Public Key Broadcast Encryption with Low Number of Keys and Constant Decryption Time *

Yi-Ru Liu, Wen-Guey Tzeng
Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan 30050

Abstract

In this paper we propose two public-key BE schemes that have efficient complexity measures. The first scheme, called the BE-PI scheme, has $O(r)$ header size, $O(1)$ public keys and $O(\log N)$ private keys, where r is the number of revoked users. This is the first public-key BE scheme that has both public and private keys under $O(\log N)$ while the header size is $O(r)$. These complexity measures of the header size and private keys also match those of efficient secret-key broadcast encryption schemes.

Our second scheme, called the PK-SD-PI scheme, has $O(r)$ header size, $O(1)$ public key and $O(\log^2 N)$ private keys. They are the same as those of the SD scheme. Nevertheless, the decryption time is remarkably $O(1)$. This is the first public-key BE scheme that has $O(1)$ decryption time while other complexity measures are kept low. The PK-LSD-PI scheme can be constructed in the same way. It has $O(r/\epsilon)$ ciphertext size and $O(\log^{1+\epsilon} N)$ private keys, where $0 < \epsilon < 1$. The decryption time is also $O(1)$.

Our basic schemes are one-way secure against *full collusion of revoked users*. With a slight modification, we make both schemes indistinguishably secure against the adaptive chosen ciphertext attack. The BE-PI scheme has the capability of *tracing traitors*. It is able to find out what private keys are used in a confiscated decoding box.

Keywords: Broadcast encryption, polynomial interpolation, collusion.

1 Introduction

Assume that there is a set $\mathcal{U}$ of N users. We would like to broadcast a message to a subset S of the users such that only the (authorized) users in S can obtain the message, while the (revoked) users not in S cannot get information about the message. Broadcast encryption is a bandwidth-saving method to achieve this gaol via cryptographic key-controlled access. In broadcast encryption, a dealer sets up the system and assigns each user a set of private keys such that the broadcasted messages can be decrypted by authorized users only. Broadcast encryption has many applications, such as in the pay TV system, encrypted file sharing system, digital right management of digital content, content protection of recordable data, etc.

*Supported in part by NSC project NSC 95-2221-E-009-030 and Taiwan Information Security Center TWISC@NCTU.

A broadcasted message M is usually sent in the form $\langle Hdr(S,m), E_m(M)\rangle$, where m is a session key for encrypting M via a symmetric encryption method E. An authorized user in S can use his private keys to decrypt the session key m from $Hdr(S,m)$. Since the size of $E_m(M)$ is pretty much the same for all broadcast encryption schemes, we are concerned about the header size. The performance measures of a broadcast encryption scheme are the header size, the number of private keys held by each user, the size of public parameters of the system (public keys), the time for encrypting a message, and the time for decrypting the header by an authorized user. A broadcast encryption scheme should be able to resist the collusion attack from revoked users. A scheme is *fully collusion-resistant* if even all revoked users collude, they get no information about the broadcasted message.

Broadcast encryption schemes can be *static* or *dynamic*. For a dynamic broadcast encryption scheme, the private keys of a user can be updated from time to time, while the private keys of a user in a static broadcast encryption scheme remain the same through the lifetime of the system. Broadcast encryption schemes can also be public-key or secret-key. For a public-key broadcast encryption scheme, any one (broadcaster) can broadcast a message to an arbitrary group of authorized users by using the public system parameters, while for a secret-key broadcast encryption scheme, only the special dealer, who knows the secrets of the system or the private keys of users, can broadcast a message.

In this paper we refer "static public-key broadcast encryption" as "public-key BE".

1.1 Our contribution

We propose two public-key BE schemes that have efficient complexity measures. The first scheme, called the BE-PI scheme (broadcast encryption with polynomial interpolation), has $O(r)$ header size, $O(1)$ public keys, and $O(\log N)$ private keys, where r is the number of revoked users. This is the first public-key BE scheme that has both public and private keys under $O(\log N)$ while the header size is $O(r)$. These complexity measures of the header size and private keys also match those of efficient secret-key broadcast encryption schemes [11, 20, 21]. The idea of this scheme is to run $\log N$ copies of the basic scheme in [17, 19, 22] in parallel for lifting the restriction on a priori fixed number of revoked users. If we implement the $\log N$ copies straightforwardly, we would get a scheme of public key size $O(N)$. Nevertheless, we are able to use the properties of bilinear maps as well as special private key assignment to eliminate the need of $O(N)$ public keys and make it a constant number.

Our second scheme, called the PK-SD-PI scheme (public-key SD scheme with polynomial interpolation), is constructed by combining the polynomial interpolation technique used in the BE-PI scheme and the subset cover method used in the SD scheme [16]. The PK-SD-PI scheme has $O(r)$ header size, $O(1)$ public key and $O(\log^2 N)$ private keys. They are the same as those of the SD scheme. Nevertheless, the decryption time is remarkably $O(1)$. This is the first public-key broadcast encryption scheme that has $O(1)$ decryption time while other complexity measures are kept low. The PK-LSD-PI scheme can be constructed in the same way. It has $O(r/\epsilon)$ ciphertext size and $O(\log^{1+\epsilon} N)$ private keys, where $0 < \epsilon < 1$. The decryption time is also $O(1)$.

Our basic schemes are one-way secure against *full collusion of revoked users*. With a slight modification, we make both schemes indistinguishably secure against the adaptive chosen ciphertext attack. The BE-PI scheme has the capability of *tracing traitors*. It is able to find out what private keys are used in a confiscated decoding box. The comparison with some other public-key BE

Table 1: Comparison of some fully collusion-resistant public-key BE schemes.

	header size	public-key size	private-key size	decryption cost$^\natural$
PK-SD-HIBE†	$O(r)$	$O(1)$	$O(\log^2 N)$	$O(\log N)$
BGW-I [4]	$O(1)$	$O(N)^\flat$	$O(1)$	$O(N-r)$
BGW-II [4]	$O(\sqrt{N})$	$O(\sqrt{N})^\flat$	$O(1)$	$O(\sqrt{N})$
BW[5]	$O(\sqrt{N})$	$O(\sqrt{N})^\flat$	$O(\sqrt{N})$	$O(\sqrt{N})$
LHL§ [15]	$O(rD)$	$O(2C)^\flat$	$O(D)$	$O(C)$
P-NP, P-TT, P-YF‡	$O(r)$	$O(N)$	$O(\log N)$	$O(r)$
Our work: BE-PI	$O(r)$	$O(1)$	$O(\log N)$	$O(r)$
Our work: PK-SD-PI	$O(r)$	$O(1)$	$O(\log^2 N)$	$O(1)$
Our work: PK-LSD-PI	$O(r/\epsilon)$	$O(1)$	$O(\log^{1+\epsilon} N)$	$O(1)$

N - the number of users.
r - the number of revoked users.
† - the transformed SD scheme [6] instantiated with constant-size HIBE [2].
‡ - the parallel extension of [17, 19, 22].
$^\flat$ - the public keys are needed for decrypting the header by a user.
§ - $N = C^D$.
$^\natural$ - group operation/modular exponentiation and excluding the time for scanning the header.

schemes with full collusion resistance is shown in Table 1.

There have been many proposed broadcast encryption schemes. One can choose a suitable scheme to fit his need for some particular environment and restrictions. Our schemes can be used in the scenario of mobile TV. Many channel (content) providers broadcast their programs to the mobile TV subscribers. If the secret-key system is used, the system needs to give the system secrets or the private keys of users to channel providers. This is really undesirable and may jeopardize security of the system. Thus, we prefer the public-key BE system here. Mobile subscribers usually use portable devices to watch TV. But, the storage of today's portable devices is quite limited due to manufacturing cost consideration. It is sometimes not possible to store a large number of keys in their local storage. Furthermore, two-way communication with the outside is either unavailable or of low bandwidth. If the public keys are needed in decryption, it is not practical for the portable devices to put the public keys outside and access them on demand. Thus, we prefer the system with *low* public and private keys [11]. Our schemes have $O(1)$ public key, and $O(\log N)$ or $O(\log^2 N)$ private keys. A mobile TV device of using the PK-SD-PI scheme could save quite a lot of time in processing the header of broadcasted stream video since the decryption time of the header is $O(1)$.

1.2 Related work

Fiat and Naor [8] formally proposed the concept of static secret-key broadcast encryption. Many researchers followed to propose various broadcast encryption schemes, e.g., see [11, 12, 16, 17, 20].

Kurosawa and Desmedt [13] proposed a pubic-key BE scheme that is based on polynomial interpolation and traces at most k traitors. The similar schemes of Noar and Pinkas [17], Tzeng and Tzeng [19], and Yoshida and Fujiwara [22] allow revocation of up to k users. Kurosawa and Yoshida [14] generalized the polynomial interpolation (in fact, the Reed-Solomon code) to any linear

code for constructing public-key BE schemes. The schemes in [7, 13, 14, 17, 19, 22] all have $O(k)$ public keys, $O(1)$ private keys, and $O(r)$ header size, $r \leq k$. However, k is a-priori fixed during the system setting and the public key size depends on it. These schemes can withstand the collusion attack of up to k revoked users only. Thus, they are not fully collusion-resistant.

Yoo, et al. [21] observed that the restriction of a pre-fixed k can be lifted by running $\log N$ copies of the basic scheme with different degrees (from 2^0 to $2^{\lceil \log N \rceil}$) of polynomials. This results in a scheme of $O(\log N)$ private keys and $O(r)$ header size such that r is not restricted. However, their scheme is secret-key and the system has $O(N)$ secret values. In the public-key setting, the public-key size is $O(N)$.

Recently Boneh, et al. [4] proposed a public-key BE scheme that has $O(1)$ header size, $O(1)$ private keys, and $O(N)$ public keys. With some modification by trading off the header size and public keys, they gave another scheme with $O(\sqrt{N})$ header size, $O(1)$ private keys and $O(\sqrt{N})$ public keys. Lee, et al. [15] proposed a better trade-off by using receiver identifiers in the scheme. It can achieve $O(1)$ public key, $O(\log N)$ private keys, but, $O(r \log N)$ header size. Boneh and Waters [5] combined the scheme in [4] and augmented broadcast encryption to form a public-key BE scheme that has the traitor tracing capability. Their scheme is secure against adaptive adversaries and its performance complexities are all $O(\sqrt{N})$. This type of schemes [4, 5, 15] has the disadvantage that the public keys are needed by a user in decrypting the header. Thus, the *de-facto* private key of a user is the combination of the public key and his private key.

It is possible to transform a secret-key broadcast encryption scheme into a public-key BE scheme. For example, Dodis and Fazio [6] transformed the SD and LSD schemes [12, 16] into public-key SD and LSD schemes, shorted as PK-SD and PK-LSD. The transformation employs the technique of hierarchical identity-based encryption [10] to substitute for the hash function that is used for private key derivation. Instantiated with the newest constant-size hierarchical identity-based encryption [2], the PK-SD scheme has $O(r)$ header size, $O(1)$ public keys and $O(\log^2 N)$ private keys. The PK-LSD scheme has $O(r/\epsilon)$ header size, $O(1)$ public keys and $O(\log^{1+\epsilon} N)$ private keys, where $0 < \epsilon < 1$ is a constant. The decryption costs of the PK-SD and PK-LSD schemes are both $O(\log N)$, which is the time for key derivation incurred by the original relation of private keys. If we apply the HIBE technique to the secret-key broadcast encryption schemes of $O(\log N)$ or $O(1)$ private keys [1, 11, 20], we would get their public-key versions with $O(N)$ private keys and $O(N)$ decryption time.

2 Preliminaries

Bilinear map. We use the properties of bilinear maps. Let G_q and G_1 be two (multiplicative) cyclic groups of prime order q and $\hat{e}$ be a bilinear map from $G_q \times G_q$ to G_1. Then, $\hat{e}$ has the following properties.

1. For all $u, v \in G_q$ and $x, y \in Z_q$, $\hat{e}(u^x, v^y) = \hat{e}(u, v)^{xy}$.
2. Let g be a generator of G_q, we have $\hat{e}(g, g) = g_1 \neq 1$ is a generator of G_1.

BDH hardness assumption. The BDH problem is to compute $\hat{e}(g, h)^{s^2}$ from given (g, h, g^s), where g, h are random generators of G_q and s is random over Z_q. We say that BDH is (t, ϵ)-hard if for any probabilistic algorithm A with time bound t, there is some k_0 such that for any $k \geq k_0$,

$$\Pr[A(g, h, g^s) = \hat{e}(g, h)^{s^2} : g, h \xleftarrow{u} G_q \backslash \{1\}, s \xleftarrow{u} Z_q] \leq \epsilon.$$

This version of the BDH problem is equivalent to computing $\hat{e}(g,g)^{abc}$ from given (g^a, g^b, g^c). To take a quick look for one direction, we let $h = g^c$, $g^{s_1} = g^{a+b}$ and $g^{s_2} = g^{a-b}$. Then,

$$\hat{e}(g,g)^{abc} = (\hat{e}(g,h)^{s_1^2}/\hat{e}(g,h)^{s_2^2})^{1/4}.$$

Broadcast encryption. A public-key BE scheme Π consists of three probabilistic polynomial-time algorithms:

- Setup(1^z, ID, $\mathcal{U}$). Wlog, let $\mathcal{U} = \{U_1, U_2, \ldots, U_N\}$. It takes as input the security parameter z, a system identity ID and a set $\mathcal{U}$ of users and outputs a public key PK and N private key sets $SK_1, SK_2, \ldots, SK_N$, one for each user in $\mathcal{U}$.

- $Enc(PK, S, M)$. It takes as input the public key PK, a set $S \subseteq \mathcal{U}$ of authorized users and a message M. It outputs a pair $\langle Hdr(S,m), C\rangle$ of the ciphertext header and body, where m is a randomly generated session key and C is the ciphertext of M encrypted by m via some standard symmetric encryption scheme, e.g., AES.

- $Dec(SK_k, Hdr(S,m), C)$. It takes as input the private key SK_k of user U_k, the header $Hdr(S,m)$ and the body C. If $U_k \notin S$, he cannot decrypt C to obtain the message M. If $U_k \in S$, it can decrypt the header $Hdr(S,m)$ to obtain the session key m and then uses m to decrypt the ciphertext body C for the message M.

The system is correct if all users in S can get the broadcasted message M.

Security. We describe the indistinguishability security against the adaptive chosen ciphertext attack (IND-CCA security) for broadcast encryption as follows [4]. Here, we focus on the security of the session key, which in turn guarantees the security of the ciphertext body C. Let Enc^* and Dec^* be like Enc and Dec except that the message M and the ciphertext body C are ignored. The security is defined by an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$ via the following game.

Init. The adversary $\mathcal{A}$ chooses a system identity ID and a subset $S^* \subseteq \mathcal{U}$ of users that it wants to attack.

Setup. The challenger $\mathcal{C}$ runs Setup(1^z, ID, $\mathcal{U}$) to generate a public key PK and private key sets $SK_1, SK_2, \ldots, SK_N$. The challenger $\mathcal{C}$ gives $SK_i, U_i \notin S^*$ to $\mathcal{A}$.

Query phase 1. The adversary $\mathcal{A}$ issues decryption queries Q_i, $1 \le i \le n$, of form (U_k, S, Hdr), $S \subseteq S^*$, $U_k \in S$, and the challenger $\mathcal{C}$ responds with $Dec^*(SK_k, Hdr)$, which is the session key encrypted in Hdr.

Challenge. The challenger $\mathcal{C}$ runs $Enc^*(PK, S^*)$ and outputs $Hdr^*(S^*, m)$, where m is randomly chosen. Then, $\mathcal{C}$ chooses a random bit b and a random session key m^* and sets $m_b = m$ and $m_{1-b} = m^*$. $\mathcal{C}$ gives $Hdr^*(S^*, m), m_0, m_1)$ to $\mathcal{A}$.

Query phase 2. The adversary $\mathcal{A}$ issues more decryption queries Q_i, $n+1 \le i \le q_D$, of form (U_k, S, Hdr), $S \subseteq S^*, U_k \in S, Hdr \neq Hdr^*$, and the challenger $\mathcal{C}$ responds with $Dec^*(SK_k, Hdr)$.

Guess. $\mathcal{A}$ outputs a guess b' for b.

In the above the adversary A is static since it chooses the target user set S^* before the system setup. Let $\mathrm{Adv}^{ind\text{-}cca}_{\mathcal{A},\Pi}(z)$ be the advantage that $\mathcal{A}$ wins the above game, that is,

$$\mathrm{Adv}^{ind\text{-}cca}_{\mathcal{A},\Pi}(z) = 2 \cdot \Pr[\mathcal{A}^{\mathcal{O}}(PK, SK_{\mathcal{U}\setminus S^*}, Hdr^*, m_0, m_1) = b : \\ S^* \subseteq \mathcal{U}, (PK, SK_{\mathcal{U}}) \leftarrow Setup(1^z, \textsc{Id}, \mathcal{U}), Hdr^* \leftarrow Enc^*(PK, S^*), b \stackrel{u}{\leftarrow} \{0,1\}] - 1,$$

where $SK_{\mathcal{U}} = \{SK_i : 1 \le i \le N\}$ and $SK_{\mathcal{U}\setminus S^*} = \{SK_i : U_i \notin S^*\}$.

Definition 1. *A public-key BE scheme* Π*=(Setup, Enc, Dec) is* (t, ϵ, q_D)*-IND-CCA secure if for all* t*-time bounded adversary* $\mathcal{A}$ *that makes at most* q_D *decryption queries, we have* $Adv^{ind\text{-}cca}_{\mathcal{A},\Pi}(z) < \epsilon$.

3 The BE-PI scheme

We now present our BE-PI scheme as follows. The encrypted session key is one-way secure against the collusion attack of all revoked users. We shall discuss how to transform it to be indistinguishably secure against the adaptive chosen ciphertext attack in Section 4.

1. **Setup**(1^z, Id, $\mathcal{U}$): z is the security parameter, Id is the identity name of the system, and $\mathcal{U} = \{U_1, U_2, \ldots, U_N\}$ is the set of users in the system. Let G_q and G_1 be the bilinear groups with the pairing function $\hat{e}$, where q is a large prime. This bilinear system as described above is of security parameter z. Then, the system dealer does the following:

   - Choose a cryptographically secure hash function $H : \{0,1\}^* \to G_q$.
   - Choose a secure symmetric encryption scheme E with key space G_q.
   - Choose a generator g of group G_q, and let $\lg = \log_g$ and $g_1 = \hat{e}(g, g)$.
   - Compute
   $$h_i = H(\textsc{Id} \| \text{"}h\text{"} \| i)$$
   for $1 \le i \le \lceil \log_2 N \rceil$, where "$h$" indicates the h-related hash values.
   - Compute
   $$g^{a_j^{(i)}} = H(\textsc{Id} \| \text{"}f\text{"} \| i \| j)$$
   for $0 \le i \le \lceil \log_2 N \rceil$ and $0 \le j \le 2^i$, where "f" means polynomial-related parameters.
   Remark. The underlined polynomials, are, $0 \le i \le \lceil \log N \rceil$,
   $$f_i(x) = \sum_{j=0}^{2^i} a_j^{(i)} x^j \pmod q.$$
   The system dealer does not know the coefficients $a_j^{(i)} = \lg H(\textsc{Id} \| \text{"}f\text{"} \| i \| j)$. But, this does not matter.
   - Randomly choose a secret $\rho \in Z_q$ and compute g^ρ.
   - Publish the public key $PK = (\textsc{Id}, H, E, G_q, G_1, \hat{e}, g, g^\rho)$.

- Assign a set $SK_k = \{s_{k,0}, s_{k,1}, \ldots, s_{k,\lceil \log N \rceil}\}$ of private keys to user U_k, $1 \leq k \leq N$, where
$$s_{k,i} = (g^{r_{k,i}}, g^{r_{k,i} f_i(k)}, g^{r_{k,i} f_i(0)} h_i^{\rho})$$
and $r_{k,i}$ is randomly chosen from Z_q, $1 \leq i \leq \lceil \log N \rceil$.

2. **Enc**(PK, S, M): $S \subseteq \mathcal{U}$, $R = \mathcal{U} \backslash S = \{U_{i_1}, U_{i_2}, \ldots, U_{i_l}\}$ is the set of revoked users, where $l \geq 1$. M is the sent message. The broadcaster does the following:

- Let $\alpha = \lceil \log_2 l \rceil$ and $L = 2^{\alpha}$.
- Compute $h_\alpha = H(\text{ID} \| \text{"}h\text{"} \| \alpha)$.
- Randomly select distinct $i_{l+1}, i_{l+2}, \ldots, i_L > N$. These $U_{i_t}, l+1 \leq t \leq L$, are dummy users.
- Randomly select a session key $m \in G_q$.
- Randomly select $r \in Z_q$ and compute, $1 \leq t \leq L$,
$$g^{r f_\alpha(i_t)} = (\prod_{j=0}^{L} H(\text{ID} \| \text{"}f\text{"} \| \alpha \| j)^{i_t^j})^r.$$
- The ciphertext header $Hdr(S, m)$ is
$$(\alpha, m\hat{e}(g^{\rho}, h_\alpha)^r, g^r, (i_1, g^{r f_\alpha(i_1)}), (i_2, g^{r f_\alpha(i_2)}), \ldots, (i_L, g^{r f_\alpha(i_L)})).$$
- The ciphertext body is $C = E_m(M)$.

3. **Dec**$(SK_k, Hdr(S, m), C)$: $U_k \in S$. The user U_k does the following.

- Compute $b_0 = \hat{e}(g^r, g^{r_{k,\alpha} f_\alpha(k)}) = g_1^{r r_{k,\alpha} f_\alpha(k)}$.
- Compute $b_j = \hat{e}(g^{r_{k,\alpha}}, g^{r f_\alpha(i_j)}) = g_1^{r r_{k,\alpha} f_\alpha(i_j)}$, $1 \leq j \leq L$.
- Use the Lagrange interpolation method to compute
$$g_1^{r r_{k,\alpha} f_\alpha(0)} = \prod_{j=0}^{L} b_j^{\lambda_j}, \tag{1}$$
where $\lambda_j = \frac{(-i_0)(-i_1)\cdots(-i_{j-1})(-i_{j+1})\cdots(-i_L)}{(i_j - i_0)(i_j - i_1)\cdots(i_j - i_{j-1})(i_j - i_{j+1})\cdots(i_j - i_L)} \pmod q$, $i_0 = k$.
- Compute the session key
$$\frac{m\hat{e}(g^{\rho}, h_\alpha)^r \cdot g_1^{r r_{k,\alpha} f_\alpha(0)}}{\hat{e}(g^r, g^{r_{k,\alpha} f_\alpha(0)} h_\alpha^{\rho})} = \frac{m\hat{e}(g^{\rho}, h_\alpha)^r \cdot g_1^{r r_{k,\alpha} f_\alpha(0)}}{\hat{e}(g^r, h_\alpha^{\rho}) \cdot g_1^{r r_{k,\alpha} f_\alpha(0)}} = m. \tag{2}$$
- Use m to decrypt the ciphertext body C to obtain the message M.

Correctness. We can easily see that the scheme is correct by Equation (2).

3.1 Performance analysis

For each system, the public key is $(\text{ID}, H, E, G_q, G_1, \hat{e}, g, g^\rho)$, which is of size $O(1)$. Since all systems can use the same $(H, E, G_q, G_1, \hat{e}, g)$, the real public key specific to a system is simply (ID, g^ρ). Each system dealer has a secret ρ for assigning private keys to its users. Each user U_k holds private keys $SK_k = \{s_{k,0}, s_{k,1}, \ldots, s_{k,\lceil \log N \rceil}\}$, which each corresponds to a share of polynomial f_i in the masked form, $0 \leq i \leq \lceil \log N \rceil$. The number of private keys is $O(\log N)$. When r users are revoked, we choose the polynomial f_α of degree 2^α for encrypting the session key, where $2^{\alpha-1} < r \leq 2^\alpha$. Thus, the header size is $O(2^\alpha) = O(r)$. It is actually no more than $2r$.

Since evaluation of a hash function is much faster than computation of a pairing and a modular exponentiation, we omit the cost of evaluating hash functions. To prepare a header, the broadcaster needs to do one pairing function and $2^\alpha + 2$ modular exponentiations, which is $O(r)$ modular exponentiations.

For a user in S to decrypt a header, with a little re-arrangement of Equation (1) as

$$\prod_{j=0}^{L} b_j^{\lambda_j} = b_0^{\lambda_0} \cdot \hat{e}(g^{r_{k,\alpha}}, \prod_{j=1}^{L} (g^{r f_\alpha(i_j)})^{\lambda_j}),$$

the user needs to perform 3 pairing functions and 2^α modular exponentiations, which is $O(r)$ modular exponentiations. The evaluation of λ_j's can be done in $O(L) = O(2r)$ if the header consists of $\tilde{\lambda}_j = \frac{(-i_1)\cdots(-i_{j-1})(-i_{j+1})\cdots(-i_L)}{(i_j-i_1)\cdots(i_j-i_{j-1})(i_j-i_{j+1})\cdots(i_j-i_L)} \pmod q$, $1 \leq j \leq L$. The user can easily compute λ_j's from $\tilde{\lambda}_j$'s. Inclusion of $\tilde{\lambda}_j$'s in the header does not affect the order of the header size.

3.2 Security analysis

We show that the BE-PI scheme is fully collusion-resistant. No matter how many revoked users collude, they cannot compute the session key m. We show that it is one-way secure (without decryption queries). The definition of one-wayness security is similar to the indistinguishability security except that the adversary, who controls the set $\mathcal{U} \backslash S^*$ of revoked users, is required to compute the session key m from the challenge $Hdr^*(S^*, m)$, where S^* is chosen by the adversary in advance. Later, we shall show how to achieve the IND-CCA security. Let q_H be the number of queries to the hash function H by the collusion of the revoked users.

Theorem 1. *Assume that the BDH problem is (t_1, ϵ_1)-hard. For any $0 \leq \alpha \leq \lceil \log_2 N \rceil$, if the number of revoked users is no more than $L = 2^\alpha$, any collusion of them cannot decrypt the header to obtain the session key with probability $\epsilon = \epsilon'$, time bound $t = t_1 - t'$ and q_H hash oracles under the random oracle model, where t' is polynomially bounded and $q_H \leq t$.*

Proof. We reduce the BDH problem to the problem of computing the session key from the header by the revoked users. Since the polynomials $f_i(x) = \sum_{j=0}^{L} a_j^{(i)} x^j$ and secret shares of users for the polynomials are independent for different i's. We simply discuss security for a particular α. Without loss of generality, let $R = \{U_1, U_2, \ldots, U_L\}$ be the set of revoked users and $S^* = \mathcal{U} \backslash R$. Note that S^* was chosen by the adversary in advance. Let the input of the BDH problem be (g, h, g^s), where the pairing function is implicitly known. We set the parameters of decrypting the header as follows:

1. Randomly select $\tau, \kappa, \mu_1, \mu_2, \ldots, \mu_L, w_1, w_2, \ldots, w_L \in Z_q$.

2. Set the public key of the system:
   (a) Let the input g be the generator g in the system.
   (b) Set $f_\alpha(i) = w_i, 1 \le i \le L$.
   (c) Let $g^{a_0^{(\alpha)}} = g^{f_\alpha(0)} = g^s \cdot g^\tau = g^{s+\tau}$.
   (d) Compute $g^{a_i^{(\alpha)}}$, $1 \le i \le L$, from $g^{a_0^{(\alpha)}}$ and $g^{f_\alpha(j)} = g^{w_j}, 1 \le j \le L$. This can be done by the Lagrange interpolation method over exponents.
   (e) Set $h_\alpha = g^s \cdot g^\kappa = g^{s+\kappa}$.
   (f) Set $g^\rho = g^s$.
3. Set the secret key $(g^{r_{i,\alpha}}, g^{r_{i,\alpha} f_\alpha(i)}, g^{r_{i,\alpha} f_\alpha(0)} h_\alpha^s)$ of the revoked user $U_i, 1 \le i \le L$, as follows:
   (a) Let $g^{r_{i,\alpha}} = g^{-s} \cdot g^{\mu_i} = g^{-s+\mu_i}$.
   (b) Compute $g^{r_{i,\alpha} f_\alpha(i)} = (g^{r_{i,\alpha}})^{w_i}, 1 \le i \le L$.
   (c) Compute $g^{r_{i,\alpha} f_\alpha(0)} h_\alpha^s = g^{(-s+\mu_i)(s+\tau)} (g^{s+\kappa})^s = g^{-s(\mu_i - \tau + \kappa)}$.
4. Set the header $(\alpha, m\hat{e}(g^\rho, h_\alpha)^r, g^r, (1, g^{r f_\alpha(1)}), (2, g^{r f_\alpha(2)}), \ldots, (L, g^{r f_\alpha(L)}))$ as follows:
   (a) Let $g^r = h$.
   (b) Compute $g^{r f_\alpha(i)} = (g^r)^{w_i}, 1 \le i \le L$.
   (c) Randomly select $y \in G_1$ and set $m\hat{e}(g^\rho, h_\alpha)^r = y$. We do not know what m is. But, this does not matter.

Assume that the revoked users together can compute the session key m. During computation, the users can query hash oracles $H(\cdot)$. If the query is of the right form $H(\text{ID}\|\text{"}f\text{"}\|\alpha\|j)$ or $H(\text{ID}\|\text{"}h\text{"}\|\alpha)$, we set them to be $g^{a_j^{(\alpha)}}$ and h_α, respectively. If the query has ever been asked, we return the stored hash value for the query. For other non-queried inputs, we return random values in G_q.

We should check whether the distributions of the parameters in our reduction and those in the system are equal. Since $\tau, w_1, w_2, \ldots, w_L$ are randomly chosen, $g^{a_i^{(\alpha)}}, 0 \le i \le L$ are distributed uniformly over G_q^{L+1}. Due to the random oracle model, they are distributed in the same way as those in the system. Since $\kappa, \mu_1, \mu_2, \ldots, \mu_L$ are randomly chosen, the distribution of h_α and $g^{r_{i,\alpha}}, 1 \le i \le L$ are uniform over G_q^{L+1}, which is again the same as that of the corresponding system parameters. The distributions of g^r in the header and g^ρ in the public key are both uniform over G_q. They are the same as the distributions of the given input h and g^s, respectively. Since the session key m is chosen randomly from G_1, $m\hat{e}(g^\rho, h_\alpha)^r$ is distributed uniformly over G_1. We set it to a random value $y \in G_1$. Even though we don't know about m, it does not affect the reduction. Other parameters are dependent on what have been discussed. We can check that they are all computed correctly. So, the reduction preserves the right distribution.

If the revoked users compute m from the header with probability ϵ, we can solve the BDH problem with probability $\epsilon_1 = \epsilon$ by computing the following:

$$\begin{aligned} y \cdot m^{-1} \cdot \hat{e}(g^s, (g^r)^\kappa)^{-1} &= \hat{e}(g^\rho, h_\alpha)^r \cdot \hat{e}(g^s, g^{r\kappa})^{-1} \\ &= \hat{e}(g^s, g^{s+\kappa})^r \cdot \hat{e}(g^s, g^{r\kappa})^{-1} \\ &= \hat{e}(g^s, g^s)^r \\ &= \hat{e}(g, h)^{s^2}. \end{aligned} \tag{3}$$

Let t' be the time for this reduction and the solution computation in Equation (3). We can see that t' is polynomially bounded. Thus, if the collusion attack of the revoked users takes $t_1 - t'$ time, we can solve the BDH problem within time t_1.

Since each query takes a constant time, q_H cannot exceed the runtime t. This completes the proof. □

3.3 Traitor tracing

Some authorized users may conspire to construct a decoding box and sells it for profits. There are many ways that a pirate decoder is constructed. For example, a user may simply put its private keys into it. It is also possible that some users put their private keys into the pirate decoder and the decoding algorithm randomly uses one of them for decoding each time.

In our scheme, the private keys of users are all distinct. A private key is associated with a user. We can use the general black-box traitor confirmation algorithm to find the private keys in a confiscated decoding box [19]. The difference is that our scheme uses $\lceil \log N \rceil$ polynomials. We have to run the traitor tracing algorithm for each of the polynomials. The traitor tracing algorithm can find at least one traitor among k traitors in time $O(\binom{n}{k})$ for any $1 \leq k \leq N$.

We could have set the private key, for a particular α, $s_{k,\alpha} = g^{\rho f_\alpha(k)}$ for user U_k and let the header $Hdr(S, m)$ be

$$(\alpha, m\hat{e}(g^\rho, g^{f_\alpha(0)})^r, g^r, (i_1, g^{r f_\alpha(i_1)}), (i_2, g^{r f_\alpha(i_2)}), \ldots, (i_L, g^{r f_\alpha(i_L)})).$$

However, the type of private keys has the problem of key derivation, that is, the private keys $s_{k_1,\alpha}, s_{k_2,\alpha}, \ldots, s_{k_n,\alpha}$ together could produce another private key $s_{k_{n+1},\alpha}$ for $n \geq 2^\alpha + 1$. Thus, the scheme loses the capability of tracing traitors. Nevertheless, this type of private keys does not affect its security against collusion of revoked users.

4 The BE-PI scheme with IND-CCA security

In Theorem 1, we show that the session key in the header is one-way secure against any collusion of revoked users. There are some standard techniques that transfer one-wayness security to indistinguishability security against the adaptive chosen ciphertext attack. Here we present such a scheme Π' based on the technique in [9]. The modification is as follows.

- In the **Setup** algorithm, the system dealer selects another symmetric encryption scheme $\mathcal{E} : K \times G_q \to G_q$, where K is the key space. The symmetric encryption $\mathcal{E}$ is Find-Guess (FG) secure, which is the counterpart of the IND-security for asymmetric encryption. The system dealer also chooses two additional hash functions $H_1 : \{0,1\}^* \to Z_q$ and $H_2 : G_q \to K$. The system dealer incorporates $\mathcal{E}$, H_1 and H_2 into the public key PK.

- In the **Enc** algorithm,

$$Hdr(S,m) = (g^r, \sigma\hat{e}(g^\rho, h_\alpha)^r, \mathcal{E}_{H_2(\sigma)}(m), (i_1, g^{rf_\alpha(i_1)}), (i_2, g^{rf_\alpha(i_2)}), \ldots, (i_L, g^{rf_\alpha(i_L)})),$$

  where σ is randomly chosen from G_q and $r = H_1(\sigma\|m)$.

- In the **Dec** algorithm, we first compute $\bar{\sigma}$ as described in the BE-PI scheme. Then, we compute the session key $\bar{m}$ from $E_{H_2(\sigma)}(m)$ by using $\bar{\sigma}$. We check whether $\sigma\hat{e}(g^\rho, h_\alpha)^r = \bar{\sigma}\hat{e}(g^\rho, h_\alpha)^{H_1(\bar{\sigma}\|\bar{m})}$. If they are equal, $\bar{m}$ is outputted. Otherwise, $\perp$ is outputted.

Before applying the result of Theorem 12 in [9], we need to show that (m, Hdr) of Π is γ-uniform. This is easy to check since for any PK and $(m, y) \in G_1^2$, $\Pr[Hdr(S,m) = y] = 1/q \simeq 2^{-z}$, where z is the security parameter. Thus, the encryption part Hdr for the session key m is 2^{-z}-uniform.

Let q_{H_1}, q_{H_2} and q_D be the numbers of queries to H_1, H_2 and the decryption oracle, respectively. Recall that t' and q_H are described in Theorem 1.

Theorem 2. *Assume that the BDH problem is (t_1, ϵ_1)-hard and the symmetric encryption $\mathcal{E}$ is (t_2, ϵ_2) FG-secure. The scheme Π' is $(t, \epsilon, q_H, q_{H_1}, q_{H_2}, q_D)$ IND-CCA secure under the random oracle model, where*

$$t = \min\{t_1 - t', t_2\} - O(2z(q_{H_1} + q_{H_2})) \text{ and}$$
$$\epsilon = (1 + 2(q_{H_1} + q_{H_2})\epsilon_1 + \epsilon_2)(1 - 2\epsilon_1 - 2\epsilon_2 - 2^{-z+1})^{-q_D} - 1.$$

5 A public-key SD scheme

In the paradigm of subset cover [16] for broadcast encryption, the system chooses a collection $\mathcal{C}$ of subsets $S_1, S_2, \ldots, S_w \subseteq \mathcal{U}$. Each subset $S_j \in \mathcal{C}$ is assigned a *private key* K_j such that every user in S_j holds K_j. For a target set S of receivers, we find a subset cover $S_{i_1}, S_{i_2}, \ldots, S_{i_l}$ with $S = \bigcup_j S_{i_j}$ and use the private keys associated with the cover to encrypt the session key. Then, the header size is linearly proportional to maximum l.

Since a user may belong to a lot of subsets in the collection, he may hold a lot of private keys. In order to save the storage cost for private keys, it is preferable that these keys have some relations, for example, some of them can derive some others. A subset-cover based broadcast encryption scheme plays the art of choosing a collection $\mathcal{C}$ of subsets, assigning subset keys and finding subset covers.

5.1 The PK-SD-PI scheme

We now present our PK-SD-PI scheme, which is constructed by using the polynomial interpolation technique on the collection of subsets in [16]. The system setup is similar to that of the BE-PI scheme. Consider a complete binary tree T of $\lceil \log N \rceil + 1$ levels. The nodes in T are numbered differently. Each user in $\mathcal{U}$ is associated with a different leaf node in T. We call a complete subtree rooted at node i as "subtree T_i". For each subtree T_i of η levels (level 1 to level η from top to bottom), we define the degree-1 polynomials

$$f_j^{(i)}(x) = a_{j,1}^{(i)} x + a_{j,0}^{(i)} \pmod{q},$$

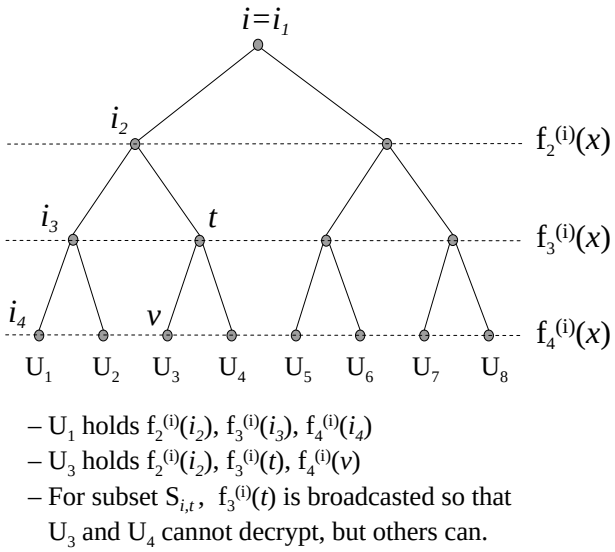

Figure 1: Level polynomials, private keys and broadcasted shares for subtree T_i.

where $a_{j,0}^{(i)} = \lg H(\text{ID}\|\text{"}sd\text{"}\|i\|j\|0)$ and $a_{j,1}^{(i)} = \lg H(\text{ID}\|\text{"}sd\text{"}\|i\|j\|1)$, $2 \le j \le \eta$. For a user U_k in the subtree T_i of η levels, he is given the private keys

$$s_{k,i,j} = (g^{r_{k,i,j}}, g^{r_{k,i,j} f_j^{(i)}(i_j)}, g^{r_{k,i,j} f_j^{(i)}(0)} h^{\rho})$$

for $2 \le j \le \eta$, where nodes $i_1, i_2, \ldots, i_\eta$ are the nodes in the path from node i to the leaf node for U_k (including both ends). We can read $s_{k,i.j}$ as the private key of U_k for the jth level of subtree T_i. In Figure 1, the private keys (in the unmasked form) of U_1 and U_3 for subtree T_i with $\eta = 4$ are given. Here, we use h^ρ in all private keys in order to save space in the header. If h_i^ρ is used for each subtree T_i, the header size is double, but still in $O(r)$.

Recall that in the SD scheme, the collection $\mathcal{C}$ of subsets is

$$\{S_{i,t} : \text{ node } i \text{ is a parent of node } t, i \ne t\},$$

where $S_{i,t}$ denotes the set of users in subtree T_i, but not in subtree T_t. By our design, if the header contains a masked share for $f_j^{(i)}(t)$, where node t is in the j-th level of subtree T_i, only user U_k in $S_{i,t}$ can decrypt the header by using his private key $s_{k,i,j}$, that is, the masked form of $f_j^{(i)}(s)$, for some $s \ne t$. In Figure 1, the share $f_3^{(i)}(t)$ is broadcasted so that only the users in $S_{i,t}$ can decrypt the header.

For a set R of revoked users, let S_{i_1,t_1}, $S_{i_2,t_2}, \ldots, S_{i_z,t_z}$ be a subset cover for $\mathcal{U}\backslash R$, the header is like

$$(m\hat{e}(g^\rho, h)^r, g^r, (i_1, t_1, g^{r f_{j_1}^{(i_1)}(t_1)}), \ldots, (i_z, t_z, g^{r f_{j_z}^{(i_z)}(t_z)})),$$

where node t_k is in the j_k-th level of subtree T_{i_k}, $1 \le k \le z$.

For decryption, a non-revoked user finds an appropriate subset S_{i_j,t_j} in the header and applies the Lagrange interpolation to compute the session key m.

Performance. The public key is $O(1)$, which is the same as that of the BE-PI scheme. Each user belongs to at most $\lceil \log N \rceil + 1$ subtrees and each subtree has at most $\lceil \log N \rceil + 1$ levels. For the subtree of η levels, the user in the subtree holds $\eta - 1$ private keys. Thus, the total number of shares (private keys) held by each user is $\sum_{i=1}^{\lceil \log N \rceil} i = (\lceil \log N \rceil^2 + \lceil \log N \rceil)/2$, which is $O(\log^2 N)$. According to [16], the number z of subsets in a subset cover is at most $2|R| - 1$, which is $O(r)$

When the header streams in, a non-revoked user U_k needs to find his containing subset S_{i_j,t_j}, $U_k \in S_{i_j,t_j}$. With a proper numbering of the nodes in T, this can be done very fast, for example, in $O(\log \log N)$ time. Without considering the time of scanning the header to find his containing subset, each user needs to perform 2 modular exponentiations and 3 pairing functions. Thus, the decryption cost is $O(1)$.

Security. The polynomials associated with the subtrees are all distinct and independent (assuming the random oracle model). For each such polynomial used in the header, the revoked users have at most one of its shares. By the argument of Theorem 1, the revoked users together cannot compute the session key m. Similarly, we can make our PK-SD-PI scheme have the IND-CCA security like Section 4

5.2 The PK-LSD-PI scheme

We can construct the PK-LSD-PI scheme in the same way since the collection $\mathcal{C}$ in the LSD scheme is a subset of that of the SD scheme. The numbers of public and private keys are $O(1)$ and $O(\log^{1+\epsilon})$, respectively, for any constant $0 < \epsilon < 1$. The header size is $O(r/\epsilon)$, which is $O(r)$ for constant ϵ. The decryption cost is again $O(1)$.

6 Conclusion

We have presented two very efficient public-key BE schemes. They have low public and private keys. One of them even has a constant decryption time. Our results show that the efficiency of public-key BE schemes is comparable to that of private-key BE schemes.

We are interested in reducing the ciphertext size while keeping other complexities low in the future.

References

[1] N. Attrapadung, H. Imai. Graph-decomposition-based frameworks for subset-cover broadcast encryption and efficient instantiations.

[2] D. Boneh, X. Boyen, E.-J. Goh. Hierarchical identity based encryption with constant size ciphertext. In *Proceedings of Advances in Cryptology - Eruocrypt 05*, Lecture Notes in Computer Science 3494, pp.440-456, Springer, 2005.

[3] D. Boneh, M. Franklin. An efficient public key traitor tracing scheme. In *Proceedings of Advances in Cryptology - Crypto 99*, Lecture Notes in Computer Science 1666, pp.338-353, Springer, 1999.

[4] D. Boneh, C. Gentry, B. Waters. Collusion resistant broadcast encryption with short ciphertexts and private keys. In *Proceedings of Advances in Cryptology - Crypto 05*, Lecture Notes in Computer Science 3621, pp.258-275, Springer, 2005.

[5] D. Boneh, B. Waters. A fully collusion resistant broadcast, trace, and revoke system. In *Proceedings of the ACM Conference on Computer and Communications Security - CCS 06*, pp.211-220, ACM Press, 2006.

[6] Y. Dodis, N. Fazio. Public key broadcast encryption for stateless receivers. In *Proceedings of Digital Right Management 02 - DRM 02*, Lecture Notes in Computer Science 2696, pp.61-80, Springer, 2002.

[7] Y. Dodis, N. Fazio. Public key broadcast encryption secure against adaptive chosen ciphertext attack. In *Proceedings of Public Key Cryptography - PKC 03*, Lecture Notes in Computer Science 2567, pp.100-115, Springer, 2003.

[8] A. Fiat, M. Naor. Broadcast encryption. In *Proceedings of Advances in Cryptology - Crypto 93*, Lecture Notes in Computer Science 773, pp.480-491, Springer, 1993.

[9] E. Fujisaki, T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Proceedings of Advances in Cryptology - Crypto 99*, Lecture Notes in Computer Science 1666, pp.537-554, Springer, 1999.

[10] C. Gentry, A. Silverberg. Hierarchical ID-based cryptography. In *Proceedings of Advances in Cryptology - Asiacrypt 02*, Lecture Notes in Computer Science 2501, pp.548-566, Springer, 2002.

[11] M.T. Goodrich, J.Z. Sun, R. Tamassia. Efficient Tree-Based Revocation in Groups of Low-State Devices. In *Proceedings of Advances in Cryptology - Crypto 04*, Lecture Notes in Computer Science 3152, pp.511-527, Springer, 2004.

[12] D. Halevy, A. Shamir. The LSD broadcast encryption scheme. In *Proceedings of Advances in Cryptology - Crypto 02*, Lecture Notes in Computer Science 2442, pp.47-60, Springer, 2002.

[13] K. Kurosawa, Y. Desmedt. Optimum traitor tracing and asymmetric schemes. In *Proceedings of Advances in Cryptology - Eurocrypt 98*, Lecture Notes in Computer Science 1403, pp.145-157, Springer, 1998.

[14] K. Kurosawa, T. Yoshida. Linear code implies public-key traitor tracing. In *Proceedings of Public Key Cryptography*, Lecture Notes in Computer Science 2274, pp.172-187, Springer, 2002.

[15] J.W. Lee, Y.H. Hwang, P.J. Lee. Efficient public key broadcast encryption using identifier of receivers. In *Proceedings of International Conference on Information Security Practice and Experience - ISPEC 06*, Lecture Notes in Computer Science 3903, pp.153-164, Springer, 2006.

[16] D. Naor, M. Naor, J. Lotspiech. Revocation and tracing schemes for stateless receivers. In *Proceedings of Advances in Cryptology - Crypto 01*, Lecture Notes in Computer Science 2139, pp.41-62, Springer, 2001.

[17] M. Naor, B. Pinkas. Efficient trace and revoke schemes. In *Proceedings of Financial Cryptography 00*, Lecture Notes in Computer Science 1962, pp.1-20, Springer, 2000.

[18] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11), pp.612-613, 1979.

[19] W.-G. Tzeng, Z.-J. Tzeng. A public-key traitor tracing scheme with revocation using dynamic shares. In *Proceedings of Public Key Cryptography - PKC 01*, Lecture Notes in Computer Science 1992, pp.207-224, Springer, 2001.

[20] P. Wang, P. Ning, D.S. Reeves. Storage-efficient stateless group key revocation. In *Proceedings of the 7th Information Security Conference - ISC 04*, Lecture Notes in Computer Science 3225, pp.25-38, Springer, 2005.

[21] E.S. Yoo, N.-S. Jho, J.J. Cheon, M.-H. Kim. Efficient broadcast encryption using multiple interpolation methods. In *Proceedings of ICISC 04*, Lecture Notes in Computer Science 3506, pp.87-103, Springer, 2005.

[22] M. Yoshida, T. Fujiwara. An efficient traitor tracing scheme for broadcast encryption. In *Proceedings of 2000 IEEE International Symposium on Information Theory*, pp. 463, IEEE Press, 2000.