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
Email: wgtzeng@cs.nctu.edu.tw

Abstract

In this paper we propose two public key BE schemes that have efficient complexity measures. The first scheme, called the PBE-PI scheme, has $O(r)$ header size, $O(1)$ public keys and $O(\log N)$ private keys per user, 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 match those of efficient secret key BE schemes.

Our second scheme, called the PBE-SD-PI scheme, has $O(r)$ header size, $O(1)$ public key and $O(\log N)$ private keys per user also. However, its 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. Overall, this is the most efficient public key BE scheme up to now.

Our basic schemes are one-way secure against *full collusion of revoked users* in the random oracle model under the BDH assumption. We modify our schemes to have indistinguishably security against adaptive chosen ciphertext attacks.

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 them 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 goal 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 pay-TV systems, encrypted file sharing systems, digital right management, content protection of recordable data, etc.

A broadcasted message M is 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

*Version 2 (March 12, 2008). Research supported in part by NSC projects 96-2628-E-009-011-MY3, 96-3114-P-001-002-Y (iCAST), and 96-2219-E-009-013 (TWISC).

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 stateless or stateful. For a stateful 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 stateless 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 BE scheme, any one (broadcaster) can broadcast a message to an arbitrary group of authorized users by using the public parameters of the system, while for a secret key broadcast encryption scheme, only the special dealer, who knows the system secrets, can broadcast a message.

In this paper we refer "stateless 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 PBE-PI scheme (broadcast encryption with polynomial interpolation), has $O(r)$ header size, $O(1)$ public keys, and $O(\log N)$ private keys per user[1], 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 match those of efficient secret key BE schemes [11, 20, 21]. The idea 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. Nevertheless, if we implement the $\log N$ copies straightforwardly, we would get a scheme of $O(N)$ public keys. 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 PBE-SD-PI scheme (public key SD broadcast encryption with polynomial interpolation), is constructed by combining the polynomial interpolation technique and the subset cover method in the SD scheme [16]. The PBE-SD-PI scheme has $O(r)$ header size, $O(1)$ public key and $O(\log N)$ private keys per user. They are the same as those of the PBE-PI 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.

Our basic schemes are one-way secure against *full collusion of revoked users* in the random oracle model under the BDH assumption. We modify our schemes to have indistinguishably security against adaptive chosen ciphertext attacks. The comparison with some other public key BE schemes with full collusion resistance is shown in Table 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

[1] log is based on 2 if the base is not specified.

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

	header size	public-key size	private-key size	decryption cost$^\natural$
PBE-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: PBE-PI	$O(r)$	$O(1)$	$O(\log N)$	$O(r)$
Our work: PBE-SD-PI	$O(r)$	$O(1)$	$O(\log N)$	$O(1)$

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

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. 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 N) of polynomials. They proposed 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. 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 achieves $O(1)$ public key, $O(\log N)$ private keys, but, $O(r \log N)$ header size. Boneh and Waters [5] proposed a scheme that has the traitor tracing capability. 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 BE scheme into a public key one. For example, Dodis and Fazio [6] transformed the SD and LSD schemes [12, 16] into public key SD and LSD schemes, shorted as PBE-SD and PBE-LSD. The transformation employs the technique of hierarchical identity-based encryption to substitute for the hash function. Instantiated with the newest constant-size hierarchical identity-based encryption [2], the PBE-SD scheme has $O(r)$ header size, $O(1)$ public keys and $O(\log^2 N)$ private keys. The PBE-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 PBE-SD and PBE-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 BE 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 and G_1 be two (multiplicative) cyclic groups of prime order q and $\hat{e}$ be a bilinear map from $G \times G$ to G_1. Then, $\hat{e}$ has the following properties.

1. For all $u, v \in G$ 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, $\hat{e}(g, g) = g_1 \neq 1$ is a generator of G_1.

BDH hardness assumption. The BDH problem is to compute $\hat{e}(g, g)^{abc}$ from given (g, g^a, g^b, g^c). 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 $k \geq k_0$,

$$\Pr[A(g, g^a, g^b, g^c) = \hat{e}(g, g)^{abc} : g \stackrel{u}{\leftarrow} G; a, b, c \stackrel{u}{\leftarrow} Z_q] \leq \epsilon.$$

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 and 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 $U_k \in S$, it computes the session key m and then uses m to decrypt C for the message M. If $U_k \notin S$, it cannot decrypt the ciphertext.

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

Security. We describe the indistinguishability security against adaptive chosen ciphertext attacks (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 omitted. 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 target set $S^* \subseteq \mathcal{U}$ of users 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 to $\mathcal{A}$, where $U_i \notin S^*$.

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

Challenge. The challenger $\mathcal{C}$ runs $Enc^*(PK, S^*)$ and outputs $y = Hdr(S^*, m)$, 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 $(m_0, m_1, Hdr(S^*, m))$ to $\mathcal{A}$.

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

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

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

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

where $SK_{\mathcal{U}} = \{SK_i : 1 \leq i \leq N\}$ 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* $Adv_{\mathcal{A},\Pi}^{\textit{ind-cca}}(z) < \epsilon$.

In this paper we first give schemes with one-way security against chosen plaintext attacks (OW-CPA security) and then transform them to have IND-CCA security via the Fujisaki-Okamoto transformation [9]. The OW-CPA security is defined as follows.

Init. The adversary $\mathcal{A}$ chooses a system identity $\textsc{Id}$ and a target set $S^* \subseteq \mathcal{U}$ of users to attack.

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

Challenge. The challenger $\mathcal{C}$ runs $Enc^*(PK, S^*)$ and outputs $Hdr(S^*,\ m)$, where m is randomly chosen.

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

Since $\mathcal{A}$ can always encrypt a chosen plaintext by himself, the oracle of encrypting a chosen plaintext does not matter in the definition. Let $\mathrm{Adv}_{\mathcal{A},\Pi}^{\textit{ow-cpa}}(z)$ be the advantage that $\mathcal{A}$ wins the above game, that is,

$$\begin{aligned}\mathrm{Adv}_{\mathcal{A},\Pi}^{\textit{ow-cpa}}(z) = \Pr[&\mathcal{A}(PK, SK_{\mathcal{U}\setminus S^*}, Hdr(S^*, m)) = m : S^* \subseteq \mathcal{U}, \\ &(PK, SK_{\mathcal{U}}) \leftarrow Setup(1^z, \textsc{Id}, \mathcal{U}), Hdr(S^*, m) \leftarrow Enc^*(PK, S^*)].\end{aligned}$$

Definition 2 *A public key BE scheme* Π*=(Setup, Enc, Dec) is* (t, ϵ)*-OW-CPA secure if for all* t*-time bounded adversary* $\mathcal{A}$*, we have* $Adv_{\mathcal{A},\Pi}^{\textit{ow-cpa}}(z) < \epsilon$.

3 The PBE-PI Scheme

Let G and G_1 be the bilinear groups with the pairing function $\hat{e}$, where q is a large prime. Let $H_1, H_2 : \{0,1\}^* \rightarrow G_1$ be two hash functions and E be a symmetric encryption with key space G_1.

The idea of our construction is as follows. For a polynomial $f(x)$ of degree t, we assign each user U_i a share $f(i)$. The secret is $f(0)$. We can compute the secret $f(0)$ from any $t+1$ shares. If we want to revoke t users, we broadcast their shares. Any non-revoked user can compute the secret $f(0)$ from his own share and the broadcasted ones, totally $t+1$ shares. On the other hand, any collusion of revoked users cannot compute the secret $f(0)$ since they have t shares only, including the broadcasted ones. If less than t users are revoked, we broadcast the shares of some dummy users such that t shares are broadcasted totally. In order to achieve $O(r)$ ciphertexts, we use $\log N$ polynomials, each for a range of the number of revoked users.

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. Wlog, let N be a power of 2. Then, the system dealer does the following:

   - Choose a generator g of group G, and let $\lg = \log_g$ and $g_1 = \hat{e}(g, g)$.
   - Compute $h_i = H_1(\text{ID}\|i)$ for $1 \le i \le \log N$.
   - Compute $g^{a_j^{(i)}} = H_2(\text{ID}\|i\|j)$ for $0 \le i \le \log N$ and $0 \le j \le 2^i$.
     <u>*Remark*</u>. The underlying polynomials are, $0 \le i \le \log N$,
     $$f_i(x) = \sum_{j=0}^{2^i} a_j^{(i)} x^j \pmod q.$$
     The system dealer does not know the coefficients $a_j^{(i)} = \lg H_2(\text{ID}\|i\|j)$. But, this does not matter.
   - Randomly choose a secret $\rho \in Z_q$ and compute g^ρ.
   - Publish the public key $PK = (\text{ID}, H_1, H_2, E, G, G_1, \hat{e}, g, g^\rho)$.
   - Assign a set $SK_k = \{s_{k,0}, s_{k,1}, \ldots, s_{k,\log N}\}$ of private keys to user U_k, $1 \le k \le N$, 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, $1 \le i \le \log N$.

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 $l \ge 1$. M is the sent message. The broadcaster does the following:

   - Let $\alpha = \lceil \log l \rceil$ and $L = 2^\alpha$.
   - Compute $h_\alpha = H_1(\text{ID}\|\alpha)$.
   - Randomly select distinct $i_{l+1}, i_{l+2}, \ldots, i_L > N$. These $U_{i_t}, l+1 \le t \le L$, are dummy users.
   - Randomly select a session key $m \in G_1$.
   - Randomly select $r \in Z_q$ and compute, $1 \le t \le L$,
     $$g^{r f_\alpha(i_t)} = (\prod_{j=0}^{L} H_2(\text{ID}\|\alpha\|j)^{i_t^j})^r.$$
   - 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 $b_j = \hat{e}(g^{r_{k,\alpha}}, g^{r f_\alpha(i_j)}) = g_1^{r r_{k,\alpha} f_\alpha(i_j)}$, $1 \le j \le L$.
   - 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

$$\frac{m\hat{e}(g^{\rho}, h_{\alpha})^{r} \cdot g_1^{rr_{k,\alpha}f_{\alpha}(0)}}{\hat{e}(g^{r}, g^{r_{k,\alpha}f_{\alpha}(0)}h_{\alpha}^{\rho})} = \frac{m\hat{e}(g^{\rho}, h_{\alpha})^{r} \cdot g_1^{rr_{k,\alpha}f_{\alpha}(0)}}{\hat{e}(g^{r}, h_{\alpha}^{\rho}) \cdot g_1^{rr_{k,\alpha}f_{\alpha}(0)}} = m. \qquad (2)$$

- 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 $(\textsc{Id}, H_1, H_2, E, G, G_1, \hat{e}, g, g^{\rho})$, which is of size $O(1)$. Since all systems can use the same $(H, E, G, G_1, \hat{e}, g)$, the public key specific to a system is simply $(\textsc{Id}, g^{\rho})$. Each system dealer has a secret ρ for assigning private keys to its users. Each user U_k holds private keys $SK_k = \{s_{k,0}, s_{k,1}, \ldots, s_{k,\log N}\}$, each corresponding to a share of polynomial f_i in the masked form, $0 \le i \le \log N$. 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 $2^{\alpha-1} < r \le 2^{\alpha}$. Thus, the header size is $O(2^{\alpha}) = O(r)$. It is actually no more than $2r$.

To prepare a header, the broadcaster needs to compute one pairing function, $2^{\alpha} + 2$ hash functions, 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

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

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

$$\tilde{\lambda}_j = \frac{(-i_1)\cdots(-i_{j-1})(-i_{j+1})\cdots(-i_L)}{(i_j - i_1)\cdots(i_j - i_{j-1})(i_j - i_{j+1})\cdots(i_j - i_L)} \bmod q, 1 \le j \le L.$$

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 it has OW-CPA security in the random oracle model under the BDH assumption.

Theorem 1 *Assume that the BDH problem is (t_1, ϵ_1)-hard. Our PBE-PI scheme is $(t_1 - t', \epsilon_1)$-OW-CPA secure in the random oracle model, where t' is some polynomially bounded time.*

Proof 1 *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. Wlog, let $R = \{U_1, U_2, \ldots, U_L\}$ be the set of revoked users and the target set of attack be $S^* = \mathcal{U} \backslash R$. Note that S^* was chosen by the adversary in the* **Init** *stage. Let the input of the BDH problem be (g, g^a, g^b, g^c), where the pairing function is implicitly known. We set the system parameters 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 $g^\rho = g^a$.*

(c) *The public key is $(\mathrm{ID}, H_1, H_2, E, G, G_1, \hat{e}, g, g^a)$.*

(d) *The following is implicitly computed.*

- *Set $f_\alpha(i) = w_i, 1 \le i \le L$.*
- *Let $g^{a_0^{(\alpha)}} = g^{f_\alpha(0)} = g^a \cdot g^\tau = g^{a+\tau}$.*
- *Compute $g^{a_i^{(\alpha)}}$, $1 \le i \le L$, from $g^{a_0^{(\alpha)}}$ and $g^{f_\alpha(j)} = g^{w_j}, 1 \le j \le L$, by the Lagrange interpolation method over exponents.*
- *Set $h_\alpha = g^b \cdot g^\kappa = g^{b+\kappa}$.*
- *For $j \ne \alpha$, choose a random polynomial $f_j(x)$ and set $h_j = g^{z_j}$, where z_j is randomly chosen from Z_q.*

3. *Set the secret keys ($g^{r_{i,j}}$, $g^{r_{i,j}f_j(i)}$, $g^{r_{i,j}f_j(0)}h_j^\rho$), $0 \le j \le \log N$, of the revoked user $U_i, 1 \le i \le L$, as follows:*

(a) *For $j = \alpha$, let $g^{r_{i,\alpha}} = g^{-b+\mu_i}$, $g^{r_{i,\alpha}f_\alpha(i)} = (g^{r_{i,\alpha}})^{w_i}$, and $g^{r_{i,\alpha}f_\alpha(0)}h_\alpha^\rho = g^{(-b+\mu_i)(a+\tau)}(g^{b+\kappa})^a = g^{a(\mu_i+\kappa)-b\tau+\mu_i\tau}$.*

(b) *For $j \ne \alpha$, randomly choose $r_{i,j} \in Z_q$ and compute $g^{r_{i,j}}$, $g^{r_{i,j}f_j(i)}$ and $g^{r_{i,j}f_j(0)}h_j^\rho = g^{r_{i,j}f_j(0)}(g^a)^{z_j}$.*

4. *Set the header (α, $m\hat{e}(g^\rho, h_\alpha)^r$, g^r, $(1, g^{rf_\alpha(1)})$, $(2, g^{rf_\alpha(2)})$, …, $(L, g^{rf_\alpha(L)})$) as follows:*

(a) *Let $g^r = g^c$.*

(b) *Compute $g^{rf_\alpha(i)} = (g^c)^{w_i}, 1 \le i \le L$.*

(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 H_1 and H_2 hash oracles. If the query is of the form $H_2(\mathrm{ID}\|i\|j)$ or $H_1(\mathrm{ID}\|i)$, we set them to be $g^{a_j^{(i)}}$ and h_i, 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.

We should check whether the distributions of the parameters in our reduction and those in the system are equal. We only check those related to α since the others are correctly distributed. Since $\tau, w_1, w_2, \dots, w_L$ are randomly chosen, $g^{a_i^{(\alpha)}}, 0 \le i \le L$ are uniformly distributed over G^{L+1}. Due to the random oracle model, their corresponding system parameters are also uniformly distributed over G^{L+1}. Since $\kappa, \mu_1, \mu_2, \dots, \mu_L$ are randomly chosen, the distribution of h_α and $g^{r_{i,\alpha}}, 1 \le i \le L$, are uniform over G^{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 since they are set from the given input g^c and g^a, 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 the same probability $\epsilon_1 = \epsilon$ by computing the following:

$$\begin{aligned} y \cdot m^{-1} \cdot \hat{e}(g^a, g^c)^{-\kappa} &= \hat{e}(g^\rho, h_\alpha)^r \cdot \hat{e}(g, g)^{-ac\kappa} \\ &= \hat{e}(g^a, g^{b+\kappa})^c \cdot \hat{e}(g, g)^{-ac\kappa} \\ &= \hat{e}(g, g)^{abc}. \end{aligned} \tag{3}$$

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.

4 The PBE-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 of transforming OW-CPA security to IND-CCA security. Here we present such a scheme Π' based on the technique in [9].

The IND-CCA security of the Fujisaki-Okamoto transformation depends only on the OW-CPA security of the public key encryption scheme, the FG security of a symmetric encryption scheme $\mathcal{E}$, and the γ-uniformity of the public key encryption scheme. The FG-security is the counterpart of the IND-security for symmetric encryption. A public key encryption scheme is γ-uniform if for every key pair (pk, sk), every message x, and $y \in \{0,1\}^*$, $\Pr[E_{pk}(x) = y] \leq \gamma$. Before applying the transformation, we check the following things:

1. The transformation applies to public key encryption, while ours is public key broadcast encryption. Nevertheless, if the authorized set S is fixed, our public key broadcast encryption scheme is a public key encryption scheme with public key $pk = (PK, S)$. In the definition of IND-CCA security (Definition 1), the adversary $\mathcal{A}$ selects a target set S^* of users to attack in the **Init** stage and S^* is fixed through the rest of the attack. Thus, we can discuss the attack of $\mathcal{A}$ with a fixed target set S^*. Note that $\mathcal{A}$ is a static adversary.

2. Let S be a fixed authorized set of users. For every m and every $y \in \{0,1\}^*$, $\Pr[Hdr(S,m) = y]$ is either 0 or $1/q \simeq 1/2^z$, where z is the security parameter (the public key size). Thus, our broadcast encryption scheme is 2^{-z}-uniform if the authorized set is fixed.

Let $\mathcal{E} : K \times G_1 \to G_1$ be a symmetric encryption scheme with FG-security, where K is the key space of $\mathcal{E}$. Let $H_3 : G_1 \times G_1 \to Z_q$ and $H_4 : G_1 \to K$ be two hash functions. The modification of Π for Π' is as follows.

- In the **Setup** algorithm, add $\mathcal{E}, H_3, H_4$ to PK.

- In the **Enc** algorithm,

$$\begin{aligned} Hdr(S,m) = (&g^r, \sigma\hat{e}(g^\rho, h_\alpha)^r, \mathcal{E}_{H_4(\sigma)}(m), \\ &(i_1, g^{rf_\alpha(i_1)}), (i_2, g^{rf_\alpha(i_2)}), \ldots, (i_L, g^{rf_\alpha(i_L)})), \end{aligned}$$

  where σ is randomly chosen from G_1 and $r = H_3(\sigma, m)$.

- In the **Dec** algorithm, we first compute $\bar{\sigma}$ as described in the PBE-PI scheme. Then, we compute the session key $\bar{m}$ from $\mathcal{E}_{H_4(\sigma)}(m)$ by using $\bar{\sigma}$. We check whether $\sigma\hat{e}(g^\rho, h_\alpha)^r = \bar{\sigma}\hat{e}(g^\rho, h_\alpha)^{H_3(\bar{\sigma},\bar{m})}$ and $g^{rf_\alpha(i_j)} = g^{f_\alpha(i_j)H_3(\bar{\sigma},\bar{m})}$, $1 \leq j \leq L$. If they are all equal, $\bar{m}$ is outputted. Otherwise, $\perp$ is outputted.

Let q_{H_3}, q_{H_4} and q_D be the numbers of queries to H_3, H_4 and the decryption oracles, respectively. Our scheme Π' is IND-CCA-secure.

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 $(t, \epsilon, q_{H_3}, q_{H_4}, q_D)$-IND-CCA secure in the random oracle model, where t' is some polynomially bounded time,*

$$\begin{aligned} &t = \min\{t_1 - t', t_2\} - O(2z(q_{H_3} + q_{H_4})) \quad \textit{and} \\ &\epsilon = (1 + 2(q_{H_3} + q_{H_4})\epsilon_1 + \epsilon_2)(1 - 2\epsilon_1 - 2\epsilon_2 - 2^{-z+1})^{-q_D} - 1. \end{aligned}$$

This theorem is proved by showing that if Π' is not IND-CCA-secure, then either Π is not OW-CPA-secure or $\mathcal{E}$ is not FG-secure directly. The OW-CPA security of Π is based on the BDH assumption. We note that the application of the transformation to other types of schemes could be delicate. Galindo [10] pointed out such a case. Nevertheless, the problem occurs in the proof and is fixable without changing the transformation or the assumption. The detailed proof will be given in the full version of the paper.

5 A Public Key SD Scheme

In the paradigm of subset cover for broadcast encryption [16], the system chooses a collection $\mathcal{C}$ of subsets of users such that each set S of users can be covered by the subsets in $\mathcal{C}$, that is, $S = \cup_{i=1}^{w} S_w$, where $S_i \in \mathcal{C}$ are disjoint, $1 \le i \le w$. Each subset S_i in $\mathcal{C}$ is associated with a private key k_i. A user is assigned a set of keys such that he can derive the private keys of the subsets to which he belongs. The subset keys k_i cannot be independent. Otherwise, each user may hold too many keys. It is preferable that the subset keys have some relations, for example, one can be derived from another. Thus, each user U_k is given a set SK_k of keys so that he can derive the private key of a subset to which he belongs. A subset-cover based broadcast encryption scheme plays the art of choosing a collection $\mathcal{C}$ of subsets, assigning subset and user keys, and finding subset covers.

5.1 Basic PBE-SD-PI Scheme

We now present our PBE-SD-PI scheme, which is constructed by using the polynomial interpolation technique on the collection of subsets in [16]. We first give such a scheme with $O(\log^2 N)$ private keys and then show how to get the one with $O(\log N)$ private key.

The system setup is similar to that of the PBE-PI scheme. Consider a complete binary tree T of $\log N + 1$ levels. The nodes in T are numbered differently. Each user in $\mathcal{U}$ is associated with a different leaf node in T. We refer to 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

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

where $a_{j,0}^{(i)} = \lg H_2(\textsc{Id}\|i\|j\|0)$ and $a_{j,1}^{(i)} = \lg H_2(\textsc{Id}\|i\|j\|1)$, $2 \le j \le \eta$. For a user U_k in the subtree T_i of η levels, he is given the private keys

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

for $2 \le j \le \eta$, where $h_{i,j} = H_1(\textsc{Id}\|i\|j)$ and 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.

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.

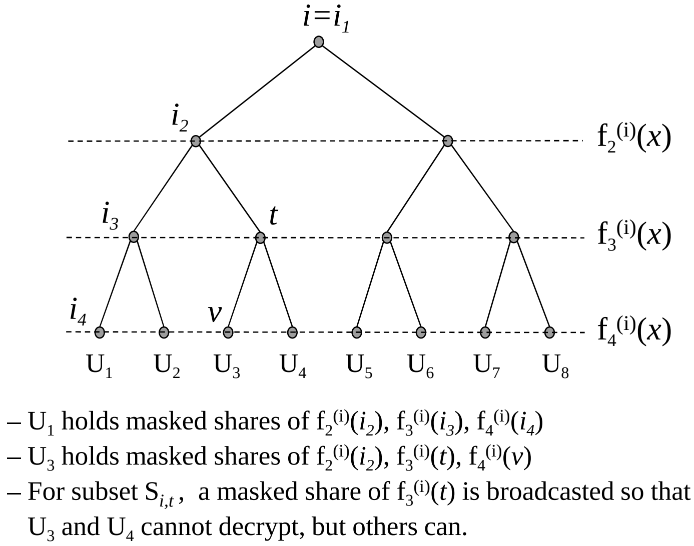

Figure 1: Level polynomials, private keys and broadcasted shares for subtree T_i.

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

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

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

For decryption, a non-revoked user finds $i_k, t_k, g^{rf_{j_k}^{(i_k)}(t_k)}$ (corresponding to S_{i_k,t_k} where he is in) from 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 PBE-PI scheme. Each user belongs to at most $\log N + 1$ subtrees and each subtree has at most $\log N + 1$ 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 $\sum_{i=1}^{\log N} i = O(\log^2 N)$. 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 looks for his containing subset S_{i_j,t_j} to which he belongs. 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 out his containing subset, each user needs to perform 2 modular exponentiations and 3 pairing functions. Thus, the decryption cost is $O(1)$.

Security. We first show that the scheme is one-way secure.

Theorem 3 *Assume that the BDH problem is (t_1, ϵ_1)-hard. Our PBE-SD-PI scheme is $(t_1 - t', \epsilon_1)$-OW-CPA secure in the random oracle model, where t' is some polynomially bounded time.*

Proof 2 *The one-way security proof for the PBE-SD-PI scheme is similar to that for the PBE-PI scheme. In the PBE-SD-PI scheme, all polynomials $f_j^{(i)}(x)$ are of degree one. Let (g, g^a, g^b, g^c) be the input to the BDH problem. Let $S_{i_1,t_1}, S_{i_2,t_2}, \ldots, S_{i_z,t_z}$ be a subset cover for $S^* = \mathcal{U}\backslash R$. Due to the random oracle assumption for H_1 and H_2, all polynomials are independent. Thus, we can simply consider a particular $S_{\alpha,t}$ in the subset cover for $S^* = \mathcal{U}\backslash R$, where t is at level β of subtree T_α. The corresponding polynomial is $f(x) = f_\beta^{(\alpha)}(x) = a_1 x + a_0 \pmod q$. Wlog, let $\{U_1, U_2, \ldots, U_l\}$ be the set of revoked users that have the secret share about $f(t)$. The reduction to the BDH problem is as follows. Recall that the public key of the PBE-SD-PI method is $(\textsc{Id}, H_1, H_2, E, G, G_1, \hat{e}, g, g^{\rho})$.*

1. *Let g be the generator in the system and $g^\rho = g^a$.*

2. *Set $f(t) = w$ and compute $g^{f(t)} = g^w$, where w is randomly chosen from Z_q.*

3. *Let $g^{a_0} = g^{f(0)} = g^a \cdot g^\tau$, where τ is randomly chosen from Z_q.*

4. *Compute g^{a_1} from $g^{f(t)}$ and g^{a_0} via the Lagrange interpolation.*

5. *The (random) hash values $H_2(\textsc{Id}\|\alpha\|\beta\|0)$ and $H_2(\textsc{Id}\|\alpha\|\beta\|1)$ are set as g^{a_0} and g^{a_1} respectively.*

6. *Set $h_{\alpha,\beta} = g^b \cdot g^\kappa$, where κ is randomly chosen from Z_q.*

7. *The $f(x)$-related secret share of $U_i, 1 \le i \le l$, is computed as $(g^{r_i},\ g^{r_i f(t)},\ g^{r_i f(0)} h^\rho_{\alpha,\beta})$, where $g^{r_i} = g^{-b} \cdot g^{\mu_i}$ and μ_i is randomly chosen from Z_q. Note that $g^{r_i f(0)} h^\rho_{\alpha,\beta} = g^{a(\mu_i+\kappa)-b\tau+\mu_i\tau}$ can be computed from the setting in the previous steps.*

8. *The non-$f(x)$-related secret shares of $U_i, 1 \le i \le l$, can be set like those in Theorem 1, where $h_{i,j}$ is set as $g^{z_{i,j}}$ for a randomly chosen $z_{i,j}$.*

9. *Set the challenge as*

$$(y, g^c, (i_1, t_1, g^{cf^{(i_1)}_{j_1}(t_1)}), (i_2, t_2, g^{cf^{(i_2)}_{j_2}(t_2)}), \ldots, (i_z, t_z, g^{cf^{(i_z)}_{j_z}(t_z)})),$$

   where y is randomly chosen from G and thought as $m\hat{e}(g^\rho, h)^c$. Note that $g^{cf^{(i_k)}_{j_k}(t_k)}, 1 \le k \le z$, can be computed since $f^{(i_k)}_{j_k}(t_k)$ is a number randomly chosen from Z_q, as described in Step 2.

If the revoked users $U_1, U_2, \ldots, U_l$ can together compute the session key m from the challenge with probability ϵ_1, we can compute

$$\begin{aligned} y \cdot m^{-1} \cdot \hat{e}(g^a, g^c)^{-\kappa} &= \hat{e}(g^\rho, h_{\alpha,\beta})^c \cdot \hat{e}(g, g)^{-ac\kappa} \\ &= \hat{e}(g^a, g^{b+\kappa})^c \cdot \hat{e}(g, g)^{-ac\kappa} = \hat{e}(g, g)^{abc} \end{aligned} \tag{4}$$

with the same probability ϵ_1. This contradicts the BDH assumption.

Let t' be the time for the reduction and solution computation in Equation (4), where t' is polynomially bounded. Thus, if the collusion attack takes $t_1 - t'$, we can solve the BDH problem in time t_1.

Similarly, we can modify our PBE-SD-PI scheme to have IND-CCA security like Section 4

5.2 Efficient PBE-SD-PI scheme

We now show how to get the scheme with $O(\log N)$ private keys. Let node 0 be the root of T Instead of using independent polynomials $f^{(i)}_j$, we compute those $f^{(i')}_{j'}(x)$ from $f^{(0)}_j(x)$. This could save the stored private keys for each user. We shall give the construction in the final version of the paper.

6 Conclusion

We have presented two very efficient public key BE schemes. Both of them have low public and private keys. One of them even have 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. In *Proceedings of Advances in Cryptology - Asiacrypt 05*, Lecture Notes in Computer Science 3788, pp.100-120, Springer, 2005.

[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] D. Galindo. Boneh-Franklin identity based encryption revisited. In Proceedings of ICALP 05, Lecture Notes in Computer Science 3580, pp.791-802, Springer, 2005.

[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.