Adaptively Secure Broadcast Encryption with Short Ciphertexts

Behzad Malek and Ali Miri

{bmalek,samiri}@site.uottawa.ca
School of Information Technology and Engineering
University of Ottawa, Ottawa, ON, Canada

Abstract. We propose an adaptively secure broadcast encryption scheme with short ciphertexts. That is the size of broadcast encryption message to share secret keys between all members of the broadcast group is fixed, regardless of the size of the broadcast group. In our proposed scheme, members can join and leave the group without requiring any change to public parameters of the system or private keys of existing members. Our construction has a twofold improvement over best previously known broadcast encryption schemes. First, we propose a scheme that immediately yields adaptive security in the CCA model without any (sub-linear) increase in the size of ciphertexts or use of a random oracle. Secondly, the security model in our system includes decryption queries for any member, even including the members in the challenge set. This a more secure model, as it is closer to the adversary in real world.

1 Introduction

Sharing secrets or common keys between two parties has been solved by public key cryptosystems in the 70's, but efficiently extending the secret sharing beyond two parties is still a challenge. This is a problem that we are facing in so many applications today, such as video conferencing, Digital Rights Management systems, secure IP multicasting, or any other applications in general where different parties belong to the same group and share the same credentials. They are involved in the same activity and need to securely communicate with each other. A naive solution would be to unicast the same key to all the members of the group using conventional public key cryptosystems. This solution is not efficient as the manager of the group has to store and communicate as many messages as the number of group members. Moreover, every time there is any change in the group membership, a new set of messages must be communicated with the new members of the same group. We are looking for non-trivial solutions that can share a secret key to dynamic members of group with minimum computations and communications needed. As it can be seen from Fig. 1, establishing a secure communication channel among a very disperse and dynamic set of users is a very challenging tasks. This is mainly because traditional access control systems, e.g.

Role Based Access Control (RBAC), often give full access to users depending on the roles they take. This results in an all-or-nothing authorization, which does not have the capability to set permissions selectively per user. Therefore, we want a solution that can efficiently share access keys to all the users of a specific resource (document). The solution has to be adapted to any arbitrary set of users over any domain that can change dynamically.

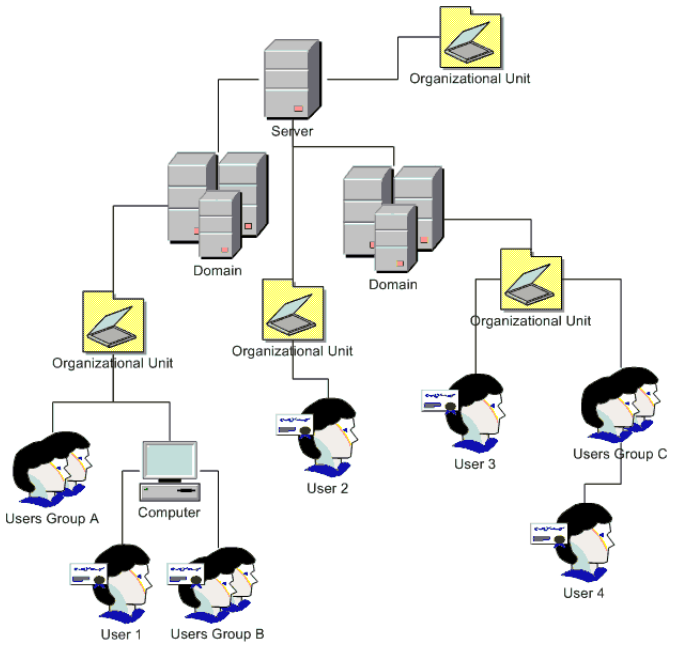

Fig. 1. Sharing secrets among a dynamic group of users

In Fig. 1, we have supposed that a secret message needs to be communication with Users 1,2,3,4 and all the users in Users Group B. Note that the users might have different roles or be located at different domains. Nevertheless, the server in Fig. 1 must be able to send a short broadcast message in the network that can only be accessed by the privileged users. The broadcast message has to be encrypted to be secure and short to avoid flooding the network. Users should have different public/private keys pais to be able to securely communicate with other members. The broadcast message needs to be generated using each user's public key, while any colluding subset of users in the network would not be able to access the broadcast message if they are excluded from the set of privileged recipients.

The cryptographic solutions in this area are usually categorized into Group Key Multicasting (GKM) and *broadcast encryption* schemes. In GKM, distribu-

tion of keys to an exclusive set of members, while the group membership changes dynamically, is the main challenge. Emphasis is mostly on the security of the current group members against adversaries over time in the past, present and future. Members leave and join the group depending on the credentials they receive from the group manager. Small subgroups can form within a group, and members should be able to securely communicate with any subgroup of members. We are looking for ways to avoid or reduce re-sending keys to members after every time there is a change in the group membership. We refer to the group manager as administrator (`Admin`) who is responsible for managing the group and distributing keys to group members. The `Admin` is usually interested in minimizing the communication/computation overheads imposed on group members, in order to share a secret among themselves.

In broadcast encryption, security of the system and efficiency of sharing new sessions keys are the main concern. It involves sharing a key between multiple (more than two) parties in a group, where they join or leave the group at any time. There are many security requirements in sharing new secret keys between members of a group that dynamically change their membership. Generally, new members of the group should not be able to access previous messages, or the revoked members should be not be able to access the current messages communicated in the group. This usually requires updating or distributing new keys to group members after any membership changes, affecting keys of all other members and incurring extra communication overheads. This is referred to as 1-affects-all effect. Regarding the communication bandwidth as a limited natural resource, we would like to design a protocol that trades off computations or storage complexity for minimal communication overheads.

Contributions: Within the given requirements, we propose the first broadcast encryption scheme that is secure in a fully adaptive model. The proposed scheme is proved secure in a formal model based on a known complexity assumption, while there is no need for using random oracle (hash functions) in the proof. The security model under which the security proof is provided is a very strong model better simulating the adversary in the real world.

Outline: In this work, we first review some of the related work in Section 2. We give the preliminaries to understand this work in Section 3, which is followed by Section 4, where the main protocol is given. Security of the proposed protocol in its underlying attack model is formally proved in Section 5. Performance of our broadcast encryption scheme is discussed in Section 6. Finally, conclusions and future work are given in Section 7.

2 Related Work

There exist various GKM or broadcast encryption schemes that can be used for secure group communication. For a comprehensive survey of most recent group key multicast protocols, the reader can refer to [8]. In a centralized scheme, there is an authority that manages the entire group membership and is responsible to share the corresponding keys to members of the group. With the exception of

Secure Lock protocol [10], any change in the group memberships often requires private keys of all other members to be changed creating extra communication overheads in the group. In the Secure Lock protocol [10], this is avoid, but at the expense of broadcast ciphertexts that are linear in the size of the group.

Our work is based on a centralized authority, which we refer to as `Admin` in this work. The `Admin` is responsible for setting up and supervising the group. The `Admin` can also be viewed as a central authority in our protocol. In Table 1, we compare the performances of protocols that we find relevant to our scheme.

Table 1. Comparison of centralized group key sharing protocols

Scheme	1-Not-All	Communication	Computation	Storage	Update
Secure Lock[10]	✓	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Burmerster et al.[7]	-	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$
Perrig et al.[18]	-	$\mathcal{O}(\log_2 n)$	$\mathcal{O}(\log_2 n)$	$\mathcal{O}(\log_2 n)$	$\mathcal{O}(\log_2 n)$
Barua et al.[2]	-	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(\log_3 n)$
Choi et al.[11]	-	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$
Boneh et al.[6]	✓	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$
Gentry & Waters[13]	✓	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(1)$
Our scheme	✓	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

	Legend
Size of the broadcast group	n
Does not have the 1-affects-all effect	1-Not-All
Communication complexity of broadcast messages	Communication
Computation complexity to send broadcast messages	Computation
Storage complexity to store private/public keys	Storage
Size of update messages	Update

An attempt to efficiently scale group key sharing from two (as in public key cryptography) to many entities is found in [15], where bilinear pairings (Weil or Tate) can be used for a one-round tripartite key agreement. The protocol is extended to large groups consisting of n members in [2]. It uses ternary trees and therefore requires $\mathcal{O}(\log_3 n)$ communication rounds for n members. Choi et al. [11] propose a constant-round GKM protocol from bilinear pairings, based on Burmester and Desmedt's scheme [7]. A good collection of identity-based GKM protocols from pairings is gathered in [9, 16] and their security is compared to each other. The reader interested in efficiency comparison can be referred to [1, 19]. The key sharing scheme has to be collusion resistant; the excluded members must not be able to cooperate together, in order to obtain the session key or private keys of other members in the group. There are some fully collusion resistant systems [12, 14, 17] in which the ciphertext grows linearly with the number of privileged receivers in the broadcast group. Boneh et al. [6] have proposed a

system that has short ciphertexts, i.e. the size of the broadcast message is fixed and does not change with the size of the broadcast group.

Their collusion resistant broadcast encryption is designed in a static security model, where the adversary must commit to the set of identities S' that it will attack before the setup phase, i.e. before seeing public parameters (PK) of the broadcast group. The adversary is prohibited from querying private keys for any $i \in [1, n] \setminus S'$. Gentry and Waters [13] propose a semi-static security model, where the adversary similarly must commit to a set S' of indices before the setup phase, but can query an arbitrary subset of S'. Note that $S' \cup S^* = [1, n]$ and the adversary cannot query the private keys for any $i \in S^*$. It is claimed that "a semi-static adversary is weaker than an adaptive adversary, but it is stronger than a static adversary, in that its choice of which subset of S' to attack can be adaptive"[13]. On the contrary, no initial commitment is required from the adversary in an adaptively secure system. The adversary is allowed to see the public parameters (including the identity of members) and can then ask for private keys of members that it wishes to attack. Attacker in reality is an adaptive attacker that can collude with any subset of group members and ask for decryption queries of a ciphertext at any time. We address this attacker and present a more general definition of adaptive security.

3 Preliminaries

In this section, we begin by defining a broadcast encryption system. Then, we present the adaptive security model devised for a broadcast encryption system. Later in this section, we introduce the cryptographic primitives used as the basis of this work.

3.1 Broadcast Encryption Systems

We need to begin by formally defining a public-key broadcast encryption system. We use the formal definition of Gentry and Waters [13] and propose our broadcast encryption protocol in the same framework. The broadcast encryption scheme is comprised of four algorithms: ***Setup***(λ, n), ***KeyGen***(i, SK), ***Encrypt***(S, PK) and ***Decrypt***$(S, i, D_i, \mathrm{Hdr}, PK)$.

Setup(λ, n) Takes as input the number of receivers (n) and the security parameter λ of a broadcast recipient group. It outputs a public/secret key pair $\langle PK, SK \rangle$ belonging to the `Admin`. Note that SK is called a secret key, as the security of the given broadcast encryption system depends on it.

KeyGen(i, SK) Takes an input an index $i \in \{1, \cdots, n\}$ and the secret key SK. It outputs a private key d_i for the i-th member (identity). We will see later that this private key is used for decryption in the ***Decrypt()*** algorithm.

Encrypt(S, PK) Takes as input a subset $S \subseteq [1, n]$ and a public key PK. If the size of the subset ($|S|$) satisfies $|S| \leq l$, it outputs a pair $\langle \text{Hdr}, K \rangle$, where Hdr is called the header and $K \in \mathcal{K}$ is a message encryption key. We will show later that K is used as the encryption key and Hdr contains data for intended recipients to find the encryption key. The broadcast to members in S consists of $\langle S, \text{Hdr} \rangle$.

Decrypt$(S, i, D_i, \text{Hdr}, PK)$ Takes as input a subset $S \subseteq [1, n]$, an index $i \in \{1, \cdots, n\}$, private key D_i corresponding to i, a header Hdr for the given S and the public key PK. If $|S| \leq l$ and $i \in S$, then the algorithm outputs the message encryption key $K \in \mathcal{K}$.

3.2 Security Model

The security of our protocol is defined in the chosen ciphertext security against an adaptive adversary. Adaptive security in broadcast encryption is defined using the following game between an attack algorithm $\mathcal{A}$ and a challenger. Both $\mathcal{A}$ and the challenger are given n and λ in the beginning. The adversary is adaptive; that is it does not to commit to a subset of members before seeing the public parameters PK. We improve the security model of Gentry and Waters' [13] by adding the decryption query round in which the adversary, in addition to adaptively obtaining the private keys of the attack set, can send decryption queries to the challenger for the challenge set. We believe that this is a stronger security model, as it is captures a wider range of attacks and is therefore closer to the adversary in real world. Our model is defined as follows:

Setup. The challenger runs ***Setup***(λ, n) to obtain a public key PK, which is then revealed to the adversary.

Key Query Phase. Algorithm $\mathcal{A}$ adaptively issues private key (D_i) queries for any set of indices $S' \subset [1, n]$.

Challenge. The challenge set is specified as $S^* = [1, n] \setminus S'$. Note that for all private keys (D_i) of member i queried in the **Key Query Phase**, we have $i \notin S^*$. The challenger then runs ***Encrypt***(S^*, PK) and outputs $\langle \text{Hdr}^*, K \rangle$. The challenger secretly picks a random $Z \xleftarrow{R} \in \mathcal{K}$. It then sets $b \xleftarrow{R} \in \{0, 1\}$ and returns $\langle \text{Hdr}^*, K^* \rangle$ to the adversary, where $K^* \leftarrow K$ if $b = 0$, otherwise $K^* \leftarrow Z$.

Decryption Query Phase. The adversary issues adaptively decryption queries $q_1, \cdots, q_D$, where a decryption query consists of the triple (i, S, Hdr) for any $S \subset [1, n]$ including $S \subset S^*$. The only constraint is that $\text{Hdr} \neq \text{Hdr}^*$. The challenger responds with ***Decrypt***$(S, i, D_i, \text{Hdr}, PK)$.

Guess. The adversary uses algorithm $\mathcal{A}$ to output its guess $b' \in \{0, 1\}$ for b and wins the game if $b' = b$.

We refer to the game described above as the adaptive Chosen Ciphertext Attack (CCA). Using an algorithm $\mathcal{A}$ to break the broadcast encryption system (BE) with parameters (λ, n), i.e. to guess the correct value of b, the adversary's advantage is defined as follows:

$$Adv_{\mathcal{A},\mathrm{BE}}(\lambda, n) = |Pr[b' = b] - \frac{1}{2}|,$$

where b' is the algorithm $\mathcal{A}$'s guess of b.

Definition 1 *A broadcast encryption system* BE *is adaptively* $(\mathrm{negl}(\lambda), n, q_D)$ *CCA secure if for all polynomial-time algorithms* $\mathcal{A}$ *that make a total of* q_D *decryption queries, we have* $Adv_{\mathcal{A},\mathrm{BE}}(\lambda, n) = \mathrm{negl}(\lambda)$. *The adversary has a negligible advantage if* $\mathrm{negl}(\lambda)$ *can be made smaller than* $\frac{1}{\mathrm{poly}(\lambda)}$ *for any arbitrary polynomial* $\mathrm{poly}()$.

3.3 Bilinear Maps

We make extensive use of *bilinear maps* at the core of our proposed schemes, but first we have to define it.

Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order p, and let g be a generator of $\mathbb{G}$. A bilinear map is an efficiently computable function from $\mathbb{G} \times \mathbb{G}$ onto $\mathbb{G}_T$, such that it has the following properties:

1. *Bilinearity*: For all $g, g', h, h' \in \mathbb{G}$,

$$\begin{aligned} &e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T, \\ &e(gg', h) = e(g, h)e(g', h), \\ &e(g, hh') = e(g, h)e(g, h') \end{aligned}$$

   Note that $e(\cdot, \cdot)$ is symmetric, that is $e(g^a, g^b) = e(g^b, g^a) = e(g, g)^{ab} \quad \forall a, b$.

2. *Non-degeneracy*: If $e(g, h) = 1$ for all $h \in \mathbb{G}$, then $g = I$ (identity).

Weil pairing and Tate pairing are two implementations of an efficient bilinear map over elliptic curve groups useful for cryptography. For a more detailed discussion on bilinear maps and pairings, we refer the reader to [3]. Bilinear maps for cryptography has to have certain complexities to be used in cryptographic algorithms. This is explained in the following section.

3.4 Complexity Assumptions

The security of our schemes is based on a complexity assumption that has appeared in prior art [4–6, 13]. The many complexity assumptions found in literature have slightly different settings, but they are all related to the difficulty of solving Discrete Logarithm Problem (DLP) over large algebraic group.

Our main construction, which is given later in Section 4, is based on a narrower variant of the DLP assumption, referred to as the Bilinear Diffie-Hellman Exponent (BDHE)-Sum assumption. This is the same complexity assumption that has been used in Gentry and Waters' adaptive scheme [13]. We have simplified the definition to relate directly to our security proof.

Definition 2 (BDHE-Sum Assumption (for n):) *As usual, let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order p with bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, g a generator for $\mathbb{G}$ and $\alpha, s \xrightarrow{R} \mathbb{Z}^*{}_p$. Set $S = [-2n, 2n]$. Given $\{y_i = g^{\alpha^i} : i \in S\}$, compute $e(g,g)^{\alpha^{4n+1}}$, without knowing α.*

The decision assumption variant of above assumptions is stated as follows:

Definition 3 *Let $\widehat{y}_{g,\alpha,n} = \{y_i = g^{\alpha^i}\ \forall i \in S\}$. An algorithm $\mathcal{B}$ that outputs $b \in \{0,1\}$ has advantage ϵ in solving the decision BDHE(-Sum) for n in $\mathbb{G}$ if*

$$Pr\left[\mathcal{B}(g, \widehat{y}_{g,\alpha,n}, e(g,g)^{\alpha^{4n+1}}) = 0\right] - Pr\left[\mathcal{B}(g, \widehat{y}_{g,\alpha,n}, Z) = 0\right] \geq \epsilon,$$

where the probability is over the random choice of the generator $g \in \mathbb{G}$, the random choice of $\alpha \in \mathbb{Z}^*{}_p$, the random choice of $Z \in \mathbb{G}_T$, and the random bits consumed by $\mathcal{B}$. We refer to the distribution on the left as Pr_{BDHE} and the distribution on the right as R_{BDHE}.

We say that the (decision) (ϵ, n)-BDHE-Sum assumption holds in $\mathbb{G}$ if no polynomial-time algorithm has advantage of at least ϵ in solving the (decision) BDHE-Sum problem for n in $\mathbb{G}_T$.

4 Adaptively Secure BE Construction

By increasing the number of private keys, we manage to derive a fully adaptively secure BE scheme with short ciphertexts, which is referred to as BE_A. As before, we denote the maximum number of members in the multicast group by n. Our BE_A scheme is given in the following:

Setup(λ, n) Run $\langle \mathbb{G}, \mathbb{G}_T, e \xleftarrow{R} \mathit{GroupGen}(\lambda, n)\rangle$. Set $\alpha \xleftarrow{R} \mathbb{Z}^*{}_p$, the generator $g \in \mathbb{G}$, identity values $x_1, \cdots, x_n \xleftarrow{R} \mathbb{G}^n$ and two secret values $A, \gamma \xleftarrow{R} \mathbb{Z}^*{}_p$. Set PK to include a description of $\mathbb{G}, \mathbb{G}_T, e$, $\{x_1, \cdots, x_n\}$, $\{g^{A\alpha^i}, \forall i \in [0,n]\}$ and $e(g,g)^{A\alpha^{2n+1}}$ as the session key. The group's secret SK is set as $\langle A, \gamma, \alpha\rangle$, which is known by `Admin` only. Output $\langle PK, SK\rangle$.

KeyGen(i, SK) Pick $r_i \xleftarrow{R} \mathbb{Z}^*{}_p$. Release to member i the following set of private keys $D_i \leftarrow \{r_i, d_{i,j}, T_{i,j}\}$, where:

$$d_{i,j} = g^{\alpha^n \frac{\gamma\alpha^2 r_j}{\alpha - x_i}}\ \forall\ j \in [1,n] \text{ and } j \neq i$$

$$T_{i,j} = g^{\frac{A\alpha^j}{(\gamma r_j)}}\ \forall\ j \in [0,n]$$

We emphasis that r_i and $d_{i,j}$ values are used for decryption and $T_{i,j}$ values are used to create the broadcast encryption message.

Encrypt(S, i, D_i, PK) The set S includes the index of members for which the message will be sent, as well as the index of the encrypting member i. Pick $t \xleftarrow{R} \mathbb{Z}^*{}_p$ and set $\text{Hdr} \leftarrow \langle C_1, C_2 \rangle$, where $C_1 \leftarrow g^{t\alpha^n}$ and

$$C_2 \leftarrow g^{tA\alpha^{n-|S|}(\gamma r_i)^{-1}\prod_{j\in S}(\alpha - x_j)},$$

where i is the sender's index i. Let's denote $p(\alpha) = \alpha^{n-|S|}\prod_{j\in S}(\alpha - x_j)$. It should be clear that $p(\alpha)$ is a polynomial of degree n, and therefore $g^{A(\gamma r_i)^{-1}p(\alpha)}$ can be readily calculated from $T_{i,j}$-s and x_j-s. The session key (K) is set as follows:

$$K \leftarrow e(g,g)^{tA\alpha^{2n+1}}.$$

Output $\langle \text{Hdr}, K \rangle$.

Decrypt$(S, i, D_i, \text{Hdr}, PK)$ If $i \in S$, find the sender's index (j) and then expand Hdr to $\langle C_1, C_2 \rangle$ and output

$$K \leftarrow e(C_1, g^{Ap_i(\alpha)})e(C_2, d_{i,j}).$$

where $p_i(\alpha) = \alpha^{n+1} - \frac{\alpha^2 p(\alpha)}{\alpha - x_i}$. Note that for $i \in S$, p_i is a polynomial of degree n and therefore $g^{Ap_i(\alpha)}$ can be easily calculated from $g^{A\alpha^j}$-s and x_j-s, when $i \in S$.

Correctness: Let's check that decryption recovers the correct value of K. Recall that the secret key of a member is set as $d_{i,j} = g^{\alpha^n \frac{\gamma\alpha^2 r_j}{\alpha - x_i}}$. Then, we have the following proceedings:

$$\begin{aligned}
e(C_1, g^{Ap_i(\alpha)})e(C_2, d_{i,j}) &= e(g^{t\alpha^n}, g^{Ap_i(\alpha)}) \\
&\times e(g^{tA(\gamma r_j)^{-1}p(\alpha)}, g^{\alpha^n \frac{\gamma\alpha^2 r_j}{\alpha - x_i}}) \\
&= e(g,g)^{tA\alpha^n p_i(\alpha)} e(g,g)^{tA\alpha^n \frac{\alpha^2 p(\alpha)}{\alpha - x_i}} \\
&= e(g,g)^{tA\alpha^n(\alpha^{n+1} - \frac{\alpha^2 p(\alpha)}{\alpha - x_i} + \frac{\alpha^2 p(\alpha)}{\alpha - x_i})} \\
&= e(g,g)^{tA\alpha^{2n+1}}
\end{aligned}$$

as required.

Authentication: Let SymEnc and SymDec be symmetric encryption and decryption, respectively. Let M be a random verification message to be broadcast to the set S, and let $C_M \xleftarrow{R} \text{SymEnc}(K, M)$ be the randomized encryption of M under the session key K, which is broadcast to the set S. The broadcast to members in S consists of $\langle S, \text{Hdr}, M, C_M \rangle$. The privileged receiver, a member in the set S, can easily verify the sender of the broadcast message as follows:

First, member i (if $i \in S$) retrieves the session key K from Hdr by using the decryption key ($d_{i,j}$) corresponding to the sender (member j) of the message. Then, member i checks if $M = \text{SymDec}(K, C_M)$. If it passes, it verifies the sender, otherwise, it refuses the authentication.

5 Security Analysis

In this section, we prove the fully security of the proposed BE_A scheme in the CCA model.

Theorem 1 *Let $\mathbb{G}_T$ be a bilinear group of prime order p. For any positive integers $n, 2n (s.t.\ n < 2n < p)$ our n-broadcast encryption system is $(\text{negl}(\lambda), 2n)$ adaptively secure assuming the decision $(\text{negl}(\lambda), 2n)$-BDHE-Sum assumption holds in $\mathbb{G}_T$.*

Table 2. Comparison of identity-based broadcast encryption schemes

Scheme	Ciphertext	Private Keys	Public Parameters	Security	ROM
Gentry & Waters[13]	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	Semi-Static	–
Gentry & Waters[13]	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	Adaptive	✓
Gentry & Waters[13]	$\mathcal{O}(\sqrt{n})$	$\mathcal{O}(1)$	$\mathcal{O}(n)$	Adaptive	–
Our scheme	$\mathcal{O}(1)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$	Adaptive	–

Proof. As usual, we start by the assumption that there is an algorithm $\mathcal{A}$ with advantage $\epsilon = \text{negl}(\lambda)$ in attacking the proposed BE_A scheme. If this is true, we prove that $\mathcal{A}$ can be used to solve the decision n-BDHE-Sum in $\mathbb{G}$. We build a simulation machine $\mathcal{B}$ that receives an instance of the decision n-BDHE-Sum problem. This is comprised of $Z \in \mathbb{G}$ and the set of $\{g^{a_i} : i \in [-2n, 2n]\}$.

No Commit. It has to be emphasized that the adversary's algorithm $\mathcal{A}$ does not commit to a predetermined set of indices S^* to attack, before seeing the public parameters of the scheme. Without loss of generality, we assume $|S^*| = 2$. This implies that the adversary can attack and retrieve the private keys of all members, except two members that will be used in the challenge round. One un-attacked member is used to generate a broadcast message (Hdr^*) only for the other un-attacked member. Otherwise, it is obvious that the adversary will be able to recover the session key, as it already has the private key of other members.

Setup. $\mathcal{B}$ disguises the parameters of the challenge problem into parameters of the proposed BE_A scheme. $\mathcal{B}$ puts $\alpha = a$ and using the challenge instance, it sets the public parameters as: $g^{\alpha^i} = g^{a^i}$ for $i \in [0, 2n]$. For the public identity's of members (x_i), $\mathcal{B}$ picks $x_i \xleftarrow{R} \mathbb{Z}^*{}_p$ and publishes PK as $\mathbb{G}, \mathbb{G}_T, e, \{x_1, \cdots, x_n\}$,

and $\{g^{\alpha^i}, \forall i \in [0, 2n]\}$. Then, $\mathcal{B}$ picks a random $y_0 \xleftarrow{R} \mathbb{Z}^*_p$ and sets $\gamma = y_0 a^{-2n}$. The session key, as before, is the following $K = e(g,g)^{\alpha^{2n+1}} = e(g,g)^{a^{2n+1}}$. The secret key SK includes the set $\{\alpha = a, \gamma = y_0 a^{-2n}\}$.

Private Keys Query. Algorithm $\mathcal{A}$ queries private keys $(d_{i,j})$ for any arbitrary subset S' of $[1,n]$, where $\max(|S'|) = n-2$. We make the restriction that $\mathcal{A}$ has to query all private keys at once for members it will attack. Let's denote the set of un-attacked members by S^*. Thus, we have $S^* \cup S' = [1,n]$. We have assumed that $|S^*| = 2$, so the notation $j \oplus 1$ refers to the other index in S^* than j in our notes.

Having known the set of attacked members (S') and the set of un-attacked members (S^*), $\mathcal{B}$ picks a random $b_j \xleftarrow{R} \mathbb{Z}^*_p$ and sets $B_j = b_j a^n$ for $j \in [1,n]$, but it sets $B_j = b_j a^{n-1}(a - x_{j\oplus 1})$ and $B_{j\oplus 1} = b_{j\oplus 1} a^{n-1}(a - x_j)$. For $i \in S'$, $\mathcal{B}$ responds to the query for member i's private keys as follows: it returns $D_i \leftarrow \langle r_i, d_{i,j}, T_{i,j}\rangle$, recall that in the real protocol, we have r_i as a random value, $d_{i,j} = g^{\gamma \alpha^n B_j \frac{\alpha^2 - r_j}{\alpha - x_i}}$ and $T_{i,j} = g^{\frac{\alpha^j}{\gamma B_j}}$. Accordingly, $\mathcal{B}$ returns $r_i = x_i^2$ and $d_{i,j} = g^{y_0 b_j (a + x_i)}$ for $j \in S'$, but for $j \in S^*$, $\mathcal{B}$ returns $d_{i,j} = g^{y_0 b_j a^{-1}(a - x_{j\oplus 1})(a + x_i)}$ and $d_{i,j\oplus 1} = g^{y_0 b_j a^{-1}(a - x_j)(a + x_i)}$. Finally, $T_{i,j} = g^{(y_0 b_i)^{-1} a^{n+j}}$ for all $j \in [0,n]$. Note that all these parameters can be readily calculated from the BDHE-Sum instance. It is easy to check that the private keys are matched with the parameters in the real protocol, and they are valid. The set of indices in the un-attacked set S^*, which have not been queried, will be used in the challenge phase.

Challenge. In the challenge phase, $\mathcal{B}$ creates a broadcast encryption message for $i \in S^*$. It sets $S \subset S^*$ and generates $C_1^* = g^t$ and $C_2^* = g^{t(\gamma b_i)^{-1} p(\alpha)}$, where $i \in S^*$. Let's suppose that the broadcast message is generated by member i, $i \in S^*$ and $S = \{i \oplus 1\}$. The challenge is then calculated as follows: pick a random $t_0 \xleftarrow{R} \mathbb{Z}^*_p$, and set $t = t_0 a^{2n}$. Then, calculate the broadcast $\mathrm{Hdr}^* \leftarrow \langle C_1^*, C_2^*\rangle$, where $C_1^* = g^{t_0 a^{2n}}$ and $C_2^* = g^{t_0 a^{2n} (y_0 a^{-2n} B_i)^{-1} a^{n-1}(a - x_{i\oplus 1})}$, which yields $g^{t_0 (y_0 b_i)^{-1}}$ for $B_i = b_i a^{n-1}(a - x_{i\oplus 1})$. Note that both C_1^* and C_2^* can be directly calculated from the BDHE-Sum instance: $C_1^* = (g^{a^{2n}})^{t_0}$ and $C_2^* = (g)^{t_0 (y_0 b_i)^{-1}}$. It should be noted that $\mathrm{Hdr}^* = \{C_1^*, C_2^*\}$ as set above is a valid ciphertext for indices in $S \subset S^*$ and the session key $K = e(g,g)^{t_0 a^{2n} a^{2n+1}} = e(g,g)^{t_0 a^{4n+1}}$. $\mathcal{B}$ outputs Hdr^* and $K^* = Z^{t_0}$, where Z is the challenge from the BDHE-Sum instance, as the new challenge to $\mathcal{A}$.

Decryption Query. we further allow $\mathcal{A}$ to use the set of private keys it received to generate a broadcast message for any $i \in [1,n]$ and even for $i \in S^*$. $\mathcal{B}$ is able to derive the private keys $d_{i,j}$ for $i \in S^*$, in the same way as in the **Private Keys Query** phase, except $T_{i,j}$ values for $i \in S^*$. Nevertheless, this does not stop $\mathcal{B}$ from returning correct decryptions, since only r_i and $d_{i,j}$ are need for decrypting and $T_{i,j}$ values are used to create the broadcast encryption. By setting $r_i = x_i^2$ and $d_{i,j} = g^{y_0 b_j (a + x_i)}$ for any $i \in S^*$ and $j \in S'$, $\mathcal{B}$ is able to

respond correctly to the decryption queries as in the real application.

Guess. The algorithm $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$ and wins the game if $b' = b$. $\mathcal{B}$ sends b' to the challenger in the proposed BE_A scheme to solve the BDHE-Sum instance. From $\mathcal{A}$'s perspective, $\mathcal{B}$'s simulation has almost the same distribution as the adaptive security model defined earlier in Section 3. The public and private keys are appropriately distributed, since x_i-s and therefore r_i-s are uniformly random. When $b = 0$ in the adaptive game, $\langle \text{Hdr}^*, K^* \rangle$ is generated according to the same distribution as in the real application with a valid session key $K^* = e(g,g)^{ta^{2n+1}}$, where $t = t_0 a^{2n}$. Thus, the challenge is a valid ciphertext under the randomness of t_0. From $\mathcal{B}$'s simulation, when $b = 0$, we can easily find the solution to BDHE-Sum problem, by outputting $Z = {K^*}^{1/t_0} = e(g,g)^{a^{4n+1}}$.

When $b = 1$ in the adaptive game, $\langle \text{Hdr}^*, K^* \rangle$ is generated with K^* being replaced by a random key. This distribution is identical to that of $\mathcal{B}$'s simulation, where Hdr^* is valid for randomness of t_0, but $K^* \xleftarrow{R} \mathbb{G}_T$ is a uniformly random element of $\mathbb{G}_T$. From this, we see that $\mathcal{B}$'s advantage in deciding n-BDHE-Sum problem is precisely $\mathcal{A}$'s advantage against the BE_A scheme.

6 Performance Analysis

In this section, we analyze the overheads of increased security over previously known schemes. We have seen that adding new members causes extra communications and updates of private keys of existing members. Nevertheless, the maximum size of the group n in the proposed scheme is bounded by size of the pairing group, i.e. $n < |\mathbb{G}|$. For current levels of security, it is suggested to have $|\mathbb{G}| \geq 2^{160}$ [3]. More efficiency and security analysis of pairing groups are given in [9]. This implies that the size of the underlying pairing groups increases linearly with the maximum size of the broadcast group. This is a challenge that we leave for future work.

We proposed a fully adaptive BE with short ciphertexts to meet the prime objective of this work. The design is aimed at constructing a fully secure BE scheme with $\mathcal{O}(1)$ session key messages (Hdr), regardless of the size of the broadcast group. In Table 2, we have compared our work to Gentry and Waters' BE scheme and its variants [13]. It can be seen from Table 2 that the increased security in our scheme has led to an increase in the size of private key parameters of each member. Moreover, the adaptive security in Gentry and Waters' work [13] with short ciphertext $\mathcal{O}(1)$ is based on Random Oracle Model (ROM) and hash functions.

It should be noted that the adaptive security of the proposed scheme is obtained in an attack model that is stronger that the one appeared in [13]. In our security model, we allow the adversary not only to query private keys of all members under attack, but also to query decryption of messages intended for all other members – the ones it did not query for private keys. We believe that this a better security model as it is closer to the adversary in real world.

Table 3. Satisfying requirements for broadcast encryption schemes

Requirement	Boneh et al.[6]	Gentry & Waters[13]	Our scheme
Efficiency in communication	✓	✓	✓
Efficiency in computation	✓	♯	✓
Collusion resistant	✓	✓	✓
Scalability	✓	♯	♯
No 1-affects-all	✓	✓	♯
Ephemeral secrecy	✓	✓	✓
Long-term secrecy	✓	✓	✓
Forward secrecy	✓	✓	✓
Group forward secrecy	✓	✓	✓
Backward secrecy	✓	✓	✓
Provable security	✓	✓	✓
Symmetry	✓	✓	✓
Authentication	–	–	✓

	Legend
Fully satisfies the requirements	✓
Partially satisfies the requirements	♯
Does not satisfies the requirements	–

6.1 Group Operations

In the proposed BE_A scheme also, removing from the group membership do not affect existing members. Excluding a member simply means not including the index of excluded member in calculate the ciphertexts (Hdr). If a member is permanently removed from the group, only the identity parameter (x_k) of the excluded member is removed and no further changes to private keys of members are required. Keys of members remain the same as the group membership changes without compromising security of the BE_A protocol. It should be added that member removal is performed at no extra communication or computation cost to group members.

Removal. membership removal is inherent in the BE_A scheme. Excluding a member is as usual and is performed by not including the index of the excluded member in S. Thus, no extra communication or computation overhead is incurred for removing a member.

Addition. adding a member is authorized by `Admin`. If the group's maximum capacity, set by n, is not reached, any new member i' can be added to the group. The `Admin` simply generates a new set of private keys $\{d_{i',j}, T_{i',j}\}$ for the new member and publishes its identity ($x_{i'}$) to the group. Unlike the semi-static scheme of Boneh et al. [6] that did not require further key update at the existing members, we have to send the new decryption key $d_{i,i'}$ for the existing member i to be able to communicate with new member i'. We further compare

the proposed scheme in Table 3 with regards to the general requirements that we provided earlier in this work.

As listed in Table 3, the proposed broadcast encryption scheme satisfies almost all the requirements. As said earlier, the maximum size of the group is limited by the order of bilinear underlying group. Moreover, adding new members requires the `Admin` to broadcast new decryption keys to all current members of the group, where as removing any member does not require any change to keys of existing members. Our proposed scheme is the only scheme that provides authentication of the sender without any increase in the size of the broadcast encryption message.

7 Conclusions and Future Work

In this work, we have proposed a broadcast encryption scheme based on pairings over elliptic curves. We have proposed the first adaptively secure broadcast encryption scheme with short ciphertexts without the use of random oracles. The security model of the proposed broadcast encryption scheme is a strong model that simulates the adversary in the real world as closely as possible. In our model, we prove adaptive security of the proposed scheme, where the adversary not only receives private keys of its selected members, but also it can send decryption queries for members in the challenge set. We believe this is a better security model, as it captures a wider range of attacks in practice.

Increase security has resulted in an increase in the size of private keys. However, this increase yields an authentication service with no extra overheads. In our proposed scheme, the sender of a broadcast message can be verified to the members of the broadcast group. As a side effect, if new members are added to the broadcast group, verification keys of the added members need to be communicated with existing group members.

We have also shown that the communication and computation needed for the protocol to actively exclude or include memberships are very minimal, i.e. with $\mathcal{O}(1)$ communication and $\mathcal{O}(n)$ computations, where n is the size of the broadcast group. The amount of storage required for each member can be trivial when compared to other protocols. Members can join or leave the group, while the security keys of other members will not be affected by the change in group memberships. The maximum number of that can join the group is limited by the underlying algebraic group structure. Thus, we believe that the protocol is suitable for small groups with limited bandwidth. Our protocol will drastically reduce the communication and computation overheads needed to establish secure key exchange between the group members.

The maximum size of the broadcast group is bounded by the size of the underlying bilinear group. In future, we intend to design adaptively secure broadcast encryption schemes that has short ciphertext and exponential size groups. It is still an open question if one could design a broadcast encryption scheme with both short ciphertexts and short private keys.

References

1. Yair Amir, Yongdae Kim, Cristina Nita-Rotaru, and Gene Tsudik. *On the performance of group key agreement protocols.* ACM Transactions on Information and System Security, 7(3):457–488, 2004.
2. Rana Barua, Ratna Dutta, and Palash Sarkar. *Extending Jouxs protocol to multi party key agreement.* In Advances in Cryptology: INDOCRYPT'03, volume 2904 of LNCS, pages 205–217. Springer-Verlag, 2003.
3. Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart, editors. *Advances in Elliptic Curve Cryptography*. Cambridge University Press, 2005.
4. Dan Boneh and Xavier Boyen. *Efficient selective-ID identity based encryption without random oracles.* In Christian Cachin and Jan Camenisch, editors, Advances in Cryptology: EUROCRYPT'04, volume 3027 of LNCS, pages 223–238. Springer, 2004.
5. Dan Boneh, Xavier Boyen, and Eu-Jin Goh. *Hierarchical Identity Based Encryption with Constant Size Ciphertext.* In Advances in Cryptology:EUROCRYPT'05, volume 3494 of LNCS, pages 440–456. Springer-Verlag, 2005. Available at `http://www.cs.stanford.edu/~xb/eurocrypt05a/`.
6. Dan Boneh, Craig Gentry, and Brent Waters. *Collusion Resistant Broadast Encryption with short ciphrertexts and private keys.* In Advance in Cryptology: CRYPTO'05, LNCS, pages 258–275, 2005.
7. Mike Burmester and Yvo Desmedt. *A secure and efficient conference key distribution system.* In Advances in Cryptology: Eurocrypt'94, volume 950 of LNCS, pages 275–286. Springer-Verlag, 1995.
8. Yacine Challal and Hamida Seba. *Group Key Management Protocols: A Novel Taxonomy.* International Journal of Information Theory, 2(1):105–118, 2005.
9. Liqun Chen, Michael Cheng, and Nigel P. Smart. *Identity-based Key Agreement Protocols From Pairings.* International Journal of Information Security, 6(4):213–241, 2007.
10. Guang-Huei Chiou and Wen-Tsuen Chen. *Secure Broadcasting Using the Secure Lock.* IEEE Transactions on Software Engineering, 15(8):929–934, 1989.
11. Kyu Young Choi, Jung Yeon Hwang, and Dong Hoon Lee. *Efficient ID-based Group Key Agreement with Bilinear Maps.* In Public Key Cryptography, volume 2947 of LNCS, pages 130–144. Springer-Verlag, 2004.
12. Yevgeniy Dodis and Nelly Fazio. *Public Key Broadcast Encryption for Stateless Receivers.* In Digital Rights Management: DRM'02, volume 2696 of LNCS, pages 61–80. Springer Verlag, 2002.
13. Craig Gentry and Brent Waters. *Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts).* In Advances in Cryptology: EUROCRYPT'09, volume 5479 of LNCS, pages 171–188. Springer-Verlag, 2009.
14. Dani Halevy and Adi Shamir. *The LSD Broadcast Encryption Scheme.* In Advances in Cryptology: CRYPTO'02, volume 2442 of LNCS, pages 145–161. Springer, 2002.
15. Antoine Joux. *A One Round Protocol for Tripartite Diffie-Hellman.* In the 4th International Symposium on Algorithmic Number Theory, volume 1838 of LNCS, pages 385–394. Springer-Verlag, 2000.
16. Mark Manulis. *Security-Focused Survey on Group Key Exchange Protocols.* Technical Report November, HGI Network and Data Security Group, 2006.
17. Dalit Naor, Moni Naor, and Jeffrey B. Lotspiech. *Revocation and Tracing Schemes for Stateless Receivers.* In Advances in Cryptology: CRYPTO'01, volume 2139, pages 41–62. Springer-Verlag, 2001.

18. Adrian Perrig, Dawn Song, and J. D. Tygar. *ELK, a new protocol for Efficient Large-group Key Distribution.* In IEEE Security and Priavcy Symposium, pages 247–262, 2001.
19. Sandro Rafaeli and David Hutchison. *A Survey of Key Management for Secure Group Communication.* ACM Computer Surveys, 35(3):309–329, 2003.