Adaptively Secure Broadcast Encryption with Short Ciphertexts

Behzad Malek[1] and Ali Miri[2]
(Corresponding author: Behzad Malek)

School of Information Technology and Engineering, University of Ottawa[1]
Ottawa, ON, K1N 6N5, Canada (Email: bmalek@site.uottawa.ca)
Department of Computer Science, Ryerson University[2]
Toronto, ON, M5B 2K3, Canada (Email: ali.miri@ryerson.ca)

(Received July 15, 2010; revised and accepted Sept. 19, 2010)

Abstract

We propose an adaptively secure broadcast encryption scheme with short ciphertexts, where the size of broadcast encryption message 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 previously known best broadcast encryption schemes. First, we propose a scheme that immediately yields adaptive security without any increase in the size of ciphertexts or use of a random oracle. Secondly, the proof of security in the proposed scheme is defined in a stronger security model closely simulating an adversary in real world. In our security model, the adversary can selectively query private keys of the group members after the setup and can receive decryption of broadcast encryption messages at any given time.

Keywords: Adaptive adversary, broadcast encryption, short ciphertext

1 Introduction

Sharing secrets or common keys between two parties has been solved by public key cryptosystems, but extending the secret sharing operations in an efficient manner beyond two parties remains a challenging task. Securely sharing secrets over an insecure, public channel is a problem that rises in many applications, such as satellite broadcasts, cable TV subscriptions, digital rights management systems and secure IP multicasting. Various entities belonging to the same group share the same (digital) resource that needs to be protected from intruders. Encryption is usually the preferred means of securing digital assets in an open environment. There are many efficient encryption schemes available for a full range of digital contents, including files, folders, audio and video streams. However, members of the broadcast group must have the decryption key, in order to access the protected content. A naive solution would be to share individually the decryption key with all members of the group using conventional public key of every member. Despite of being very common, this approach is not efficient as the manager of the group has to communicate the decryption key as many times as the number of group members. Moreover, every time there is a change in the group membership, a new set of messages to share the decryption key must be communicated with the members of the group. We are looking for non-trivial solutions to share a secret key among dynamic members of a group with minimum computation and communication overheads. A distributed system is displayed in Figure 1, where establishing a secure communication channel among a very disperse and dynamic set of users is desired. This is a challenging task, 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, not being flexible in granting access to resources. The solution should be able to set permissions dynamically per document and *efficiently* share the cryptographic access-keys to intended users. The solution has to be adapted to any arbitrary set of users over any domain and must allow for dynamic changes in the group memberships.

As it can be seen from Figure 1, we have supposed that a secret message (e.g. the decryption key) needs to be communicated with Users 1,2,3,4 and all the users in Users Group B. Note that the users might have different roles or might be located at different domains. Nevertheless, the server in Figure 1 should be capable of sending a short broadcast message to share the access-key with the privileged users in the network. The broadcast message has to be encrypted to be secure and should be short to avoid flooding the network. Users have different public/private keys pairs unique to them. The broadcast message has to incorporate all the receiving members, and therefore it needs to be generated using each user's public key. Obviously, any subset of colluding users in the network should not be able to access the broadcast message if the users are not included the calculation of the broadcast message.

Broadcast encryption is a cryptographic solution that involves sharing a cryptographic key between multiple (more than two) members in a group. Members can arbitrarily select any subset of members for sharing a cryptographic key. Members leave and join the group depending on the credentials they receive from the group manager at any time. We refer to the group manager as administrator (Admin), and he is responsible for managing the group and distributing keys to group members. A change in the group membership usually requires updating or distributing new keys to all of the group members including the existing members. This incurs extra communication overheads in the group and should be avoided. This is referred to as *1-affects-all* effect. Regarding the communication bandwidth as a limited natural resource, our main design goal is to reduce the communication overheads in a dynamic broadcast group, while preserving a very high level of security for all members.

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 the formal model based on a known, strong complexity assumption. Our security model under which the security proof is provided is devised to simulate the adversary in the real world as closely as possible. We achieve an adaptively secure broadcast encryption with short ciphertexts without using a random oracle (hash functions).

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 in Section 7, conclusions and future work are provided.

2 Related Work

There exist various 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]. There is usually a central authority who manages the entire multicast group and its memberships. The central authority is also responsible for distributing the public/private key pairs to members of the group. With the exception of the *Secure Lock* protocol [10], in (almost all) broadcast encryption schemes, any change in the group memberships requires changing the public/private keys of all the other members, creating huge communication overheads in the group. In the Secure Lock protocol [10], this is avoided at the expense of increasing the size of broadcast messages to the size of the entire broadcast group. In our work, we refer to the central authority as *Admin*. In Table 1, we have tersely listed some of the protocols that are relevant to our scheme and briefly compared their performances in the rest of this section.

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) are used in a key agreement protocol between three entities. The protocol is extended to large groups consisting of n members in [2], where ternary trees are used to expand to n members requiring $\mathcal{O}(\log_3 n)$ communication messages. Choi et al. [11] propose a constant-round broadcast encryption protocol with short ciphertexts from bilinear pairings, but their proposal suffers from 1-affects-all effect and any change in the group membership requires $\mathcal{O}(n)$ updates to be broadcast in the network. A good collection of identity-based protocols from pairings is listed in [9, 16] and their security is compared to each other in full details. We avoid repetition by referring the reader to [9, 16] for performance analysis and to [1, 19] for parameters and efficiency comparisons.

One of the key requirements of a secure broadcast encryption scheme is resistance against group members' collusion. In a collusion resistant broadcast encryption scheme, excluded members are not able to cooperate together to obtain the current encrypted message in the broadcast or to compromise other members in the group. There are a few collusion resistant protocols in literature [12, 14, 17], but in most cases collusion resistance has resulted in an increase in communication overheads. That is the ciphertext in these schemes grows (usually) linearly with the number of privileged members in the broadcast group. Nevertheless, Boneh et al. [6] have proposed a collusion resistant scheme 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 seeing the public parameters of the protocol (denoted by PK). This is considered a weak security model that is not reflective of the adversary in real world and does not capture all possible attacks. That is the adversary is prohibited from querying private keys of protected members with index i for any $i \in [1,n] \setminus S'$. In other words, the security assumption is based on the fact that prior to organizing the broadcast group, the adversary and the compromised members are known to the system's *Admin*.

In a more realistic simulation of the adversary, Gentry and Waters [13] propose a *semi-static* security model, where the adversary still commits to a set S' of indices before the setup phase as before, but it can query an arbitrary subset of S' after the setup. Note that $S' \cup S^* = [1,n]$ and the adversary cannot query the private keys of any $i \in S^*$. It is claimed in [13] that *"a semi-static adversary is weaker than an adaptive adversary, but it is stronger than a static adversary, in that its*

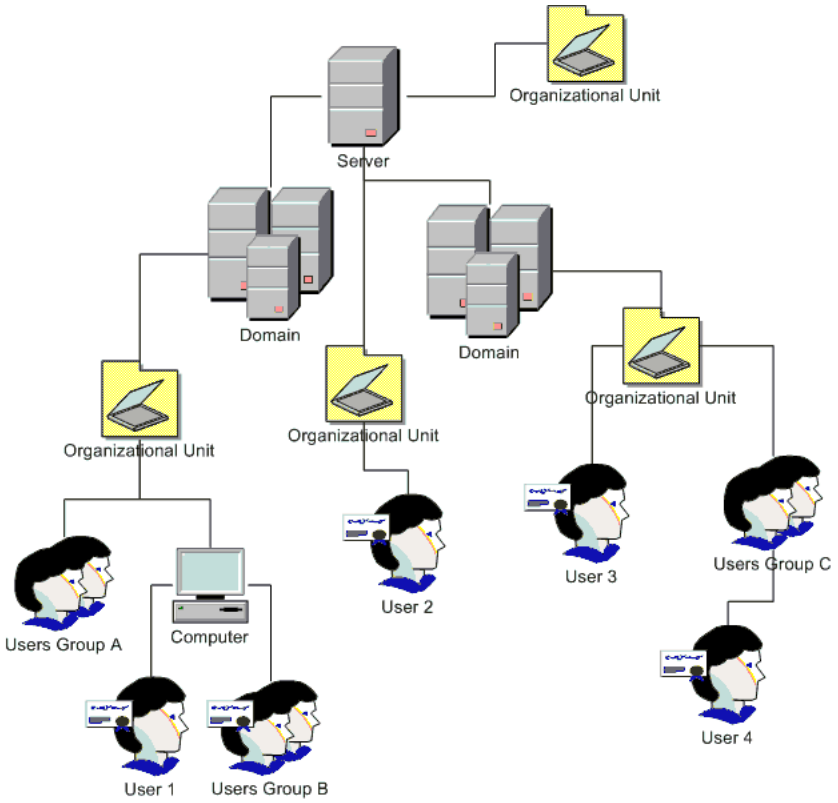

Figure 1: Sharing secrets among a dynamic group of users

choice of which subset of S' to attack can be adaptive". In an adaptively secure system, no initial commitment is required from the adversary. The adversary is allowed to see the public parameters, to take the identity of any member and to receive the private keys of any compromised member. The attacker in real world is an adaptive adversary that can take the identity of any subset of group members, receive decryption of ciphertexts at any time or exploit the private keys of an arbitrary set of compromised members. We propose a secure broadcast encryption scheme that addresses this attacker and presents a scheme in the adaptive security model.

3 Preliminaries

In this section, we begin by formally 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 that are used as the basis of our work.

3.1 Broadcast Encryption Systems

To define a broadcast encryption system, we use the formal definition of Gentry and Waters [13] for the broadcast encryption protocol. The broadcast encryption scheme is comprised of four algorithms: ***Setup***(λ, n), ***KeyGen***(i, SK), ***Encrypt***(S, PK) and *Decrypt*(S, i, D_i, Hdr, PK).

Setup(λ, n) Takes as input the number of receivers (n) and the security parameter λ of a broadcast group. Note that λ implicitly determines the maximum size of the broadcast group for which a secure broadcast encryption can be built. 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 as inputs an index $i \in \{1, \cdots, n\}$ that denotes the member's identity and the secret key SK. It outputs a private key D_i for the i-th member. We will see later that the 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 $|S| \leq n$, it outputs a pair $\langle Hdr, K\rangle$, where Hdr is called the header and $K \in \mathcal{K}$ is a message encryption key that will be shared among all the intended recipients. We will show later that K is used as the session (encryption) key and Hdr contains data to help find the encryption key by intended recipients only. The broadcast message to all members in S is fixed in size and only consists of $\langle S, Hdr\rangle$.

Decrypt(S, i, D_i, Hdr, PK) Takes as input a subset $S \subseteq [1, n]$, an index $i \in \{1, \cdots, n\}$, private key D_i correspond-

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

ing to member i, a header Hdr for the given S and the public key PK. If $|S| \leq n$ and $i \in S$, then the algorithm outputs the message encryption key $K \in \mathcal{K}$.

3.2 Security Model

Having defined the framework of the broadcast encryption scheme, we can devise the security model. The security of our protocol is defined in the *adaptive* adversary model. 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 commit to a subset of members before seeing the public parameters PK. In addition, our security model permits the adversary to adaptively obtain the private keys of the compromised members; the adversary can send decryption queries to receive the decryption of any broadcast message that was sent for members in the challenge set. The security model presented in this paper represents a stronger model, as it captures a wider range of attacks. It is therefore closer to the adversary in real world as compared to others in the literature. Our model is defined as follows:

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

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) queried in the **Key Query Phase**, we have $S' \subset [1, n]$ and $i \notin S^*$. The challenger then runs ***Encrypt(***S^*, PK***)*** and outputs $\langle Hdr^*, K \rangle$. The challenger secretly picks a random $Z \overset{R}{\longleftarrow}\in \mathcal{K}$. It then sets $b \overset{R}{\longleftarrow}\in \{0, 1\}$ and returns $\langle Hdr^*, K^* \rangle$ to the adversary, where $K^* \leftarrow K$ if $b = 0$, otherwise $K^* \leftarrow Z$.

Decryption Query Phase. The adversary issues arbitrary decryption queries $q_1, \cdots, q_D$, where a decryption query consists of the triple (i, S, Hdr) for any $S \subset [1, n]$, even including $S \subset S^*$. The only constraint is that $Hdr \neq Hdr^*$. The challenger responds with ***Decrypt(***S, i, D_i, 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 security model.* The adversary's advantage by using algorithm A to break the broadcast encryption system (BE) with parameters (λ, n) is defined as follows:

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

where b' is the algorithm $\mathcal{A}$'s guess of b in the adaptive security model. Let's define a helpful notation that will be used in the rest of this work.

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

We refer to the BE scheme that is secure in the adaptive model as BE_A in the rest of this work.

3.3 Bilinear Maps

We make extensive use of *bilinear maps* at the core of our proposed schemes, so let's first properly define a bilinear pairing.

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 $g, g', h, h' \in \mathbb{G}$,

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

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

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]. Cryptographic bilinear maps must have certain complexity properties that are explained in the following section.

3.4 Complexity Assumptions

In general, cryptographic bilinear maps need to be one-way functions, that is computing the bilinear pairing should be efficient, but the inverse has to be difficult [4, 5, 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 groups. 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 proof of security.

Definition 2 (BDHE-Sum Assumption (for n):). *As usual, let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order p with a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$, g a generator for $\mathbb{G}$ and $\alpha \xrightarrow{R} \mathbb{Z}^*{}_p$. 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-variant assumption 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 any advantage greater than ϵ in solving the decision BDHE-Sum problem for n in $\mathbb{G}_T$.

4 Adaptively Secure BE Construction

To derive a fully adaptively secure BE$_A$ scheme with short ciphertexts, our protocol increases the number of private keys. As before, we denote the maximum number of members in the broadcast group by n. Our BE$_A$ scheme is given as follows:

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

KeyGen(**i, SK**) Pick $r_i \xleftarrow{R} \mathbb{Z}^*{}_p$ and for all $j \in [0, n]$, pick randomly $B_j \xleftarrow{R} \mathbb{Z}^*{}_p$. Release to member i its private key as $D_i \leftarrow \{r_i, d_{i,j}, T_{i,j}\}$, where:

$$\begin{aligned} d_{i,j} &= g^{\gamma \alpha^n B_j \frac{\alpha^2 - r_i}{\alpha - x_i}}\ \forall\ j \in [1, n] \text{ and } j \neq i \\ T_{i,j} &= g^{\frac{\alpha^j}{\gamma B_i}}\ \forall\ j \in [0, n] \end{aligned}$$

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

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

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

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 $g^{t(\gamma B_i)^{-1} p(\alpha)}$ is in the following form:

$$g^{t(\gamma B_i)^{-1} p(\alpha)} = g^{t \frac{\sum_{j=0}^{n} e_j \alpha^j}{\gamma B_i}}$$

where e_j is a multiple of a few x_j values depending on the indices included in the set S. By knowing the random t, all the $T_{i,j}$ and x_j ($\forall\, j \in [0, n]$) values, the sender (member i) can readily calculate $g^{t(\gamma B_i)^{-1}p(\alpha)}$. The session key ($K$) is set as follows:

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

Output $\langle Hdr, K\rangle$.

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

$$K \leftarrow e(C_1, g^{p_i(\alpha)} \cdot g^{r_i h_i(\alpha)})e(C_2, d_{i,j})$$

where $p_i(\alpha) = \alpha^{2n+1} - \frac{\alpha^{n+2}p(\alpha)}{\alpha - x_i}$ and $h_i(\alpha) = \alpha^n \frac{p(\alpha)}{\alpha - x_i}$. Note that only if $i \in S$, p_i is a polynomial of degree $2n$ and $g^{p_i(\alpha)}$ can be directly calculated from g^{α^j} and x_j values. If $i \notin S$, then $p_i(\alpha)$ is polynomial of degree $2n + 1$, and it cannot be directly calculated from g^{α^j} as the maximum power i in g^{α^i} available is $2n$ (i.e. $g^{\alpha^{2n}}$). Similarly if $i \in S$, then $r_i h_i(\alpha)$ is polynomial of degree $2n - 1$ that can be directly calculated by knowing r_i, g^{α^j} and x_j values.

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

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

as required.

It is very easy to see how the proposed BE$_A$ scheme can be adapted to add sender's authentication to the broadcast encryption message.

Sender's Authentication: Let SymEnc and SymDec be symmetric encryption and decryption functions, respectively. Let M be a random verification message to be broadcast to the set S, and let $C_M \xleftarrow{R}$ 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 $\langle S, Hdr, M, C_M\rangle$. 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 via the ***Decrypt(**S, i, D_i, Hdr, PK**)*** function. Then, member i checks if $M =$ SymDec(K, C_M). If it passes, it verifies the sender, otherwise, it refuses the sender's authentication.

5 Security Analysis

In this section, we prove the full security of the proposed BE$_A$ scheme in the adaptive security model.

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

Proof. As usual, we start by the assumption that there is an algorithm $\mathcal{A}$ with advantage $\epsilon > negl(\lambda)$ 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 $\mathbb{G}$ with an advantage ϵ, which contradicts the presumed advantage $negl(\lambda)$ of the initial complexity assumption. We build a simulation machine $\mathcal{B}$ that receives an instance of the decision n-BDHE-Sum problem 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 non-compromised (non-attacked) member is used to generate a broadcast message (Hdr^*) only for the other non-compromised member. Otherwise, it is obvious that the adversary will be able to recover the session key, as it already has the private key of all the other members.

Setup. $\mathcal{B}$ disguises the parameters of the challenge problem into parameters of the proposed BE$_A$ scheme. $\mathcal{B}$ replaces $\alpha = a$ and using the challenge instance, it sets the public parameters as: $g^{\alpha^i} = g^{a^i}$ for $i \in [0, 2n]$. For public identities x_i, $\mathcal{B}$ as usual picks $x_i \xleftarrow{R} \mathbb{Z}^*{}_p$ and publishes public parameters 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 $\max(|S'|) = n - 2$. Let's denote the set of non-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 index in S^* other than j in our notations. Having known the set of attacked members (S') and the set of non-attacked members (S^*), $\mathcal{B}$ picks a random $b_j \xleftarrow{R} \mathbb{Z}^*{}_p$ and sets $B_j = b_j a^n$ for $j \in S'$, 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 $j \in S^*$.

For $i \in S'$, $\mathcal{B}$ responds to the query for member i's private keys as follows: it returns $D_i \leftarrow \langle r_i, d_{i,j}, T_{i,j} \rangle$, where in the real protocol r_i as a random value, $d_{i,j} = g^{\gamma\alpha^n B_j \frac{\alpha^2 - r_i}{\alpha - x_i}}$ and $T_{i,j} = g^{\frac{\alpha^j}{\gamma B_i}}$. Therefore for all $i \in S'$ and for $j \in S'$, $\mathcal{B}$ sets $r_i = x_i^2$ and returns $d_{i,j} = g^{y_0 b_j (a+x_i)}$. For $j \in S^*$, it returns $d_{i,j} = g^{y_0 b_j a^{-1}(a - x_{j\oplus 1})(a+x_i)}$ and $d_{i,j\oplus 1} = g^{y_0 b_j a^{-1}(a-x_j)(a+x_i)}$ to the adversary. Finally, $T_{i,j} = g^{(y_0 b_i)^{-1} a^{n+j}}$ for all $i \in S'$ and for $j \in [0, n]$. It should be noted that $\mathcal{B}$ does not need to calculate and return any $T_{i,j}$ for $i \in S^*$. All these parameters can be readily calculated from the BDHE-Sum instance and the random parameters picked by $\mathcal{B}$. 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 challenge (non-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 $C_2^* = g^{t(\gamma B_i)^{-1}\alpha^{n-|S|}\prod_{j\in S}(\alpha - x_j)}$, where $i \in S^*$. Let's suppose that the broadcast message is generated by member i, where $i \in S^*$ and therefore $S = \{i \oplus 1\}$. The challenge is then calculated as follows: pick a random $t_0 \xleftarrow{R} \mathbb{Z}^*{}_p$, and set $t = t_0 a^{2n}$. By applying the $\mathcal{B}$'s simulation parameters, the broadcast message $Hdr^* \leftarrow \langle C_1^*, C_2^* \rangle$ is calculated as $C_1^* = g^{t_0 a^{2n}}$. Since $B_i = b_i a^{n-1}(a - x_{i\oplus 1})$, we have C_2^* given as follows:

$$\begin{aligned} C_2^* &= g^{t_0 a^{2n}(y_0 a^{-2n} B_i)^{-1} a^{n-1}(a - x_{i\oplus 1})}, \\ &= g^{t_0 a^{2n}(y_0 a^{-2n} b_i a^{n-1}(a-x_{i\oplus 1}))^{-1} a^{n-1}(a-x_{i\oplus 1})}, \\ &= g^{t_0 (y_0 b_i)^{-1}}. \end{aligned}$$

Note that both C_1^* and C_2^* can be directly calculated from the BDHE-Sum instance and the random parameters picked by $\mathcal{B}$. Therefore, $Hdr^* = \{C_1^*, C_2^*\}$, where $C_1^* = (g^{a^{2n}})^{t_0}$ and $C_2^* = (g)^{t_0 (y_0 b_i)^{-1}}$, is a valid ciphertext for indices in $S \subset S^*$. The corresponding session key would then be $K = e(g,g)^{t_0 a^{2n} a^{2n+1}} = e(g,g)^{t_0 a^{4n+1}}$. $\mathcal{B}$ 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 used in decryption and $T_{i,j}$ values are used only 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 and therefore r_i values are uniformly random. When $b = 0$ in the adaptive game, $\langle Hdr^*, K^* \rangle$ 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, $\langle Hdr^*, K^* \rangle$ is generated with K^* being replaced by a random key. Since $K^* \xleftarrow{R} \mathbb{G}_T$ is a uniformly random element of $\mathbb{G}_T$, this distribution is identical to that of $\mathcal{B}$'s simulation where Hdr^* is a valid ciphertext. Therefore, $\mathcal{B}$'s advantage (ϵ) in deciding n-BDHE-Sum problem is precisely $\mathcal{A}$'s advantage ($negl(\lambda)$) in attacking the proposed BE$_A$ scheme. □

6 Performance Analysis

In this section, we analyze the overheads of the proposed scheme over previously known schemes. A fully adaptive BE$_A$ with short ciphertexts is achieved in this work. The design is aimed at constructing a fully secure BE scheme with $\mathcal{O}(1)$ communication overheads (Hdr), regardless of the size of the broadcast group. This is achieved without using the Random Oracle Model (ROM) and hash functions. In comparison with Gentry and Waters' BE scheme and its variants [13], the security of the proposed scheme is proved in an attack model that is stronger than Gentry and Waters' BE scheme. In our security model, we allow the adversary to query the private keys of all members under attack and also to receive decryption of broadcast messages intended for all members. Gentry and Waters' semi-static BE scheme requires $\mathcal{O}(1)$ private keys and $\mathcal{O}(n)$ public keys to be stored by each member in the broadcast group. They extend the security from semi-static to fully adaptive by increasing the size of broadcast message to $\mathcal{O}(\sqrt{n})$ without using random oracles (hash functions). In our scheme, the increase in security has led to an increase in the size of private keys – each member in our scheme has to store $\mathcal{O}(n)$ private keys. We manage to keep the size of the ciphertext remains fixed, and it does not increase with the size of the broadcast group.

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 at no extra communication overhead. In the proposed scheme, if more members (greater than n) need to be added to the broadcast group, adding new members causes extra communications and requires updates in 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. $n < |\mathbb{G}|$. 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 a key-update for existing members, we have to send the new decryption key $d_{i,i'}$ to the existing member (i) to allow communication with the new member i'.

7 Conclusions and Future Work

A broadcast encryption scheme based on cryptographic pairings is proposed in this work. The scheme is the first adaptively secure broadcast encryption with short ciphertexts that does not use the random oracle model. The security model of the proposed broadcast encryption scheme is a strong model simulating the adversary in the real world as closely as possible. In our model, the adversary can receive the private keys of any subset of members in the broadcast group as well as decryption of previous broadcast messages. The increase of security in our scheme has resulted in an increase in the size of private keys. However, this increase is compensated with the ability of our scheme to offer sender's authentication at no additional overheads. In our scheme, we have showed that how the sender of a broadcast message can be readily verified to all the members of the broadcast group.

It has also been shown that the communication and computation overheads 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 is 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 changes in the group. The maximum number of members that can join the group is limited by the underlying algebraic group structure. The maximum size of the broadcast group is bounded by the size of the underlying bilinear group. This implies that the size of the underlying pairing group increases linearly with the maximum size of the broadcast group.

References

[1] Y. Amir, Y. Kim, C. Nita-Rotaru, and G. Tsudik, "On the performance of group key agreement protocols," *ACM Transactions on Information and System Security*, vol. 7, no. 3, pp. 457–488, 2004.

[2] R. Barua, R. Dutta, and P. Sarkar, "Extending JOUX protocol to multi party key agreement," *Advances in Cryptology: INDOCRYPT'03*, Springer-Verlag, LNCS 2904, pp. 205–217, 2003.

[3] I. F. Blake, G. Seroussi, and N. P. Smart, *Advances in Elliptic Curve Cryptography*, Cambridge University Press, 2005.

[4] D. Boneh and X. Boyen, "Efficient selective-ID identity based encryption without random oracles," *Advances in Cryptology: EUROCRYPT'04*, Springer-Verlag, LNCS 3027, pp. 223–238, 2004.

[5] D. Boneh, X. Boyen, and E. J. Goh, "Hierarchical identity based encryption with constant size ciphertext," *Advances in Cryptology: EUROCRYPT'05*, Springer-Verlag, LNCS 3494, pp. 440–456, 2005.

[6] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadast encryption with short ciphrertexts and private keys," *Advances in Cryptology: CRYPTO'05*, Springer-Verlag, LNCS 3621, pp. 258–275, 2005.

[7] M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system," *Advances in Cryptology: Eurocrypt'94*, Springer-Verlag, LNCS 950, pp. 275–286, 1995.

[8] Y. Challal and H. Seba, "Group key management protocols: A novel taxonomy," *International Journal of Information Theory*, vol. 2, no. 1, pp. 105–118, 2005.

[9] L. Chen, M. Cheng, and N. P. Smart, "Identity-based key agreement protocols from pairings," *International Journal of Information Security*, vol. 6, no. 4, pp. 213–241, 2007.

[10] G. H. Chiou and W. T. Chen, "Secure broadcasting using the secure lock," *IEEE Transactions on Software Engineering*, vol. 15, no. 8, pp. 929–934, 1989.

[11] K. Y. Choi, J. Y. Hwang, and D. H. Lee, "Efficient ID-based group key agreement with bilinear maps," *Proceedings of Public Key Cryptography*, Springer-Verlag, LNCS 2947, pp. 130–144, 2004.

[12] Y. Dodis and N. Fazio, "Public key broadcast encryption for stateless receivers," *Digital Rights Management: DRM'02*, Springer Verlag, LNCS 2696, pp. 61–80, 2002.

[13] C. Gentry and B. Waters, "Adaptive security in broadcast encryption systems (with short ciphertexts)," *Advances in Cryptology: EUROCRYPT'09*, Springer-Verlag, LNCS 5479, pp. 171–188, 2009.

[14] D. Halevy and A. Shamir, "The LSD broadcast encryption scheme," *Advances in Cryptology: CRYPTO'02*, Springer-Verlag, LNCS 2442, pp. 145–161, 2002.

[15] A. Joux, "A one round protocol for tripartite Diffie-Hellman," *the 4th International Symposium on Algorithmic Number Theory*, Springer-Verlag, LNCS 1838, pp. 385–394, 2000.

[16] M. Manulis, "Security-Focused survey on group key exchange protocols," *Technical Report November*, HGI Network and Data Security Group, 2006. (http://www.manulis.eu/papers/TR0603-GKEPS.pdf)

[17] D. Naor, M. Naor, and J. B. Lotspiech, "Revocation and tracing schemes for stateless receivers," *Advances in Cryptology: CRYPTO'01*, Springer-Verlag, LNCS 2139, pp. 41–62, 2001.

[18] A. Perrig, D. Song, and J. D. Tygar, "ELK, a new protocol for efficient large-group key distribution," *IEEE Security and Privacy Symposium*, pp. 247–262, 2001.

[19] S. Rafaeli and D. Hutchison, "A Survey of Key Management for secure group communication," *ACM Computer Surveys*, vol. 35, no. 3, pp. 309–329, 2003.

Behzad Malek received his Master's degree from University of Ottawa, Canada in Electrical Engineering in 2005. He is currently a PhD Candidate in Electrical and Computer Engineering at the University of Ottawa, Canada. His research interests are communications security and privacy. He is specialization is in applied cryptography and design of cryptographic security/privacy systems. He has been a student IEEE member and also a member of Computational Laboratory in Coding and Cryptography (CLiCC) at the University of Ottawa since 2003.

Ali Miri has been a Full Professor at School of Computer Science, Ryerson University, Toronto since 2009. He has also been with the School of Information Technology and Engineering and Department of Mathematics and Statistics since 2001 as an Assistant Professor, and later as an Associate Professor in 2005 and a Full Professor at 2008. He is the founder and the director of Computational Laboratory in Coding and Cryptography (CLiCC) at the University of Ottawa. His research interests include computer networks, digital communication, and security and privacy technologies and their applications. He is the author and co-authoring of more than 120 peer-reviewed papers in international conference and journals. Dr. Ali Miri has served in more than 50 organizing and technical program committees of international conferences and workshops. He has served as a guest editor for Journal of Ad Hoc and Sensor Wireless Networks, Journal of Telecommunications Systems, and is currently serving on the editorial board of International Journal On Advances in Internet Technology. He is a member of Professional Engineers Ontario, ACM and a senior member of IEEE.