Bridging Broadcast Encryption and Group Key Agreement

Qianhong Wu[1,2], Bo Qin[1,3], Lei Zhang[4],
Josep Domingo-Ferrer[1], and Oriol Farràs[1,5]

[1] Universitat Rovira i Virgili, Department of Computer Engineering and Mathematics, UNESCO Chair in Data Privacy, Tarragona, Catalonia
{qianhong.wu,bo.qin,josep.domingo,oriol.farras}@urv.cat
[2] Key Lab. of Aerospace Information Security and Trusted Computing, Ministry of Education School of Computer, Wuhan University, China
[3] Dept. of Maths, School of Science, Xi'an University of Technology, China
[4] Software Engineering Institute, East China Normal University, Shanghai, China
leizhang@sei.ecnu.edu.cn
[5] Department of Computer Science, Ben Gurion University, Be'er-Sheva, Israel

Abstract. Broadcast encryption (BE) schemes allow a sender to securely broadcast to any subset of members but requires a trusted party to distribute decryption keys. Group key agreement (GKA) protocols enable a group of members to negotiate a common encryption key via open networks so that only the members can decrypt the ciphertexts encrypted under the shared encryption key, but a sender cannot exclude any particular member from decrypting the ciphertexts. In this paper, we bridge these two notions with a hybrid primitive referred to as contributory broadcast encryption (CBE). In this new primitive, a group of members negotiate a common public encryption key while each member holds a decryption key. A sender seeing the public group encryption key can limit the decryption to a subset of members of his choice. Following this model, we propose a CBE scheme with short ciphertexts. The scheme is proven to be fully collusion-resistant under the decision n-Bilinear Diffie-Hellman Exponentiation (BDHE) assumption in the standard model. We also illustrate a variant in which the communication and computation complexity is sub-linear with the group size. Of independent interest, we present a new BE scheme that is aggregatable. The aggregatability property is shown to be useful to construct advanced protocols.

Keywords: Broadcast encryption; Group key agreement; Contributory broadcast encryption; Provable Security.

1 Introduction

With the fast advance and pervasive deployment of the communication technologies, there is an increasing demand of versatile cryptographic primitives to protect modern communication and computation platforms. These new platforms, including instant-messaging tools, collaborative computing, mobile *ad hoc*

D.H. Lee and X. Wang (Eds.): ASIACRYPT 2011, LNCS 7073, pp. 143–160, 2011.

networks and social networks, allow exchanging data within *any subset* of their users. These new information technologies provide potential opportunities for organizations and individuals. For instance, the users of a social network may wish to share their private photos/videos with their friends; scientists from different places may want to collaborate in a research project by means of an insecure third-party platform.

These new applications call for cryptographic primitives allowing a sender to securely encrypt to any subset of the users of the services without relying on a fully trusted dealer. Broadcast encryption (BE) [15] is a well-studied primitive intended for secure group-oriented communications. It allows a sender to securely broadcast to any subset of the group members. Nevertheless, its security heavily relies on a trusted key server to generate and distribute secret decryption keys for the members; both the sender and the receivers must fully trust the key server who can read all communications to any subset of the group members.

Group key agreement (GKA) [20] is another well-established primitive to secure group-oriented communications. A conventional GKA protocol allows a group of members to establish a common secret key via open networks. However, whenever a sender wants to broadcast to a group, he must first join the group and run a GKA protocol to share a secret key with the intended members. To overcome this limitation, Wu *et al.* recently introduced asymmetric GKA [32] in which only a common group public key is negotiated and each group member holds a different decryption key. However, neither conventional symmetric GKA nor newly-introduced asymmetric GKA allows *the sender* to exclude any particular member *on demand*[1]. Hence, it is essential to find more flexible cryptographic primitives allowing dynamic broadcasts without a fully trusted dealer.

1.1 Our Contributions

In this paper we present the Contributory Broadcast Encryption (CBE) primitive, which is a hybrid of GKA and BE. The new cryptographic primitive is motivated by the emerging communication and computation platforms. In CBE, a group of members contribute to the public group encryption key, and a sender can securely broadcast to any subset of the group members chosen in an *ad hoc* way. Specifically, our main contributions can be summarized as follows.

First, we present a model of CBE and formalize its security definitions. CBE incorporates the underlying ideas of GKA and BE. In the set-up stage of a CBE scheme, a group of members interact via open networks to negotiate a common encryption key while each member holds a different secret decryption key. Using the common encryption key, anyone can encrypt any message to any subset of the group members and only the intended receivers can decrypt. Unlike GKA, CBE allows the sender to exclude some members from reading the ciphertexts.

[1] Dynamic GKA equipped with a leave sub-protocol allows a sender to exclude some members from decrypting ciphertexts. In this case, the sender has to negotiate with the remaining members for their agreement to run the leave sub-protocol. The sender cannot exclude any member on his own demand.

Compared to BE, CBE does not need a fully trusted third party to set up the system. We formalize collusion resistance by defining an attacker who can adaptively corrupt some members during the set-up stage and can also query the decryption keys of the group members after the system is set up. Even if the attacker fully controls all members outside the intended receivers, she cannot extract useful information from the ciphertext. A trivial CBE scheme can be constructed by concurrently encrypting to each member with her/his regular public key. Unfortunately, the trivial solution incurs a heavy encryption cost and produces linear-size ciphertexts. The challenge is to design CBE schemes with efficient encryption and short ciphertexts.

Second, we present the notion of aggregatable broadcast encryption (ABE) and construct a concrete ABE scheme. The construction is based on the newly introduced aggregatable signature-based broadcast (ASBB) primitive [32]. Our ABE construction is tightly proven to be fully collusion-resistant under the decision BDHE assumption, and offers short ciphertexts and efficient encryption. Further, the proposed ABE scheme is equipped with aggregatability, which means that different instances of the ABE scheme can be aggregated into a new instance. We observe that the BE schemes in the literature are not aggregatable. However, the aggregatability of ABE schemes seems very useful to design advanced protocols, as illustrated in the construction of our CBE scheme.

Finally, we construct an efficient CBE scheme with our ABE scheme as a building block. The CBE construction is proven to be semi-adaptively secure under the decision BDHE assumption in the standard model. Only one round is required to establish the public group encryption key and set up the CBE system. After the system set-up, the storage cost of both the sender and the group members is $O(n)$, where n is the number of group members participating in the set-up stage. However, the online complexity (which dominates the practicality of a CBE scheme) is very low. Indeed, at the sender's side, the encryption needs only $O(1)$ exponentiations and generates $O(1)$-size ciphertexts; and at the receivers' side, the decryption requires only $O(1)$ exponentiations and $O(1)$ bilinear map operations. We also illustrate a trade-off between the set-up complexity and the online performance. After the trade-off, the variant has $O(n^{2/3})$ complexity in communication, computation and storage. This is comparable to up-to-date regular BE schemes which have $O(n^{1/2})$ complexity in the same performance metrics, but our scheme does not require a trusted key dealer. As a versatile GKA scheme, our CBE does not require additional rounds to enable a *new sender* to broadcast to the group members or to let a sender *revoke* any subset of group members. These features are desirable for applications in which the sender and the group members may change frequently.

1.2 Related Work

Considerable efforts have been devoted to protect group communications. Among them, the most prominent notions are key agreement and broadcast encryption. Since the inception of the Diffie-Hellman protocol [14] in 1976, a number of proposals have addressed key agreement protocols for multiple parties. The schemes

due to Ingemarsson *et al.* [20] and Steiner *et al.* [29] are designed for n parties and require $O(n)$ rounds. Tree key structures have been further proposed and reduced the number of rounds to $O(\log n)$ [23, 24, 27]. A multi-round GKA protocol poses a synchronism requirement on group members and it needs all group members to simultaneously stay online to complete the protocol. Several proposals (*e.g.*, [8, 18, 30]) have been motivated to optimize round complexity in GKA protocols. Burmester and Desmedt [12] proposed a two-round n-party GKA protocol for n parties. The Joux protocol [21] is one-round and only applicable to three parties. The work of Boneh and Silverberg [5] shows that a one-round $(n+1)$-party GKA protocol can be constructed from n-linear pairings. However, it remains unknown whether there exist n-linear pairings for $n > 2$.

Dynamic GKA protocols provide extra mechanisms to cope with member changes. Bresson *et al.* [9, 10] extended the protocol in [11] to dynamic GKA protocols which allow members to leave and join the group. The number of rounds in `set-up/join` algorithms of their protocols [9, 10] is linear with the group size, but the number of rounds in the `leave` algorithm is constant. The theoretical analysis [28] proves that, for any tree-based group key agreement scheme, the lower bound of the worst-case cost is $O(\log n)$ rounds for a member to join or leave. Without relying on a tree-based structure, Kim *et al.* [22] proposed a two-round dynamic GKA protocol. Recently, Abdalla *et al.* [1] presented a two-round dynamic GKA protocol in which only one round is required to cope with the change of members if they are in the initial group. Observing that existing GKA protocols cannot handle sender changes efficiently, Wu *et al.* presented the notion of asymmetric GKA [32] to support sender changes and their instantiated protocol allows anyone to securely broadcast to the group members.

BE is another well-established cryptographic primitive developed for secure group communications. BE schemes in the literature can be classified into two categories, *i.e.*, symmetric-key BE and public-key BE. In the symmetric-key setting, only the trusted center generates all the secret keys and broadcasts messages to users. Hence, only the key generation center can be the broadcaster or the sender. Fiat and Naor [15] first formalized broadcast encryption in the symmetric-key setting and proposed a systematic BE method. Similarly to the GKA setting, tree-based key structures were subsequently proposed to improve efficiency in symmetric-key BE systems [19, 31]. The state of the art along this research line is presented in [13].

Public-key BE schemes are more flexible in practice. In this setting, in addition to the secret keys for each user, the trusted center also generates a public key for all the users so that any one can play the role of a broadcaster or sender. Naor and Pinkas presented in [25] the first public-key BE scheme in which up to a threshold of users can be revoked. If more than this threshold of users are revoked, the scheme will be insecure and hence not fully collusion-resistant. Subsequently, by exploiting newly developed bilinear pairing technologies, a fully collusion-resistant public-key BE scheme was presented in [3] which has $O(\sqrt{n})$ complexity in key size, ciphertext size and computation cost. A recent scheme [26] slightly reduces the size of the key and the ciphertexts, although it still has sub-

linear complexity. The schemes presented in [4, 6, 17] strengthen the security concept of public-key BE schemes. However, as to performance, the sub-linear barrier $O(\sqrt{n})$ has not yet been broken.

Although both GKA and BE are used to secure group communications, they have very different features as they were initially developed for different types of group-oriented applications. First, GKA can be applied to *ad hoc* groups where there is no fully trusted party while BE is usually deployed to secure group communications where a fully trusted third party is available. Second, the encryption key in GKA protocols is usually established by group members in a contributory way, regardless of conventional symmetric GKAs or newly-introduced asymmetric GKAs. On the contrary, the encryption key in BE schemes is usually generated by a centralized key server. Third, the secret decryption key in GKA protocols is computed by each member with public inputs from other members and his/her own private inputs. Contrary to GKA protocols, the decryption key of each member in BE schemes is assigned by the dealer, which implies that the dealer can read all communications to any subset of the group members and n secure unicast channels have to be established before a BE scheme is set up. Finally, in a GKA protocol group members need to interact to update their keys if the membership changes, which implies that a sender cannot exclude some members from reading the ciphertexts. Unlike GKA, BE supports a much more flexible revocation mechanism. It allows a sender to choose the intended receivers on demand to read the ciphertexts. This revocation mechanism does not require cooperation among group members or extra interactions between the dealer and the group members. For the newly-emerging applications, the contributory feature of GKA protocols is desirable but GKA protocols do not allow a sender to exclude receivers from reading specific ciphertexts on demand; the flexible revocation mechanism of BE schemes is desirable but BE schemes heavily relies on a fully trusted authority that is hard to implement in the motivated scenarios. These observations inspire us to investigate more versatile cryptographic primitives to bridge the gap.

1.3 Paper Organization

The rest of the paper is organized as follows. In Section 2, we model CBE and define its security. In Section 3, we present a collusion-resistant regular public-key BE scheme with aggregatability. Efficient CBE schemes are realized in Section 4, and Section 5 concludes the paper.

2 Modeling Contributory Broadcast Encryption

We begin by formalizing the CBE notion bridging the GKA and BE primitives. In CBE, a group of members first jointly establish a public encryption key, then a sender can freely select which subset of the group members can decrypt the ciphertext. Our definition incorporates the up-to-date definitions of GKA [32] protocols and BE [3] schemes. Since the negotiated public key is usually employed

to transmit session keys, we define a CBE scheme as a key encapsulation mechanism (KEM). Knowing this public encryption key, anyone can send a session key ξ to any subset of the initial group members. Only the intended receivers can extract ξ. Even if all the outsiders including group members not in the intended subset collude, they receive no information about ξ.

2.1 Syntax

We first define the algorithms that compose a CBE scheme. Let $\lambda \in \mathbb{N}$ denote the security parameter. Suppose that a group of members $\{\mathcal{U}_1, \cdots, \mathcal{U}_n\}$ wants to jointly establish a CBE system, where n is a positive integer and each member $\mathcal{U}_i$ is indexed by i for $1 \leq i \leq n$. We focus on bridging BE and GKA and we assume that the communications between members are authenticated, but we do not further elaborate on the authentication of the group members. Formally, a CBE scheme is a tuple $\mathcal{CBE}$ =(ParaGen, CBSetup, CBEncrypt, CBDecrypt) of polynomial-time algorithms defined as follows.

ParaGen(1^λ). This algorithm is used to generate global parameters. It takes as input a security parameter λ and it outputs the system parameters, including the group size n.

CBSetup($\mathcal{U}_1(x_1), \cdots, \mathcal{U}_n(x_n)$). This interactive algorithm is jointly run by members $\mathcal{U}_1, \cdots, \mathcal{U}_n$ to set up a BE scheme. Each member $\mathcal{U}_i$ takes private input x_i (and her/his random coins representing the member's random inner state information). The communications between members go through public but authenticated channels. The algorithm will either abort or successfully terminate. If it terminates successfully, each user $\mathcal{U}_i$ outputs a decryption key dk_i securely kept by the user and a common group encryption key gek shared by all group members. The group encryption gek is publicly accessible. If the algorithm aborts, it outputs NULL. Here, we leave the input system parameters implicitly. We denote this procedure by $(\mathcal{U}_1(dk_1), \cdots, \mathcal{U}_n(dk_n); gek) \leftarrow \mathsf{CBSetup}(\mathcal{U}_1(x_1), \cdots, \mathcal{U}_n(x_n))$.

CBEncrypt($\mathbb{R}, gek$). This group encryption algorithm is run by a sender who is assumed to know the public group encryption key. The sender may or may not be a group member. The algorithm takes as inputs a receiver set $\mathbb{R} \subseteq \{1, \cdots, n\}$ and the public group encryption key gek, and it outputs a pair $\langle c, \xi \rangle$, where c is the ciphertext and ξ is the secret session key in a key space $\mathbb{K}$. Then $(c, \mathbb{R})$ is sent to the receivers.

CBDecrypt($\mathbb{R}, j, dk_j, c$). This decryption algorithm is run by each intended receiver. It takes as inputs the receiver set $\mathbb{R}$, an index $j \in \mathbb{R}$, the receiver's decryption key dk_j, a ciphertext c, and it outputs the secret session key ξ.

2.2 Security Definitions

The correctness of a CBE scheme means that if all members and the sender follow the scheme honestly, then the members in the receiver set can always correctly decrypt. Formally, the *correctness* of a CBE scheme is defined as follows.

Definition 1 (Correctness). *A CBE scheme is correct if for any parameter* $\lambda \in \mathbb{N}$ *and any element* ξ *in the session key space,* $(\mathcal{U}_1(dk_1), \cdots, \mathcal{U}_n(dk_n); gek) \leftarrow \mathsf{CBSetup}(\mathcal{U}_1(x_1), \cdots, \mathcal{U}_n(x_n))$*, and* $(c, \xi) \leftarrow \mathsf{CBEncrypt}(\mathbb{R}, gek)$*, it holds that* $\mathsf{CBDecrypt}(\mathbb{R}, j, dk_j, c) = \xi$ *for any* $j \in \mathbb{R}$*.*

We next define the secrecy of a CBE scheme. In the above, to achieve better practicality, a CBE scheme is modeled as a KEM in which a sender sends a (short) secret session key to the intended receivers and simultaneously, (long) messages can be encrypted using a secure symmetric encryption algorithm with the session key. Hence, we define the secrecy of a CBE scheme by the indistinguishability of the encrypted session key from a random element in the session key space. Since there exist standard conversions (*e.g.*, [16]) from secure KEM against chosen-plaintext attacks (CPA) to secure encryption against adaptively chosen-ciphertext attacks (CCA2), it is sufficient to only define the CPA secrecy of CBE schemes. However, noting that CBE is designed for distributed applications where the users are likely to be corrupted, we include full collusion resistance into our secrecy definition.

The fully collusion-resistant secrecy of a CBE scheme is defined by the following secrecy game between a challenger $\mathcal{CH}$ and an attacker $\mathcal{A}$. The secrecy game is defined as follows.

Initial. The challenger $\mathcal{CH}$ runs $\mathsf{ParaGen}$ with a security parameter λ and obtains the system parameters. The system parameters are given to the attacker $\mathcal{A}$.

Queries. The attacker $\mathcal{A}$ can make the following queries to challenger $\mathcal{CH}$.

- **Execute.** The attacker $\mathcal{A}$ uses the identities of n members $\mathcal{U}_1, \cdots, \mathcal{U}_n$ to query the challenger $\mathcal{CH}$. The challenger runs $\mathsf{CBSetup}(\mathcal{U}_1(x_1), \cdots, \mathcal{U}_n(x_n))$ on behalf of the n members, and responds with the group encryption key gek and the transcripts of $\mathsf{CBSetup}$ to the attacker $\mathcal{A}$.
- **Corrupt.** The attacker $\mathcal{A}$ sends i to the $\mathsf{Corrupt}$ oracle maintained by the challenger $\mathcal{CH}$, where $i \in \{1, \cdots, n\}$. The challenger $\mathcal{CH}$ returns the private input and inner random coins of $\mathcal{U}_i$ during the execution of $\mathsf{CBSetup}$.
- **Reveal.** The attacker $\mathcal{A}$ sends i to the Reveal oracle maintained by the challenger $\mathcal{CH}$, where $i \in \{1, \cdots, n\}$. The challenger $\mathcal{CH}$ responds with dk_i, which is the decryption key of $\mathcal{U}_i$ after execution of $\mathsf{CBSetup}$.

Challenge. At any point, the attacker $\mathcal{A}$ can choose a target set $\mathbb{R}^* \subseteq \{1, \cdots, n\}$ to attack, with a constraint that the indices in $\mathbb{R}^*$ have never been queried to the $\mathsf{Corrupt}$ oracle or the Reveal oracle. Receiving $\mathbb{R}^*$, the challenger $\mathcal{CH}$ randomly selects $\rho \in \{0, 1\}$ and responds with a challenge ciphertext c^*, where c^* is obtained from $(c^*, \xi) \leftarrow \mathsf{CBEncrypt}(\mathbb{R}, gek)$ if $\rho = 1$, else if $\rho = 0$, c^* is randomly sampled from the image space of $\mathsf{CBEncrypt}$.

Output. Finally, $\mathcal{A}$ outputs a bit ρ', its guess of ρ. The adversary wins if $\rho' = \rho$.

We define $\mathcal{A}$'s advantage $Adv^{secrecy-fc}_{CBE,\mathcal{A}}$ in winning the above fully collusion-resistant secrecy game as

$$Adv^{secrecy-fc}_{CBE,\mathcal{A}} = |\Pr[\rho = \rho'] - 1/2|.$$

Definition 2. *An n-party CBE scheme has adaptive (τ, n, ϵ)-secrecy against a full-collusion attack if there is no adversary $\mathcal{A}$ which runs in time at most τ and has advantage $Adv_{CBE,\mathcal{A}}^{secrecy-fc}$ at least ϵ in the above secrecy game. An n-party CBE scheme has semi-adaptive (τ, n, ϵ)-secrecy against a full-collusion attack if, for any attacker $\mathcal{A}'$ running in time τ, $\mathcal{A}'$'s advantage $Adv_{CBE,\mathcal{A}'}^{secrecy-fc}$ is less than ϵ in the above secrecy game, with extra constraints that $\mathcal{A}'$ (1) must commit to a set of indices $\tilde{\mathbb{R}} \subseteq \{1, \cdots, n\}$ before the* Queries *stage, (2) can only query* Corrupt *and* Reveal *with $i \notin \tilde{\mathbb{R}}$ and (3) can only choose $\mathbb{R}^* \subseteq \tilde{\mathbb{R}}$ to query $\mathcal{CH}$ in the* Challenge *stage.*

The above definition captures the full collusion resistance since the attacker is allowed to access the `Corrupt` and `Reveal` oracles. The `Corrupt` oracle is used to model an attacker who compromises some members during the set-up stage to establish the group encryption key. The `Corrupt` oracle is used to capture the decryption key leakage after the CBE system has been established. This difference can be used to differentiate the secrecy against attacks during the set-up stage from the secrecy against attacks after a CBE system is deployed.

2.3 Remarks on Complexity Bounds of CBE and BE Schemes

Before concrete CBE schemes are constructed, it is meaningful to examine the complexity bound of a CBE scheme for the purpose of guiding the design of CBE schemes.

A CBE scheme consists of an offline stage (consisting of ParaGen and CBSetup) to establish the group encryption key and an online stage enabling a sender to securely encrypt to intended receivers. Since CBE allows to revoke members, the members do not need to reassemble for a new run of the CBSetup procedure until some new members join. This implies that the practicality of a CBE scheme critically depends on the overheads of the CBEncrypt and CBDecrypt procedures for online encryption of session keys and decryption of ciphertexts. Hence, special efforts should be devoted to improve this online performance.

It is easy to see that there exists a trivial construction of CBE schemes. A group of n members independently generate public/secret key pairs in a standard public-key cryptosystem. The public group encryption key is a concatenation of each member's public key, and each member's decryption key is his/her secret key. To broadcast to a subset of the members, a sender first encrypts the session key using each member's public key and obtains the CBE ciphertext by concatenating the generated n ciphertexts in the underlying public-key cryptosytems. This trivial CBE has $n\tau_{\mathtt{PKE}}$ online encryption cost, $n\ell_{\mathtt{PKC}}$-size ciphertext, where $\ell_{\mathtt{PKC}}$ is the binary length of the ciphertext in the standard public-key cryptosystem, and $\tau_{\mathtt{PKE}}$ is the time to perform a standard public-key encryption operation. Hence, the upper bound of online complexity of a CBE scheme is $O(n)$.

We next analyze whether there exist CBE schemes with online complexity less than $O(n)$. From the definition of CBEncrypt, a sender has to read the indices in $\mathbb{R} \subseteq \{1, \cdots, n\}$ and perform some operations involving each index. This implies that the CBEncrypt procedure has a cost $|\mathbb{R}|\tau_{\mathtt{CEO}}$, where $|\mathbb{R}| = n$ in the worst

case and $\tau_{\texttt{CEO}}$ is the time to perform a basic cryptographic encryption operation involving each index. Also, the sender needs to send $(c, \mathbb{R})$ to the receivers. This requires $\ell_c + n$ bits, where ℓ_c is the binary size of the CBE ciphertext. The analysis shows that the lower bound of the online complexity of a CBE scheme is also $O(n)$.

From the above analysis, it would seem that no better than a trivial CBE can be done. However, a closer look shows this is not the case. First, a well-designed CBE can be more efficient than a trivial CBE if $\tau_{\texttt{CEO}} \ll \tau_{\texttt{PKE}}$ and the performance difference can be further amplified by the factor n. Second, $\ell_{\texttt{PKC}}$ is usually hundreds to thousands, thus a trivial CBE may consume hundreds to thousands times more bits than an elegantly-developed CBE if ℓ_c is independent of the group size n. Hence, the efforts to achieve non-trivial CBE schemes are meaningful in practice.

To highlight this point, we further look at regular public-key BE schemes. The definitions of encryption and decryption in our CBE are exactly the same as those of standard public-key BE schemes [3]. Hence, the above online complexity bounds also apply to regular BE systems. Furthermore, by slightly modifying the above trivial CBE, one can also obtain a trivial public-key BE scheme. To strictly follow the public-key BE definition, one just needs to let a trusted key dealer generate the public/secret key pairs for all members. The rest is the same as the trivial CBE. This implies that a trivial public-key BE scheme has exactly the same asymptotical complexity as the trivial one. However, as discussed above, it is still meaningful to construct non-trivial public-key BE schemes. Indeed, this work has attracted a lot of attention and numerous efforts (*e.g.*, [3, 4, 6, 26, 17]) have been devoted to reduce the ℓ_c size and the $\tau_{\texttt{CEO}}$ complexity. We do a parallel work in the CBE setting.

3 An Aggregatable BE Scheme

Previously, aggregatability was mainly considered in the signature setting [7] and exploited to reduce the signature verification time and the storage overhead when numerous signatures need to be verified and stored. In [32], Wu *et al.* first presented the ASBB notion and considered aggregatability in the static BE setting. In this section, we integrate aggregatability into dynamic BE schemes and instantiate an aggregatable BE (ABE) scheme.

3.1 Review of Aggregatable Signature-Based Broadcast

Our ABE scheme is based on the ASBB primitive [32]. An ASBB scheme consists of the algorithms *ParaGen*, *KeyGen*, *Sign*, *Verify*, *Encrypt* and *Decrypt*. *ParaGen* takes as input a security parameter λ and outputs the public parameters π. *KeyGen* takes input π and outputs a public/secret key pair (pk, sk). *Sign* takes as input the key pair (pk, sk) and a string s, and outputs a signature $\sigma(s)$. *Verify* takes as input the public key pk and the signature $\sigma(s)$ of the string s, and outputs 0 or 1. *Encrypt* takes as input a public key pk and a plaintext m,

and outputs a ciphertext c. *Decrypt* takes as input the public key pk, a valid string-signature $(s, \sigma(s))$ and a ciphertext c, and outputs the plaintext m.

An ASBB scheme has a key-homomorphic property. This property states that, for any two public/secret key pairs (pk_1, sk_1) and (pk_2, sk_2) generated by running $KeyGen(\pi)$, two signatures $\sigma_1 = Sign(pk_1, sk_1, s)$, $\sigma_2 = Sign(pk_2, sk_2, s)$ on any message string s with respect to the two public keys, it holds that $Verify(pk_1 \otimes pk_2, s, \sigma_1 \odot \sigma_2) = 1$, where $\otimes : \Gamma \times \Gamma \to \Gamma$ and $\odot : \Omega \times \Omega \to \Omega$ are two efficient operations in the public key space Γ and the signature space Ω, respectively. Clearly, from the key-homomorphic property, we have that $Decrypt(pk_1 \otimes pk_2, s, \sigma_1 \odot \sigma_2, c) = m$ for any plaintext m and the corresponding ciphertext $c = Encrypt(pk_1 \otimes pk_2, m)$.

Furthermore, an ASBB scheme has an interesting property referred to as aggregatability. Assume that an adversary $\mathcal{A}$ knows $(\pi, pk_1, \cdots, pk_n)$, where π is the system parameters, and $pk_1, \cdots, pk_n$ are n different public keys generated by independently invoking *KeyGen* of the ASBB scheme. For n public binary strings $s_1, \cdots, s_n \in \{0,1\}^*$, the adversary $\mathcal{A}$ is provided with valid signatures $\sigma_i(s_j)$ under pk_i for $1 \leq i, j \leq n$ and $i \neq j$. Due to the key-homomorphic property, $pk = pk_1 \otimes \cdots \otimes pk_n$ forms the public key of the aggregated ASBB instance. Aggregatability states that the new ASBB instance related to the aggregated public key pk is secure against any polynomial-time adversary $\mathcal{A}$. Wu *et al.*'s ASBB scheme [32] is briefly reviewed next.

- **ParaGen**(π). Let `PairGen` be an algorithm that, on input a security parameter 1^λ, outputs a tuple $\Upsilon = (p, \mathbb{G}, \mathbb{G}_T, e)$, where $\mathbb{G}$ and $\mathbb{G}_T$ have the same prime order p, and $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is an efficient non-degenerate bilinear map such that $e(g, g) \neq 1$ for any generator g of $\mathbb{G}$, and for all $u, v \in \mathbb{Z}$, it holds that $e(g^u, g^v) = e(g, g)^{uv}$. Let $\Upsilon = (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \texttt{PairGen}(1^\lambda)$, and g be a generator of $\mathbb{G}$, and $H : \{0,1\}^* \to \mathbb{G}$ be a cryptographic hash function. The system parameters are $\pi = (\Upsilon, g, H)$.
- **KeyGen**(π). Select at random $r \in \mathbb{Z}_p^*$, $X \in \mathbb{G} \setminus \{1\}$. Compute $R = g^{-r}$, $A = e(X, g)$. Output a public key $pk = (R, A)$ and its associating secret key $sk = (r, X)$.
- **Sign**(pk, sk, s). Take as inputs public key $pk = (R, A)$, secret key $sk = (r, X)$ and a string $s \in \{0,1\}^*$, and output a signature $\sigma = XH(s)^r$ on s.
- **Verify**(pk, s, σ). Take as inputs public key $pk = (R, A)$, a message-signature pair (s, σ), and output 1 if $e(\sigma, g)e(H(s), R) = A$ holds; else output 0.
- **Encryption**(pk, ξ). Given public key $pk = (R, A)$, for a plaintext $\xi \in \mathbb{G}_T$, randomly select $t \in \mathbb{Z}_p^*$ and compute $c_1 = g^t, c_2 = R^t, c_3 = \xi A^t$. Output $c = (c_1, c_2, c_3)$.
- **Decryption**(pk, s, σ, c). Given public key $pk = (R, A)$ and ciphertext $c = (c_1, c_2, c_3)$, anyone with a valid message-signature pair (s, σ) can extract $\xi = \frac{c_3}{e(\sigma, c_1)e(H(s), c_2)}$.

In the ASBB scheme, every signature under the public key can be used as a decryption key to decrypt ciphertexts generated with the same public key. This feature allows ASBB to be used as static broadcast schemes.

3.2 An Aggregatable BE Scheme Based on ASBB

We construct a BE scheme from the the ASBB scheme [32] and show the resulting BE scheme preserves aggregatability as that of the underlying ASBB scheme. The construction is conceptually simple. Assume that the j-th user holds decryption keys[2] corresponding to the indices $\{0, ..., n\} \setminus \{j\}$. An encrypter knows which public key he should use. For instance, if the encrypter doesn't want to revoke anybody, he encrypts using pk_0. If he wants to exclude i from decrypting, he encrypts using pk_i. If he wants to exclude i and j from decrypting, he encrypts by using an *aggregated* public key $pk_i \otimes pk_j$. In the same way, more users can be excluded from decrypting. With the parameters in the above setting, the proposal is realized as follows.

- $\mathsf{BSetup}(n, N)$: The dealer randomly chooses $X_i \in \mathbb{G}, r_i \in \mathbb{Z}_p^*$ and computes $R_i = g^{-r_i}, A_i = e(X_i, g)$. The BE public key is $PK = ((R_0, A_0), \cdots, (R_n, A_n))$ and the BE secret key is $sk = ((r_0, X_0), \cdots, (r_n, X_n))$.
- $\mathsf{BKeyGen}(j, SK)$: For $j = 1, \cdots, n$, the private key of the user j is $d_j = (\sigma_{0,j}, \cdots, \sigma_{j-1,j}, \sigma_{j+1,j}, \cdots, \sigma_{n,j}) : \sigma_{i,j} = X_i H(ID_j)^{r_i}$.
- $\mathsf{BEncryption}(\mathbb{R}, PK)$: Set $\overline{\mathbb{R}} = \{0, 1, \cdots, n\} \setminus \mathbb{R}$. Randomly pick t in $\mathbb{Z}_p$ and compute $c = (c_1, c_2) : c_1 = g^t, c_2 = (\prod_{i \in \overline{\mathbb{R}}} R_i)^t$. Set the session key $\xi = (\prod_{i \in \overline{\mathbb{R}}} A_i)^t$. Output (c, ξ) and send $(\mathbb{R}, c)$ to receivers.
- $\mathsf{BDecryption}(\mathbb{R}, j, d_j, c, PK)$: If $j \in \mathbb{R}$, the receiver j extracts ξ from c with private key d_j by computing $e(\prod_{i \in \overline{\mathbb{R}}} \sigma_{i,j}, c_1) e(H(ID_j), c_2) = \xi$.

The correctness of the BE scheme above follows from direct verification of the following equations
$e(\prod_{i \in \overline{\mathbb{R}}} \sigma_{i,j}, c_1) e(H(ID_j), c_2) = e(\prod_{i \in \overline{\mathbb{R}}} X_i H(ID_j)^{r_i}, g^t) e(H(ID_j), \prod_{i \in \overline{\mathbb{R}}} g^{-r_i t})$
$= e(\prod_{i \in \overline{\mathbb{R}}} X_i, g)^t = (\prod_{i \in \overline{\mathbb{R}}} A_i)^t = \xi$.

The security of our BE scheme relies on the decision n-BDHE assumption which was shown to be sound by Boneh *et al.* [2] in the generic group model.

Definition 3 (Decision n-BDHE Assumption). *Let $\mathbb{G}$ be a bilinear group of prime order p as defined above, g a generator of $\mathbb{G}$, and $h = g^t$ for some unknown $t \in \mathbb{Z}_p$. Denote $\overrightarrow{y}_{g,\alpha,n} = (g_1, \cdots, g_n, g_{n+2}, \cdots, g_{2n}) \in \mathbb{G}^{2n-1}$, where $g_i = g^{\alpha^i}$ for some unknown $\alpha \in \mathbb{Z}_p$. We say that an algorithm $\mathcal{B}$ that outputs $b \in \{0, 1\}$ has advantage ε in solving the decision n-BDHE assumption if $|\Pr[\mathcal{B}(g, h, \overrightarrow{y}_{g,\alpha,n}, e(g_{n+1}, h)) = 0] - \Pr[\mathcal{B}(g, h, \overrightarrow{y}_{g,\alpha,n}, Z) = 0)]| \geq \varepsilon$, where the probability is over the random choice of g in $\mathbb{G}$, the random choice $t, \alpha \in \mathbb{Z}_p$, the random choice of $Z \in \mathbb{G}_T$, and the random bits consumed by $\mathcal{B}$. We say that the decision (τ, ε, n)-BDHE assumption holds in $\mathbb{G}$ if no τ-time algorithm has advantage at least ε in solving the decision n-BDHE assumption.*

According to the BE security definition in [17], our scheme is fully collusion-resistant under the Decision BDHE assumption. The proof is given in the full

[2] Here, user j's i-th decryption key corresponding to index $i \in \{0, ..., n\} \setminus \{j\}$ is a signature $\sigma_{i,j} = \sigma_i(ID_j)$ on user j's identity ID_j verifiable under the public key pk_i.

version of the paper [33]. One can further apply the generic Gentry-Waters transformation [17] to convert our semi-adaptive BE schemes into an adaptively secure one. The cost is to double the size of the public keys and the ciphertexts.

Theorem 1. *The proposed BE scheme for dynamic groups has full collusion resistance against semi-adaptive attacks in the random oracle model if the decision n-BDHE assumption holds. More formally, if there exists a semi-adaptive attacker $\mathcal{A}$ breaking our scheme with advantage ϵ in time τ, then there exists an algorithm $\mathcal{B}$ breaking the n-BDHE assumption with advantage ϵ in time $\tau' = \tau + \mathcal{O}((q_H + n^2)\tau_{Exp})$, where q_H is the number of queries to the random oracle from $\mathcal{A}$, and τ_{Exp} is the time to compute an exponentiation in $\mathbb{G}$ or $\mathbb{G}_T$.*

One may observe that, in the above BE scheme, if we replace $H(ID_j)$ with a random element h_j in $\mathbb{G}$, we obtain a semi-adaptive BE scheme with short ciphertexts in the standard model. In this case, to simulate h_j in the security proof, we just need to set $h_j = g^{\alpha^j} g^{v_j}$ for a randomly chosen value $v_j \in Z_p$, where g^{α^j} is obtained from the decision n-BDHE assumption.

3.3 Useful Properties

Our BE scheme inherits the key-homomorphic property of the underlying ASBB scheme. Consider the system parameters defined above. Let $PK_1 = (R_{0,1}, A_{0,1}), \cdots, (R_{n,1}, A_{n,1}))$ and $PK_2 = ((R_{0,2}, A_{0,2}), \cdots, (R_{n,2}, A_{n,2}))$ be the respective public keys of two random instances of the above BE scheme, and for $j = 1, \cdots, n$, let $d_{j,1} = (\sigma_{0,j,1}, \cdots, \sigma_{j-1,j,1}, \sigma_{j+1,j,1}, \cdots, \sigma_{n,i,1}) \in \mathbb{G}^n$ and $d_{j,2} = (\sigma_{0,j,2}, \cdots, \sigma_{j-1,j,2}, \sigma_{j+1,j,2}, \cdots, \sigma_{n,j,2}) \in \mathbb{G}^n$ be the respective decryption keys corresponding to index j under PK_1 and PK_2. Define $PK = PK_1 \circledast PK_2 = ((R_{0,1}R_{0,2}, A_{0,1}A_{0,2}), \cdots, (R_{n,1}R_{n,2}, A_{n,1}A_{n,2}))$ and define $dk_j = d_{j,1} \boxdot d_{j,2} = (\sigma_{0,j,1}\sigma_{0,j,2}, \cdots, \sigma_{j-1,j,1}\sigma_{j-1,j,2}, \sigma_{j+1,j,1}\sigma_{j+1,j,2}, \cdots, \sigma_{n,j,1}\sigma_{n,j,2})$. Then PK is the public key of a new instance of the above BE scheme and dk_j is the new decryption key corresponding to the index j. This fact can be directly verified.

Our BE scheme also preserves the aggregatability of the underlying ASBB scheme. Roughly speaking, a BE scheme is aggregatable if n instances of the BE scheme can be aggregated into a new BE instance secure against an attacker accessing some decryption keys of each instance, provided that the i-th decryption key corresponding to the i-th instance is unknown to the attacker for $i = 1, \cdots, n$. More formally, this property can be defined as follows.

Definition 4 (Aggregatability). *Consider the following game between an adversary $\mathcal{A}$ and a challenger $\mathcal{CH}$:*

- **Setup:** *$\mathcal{A}$ initializes the game with an integer n. $\mathcal{CH}$ replies with $(\pi, PK_1, \cdots, PK_n)$ which are the system parameters and the n independent public keys of the BE scheme.*
- **Corruption:** *For $1 \leq i, j \leq n$, where $i \neq j$, the adversary $\mathcal{A}$ is allowed to know the decryption keys $dk_{j,i}$ corresponding to index j with respect to the public key PK_i.*

– **Challenge:** *$\mathcal{CH}$ and $\mathcal{A}$ run a standard Ind-CPA game under the aggregated public key $PK = PK_1 \circledast \cdots \circledast PK_n$. $\mathcal{A}$ wins if $\mathcal{A}$ outputs a correct guess bit. Denote $\mathcal{A}$'s advantage by $Adv_{\mathcal{A}} = |\Pr[win] - \frac{1}{2}|$.*

A BE scheme is said to be (τ, ε, n)-aggregatable if no τ-time algorithm $\mathcal{A}$ has advantage $Adv_{\mathcal{A}} \geq \varepsilon$ in the above aggregatability game.

Theorem 2. *If there exists an attacker $\mathcal{A}$ who wins the aggregatability game with advantage ϵ in time τ, then there exists an algorithm $\mathcal{B}$ breaking the n-BDHE assumption with advantage ϵ in time $\tau' = \tau + \mathcal{O}((n^3)\tau_{Exp})$.*

For the proof of the previous theorem, we refer to Theorem 3 where we prove a stronger property in the sense that the attacker is additionally allowed to know the internal randomness used to compute $dk_{j,i}$ corresponding some PK_i for $1 \leq i, j \leq n$ where $i \neq j$.

4 Proposed CBE Scheme

In this section, we propose a CBE based on the above aggregatable BE scheme. The basic construction has short ciphertexts and long protocol transcripts. Then we show an efficient trade-off between ciphertexts and protocol transcripts.

4.1 High-Level Description

Our basic idea is to introduce the revocation mechanism of a regular BE scheme into the asymmetric GKA scheme [32]. To this end, each member acts as the dealer of the aggregatable BE scheme above. The k-th user publishes PK_k and $d_{j,k}$, where $d_{j,k}$ is the decryption key of PK_k corresponding to the index $j \in \{1, \cdots, n\} \setminus \{k\}$. Then the negotiated public key is $PK = PK_0 \circledast \cdots \circledast PK_n$. Each member j can compute the decryption key $dk_j = dk_{j,j} \boxdot_{k=1,k\neq j}^{n} dk_{j,k}$. Observe that $dk_{j,j}$ has never been published. Due to the key homomorphism of the BE scheme above, dk_j is a valid decryption key corresponding to PK. Hence, anyone knowing PK can encrypt to any subset of the members and the intended receivers can decrypt.

To guarantee the security of the resulting CBE scheme, we also need to show that *only* the intended receivers can decrypt. This is ensured by the fact that the underlying BE scheme is aggregatable. Indeed, although the Gentry-Waters BE scheme [17] is key-homomorphic, an analog of our CBE scheme using the Gentry-Waters BE scheme as a building block is shown to be insecure in [33], because the Gentry-Waters BE scheme is not aggregatable. We note that a *static* PKBE scheme without a dealer can be trivially obtained from the ASGKA protocol in [32]. This is realized by letting each member to register his/her published string as her public key. Then anyone knowing the public keys of all members can send encrypted messages to the group and only the group members can decrypt the message. However, no revocation mechanism is provided. To exclude some members, one may be motivated to modify the above trivial construction by using the aggregation of the public keys of the intended receivers as the

sub-group public key. Clearly, this will allow the intended receivers to decrypt ciphertexts generated with this sub-group public key. Unfortunately, anyone (not necessary to be a revoked member) knowing the receivers' public keys can also decrypt, as shown in [33].

4.2 The Proposal

Based on our aggregatable BE scheme, we implement a CBE scheme with short ciphertexts. Assume that the group size is at most n. Let $\Upsilon = (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \mathtt{PairGen}(1^\lambda)$, and $g, h_1, \cdots, h_n$ be independent generators of $\mathbb{G}$. The system parameters are $\pi = (\lambda, n, \Upsilon, g, h_1, \cdots, h_n)$.

- **Setup.** The set-up of a CBE system consists of the following three procedures:
  - Group Key Agreement Execution: For $1 \leq k \leq n$, member k does the following:
    Randomly choose $X_{i,k} \in \mathbb{G}, r_{i,k} \in \mathbb{Z}_p^*$;
    Compute $R_{i,k} = g^{-r_{i,k}}, A_{i,k} = e(X_{i,k}, g)$;
    Set $PK_k = ((R_{0,k}, A_{0,k}), \cdots, (R_{n,k}, A_{n,k}))$;
    For $1 \leq j \leq n$, $j \neq k$, compute $\sigma_{i,j,k} = X_{i,k} h_j^{r_{i,k}}$ for $0 \leq i \leq n, i \neq j$;
    Set $d_{j,k} = (\sigma_{0,j,k}, \cdots, \sigma_{j-1,j,k}, \sigma_{j+1,j,k}, \cdots, \sigma_{n,j,k})$;
    Publish $(PK_k, d_{1,k}, \cdots, d_{k-1,k}, d_{k+1,k}, \cdots, d_{n,k})$ and keep $d_{k,k}$ secret.
  - Group Encryption Key Derivation: The group encryption key is $PK = PK_0 \circledast \cdots \circledast PK_n = ((R_0, A_0), \cdots, (R_n, A_n))$, where $R_i = \prod_{k=1}^n R_{i,k}$, $A_i = \prod_{k=1}^n A_{i,k}$ for $i = 0, \cdots, n$. The group encryption key PK is publicly computable.
  - Member Decryption Key Derivation: For $0 \leq i \leq n$, $1 \leq j \leq n$ and $i \neq j$, member j can compute decryption key $d_j = (\sigma_{0,j}, \cdots, \sigma_{j-1,j}, \sigma_{j+1,j}, \cdots, \sigma_{n,j})$, where $\sigma_{i,j} = \sigma_{i,j,j} \prod_{k=1,k\neq j}^n \sigma_{i,j,k} = \prod_{k=1}^n \sigma_{i,j,k} = \prod_{k=1}^n X_{i,k} h_j^{r_{i,k}}$.
- **CBEncrypt.** Assume that a sender (not necessarily a group member) wants to send to receivers in $\mathbb{R} \subseteq \{1, \cdots, n\}$ a session key ξ. Set $\overline{\mathbb{R}} = \{0, 1, \cdots, n\} \setminus \mathbb{R}$. Randomly pick t in $\mathbb{Z}_p$ and compute the ciphertext $c = (c_1, c_2)$ where $c_1 = g^t, c_2 = (\prod_{i \in \overline{\mathbb{R}}} R_i)^t$. Output (c, ξ) where $\xi = (\prod_{i \in \overline{\mathbb{R}}} A_i)^t$. Send $(\mathbb{R}, c)$ to the receivers.
- **CBDecrypt.** If $j \in \mathbb{R}$, receiver j can extract ξ from the ciphertext c with decryption key d_j by computing $e(\prod_{i \in \overline{\mathbb{R}}} \sigma_{i,j}, c_1) e(h_j, c_2) = \xi$.

The correctness of the proposed CBE scheme is correct directly follows from the fact that the underlying BE scheme is correct and key-homomorphic. As to security, we have the following theorem, whose proof is given in [33].

Theorem 3. *The proposed CBE scheme has fully collusion-resistant secrecy against semi-adaptive attacks in the standard model if the decision n-BDHE assumption holds. More formally, if there exists a semi-adaptive attacker $\mathcal{A}$ breaking our scheme with advantage ϵ in time τ, then there exists an algorithm $\mathcal{B}$ breaking the n-BDHE assumption with advantage ϵ in time $\tau' = \tau + O((n^3)\tau_{Exp})$.*

4.3 Discussion

We first examine the online complexity our scheme which is critical for the practicality of a CBE scheme. We use the widely-adopted metrics [3, 4, 6, 26, 17] for regular BE schemes. After the CBSetup procedure, a sender needs to retrieve and store the group public key PK consisting of n elements in $\mathbb{G}$ and n elements in $\mathbb{G}_T$. This requires about $150n$ bytes to achieve the security level of an RSA-1024 cryptosystem. Note that in the motivated applications, the group size is usually not very large. Consider an initial group of 100 users. The group public key is about $15K$ bytes long and acceptable in practice. Moreover, for encryption, the sender needs only two exponentiations and the ciphertext merely contains two elements in $\mathbb{G}$. This is about n times more efficient than the trivial solution. At the receiver's side, in addition to the description of the bilinear pair which may be shared by many other security applications, a receiver needs to store n elements in $\mathbb{G}$ for decryption. The storage cost of a receiver is about $22n$ bytes. For decryption, a receiver needs to compute two single-base bilinear pairings (or one two-base bilinear pairing). The online costs on the sides of both the sender and the receivers are really low.

We next discuss the complexity of the CBSetup procedure to set up a CBE system. The overhead incurred by this procedure is $O(n^2)$. However, in most cases, this procedure needs to be run only once and this can be done offline before online transmission of secret session keys. For instance, in the social networks example, a number of friends exchange their CBSetup transcripts and establish a CBE system to secure their subsequent sharing of private picture/videos. Since CBE allows revoking members, the members do not need to reassemble for a new run of the CBSetup procedure until some new friends join. From our personal experience, the group lifetime usually lasts from weeks to months. These observations imply that our protocol is practical in the real world.

Furthermore, if the initial group is too large, an efficient trade-off can be employed [3] to balance the online and offline costs. Suppose that n is a cube, *i.e.*, $n = n_1^3$, and the initial group has n members. We divide the full group into n_1^2 subgroups, each of which has n_1 members. By applying our basic CBE to each subgroup, we obtain a CBE scheme with $O(n_1^2)$-size transcripts per member during the offline stage of group key establishment; a sender needs to do $O(n_1^2)$ encryption operations of the basic CBE scheme, which produces $O(n_1^2)$-size ciphertexts. Consequently, we obtain a CBE scheme with $O(n^{\frac{2}{3}})$ complexity. This is comparable to up-to-date public-key BE systems whose complexity is $O(n^{\frac{1}{2}})$. For a group of 1000 users, our dealer-free BE scheme is about 10 times more efficient than the trivial solution. It is about 3 times less efficient than a public-key BE scheme, but our CBE does not require a trusted key dealer. The cost of versatileness is acceptable.

One may notice a subtlety in the above trade-off. When the basic CBE scheme is applied to each subgroup, members in each subgroup will extract the same session key, but members in different subgroups will have different session keys. This is inconsistent with the CBE definition in which all members should extract the same session key, even if the members are in different subgroups. This can

be trivially addressed as follows. The sender additionally selects a string from the session key space and encrypts it for each subgroup with the session keys shared by each subgroup. Then all members can extract the same resulting session key. This introduces an additional $O(n^{\frac{2}{3}})$-size ciphertext if there are $O(n^{\frac{2}{3}})$ subgroups, but it does not affect the asymptotical complexity of the scheme after a trade-off.

Finally, we assume that the communication channels between members are authenticated during the CBSetup stage to establish the group encryption key. In practice, these authenticated channels can be the pre-existing ones between members (*e.g.*, in instant-messaging system and cooperative scientific computation) or be established by personal interaction (*e.g.*, some *ad hoc* network applications). This is plausible since CBE is usually deployed for cooperative members who may be friends. Note that the CBSetup sub-protocol requires only one round. An alternative option to achieve authentication is to let a partially trusted third party certify each member's protocol transcript. The third party plays a role similar to a certification authority in the popular PKI setting, and cannot read the plaintexts encrypted to the members. This is different from regular BE systems where the fully trusted dealer can decrypt all communications to the members. For instance, in a social network application, the service provider can serve as the partially trusted third party. This is also plausible since this kind of applications usually require users to register for service. In this case, the CBSetup transcript of each member can be viewed as her public key.

5 Conclusions

In this paper, we formalized the CBE primitive, which bridges the GKA and BE notions. In CBE, anyone can send secret messages to any subset of the group members, and the system does not require a trusted key server. Neither the change of the sender nor the dynamic choice of the intended receivers require extra rounds to negotiate group encryption/decryption keys. Following the CBE model, we instantiated an efficient CBE scheme that is secure in the standard model. As a versatile cryptographic primitive, our novel CBE notion opens a new avenue to establish secure broadcast channels and can be expected to secure numerous emerging distributed computation applications.

Acknowledgments. The authors gratefully acknowledge the anonymous reviewers for their invaluable comments. The authors are partly supported by the EU 7FP through project "DwB", the Spanish Government through projects CTV-09-634, PTA2009-2738-E, TSI-020302-2010-153, PT-430000- 2010-31, TIN2009-11689, CONSOLIDER INGENIO 2010 "ARES" CSD2007-0004 and TSI2007-65406-C03-01, by the Government of Catalonia under grant SGR2009-1135, and by the NSF of China through projects 60970114, 60970115, 60970116, 61173154, 61003214, 61173192, 91018008, 61021004 and 11061130539. The authors also acknowledge support by the Fundamental Research Funds for the Central Universities of China to Project 3103004, and Shaanxi Provincial Education Department through Scientific Research Program 2010JK727. The fourth

author is partially supported as an ICREA-Acadèmia researcher by the Catalan Government. The fifth author is partially supported by ISF grant 938/09. The authors are with the UNESCO Chair in Data Privacy, but this paper does not necessarily reflect the position of UNESCO nor does it commit that organization.

References

1. Abdalla, M., Chevalier, C., Manulis, M., Pointcheval, D.: Flexible Group Key Exchange with On-demand Computation of Subgroup Keys. In: Bernstein, D.J., Lange, T. (eds.) AFRICACRYPT 2010. LNCS, vol. 6055, pp. 351–368. Springer, Heidelberg (2010)
2. Boneh, D., Boyen, X., Goh, E.J.: Hierarchical Identity Based Encryption with Constant Size Ciphertext. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440–456. Springer, Heidelberg (2005)
3. Boneh, D., Gentry, C., Waters, B.: Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 258–275. Springer, Heidelberg (2005)
4. Boneh, D., Sahai, A., Waters, B.: Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 573–592. Springer, Heidelberg (2006)
5. Boneh, D., Silverberg, A.: Applications of Multilinear Forms to Crytography. Contemporary Mathematics, vol. 324, pp. 71–90 (2003)
6. Boneh, D., Waters, B.: A Fully Collusion Resistant Broadcast, Trace, and Revoke System. In: ACM CCS 2006, pp. 211–220. ACM Press (2006)
7. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003)
8. Boyd, C., González-Nieto, J.M.: Round-Optimal Contributory Conference Key Agreement. In: Desmedt, Y.G. (ed.) PKC 2003. LNCS, vol. 2567, pp. 161–174. Springer, Heidelberg (2002)
9. Bresson, E., Chevassut, O., Pointcheval, D.: Provably Authenticated Group Diffie-Hellman Key Exchange - The Dynamic Case. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 290–309. Springer, Heidelberg (2001)
10. Bresson, E., Chevassut, O., Pointcheval, D.: Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 321–336. Springer, Heidelberg (2002)
11. Bresson, E., Chevassut, O., Pointcheval, D., Quisquater, J.-J.: Provably Authenticated Group Diffie-Hellman Key Exchange. In: ACM CCS 2001, pp. 255–264. ACM Press (2001)
12. Burmester, M., Desmedt, Y.: A Secure and Efficient Conference Key Distribution System. In: De Santis, A. (ed.) EUROCRYPT 1994. LNCS, vol. 950, pp. 275–286. Springer, Heidelberg (1995)
13. Cheon, J.H., Jho, N.S., Kim, M.H., Yoo, E.S.: Skipping, Cascade, and Combined Chain Schemes for Broadcast Encryption. IEEE Transactions Information Theory 54(11), 5155–5171 (2008)
14. Diffie, W., Hellman, M.: New Directions in Cryptography. IEEE Transactions on Information Theory 22(6), 644–654 (1976)
15. Fiat, A., Naor, M.: Broadcast Encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)

16. Fujisaki, E., Okamoto, T.: Secure Integration of Asymmetric and Symmetric Encryption Schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 537–554. Springer, Heidelberg (1999)
17. Gentry, C., Waters, B.: Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 171–188. Springer, Heidelberg (2009)
18. Gorantla, M.C., Boyd, C., González Nieto, J.M., Manulis, M.: Generic One Round Group Key Exchange in the Standard Model. In: Lee, D., Hong, S. (eds.) ICISC 2009. LNCS, vol. 5984, pp. 1–15. Springer, Heidelberg (2010)
19. Halevy, D., Shamir, A.: The LSD Broadcast Encryption Scheme. In: Yung, M. (ed.) CRYPTO 2002. LNCS, vol. 2442, pp. 47–60. Springer, Heidelberg (2002)
20. Ingemarsson, I., Tang, D.T., Wong, C.K.: A Conference Key Distribution System. IEEE Transactions on Information Theory 28(5), 714–720 (1982)
21. Joux, A.: A One Round Protocol for Tripartite Diffie-Hellman. J. of Cryptology 17, 263–276 (2004)
22. Kim, H.J., Lee, S.M., Lee, D.H.: Constant-Round Authenticated Group Key Exchange for Dynamic Groups. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 245–259. Springer, Heidelberg (2004)
23. Kim, Y., Perrig, A., Tsudik, G.: Tree-Based Group Key Agreement. ACM Transactions on Information System Security 7(1), 60–96 (2004)
24. Mao, Y., Sun, Y., Wu, M., Liu, K.J.R.: JET: Dynamic Join-Exit-Tree Amortization and Scheduling for Contributory Key Management. IEEE/ACM Transactions on Networking 14(5), 1128–1140 (2006)
25. Naor, M., Pinkas, B.: Efficient Trace and Revoke Schemes. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 1–20. Springer, Heidelberg (2001)
26. Park, J.H., Kim, H.J., Sung, M.H., Lee, D.H.: Public Key Broadcast Encryption Schemes With Shorter Transmissions. IEEE Transactions on Broadcasting 54(3), 401–411 (2008)
27. Sherman, A., McGrew, D.: Key Establishment in Large Dynamic Groups Using One-way Function Trees. IEEE Transactions on Software Engineering 29(5), 444–458 (2003)
28. Snoeyink, J., Suri, S., Varghese, G.: A Lower Bound for Multicast Key Distribution. In: INFOCOM 2001, pp. 422–431. IEEE Press (2001)
29. Steiner, M., Tsudik, G., Waidner, M.: Key Agreement in Dynamic Peer Groups. IEEE Transactions on Parallel and Distributed Systems 11(8), 769–780 (2000)
30. Tzeng, W.-G., Tzeng, Z.-J.: Round-Efficient Conference Key Agreement Protocols with Provable Security. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 614–627. Springer, Heidelberg (2000)
31. Wong, C.K., Gouda, M., Lam, S.: Secure Group Communications Using Key Graphs. IEEE/ACM Transactions on Networking 8(1), 16–30 (2000)
32. Wu, Q., Mu, Y., Susilo, W., Qin, B., Domingo-Ferrer, J.: Asymmetric Group Key Agreement. In: Joux, A. (ed.) EUROCRYPT 2009. LNCS, vol. 5479, pp. 153–170. Springer, Heidelberg (2009)
33. Wu, Q., Qin, B., Zhang, L., Domingo-Ferrer, J., Farras, O.: Bridging Broadcast Encryption and Group Key Agreement (full version), `http://eprint.iacr.org`