Ad Hoc Broadcast Encryption

Qianhong Wu
Dept. of Comp. Eng. and Maths
Universitat Rovira i Virgili
Tarragona, Spain
qianhong.wu@urv.cat

Bo Qin
Dept. of Comp. Eng. and Maths
Universitat Rovira i Virgili
Tarragona, Spain
bo.qin@urv.cat

Lei Zhang
Dept. of Comp. Eng. and Maths
Universitat Rovira i Virgili
Tarragona, Spain

Josep Domingo-Ferrer
Dept. of Comp. Eng. and Maths
Universitat Rovira i Virgili
Tarragona, Spain

ABSTRACT

Existing broadcast encryption (BE) requires a trusted dealer to generate and distribute secret keys to all users. The applications in ad hoc networks, peer-to-peer networks, and on-the-fly data sharing call for confidential broadcast channel without relying on a dealer. To cater for such applications, Wu *et al.* recently introduced the primitive of asymmetric group key agreement and realized a one-round scheme. However, their solution is only suitable for static case in which the group members are assumed to keep unchanged for a long period.

This paper resolves the main open question left in Wu *et al.*'s work by providing rational solutions to fully dynamic case. To meet the end, we first introduce a new notion referred to as ad hoc broadcast encryption (AHBE). In an AHBE system, each user possesses a public key; seeing the public keys of the users, a sender can securely broadcast to any subset of the users; only the user in the receiver set can decrypt. Then we propose a generic transformation from any key homomorphic BE scheme to an AHBE scheme, and implement a concrete AHBE by showing that the recent Gentry-Waters BE scheme is key-homomorphic. This AHBE scheme enjoys non-interactive decryption and sub-linear complexity, comparable to up-to-date broadcast systems which have also sub-linear complexity but require a trusted dealer to initialize the system. Lastly, observing the inherent sub-linear complexity of AHBE converted from regular BE schemes, we propose a direct construction of AHBE which has constant complexity, at a cost of a one-round interaction for decryption. As to security, both schemes are shown to be adaptively secure in the standard model under well-studied assumptions.

ACM CCS 2010, Chicago, USA

Categories and Subject Descriptors

E.3 [**Data Encryption**]: Public Key Cryptosystems

General Terms

Security, Algorithm

Keywords

Broadcast Encryption, Asymmetric Group Key Agreement, Ad Hoc Broadcast

1. INTRODUCTION

Broadcasting is one of the most useful, versatile, and well-studied communication primitive in distributed computing with many applications. Existing broadcast encryption (BE) systems [2, 16, 18] require a trusted dealer to produce and distribute secret keys to each user. Such systems provide efficient solutions to applications such as pay-TV and priced video distribution. However, the existing BE systems are not suitable for applications where the trusted dealer is unavailable. For instance, in *ad hoc*/peer-to-peer networks emerging in recent years, it is difficult to find an entity who can play the role of a trusted dealer to deploy a BE system. As a second example, let us consider the following scenario in the traditional Internet. A company provides remote data storage services to registered users; users wish to be able to share their private files with some other registered users but do not want the company to see the contents of the shared files. In both scenarios, it is needed to build BE systems without requiring a trusted dealer. A trivial solution can be achieved with any public cryptosystems, but suffers from long ciphertexts and heavy encryption overhead. The challenge is to design solutions with short ciphertexts and efficient encryption.

1.1 Related Work

Public key cryptosystems. As mentioned above, any public key cryptosystem, e.g., the known RSA cryptosystem [23] and the ElGamal cryptosystem [15], can be used to implement a trivial AHBE/BE scheme. In such a construction, each user has a public/private key pair; to broadcast a secret message, the sender uses the respective public keys of the users in the receiver set to encrypt the message; the resulting ciphertext is the concatenation of the ciphertexts under

each public key; each receiver can select out her piece of ciphertext and then recover the message by invoking the decryption algorithms of the underlying public key cryptosystems. Unfortunately, this trivial construction suffers from $O(n)$ ciphertexts and $O(n)$ encryption operations, where n is the maximum number of allowable receivers.

Conventional broadcast cryptosystems. Conventional public key broadcast encryption is a cryptographic primitive more related to our AHBE systems. In a broadcast encryption scheme [16], a trusted dealer generates and distributes private keys to n users; a sender can send a message to a dynamically chosen subset of receivers $\mathbb{R} \subseteq \{1, \cdots, n\}$ of users such that only users in $\mathbb{R}$ can decrypt the ciphertext; the sender can then safely transmit this ciphertext over a broadcast channel to all users, assuming that the public key of dealer is authentic. It is desirable if the system is public-key so that anybody can encrypt, allows stateless receivers who do not need to update their private keys, and has collusion resistance in the sense that even if all users outside $\mathbb{R}$ collude, they cannot decrypt. Similarly to our AHBE systems, the main challenge in building broadcast cryptosystems is to encrypt messages with short ciphertexts and efficient encryption.

The earlier broadcast encryption systems have relied on combinatorial techniques. Such systems include a collusion bound t. if an adversary compromises more than t keys, the system would no longer guarantee security even for encryptions solely sent to uncompromised users. Fiat and Naor [16] were the first to formally explore broadcast encryption. Further improvements [20, 21] reduce the private key size. Dodis and Fazio [14] extend the subtree difference method into a public key broadcast system for a small size public key. Wallner *et al.* [25] and Wong [26] independently discovered the logical-tree-hierarchy scheme for group multicast. The parameters of the original schemes are improved in further work [12, 13, 24].

Recently, more efficient broadcast systems have been realized from bilinear pairings. Boneh *et al.* [2] proposed two efficient broadcast encryption schemes proven to be secure. Their basic scheme has linear public keys but constant secret keys and ciphertexts. After a tradeoff, they achieve a scheme with $O(\sqrt{n})$ public keys, private keys, and ciphertexts. However, they use a static model of security in which an adversary declares the target set $\mathbb{R}^*$ of his challenge ciphertext before even seeing the system parameters. The subsequent efforts [18, 4, 5] have been devoted to improve security, but the sub-linear barrier $O(\sqrt{n})$ has been not broken in existing conventional BE schemes.

These conventional BE schemes are not suitable for the applications motivated at the beginning of this paper. In existing broadcast schemes, a trusted dealer is required to generate the system parameters, the public key. The dealer is also required to produce the secret decryption keys and distribute them to each user, which implies that (i) confidential channels from the dealer to each user have to be established before distributing decryption keys of the broadcast scheme, and (ii) the privileged dealer knows all the decryption keys and must be fully trusted by all users. These features are not always desirable in practice. If such a trusted party does not exist or some users do not trust the dealer, e.g., in *ad hoc* networks, then the existing schemes cannot work.

Group key exchanges. Another related area to our work is group key exchanges (GKE). A number of GKE protocols have been proposed [6, 7, 8, 9, 10, 11]. In such systems, via open networks, a group of users can negotiate a common secret key shared among the group members; any group member can broadcast messages encrypted with the shared key to other members such that only the group members can decrypt. Note that the up-to-date GKE protocols requires at least two rounds and the negotiated secret key can only be shared among group members. GKE protocols do not provide efficient solutions to our motivated applications, because whenever a sender wants to send a message to a group of receivers, she has to firstly join the group of receivers and run a GKE protocol to obtain a secret encryption key. Since the sender can be potentially anyone even if the receivers keep unchanged, the solution is not efficient. To address the above limitations, asymmetric group key agreement [27] has been proposed in which a public encryption key is negotiated. However, the protocol can only deal with static case where the group of receivers cannot be chosen by the sender. Indeed, it is left open in that work [27] to construct a protocol allowing the sender to dynamically broadcast to any subset of potential receivers. We solve this problem with efficient AHBE proposals.

1.2 Our Contribution

In this paper we consider the problem of dynamic broadcasting to ad hoc groups where a trusted dealer is unavailable. A new primitive referred to as *ad hoc* broadcast encryption (AHBE) is proposed to cater for such applications. In an AHBE system, each user has a public/private key pair; knowing the public keys of the users, a sender can choose any subset of the users to broadcast, provided that the number of the receivers is less than n; only the users in the receiver set can decrypt. We define an adaptive security notion in AHBE systems where the attacker adaptively corrupts users before choosing the receiver set to attack. It is easy to see that any regular public key encryption system implies an *ad hoc* broadcast encryption system in which, for n receivers, $O(n)$ encryption operations and $O(n)$ ciphertexts are required. *The challenge is to design AHBE systems with short ciphertexts and efficient encryption.*

This paper focuses on addressing the above challenge. We present the notion of key homomorphic broadcast encryption (KHBE) in which the public keys and decryption keys of different KHBE instances have a key homomorphic relationship. The decryption keys of different KHBE instances can be aggregated as the according decryption keys corresponding to the public key from aggregation of the public keys of the underlying KHBE instances. By exploiting KHBE, we propose a generic construction of AHBE and instantiate a concrete AHBE scheme, by showing that the recent Gentry-Waters BE scheme [18] is key-homomorphic. The proposed scheme is shown to be adaptively secure in the standard model under the decision bilinear Diffie-Hellman exponentiation (BDHE) assumption [18, 27]. Our basic construction has $O(n)$ size private keys and $O(n^2)$ size public keys but constant size ciphertexts. To decrypt, each receiver does not need the help from other receivers and the decryption is non-interactive, provided that the public keys of other receivers are known. Observing that a system with $O(n^2)$ public keys cannot be deployed for a practical scale of AHBE in practice, we provide a tradeoff between ciphertexts and public keys. The resulting AHBE enjoys sub-linear complexity $O(n^{2/3})$ regarding both public keys and ciphertexts and

$O(n^{1/3})$ private keys. Our result is comparable to up-to-date regular broadcast systems, in which each receiver has also sub-linear overhead (i.e., $O(\sqrt{n})$) regarding public/private keys and/or ciphertexts but require a fully trusted dealer.

1.3 Paper Organization

Section 2 reviews background information pertaining to our constructions. In Section 3, we formalize the definition of AHBE systems. Section 4 proposes an AHBE scheme with sub-linear complexity and non-interactive decryption. An AHBE scheme with constant complexity is instantiated in Section **??**. Section 5 is a conclusion.

2. PRELIMINARY

2.1 Notations

We summarize some notations used throughout the paper. N is the number of all potential users, and n is the maximum number of receivers in a broadcast system. In regular BE, it usually happens that $N = n$. In AHBE, N might be very large up to millions but n is usually very small. $\mathbb{R}$ is the receiver set in a broadcast. $a \leftarrow A$ represents to sample a random point from a space A and assign its value to a. If A is a probabilistic algorithm, it means that a is the output of an independent run of A. $|A|$ denotes the cardinality of set A. PPT represents probabilistic polynomial time. Hdr is said to be a header of k under $\langle \mathbb{R}, PK \rangle$ in a broadcast if Hdr is the ciphertext of k encrypted with the public key PK and only the receivers in $\mathbb{R}$ can extract k.

2.2 Bilinear Maps

We briefly review a few facts related to groups with efficiently computable bilinear maps [3, 17]. Let `PairGen` be a PPT 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 a bilinear map. The bilinear map e satisfies the following properties.

1. Non-degeneracy. $e(g, g) \neq 1$ for any generator g of $\mathbb{G}$.
2. Bilinearity. For all $u, v \in \mathbb{Z}$, it holds that $e(g^u, g^v) = e(g, g)^{uv}$.

We say that $\mathbb{G}$ is a bilinear map if the group operations in $\mathbb{G}$ and the bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ are both efficiently computable. Note that the map e is symmetric since $e(g^u, g^v) = e(g, g)^{uv} = e(g^v, g^u)$.

2.3 Complexity Assumptions

We recall the Decision Bilinear Diffie-Hellman (DBDH) assumption [1] and the decision n-Bilinear Diffie-Hellman Exponentiation (BDHE) assumption [18, 27] in the following definitions.

Definition 1. **(DBDH Assumption.)** Let $\mathbb{G}$ be bilinear group of prime order p as defined above and g be a generator of $\mathbb{G}$. In the DBDH game, an attacker $\mathcal{A}(\cdot)$ is given $g, g^x, g^y, g^z, e(g, g)^{bxyz+(1-b)\delta}$, where x, y, z, δ are randomly chosen from $\mathbb{Z}_p^*$ and b is randomly chosen from $\{0, 1\}$; $\mathcal{A}$ is required to output a bit b' and wins if $b' = b$. The DBDH assumption states that for any polynomial time attacker $\mathcal{A}$, her advantage $Adv_{\mathcal{A}} = |\Pr(\mathcal{A} \text{ wins}) - \frac{1}{2}|$ is negligible in λ.

Definition 2. **(Decision n-BDHE Assumption.)** Let $\mathbb{G}$ be bilinear group of prime order p as defined above and g, h two independent generators of $\mathbb{G}$. 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 $\alpha \in \mathbb{Z}_p^*$. Randomly select $b \leftarrow \{0, 1\}$. If $b = 0$, set $Z = e(g_{n+1}, h)$; else if $b = 1$, randomly choose $Z \leftarrow \mathbb{G}_T$. An attacker $\mathcal{A}$ is provided with $(g, h, \overrightarrow{y}_{g,\alpha,n}, Z)$ and required to output a bit b'. The attacker wins if $b' = b$. The decision n-BDHE assumption states that for any polynomial time attacker $\mathcal{A}$, her advantage $Adv_{\mathcal{A}} = |\Pr(\mathcal{A} \text{ wins}) - \frac{1}{2}|$ is negligible in λ.

2.4 BE Systems

In a conventional BE system, a dealer first setups the system parameters including a public/secret key pair. All these parameters are publicly accessible except that the system secret key is kept confidential by the dealer. Then the dealer generate a decryption key for each possible user and distribute the decryption key to each subscriber. Finally, any sender who knows the system public key can broadcast confidential messages to any subset of the subscribers. Only the subscribers in the subset chosen by the sender can decrypt. More formally, a BE system consists of the following probabilistic (interactive) algorithms [18].

BSetup(n, N) Takes as input the number of receivers N and the maximal size n of a broadcast recipient group. It outputs a BE public/secret key pair $\langle PK, SK \rangle$. Here, for simplicity, we leave another input, the input security parameter λ, implicitly.

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

BEnc$(\mathbb{R}, PK)$ It takes as input a recipient set $\mathbb{R} \subseteq \{1, \cdots, N\}$ and the public key PK. If $|\mathbb{R}| \leq n$, it outputs a pair $\langle Hdr, k \rangle$ where Hdr is called the header and $k \in \mathbb{K}$ is the message encryption key.

BDec$(\mathbb{R}, i, d_i, Hdr, PK)$ This algorithm allows each receiver to decrypt the message encryption key k hidden in the header. It takes as input the receiver set $\mathbb{R}$, an index $i \in \{1, \cdots, N\}$, the receiver's secret key d_i, a header Hdr, the public key PK. If $|\mathbb{R}| \leq n, i \in \mathbb{R}$, then the algorithm outputs the message encryption key k.

A BE system is correct if for all $\mathbb{R} \subseteq \{1, \cdots, N\}$ and all $i \in \mathbb{R}$, if $\langle PK, SK \rangle \leftarrow \mathsf{BSetup}(n, N)$, $d_i \leftarrow \mathsf{BKeyGen}(i, SK)$, and $\langle Hdr, k \rangle \leftarrow \mathsf{BEnc}(\mathbb{R}, PK)$, then $\mathsf{BDec}(\mathbb{R}, i, d_i, Hdr, PK) = k$.

We define security in broadcast encryption by using the following game between an attacker $\mathcal{A}$ and a challenger $\mathcal{CH}$.

BSetup. The attacker commits to a set $\tilde{\mathbb{R}} \subseteq \{1, \cdots, N\}$. The challenger runs $\mathsf{BSetup}(n, N)$ to obtain the the public key PK, which is given to the attacker.

BCorruption. Attacker $\mathcal{A}$ adaptively issues private key queries for some indices $i \in \{1, \cdots, N\} \setminus \tilde{\mathbb{R}}$.

BChallenge. At some point, the attacker then specifies a challenge set $\mathbb{R}^* \subseteq \tilde{\mathbb{R}}$. Note that for the private key of any user i queried in $\mathsf{BCorruption}$ we have that $i \notin \mathbb{R}^*$. The challenger sets $\langle Hdr^*, k_0 \rangle \leftarrow \mathsf{BEnc}(\mathbb{R}^*, PK)$ and $k_1 \leftarrow \mathbb{K}$. It sets $b \leftarrow \{0, 1\}$ and gives (Hdr^*, k_b) to attacker $\mathcal{A}$.

BGuess. Attacker $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$ for b and wins the game if $b = b'$.

We define $\mathcal{A}$'s advantage in attacking the BE system with system parameters (n, N) and security parameter λ as

$$Adv^{\mathsf{BE}}_{\mathcal{A},n,N}(1^\lambda) = |\Pr[b = b'] - \frac{1}{2}|$$

Definition 3. We say that a BE system has semi-static adaptive security if for all polynomial time algorithms $\mathcal{A}$ we have that $\mathrm{Adv}^{\mathsf{BE}}_{\mathcal{A},n,N}(1^\lambda)$ is negligible in λ, and a BE system has adaptive security if $\mathcal{A}$ does not need to commit to $\tilde{\mathbb{R}}$ in the BSetup stage, and in the $\mathsf{BChallenge}$ state, any index $i \in \mathbb{R}^*$ has not been queried in the $\mathsf{BCorruption}$ stage.

3. MODELING AHBE

We present the model of AHBE systems and then formalize the security definitions.

3.1 AHBE Systems

Compared to regular BE, AHBE eliminates the requirement of a fully trusted dealer. In AHBE, each user publish a public key so that any potential sender and other possible users can access. Then a sender can encrypt any confidential message to any chosen group of users, provided that the group size is less than n. Finally, only users in the group chosen by the sender can decrypt. The challenge is to guarantee that the encryption is efficient and the ciphertext is short. AHBE is also conceptually simple, compared to group key exchanges which usually require multiple rounds and a sender has to first join the group to which she wants to broadcast a message.

For clarity, we define AHBE as a key encapsulation mechanism. An AHBE system consists of the following probabilistic (interactive) algorithms:

KeyGen(i, n, N) Let N be the number of potential receivers, and $n \leq N$ be the maximal size of an *ad hoc* broadcast recipient group. This key generation algorithm is run by each user $i \in \{1, \cdots, N\}$ to generate her public/private key pair. A user takes as input the system parameters n, N and her index $i \in \{1, \cdots, N\}$, and outputs $\langle pk_i, sk_i\rangle$ as her public/secret key pair. Denote $\{\langle pk_i, sk_i\rangle | i \in \mathbb{R} \subseteq \{1, \cdots, N\}\}$ by $\langle pk_i, sk_i\rangle_{\mathbb{R}}$ and similarly, $\{\langle pk_i\rangle | i \in \mathbb{R} \subseteq \{1, \cdots, N\}\}$ by $\langle pk_i\rangle_{\mathbb{R}}$. Here, similarly to the BE definition, we also leave the input security parameter λ, implicitly.

AHBEnc$(\mathbb{R}, \langle pk_i\rangle_{\mathbb{R}})$ This is the *ad hoc* broadcast encryption algorithm. It is run by any sender who may or may not be in $\{1, \cdots, N\}$, provided that the sender knows the public keys of the potential receivers. It takes as input a recipient set $\mathbb{R} \subseteq \{1, \cdots, N\}$ and the public key pk_i for $i \in \mathbb{R}$. If $|\mathbb{R}| \leq n$, it outputs a pair $\langle Hdr, k\rangle$ where Hdr is called the header and k is the message encryption key.

Let $\mathcal{E}_{sym}$ be a symmetric encryption scheme with keyspace $\mathbb{K}$, and $E(\cdot)$ and $D(\cdot)$ be the encryption and decryption algorithms, respectively. Let M be a message to be broadcasted to the set $\mathbb{R}$, and let $C = E(M, k)$ be the encryption of M under the symmetric key $k \in \mathbb{K}$. The broadcast to users in $\mathbb{R}$ consists of $(\mathbb{R}, Hdr, C)$. The definition of the *ad hoc* broadcast encryption procedure implies that this procedure is only used to generate the message encryption key, but it is non-interactive in the sense that, the sender can broadcast to an *ad hoc* group once she knows the receivers' public keys.

AHBDec$(\mathbb{R}, j, sk_j, Hdr, \langle pk_i\rangle_{\mathbb{R}})$ This algorithm allows each receiver to decrypt the message encryption key k hidden in the header. It may be a non-interactive (or an interactive) algorithm if each receiver does not need (or needs) other receivers' help to extract her decryption key. It takes as input the receiver set $\mathbb{R}$, an index $j \in \{1, \cdots, N\}$, the receiver's secret key sk_j, a header Hdr, the public/private key pairs of receivers in the recipient set $\mathbb{R}$. If $|\mathbb{R}| \leq n$, $j \in \mathbb{R}$, then the algorithm outputs the message encryption key k. The key k can then be used to decrypt C to obtain M by computing $M = D(C, k)$.

3.2 Security Definitions

For purpose of focusing on the functionality of confidentiality, we implicitly assume that the public keys of users are authentic without distractions to authenticate the public key of each user. In traditional Internet, the authentication can be achieved with certificates by employing existing public key infrastructure. In newly emerging mobile ad hoc networks, it might rely on a web of trust because the users may meet in person.

As usual, we first define the correctness of an AHBE scheme. It states that any user in the receiver set can decrypt a valid header. Formally, it is defined as follows.

Definition 4. **(Correctness.)** Assume the same setting as the previous section. An AHBE scheme is correct if for $\{\langle pk_i, sk_i\rangle\} \leftarrow \mathsf{KeyGen}(i, n, N)$, all $\mathbb{R} \subseteq \{1, \cdots, N\}(|\mathbb{R}| \leq n)$ and all $i \in \mathbb{R}$, $\langle Hdr, k\rangle \leftarrow \mathsf{AHBEnc}(\mathbb{R}, \langle pk_i\rangle_{\mathbb{R}})$, then it holds that $\mathsf{AHBDec}(\mathbb{R}, j, sk_j, Hdr, \langle pk_i\rangle_{\mathbb{R}}) = k$ for any $j \in \mathbb{R}$.

We only define security against chosen plaintext attacks. However, our definition can readily be extended to capture chosen ciphertext attacks.

In an *adaptively* secure *ad hoc* broadcast encryption system, the adversary is allowed to see the public keys of all the receivers and then ask for several secret keys before choosing the set of indices that it wishes to attack.

Adaptive security in *ad hoc* broadcast encryption is defined using the following game between an attacker $\mathcal{A}$ and a challenger $\mathcal{CH}$. Both $\mathcal{CH}$ and $\mathcal{A}$ are given λ as input.

Setup. The challenger runs $\mathsf{KeyGen}(i, n, N)$ to obtain the users' public keys. The challenger gives the public keys and public system parameters to the attacker.

Corruption. Attacker $\mathcal{A}$ adaptively issues private key queries for some indices $i \in \{1, \cdots, N\}$.

Challenge. At some point, the attacker specifies a challenge set $\mathbb{R}^*$, such that for the private key of any user i queried in the corruption step we have that $i \notin \mathbb{R}^*$. The challenger sets $\langle Hdr^*, k_0\rangle \leftarrow \mathsf{AHBEnc}(\mathbb{R}^*, \langle pk_i\rangle_{\mathbb{R}^*})$ and $k_1 \leftarrow \mathbb{K}$. It sets $b \leftarrow \{0,1\}$ and gives (Hdr^*, k_b) to attacker $\mathcal{A}$.

Guess. Attacker $\mathcal{A}$ outputs a guess bit $b' \in \{0,1\}$ for b and wins the game if $b = b'$.

We define $\mathcal{A}$'s advantage in attacking the *ad hoc* broadcast encryption (AHBE) system with security parameter λ as

$$Adv^{\mathtt{AHBE}}_{\mathcal{A},n,N}(1^\lambda) = |\Pr[b = b'] - \frac{1}{2}|.$$

Definition 5. **(Adaptive security.)** We say that an AHBE scheme is adaptively secure if for all polynomial time algorithms $\mathcal{A}$ we have that $\mathrm{Adv}^{\mathtt{AHBE}}_{\mathcal{A},n,N}(1^\lambda)$ is negligible in λ.

Similarly to BE [18], in addition to the adaptive game for AHBE security, we consider two other weaker security notions. The first is static security, where the adversary must commit to the set $\mathbb{R}^*$ of identities that it will attack in an Initialization phase before the Setup algorithm is run. This is the security definition that is used by recent BE systems [2, 18]. Another useful security definition is referred to as *semi-static* security. In this game the adversary must commit to a set $\tilde{\mathbb{R}}$ of indices at the Initialization phase. The adversary cannot query the private key for any $i \in \tilde{\mathbb{R}}$, and it must choose a target group $\tilde{\mathbb{R}}^*$ for the challenge ciphertext that is a subset of $\tilde{\mathbb{R}}$. A semi-static adversary is weaker than an adaptive adversary, but it is stronger than a static adversary since the attacker's choice of which subset of $\tilde{\mathbb{R}}$ to attack can be adaptive.

3.3 From Semi-static Security to Adaptive Security

Static security seems too weak to capture the attackers against AHBE systems, since the challenger does not know in practice the target set that the attacker will corrupt. Adaptive security might be a correct definition for security in AHBE systems. However, it seems hard to achieve adaptive security in AHBE systems since the simulator does not know which users the attacker will corrupt so that it can prepare secret keys for them. A usual way is to let the simulator guess the target set before initializing the adaptive security game. Nevertheless, such a reduction suffers from an exponentially small probability of correctly guessing the target set.

Recently, Gentry and Waters developed a modular proof approach [18] to achieve adaptive security in regular BE systems. Their technique is derived from the two-key simulation technique introduced by Katz and Wang [22] which was initially used to obtain tightly secure signature and identity-based encryption schemes in the random oracle model. They further exploit this idea to achieve adaptively secure regular broadcast encryption systems from semi-statically secure ones. We observe that this idea can also be employed to convert any semi-static secure AHBE system into an adaptively secure system.

In the sequel we show how to convert an AHBE system with semi-static security into one with adaptive security. The cost is doubling public keys and ciphertexts. Suppose we are given a semi-static secure AHBE system $\mathtt{AHBE}_{SS}$ with algorithms $KeyGen_{SS}$, $AHBEnc_{SS}$, $AHBDec_{SS}$. Then we can build an adaptively secure $\mathtt{AHBE}_A$ system as follows.

- KeyGen. A user generates his public/secret key pair as the following:

  $s_i \leftarrow \{0,1\}$.

  $(pk'_{2i-1}, sk'_{2i-1}) \leftarrow KeyGen_{SS}(2i-1, 2n, N)$,

  $(pk'_{2i}, sk'_{2i}) \leftarrow KeyGen_{SS}(2i, 2n, N)$.

  Set $pk_i = (pk'_{2i-1}, pk'_{2i})$, $sk_i = (sk'_{2i-s_i}, s_i)$.

  Output (pk_i, sk_i).

- AHBEnc. For any message M to be broadcasted, the sender does the following:

  Generate a random set of $|\mathbb{R}|$ bits: $t \leftarrow \{t_i \leftarrow \{0,1\} : i \in \mathbb{R}\}$.

  Set

  $\mathbb{R}_0 = \{2i - t_i : i \in \mathbb{R}\}$,

  $\langle Hdr_0, k_0\rangle = AHBEnc_{SS}(\mathbb{R}_0, \langle pk'_\ell\rangle_{\mathbb{R}_0})$,

  $\mathbb{R}_1 = \{2i - (1 - t_i) : i \in \mathbb{R}\}$,

  $\langle Hdr_1, k_1\rangle = ABHEnc_{SS}(\mathbb{R}_1, \langle pk'_\ell\rangle_{\mathbb{R}_1})$,

  $C_0 = E(M, k_0)$,

  $C_1 = E(M, k_1)$,

  $Hdr = \langle Hdr_0, C_0, Hdr_1, C_1, t\rangle$.

  Output $\langle Hdr, K\rangle$.

- AHBDec. Receiving $\langle Hdr, K\rangle$, a user in $\mathbb{R}$ does the following:

  Parse sk_j as $\langle sk'_j, s_j\rangle$,

  Parse Hdr as $\langle Hdr_0, C_0, Hdr_1, C_1, t\rangle$.

  Set $\mathbb{R}_0$ and $\mathbb{R}_1$ as above.

  Compute

  $k_{s_j \oplus t_j} \leftarrow AHBDec_{SS}(\mathbb{R}_{s_j \oplus t_j}, j, sk'_j, Hdr_{s_j \oplus t_j}, \langle pk'_\ell\rangle_{\mathbb{R}_{s_j \oplus t_j}})$,

  $M = D(C_{s_j \oplus t_j}, k_{s_j \oplus t_j})$.

  Output M.

We briefly compare the Gentry-Waters conversion for regular BE systems with ours for AHBE systems. In the Gentry-Water conversion for regular BE systems, each user is associated with two potential secret keys; however, the dealer gives her only one of the two. An encryptor (who does not know which secret key the receiver possesses) encrypts the ciphertext twice, one for each key. The main benefit of this idea is that a simulator will have private keys for every user, and then it can always correctly answer the corruption queries from the attacker, hence circumventing the need of guessing of the target set in advance.

In our conversion, since no dealer will generate private keys for the users, each user generates by herself two public/private key pairs; only one of the two secret keys will be kept while the other is erased. The combination of each user's public keys can work as the public key of the regular BE system. Then the encryptor and the users can do the same as that in the Gentry-Water conversion. It is easy to see that, from a viewpoint of security proof, the two conversions are identical. This is due to the fact that, in the security proof of a regular BE system, a simulator will generate all the system parameters and the public/private keys on behalf of the dealer. In the context of AHBE systems, the simulator will do the same job on behalf of each user. In both cases, the attacker only communicates with the simulator. Thus, there is no difference for the attacker to communicate with the simulator in a regular BE or an AHBE system. Hence, the proof of the Gentry-Water conversion applies to the following theorem.

THEOREM 1. *Let $\mathcal{A}$ be an adaptive attacker against* $\mathtt{AHBE}_A$. *Then, there exist algorithms $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$, and $\mathcal{B}_4$, each running in about the same time as $\mathcal{A}$, such that*

$$Adv^{\mathtt{AHBE}_A}_{\mathcal{A},n,N}(\lambda) \leq Adv^{\mathtt{AHBE}_{SS}}_{\mathcal{B}_1,2n,N}(\lambda) + Adv^{\mathtt{AHBE}_A}_{\mathcal{B}_2,2n,N}(\lambda) + Adv^{\mathcal{E}_{sym}}_{\mathcal{B}_3}(\lambda) + Adv^{\mathcal{E}_{sym}}_{\mathcal{B}_4}(\lambda)$$

PROOF. It is omitted to avoid repetition. □

4. AHBE WITH SHORT CIPHERTEXTS

In this section, we propose a generic construction of AHBE schemes. The construction is based on a new notion of key homomorphic broadcast encryption. Then we describe a concrete implementation by showing that the Gentry-Waters BE scheme [18] is key homomorphic. The decryption in our scheme is non-interactive. The basic construction has $O(n^2)$ size public keys but constant size ciphertexts. After a tradeoff between ciphertexts and public keys we obtain a scheme with sub-linear ciphertexts, public keys and private keys.

4.1 Key Homomorphism

Coarsely speaking, the key homomorphism of a BE scheme means that, given two instances of the BE scheme, both their public keys PK_1, PK_2 and decryption keys $d_1(i), d_2(i)$ can be aggregated such that the aggregation of $d_1(i)$ and $d_2(i)$ is a decryption key corresponding to the aggregation of PK_1 and PK_2. Formally, the key homomorphism is defined as follows.

Definition 6. **(Key homomorphism.)** Let $\otimes : \Gamma \times \Gamma \to \Gamma$ and $\odot : \Omega \times \Omega \to \Omega$, $\bigcirc : \mathbb{K} \times \mathbb{K} \to \mathbb{K}$ be appropriately defined efficient operations in the public key space Γ, the decryption key space Ω and the message encryption key space $\mathbb{K}$, respectively. A BE scheme is said to be key homomorphic if the following conditions hold for all $\mathbb{R} \subseteq \{1, \cdots, N\} (|\mathbb{R}| \leq n)$ and all $i \in \mathbb{R}$:

1. If $\langle PK_1, SK_1 \rangle \leftarrow \mathsf{BSetup}(n, N)$,

   $\langle PK_2, SK_2 \rangle \leftarrow \mathsf{BSetup}(n, N)$,

   $d_1(i) \leftarrow \mathsf{BKeyGen}(i, SK_1)$,

   $d_2(i) \leftarrow \mathsf{BKeyGen}(i, SK_2)$,

   $\langle Hdr, k \rangle \leftarrow \mathsf{BEnc}(\mathbb{R}, PK_1 \otimes PK_2)$,

   then $\mathsf{BDec}(\mathbb{R}, i, d_1(i) \odot d_2(i), Hdr, PK_1 \otimes PK_2) = k$.

2. If Hdr is a header of k_1 under $\langle \mathbb{R}, PK_1 \rangle$, then it is header of some k_2 under $(\mathbb{R}, PK_2)$ and a header of $k_1 \bigcirc k_2$ under $\langle \mathbb{R}, PK_1 \otimes PK_2 \rangle$.

Hereafter, if a BE scheme is key homomorphic, we will call it a key homomorphic BE (KHBE) scheme.

4.2 Transformation from KHBE to AHBE

The main idea is to exploit the key homomorphism of the underlying BE scheme, and illustrated in Matrix (1), where ? means that $d_i(i) (i = 1, \cdots, n)$ is not published.

$$\begin{pmatrix} \mathcal{U}_1 & \mathcal{U}_2 & \mathcal{U}_3 & \cdots & \mathcal{U}_n & \mathtt{Sender} \\ ? & d_1(2) & d_1(3) & \cdots & d_1(n) & PK_1 \\ d_2(1) & ? & d_2(3) & \cdots & d_2(n) & PK_2 \\ d_3(1) & d_3(2) & ? & \cdots & d_3(n) & PK_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ d_n(1) & d_n(2) & d_n(3) & \cdots & ? & PK_n \end{pmatrix} \quad (1)$$

We briefly explain the above matrix. PK_i is the public key of a BE instance generated by user i. $d_i(j)$ is the decryption key allocated to user j in the underlying BE instance. Each row is published by a correspondent member of an ad hoc group of broadcast receivers, but user $\mathcal{U}_i$ does not publish $d_i(i)$. Due to the key homomorphism, for any receiver set $\mathbb{R}$, PK_i can be publicly aggregated into $K = \bigotimes_{i \in \mathbb{R}}$ as the public key of a new instance of the underlying BE, and the j-column $\{d_i(j)\}_{i=1}^n$ can be aggregated into a decryption key $d(j) = \bigodot_{i \in \mathbb{R}} d_i(j)$ corresponding to the public key K of the new BE instance. Since $d_j(j)$ is not published, $d(j)$ can only be obtained by user $j (j = 1, \cdots, n)$. Finally, a sender can choose any receiver set $R \subseteq \{1, \cdots, n\}$ to broadcast and only user $j \in \mathbb{R}$ can decrypt with her decryption key $d(j)$. For any user $j' \notin \mathbb{R}$, she cannot use $d(j') = \bigodot_{i \in \mathbb{R}} d_i(j')$ to decrypt correctly. This is because K fully functions as the public key of a regular BE systems in which all users have decryption keys but only the intended receivers can decrypt.

Here, we implicitly require that PK_i's are (computationally) independent and different. Else, the conversion is not secure. For instance, if $PK_1 = PK_2$, then $d_1(1)$ can be trivially computed from the published data as $d_1(1) = d_2(1)$ and the latter $d_2(1)$ is public. This requirement is rational in practice because PK_i's are independently generated by different users and the coincidence of equality is negligible. The detailed conversion is described as follows.

- **KeyGen.** Assume that the potential receivers form a set $\{1, \cdots, N\}$. Let $n \leq N$ be the maximum number of receivers in a broadcast. For simplicity and without loss of generality, we hereafter assume that $n = N$. Generate an instance π of a KHBE scheme as system parameter. Then the KeyGen algorithm works as follows.
  - For receiver $i \in \{1, \cdots, n\}$, invoke BSetup to generate a public-private key pair (PK_i, SK_i) of the underlying KHBE scheme.
  - Receiver i runs $\mathsf{BKeyGen}$ and obtains $d_i(j) \leftarrow \mathsf{BKeyGen}(j, SK_i)$ for $j = 1, \cdots, n$. The public key of the receiver i in the AHBE scheme is set as
    $$K_i = \{d_i(j) | 1 \leq i \neq j \leq n\} \cup \{PK_i\}.$$
    Note that $d_i(i)$ is not published.
  - Set receiver i's private key as $d_i(i)$ which will be used for later decryption regarding the AHBE scheme.
- **AHBEnc.** This procedure works as follows.

– Decide the receiver set $\mathbb{R} \subseteq \{1, \cdots, n\}$.

– Extract the broadcast public key:

$$K = \bigotimes_{i \in \mathbb{R}} PK_i.$$

Since PK_i's are public, any sender can retrieve them and compute the group public key K for broadcast.

– Invoke the underlying KHBE encryption algorithm $\mathsf{BEnc}(\cdot)$ to compute the header

$$\langle Hdr, k \rangle \leftarrow \mathsf{BEnc}(\mathbb{R}, K).$$

Send $(\mathbb{R}, Hdr)$ to the receivers.

- **AHBDec** Due to the key homomorphism of the underlying KHBE scheme, the receiver $i \in \mathbb{R}$ can extract a decryption key under the AHBE public key K by computing

$$d(i) = d_j(i) \bigodot [\bigodot_{j \in \mathbb{R}}^{j \neq i} d_j(i)] = \bigodot_{j \in \mathbb{R}} d_j(i).$$

Note that $d_i(i)$ is not published and only the receiver i in the receiver set $\mathbb{R}$ can compute it. It is easy to see that $d(i) = \bigodot_{j \in \mathbb{R}} d_j(i)$ is a valid decryption key under the aggregated public key $K = \bigotimes_{i \in \mathbb{R}} PK_i$ of the underlying KHBE scheme. Hence, $d(i)$ can be used by user i to decrypt header encrypted by K, provided that user i is in the receiver set. To decrypt the header, each receiver $i \in \mathbb{R}$ can invoke the KHBE decryption algorithm $\mathsf{BDec}(\cdot)$ and compute

$$k = \mathsf{BDec}(\mathbb{R}, i, d(i), Hdr, K).$$

THEOREM 2. *The generic AHBE scheme has semi-static security if the underlying KHBE has adaptive security.*

PROOF. We construct an algorithm $\mathcal{B}$ to break the semi-static security of the underlying KHBE scheme by invoking the attacker $\mathcal{A}$ against our AHBE scheme.

In the Initialization phase, the attacker $\mathcal{A}$ commits to a set $\tilde{\mathbb{R}} \subseteq \{1, \cdots, n\}$.

In the Setup phase, $\mathcal{B}$ randomly selects $i^* \in \tilde{\mathbb{R}}$. $\mathcal{B}$ setups the semi-static security game with the KHBE challenger $\mathcal{CH}$. $\mathcal{CH}$ will return the system parameters and a KHBE public key denoted by PK_{i^*}. Then $\mathcal{B}$ queries $\mathcal{CH}$ for the secret key $d_{i^*}(j)$ for each index $j \notin \tilde{\mathbb{R}}$. For $i \in \{1, \cdots, n\} \setminus \{i^*\}$, $\mathcal{B}$ generates the KHBE public/private key (PK_i, SK_i) as the real scheme, and can compute the corresponding decryption key $d_i(j)$ for each index $j \in \{1, \cdots, n\} \setminus \{i\}$. For $i = 1, \cdots, n$, $\mathcal{B}$ provides $\mathcal{A}$ with $K_i = \{d_i(j) | 1 \leq i \neq j \leq n\} \cup \{PK_i\}$ as the n receivers' public keys of the AHBE scheme. Clearly, the simulation of each user's public key in the AHBE scheme is perfect.

In the Corruption phase, the attacker $\mathcal{A}$ can query the private key of any user $i \in \{1, \cdots, n\} \setminus \tilde{\mathbb{R}}$. Since the public/private key pairs of the users outside of $\tilde{\mathbb{R}}$ have been generated by following the real scheme, $\mathcal{B}$ can answer the corruption queries correctly.

In the Challenge phase, the attacker $\mathcal{A}$ specifies a challenge set $\mathbb{R}^* \subseteq \mathbb{R}$. If $i^* \notin \mathbb{R}^*$, $\mathcal{B}$ claims failure because in this case, $\mathcal{A}$'s answer will not help $\mathcal{B}$ to break the underlying KHBE scheme. Else if $i^* \in \mathbb{R}^*$, $\mathcal{B}$ forwards $\mathbb{R}^*$ to $\mathcal{CH}$ and requests for a challenge KHBE header from $\mathcal{CH}$. $\mathcal{B}$ will obtain $\langle Hdr^*, k_b \rangle$ under $\langle \mathbb{R}^*, PK_{i^*} \rangle$. The task of $\mathcal{B}$ is to efficiently covert $\langle Hdr^*, k_b \rangle$ into a well-formed challenge under $\langle \mathbb{R}^*, \bigotimes_{i \in \mathbb{R}^*} PK_i \rangle$ to attacker $\mathcal{A}$.

To this end, $\mathcal{B}$ computes $\mathsf{BDec}(\mathbb{R}^*, i^*, d_i(i^*), Hdr^*, PK_i) = k_{b,i}$ for $i \neq i^*$ and $i \in \mathbb{R}^*$, noting that for all $j \in \mathbb{R}^*$, we have that $\mathsf{BDec}(\mathbb{R}^*, j, d_i(j), Hdr^*, PK_i) = k_{b,i}$ due to the second property of the key homomorphism. $\mathcal{B}$ sets $k_b^* = k_b \bigodot_{i \neq i^*}^{i \in \mathbb{R}^*} k_{b,i}$ and sends $\langle Hdr^*, k_b^* \rangle$ to challenge $\mathcal{A}$. Due to the key homomorphism, if k_b is hidden in Hdr^* under $\langle \mathbb{R}^*, PK_{i^*} \rangle$, then k_b^* is hidden in Hdr^* under $\langle \mathbb{R}^*, \bigotimes_{i \in \mathbb{R}^*} PK_i \rangle$; else k_b^* is independent of Hdr^* as a header under the aggregated public key $\bigotimes_{i \in \mathbb{R}^*} PK_i$. Hence, $\langle Hdr^*, k_b^* \rangle$ is a well-formed challenge to $\mathcal{A}$ and has the same distribution as that in the real world.

In the Guess phase, the attacker $\mathcal{A}$ will output a guess bit b'. $\mathcal{B}$ directly uses b' to answer the KHBE challenge. Since k_b^* is a message encryption key hidden in the header Hdr^* under the aggregated public key $\bigotimes_{i \in \mathbb{R}^*} PK_i$ if and only if k_b is a message encryption key hidden in the header Hdr^* under the KHBE public key PK_{i^*}, $\mathcal{B}$ answers correctly if and only if $\mathcal{A}$'s guess is correct. Hence, if $\mathcal{A}$ breaks our AHBE scheme with advantage ε, then B breaks the underlying KHBE scheme with advantage at least $\frac{1}{n}\varepsilon$, where the factor $\frac{1}{n}$ of the reduction loss is introduced by the event $i^* \notin \mathbb{R}^*$ which happens with probability at most $\frac{1}{n}$. As to time complexity, the additional overhead for $\mathcal{B}$ is to generate the public keys for n receivers. This extra overhead is $O(n^2)$. □

The above construction only achieves semi-static security. However, by applying the generic transformation from semi-static security to fully-adaptive security in Section 3.3, the above scheme can be readily improved to meet fully-adaptive security, at a cost of double public keys and ciphertexts.

4.3 An Implementation

4.3.1 Adaptively secure version of the Gentry-Waters BE scheme

Gentry and Waters presented a regular BE with semi-static security and a transformation from semi-static security to adaptive security [18]. In the following, we provide an adaptively secure version of the Gentry-Waters BE scheme by implementing their transformation. Let $g, h_{i,s} (i \in \{1, \cdots, n\}, s \in \{0, 1\})$ be independent generators of bilinear group $\mathbb{G}$ of prime order p. The scheme is as follows.

- $\mathsf{BSetup}(n, n)$: 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 public key is $SK = g^x$.

- $\mathsf{BKeyGen}(i, SK)$: Run $r_i \leftarrow \mathbb{Z}_p, s_i \leftarrow \{0, 1\}$ and output user i's private key $d_i = \langle d_{i,0}, \cdots, d_{i,n}, s_i \rangle$:

$$d_{i,0} = g^{-r_i}, \quad d_{i,i} = g^x h_{i,s_i}^{r_i}, \quad d_{i,j} = h_{j,s_i}^{r_i} (\forall_{j \neq i}).$$

- $\mathsf{BEnc}(\mathbb{R}, PK)$: Randomly pick t in $\mathbb{Z}_p$ and compute $Hdr = (c_1, c_2, c_3)$:

$$c_1 = g^t, c_2 = (\prod_{j \in \mathbb{R}} h_{j,0})^t, c_3 = (\prod_{j \in \mathbb{R}} h_{j,1})^t.$$

Set $k = e(g,g)^{xt}$ and output $\langle Hdr, k\rangle$. Send $\langle \mathbb{R}, Hdr\rangle$ to receivers.

- $\mathsf{BDec}(\mathbb{R}, i, d_i, Hdr, PK)$: If $i \in \mathbb{R}$, receiver i extract k from Hdr with private key d_i by computing

$$\begin{aligned} & e(d_{i,i}\prod_{j\in\mathbb{R}\setminus\{i\}} d_{i,j}, c_1)e(d_{i,0}, c_2) \\ &= e(\prod_{j\in\mathbb{R}} d_{i,j}, c_1)e(d_{i,0}, c_2) \\ &= e(g^x \prod_{j\in\mathbb{R}} h_{j,s_i}^{r_i}), g^t)(g^{-r_i}, (\prod_{j\in\mathbb{R}} h_{j,s_i})^t) \\ &= e(g,g)^{xt} = k. \end{aligned}$$

  Note that the header component $(\prod_{j\in\mathbb{R}} h_{j,1-s_i})^t$ is not used in the decryption procedure. It is just a trick for their security proof.

We define $\odot, \otimes, \bigcirc$ by $d_{1_i} \odot d_{2_i} = \langle d_{1_{i,0}} d_{2_{i,0}}, \cdots, d_{1_{i,n}} d_{2_{i,n}}\rangle$, $PK_1 \otimes PK_2 = PK_1 PK_2$, $k_1 \bigcirc k_2 = k_1 k_2$, respectively. Then we have following lemma.

Lemma 1. For any positive integers n, the following claims hold. (1) The above BE scheme has semi-static security under the decision n-BDHE assumption. (2) The above BE scheme is key homomorphic.

Proof. Claim 1 follows from a combination of Theorem 2.2 and Theorem 3.2 in [18]. Claim 2 follows from a straightforward verification. □

4.3.2 A Basic AHBE Instantiation

Following the generic construction, we instantiate an AHBE scheme with constant size ciphertexts.

- **KeyGen.** Assume the same system parameters as the above KHBE scheme. Then the KeyGen algorithm works as follows.
  - For receiver $i \in \{1, \cdots, n\}$, invoke BSetup to generate a public-private key pair
$$(PK_i, SK_i) = (e(g,g)^{x_i}, g^{x_i})$$
    of the underlying KHBE scheme for randomly chosen x_i in $\mathbb{Z}_p$.
  - Receiver i runs $\mathsf{BKeyGen}$ and obtains $d_i(j) \leftarrow \mathsf{BKeyGen}(j, SK_i)$ for $i, j, l = 1, \cdots, n$, where $d_i(j) = \langle d_{i,0,j}, \cdots, d_{i,n,j}\rangle$:
$$d_{i,0,j} = g^{-r_{i,j}}, d_{i,j,j} = g^{x_i} h_{j,s_i}^{r_{i,j}}, d_{i,l,j} = h_{l,s_i}^{r_{i,j}} (\forall_{l\neq j})$$
    for some $r_{i,j} \leftarrow \mathbb{Z}_p, s_i \leftarrow \{0,1\}$. Receiver i's private key is $d_i(i)$ for the AHBE decryption.
  - Output the receiver i's public key
$$K_i = \{d_i(j) | 1 \le i \neq j \le n\} \cup \{PK_i\}$$
    of the resulting AHBE scheme.

- **AHBEnc.** This procedure works as follows.
  - Decide the receiver set $\mathbb{R} \subseteq \{1, \cdots, n\}$.
  - Extract the broadcast public key for the receivers in $\mathbb{R}$:
$$K = \prod_{i\in\mathbb{R}} PK_i = e(g,g)^{\sum_{i\in\mathbb{R}} x_i}.$$
    Since PK_i's are public, any sender can retrieve them and compute the group public key K for broadcast.
  - Invoke the underlying KHBE encryption algorithm $\mathsf{BEnc}(\cdot)$ to compute the header $Hdr = \mathsf{BEnc}(K, k) = (c_1, c_2, c_3)$:
$$c_1 = g^t, c_2 = (\prod_{j\in\mathbb{R}} h_{j,0})^t, c_3 = (\prod_{j\in\mathbb{R}} h_{j,1})^t,$$
    where t is randomly chosen from $\mathbb{Z}_p$. Set
$$k = K^t = e(g,g)^{t\sum_{i\in\mathbb{R}} x_i}$$
    and send $(\mathbb{R}, Hdr)$ to the receivers.

- **AHBDec** Receiver $i \in \mathbb{R}$ can extract a decryption key under the AHBE public key K by computing

$$\begin{aligned} d(i) &= d_i(i) \bigodot [\bigodot_{j\in\mathbb{R}}^{j\neq i} d_j(i)] = \bigodot_{j\in\mathbb{R}} d_j(i) \\ &= \langle \prod_{j\in\mathbb{R}} d_{j,0,i}, \cdots, \prod_{j\in\mathbb{R}} d_{j,n,i}\rangle. \end{aligned}$$

  Here, $d_i(i)$ is not published and only the receiver i in the receiver set $\mathbb{R}$ can compute it. Due to the key homomorphism of the underlying KHBE scheme, it is easy to see that $d(i) = \bigodot_{j\in\mathbb{R}} d_j(i)$ is a valid decryption key under the aggregated public key $K = \prod_{j\in\mathbb{R}} PK_j$ of the underlying KHBE scheme. Hence, $d(i)$ can be used by user i to decrypt the header under $\langle \mathbb{R}, K\rangle$. To decrypt the header, each receiver $i \in \mathbb{R}$ can invoke the KHBE decryption algorithm $\mathsf{BDec}(\cdot)$ and compute
$$k = \mathsf{BDec}(\mathbb{R}, i, d(i), Hdr, K).$$

From Theorems 2 and Lemma 1, we have the following claim regarding the security of the instantiation. The detailed proof is omitted to avoid repetition.

Corollary 1. The above *ad hoc* broadcast scheme has semi-static security in the standard model under the decision B-DHE assumption.

The above construction only achieves semi-static security. One can follow the generic conversion in Section 3.3 to obtain fully-adaptive security.

4.4 Tradeoff between Ciphertexts and Public keys

In the above basic AHBE construction, the public key of each user consists of $O(n^2)$ elements and the private key includes $O(n)$ elements. This is a heavy burden for an AHBE system of realistic scale, although the ciphertext is of constant size. In the following, we illustrate an efficient tradeoff between the public/private keys and ciphertexts.

Let $n = n_1^3$ and we divide the maximal receiver group $\{i_1, \cdots, i_n\}$ into n_1^2 subgroups each of which hosts at most n_1 receivers. Then we apply our basic AHBE scheme to each subgroup concurrently when a sender wants to broadcast to a set of users $\mathbb{R} \subseteq \{i_1, \cdots, i_n\}$. After employing this approach, the public key of each user consists of $O(n_1^2)$ elements and the secret key contains $O(n_1)$ elements, at a cost that the AHBE ciphertext includes $O(n_1^2)$ KHBE ciphertexts. Hence, the resulting AHBE scheme has sub-linear complexity, i.e., $O(n^{\frac{2}{3}})$ size public keys and ciphertexts, and $O(n^{\frac{1}{3}})$ size private keys. This performance is comparable to the up-to-date conventional broadcast schemes [2, 4, 5, 18] which exploit a similar subgroup partition approach to obtain sub-linear complexity $O(\sqrt{n})$.

5. CONCLUSION

We proposed the notion of AHBE which allows a sender to dynamically broadcast to any *ad hoc* group without the help of a trusted dealer. We presented the first AHBE schemes which are proven to be adaptively secure in the standard model under some well-understood computational assumptions. The scheme enjoys non-interactive decryption and has sub-linear complexity comparable to the up-to-date broadcast systems requiring a trusted dealer to initialize the system.

6. REFERENCES

[1] D. Boneh, M. Franklin. Identity Based Encryption from the Weil Pairing. *SIAM J. of Computing*, vol. 32(3), 586-615, 2003.

[2] D. Boneh, C. Gentry, B. Waters. Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys. In: Shoup, V. (ed.) *CRYPTO'05. LNCS*, vol. 3621, pp. 258-275. Springer, Heidelberg, 2005.

[3] D. Boneh, B. Lynn, H. Shacham. Short Signatures from the Weil Pairing. In: Boyd, C. (ed.) *ASIACRYPT'01. LNCS*, vol. 2248, pp. 514-532. Springer, Heidelberg, 2001.

[4] D. Boneh, A. Sahai, B. Waters. Fully Collusion Resistant Traitor Tracing with Short Ciphertexts and Private Keys. In: Vaudenay, S. (ed.) *EUROCRYPT'06. LNCS*, vol. 4004, pp. 573-592. Springer, Heidelberg, 2006.

[5] D. Boneh, B. Waters. A Fully Collusion Resistant Broadcast, Trace, and Revoke System. In: Juels A., Wright R.-N., De Capitani di V.S. (Eds.) *ACM CCS'06*, pp. 211-220. ACM Press, 2006.

[6] E. Bresson, O. Chevassut, D. Pointcheval. Provably Authenticated Group Diffie-Hellman Key Exchange – The Dynamic Case. In: Boyd, C. (Ed.) *ASIACRYPT'01. LNCS*, vol. 2248, pp. 290-309. Springer, Heidelberg, 2001.

[7] E. Bresson, O. Chevassut, D. Pointcheval. Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions. In: Knudsen L.R. (Ed.) *EUROCRYPT'02. LNCS*, vol. 2332, pp. 321-336. Springer, Heidelberg, 2002.

[8] E. Bresson, O. Chevassut, D. Pointcheval, J.J. Quisquater. Provably Authenticated Group Diffie-Hellman Key Exchange. In: Samarati, P. (Ed.) *ACM CCS'01*, pp. 255-264. ACM Press, 2001.

[9] M. Burmester, Y. Desmedt. A Secure and Efficient Conference Key Distribution System. In: Santis, A.D. (Ed.) *EUROCRYPT'94. LNCS*, vol. 950, pp. 275-286. Springer, Heidelberg, 1994.

[10] C. Cachin, K. Kursawe, V. Shoup. Random Oracles in Constantinople: Practical Asynchronous Byzantine Agreement Using Cryptography. In: Neiger G. (Ed.) *PODC'00*, pp. 123-132. ACM Press, 2000.

[11] C. Cachin, Reto. Strob. Asynchronous Group Key Exchange with Failures. In: Chaudhuri S., Kutten S. (Eds.) *PODC'04*, pp. 357-366. ACM Press, 2004.

[12] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, B. Pinkas. Multicast Security: A Taxonomy and some Efficient Constructions. In: *IEEE INFOCOM'99*, vol. 2, pp. 708-716. New York, NY, 1999.

[13] R. Canetti, T. Malkin, K. Nissim. Efficient Communication-storage Tradeoffs for Multicast Encryption. In: Stern, J. (ed.) *EUROCRYPT'99, LNCS*, vol. 1592, pp. 459-474. Springer, Heidelberg, 1999.

[14] Y. Dodis, N. Fazio. Public Key Broadcast Encryption for Stateless Receivers. In: Feigenbaum J. (ed.) *DRM'02. LNCS*, vol. 2696, pp. 61-80. Springer, Heidelberg (2003)

[15] T. ElGamal. A Public Key Cryptosystem and Signature Scheme Based on Discrete Logarithms. *IEEE Transaction on Information Theory*, vol. 31: 467-472, 1985.

[16] A. Fiat, M. Naor. Broadcast Encryption. In: Stinson, D.R. (ed.) *CRYPTO'93. LNCS*, vol. 773, pp. 480-491. Springer, Heidelberg, 1994.

[17] S.D. Galbraith, V. Rotger. Easy Decision Diffie-Hellman Groups. *Journal of Computation and Mathematics* vol. 7: 201-218, 2004.

[18] C. Gentry, B. Waters. Adaptive Security in Broadcast Encryption Systems (with Short Ciphertexts). In: Joux, A. (ed.) *EUROCRYPT'09. LNCS*, vol. 5479, pp. 171-188. Springer, Heidelberg, 2009.

[19] S. Goldwasser, S. Micali, R. Rivest. A Digital Signature Scheme Secure against Adaptive Chosen-message Attacks. *SIAM J. Computing*, vol.17(2), 281-308, 1988.

[20] M.T. Goodrich, J.Z. Sun, R. Tamassia. Efficient Tree-Based Revocation in Groups of Low-State Devices. In: Franklin, M. (ed.) *CRYPTO'04. LNCS*, vol. 3152, pp. 511-527. Springer, Heidelberg, 2004.

[21] D. Halevy, A. Shamir. The LSD Broadcast Encryption Scheme. In: Yung, M. (ed.) *CRYPTO'02. LNCS*, vol. 2442, pp. 47-60. Springer, Heidelberg, 2002.

[22] J. Katz, N. Wang. Efficiency Improvements for Signature Schemes with Tight Security Reductions. In: Jajodia S., Atluri V., Jaeger T. (Eds.) *ACM CCS'03*, pp. 155-164, ACM Press, 2003.

[23] R. Rivest, A. Shamir, L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, vol. 21(2): 120ÍC126, 1978.

[24] A.T. Sherman, D.A. McGrew. Key Establishment in Large Dynamic Groups using One-way Function Trees. *IEEE Trans. Softw. Eng.*, vol. 29(5): 444-458, 2003.

[25] D.M. Wallner, E.J. Harder, R.C. Agee. Key Management for Multiast: Issues and Architectures. *IETF draft wallner-key*, 1997.

[26] C.K. Wong, M. Gouda, S. Lam. Secure Group Communications using Key Graphs. *IEEE/ACM Transactions on Networking*, vol. 8(1): 16-30, 2000.

[27] Q. Wu, Y. Mu, W. Susilo, B. Qin, J. Domingo-Ferrer. Asymmetric Group Key Agreement. In: Joux, A. (ed.) *EUROCRYPT'09, LNCS*, vol. 5479, pp. 153-170. Springer, Heidelberg, 2009.