Contributory Broadcast Encryption with Efficient Encryption and Short Ciphertexts

Qianhong Wu, *Member, IEEE,* Bo Qin, Lei Zhang, *Member, IEEE,* Josep Domingo-Ferrer, *Fellow, IEEE* Oriol Farràs, and Jesús A. Manjón

***Abstract*—Traditional broadcast encryption (BE) schemes allow a sender to securely broadcast to any subset of members but require 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 group 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 (ConBE). 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 ConBE 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. Of independent interest, we present a new BE scheme that is aggregatable. The aggregatability property is shown to be useful to construct advanced protocols.**

***Index Terms*—Broadcast encryption, group key agreement, contributory broadcast encryption, provable security.**

I. Introduction

With the fast advance and pervasive deployment of communication technologies, there is an increasing demand of versatile cryptographic primitives to protect group communications and computation platforms. These new platforms include instant-messaging tools, collaborative computing, mobile *ad hoc* networks and social networks. 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) [1] 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, a BE system heavily relies on a fully trusted key server who generates secret decryption keys for the members and can read all the communications to any members.

Group key agreement (GKA) is another well-understood cryptographic primitive to secure group-oriented communications. A conventional GKA [2] allows a group of members to establish a common secret key via open networks. However, whenever a sender wants to send a message to a group, he must first join the group and run a GKA protocol to share a secret key with the intended members. More recently, and to overcome this limitation, Wu *et al.* introduced asymmetric GKA [3], 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 the newly introduced asymmetric GKA allow *the sender* to *unilaterally* exclude any particular member from reading the plaintext[1]. Hence, it is essential to find more flexible cryptographic primitives allowing dynamic broadcasts without a fully trusted dealer.

A. Our Contributions

We present the Contributory Broadcast Encryption (ConBE) primitive, which is a hybrid of GKA and BE. Compared to its preliminary Asiacrypt 2011 version [5], this full paper provides complete security proofs, illustrates the necessity of the aggregatability of the underlying BE building block and shows the practicality of our ConBE scheme with experiments. Specifically, our main contributions are as follows.

First, we model the ConBE primitive and formalize its security definitions. ConBE incorporates the underlying ideas of GKA and BE. A group of members interact via open networks to negotiate a public encryption key while each member holds a different secret decryption key. Using the public encryption key, anyone can encrypt any message to any subset of the group members and only the intended receivers can decrypt. Unlike GKA, ConBE allows the sender to exclude some members from reading the ciphertexts. Compared to BE, ConBE does not need a fully trusted third party to set up the system. We formalize collusion resistance by defining an attacker who can fully control all the members outside the intended receivers but cannot extract useful information from the ciphertext.

Q. Wu is with the School of Electronics and Information Engineering, Beihang University, and The State Key Laboratory of Integrated Services Networks, Xidian University and State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: qianhong.wu@buaa.edu.cn).

B. Qin is with Key Laboratory of Data Engineering and Knowledge Engineering (Renmin University of China) Ministry of Education, School of Information, Renmin University of China, ZhongGuanCun Street No. 59, Haidian District, Beijing, China, Beijing, China (e-mail: bo.qin@ruc.edu.cn).

L. Zhang is with Shanghai Key Laboratory of Trustworthy Computing, Software Engineering Institute, East China Normal University, Shanghai, China (e-mail: leizhang@sei.ecnu.edu.cn).

J. Domingo-Ferrer, O. Farràs and J. A. Manjón are with Universitat Rovira i Virgili, Department of Computer Engineering and Mathematics, UNESCO Chair in Data Privacy, Tarragona, Catalonia (e-mail: {josep.domingo, oriol.farras, jesus.manjon}@urv.cat).

[1]Dynamic symmetric GKA equipped with a leave sub-protocol allows the members to exclude some members from decrypting ciphertexts. In this case, if the sender (who is also a group member) wants to exclude some other members, he/she has to seek the agreement of the remaining members to run the leave sub-protocol. The sender cannot exclude any member unilaterally.

Second, we present the notion of aggregatable broadcast encryption (AggBE). Coarsely speaking, a BE scheme is aggregatable if its secure instances can be aggregated into a new secure instance of the BE scheme. Specifically, only the aggregated decryption keys of the same user are valid decryption keys corresponding to the aggregated public keys of the underlying BE instances. We observe that the aggregatability of AggBE schemes is necessary in the construction of our ConBE scheme and the BE schemes in the literature are not aggregatable. We construct a concrete AggBE scheme tightly proven to be fully collusion-resistant under the decision BDHE assumption. The proposed AggBE scheme offers efficient encryption/decryption and short ciphertexts.

Finally, we construct an efficient ConBE scheme with our AggBE scheme as a building block. The ConBE 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 ConBE 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 ConBE scheme) is very low. We also illustrate a trade-off between the set-up complexity and the online performance. After a 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. We conduct a series of experiments and the experimental results validate the practicality of our scheme.

B. Potential Applications

A potential application of our ConBE is to secure data exchanged among friends via social networks. Since the Prism scandal [4], people are increasingly concerned about the protection of their personal data shared with their friends over social networks. Our ConBE can provide a feasible solution to this problem. Indeed, Phan *et al.* [6] underlined the applications of our ConBE [5] to social networks. In this scenario, if a group of users want to share their data without letting the social network operator know it, they can use our ConBE scheme. Since the setup procedure of our ConBE only requires one round of communication, each member of the group just needs to broadcast one message to other intended members in a send-and-leave way, without the synchronization requirement. After receiving the messages from the other members, all the members share the encryption key that allows any user to selectively share his/her data to any subgroup of the members. Furthermore, it also allows sensitive data to be shared among different groups. Other applications may include instant messaging among family members, secure scientific research tasks jointly conducted by scientists from different places, and disaster rescue using a mobile *ad hoc* network. A common feature of these scenarios is that a group of users would like to exchange sensitive data but a fully trusted third party is unavailable. Our ConBE provides an efficient solution to these applications.

C. Related Work

A number of works have addressed key agreement protocols for multiple parties. The schemes due to Ingemarsson *et al.* [2] and Steiner *et al.* [7] are designed for n parties and require $O(n)$ rounds. Tree key structures have been further proposed, reducing the number of rounds to $O(\log n)$ [8], [9], [10]. Multi-round GKA protocols pose a synchronism requirement: in order to complete the protocol, all the group members have to stay online simultaneously. How to optimize the round complexity of GKA protocols has been studied in several works (*e.g.*, [11], [12], [13]). In [14], Tzeng presented a constant-round GKA protocol that can identify cheaters. Subsequently, Yi [15] constructed a fault-tolerant protocol in an identity-based setting. Burmester and Desmedt [16] proposed a two-round n-party GKA protocol for n parties. The Joux protocol [17] is one-round and only applicable to three parties. The work of Boneh and Silverberg [18] shows a one-round $(n+1)$-party GKA protocol with n-linear pairings.

Dynamic GKA protocols provide extra mechanisms to handle member changes. Bresson *et al.* [19], [20] extended the protocol in [21] to dynamic GKA protocols that allow members to leave and join the group. The number of rounds in the set-up/join algorithms of the Bresson *et al.*'s protocols [19], [20] is linear with the group size, but the number of rounds in the leave algorithm is constant. The theoretical analysis in [22] shows that for any tree-based group key agreement scheme, the lower bound of the worst-case cost is $O(\log n)$ rounds of interaction for a member to join or leave. Without relying on a tree-based structure, Kim *et al.* [23] proposed a two-round dynamic GKA protocol. Recently, Abdalla *et al.* [24] 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. Jarecki *et al.* [25] presented a robust two-round GKA protocol in which a session key can be established even if some participants fail during the execution of the protocol. Observing that existing GKA protocols cannot handle sender/member changes efficiently, Wu *et al.* presented a group key management protocol [26] in which a change of the sender or monotone exclusion of group members does not require extra communication, and changes of other members require one extra round.

BE is another well-established cryptographic primitive developed for secure group communications. As the core of BE is to generate and distribute the key materials to the participants, BE schemes are also referred to as key distribution schemes in some scenarios. While digital rights management motivated most previous BE schemes [27], [28], recent efforts [29], [30], [31], [32], [33], [34], [35] are devoted to modifying BE or key distribution technologies in view of securing emerging information systems such as sensor networks, mobile *ad hoc* networks, vehicular networks, etc.

BE schemes in the literature can be classified into two categories, *i.e.*, symmetric-key BE [1] and public-key BE [36]. 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. Similarly to the GKA setting, tree-based key

structures were independently proposed to improve efficiency in symmetric-key BE systems [37], [38], and further improved in [39] with $O(\log n)$ keys. Cheon *et al.* [40] presented an efficient symmetric BE scheme allowing new members to join the protocol anytime. Harn and Lin [41] proposed a group key transfer protocol. Their protocol is based on secret sharing and is considerably efficient, albeit it cannot revoke (compromised) users.

In the public-key BE setting, 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 [36] the first public-key BE scheme in which up to a threshold of users can be revoked. Subsequently, [42] presented a fully collusion-resistant public-key BE scheme exploiting new bilinear pairing technologies in which the key size, the ciphertext size, and the computation costs are $O(\sqrt{n})$. The scheme in [43] slightly reduces the size of the key and the ciphertexts, although it still has sub-linear complexity. The schemes presented in [44] strengthen the security concept of public-key BE schemes. As to performance, the sub-linear barrier $O(\sqrt{n})$ has not yet been broken. In [45], Lewko *et al.* proposed two elegant schemes with constant public and secret keys, although their ciphertext size is linear with the number of the revoked users, which is $O(n)$ in the worst case.

D. Paper Organization

The rest of this paper is organized as follows. In Section II, we model ConBE and define its security. In Section III, we present a collusion-resistant regular public-key BE scheme with aggregatability. Efficient ConBE schemes are realized in Section IV. We analyze the performance of our scheme in Section V and provide detailed proofs for the security results in Section VI. Finally, Section VII concludes the paper.

II. Modeling Contributory Broadcast Encryption

We begin by formalizing the ConBE notion bridging the GKA and BE primitives. In ConBE, 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. Since the negotiated public key is usually employed to transmit session keys, we define a ConBE scheme as a key encapsulation mechanism (KEM).

A. Syntax

We first define the algorithms that compose a ConBE scheme. Let $\lambda \in \mathbb{N}$ denote the security parameter. Suppose that a group of members $\{\mathcal{U}_1, \cdots, \mathcal{U}_n\}$ want to jointly establish a ConBE system, where n is a positive integer and each member $\mathcal{U}_i$ is indexed by i for $1 \leq i \leq n$. To focus on ConBE, we assume that the communications between members are authenticated. However, we do not assume any confidential channel during the execution of the protocol. Formally, a ConBE scheme $(\mathsf{ParaGen}, \mathsf{CBSetup}, \mathsf{CBEncrypt}, \mathsf{CBDecrypt})$ consists of the following four polynomial-time algorithms.

♦ $\mathsf{ParaGen}(1^\lambda)$. 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.

♦ $\mathsf{CBSetup}(\mathcal{U}_1(x_1), \cdots, \mathcal{U}_n(x_n))$. 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 authenticated and public 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 the group members. The group encryption gek is publicly accessible. If the algorithm aborts, it outputs NULL. Here, we leave the input system parameters implicit. We denote this procedure by $(\mathcal{U}_1(dk_1), \cdots, \mathcal{U}_n(dk_n); gek) \leftarrow \mathsf{CBSetup}(\mathcal{U}_1(x_1), \cdots, \mathcal{U}_n(x_n))$.

♦ $\mathsf{CBEncrypt}(\mathbb{S}, gek)$. 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 $\mathbb{S} \subseteq \{1, \cdots, n\}$ and the public group encryption key gek, and it outputs a pair (c, ξ), where c is the ciphertext and ξ is the secret session key in a key space $\mathbb{K}$. Then $(c, \mathbb{S})$ is sent to the receivers.

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

A ConBE scheme is correct if the members in the receiver set can always correctly decrypt when the members and the sender follow the scheme honestly. Formally, it is defined as follows.

Definition 1 (Correctness)**.** *A ConBE scheme is said to be 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 $(c, \xi) \leftarrow \mathsf{CBEncrypt}(\mathbb{S}, gek)$, it holds that $\mathsf{CBDecrypt}(\mathbb{S}, j, dk_j, c) = \xi$ for any $j \in \mathbb{S}$.*

A trivial ConBE scheme can be constructed by concurrently encrypting to each member with her/his public key in a traditional public-key cryptosystem. Unfortunately, the trivial solution incurs a heavy encryption cost and produces ciphertexts whose size grows linearly with the number of receivers. Another option would be a BE scheme in which the public key is obtained by means of a multiparty computation protocol, but it would require extra communication and point-to-point confidential channels between the users. The challenge is to design ConBE schemes with efficient encryption and short ciphertexts.

B. Security Definitions

We next define the security of a ConBE scheme. Several methods have been proposed to transform public key encryption (PKE) with security against chosen-plaintext attacks (CPA) into encryption against adaptively chosen-ciphertext attacks (CCA2) in the standard model. In [48], Canetti *et al.* suggested conversion from CPA-secure IBE to CCA2-secure PKE using a one-time signature. In [49], Matsuda and Hanaoka proposed to obtain a CCA2-secure PKE from any CPA-secure PKE with a universal computational extractor. In

[50], Liu *et al.* obtained a CCA2-secure ABE from a CPA-secure ABE without extra cryptographic primitives, but with an additional on-the-fly dummy attribute. We note that these methods are applicable to our ConBE setting with/without modification (*e.g.*, by adding an on-the-fly dummy receiver). The cost depends on the methods, *i.e.*, a universal computational extractor, a one-time signature or a dummy user. Hence, it is sufficient to only define the CPA security of a ConBE scheme. However, noting that ConBE is designed for distributed applications where the users are likely to be corrupted, we include full collusion resistance into our security definition.

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

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

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

- Execute. $\mathcal{A}$ uses the identities of n members $\mathcal{U}_1, \cdots, \mathcal{U}_n$ to query $\mathcal{CH}$. The challenger runs 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 CBSetup to $\mathcal{A}$.
- Corrupt. $\mathcal{A}$ sends i to the Corrupt oracle maintained by $\mathcal{CH}$, 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 CBSetup.
- Reveal. $\mathcal{A}$ sends i to the Reveal oracle maintained by $\mathcal{CH}$, 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 CBSetup.

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

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

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

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

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

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 Reveal oracle is used to capture the decryption key leakage after the ConBE system has been established. This difference can be used to differentiate the security against attacks during the set-up stage from the security against attacks after a ConBE system is deployed.

We assume that the communication channels between members are authenticated during the CBSetup stage to establish the group encryption key. This is to allow each user to validate that the received protocol transcripts are from authentic members. The most usual way to establish authenticated channels is through a public-key infrastructure (PKI): each user registers a public key to a certification authority CA and uses the corresponding private key to sign any message she generates during the CBSetup stage. Hence, the authenticity of the CBSetup transcript from a user can be verified by all other users. Note that after this stage has been completed and the group encryption key gek has been agreed upon, messages encrypted under this group key cannot be understood by CA, because the latter does not know the corresponding decryption keys. For instance, in a social network application, the social network operator can serve as the CA and certify the users' public keys used to authenticate communication. In this way, the operator is only partially trusted and cannot decrypt the encrypted messages subsequently shared among the users under gek.

III. An Aggregatable BE Scheme

In this section, we propose an efficient AggBE scheme that is essential to construct ConBE schemes.

A. Definitions of AggBE

A BE scheme [42], [1], [44] consists of the following probabilistic algorithms.

♦ BSetup(1^λ). Take as input a security parameter λ. Output the maximal size n of a group of broadcast receivers, and a BE public/secret key pair (PK, SK).

♦ BKeyGen(i, SK). Take as input an index $i \in \{1, \cdots, n\}$ and the secret key SK. Output a private key d_i for user i.

♦ BEncryption$(\mathbb{S}, PK)$. Take as input a receiver set $\mathbb{S} \subseteq \{1, \cdots, n\}$ and the public key PK. If $|\mathbb{S}| > n$, abort the protocol. Else if $|\mathbb{S}| \leq n$, output a pair (c, ξ) where c is called the ciphertext and $\xi \in \mathbb{K}$ is the message encryption key.

♦ BDecryption$(\mathbb{S}, i, d_i, c, PK)$. This algorithm allows each receiver to extract the message encryption key ξ from the ciphertext. Take as input the receiver set $\mathbb{S}$, the index $i \in \{1, \cdots, n\}$, the receiver's secret key d_i, the ciphertext c and the public key PK. If $|\mathbb{S}| \leq n$ and $i \in \mathbb{S}$, output the message encryption key ξ.

The security for BE is defined by an experiment between an attacker $\mathcal{A}$ and a challenger $\mathcal{CH}$. $\mathcal{A}$ is given the dealer's public key including the system parameters. $\mathcal{A}$ can adaptively query the decryption key of any user. At some point, the attacker specifies a challenge set $\mathbb{S}^*$. The constraint is that, for any $i \in \mathbb{S}^*$, the decryption key of user i has never been queried. The challenger sets $(c^*, \xi_0) \leftarrow$ BEncryption$(\mathbb{S}^*, PK)$ and $\xi_1 \leftarrow$

$\mathbb{K}$. It sets $b \leftarrow \{0,1\}$ and gives (c^*, ξ_b) to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$ for b and wins the game if $b = b'$. The adversary $\mathcal{A}$'s advantage in the game above is defined as $Adv^{\mathrm{BE}}_{\mathcal{A},n,N}(1^\lambda) = |\Pr[b = b'] - \frac{1}{2}|$.

Definition 3 (Adaptive security)**.** *We say that a BE scheme has adaptive security if, for any polynomial-time algorithm $\mathcal{A}$, its advantage $Adv^{BE}_{\mathcal{A},n,N}(1^\lambda)$ is negligible in λ.*

In [44], a slightly weaker notion of semi-adaptive security is defined. In this case, the attacker must commit to a set of indices $\mathbb{C}$ at the beginning of the above game. The attacker is allowed to query the decryption key of any user not in $\mathbb{C}$, and can choose any $\mathbb{S}^* \subseteq \mathbb{C}$ for a challenge ciphertext. Gentry and Waters also illustrate a generic transformation [44] to convert any semi-adaptively secure BE scheme into an adaptively secure one.

Before formalizing aggregatability, we define a weaker key homomorphic property for BE schemes. The key homomorphic property was first defined in the static broadcast encryption scenario by Wu *et al.* [3]. Recently, Boneh *et al.* extended this concept to the attribute-based encryption scenario [46]. For our dynamic BE scenario, the key homomorphism states that, by combining the decryption keys associated with the same index of different BE instances, one can obtain a functional decryption key associated with the same index of the combined BE instances.

Definition 4 (Key homomorphism)**.** *A BE scheme is said to be key homomorphic if for any two public/secret key pairs $(PK_1, SK_1), (PK_2, SK_2) \leftarrow$BSetup$(1^\lambda)$, any index $i \in \mathbb{S} \subseteq \{1, \cdots, n\}$, any $d_{1,i} =$BKeyGen(i, SK_1) and $d_{2,i} =$BKeyGen(i, SK_2), it holds that* BDecryption$(\mathbb{S}, i, d_{1,i} \boxdot d_{2,i}, c, PK_1 \circledast PK_2) = \xi$ *for any KEM ciphertext $(c, \xi) \leftarrow$BEncryption$(\mathbb{S}, PK_1 \circledast PK_2)$, where $\circledast : \Gamma \times \Gamma \to \Gamma$ and $\boxdot : \Omega \times \Omega \to \Omega$ are two efficient operations in the public key space Γ and the decryption key space Ω, respectively.*

The key homomorphic property just indicates that the combined decryption key works for the combined BE instance. It does not exclude the possibility that valid decryption keys for the combined BE instance might be obtained in other ways; in contrast, aggregatability excludes this possibility. 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 $i = 1, \cdots, n$. Formally, this property is defined as follows.

Definition 5 (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 $(\pi, PK_1, \cdots, PK_n)$, that is, the system parameters and the n independent public keys of the BE scheme.*

♦ **Corruption:** *For $1 \le i, j \le n$, 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 (indistinguishability under chosen-plaintext attack) 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 $Adv_{\mathcal{A}} \ge \varepsilon$ in the above aggregatability game.

B. An AggBE Scheme

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 $u, v \in \mathbb{Z}_p^*$, it holds that $e(g^u, g^v) = e(g, g)^{uv}$. Let $\Upsilon = (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow$ `PairGen`(1^λ), g be a generator of $\mathbb{G}$. Let $h_j \in \mathbb{G}$ be randomly chosen for $j = 1, \cdots, n$. The system parameters are $\pi = (\Upsilon, g, h_1, \cdots, h_n)$. Assume n users in the system. Our AggBE scheme extends the aggregatable signature-based broadcast [3] with user revocation and is constructed as follows.

♦ BSetup(1^λ): 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 $sk = ((r_0, X_0), \cdots, (r_n, X_n))$.

♦ BKeyGen(j, SK): For $j = 1, \cdots, n$, the private key of the user j is $d_j = (\sigma_{0,j}, \cdots, \sigma_{j-1,j}, \sigma_{j+1,j}, \cdots, \sigma_{n,j})$:

$$\sigma_{i,j} = X_i h_j^{r_i}.$$

♦ BEncryption$(\mathbb{S}, PK)$: Set $\overline{\mathbb{S}} = \{0, 1, \cdots, n\} \setminus \mathbb{S}$. Randomly pick t in $\mathbb{Z}_p^*$ and compute $c = (c_1, c_2)$:

$$c_1 = g^t, c_2 = (\prod_{i \in \overline{\mathbb{S}}} R_i)^t.$$

Set the session key $\xi = (\prod_{i \in \overline{\mathbb{S}}} A_i)^t$. Output (c, ξ) and send $(\mathbb{S}, c)$ to receivers.

♦ BDecryption$(\mathbb{S}, j, d_j, c, PK)$: If $j \in \mathbb{S}$, the receiver j extracts ξ from c with private key d_j by computing

$$e(\prod_{i \in \overline{\mathbb{S}}} \sigma_{i,j}, c_1) e(h_j, c_2) = \xi.$$

The correctness of the BE scheme above follows from direct verification of the following equalities

$$\begin{aligned} & e(\prod_{i \in \overline{\mathbb{S}}} \sigma_{i,j}, c_1) e(h_j, c_2) \\ = \; & e(\prod_{i \in \overline{\mathbb{S}}} X_i h_j^{r_i}, g^t) e(h_j, \prod_{i \in \overline{\mathbb{S}}} g^{-r_i t}) \\ = \; & e(\prod_{i \in \overline{\mathbb{S}}} X_i, g)^t = (\prod_{i \in \overline{\mathbb{S}}} A_i)^t = \xi. \end{aligned}$$

The security of our BE scheme relies on the well-established decision n-BDHE assumption [47].

Definition 6 (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 [44], our scheme is fully collusion-resistant under the decision n-BDHE assumption. The proof is given in Section VI-A. One can further apply the generic Gentry-Waters transformation [44] to convert our semi-adaptive BE scheme into an adaptively secure one.

Theorem 1. *The proposed BE scheme for dynamic groups has full collusion resistance 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 $\tau' = \tau + \mathcal{O}(n^2)\tau_{Exp}$, where τ_{Exp} is the time to compute an exponentiation in $\mathbb{G}$ or $\mathbb{G}_T$.*

One may observe that our BE scheme is key-homomorphic. Consider the system parameters defined as 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 $d_{j,1} = (\sigma_{0,j,1}, \cdots, \sigma_{j-1,j,1}, \sigma_{j+1,j,1}, \cdots, \sigma_{n,j,1}) \in \mathbb{G}^n$ 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 $dk_j = d_{j,1} \boxdot d_{j,2} = (\sigma_{0,j,1}\sigma_{0,j,2}, \cdots, \sigma_{j-1,j,1}\sigma_{j-1,j,2}, \sigma_{j+1,j,1}\sigma_{j+1,j,2}, \cdots, \sigma_{n,j,1}\sigma_{n,j,2})$. 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. Indeed, the following theorem shows that our BE scheme enjoys the stronger notion of aggregatability.

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 $i \neq j$.

IV. Proposed ConBE Scheme

In this section, we propose a ConBE 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.

A. High-Level Description

Our basic idea is to introduce the revocation mechanism of a regular BE scheme into the asymmetric GKA scheme [3]. 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 ConBE scheme, we also need to show that *only* the intended receivers can decrypt. This is ensured by the aggregatabilty of the underlying BE scheme.

B. The Proposal

Based on our aggregatable BE scheme, we implement a ConBE scheme with short ciphertexts. Assume that the group size is at most n. Let $\Upsilon = (p, \mathbb{G}, \mathbb{G}_T, e) \leftarrow \texttt{PairGen}(1^\lambda)$, 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 ConBE system consists of the following three procedures:

- Group Key Agreement. 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 $j = 1, \cdots, n$, $j \neq k$, compute $\sigma_{i,j,k} = X_{i,k} h_j^{r_{i,k}}$ for $i = 0, \cdots, n$, with $i \neq j$;
  - 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 $(PK_k, d_{1,k}, \cdots, d_{k-1,k}, d_{k+1,k}, \cdots, d_{n,k})$;
  - Compute $d_{k,k}$ accordingly and keep it 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 her 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 $\mathbb{S} \subseteq \{1, \cdots, n\}$ a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2015.2419662, IEEE Transactions on Computers

session key ξ. Set $\overline{\mathbb{S}} = \{0, 1, \cdots, n\} \setminus \mathbb{S}$. Randomly pick t in $\mathbb{Z}_p^*$ and compute the ciphertext $c = (c_1, c_2)$ where

$$c_1 = g^t, c_2 = (\prod_{i \in \overline{\mathbb{S}}} R_i)^t.$$

Output (c, ξ) where $\xi = (\prod_{i \in \overline{\mathbb{S}}} A_i)^t$. Send $(\mathbb{S}, c)$ to the receivers.

♦ CBDecrypt. If $j \in \mathbb{S}$, receiver j can extract ξ from the ciphertext c with decryption key d_j by computing

$$e(\prod_{i \in \overline{\mathbb{S}}} \sigma_{i,j}, c_1) e(h_j, c_2) = \xi.$$

The correctness of the scheme 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 Section VI-B.

Theorem 3. *The proposed ConBE scheme has fully collusion-resistant security 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 $\tau' = \tau + O((n^3)\tau_{Exp})$.*

C. Insecure Analog of ConBE Using Gentry-Waters BE

The above BE scheme bears some similarities to the Gentry-Waters BE scheme [44]. However, our BE scheme is aggregatable while the Gentry-Waters BE scheme is not. In this section, with the Gentry-Waters BE scheme as an example, we show that an analog of our ConBE scheme is insecure due to the lack of aggregatability of the Gentry-Waters BE scheme.

1) Review of the Gentry-Waters BE Scheme: Gentry and Waters presented a semi-adaptively secure BE scheme [44]. Let $h_1, \cdots, h_n$ and g be independent generators of a group $\mathbb{G}$ equipped with a bilinear map e. Assume that the order of $\mathbb{G}$ is a prime p. The Gentry-Waters BE scheme is as follows.

♦ BSetup(n, n): Randomly select x in $\mathbb{Z}_p^*$ and compute $g^x, e(g, g)^x$. The BE public key is $PK = e(g, g)^x$ and the BE secret key is $SK = g^x$.

♦ BKeyGen(i, SK): Run $r_i \leftarrow \mathbb{Z}_p^*$ and output user i's secret decryption key $s_i = (s_{i,0}, \cdots, s_{i,n})$ where

$$s_{i,0} = g^{-r_i}, s_{i,1} = h_1^{r_i}, \cdots, s_{i,i-1} = h_{i-1}^{r_i},$$
$$s_{i,i} = g^x h_i^{r_i}, s_{i,i+1} = h_{i+1}^{r_i}, \cdots s_{i,n} = h_n^{r_i}.$$

♦ BEnc$(\mathbb{S}, PK)$: Randomly pick t in $\mathbb{Z}_p^*$ and compute $c = (c_1, c_2)$ where

$$c_1 = g^t, c_2 = (\prod_{j \in \mathbb{S}} h_j)^t.$$

Set $\xi = e(g, g)^{xt}$ and output (c, k). Send $(\mathbb{S}, c)$ to the receivers.

♦ BDec$(\mathbb{S}, i, s_i, c, PK)$: If $i \in \mathbb{S}$, the receiver i extracts ξ from c with private key d_i by computing

$$e(s_{i,i} \prod_{j \in \mathbb{S} \setminus \{i\}} s_{i,j}, c_1) e(s_{i,0}, c_2) = e(\prod_{j \in \mathbb{S}} s_{i,j}, c_1) e(s_{i,0}, c_2)$$
$$= e((g^x \prod_{j \in \mathbb{S}} h_j^{r_i}), g^t) e(g^{-r_i}, (\prod_{j \in \mathbb{S}} h_j)^t)$$
$$= e(g, g)^{xt} = \xi.$$

We define $\boxdot$, $\circledast$, $\bigcirc$ as $s_{1_i} \boxdot s_{2_i} = (s_{1_{i,0}} s_{2_{i,0}}, \cdots, s_{1_{i,n}} s_{2_{i,n}})$, $PK_1 \circledast PK_2 = PK_1 PK_2$, $k_1 \bigcirc k_2 = k_1 k_2$, respectively. Then it is easy to verify that the Gentry-Waters BE scheme is key-homomorphic.

2) Analog of Our ConBE Using the Gentry-Waters BE Scheme: Following the same paradigm, it is easy to give an analog of our ConBE scheme by using the Gentry-Waters BE scheme. Assume the same system parameters as above. The analog of the ConBE can work as follows.

♦ CBSetup. This algorithm consists of the following procedures.

- Group Key Agreement. For $1 \le k \le n$, user k randomly chooses $x_k \in \mathbb{Z}_p^*$ and computes $PK_k = e(g, g)^{x_k}$ and

$$d_{i,k} = (s_{i,0,k}, s_{i,1,k}, \cdots, s_{i,k-1,k}, s_{i,k,k}, s_{i,k+1,k}, \cdots, s_{i,n,k}), \quad (1)$$

  where

$$s_{i,0,k} = g^{-r_{i,k}}, s_{i,1,k} = h_1^{r_{i,k}}, \cdots, s_{i,k-1,k} = h_{k-1}^{r_{i,k}},$$
$$s_{i,k,k} = g^{x_k} h_k^{r_{i,k}}, s_{i,k+1,k} = h_{k+1}^{r_{i,k}}, \cdots, s_{i,n,k} = h_n^{r_{i,k}}$$

  for randomly chosen $r_{i,k}$ from $\mathbb{Z}_p^*$. User k's private key is $d_{k,k}$. User k publicly broadcasts

$$\langle PK_k, d_{1,k}, \cdots, d_{k-1,k}, d_{k+1,k}, \cdots, d_{n,k} \rangle \quad (2)$$

- Group Encryption Key Derivation. Anyone can compute the group encryption key:

$$K = PK_1 \cdots PK_n = e(g, g)^{x_1 + \cdots + x_n} = e(g, g)^x,$$

  where we define $x = x_1 + \cdots + x_n$.
- Member Decryption Key Derivation. For $i = 1, \cdots, n$, user i can compute her decryption key

$$d_i = (s_{i,0}, s_{i,1}, \cdots, s_{i,i-1}, s_{i,i}, s_{i,i+1}, \cdots, s_{i,n}),$$

  where

$$s_{i,0} = \prod_{k=1}^{n} s_{i,0,k}, \cdots, s_{i,n} = \prod_{k=1}^{n} s_{i,n,k}.$$

  Define $r_i = r_{i,1} + \cdots + r_{i,n}$ for $1 \le i \le n$. Then we have that

$$s_{i,0} = g^{-r_i}, s_{i,1} = h_1^{r_i}, \cdots, s_{i,i-1} = h_{i-1}^{r_i},$$
$$s_{i,i} = g^x h_i^{r_i}, s_{i,i+1} = h_{i+1}^{r_i}, \cdots, s_{i,n} = h_n^{r_i}.$$

♦ CBEncrypt. Decide the receiver set $\mathbb{S} \subseteq \{1, \cdots, n\}$. Invoke the underlying Gentry-Waters encryption algorithm to compute the ciphertext $c = (c_1, c_2)$:

$$c_1 = g^t, c_2 = (\prod_{j \in \mathbb{S}} h_j)^t$$

where t is randomly chosen from $\mathbb{Z}_p^*$. Set

$$\xi = K^t = e(g, g)^{t(x_1 + \cdots + x_n)} = e(g, g)^{tx}$$

and send $(\mathbb{S}, c)$ to the receivers.

♦ CBDecrypt. If $i \in \mathbb{S}$, the user i can extract ξ from c with her decryption key d_i by computing

$$e(s_{i,i} \prod_{j \in \mathbb{S} \setminus \{i\}} s_{i,j}, c_1) e(s_{i,0}, c_2) = e(\prod_{j \in \mathbb{S}} s_{i,j}, c_1) e(s_{i,0}, c_2)$$

$$= e((g^x \prod_{j \in \mathbb{S}} h_j^{r_i}), g^t) e(g^{-r_i}, (\prod_{j \in \mathbb{S}} h_j)^t) = e(g,g)^{xt} = \xi.$$

3) Attack on the Analog: In the sequel we show that the above Gentry-Waters BE-based ConBE scheme is insecure. An explicit attack is presented to allow an attacker to decrypt any ciphertext encrypted to any subset of the group members. The attacker only needs to see the public key of the users and the ciphertext, both of which are transmitted over public channels. The attack proceeds as follows.

Seeing the public protocol transcripts (Formula (2)) $\langle PK_k, d_{1,k}, \cdots, d_{k-1,k}, d_{k+1,k}, \cdots, d_{n,k} \rangle$ from users $k = 1, \cdots, n$, the attacker can know (from Formula (1)):

$$s_{i,0,k} = g^{-r_{i,k}}, s_{i,1,k} = h_1^{r_{i,k}}, \cdots, s_{i,k-1,k} = h_{k-1}^{r_{i,k}},$$

$$s_{i,k,k} = g^{x_k} h_k^{r_{i,k}}, s_{i,k+1,k} = h_{k+1}^{r_{i,k}}, \cdots, s_{i,n,k} = h_n^{r_{i,k}}$$

for $i = 1, \cdots, n, i \neq k$. The attacker also knows the ciphertext $(c_1, c_2) = (g^t, (\prod_{j \in \mathbb{S}} h_j)^t)$. For each $k = 1, \cdots, n$, the attacker can compute

$$\xi_k = e(\prod_{j \in \mathbb{S}} s_{i,j,k}, c_1) e(s_{i,0,k}, c_2)$$

$$= e(g^{x_k} (\prod_{j \in \mathbb{S}} h_j)^{r_{i,k}}, g^t) e(g^{-r_{i,k}}, (\prod_{j \in \mathbb{S}} h_j)^t)$$

$$= e(g,g)^{x_k t}.$$

Then the attacker can decrypt the ciphertext by computing

$$\prod_{k=1}^{n} \xi_k = \prod_{k=1}^{n} e(g,g)^{x_k t} = e(g,g)^{(x_1 + \cdots + x_n)t} = \xi.$$

The attacker obtains the secret session key if he knows the public transcripts of the CBSetup sub-protocol and the ciphertext. Hence, the construction based on the Gentry-Waters BE scheme is insecure.

We observe that the above attack roots in a specific property (which we call shadow property) of the Gentry-Waters BE scheme. Suppose that there are two instances sharing the system parameters of the Gentry-Waters BE scheme. Their public keys are $PK = e(g,g)^x$ and $PK' = e(g,g)^{x'}$, respectively. Assume that a user indexed by i in the first instance has secret decryption key s_i computed from secret value r_i and the master secret key x corresponding to $PK = e(g,g)^x$, and a user also indexed by i (the users identified by the same index in two BE instances can be different or not) in another instance has secret decryption key s'_i computed from secret value r'_i and the master secret key x' corresponding to $PK' = e(g,g)^{x'}$, as defined in the Gentry-Waters BE scheme. Let $(c_1, c_2) = (g^t, (\prod_{j \in \mathbb{S}} h_j)^t)$ be the ciphertext sent to a receiver group $\mathbb{S}$ in the first instance. Then any receiver in $\mathbb{S}$ in the first instance can decrypt the session key $e(g,g)^{xt}$. However, a user with the same index in $\mathbb{S}$ in the second instance can extract a value $e(g,g)^{x't}$ which has a meaningful relationship with the decrypted value $e(g,g)^{xt}$ by the intended receivers. Hence, the value $e(g,g)^{x't}$ extracted by the attackers can be viewed as a shadow of the original value $e(g,g)^{xt}$.

The above shadow property of the Gentry-Waters BE scheme does not affect the security of their proposal as a regular BE scheme. However, this property may prevent the Gentry-Waters BE scheme from being used as a building block for certain advanced protocols.

V. Performance Analysis

A. Theoretical Analysis

We first examine the online complexity that is critical for the practicality of a ConBE scheme. When evaluating the performance, we use the widely adopted metrics [42], [43], [44] for regular BE schemes. In these metrics, the costs of simple operations (*e.g.*, read the indices of receivers and perform some simple quantifications of group elements associated to these indices) and communication (*e.g.*, the binary representation of the receivers' set) are not taken into consideration. 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 $\mathbb{G}_T$. 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. For decryption, a receiver needs to compute two single-base bilinear pairings (or one double-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 ConBE system. The overhead incurred by this procedure is $O(n^2)$. This procedure needs to be run only once and this can be done offline before the online transmission of secret session keys. For instance, in the social networks example, a number of friends exchange their CBSetup transcripts and establish a ConBE system to secure their subsequent sharing of private picture/videos. Since ConBE 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 [42] 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 ConBE to each subgroup, we obtain a ConBE 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 ConBE scheme, which produces $O(n_1^2)$-size ciphertexts. Consequently, we obtain a semi-adaptive ConBE scheme with $O(n^{\frac{2}{3}})$ complexity. This is comparable to up-to-date public-key BE systems whose complexity is $O(n^{\frac{1}{2}})$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2015.2419662, IEEE Transactions on Computers

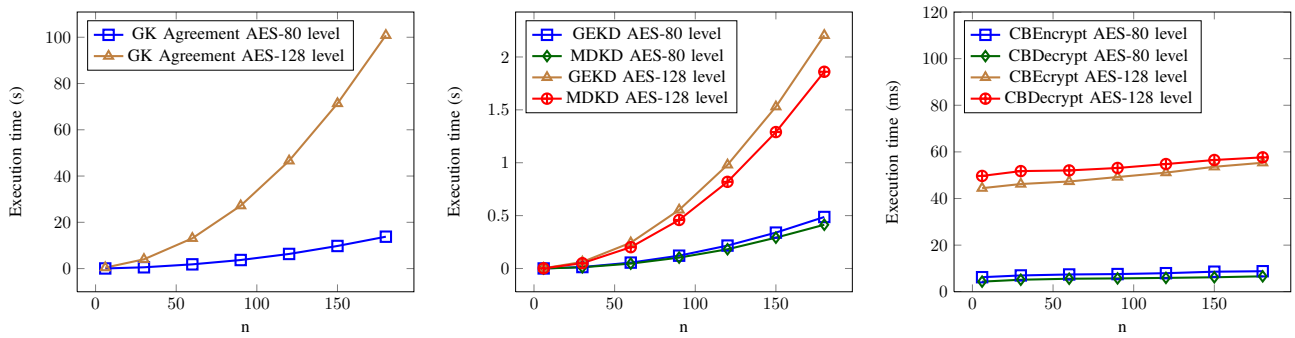

Fig. 1. Execution time of Group Key Agreement, Group Encryption Key Derivation, Member Decryption Key Derivation, CBEncrypt, and CBDecrypt for AES-80 and AES-128 levels.

B. Experimental Analysis

In this section we present experimental results on our ConBE scheme. The experiments were run on a PC with Intel Core i7-2600 CPU at 3.4GHz, using the C programming language. The cryptographic operations were implemented using the Pairing-Based Cryptography library[2]. Following the NIST-2012 key size recommendation[3], we realized our protocol for a moderate AES-80 level and a more usual AES-128 level, corresponding to the security level of an ideal symmetric cipher with 80-bit and 128-bit secret keys, respectively. We used Type A pairings constructed on the curve $y^2 = x^3 + x$ with embedding degree 2. Accordingly, in the first case for AES-80 level, $\mathbb{G}$ has 512-bit elements of a 160-bit prime order and $\mathbb{G}_T$ has 1024-bit/128-byte elements; and in the second case for AES-128 level, $\mathbb{G}$ has 1536-bit elements of a 256-bit prime order and $\mathbb{G}_T$ has 3072-bit/386-byte elements, respectively.

We performed experiments on the offline procedures including Group Key Agreement, Group Encryption Key Derivation and Member Decryption Key Derivation, and the online procedures including CBEncrypt andCBDecrypt for different group sizes $n = 6, 30, 60, 90, 120, 150, 180$. The values for CBEncrypt and CBDecrypt consider the worst case, *i.e.*, $|\mathbb{S}| = 1$. Also, we did not optimize the underlying pairing-related parameters or operations, *e.g.*, by choosing a large prime characteristic of the base field and the prime order p with most bits 0 (or 1), and by accelerating multi-base exponentiations/multi-base pairings [51]. Hence, the practical performance of our protocol can be better than the illustrated experimental results.

In Figure 1, the security level of our protocol is measured by the secret key size of AES (assumed to be an ideal symmetric cipher), *i.e.*, AES with a truncated 80-bit key and AES with a standard 128-bit key. The leftmost graph in the figure illustrates the group key agreement time for different group sizes and different security levels. The execution time grows almost quadratically with the group size, and also grows with the security level. This is consistent with our theoretical analysis, because the pairings and the exponentiations dominate the computation costs. To achieve a moderate 128-bit security, the execution time is about 3 minutes for a group of 180 users. This is realistic as the GKA procedure only needs to be run once and then one can broadcast to any subset of the users, without re-running the protocol or any extra revocation sub-protocol.

The central graph in Figure 1 shows the time to extract the group encryption key and the decryption key for different group sizes and different security levels. Similarly to the group key agreement time, the key extraction time also grows with the security level and the group size. However, even in the worst case, only about 3 seconds are required, which is affordable in practice.

The rightmost graph in Figure 1 illustrates the online session key encryption/decryption time. It can be seen that the time is almost constant for different group sizes, which is consistent with the theoretical analysis. Both the session key encryption and decryption take less than 10ms for a 80-bit security level, and less than 80ms for a 128-bit security level. After the system is set up, the session key transmission is really efficient, which is user-friendly and definitely makes our ConBE scheme practical.

We also performed experiments on cost tradeoff between set-up and online encryption. For $n = 180$ and AES-128 level, the execution times for Group Key Agreement, Group Encryption Key Derivation, Member Decryption Key Derivation, CBEncrypt and CBDecrypt are 101s, 2.20s, 1.86s, 55.3ms, and 57.6ms, respectively. However, using the trade-off described in the previous section, specifically taking subgroups of 6 users, the times become 410ms, 2.05ms, 1.63ms, 1.33s, and 57.6ms. The set-up efficiency was significantly improved, at the cost of a 1.33s encryption time, to be compared to a 55.3ms encryption time without tradeoff.

VI. Security Proofs

A. Proof of Theorem 1

Proof: A semi-adaptive attacker must commit to a set of the group members at the beginning of the game. She is allowed to corrupt all the users outside the committed set. Finally, she can choose any subset of the committed set as a target set to attack and try to get useful information sent to the target group. Suppose that $\mathcal{A}$ is a semi-adaptive τ-time adversary breaking our BE scheme with advantage ε for a system parameterized with a given n. We build an algorithm $\mathcal{B}$ with advantage ε in solving the decision n-BDHE problem in time τ'.

[2]Version 0.5.12, available at http://crypto.stanford.edu/pbc.
[3]http://www.keylength.com/en/4/.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2015.2419662, IEEE Transactions on Computers

$\mathcal{A}$ commits to a set $\mathbb{C} \subseteq \{1, \cdots, n\}$ to $\mathcal{B}$. $\mathcal{B}$ queries the decision n-BDHE challenger and obtains a random decision n-BDHE challenge $(g, g^t, \overrightarrow{y}_{g,\alpha,n}, Z)$, where $\overrightarrow{y}_{g,\alpha,n} = (g_1, \cdots, g_n, g_{n+2}, \cdots, g_{2n}) = (g^{\alpha^1}, \cdots, g^{\alpha^n}, g^{\alpha^{n+2}}, \cdots, g^{\alpha^{2n}})$ and Z is either $e(g_{n+1}, g^t)$ or a random element of $\mathbb{G}_T$. $\mathcal{B}$ proceeds as follows.

Preparation for simulation.

For $j = 1, \cdots, n$, $\mathcal{B}$ randomly selects $v_j \in \mathbb{Z}_p^*$ and computes $h_j = g_j g^{v_j}$. Denote $\overline{\mathbb{C}} = \{1, \cdots, n\} \setminus \mathbb{C}$.

For $i \in \overline{\mathbb{C}} \cup \{0\}$, randomly select $a_i, r_i \in \mathbb{Z}_p^*$. For $j \in \overline{\mathbb{C}}$, compute

$$R_0 = g^{r_0}(\prod_{k\in\overline{\mathbb{C}}} g_{n+1-k}), A_0 = e(g,g)^{a_0+\alpha^{n+1}},$$
$$\sigma_{0,j} = g^{a_0} g_j^{-r_0}(\prod_{k\in\overline{\mathbb{C}}}^{k\neq j} g^{-1}_{n+1-k+j}) R_0^{-v_j}. \quad (3)$$

For $i \in \overline{\mathbb{C}}$ and $j \neq i$, compute

$$R_i = g^{r_i} g^{-1}_{n+1-i}, A_i = e(g,g)^{a_i},$$
$$\sigma_{i,j} = g^{a_i} g_j^{-r_i} g_{n+1-i+j} R_i^{-v_j}. \quad (4)$$

For $i \in \mathbb{C}$ and $j \in \{1, \cdots, n\}$, compute

$$R_i = g^{r_i}, A_i = e(g,g)^{a_i}, \sigma_{i,j} = g^{a_i} h_j^{-r_i}. \quad (5)$$

Then $\mathcal{B}$ can answer all the queries from $\mathcal{A}$.

Query public key. $\mathcal{A}$ can query the BE public key as well as the system parameters $\pi = ((p, \mathbb{G}, \mathbb{G}_T, e), g, h_1, \cdots, h_n)$ and the maximum group size n. From the decision n-BDHE challenge, the simulation of π is straightforward. $\mathcal{B}$ needs to generate a BE public key $PK = (pk_0, pk_1, \cdots, pk_n)$, where pk_i is the public key of the underlying aggregatable signature-based broadcast [3]. $\mathcal{B}$ sets $pk_i = (R_i, A_i)$ and forwards them to $\mathcal{A}$. Note that r_i and a_i are uniformly distributed in $\mathbb{Z}_p^*$, so the simulated public keys have an identical distribution as in the real world, and the simulation is perfect.

Query decryption key. $\mathcal{A}$ can query the decryption key of any user $j \in \overline{\mathbb{C}} = \{1, \cdots, n\} \setminus \mathbb{C}$. $\mathcal{B}$ returns $(\sigma_{0,j}, \cdots, \sigma_{j-1,j}, \sigma_{j+1,j}, \cdots, \sigma_{n,j})$. Now we show that the simulated decryption keys are well formed and perfect.

For the case that $i = 0$ and $j \in \overline{\mathbb{C}}$, from Equation (3), the following equations hold.

$$\begin{aligned}
&e(\sigma_{0,j}, g) e(h_j, R_0) \\
&= e(g^{a_0} g_j^{-r_0} (\prod_{k\in\overline{\mathbb{C}}}^{k\neq j} g^{-1}_{n+1-k+j}) R_0^{-v_j}, g) e(g_j g^{v_j}, R_0) \\
&= e(g^{a_0} g_j^{-r_0} (\prod_{k\in\overline{\mathbb{C}}}^{k\neq j} g^{-1}_{n+1-k+j}), g) e(g_j, g^{r_0}(\prod_{k\in\overline{\mathbb{C}}} g_{n+1-k})) \\
&= e(g^{a_0} (\prod_{k\in\overline{\mathbb{C}}}^{k\neq j} g^{-1}_{n+1-k+j}), g) e(g, (\prod_{k\in\overline{\mathbb{C}}} g_{n+1-k+j})) \\
&= e(g^{a_0}, g) e(g, g_{n+1}) = e(g,g)^{a_0+\alpha^{n+1}} = A_0. \quad (6)
\end{aligned}$$

For the case $i \in \overline{\mathbb{C}}$ and $j \neq i$, from Equation (4), the following equations hold.

$$\begin{aligned}
&e(\sigma_{i,j}, g) e(h_j, R_i) \\
&= e(g^{a_i} g_j^{-r_i} g_{n+1-i+j} R_i^{-v_j}, g) e(g_j g^{v_j}, R_i) \\
&= e(g^{a_i} g_j^{-r_i} g_{n+1-i+j}, g) e(g_j, R_i) \\
&= e(g^{a_i} g_j^{-r_i} g_{n+1-i+j}, g) e(g_j, g^{r_i} g^{-1}_{n+1-i}) \\
&= e(g^{a_i} g_{n+1-i+j}, g) e(g_j, g^{-1}_{n+1-i}) \\
&= e(g^{a_i} g_{n+1-i+j}, g) e(g, g^{-1}_{n+1-i+j}) \\
&= e(g,g)^{a_i} = A_i. \quad (7)
\end{aligned}$$

For the case that $i \in \mathbb{C}$ and $j \in \{1, \cdots, n\}$, from Equation (5), the following equation holds.

$$\begin{aligned}
&e(\sigma_{i,j}, g) e(h_j, R_i) \\
&= e(g^{a_i} h_j^{-r_i}, g) e(h_j, g^{r_i}) \\
&= e(g,g)^{a_i} = A_i. \quad (8)
\end{aligned}$$

Hence, for $j \in \overline{\mathbb{C}}$, $i = 0, \cdots, n$ and $i \neq j$, we have that

$$e(\sigma_{i,j}, g) e(h_j, R_i) = A_i. \quad (9)$$

Since g is a generator of $\mathbb{G}$, there exist $X_i \in \mathbb{G}$ and $\gamma_i \in \mathbb{Z}_p^*$ satisfying $e(X_i, g) = A_i$ and $R_i = g^{-\gamma_i}$. The above Equation (9) further implies that $\sigma_{i,j} = X_i h_j^{\gamma_i}$. Therefore, for user $j \in \overline{\mathbb{C}} = \{1, \cdots, n\} \setminus \mathbb{C}$, her decryption key $(\sigma_{0,j}, \cdots, \sigma_{j-1,j}, \sigma_{j+1,j}, \cdots, \sigma_{n,j})$ is well formed. The simulation of decryption keys for users outside $\mathbb{C}$ is perfect.

Query challenge ciphertext. At some point, the attacker $\mathcal{A}$ submits a target set $\mathbb{S}^* \subseteq \mathbb{C} \subseteq \{1, \cdots, n\}$ for a challenge ciphertext sent to $\mathbb{S}^*$. Since $\mathbb{S}^* \subseteq \mathbb{C}$, we have that $\overline{\mathbb{S}}^* = \{0, 1, \cdots, n\} \setminus \mathbb{S}^* \supseteq \{0, 1, \cdots, n\} \setminus \mathbb{C} = \overline{\mathbb{C}} \cup \{0\}$. Notice that $\mathcal{B}$ knows Z and g^t from the decision n-BDHE challenger, and the values of $r_i, a_i \in \mathbb{Z}_p^*$ which are chosen during the preparation for the simulation for $i = 1, \cdots, n$. Hence $\mathcal{B}$ can compute

$$c_1^* = g^t, c_2^* = (g^t)^{\sum_{i\in\overline{\mathbb{S}}^*} r_i}, \xi^* = Z e(g^t, g)^{\sum_{i\in\overline{\mathbb{S}}^*} a_i}. \quad (10)$$

The algorithm $\mathcal{B}$ sets $c^* = (c_1^*, c_2^*)$ and challenges $\mathcal{A}$ with (c^*, ξ^*). In the following we show that (c^*, ξ^*) is well formed.

Define $\mathbb{S}' = \overline{\mathbb{S}}^* \setminus \{\overline{\mathbb{C}} \cup \{0\}\}$. Then $\mathbb{S}' \subseteq \mathbb{C}$ and $\overline{\mathbb{S}}^* = \overline{\mathbb{C}} \cup \{0\} \cup \mathbb{S}'$. From Equations (3, 4, 5), the following equations hold:

$$\begin{aligned}
&(\prod_{i\in\overline{\mathbb{S}}^*} R_i)^t = (\prod_{i\in\overline{\mathbb{C}}\cup\{0\}\cup\mathbb{S}'} R_i)^t = (R_0 \prod_{i\in\overline{\mathbb{C}}} R_i \prod_{i\in\mathbb{S}'} R_i)^t \\
&= (g^{r_0}(\prod_{k\in\overline{\mathbb{C}}} g_{n+1-k}) \prod_{i\in\overline{\mathbb{C}}} g^{r_i} g^{-1}_{n+1-i} \prod_{i\in\mathbb{S}'} g^{r_i})^t \\
&= (g^{r_0} \prod_{i\in\overline{\mathbb{C}}} g^{r_i} \prod_{i\in\mathbb{S}'} g^{r_i})^t = (g^{\sum_{i\in\overline{\mathbb{S}}^*} r_i})^t \\
&= (g^t)^{\sum_{i\in\overline{\mathbb{S}}^*} r_i} = c_2; \quad (11)
\end{aligned}$$

$$(\prod_{i\in\overline{\mathbb{S}}^*} A_i)^t = e(g,g)^{t\alpha^{n+1}} e(g,g)^{t\sum_{i\in\overline{\mathbb{S}}^*} a_i}. \quad (12)$$

Hence, (c_1^*, c_2^*) is a well-formed ciphertext of the session key ξ if $Z = e(g,g)^{t\alpha^{n+1}}$. Else if Z is chosen at random

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TC.2015.2419662, IEEE Transactions on Computers

from $\mathbb{G}_T$, (c_1^*, c_2^*) is also well formed but independent of ξ. Therefore, $\mathcal{B}$ can answer the decision n-BDHE challenge that $Z = e(g,g)^{t\alpha^{n+1}}$ if and only if $\mathcal{A}$ answers that c^* is a ciphertext of ξ. Algorithm $\mathcal{B}$ has the same success probability as $\mathcal{A}$ to break the above BE scheme.

Time complexity: $\mathcal{B}$'s overhead is dominated by computing h_j and $(\sigma_{i,j}, R_i, A_i)$ for $j \neq i$. Computing h_j requires $O(n)$ exponentiations in $\mathbb{G}$. Computing $\sigma_{i,j}$ requires $O(n^2)$ exponentiations in $\mathbb{G}$. Computing R_i requires $O(n)$ exponentiations in $\mathbb{G}$. $\mathcal{B}$ can compute A_i by $O(n)$ exponentiations in $\mathbb{G}_T$. Let τ_{Exp} denote the time to compute one exponentiation in $\mathbb{G}$ or $\mathbb{G}_T$. The time complexity of $\mathcal{B}$ is $\tau' = \tau + O(n^2)\tau_{\text{Exp}}$. ■

B. Proof of Theorem 3

Proof: Suppose that $\mathcal{A}$ is a semi-adaptive τ-time adversary breaking our ConBE scheme with advantage ε for a system parameterized with n. We build an algorithm $\mathcal{B}$ with advantage ε in solving the decision n-BDHE problem in time τ'.

$\mathcal{A}$ commits to a set $\mathbb{C} \subseteq \{1, \cdots, n\}$ to $\mathcal{B}$. $\mathcal{B}$ queries the decision n-BDHE challenger and obtains a random decision n-BDHE challenge $(g, g^t, \overrightarrow{y}_{g,\alpha,n}, Z)$, where

$$\begin{aligned}\overrightarrow{y}_{g,\alpha,n} &= (g_1, \cdots, g_n, g_{n+2}, \cdots, g_{2n}) \\ &= (g^{\alpha^1}, \cdots, g^{\alpha^n}, g^{\alpha^{n+2}}, \cdots, g^{\alpha^{2n}})\end{aligned}$$

and Z is either $e(g_{n+1}, g^t)$ or a random element of $\mathbb{G}_T$. Denote $\overline{\mathbb{C}} = \{1, \cdots, n\} \setminus \mathbb{C}$. $\mathcal{B}$ proceeds as follows.

Preparation for simulation. For the sake of clarity, we let $\mathcal{B}$ first prepare for all the answers of various possible queries that the attacker $\mathcal{A}$ may query. Assuming the same parameter setting as in the proof of Theorem 1, $\mathcal{B}$ prepares the answers as follows.

For $j = 1, \cdots, n$, compute $h_j = g_j g^{v_j}$ where v_j is chosen at random in $\mathbb{Z}_p^*$.

Case 0: $k \in \overline{\mathbb{C}}$. In this case, $\mathcal{B}$ does as in the real scheme. $\mathcal{B}$ randomly selects $a_{i,k}, \gamma_{i,k} \in \mathbb{Z}_p^*$ and computes

$$R_{i,k} = g^{\gamma_{i,k}}, A_{i,k} = e(g,g)^{a_{i,k}}, \sigma_{i,j,k} = g^{a_{i,k}} h_j^{-\gamma_{i,k}}.$$

In this case, we have that

$$e(\sigma_{i,j,k}, g)e(h_j, R_{i,k}) = e(g,g)^{a_{i,k}} = A_{i,k}. \quad (13)$$

Case 1: $k \in \mathbb{C}$.

Case 1.1: $i = 0$. $\mathcal{B}$ randomly selects $k^* \in \mathbb{C}$ and sets $\overline{\mathbb{C}}_{k^*} = \{1, \cdots, n\} \setminus \{k^*\}$.

Case 1.1.1: $k = k^*$. Randomly select $a_{0,k^*}, \gamma_{0,k^*} \in \mathbb{Z}_p^*$ and compute

$$R_{0,k^*} = g^{\gamma_{0,k^*}}\Big(\prod_{\ell \in \overline{\mathbb{C}}_{k^*}} g_{n+1-\ell}\Big), A_{0,k^*} = e(g,g)^{a_{0,k^*}} e(g,g)^{\alpha^{n+1}},$$

$$\sigma_{0,j,k^*} = g^{a_{0,k^*}} g_j^{-\gamma_{i,k^*}}\Big(\prod_{\ell \in \overline{\mathbb{C}}_{k^*}}^{\ell \neq j} g_{n+1-\ell+j}^{-1}\Big) R_{0,k^*}^{-v_j}, j \neq k^*.$$

In this case, one can verify that for $j \neq k^* \in \mathbb{C}$

$$\begin{aligned}&e(\sigma_{0,j,k^*}, g)e(h_j, R_{0,k^*}) = e(g,g)^{a_{0,k^*}} e(g,g)^{\alpha^{n+1}} \\ &= A_{0,k^*}. \quad (14)\end{aligned}$$

Case 1.1.2: $k \neq k^*$ and $k \in \mathbb{C}$. Randomly select $a_{0,k}, \gamma_{0,k} \in \mathbb{Z}_p^*$ and compute

$$R_{0,k} = g^{\gamma_{0,k}} g_{n+1-k}^{-1}, A_{0,k} = e(g,g)^{a_{0,k}},$$

$$\sigma_{0,j,k} = g^{a_{0,k}} g_j^{-\gamma_{0,k}} (g_{n+1-k+j}) R_{0,k}^{-v_j}, j \neq k.$$

In this case, one can verify that for $j \neq k, k \neq k^* \in \mathbb{C}$

$$e(\sigma_{0,j,k})e(h_j, R_{0,k}) = e(g,g)^{a_{0,k}} = A_{0,k}. \quad (15)$$

Case 1.2: $i = 1, \cdots, n$

Case 1.2.1: $i \in \overline{\mathbb{C}}$ and $k = k^*$. Randomly select $a_{i,k}, \gamma_{i,k} \in \mathbb{Z}_p^*$ and compute

$$R_{i,k} = g^{\gamma_{i,k}} g_{n+1-i}^{-1}, A_{i,k} = e(g,g)^{a_{i,k}},$$

$$\sigma_{i,j,k} = g^{a_{i,k}} g_j^{-\gamma_{i,k}} g_{n+1-i+j} R_{i,k}^{-v_j}, j \neq i.$$

In this case, one can verify that

$$e(\sigma_{i,j,k}, g)e(h_j, R_{i,k}) = e(g,g)^{a_{i,k}} = A_{i,k}, j \neq i. \quad (16)$$

Case 1.2.2: $i \in \mathbb{C}$ or $k \neq k^*$. Randomly select $a_{i,k}, \gamma_{i,k} \in \mathbb{Z}_p^*$ and compute

$$R_{i,k} = g^{\gamma_{i,k}}, A_{i,k} = e(g,g)^{a_{i,k}}, \sigma_{i,j,k} = g^{a_{i,k}} g_j^{-\gamma_{i,k}}.$$

In this case, one can verify that

$$e(\sigma_{i,j,k}, g)e(h_j, R_{i,k}) = e(g,g)^{a_{i,k}} = A_{i,k}, j \neq i. \quad (17)$$

By summarizing Equations (13, 14, 15, 16, 17), we have the following equations:

$$e(\sigma_{i,j,k}, g)e(h_j, R_{i,k}) = e(g,g)^{a_{i,k}} = A_{i,k}, k \in \overline{\mathbb{C}}; \quad (18)$$

$$\begin{aligned}&e(\sigma_{0,j,k^*})e(h_j, R_{0,k^*}) = e(g,g)^{a_{0,k^*}} e(g,g)^{\alpha^{n+1}} \\ &= A_{0,k^*}, j \neq k^* \in \mathbb{C}; \quad (19)\end{aligned}$$

$$\begin{aligned}&e(\sigma_{i,j,k}, g)e(h_j, R_{i,k}) = e(g,g)^{a_{i,k}} \\ &= A_{i,k}, j \neq k, k \in \mathbb{C}, k \neq k^*, j \neq i. \quad (20)\end{aligned}$$

After the preparation above, $\mathcal{B}$ can answer all the queries from $\mathcal{A}$.

Query transcript. $\mathcal{A}$ can query the system parameters and the transcripts from all the group members participating in the CBSetup sub-protocol. The system parameters except h_j can be trivially simulated from the decision n-BDHE challenge. As in the preparation for simulation, $h_j = g_j g^{v_j}$ for a randomly chosen value $v_j \in \mathbb{Z}_p^*$. Hence, all the system parameters are correctly simulated. Upon receiving the query for the transcripts from the members, $\mathcal{B}$ responds with

$$\begin{aligned}M = \{&(\sigma_{i,j,k}, R_{i,k}, A_{i,k}) | 0 \leq i \leq n, 1 \leq j \leq n, \\ &1 \leq k \leq n, j \neq i, j \neq k\}.\end{aligned}$$

Due to Equations (18, 19, 20), one can see that transcripts in M are well formed. Furthermore, since $\gamma_{i,k}$ and $a_{i,k}$ are uniformly distributed in $\mathbb{Z}_p^*$, the simulated transcripts have an identical distribution as in the real world and the simulation is perfect.

Query secret inputs and internal states. $\mathcal{A}$ can query the secret inputs and internal states of members in $\{1, \cdots, n\} \setminus$

$\mathbb{C} = \overline{\mathbb{C}}$. For these members, their transcripts are generated as in the real scheme in Case 0. Hence, $\mathcal{B}$ can answer this query correctly.

Query decryption keys. Note that in our ConBE one can always compute the decryption key of a member if one knows the member's secret inputs and internal states during the CBSetup stage. Hence, the challenger $\mathcal{B}$ can handle these queries as those for secret inputs and internal states.

Query challenge ciphertext. In the test stage, the attacker $\mathcal{A}$ submits a target set $\mathbb{S}^* \subseteq \mathbb{C} \subseteq \{1, \cdots, n\}$ for a challenge ciphertext sent to $\mathbb{S}^*$.

Similarly to the proof of Theorem 1, since $\mathbb{S}^* \subseteq \mathbb{C}$, it follows that $\overline{\mathbb{S}}^* = \{0, 1, \cdots, n\} \backslash \mathbb{S}^* \supseteq \{0, 1, \cdots, n\} \backslash \mathbb{C} = \overline{\mathbb{C}} \cup \{0\}$. Then $\mathbb{S}' = \overline{\mathbb{S}}^* \backslash \{\overline{\mathbb{C}} \cup \{0\}\} \subseteq \mathbb{C}$. Hence, $\overline{\mathbb{S}}^* = \overline{\mathbb{C}} \cup \{0\} \cup \mathbb{S}'$.

Define $\sum_{i \in \overline{\mathbb{S}}^*} \sum_{k=1}^{n} \gamma_{i,k} = r$ and $\sum_{i \in \overline{\mathbb{S}}^*} \sum_{k=1}^{n} a_{i,k} = a$ which are known to $\mathcal{B}$ because $\gamma_{i,k}$ and $a_{i,k}$ are chosen by $\mathcal{B}$. Since $\mathcal{B}$ also knows Z and g^t from the decision n-BDHE challenger, $\mathcal{B}$ can compute the challenge ciphertext as follows:

$$c_1^* = g^t, c_2^* = (g^t)^r, \xi^* = Ze(g,g)^{at}.$$

Then $\mathcal{B}$ sets $c^* = (c_1^*, c_2^*)$ and sends (c^*, ξ^*). In the following, we show that (c^*, ξ^*) is well formed.

From Case 1.1, we have that

$$\prod_{k=1}^{n} R_{0,k} = (g^{\gamma_{0,k^*}} \prod_{\ell \in \overline{\mathbb{C}}_{k^*}} g_{n+1-\ell}) \prod_{k=1, k \neq k^*}^{n} R_{0,k}$$

$$= (g^{\gamma_{0,k^*}} \prod_{\ell \in \overline{\mathbb{C}}_{k^*}} g_{n+1-\ell})(\prod_{k \in \mathbb{C}, k \neq k^*} g_{n+1-k}^{-1}) g^{\sum_{k=1, k \neq k^*}^{n} \gamma_{0,k}}$$

$$= (\prod_{\ell \in \overline{\mathbb{C}}_{k^*}} g_{n+1-\ell} \prod_{k \in \mathbb{C}, k \neq k^*} g_{n+1-k}^{-1}) g^{\sum_{k=1}^{n} \gamma_{0,k}}.$$

From Case 1.2.1 and Case 1.2.2, we have that

$$\prod_{i \in \overline{\mathbb{C}}} \prod_{k=1}^{n} R_{i,k} = \prod_{i \in \overline{\mathbb{C}}} g_{n+1-i}^{-1} g^{\sum_{i \in \overline{\mathbb{C}}} \sum_{k=1}^{n} \gamma_{i,k}}$$

$$\prod_{i \in \mathbb{S}'} \prod_{k=1}^{n} R_{i,k} = g^{\sum_{i \in \mathbb{S}'} \sum_{k=1}^{n} \gamma_{i,k}}.$$

Note that $\overline{\mathbb{C}}_{k^*} = \mathbb{C} \cup \overline{\mathbb{C}} \setminus \{k^*\}$ and $\overline{\mathbb{S}}^* = \overline{\mathbb{C}} \cup \{0\} \cup \mathbb{S}'$. We have that

$$\prod_{k=1}^{n} R_{0,k} \prod_{i \in \overline{\mathbb{C}}} \prod_{k=1}^{n} R_{i,k} \prod_{i \in \mathbb{S}'} \prod_{k=1}^{n} R_{i,k} = g^{\sum_{i \in \overline{\mathbb{S}}^*} \sum_{k=1}^{n} \gamma_{i,k}} = g^r.$$

Hence the following equalities hold:

$$(\prod_{i \in \overline{\mathbb{S}}^*} R_i)^t = (\prod_{i \in \overline{\mathbb{S}}^*} \prod_{k=1}^{n} R_{i,k})^t$$

$$= ((\prod_{k=1}^{n} R_{0,k})(\prod_{i \in \overline{\mathbb{C}}} \prod_{k=1}^{n} R_{i,k})(\prod_{i \in \mathbb{S}'} \prod_{k=1}^{n} R_{i,k}))^t$$

$$= (g^r)^t = (g^t)^r = c_2^*.$$

So far, we obtain that $c_1^* = g^t, c_2^* = (\prod_{i \in \overline{\mathbb{S}}^*} R_i)^t$. Hence, (c_1^*, c_2^*) is well formed and the simulation of the challenge ciphertext is perfect.

Success probability. At some point, $\mathcal{A}$ answers whether (c_1^*, c_2^*) is a valid ciphertext for ξ^* or is independent of ξ^*. From Equations (18, 19, 20), we have that

$$(\prod_{i \in \overline{\mathbb{S}}^*} A_i)^t = (\prod_{i \in \overline{\mathbb{S}}^*} \prod_{k=1}^{n} A_{i,k})^t$$
$$= (e(g,g)^{\alpha^{n+1} + \sum_{i \in \overline{\mathbb{S}}^*} \sum_{k=1}^{n} a_{i,k}})^t = e(g,g)^{t\alpha^{n+1} + at}.$$

Note that $\xi^* = Ze(g,g)^{at}$. Hence, (c_1, c_2) is a valid ciphertext for the session key ξ^* if and only if $Z = e(g,g)^{t\alpha^{n+1}}$. Then $\mathcal{B}$ answers the decision n-BDHE challenger with $Z = e(g,g)^{t\alpha^{n+1}}$ if and only if $\mathcal{A}$ answers that c^* is a valid ciphertext for ξ^*. Clearly, $\mathcal{B}$ has the same success probability as the success probability of $\mathcal{A}$ breaking the above ConBE scheme.

Time-complexity: $\mathcal{B}$'s overhead is dominated by computing $(\sigma_{i,j,k}, R_{i,k}, A_{i,k})$ for $j \neq i, j \neq k$. Computing $\sigma_{i,j,k}$ requires $\mathcal{O}(n^3)$ exponentiations. Computing $R_{i,k}$ requires $\mathcal{O}(n^2)$ exponentiations. Computing $A_{i,k}$ needs $O(n^2)$ exponentiations. The time for $\mathcal{B}$ to solve the decision n-BDHE problem is $\tau' = \tau + \mathcal{O}(n^3)\tau_{\text{Exp}}$. ■

VII. Conclusions

In this paper, we formalized the ConBE primitive. In ConBE, 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 ConBE model, we instantiated an efficient ConBE scheme that is secure in the standard model. As a versatile cryptographic primitive, our novel ConBE notion opens a new avenue to establish secure broadcast channels and can be expected to secure numerous emerging distributed computation applications.

Acknowledgments and Disclaimer

The authors are supported by by the Chinese National Key Basic Research Program (973 program) through project 2012CB315905, the Natural Science Foundation of China through projects 61370190, 61173154, 61472429, 61402029, 61272501, 61202465, 61321064 and 61003214, the Beijing Natural Science Foundation through project 4132056, the Fundamental Research Funds for the Central Universities, and the Research Funds (No. 14XNLF02) of Renmin University of China and the Open Research Fund of Beijing Key Laboratory of Trusted Computing, the European Union through projects FP7 "DwB", FP7 "Inter-Trust" and H2020 "CLARUS", the Spanish Government through projects TSI-020302-2010-153 and TIN2011-27076-C03-01, the Catalan Government under grant 2014 SGR 537, the Templeton World Charity Foundation under grant no. TWCF0095, the Shanghai NSF under grant 12ZR1443500; the Shanghai Chen Guang Program (12CG24); the Science and Technology Commission of Shanghai Municipality under grant 13JC1403500. The fourth author is partially supported as an ICREA-Acadèmia researcher by the Catalan Government and by a Google Faculty Research Award. The URV 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] A. Fiat and M. Naor, "Broadcast Encryption," in *Proc. Crypto 1993*, 1993, vol. LNCS 773, Lecture Notes in Computer Science, pp. 480-491.

[2] I. Ingemarsson, D.T. Tang and C.K. Wong, "A Conference Key Distribution System," *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 714-720, 1982.

[3] Q. Wu, Y. Mu, W. Susilo, B. Qin and J. Domingo-Ferrer, "Asymmetric Group Key Agreement," in *Proc. Eurocrypt 2009*, 2009, vol. LNCS 5479, Lecture Notes in Computer Science, pp. 153-170.

[4] http://en.wikipedia.org/wiki/PRISM_%28surveillance_program%29, 2014.

[5] Q. Wu, B. Qin, L. Zhang, J. Domingo-Ferrer and O. Farràs, "Bridging Broadcast Encryption and Group Key Agreement," in *Proc. Asiacrypt 2011*, 2011, vol. LNCS 7073, Lecture Notes in Computer Science, pp. 143-160.

[6] D. H. Phan, D. Pointcheval and M. Strefler, "Decentralized Dynamic Broadcast Encryption," in *Proc. SCN 2012*, 2011, vol. LNCS 7485, Lecture Notes in Computer Science, pp. 166-183

[7] M. Steiner, G. Tsudik and M. Waidner, "Key Agreement in Dynamic Peer Groups," *IEEE Transactions on Parallel and Distributed Systems*, vol. 11, no. 8, pp. 769-780, 2000.

[8] A. Sherman and D. McGrew, "Key Establishment in Large Dynamic Groups Using One-way Function Trees," *IEEE Transactions on Software Engineering*, vol. 29, no. 5, pp. 444-458, 2003.

[9] Y. Kim, A. Perrig and G. Tsudik, "Tree-Based Group Key Agreement," *ACM Transactions on Information System Security*, vol. 7, no. 1, pp. 60-96, 2004.

[10] Y. Mao, Y. Sun, M. Wu and K.J.R. Liu, "JET: Dynamic Join-Exit-Tree Amortization and Scheduling for Contributory Key Management," *IEEE/ACM Transactions on Networking*, vol. 14, no. 5, pp. 1128-1140, 2006.

[11] C. Boyd and J.M. González-Nieto, "Round-Optimal Contributory Conference Key Agreement," in *Proc. PKC 2003*, 2003, vol. LNCS 2567, Lecture Notes in Computer Science, pp. 161-174.

[12] W.-G. Tzeng and Z.-J. Tzeng, "Round Efficient Conference Key Agreement Protocols with Provable Security," in *Proc. Asiacrypt 2000*, 2000, vol. LNCS 1976, Lecture Notes in Computer Science, pp. 614-627.

[13] R. Dutta and R. Barua, "Provably Secure Constant Round Contributory Group Key Agreement in Dynamic Setting," *IEEE Transactions on Information Theory*, vol. 54, no. 5, 2007-2025, 2008.

[14] W.-G. Tzeng, "A Secure Fault-Tolerant Conference-Key Agreement Protocol," *IEEE Transactions on Computers*, vol. 51, no.4, pp. 373-379, 2002.

[15] X. Yi, "Identity-Based Fault-Tolerant Conference Key Agreement," *IEEE Transactions Dependable Secure Computing* vol. 1, no. 3, 170-178, 2004.

[16] M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System," in *Proc. Eurocrypt 1994*, 1994, vol. LNCS 950, Lecture Notes in Computer Science, pp. 275-286.

[17] A. Joux, "A One Round Protocol for Tripartite Diffie-Hellman," *Journal of Cryptology*, vol. 17, no. 4, pp. 263-276, 2004.

[18] D. Boneh and A. Silverberg, "Applications of Multilinear Forms to Crytography," *Contemporary Mathematics*, vol. 324, pp.71-90, 2003.

[19] E. Bresson, O. Chevassut and D. Pointcheval, "Provably Authenticated Group Diffie-Hellman Key Exchange – The Dynamic Case," in *Proc. Asiacrypt 2001*, 2001, vol. LNCS 2248, Lecture Notes in Computer Science, pp. 290-309.

[20] E. Bresson, O. Chevassut and D. Pointcheval, "Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions," in *Proc. Eurocrypt 2002*, 2002, vol. LNCS 2332, Lecture Notes in Computer Science, pp. 321-336.

[21] E. Bresson, O. Chevassut, D. Pointcheval and J.-J. Quisquater, "Provably Authenticated Group Diffie-Hellman Key Exchange," in *Proc. ACM CCS 2001*, 2001, pp. 255-264.

[22] J. Snoeyink, S. Suri and G. Varghese, "A Lower Bound for Multicast Key Distribution," in *Proc. INFOCOM 2001*, 2001, pp. 422-431.

[23] H.J. Kim, S.M. Lee and D. H. Lee, "Constant-Round Authenticated Group Key Exchange for Dynamic Groups," in *Proc. Asiacrypt 2004*, 2004, vol. LNCS 3329, Lecture Notes in Computer Science, pp. 245-259.

[24] M. Abdalla, C. Chevalier, M. Manulis and D. Pointcheval, "Flexible Group Key Exchange with On-demand Computation of Subgroup Keys," in *Proc. Africacrypt 2010*, 2010, vol. LNCS 6055, Lecture Notes in Computer Science, pp. 351-368.

[25] S. Jarecki, J. Kim and G. Tsudik, "Flexible Robust Group Key Agreement," *IEEE Transactions on Parallel Distributed Systetems*, vol. 22, no. 5, pp. 879-886, 2011.

[26] Q. Wu, B. Qin, L. Zhang, J. Domingo-Ferrer and J. Manjón, "Fast Transmission to Remote Cooperative Groups: A New Key Management Paradigm," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 621-633, 2013.

[27] E. Bertino, N. Shang and S.S. Wagstaff Jr., "An Efficient Time-Bound Hierarchical Key Management Scheme for Secure Broadcasting," *IEEE Transactions on Dependable Secure Computing*, vol. 5, no. 2, 65-70, 2008.

[28] A. Shoufan and S.A. Huss, "High-Performance Rekeying Processor Architecture for Group Key Management," *IEEE Transactions on Computers*, vol. 58, no. 10, 1421-1434, 2009.

[29] W. Gu, S. Chellappan, X. Bai and H. Wang, "Scaling Laws of Key Predistribution Protocols in Wireless Sensor Networks," *IEEE Transactions on Information Forensics and Security*, vol. 6, no. 4, 1370-1381, 2011.

[30] M.-H. Park, G.-P. Gwon, S.-W. Seo and H.-Y. Jeong, "RSU-Based Distributed Key Management (RDKM) For Secure Vehicular Multicast Communications," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 644-658, 2011.

[31] Y. Hao, Y. Cheng, C. Zhou and W. Song, "A Distributed Key Management Framework with Cooperative Message Authentication in VANETs," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 616-629, 2011.

[32] Z. Liu, J. Ma, Q. Pei, L. Pang and Y. Park, "Key Infection, Secrecy Transfer and Key Evolution for Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 8, 2643-2653, 2010.

[33] Z. Yu and Y. Guan, "A Key Management Scheme Using Deployment Knowledge for Wireless Sensor Networks," *IEEE Transactions Parallel Distributed Systems*, vol. 19, no. 10, pp. 1411-1425, 2008.

[34] B.-J. Chang and S.-L. Kuo, "Markov Chain Trust Model for Trust-Value Analysis and Key Management in Distributed Multicast MANETs," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 4, pp. 1846-1862, 2009.

[35] B. Rong, H.-H. Chen, Y. Qian, K. Lu, R. Q. Hu and S. Guizani, "A Pyramidal Security Model for Large-Scale Group-Oriented Computing in Mobile Ad Hoc Networks: The Key Management Study," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 1, pp. 398-408, 2009.

[36] M. Naor and B. Pinkas, "Efficient Trace and Revoke Schemes," in *Proc. FC 2000*, 2000, vol. LNCS 1962, Lecture Notes in Computer Science, pp. 1-20.

[37] C.K. Wong, M. Gouda and S. Lam, "Secure Group Communications Using Key Graphs," *IEEE/ACM Transactions on Networking*, vol. 8, no. 1, pp. 16-30, 2000.

[38] D. Wallner, E. Harder and R. Agee, "Key Management for Multicast: Issues and Architectures", *The RFC Repaort 2627*, 1999. Available at: http://www.rfc-editor.org/rfc/pdfrfc/rfc2627.txt.pdf.

[39] M.T. Goodrich, J. Z. Sun and R. Tamassia, "Efficient Tree-Based Revocation in Groups of Low-State Devices," in *Proc. Crypto 2004*, 2004, vol. LNCS 3152, Lecture Notes in Computer Science, pp. 511-527.

[40] J.H. Cheon, N.S. Jho, M.H. Kim and E.S. Yoo, "Skipping, Cascade and Combined Chain Schemes for Broadcast Encryption," *IEEE Transactions Information Theory*, vol. 54, no. 11, pp. 5155-5171, 2008.

[41] L. Harn and C. Lin, "Authenticated Group Key Transfer Protocol Based on Secret Sharing," *IEEE Transactions on Computers*, vol. 59, no. 6, pp. 842-846, 2010.

[42] D. Boneh, C. Gentry and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," in *Proc. Crypto 2005*, 2005, vol. LNCS 3621, Lecture Notes in Computer Science, pp. 258-275.

[43] J.H. Park, H.J. Kim, M.H. Sung and D.H. Lee, "Public Key Broadcast Encryption Schemes With Shorter Transmissions," *IEEE Transactions on Broadcasting*, vol. 54, no. 3, pp. 401-411, 2008.

[44] C. Gentry and B. Waters, "Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts)," in *Proc. Eurocrypt 2009*, 2009, vol. LNCS 5479, Lecture Notes in Computer Science, pp. 171-188.

[45] A. B. Lewko, A. Sahai, B. Waters, "Revocation Systems with Very Small Private Keys," in *Proc. IEEE S&P 2010*, 2010, pp. 273-285.

[46] D. Boneh, C. Gentry, S. Gorbunov, S. Halevi, V. Nikolaenko, G. Segev, V. Vaikuntanathan and D. Vinayagamurthy, "Fully Key-Homomorphic Encryption, Arithmetic Circuit ABE and Compact Garbled Circuits," in *Proc. Eurocrypt 2014*, 2014, vol. LNCS 8441, Lecture Notes in Computer Science, pp. 533-556.

[47] D. Boneh, X. Boyen and E.J. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," in *Proc. Eurocrypt 2005*, 2005, vol. LNCS 3494, Lecture Notes in Computer Science, pp. 440-456.
[48] R. Canetti, S.Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-based Encryption," in: *Proc. EUROCRYPT 2004*, 2004, vol. LNCS 3027, pp. 207-222.
[49] T. Matsuda and G. Hanaoka, "Chosen Ciphertext Security via UCE," in *Proc. PKC 2014*, 2014, vol. LNCS 8383, Lecture Notes in Computer Science, pp. 56-76.
[50] W. Liu, J. Liu, Q. Wu, B. Qin, Y Zhou, "Practical Direct Chosen Ciphertext Secure Key-Policy Attribute-based Encryption with Public Ciphertext Test," in *Proc. ESORICS 2014*, 2014, vol. LNCS 8713, pp. 91-108.
[51] M. Scott, "On the Efficient Implementation of Pairing-Based Protocols," http://eprint.iacr.org/2011/334.pdf, 2011.

Qianhong Wu received his Ph.D. in Cryptography from Xidian University in 2004. Since then, he has been with Wollongong University (Australia) as an associate research fellow, with Wuhan University (China) as an associate professor, with Universitat Rovira i Virgili (Catalonia) as a research director and now with Beihang University (China) as a full professor. His research interests include cryptography, information security and privacy, and *ad hoc* network security. He has been a holder/co-holder of 7 China/Australia/Spain funded projects. He has authored 7 patents and over 100 publications. He has served in the program committee of several international conferences in information security and privacy. He is a member of IACR, ACM and IEEE.

Bo Qin received her Ph.D. degree in Cryptography from Xidian University in 2008 in China. Since then, she has been with Xi'an University of Technology (China) as a lecturer, with Universitat Rovira i Virgili (Catalonia) as a postdoctoral researcher, and now with Renmin University of China as a lecturer. Her research interests include pairing-based cryptography, data security and privacy, and VANET security. She has been a holder/co-holder of 6 China/Spain funded projects. She has authored over 60 publications and served in the program committee of several international conferences in information security.

Lei Zhang received his Ph.D. degree in computer engineering from Universitat Rovira i Virgili, Tarragona, Spain, in 2010. He is an Associate Research Fellow with Software Engineering Institute, East China Normal University, Shanghai, China. Before this, he had been a Postdoctoral Researcher with Universitat Rovira i Virgili. His fields of activity are information security, cryptography, data privacy, and network security. He has been a holder/co-holder of 5 China/Spain funded projects. He has authored over 50 publications. He has served in the program committee of several international conferences in information security and privacy. He is a member of IEEE.

Josep Domingo-Ferrer is a Full Professor of Computer Science and an ICREA-Acadèmia Researcher at Universitat Rovira i Virgili, Tarragona, Catalonia, where he holds the UNESCO Chair in Data Privacy. His research interests are in data privacy and data security. He received his M. Sc. and Ph. D. degrees in Computer Science from the Autonomous University of Barcelona in 1988 and 1991, respectively. He also holds an M. Sc. in Mathematics. He has won several research and technology transfer awards, including the IEEE Fellow Grade, a Google Faculty Research Award, and the Government of Catalonia's "Narcís Monturiol" Medal to the scientific merit and twice the ICREA Acadèmia Prize. He has authored 5 patents and over 340 publications. He has been the co-ordinator of projects funded by the European Union and the Spanish government, among which the CONSOLIDER ARES project on security and privacy, one of Spain's 34 strongest research teams. He has been the PI of US-funded research contracts. He has held visiting appointments at Princeton, Leuven and Rome. He is a co-Editor-in-Chief of *Transactions on Data Privacy*.

Oriol Farràs is a Juan de la Cierva postdoctoral researcher at the UNESCO Chair in Data Privacy and the CRISES Research Group in the Department of Computer Engineering and Maths at Universitat Rovira i Virgili, Tarragona, Catalonia. He received his M.Sc. degree in Mathematics and his M.Sc. degree in Telecommunication Engineering from Universitat Politècnica de Catalunya in 2004 and 2005, respectively. He received his Ph.D. degree in Mathematics from Universitat Politècnica de Catalunya in 2010. He has been a postdoctoral fellow in the Department of Computer Science at Ben Gurion University of the Negev, Israel, and a Director of Research at Universitat Rovira i Virgili. His research interests include cryptography, secret sharing, and information theory.

Jesús A. Manjón is a computer engineer with the UNESCO Chair in Data Privacy and the CRISES Research Group at the Department of Computer Engineering and Maths at Universitat Rovira i Virgili, Tarragona, Catalonia. He got his B.Sc. in Computer Engineering in 2004 and his M.Sc. in Computer Security in 2008. He has participated in several Spanish-funded projects and he is a co-author of several research publications on security and privacy.