1-1-2015

Broadcast encryption with dealership

Clementine Gritti
University of Wollongong, cjpg967@uowmail.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Thomas Plantard
University of Wollongong, thomaspl@uow.edu.au

Kaitai Lang
Aalto University

Duncan S. Wong
City University of Hong Kong, dwong@uow.edu.au

Broadcast encryption with dealership

Abstract

In this paper, we introduce a new cryptographic primitive called broadcast encryption with dealership. This notion, which has never been discussed in the cryptography literature, is applicable to many realistic broadcast services, for example subscription-based television service. Specifically, the new primitive enables a dealer to bulk buy the access to some products (e.g., TV channels) from the broadcaster, and hence, it will enable the dealer to resell the contents to the subscribers with a cheaper rate. Therefore, this creates business opportunity model for the dealer. We highlight the security consideration in such a scenario and capture the security requirements in the security model. Subsequently, we present a concrete scheme, which is proven secure under the decisional bilinear Diffie-Hellman exponent and the Diffie-Hellman exponent assumptions.

Publication Details

Gritti, C., Susilo, W., Plantard, T., Lang, K. & Wong, D. S. (2016). Broadcast encryption with dealership. International Journal of Information Security, 15 (3), 271-283.

Noname manuscript No.
(will be inserted by the editor)

Broadcast Encryption with Dealership

**Clémentine Gritti, Willy Susilo, Thomas Plantard,
Kaitai Lang and Duncan S. Wong**

Received: date / Accepted: date

Abstract In this paper, we introduce a new cryptographic primitive called Broadcast Encryption with Dealership (BED). This notion, which has never been discussed in the cryptography literature, is applicable to many realistic broadcast services, for example subscription-based television service. Specifically, the new primitive enables a dealer to bulk buy the access to some products (e.g. TV channels) from the broadcaster and hence, it will enable the dealer to resell the contents to the subscribers with a cheaper rate. Therefore, this creates business opportunity model for the dealer. We highlight the security consideration in such a scenario, and capture the security requirements in the security model. Subsequently, we present a concrete scheme, which is proven secure under the Decision Bilinear Diffie-Hellman Exponent (DBDHE) and the Diffie-Hellman Exponent (DHE) assumptions.

Keywords Broadcast encryption · dealership · provable security

1 Introduction

Pay TV, or sometimes referred to as the subscription-based television service, provides a mechanism to allow the subscribers to view the broadcast encrypted channel provided by the TV broadcaster, provided that the subscribers own a special device (known as a "decoder") to join the network. In practice, a subscriber, Alice, will usually purchase a package that combines several channels offered by the broadcaster to get a better deal. Recently, the scenario of Pay TV includes the notion of "dealership". We note that the term dealership does not refer to a trusted third party, but instead, it represents an entity who deals with the broadcaster to offer better price to the subscribers. This case allows a dealer to purchase some channels from the broadcaster and to sell them with a cheaper price. For instance, the dealer, Don, would purchase 100 access to the sport channel from the broadcaster and he will be offered a discounted rate at $ 20 per month, instead of paying the full fare, which is $ 30 per month for the same contents. This way, now Don can offer his service by re-selling the channel to the subscribers (who have owned the decoder), but with a slightly cheaper rate, for example $ 25 per month. Then, Don can still make his business meanwhile the subscribers will enjoy the discounted rate. Of course, it is Don's interest to protect the identity of his subscribers from the original broadcaster or else, the broadcaster may contact the subscribers directly and offer a better rate.

Although the scenario of Broadcast Encryption has been thoroughly studied in the cryptography literature [17,15,18,29,13], there exists no work that discuss the aforementioned realistic scenario to allow Broadcast Encryption with Dealership. An unattractive solution would be to employ a fully trusted dealer, which is impractical.

Clémentine Gritti, Willy Susilo and Thomas Plantard
Centre for Computer and Information Security Research
School of Computer Science and Software Engineering
University of Wollongong, Australia
E-mail: cjpg967@uowmail.edu.au,
{wsusilo,thomaspl}@uow.edu.au

Kaitai Lang and Duncan S. Wong
The Department of Computer Science
City University of Hong Kong, Hong Kong
E-mail: liang4-c@my.cityu.edu.hk, duncan@cityu.edu.hk

Our work

In this paper, we extend the existing notion of Broadcast Encryption to enable the notion of Broadcast Encryption scheme with Dealership (BED). In the new notion, there are three types of participants, namely a broadcaster, a dealer and a set of subscribers (resp. receivers). In the setup stage, the subscribers will join as members of the broadcaster (i.e., by purchasing a decoder). Then, the dealer is allowed to choose a subgroup of subscribers and generate an auxiliary information, aux, that will need to be sent to the broadcaster. Upon receiving aux, the broadcaster encrypts the purchased channel using this information such that only a subset of subscribers which have been included in aux can decrypt the scrambled contents. The broadcaster can also check the size of aux, but will not know which subscribers that have been included in aux. This is to follow the scenario that only the dealer knows the identity of the subscribers who have purchased the contents from him/her, but not the broadcaster, since otherwise, the broadcaster will be able to approach those subscribers at a later stage and to offer a better rate compared to the dealer. We provide a sound security model to capture this scenario, and subsequently present a concrete construction. Meanwhile, our scheme is proven secure under the Decision Bilinear Diffie-Hellman Exponent (DBDHE) and the Diffie-Hellman Exponent (DHE) assumptions.

One might think that our construction could be straightforward. Nevertheless, we stress the difficulty of achieving such a scheme is due to the necessary security requirements that are required by each entity in the system. We illustrate some of the difficulties as follows. From the dealer's point of view, he/she prefers to protect the privacy of the subscribers. This is to protect the dealer's interest so that the broadcaster cannot approach the subscribers directly and offer them with a better price, and hence, disadvantaging the dealer. On the other hand, the broadcaster must be ensured that the total number of the subscribers given by the dealer is limited, and hence, the dealer cannot cheat by only purchasing the contents for n users but eventually he/she will resell the contents for $n+1$ subscribers. Furthermore, we need to guarantee that the dealer cannot decrypt the purchased contents or else, the dealer can later re-broadcast the information to other illegal subscribers. Similarly, the users who do not subscribe to the channel should not be allowed to decrypt the broadcast information as well.

We need to intelligently combine the Broadcast Encryption scheme and the Membership Encryption scheme to implement our construction. Using a simple Broadcast Encryption construction does not allow the broadcaster to verify the number of users selected by the dealer. The broadcaster would not have an interaction with the dealer to check that the dealer have chosen less than n users. That means that the broadcaster generates the public/private key pairs for all the users, the dealer selects a unknown number of users and encrypts the message for these users and then, the broadcaster forwards the ciphertext to all the users such that only the users selected by the dealer can retrieve the original message. A Membership Encryption protocol does not keep the idea of broadcasting a ciphertext to several users. Indeed, there is only an interaction between a verifier and a prover: the latter wants to prove that a user A belongs to the group of users $P(G)$ without revealing other users in G to the verifier.

In our proposed framework, we adopt some practical assumptions as follows. We assume that dealer will not collude with the users, as both entities have their own separate interest. It is of course theoretically possible to allow this collusion, but in our work, we assume that this impractical collusion will not occur. Additionally, we assume that the dealer is also rational, in the sense that the dealer will not cheat the users by selling "fake" membership, which later will allow the users to report the case to the broadcaster and having the dealer blacklisted from future business. We note that these assumptions are made in order to follow realistic situation as well as to avoid us having unnecessary complication in the scheme.

One may note that the notion of BED is similar to another concept called Digital Right Management (DRM) [24]. DRM is a system to protect high-value digital goods and control the distribution and usage of these digital goods. A DRM process engages four entities: the content provider, the distributor, the clearing house and the consumer. A DRM system is included with an e-commerce system that treats financial payoffs and launches the function of the clearing house. In a DRM model, a license is transmitted from the clearing house to the consumer (who requested it) before or at the same moment as the transfer of the digital content, for instance for temporary promotional purposes.

Furthermore, an extension of DRM systems for the purpose of managing personal data held and controlled by organizations was proposed in [23]. In [31], the authors proposed a secure solution to suit DRM in home networks consisting of consumer electronic devices. One issue is that licences can leak information about the enterprise data; and in order to overcome this issue, the Platform for Privacy Preferences Language (P3P) is used to communicate enterprise privacy policies to consumers and enable them to easily construct data licenses [32]. DRM concept can also be extended to the health domain in order to protect the privacy of patient's electronic health records stored in a cloud storage [22]. Despite all the similarities between the two notions, there are some subtle and important differences between them, which will be outlined below. We note that we will use the terms in DRM for the sake of comparison below, but without further elaboration. We refer the readers to [24] for a more complex account.

We should note that BED and DRM do not have the same goal, and therefore, there are various differences among them. First, a BED system provides a *long-term decoder* to the subscribers, whereas a DRM system offers *temporary licenses* to the consumers. These two features imply that DRM only allows for a *discrete access* whereas a BED process works for *continuous access*. Hence, DRM can handle a piece of digital contents, such as music or videoclip, in which their size have been known a priori. More precisely, in the DRM case, the temporary license specifies a n-time access (e.g. "try-before-use" business model) for a piece of digital content (e.g. music, video). However, in the BED, the digital content to be broadcasted in a flow of information, from time t_0 until time t_m without interruption (e.g. TV channel broadcast), via the decoder. Finally, the most distinct feature is that the BED concept is designed in a *multi-user setting*: to reduce the computational and communication costs, the broadcaster and the dealer interact with the subscribers as a whole. In addition, the dealer is able to *select* one part of all the subscribers; meanwhile, the broadcaster can *verify* the number of selected users. In contrast, DRM schemes propose only *one-to-one processes*.

In Fig. 1, we highlight the similarities and the differences between DRM and BED systems.

	BED	DRM
Long-time decoder/license	✓	×
Continuous stream access	✓	×
Contents distributed in an encrypted form	✓	✓
Multi-user setting	✓	×
Selection of users depending of the contents	✓	✓
Verification of the number of selected users	✓	×

Table 1 Comparison between DRM and BED systems. The terminology "user" refers to "consumer" for the DRM systems and to "subscriber" for the BED system.

1.1 Related Work

The notion of Broadcast Encryption was introduced by Fiat and Naor [17]. In a Broadcast Encryption scheme, a sender who would like to share a message with a dynamically chosen subset $S \subseteq [1, N]$ of users will construct a ciphertext such that only users in S can decrypt. Then, he/she can safely transmit this ciphertext over a broadcast channel to all users. The main challenge in building efficient broadcast systems is to encrypt messages with short ciphertexts. Usually, Broadcast Encryption systems rely on combinatorial techniques. Such systems include a collusion bound t, where using larger values of t impacts system performance.

The concept of Broadcast Encryption receives much attention since the work of Naor et al. [25]: they presented a symmetric-key structure together with security model and security analysis. Dodis and Fazio [16] presented the first public-key CCA-secure Broadcast Encryption scheme. Boneh, Gentry and Waters [3] designed a fully collusion resistant scheme and proposed a security model where the adversary can corrupt any user, except the users in the challenge target set S^*. They presented new methods for achieving fully collusion resistant systems with short ciphertexts by applying computational techniques using group with bilinear maps. With their scheme, the adversary has to provide precisely S^* prior to knowing the parameters of the system, and hence, this model is known as the *selective model*. Delerabrée constructed a selectively secure ID-based Broadcast Encryption [13] in the random oracle model. Subsequently, Gentry and Waters [18] presented the first scheme in the *adaptive model*, where the adversary can corrupt users and then adaptively choose the challenge target set. They provided adaptively secure schemes in both the standard and the random oracle models. At the same time, [33] used the dual

encryption technique to build more efficient adaptively secure Broadcast Encryption systems than those obtained from the generic construction based on schemes with semi-static security. Thereafter, a scheme achieved all desired properties (constant-size ciphertexts, adaptive CCA-security) and was presented in [29] but it relies on rather non-standard assumptions.

Phan, Pointcheval and Strefler [27] recently gave a general view of the relations between the security notions for Broadcast Encryption. However, the setting of multi-channel Broadcast Encryption is out of their consideration, because the adversary could corrupt some users of one channel to break the security of the other channels. The sessions keys of all channels should indeed be compacted into one ciphertext only, there are thus some relations between keys inside one session and the security model has to take these relations into account. Finally, Phan, Pointcheval and Trinh [30] made a significant breakthrough by proposing a formalization of the multi-channel Broadcast Encryption problem, using the *dummy-helper* technique.

The privacy-preserving membership proof allows one to prove that a user A belongs to a users group G, while the privacy is protected. There are two techniques of privacy-preserving membership proof:

- Set membership proof [11,10,8] for $A \in G$ given $P(A)$. In such a proof, the token $P(A)$ and the users in G are known by a verifier. The goal for a prover is to show that the user in $P(A)$ is in G without leaking the identity A to the verifier. This kind of proof enables us to provide the privacy of the involved user.
- Accumulator for witness [4,5,26,19,9,1] for $A \in G$ given $P(G)$. In this case, the token $P(G)$ and a user A are known by a verifier. The goal of a prover is to show that A belongs to the group of users in $P(G)$ without leaking the identities of the other users in G to the verifier. This kind of proof allows us to protect the privacy of the non-involved users.

In [20], Guo et al. proposed an encryption primitive, which is known as the *Membership Encryption*. In such a scheme, the decryption satisfies the privacy-preserving group membership $A \in G$ given $P(G)$, where the group token $P(G)$ is generated from the users group G and a secret token S. If the encryption takes A and $P(G)$ as inputs, the decryption succeeds if and only if the decryptor knows (G, S) and $A \in G$ is true. They provided a provably secure Membership Encryption such that:

- $P(G)$ is constant-size with maximum number accountability on users, and independent of the number of users in G, and
- the upper bound user number in $P(G)$ is accountable, the ciphertext is constant-size with maximum number accountability on users, and linearly dependent on the length of security parameter.

1.2 Our Contributions

In this paper, we introduce the notion of Broadcast Encryption scheme with Dealership (BED). Our construction is inspired by Gentry and Waters [18] Broadcast Encryption scheme and by Guo et al. [20] Membership Encryption scheme, with some subtle changes. We should note that a trivial merger between these two schemes will result to an insecure scheme. We proved the security of the scheme to be semi-static secure under the n-Decision Bilinear DHE problem, preserving privacy, and secure with maximum number accountability under the (f, n)-DHE problem.
Our scheme is equipped with following security properties:

- *Security against illegal users:* A user receiving a ciphertext of which he/she is not a legal subscriber is unable to decrypt it. Likewise, the dealer receiving the contents sent by the broadcaster cannot decrypt them, and thus, cannot re-broadcast them later to some illegal users.
- *Privacy:* Even though the broadcaster knows all the identities of the subscribers, the broadcaster will not know who are recipients of the ciphertexts, since the group of recipients is only known to the dealer.
- *Maximum number accountability:* The broadcaster needs to be assured that the total number of users selected by the dealer is limited, and thus, the dealer cannot cheat by enabling the encrypted contents to $n+1$ subscribers for the price of n subscribers.

Following the Broadcast Encryption scheme proposed by Gentry and Waters [18], we prove that our scheme is secure in the semi-static model. In this model, an adversary must commit to a set S before the **Setup** algorithm, but then it can attack any set $S^* \subseteq S$. Thus, the adversary has more flexibility than in the static model.

Paper Organization

The rest of this paper is organized as follows. In the next section, we will revisit the definition of some hard problems used in our security proofs. Sec. 3 introduces the syntax of BED that will be used throughout this paper defines the security models of BED. In Sec. 4, we present our construction of the Broadcast Encryption scheme with Dealership, and give its security proofs in Section 5. The proofs are provided for semi-static security, privacy and maximum number accountability. Finally, Sec. 7 concludes the paper.

2 Complexity Assumptions

We first define the two hard problems adopted in the security proofs of the system: the DBDHE problem and the DHE problem. The first one was proposed by Boneh, Gentry and Waters in [3], and the second one was established by Delerabrée in [13].

Definition 1 (n-Decision Bilinear DHE problem) Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order p with bilinear maps $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$, and let g be a generator for $\mathbb{G}$. Set $a,s\in_R\mathbb{Z}_p^*$, and $b\in_R\{0,1\}$. If $b=0$, set $Z=e(g,g)^{a^{n+1}\cdot s}$; otherwise, set $Z\in_R\mathbb{G}_T$. The problem instance consists of g^s, Z, and the set $\{g^{a^i};i\in[0,n]\cup[n+2,2n]\}$. The problem is to guess b.

The n-DBDHE problem holds with t,ε if given an problem instance generated from a security parameter 1^λ, the advantage of solving this problem in t polynomial time is ε at most which is a negligible function of λ.

Definition 2 ((f,n)-DHE problem) Let $\mathbb{G}$ and $\mathbb{G}_T$ be groups of order p with bilinear maps $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$, and let g_0,h_0,w be random generators for $\mathbb{G}$. Set $a\in_R\mathbb{Z}_p^*$. The problem instance consists of a n'-degree polynomial function $f(x)\in\mathbb{Z}_p[x]$ for $n'>n$, and of $g_0,g_0^a,g_0^{a^2},\cdots,g_0^{a^n}$. The problem is to compute $(f(x),g_0^{f(a)})$.

The (f,n)-DHE problem holds with t,ε if given an problem instance generated from a security parameter 1^λ, the advantage of solving this problem in t polynomial time is ε at most which is a negligible function of λ.

3 Definition and Security Models

3.1 Definition

Definition 3 A Broadcast Encryption scheme with Dealership (BED), as depicted in Fig. 1, can be seen as a key encapsulation mechanism with Membership Encryption. The scheme comprises the following six algorithms:

- **Setup:** (Broadcaster) On input a security parameter 1^λ, an integer n corresponding to the upper bound number of users, and a group of users $\{A_1,\cdots,A_n\}$, the **Setup** algorithm generates the public parameters PP and the master secret key MSK, where $n=poly(1^\lambda)$.
- **KeyGen:** (Broadcaster) On input the public parameters PP, the master secret key MSK, and a user A_i for $i\in[1,n]$, the **KeyGen** algorithm generates a public/private key pair (PK_i, SK_i).
- **GroupGen:** (Dealer) On input the public parameters PP, a group of selected users $G=\{A_1,\cdots,A_k\}$, such that $1\le k\le n$, and the public keys $\{PK_1,\cdots,PK_k\}$ of the users belonging to G, the **GroupGen** algorithm returns the token $P(G)$.
- **Verify:** (Broadcaster) On input the token $P(G)$, and an integer k, the **Verify** algorithm returns True if the user number in $P(G)$ satisfies $|G|\le k$; otherwise, it outputs False.
- **Encrypt:** (Broadcaster) On input the public parameters PP, a token $P(G)$, and a message M, the **Encrypt** algorithm returns a ciphertext C of M.
- **Decrypt:** (User) On input a user A_i for $i\in[1,n]$, the group of selected users G, the public/private key pair (PK_i,SK_i), and a ciphertext C, the **Decrypt** algorithm returns the message M or $\perp$.

Correctness: For public parameters PP, any token $P(G)$ corresponding to a group G, and any ciphertext $C\leftarrow$ **Encrypt**$(PP,P(G),M)$, we have:

- if $A_i\in G$, then we have **Decrypt**(**Encrypt**$(PP,P(G),M),A_i,G,(PK_i,SK_i))=M$;
- otherwise, we have **Decrypt**(**Encrypt**$(PP,P(G),M),A_i,G,(PK_i,SK_i))=\perp$.

3.2 Semi-static Security Model

Below we formalize the security notions for the BED system. The strongest security model in Broadcast Encryption systems is the adaptive secure system. In an adaptively secure system, the adversary

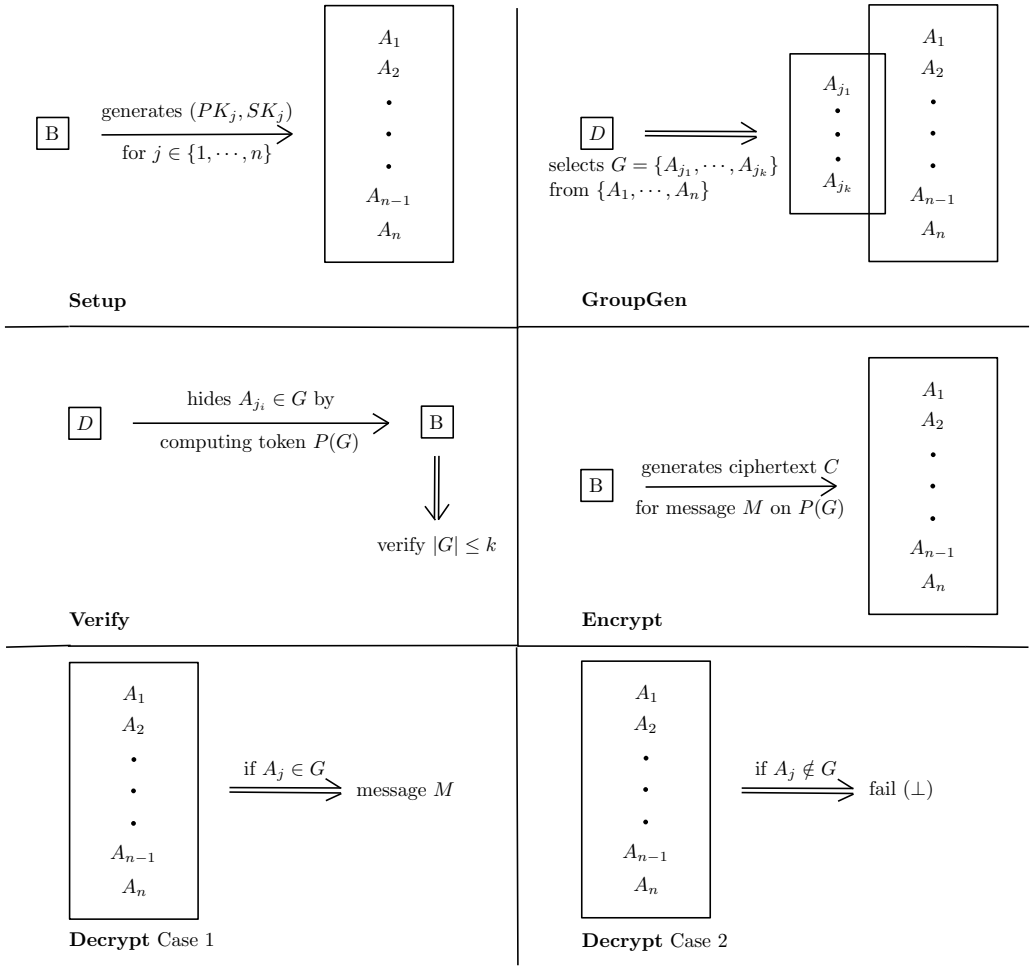

Fig. 1 Broadcast Encryption with Dealership (BED). The three participants are: the broadcaster B, the dealer D and the users A_i.

is allowed to see the public parameters (and the public keys), and then ask several private keys before choosing the set of indices that it wishes to attack.

Nevertheless, in this paper, we consider a weaker security definition: the semi-static security. This notion was introduced by Gentry and Waters [18] to prove the security of their Broadcast Encryption scheme. They also provided a generic technique to transform a semi-static secure Broadcast Encryption to an adaptive secure scheme. We note that their transformation is also applicable to our scheme, and therefore, we omit the construction of a fully-secure system in this paper.

In the game for semi-static security, the adversary must commit to a set S of users at the *Init* phase. The adversary cannot query a private key for any $A_i \in S$, and it must choose a target set S^* for the challenge ciphertext, such that $S^* \subseteq S$.

Game for Semi-static Security Model

- *Init.* The adversary commits a users set $S \subseteq \{A_1, \cdots, A_n\}$.
- *Setup.* The challenger runs the **Setup** algorithm to obtain the public parameters and the public keys of users, and sends them to the adversary.
- *Key Query Phase.* The adversary issues private key queries for users $A_i \in \{A_1, \cdots, A_n\} \setminus S$.
- *Challenge.* The adversary specifies a challenge set $S^* \subseteq S$, such that for all private keys SK_i queries corresponding to users A_i, we have that $A_i \notin S^*$. The challenger runs the **GroupGen** algorithm with input S^* and receives $P(S^*)$. It then encrypts a message M under PK and $P(S^*)$ to obtain the ciphertext C^0 and $C^1 \in_R \mathbb{G}_T$. It sets $b \in_R \{0,1\}$, and gives C^b to the adversary.
- *Guess.* The adversary outputs a guess b' for b, and wins the game if $b' = b$

We define the advantage of the adversary as

$AdvBED_{\mathcal{A}}^{SS}(1^{\lambda}) = |Pr[b' = b] - 1/2|$.

Definition 4 A BED generated with a security parameter 1^{λ} is semi-static secure with (t,ε) if for all t-polynomial time adversaries, $\varepsilon = AdvBED_{\mathcal{A}}^{SS}(1^{\lambda})$ is a negligible function of λ.

3.3 Privacy

In this paper, we use the word "privacy" to denote that the broadcaster knows the identities of all the users (as he generates their public/private key pairs), but he does not know the identities of the users selected by the dealer (he only knows the maximum number of selected users as he gave it to the dealer as a constraint). The term "privacy" here does not refer to the complete anonymity towards the users, but we emphasize that the dealer chooses the privileged users and keeps their identities secret from the broadcaster.

A BED preserves the privacy of group users. Essentially, it means that given a group token $P(G)$, where G represents a group of subscribers, and two user groups $G_0 = \{A_1, \cdots, A_{k_0}\}$ and $G_1 = \{A'_1, \cdots, A'_{k_1}\}$, it is computationally hard to decide whether $G = G_0$ or $G = G_1$.

Game for Privacy

- *Setup.* The challenger runs the **Setup** algorithm to generate the public parameters PP, and sends them to the adversary.
- *Challenge.* The adversary gives the challenger two users group $G_0 = \{A_1, \cdots, A_k\}$ and $G_1 = \{A'_1, \cdots, A'_k\}$, for $k < n$. The challenger responds by choosing a bit $b \in \{0,1\}$ at random, and generating $P(G_b)$ for G_b. Then, the challenger sends $P(G_b)$ to the adversary.
- *Win.* The adversary outputs a guess b' of b, and wins the game if $b' = b$.

We define the advantage of the adversary as

$AdvBED_{\mathcal{A}}^{P}(1^{\lambda}) = |Pr[b' = b] - 1/2|$.

Definition 5 A BED generated with a security parameter 1^{λ} preserves the privacy of group tokens with (t,ε) if for all t-polynomial time adversaries, $\varepsilon = AdvBED_{\mathcal{A}}^{P}(1^{\lambda})$ is a negligible function of λ. We say it unconditionally preserves the privacy of group tokens if $\varepsilon = 0$ for any time t and PP generated by the adversary.

3.4 Maximum Number Accountability Model

The property of maximum number accountability ensures the broadcaster that given a group token $P(G)$, the broadcaster is guaranteed that the encrypted contents will only be decryptable by maximum n users, where $|G| = n$. Intuitively, BED verifies the property of maximum number accountability, if it is computationally hard to generate a group token $P(G)$ for G with n users, but the **Verify** algorithm outputs $|G| \leq k$, for a certain value $k < n$ given by the broadcaster.

Game for Maximum Number Accountability Model

- *Challenge.* The challenger runs the **Setup** algorithm to generate the public parameters and chooses a value $k < n$, and sends them to the adversary.
- *Win.* The adversary outptus $(P(G^*), G^*)$ and wins the game if G^* contains n users but the **Verify** algorithm outputs $|G^*| \leq k$.

We define the advantage of the adversary as

$AdvBED_{\mathcal{A}}^{MNA}(1^{\lambda})$.

Definition 6 A BED generated with a security parameter 1^{λ} is (t,ε)-secure with maximum number accountability if for all t-polynomial time adversaries, $\varepsilon = AdvBED_{\mathcal{A}}^{MNA}(1^{\lambda})$ is a negligible function of λ.

4 BED Construction

In this section, we present the construction of BED . Our construction is inspired by Gentry-Waters' Broadcast Encryption scheme [18] and Guo et al.'s Membership Encryption scheme [20].

The main difficulty is due to the "right order" of public keys that is required in the original constructions, which will prohibit us to build such a scheme.

4.1 A Quick Reminder about Membership Encryption

Membership Encryption was recently initiated by [20]. Let $P(G)$ be a privacy-preserving token on a group of users. It is assumed that it is hard to know the users in G given $P(G)$. The membership encryption is used to construct the membership proof $A \in P(G)$ with privacy-preserving on users and the membership.

In [20], the authors extended their group token generation from the accumulator scheme in [26], with two secret keys a and b. The systems parameter comprises $u \in \mathbb{G}$ and $u, u^a, u^{a^2}, \cdots, u^{a^n}$, $u^{b \cdot a}, u^{b \cdot a^2}, \cdots, u^{b \cdot a^n}$. For $G = \{A_1, \cdots, A_k\} \in \mathbb{Z}_p$, they set the group token $P(G) = (w_1, w_2, w_3) = (u^{t \prod_{i=1}^{k}(a+A_i)}, u^{t \cdot b \prod_{i=1}^{k}(a+A_i)}, u^{t \cdot b \cdot a^{n-k} \prod_{i=1}^{k}(a+A_i)})$, where the secret key t is randomly chosen from $\mathbb{Z}_p$, and w_3 can be seen as the element for user number verification. They add the component $u^{\frac{1}{(a+A)(b+A)}}$ in the system parametr, and pick $r \in \mathbb{Z}_p$ at random. Their approach can be described as follows:

- If $A \in G$, then $w_2 w_1^A$ contains $(a+A)(b+A)$ such that

$$e((w_2 w_1^A)^r, u^{\frac{1}{(a+A)(b+A)}}) = e(u,u)^r \prod_{A_i \in G \setminus \{A\}} (a+A_i)$$

  is computable from u^r and the system parameter.
- If $A \notin G$, $w_2 w_1^A = u^{t(b+A)\prod_{i=1}^{k}(a+A_i)}$ such that

$$e((w_2 w_1^A)^r, u^{\frac{1}{(a+A)(b+A)}}) = e(u,u)^{r \frac{t \prod_{A_i \in G}(a+A_i)}{a+A}}$$

  contains the inversion exponent $\frac{1}{a+A}$, which is not computable from u^r and the system parameter.

Finally, the authors use these results to encrypt the message such that the decryption requires $A \in G$.

4.2 A Naive Solution

A naive solution should be to combine the Broadcast Encryption scheme in [18] and the Membership Encryption scheme proposed by [20]. Intuitively, following the technique of the latter, the **KeyGen** algorithm, the broadcaster generates the public/private key pairs of users in $\{A_1, \cdots, A_n\}$. The public key PK_i of user A_i results in the components $u_i = h^{\gamma \alpha^i}$ and $v_i = h^{\gamma \beta \alpha^i}$. Thereafter, in the **GroupGen**, the dealer chooses k users in $\{A_1, \cdots, A_n\}$ to create the group G and then, to compute the group token $P(G)$. We notice that the group token generation can be seen as an extension of the accumulator scheme in [26]. The dealer computes the coefficients F_i of the k-degree polynomial F where the roots are the identities of the selected users A_i, and then calculates the products $\prod_{i=0}^{k} u_i^{F_i t}$, $\prod_{i=0}^{k} v_i^{F_i t}$, and $\prod_{i=0}^{k} v_{n-k+i}^{F_i t}$. We notice that calculating these products requires to employ the components $u_1, \cdots, u_k$ and $v_1, \cdots, v_k$ in "right order", from the public keys of users $A_1, \cdots, A_k$. That means that the dealer has a restricted choice when selecting the users. Indeed, for a certain number $k < n$, the k first users of $\{A_1, \cdots, A_n\}$ should be selected, namely $\{A_1, \cdots, A_k\}$. Therefore, this approach is too restrictive and unrealistic for our purpose.

Instead, in the algorithm **GroupGen**, we imagine that the dealer would like to choose the privileged users without constraint of order. For instance, he picks users A_2, A_4, A_5, A_8 (for $k = 4 < n$). Following the technique proposed in [20], he cannot compute $P(G)$ for $G = \{A_2, A_4, A_5, A_8\}$. Therefore, we found another technique for letting the dealer choose the privileged users whatever their indices and keeping the form of $P(G)$ from [20], in order to verify the number of selected users. For that, we do not construct the polynomial $F(x)$ and we do not directly attribute the (u_i, v_i) to the users. Instead, we let the values (u_i, v_i) public and we construct the first three elements of $P(G)$ as

$$u_0^{t_1 \cdot \prod_{A_i \in G}(x_i+\alpha)}, v_0^{t_1 \cdot \prod_{A_i \in G}(x_i+\alpha)}, v_{n-k}^{t_1 \cdot \prod_{A_i \in G}(x_i+\alpha)}.$$

4.3 Our solution

Our BED scheme is as follows.

Let $\mathbf{GroupGen}(1^\lambda, n)$ be an algorithm that, on input the security parameter 1^λ, outputs a pairing group $\mathbb{PG} = (\mathbb{G}, \mathbb{G}_T, e, p, g)$, where $\mathbb{G}, \mathbb{G}_T$ are two cyclic groups of prime order $p = p(1^\lambda, n) > n$, g is a generator of $\mathbb{G}$, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map. The map e satisfies the following properties:

- For all $u, v \in \mathbb{G}$, for all $a, b \in \mathbb{Z}_p$, $e(u^a, v^b) = e(u,v)^{ab}$.
- $e(g,g)$ is a generator of $\mathbb{G}_T$.
- Computing $e(u,v)$ for any $u, v \in \mathbb{G}$ is computationally efficient.

Setup: On input a security parameter 1^λ, the upper bound number of users n and a group of users $\{A_1, \cdots, A_n\}$, the **Setup** works as follows:

- Choose a pairing group $\mathbb{PG} = (\mathbb{G}, \mathbb{G}_T, e, p, g)$.
- Choose $\alpha, \beta, \gamma \in \mathbb{Z}_p$ and $h \in \mathbb{G}$ at random. Compute $e(g^\gamma, g)$.
- Compute $u_i = h^{\gamma \alpha^i}$ and $v_i = h^{\gamma \beta \alpha^i}$ for $i \in [0, n]$.

The public parameters PP are defined as $PP = (\mathbb{G}, \mathbb{G}_T, e, p, g, h, e(g^\gamma, g), u_0, \cdots, u_n, v_0, \cdots, v_n)$ and the master secret key MSK is defined as $MSK = (\alpha, \beta, \gamma)$.

KeyGen: On input the public parameters PP, the master secret key MSK and a user A_i for $i \in [1,n]$, the **KeyGen** algorithm works as follows. Randomly choose $x_i, s_i \in \mathbb{Z}_p$ and $f_i \in \mathbb{G}$. Compute the private key $SK_i = (d_{i0}, \cdots, d_{in},)$ as

$$d_{i0} = g^{-s_i}, d_{ii} = g^{\gamma} f_i^{s_i}, \text{for } i \neq j, d_{ij} = f_j^{s_i}.$$

Set the public key as $PK_i = (x_i + \alpha, f_i)$.

GroupGen: On input the public parameters PP, the group $G = \{A_1, \cdots, A_k\} \in \mathbb{Z}_p$ for any $1 \leq k \leq n$, and the public keys $\{PK_1, \cdots, PK_k\}$, the **GroupGen** algorithm randomly chooses $t_1, t_2 \in \mathbb{Z}_p$ and computes $P(G)$ as

$$\begin{aligned} P(G) &= (w_1, w_2, w_3, w_4, w_5, w_6) \\ &= (u_0^{t_1 \cdot \prod_{A_i \in G}(x_i+\alpha)}, v_0^{t_1 \cdot \prod_{A_i \in G}(x_i+\alpha)}, \\ &\quad v_{n-k}^{t_1 \cdot \prod_{A_i \in G}(x_i+\alpha)}, \prod_{A_i \in G} f_i^{t_2}, g^{t_2}, e(g^{\gamma}, g)^{t_2}) \\ &= (h^{\gamma t_1 \cdot \prod_{A_i \in G}(x_i+\alpha)}, h^{\gamma \beta t_1 \cdot \prod_{A_i \in G}(x_i+\alpha)}, \\ &\quad h^{\gamma \beta t_1 \alpha^{n-k} \cdot \prod_{A_i \in G}(x_i+\alpha)}, \prod_{A_i \in G} f_i^{t_2}, g^{t_2}, \\ &\quad e(g^{\gamma}, g)^{t_2}) \end{aligned}$$

Verify: On input $P(G)$ and k, accept $|G| \leq k$ if $e(w_2, u_n) = e(w_3, u_k)$.

Encrypt: On input the public parameters PP, a token $P(G) = (w_1, w_2, w_3, w_4, w_5)$ and a message $M \in \mathbb{G}_T$, the **Encrypt** algorithm works as follows:

- Verify that $w_2 = w_1^{\beta}$ by checking $e(w_1, v_0) = e(w_2, u_0)$.
- Randomly choose $r \in \mathbb{Z}_p$. Compute the ciphertext C on the message M as

$$\begin{aligned} C &= (c_1, c_2, c_3) \\ &= (w_5^r, w_4^r, M \cdot w_6^r) \\ &= (g^{rt_2}, \prod_{A_i \in G} f_i^{rt_2}, M \cdot e(g,g)^{rt_2 \gamma}) \end{aligned}$$

Decrypt: On input a user A_i for $i \in [1,n]$, the group of selected users G, the public/private key pair (PK_i, SK_i) and a ciphertext C, the **Decrypt** algorithm works as follows. Check the cardinality of the group G: if $|G| \leq k$ then proceed; otherwise, abort.

- Compute the pairing

$$\begin{aligned} e_1 &= e(d_{ii} \cdot \prod_{A_j \in G \setminus \{A_i\}} d_{ij}, c_1) \cdot e(d_{i0}, c_2) \\ &= e(g^{\gamma} \cdot f_i^{s_i} \cdot \prod_{A_j \in G \setminus \{A_i\}} f_j^{s_i}, g^{rt_2}) \cdot \\ &\quad e(g^{-s_i}, \prod_{A_j \in G} f_j^{rt_2}) \\ &= e(g^{\gamma}, g^{rt_2}) \cdot e(\prod_{A_j \in G} f_j^{s_i}, g^{rt_2}) \cdot \\ &\quad e(g^{-s_i}, \prod_{A_j \in G} f_j^{rt_2}) \\ &= e(g^{\gamma}, g^{rt_2}) \end{aligned}$$

- Compute M by

$$e_1^{-1} \cdot c_3 = e(g^{\gamma}, g^{rt_2})^{-1} \cdot M \cdot e(g^{\gamma}, g)^{rt_2} = M$$

5 Security Analysis

Prior to presenting our security proofs, we recall the following facts. The security of the Membership Encryption proposed in [20] relies on several properties, including the fact that the group token $P(G)$ is secure with maximum number accountability (assuming that the (f,n)-DHE problem is hard), and unconditionally preserves the privacy of all users in G. The Gentry-Waters Broadcast Encryption [18] is proved semi-statically secure under the decisional BDHE assumption.

5.1 Semi-Static Security

Intuitively, an adversary $\mathcal{A}$ must recover $e(g^{\gamma}, g)^{rt_2}$ in order to decrypt the ciphertext C. To do this, $\mathcal{A}$ must pair $c_1 = g^{rt_2}$ from the ciphertext with the components from some private keys SK_i of user $A_i \in S^*$, where S^* is the challenge set chose by $\mathcal{A}$. This will result in the desired value $e(g^{\gamma}, g)^{rt_2}$, which is hidden by $val = e(\prod_{A_i \in S^*} f_i^s, g^{rt_2})$. This value val can be retrieved if and only if the user has the correct key components to cancel out val. Since the hiding value val is randomized to the randomness s from private keys of users $A_i \in S^*$, the attack of $\mathcal{A}$ will be successful with negligible probability. Formally, it is shown as follows.

Theorem 1 (Semi-static security) *Let $\mathcal{A}$ be a semi-static adversary against the above system. Then, there is an algorithm $\mathcal{B}$, which runs in about the same time as $\mathcal{A}$, such that*

$$AdvBED_{\mathcal{A}}^{SS}(1^{\lambda}) = AdvBDHE_{\mathcal{B}}^{n}(1^{\lambda})$$

Proof $\mathcal{B}$ receives the problem instance, which includes g^s, Z, and the set $\{g^{a^i}; i \in [0,n] \cup [n+2, 2n]\}$.

Init. $\mathcal{A}$ commits to a set $S \subseteq \{A_1, \cdots, A_n\}$ of users.

Setup. First, $\mathcal{B}$ generates $y_0, \cdots, y_n \in_R \mathbb{Z}_p$, and sets $f_i = g^{y_i}$ for $A_i \in S$ and $f_i = g^{y_i + a^i}$ for $A_i \in \{A_1, \cdots, A_n\} \setminus S$. In addition, $\mathcal{B}$ sets $\gamma = y_0 \cdot a^{n+1}$. It picks at random $\alpha, \beta \in_R \mathbb{Z}_p$, and sets $u_i = h^{\gamma \alpha^i} = h^{y_0 a^{n+1} \alpha^i}$ and $v_i = h^{\gamma \beta \alpha^i} = h^{y_0 a^{n+1} \beta \alpha^i}$. It sets the public parameters as $PP = (\mathbb{G}, \mathbb{G}_T, e, p, g, h, e(g^\gamma, g), u_0, \cdots, u_n, v_0, \cdots, v_n)$, where $e(g^\gamma, g)$ can be computed as $e(g^a, g^{a^n})^{y_0}$.

Second, for $i \in [1, n]$, it randomly chooses x_i from $\mathbb{Z}_p$. It set the public key PK_i as $PK_i = (x_i + \alpha, f_i)$, Finally, $\mathcal{B}$ sends PP and PK_i to $\mathcal{A}$.

Private Keys Queries. $\mathcal{A}$ is allowed to query the private key only for users $A_i \in \{A_1, \cdots, A_n\} \setminus S$. To answer the query, $\mathcal{B}$ generates $z_i \in_R \mathbb{Z}_p$, sets $s_i = z_i - y_0 \cdot a^{n+1-i}$, and outputs $SK_i = (d_{i0}, \cdots, d_{in})$ where

$$d_{i0} = g^{-s_i}, d_{ii} = g^\gamma \cdot f_i^{s_i}, \text{for } j \neq i, d_{ij} = f_j^{s_i}.$$

$\mathcal{B}$ can compute all these components from the problem instance. For instance,

$$d_{ii} = g^\gamma \cdot f_i^{s_i} = g^{y_0 \cdot a^{n+1} + (y_i + a^i)(z_i - y_0 \cdot a^{n+1-i})}$$

can be computed since the a^{n+1} term in the exponent cancels out.

Challenge. $\mathcal{A}$ chooses a subset $S^* \subseteq S$, and $\mathcal{B}$ sets $s = rt_2$ for $r, t_2 \in \mathbb{Z}_p$, and $C = (c_1, c_2, c_3)$ where $c_1 = g^s$, $c_2 = \prod_{A_i \in S^*} f_i^s$ and $c_3 = M \cdot Z^{y_0}$. It sends C to $\mathcal{A}$.

$\mathcal{B}$ can compute these components from the problem instance, as c_1 and c_3 come directly from it. Moreover, since $\mathcal{B}$ knows $DL_g(f_i)$ for all $A_i \in S^*$, it can compute c_2 as $c_2 = \prod_{A_i \in S^*} f_i^s = \prod_{A_i \in S^*} g^{y_i \cdot s} = g^{s \cdot \sum A_i \in S^* y_j}$.

Guess. Finally, $\mathcal{A}$ outputs a bit b'. $\mathcal{B}$ sends b' to the challenger.

From $\mathcal{A}$'s point of view, $\mathcal{B}$'s simulation has exactly the same distribution as the semi-static game defined in [18]. The public and private keys are appropriately distributed, since γ, and the values $\{DL_g(f_i)\}$, $\{s_i\}$ are uniformly random and independent.

When $b = 0$ in the semi-static game, C is generated according to the same distribution as in the real scheme. In $\mathcal{B}$'s simulation, we verify that is also true: when $b = 0$, $c_3 = M \cdot e(g,g)^{\gamma \cdot s}$, and thus the challenge is valid ciphertext under the randomness $s = rt_2$. When $b = 1$ in the semi-static game, $C' = (c_1, c_2, c_3')$ is generated as in the real scheme, but c_3' is replaced by $c_3 = M \cdot c$ for $c \in_R \mathbb{G}_T$, and $C = (c_1, c_2, c_3)$ is sent to $\mathcal{A}$. This distribution is identical to that of $\mathcal{B}$'s simulation, where c_1 is valid for randomness $s = rt_2$, but $c = Z$ is a uniformly random element of $\mathbb{G}_T$.

From this, we see that $\mathcal{B}$'s advantage in deciding the BDHE problem instance is precisely $\mathcal{A}$'s advantage under the semi-static attack model.

5.2 Privacy

Theorem 2 (Privacy) *$P(G)$ unconditionally preserves the privacy of all users in G.*

Proof Let $P(G)$ be a group token generated from $G = \{A_1, A_2, \cdots, A_k\}$, for $k < n$, and t_1 be a random exponent in $\mathbb{Z}_p$. We have:

$$\begin{aligned} P(G) &= (w_1, w_2, w_3, w_4, w_5, w_6) \\ &= (h^{\gamma t_1 \cdot \prod_{A_i \in G}(x_i + \alpha)}, h^{\gamma \beta t_1 \cdot \prod_{A_i \in G}(x_i + \alpha)}, \\ &\quad h^{\gamma \beta t_1 \alpha^{n-k} \cdot \prod_{A_i \in G}(x_i + \alpha)}, \prod_{A_i \in G} f_i^{t_2}, g^{t_2}, \\ &\quad e(g^\gamma, g)^{t_2}) \end{aligned}$$

Since there exists $G' = \{A_1', A_2', \cdots, A_k'\}$ and $t_1', t_2' \in_R \mathbb{Z}_p$ satisfying

$$t_1 \cdot \prod_{A_i \in G} (x_i + \alpha) = t_1' \cdot \prod_{A_i' \in G'} (x_i' + \alpha)$$

$$\prod_{A_i \in G} f_i^{t_2} = \prod_{A_i' \in G'} f_i'^{t_2'}$$

we have $P(G)$ can be also seen as a group token generated for G', t_1' and t_2'. Therefore, the privacy of all users in $P(G)$ is unconditionally preserved.

5.3 Maximum Number Accountability

Before giving the formal proof, we first give some intuition as to why the scheme is secure with maximum number accountability. According to the setting of our construction, for k selected users, we have

$$\begin{aligned} P(G) &= (w_1, w_2, w_3, w_4, w_5, w_6) \\ &= (w_1, w_1^\beta, w_1^{\beta \alpha^{n-k}}, w_4, w_5, w_6). \end{aligned}$$

Through the **Verify** algorithm, the exponent of w_3 contains α^{n-k} which is known by the verifier. We have $w_1 = h^{\gamma t_1 \cdot \prod_{A_i \in G}(x_i+\alpha)}$ where the polynomial $\prod_{A_i \in G}(x_i + x)$ has at most k degrees. Otherwise, computing w_3 needs $h^{\gamma\beta\alpha^{n+1}}, \cdots, h^{\gamma\beta\alpha^{n'}}$ for $n' > n$ and these components are not generated during the **Setup** algorithm.

Theorem 3 (Maximum number accountability) *Suppose the (f,n)-DHE problem is hard, the group token $P(G)$ is secure with maximum number accountability.*

Proof Let $\mathcal{A}$ be an adversary who breaks the above system under maximum number accountability model. Then, there is an algorithm $\mathcal{B}$ that solves the (f,n)-DHE problem. $\mathcal{B}$ interacts with $\mathcal{A}$ as follows.

Challenge. Let $\mathbb{PG} = (\mathbb{G}, \mathbb{G}_T, e, p, g)$ be the pairing group. $\mathcal{B}$ works as follows to simulate the public parameters.

- It chooses β_0, γ_0 at random from $\mathbb{Z}_p$, and sets $\alpha = a, \beta = \beta_0, \gamma = \gamma_0$, where a is the randomness in the problem instance.
- It randomly picks $y \in \mathbb{Z}_p$, and sets $g = g_0, h = g_0^y$. Thus, we obtain:

$$e(g^\gamma, g) = e(g_0, g_0)^{y\gamma_0},$$
$$u_i = h^{\gamma\alpha^i} = (g_0^{a^i})^{y\gamma_0},$$
$$v_i = h^{\gamma\beta\alpha^i} = (g_0^{a^i})^{y\beta_0\gamma_0}.$$

- For $i \in [1,n]$, it randomly chooses s_i, x_i from $\mathbb{Z}_p$ and f_i from $\mathbb{G}$. It set the public key as $PK_i = (x_i + \alpha, f_i) = (x_i + a, f_i)$, and the private key as $SK_i = (d_{i0}, \cdots, d_{in})$, where $d_{i0} = g^{-s_i} = g_0^{-s_i}$, $d_{ii} = g^\gamma f_i^{s_i} = g_0^{\gamma_0} f_i^{s_i}$ and for $j \neq i$, $d_{ij} = f_j^{s_i}$.

All terms are computable from the problem instance. $\mathcal{B}$ generated the public parameters and the key pairs, and sends it to $\mathcal{A}$.

Win. $\mathcal{A}$ outputs $(P(G), G, t)$, where $G = \{A_1, \cdots, A_k\}$, but the **Verify** algorithm outputs $|G| < k$. Let $P(G) = (w_1, w_2, w_3, w_4, w_5, w_6)$. If the **Verify** algorithm outputs $|G| = k' < k$, then we write $P(G)$ as:

$$\begin{aligned} P(G) &= (w_1, w_2, w_3, w_4, w_5, w_6) \\ &= (w_1, w_1^\beta, w_1^{\beta\alpha^{n-k}}, w_4, w_5, w_6) \end{aligned}$$

Moreover, we write w_1 as $w_1 = h^{t_1 \cdot \prod_{A_i \in G}(\alpha + x_i)} = g_0^{y \cdot t_1 \cdot \prod_{A_i \in G}(a + x_i)}$.

$\mathcal{B}$ sets $f(x) = y\beta t_1 \cdot x^{n-k'} \prod_{A_i \in G}(x + x_i)$, which is a $(n+k-k')$-degree polynomial function in $\mathbb{Z}_p[x]$, and outputs $(f(x), w_3)$ as the solution to the (f,n)-DHE problem.

5.4 A Simple and Unsuccessful Attack by the Dealer

In addition to the security proofs we gave previously, we quickly describe a simple attack that the dealer should trigger.

The dealer could select $k \leq n$ users $G_1 = \{A_1, \cdots, A_k\}$ and to compute w_1, w_2, w_3 regarding G_1, but to add extra users $G_2 = \{A_{k+1}, \cdots, A_{k'}\}$, and to compute w_4, w_5, w_6 regarding $G_1 \cup G_2$. Therefore, the dealer is successful when the broadcaster verifies the cardinality of the groups, and the latter encrypts the message for the group $G_1 \cup G_2$, meaning for more than k selected users. However, when a user in G_1 tries to decrypt the ciphertext, he/she knows only the identities of other users in G_1 (if G_2 becomes public, then the user will discover the cheating process as the number of selected users is larger than k). Thus, the decryption fails as the message cannot be recovered due to the lack of knowledge of user identities in G_2.

More precisely, in **GroupGen**, the dealer selects two groups of users $G_1 = \{A_1, \cdots, A_k\}$ and $G_2 = \{A_{k+1}, \cdots, A_{k'}\}$ for $k \leq n$ and $k < k'$, and randomly chooses $t_1, t_2 \in \mathbb{Z}_p$. Then, he/she computes $P(G)$ as follows:

- for the values w_1, w_2, w_3, he/she considers the group of users G_1, meaning that

$$\begin{aligned} &(w_1, w_2, w_3) \\ &= (u_0^{t_1 \cdot \prod_{A_i \in G_1}(x_i+\alpha)}, v_0^{t_1 \cdot \prod_{A_i \in G_1}(x_i+\alpha)}, \\ &\quad v_{n-k}^{t_1 \cdot \prod_{A_i \in G}(x_i+\alpha)}) \\ &= (h^{\gamma t_1 \cdot \prod_{A_i \in G_1}(x_i+\alpha)}, h^{\gamma\beta t_1 \cdot \prod_{A_i \in G_1}(x_i+\alpha)}, \\ &\quad h^{\gamma\beta t_1 \alpha^{n-k} \cdot \prod_{A_i \in G_1}(x_i+\alpha)}) \end{aligned}$$

- for the values w_4, w_5, w_6, he/she considers the group of users $G_1 \cup G_2$, meaning that

$$\begin{aligned} (w_4, w_5, w_6) &= (\prod_{A_i \in G_1 \cup G_2} f_i^{t_2}, g^{t_2}, e(g^\gamma, g)^{t_2}) \\ &= (\prod_{A_i \in G_1 \cup G_2} f_i^{t_2}, g^{t_2}, e(g^\gamma, g)^{t_2}) \end{aligned}$$

Therefore, when running **Verify**, the broadcaster accepts as $e(w_2, u_n) = e(w_3, u_k)$ (this equation is obtained since $|G_1| \leq k$). However, when encrypting, the broadcaster verifies correctly that $w_2 = w_1^\beta$

and after have chosen $r \in \mathbb{Z}_p$, computes the ciphertext C as

$$\begin{aligned} C &= (c_1, c_2, c_3) \\ &= (w_5^r, w_4^r, M \cdot w_6^r) \\ &= (g^{rt_2}, \prod_{A_i \in G_1 \cup G_2} f_i^{rt_2}, M \cdot e(g,g)^{rt_2\gamma}) \end{aligned}$$

Thus, the broadcaster encrypts the message for more than k users. Finally, when running **Decrypt**, we have two cases:

– the user belongs to G_1. We recall that the user runs the **Verify** algorithm for the token $P(G)$ to make sure that G has the right cardinality. If he/she knows the identities of $G_1 \cup G_2$ then he/she aborts as $k < |G_1 \cup G_2| \leq k'$. If he/she only knows the identities in G_1 then he/she proceeds as $G_1 \leq k$ but cannot achieve to retrieve the message M as

$$\begin{aligned} e_1 &= e(d_{ii} \cdot \prod_{A_j \in G_1 \setminus \{A_i\}} d_{ij}, c_1) \cdot e(d_{i0}, c_2) \\ &= e(g^\gamma \cdot f_i^{s_i} \cdot \prod_{A_j \in G_1 \setminus \{A_i\}} f_j^{s_i}, g^{rt_2}) \cdot \\ &\quad e(g^{-s_i}, \prod_{A_j \in G_1 \cup G_2} f_j^{rt_2}) \\ &= e(g^\gamma, g^{rt_2}) \cdot e(\prod_{A_j \in G_1} f_j^{s_i}, g^{rt_2}) \\ &\quad \cdot e(g^{-s_i}, \prod_{A_j \in G_1 \cup G_2} f_j^{rt_2}) \\ &= e(g^\gamma, g^{rt_2}) \cdot e(g^{-s_i}, \prod_{A_j \in G_2} f_j^{rt_2}) \end{aligned}$$

– the user belongs to G_2. We recall that the user runs the **Verify** algorithm for the token $P(G)$ to make sure that G has the right cardinality. If he/she knows the identities of $G_1 \cup G_2$ then he/she aborts as $k < |G_1 \cup G_2| \leq k'$. If he/she only knows the identities in G_2 then he/she proceeds if $G_2 \leq k$ (otherwise, he aborts), but cannot achieve to retrieve the message M as

$$\begin{aligned} e_1 &= e(d_{ii} \cdot \prod_{A_j \in G_2 \setminus \{A_i\}} d_{ij}, c_1) \cdot e(d_{i0}, c_2) \\ &= e(g^\gamma \cdot f_i^{s_i} \cdot \prod_{A_j \in G_2 \setminus \{A_i\}} f_j^{s_i}, g^{rt_2}) \cdot \\ &\quad e(g^{-s_i}, \prod_{A_j \in G_1 \cup G_2} f_j^{rt_2}) \\ &= e(g^\gamma, g^{rt_2}) \cdot e(\prod_{A_j \in G_2} f_j^{s_i}, g^{rt_2}) \cdot \\ &\quad e(g^{-s_i}, \prod_{A_j \in G_1 \cup G_2} f_j^{rt_2}) \\ &= e(g^\gamma, g^{rt_2}) \cdot e(g^{-s_i}, \prod_{A_j \in G_1} f_j^{rt_2}) \end{aligned}$$

Eventually, after having aborted because of the wrong cardinality of the user group, the user reports to the broadcaster the possible misbehavior of the dealer.

Remarks: Identification of a cheating dealer. As mentioned previously, in our proposed framework, we adopt some practical assumptions that the dealer is indeed rational. If the dealer cheats, then the dealer will be blacklisted and he/she will be excluded from the future business. In the description of the decryption algorithm, this cheating behaviour can in fact be identified by the user. Notice that a valid user will have two cases upon conducting the decryption. For the second subcase of case 1, the user only knows the identities in G_1, and therefore the user can check whether $e(c_1, \prod_{A_i \in G1} f_i) \stackrel{?}{=} e(g, c_2)$ holds. If this equation does not hold, then c_2 must have been encrypted under a set that is not equal to G_1, and this may indicate that the dealer cheats. Similarly, the same method is applicable to the second sub-case of case 2, i.e. when the user only knows the identities of G_2. We note that we do not regard this as part of our proposed framework since in practice, this matter may not concern the user as long as he/she will be able to decrypt the contents.

6 Complexity Analysis and Evaluation of the Performance

6.1 Complexity Analysis

In Table 2, we present the complexity analysis of our scheme.

We notice that the algorithms **Setup** and **KeyGen** are linear in the number of users n. However,

Algorithm	Exp. in $\mathbb{G}$	Exp. in $\mathbb{G}_T$	Inv. in $\mathbb{G}_T$	Pairings
Setup	$2n+3$			1
KeyGen	$n+2$			
GroupGen	$k+4$	1		
Verify				2
Encrypt	3	1		1
Decrypt			1	2
Total	$3n+k$ $+12$	2	1	6

Table 2 Group exponentiation and pairing benchmarks. Exp. and Inv. denote the number of exponentiations and inversions in the groups, respectively. n and k denote the number of users and the number of selected users, respectively.

Algorithm	Exp. in $\mathbb{G}$	Exp. in $\mathbb{G}_T$	Inv. in $\mathbb{G}_T$	Pairings
Time (ms)/ computation	1.49	0.36	0.36	3.34
Setup	2984.47			3.34
KeyGen	1492.98			
GroupGen	154.96	0.36		
Verify				6.68
Encrypt	4.47	0.36		3.34
Decrypt			0.36	6.68

Table 3 Timings for symmetric and asymmetric pairing types and pairing-based systems. Times are in milliseconds (ms). Exp. and Inv. denote the number of exponentiations and inversions in the groups, respectively.

this does not impose a big problem since the computation cost of the algorithm **Setup** will only be invoked once. One may notice that an improvement to the scheme can be done towards the algorithm **KeyGen**, as this algorithm needs to be executed for every single user. Nevertheless, we also note that this algorithm will only be executed once for all the users involved. More importantly, the algorithm **GroupGen** could be more faster than the algorithms **Setup** and **KeyGen**, even if it is linear in the number of selected users k, supposed that $k << n$. The rest of algorithms are constant-time, which is ideal. Indeed, we can imagine that once the keys are generated for all users and the group of selected users is created, the system can run encryptions and decryptions fast and efficiently.

6.2 Evaluation of the Performance

In this section, we evaluate the efficiency of our scheme. We measure our results using the widely accepted cryptographic operation implementations (exponentiations and pairings) using the MIRACL library, provided by Certivox for the MIRACL Authentication Server Project Wiki. The following experiments were conducted with Borland C/C++ Compiler/Assembler and tested on a processor 2.4 GHz Intel i5 520M.

For symmetric pairing-based systems, AES with a 80-bit key and a Super Singular curve over $\mathbb{GF}_p$, for a 512-bit modulus p and an embedding degree equal to 2, are used.

In Table 3, we evaluate the efficiency of our scheme. Without losing generality, we assume that the number of users is $n = 1000$ and the number of selected users is $k = 100$.

7 Conclusion

In this paper, we introduced the notion of Broadcast Encryption scheme with Dealership. We presented a realistic scenario that requires this kind of cryptographic primitives. Furthermore, we discussed the security issues that raise in this kind of scenario. We presented a set of security models to capture these requirements, and subsequently presented a concrete scheme, which is provably secure under DBDHE and DHE assumptions. The follow up work from here is the effort on obtaining an efficient generic construction of such a scheme, and compare it with the concrete scheme that we presented in this paper.

References

1. M. H. Au, P. P. Tsang, W. Susilo, and Y. Mu, *Dynamic Universal Accumulators for DDH Groups and their Applications to Attribute-Based Anonymous Credential Systems*. In Fischlin, M. (ed.), CT-RSA 2009. LNCS, vol. 5473, pp. 295-308. Springer Heidelberg (2009)
2. J. Baek, R. Safavi-Naini, and W. Susilo, *Efficient Multi-Receiver Identity-Based Encryption and its Application to Broadcast Encryption*. In PKC 2005. LNCS, vol. 3386, pp. 380-397. Springer Heidelberg (2005)
3. D. Boneh, C. Gentry, and B. Waters, *Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys*. In CRYPTO 2005. LNCS, vol. 3621, pp. 258-275. Springer Heidelberg (2005)
4. J. C. Benaloh, and M. de Mare, *One-Way Accumulators: a Decentralized Alternative to Digital Signatures*. In Helleseth, T. (ed.), EUROCRYPT 1993. LNCS, vol. 765, pp. 274-285. Springer Heidelberg (1994)
5. N. Barić, and B. Pfitzmann, *Collision-free Accumulators and Fail-stop Signature Schemes without Trees*. In Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 480-494. Springer Heidelberg (1997)
6. D. Boneh, A. Sahai, and B. Waters, *Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys*. In EUROCRYPT 2006. Springer Heidelberg (2006)

7. D. Boneh, and B. Waters, *A Fully Collusion Resistant Broadcast, Trace, and Revoke System.* In CCS 2006. ACM (2006)
8. J. Camenisch, R. Chaabouni, and A. Shelat, *Efficient Protocols for Set Membership and Range Proofs.* In Pieprzyk, J. (ed.), ASIACRYPT 2008. LNCS, vol. 5350, pp. 440-456. Springer Heidelberg (2008)
9. J. Camenisch, M. Kohlweiss, and C. Soriente, *An Accumulator based on Bilinear Maps and Efficient Revocation of Anonymous Credentials.* In Jarecki, S., Tsudik, G. (eds.), PKC 2009. LNCS, vol. 5443, pp. 481-500. Springer Heidelberg (2009)
10. J. Camenisch, and A. Lysyanskaya, *Dynamic Accumulators and Application to Efficient Revocation of Anonymous Credentials.* In Yung, M. (ed.), CRYPTO 2002. LNCS, vol. 2442, pp. 61-76. Springer Heidelberg (2002)
11. R. Cramer, I. Damgard, and B. Schoenmakers, *Proofs of Partial Knowledge and Simplified Design of Witness Hiding Protocols.* In Desmedt, Y.G. (ed.), CRYPTO 1994. LNCS, vol. 839, pp. 174-187. Springer Heidelberg (1994)
12. C. K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, *Conditional Proxy Broadcast Re-Encryption.* In ACISP 2006. LNCS, vol. 5594, pp. 327-342. Springer Heidelberg (2006)
13. C. Delerabrée, *Identity-Based Broadcast Encryption woth Constant Size Ciphertext and Private Keys.* In ASIACRYPT 2007. LNCS, pp. 200-215. Springer Heidelberg (2007)
14. C. Delerabrée, P. Paillier, and D. Pointcheval, *Fully Collusion Secure Dynamic Broadcast Encryption with Constant-Ciphertexts or Decryption Keys.* In Pairing 2007. LNCS, vol. 4575, pp. 39-59. Springer Heidelberg (2007)
15. Y. Dodis, and N. Fazio, *Public Key Broadcast Encryption for Stateless Receivers.* In DRM 2002. LNCS, vol. 2696, pp. 61-80. Springer Heidelberg (2002)
16. Y. Dodis, and N. Fazio, *Public Key Trace and Revoke Scheme Secure against Adaptive Chosen Ciphertext Attack.* In PKC 2003. LNCS, vol. 2567, pp. 100-115. Springer Heidelberg (2003)
17. A. Fiat, and M. Naor, *Broadcast Encryption.* In CRYPTO 1993. LNCS, vol. 773, pp. 480-491. Springer Heidelberg (1993)
18. C. Gentry, and B. Waters, *Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts).* In EUROCRYPT 2009. LNCS, vol. 5479, pp. 171-188. Springer Heidelberg (2009)
19. F. Guo, Y. Mu, and Z. Chen, *Mutative Identity-Based Signatures or Dynamic Credentials without Random Oracles.* In CANS 2007. LNCS, vol. 4856, pp. 1-14. Springer Heidelberg (2007)
20. F. Guo, Y. Mu, W. Susilo, and V. Varadharajan, *Membership Encryption and Its Applications.* In ACISP 2013. LNCS, vol. 7959, pp. 219-234. Springer Heidelberg (2013)
21. D. Halevy, and A. Shamir, *The LSD Broadcast Encryption Scheme.* In CRYPTO 2002. LNCS, vol. 2442, pp. 47-60, Springer Heidelberg (2002)
22. M. Jafari, R. Safavi-Naini, and N. P. Sheppard, *A Rights Management Approach to Protection of Privacy in a Cloud of Electronic Health Records.* In DRM 2011. ACM, pp. 23-30 (2011)
23. L. Korba and S. Kenny, *Applying Digital Rights Management Systems to Privacy Rights Management.* In Digital Rights Management Workshop 2002. LNCS, pp. 118-136, Springer Heidelberg (2002)
24. Q. Liu, R. Safavi-Naini, and N. P. Sheppard, *Digital Rights Management for Content Distribution.* In AISW 2003. Research and Practice in Information Technology, vol. 21 (2002)
25. D. Naor, M. Naor, and J. Lotspiech, *Revocation and Tracing Schemes for Stateless Receivers.* In CRYPTO 2001. LNCS, vol. 5479, pp. 41-62. Springer Heidelberg (2001)
26. L. Nguyen, *Accumulators from Bilinear Pairings and Applications.* In Menezes, A. (ed.), CT-RSA 2005. LNCS, vol. 3376, pp. 275-292. Springer Heidelberg (2005)
27. D. H. Phan, D. Pointcheval, S. F. Shahandashti, and M. Strefler, *Security Notions for Broadcast Encryption.* In ACNS 2011. LNCS, vol. 6715, pp. 377-394, Springer Heidelberg (2011)
28. D. H. Phan, D. Pointcheval, and M. Strefler, *Decentralized Dynamic Broadcast Encryption.* In SCN 2012. LNCS, vol. 7485, Springer Heidelberg (2011)
29. D. H. Phan, D. Pointcheval, and M. Strefler, *Adaptive CCA Broadcast Encryption with Constant-Size Secret Keys and Ciphertexts.* In ACISP 2012. LNCS, vol. 7372, pp. 308-321, Springer Heidelberg (2012)
30. D. H. Phan, D. Pointcheval, and V. C. Trinh, *Multi-Channel Broadcast Encryption.* In ASIACCS 2013. ACM (2013)
31. B. C. Popescu, B. Crispo, A. S. Tanenbaum, and F. L.A.J. Kamperman, *A DRM Security Architecture for Home Networks.* In DRM 2004. ACM, pp. 1-10 (2004)
32. F. Salim, N. P. Sheppard, and R. Safavi-Naini, *Enforcing P3P Policies Using a Digital Rights Management System.* In PET 2007. LNCS, vol. 4776, pp. 200-217, Springer Heidelberg (2013)
33. B. Waters, *Dual System Encryption: Realizing Fully Secure IBE and HIBE under Simple Assumptions.* In CRYPTO 2009. ACM, pp. 619-636 (2009)
34. J. Weng, R. H. Deng, X. Ding, C. K. Chu, and J. Lai, *Conditional proxy re-encryption secure against chosen-ciphertext attack.* In ASIACCS 2009. ACM, pp. 322-332 (2009)