Efficient Broadcast Encryption using Multiple Interpolation Methods

Eun Sun Yoo[1], Nam-Su Jho[1], Jung Hee Cheon[1], and Myung-Hwan Kim[1] *

[1]ISaC and Department of Mathematical Sciences, Seoul National University, Seoul 151-747, Korea
{eunsun, drake, jhcheon, mhkim}@math.snu.ac.kr

Abstract. We propose a new broadcast encryption scheme based on polynomial interpolations. Our scheme, obtained from the Naor-Pinkas scheme by partitioning the user set and interpolating multiple polynomials, turns out to be better in efficiency than the best known broadcast schemes like the Subset Difference and the Layered Subset Difference methods, which are tree based schemes. More precisely, when r users are revoked among n users, our method requires $O(\log(n/m))$ user keys and $O(\alpha r + m)$ transmission overhead in the worst case, where m is the number of partitions of the user set and can be chosen to optimize its efficiency, and α is a predetermined constant satisfying $1 < \alpha < 2$. So, our scheme is always better in the storage than the tree based schemes (whose storage overhead is $O(\log^2 n)$ or $O(\log^{3/2} n)$). In the transmission overhead, our scheme beats those schemes except for a very small r/n. The computation cost is worse than the other schemes but is reasonable for systems with moderate computing power. The security proof is given based on the computational Diffie-Hellman problem.

1 Introduction

Broadcast Encryption is a cryptographic method to efficiently broadcast information to a large set of users so that only privileged users can decrypt it. We assume that each user is given a user-key, which is a set of secrets, from the center before starting the broadcast and the user key is never updated, that is, users are *stateless* receivers. In each session, a message is encrypted by a session key and the encrypted message is transmitted over an insecure channel with the encrypted value, called a *header*, of the session key, which can be decrypted only by the user-keys. When a user needs to be removed from the set of the privileged users, the center should be able to make a header to be decrypted only by the non-revoked users. Each session can have an independent set of revoked users. That is, a revoked user in one session may subscribe another session. The application covers pay TV, internet multicast of movies or news, and mobile games. The best known broadcast schemes are the *Subset Difference* method (SD) [**?**] and the *Layered Subset Difference* method (LSD) [**?**], which is a variant

* This author was partially supported by KRF Research Fund(2004-070-100001)

of SD adopting the notion of layers. There is also the *Stratified Subset Difference* method (SSD) [**?**], which was proposed at Crypto'04.

The polynomial interpolation method was first introduced by Berkovits [**?**] and improved by Naor and Pinkas [**?**]. In the improved scheme, called the NP scheme, a user-key is only one field element and the header size is very small. However, since this scheme is originally designed for multicast with traitor tracing rather than broadcast, it has the best performance for small size user sets (of several hundred users, say) and requires key refreshment from time to time. The scheme can also be used for stateless receivers without key refreshment. But in this case, the NP scheme has shortcomings. First, the system cannot revoke more than d users, where d is the degree of a polynomial predetermined in the setup stage. Second, its header size depends on d, not on the number r of revoked users. If we increase d large enough to cover the maximum number of revoked users, the header size increases along with d even if r is small. Third, the computation cost for each user increases in the square order of d, which is too large to be practical even for $d \approx 1000$.

In this paper, we adopt two simple ideas, partitioning the users and interpolating multiple polynomials, to overcome the weaknesses of the NP scheme mentioned above as a broadcast encryption for a large number of stateless receivers. Our scheme covers any number n of users and is more efficient than SD, LSD and SSD in the user-key size and in the header size. More precisely, our method requires $O(\log(n/m))$ key storage and the $O(\alpha r + m)$ header size in the worst case, where m, which can be chosen to optimize efficiency, is the number of partitions of the user set, and α is a predetermined constant satisfying $1 < \alpha < 2$. So, the user-key size is always smaller than the tree based schemes whose key storage is $O(\log^{\beta} n)$ with $1 \leq \beta \leq 2$. Our scheme satisfies, in fact, the *log-key restriction* (the storage size is bounded by $\log n$) [**?**] with much smaller transmission overhead than SSD. The header size is also smaller than those schemes except when r/n is very small. The computation cost is worse than those schemes but can be adjusted according to computing power of the user device.

Outline of this paper is as follows: After a brief preliminaries in Section 2, we propose the basic scheme interpolating multiple polynomials and the extended scheme partitioning the user set, respectively, in Sections 4 and 5. We analyze the performance of our scheme in Section 5, and compare our scheme with SD, LSD and SSD in Section 6. We discuss a security proof of our scheme in Section 7, and conclude in Section 8. We present a detailed proof in Appendix.

2 Preliminaries

We use the following parameters:

n: the number of total users
r: the number of revoked users
I_u: The identifier of a user u
K_u: the set of keys stored by a user u, i.e., the *user-key* of u

A *session* is one broadcast of data to all users and the *session key* k is the key used to encrypt the data in the session. In order to broadcast a message M, the center encrypts M using the session key k and sends the encrypted message together with a *header* to the users. That is, the center sends

$$\langle\ \langle header\rangle,\ E_k(M)\ \rangle$$

to the users, where $E_k(M)$ is a symmetric encryption of M by k. Then, a privileged user u can easily compute k from a pre-defined function F satisfying $F(K_u, \langle header\rangle) = k$. With this k, u can recover M by

$$D_k(E_k(M)) = M.$$

But any revoked user u should not be able to rend k from K_u and $\langle header\rangle$. Furthermore, there should be no efficient polynomial time algorithm O such that

$$O(K_1, K_2, \ldots, K_r, \langle header\rangle) = k,$$

where $K_a = K_{u_a}$ and u_a's are revoked users for $a = 1, 2, \cdots, r$. The fixed function F is pre-distributed before the system starts to operate, and is computable in polynomial time under user level computing power. We call the length of the header the *transmission overhead* or *message overhead*. The computing time of F is called the *computation overhead* or *computation cost*. In a broadcast encryption scheme, the user-key size, called the *storage overhead*, the transmission overhead, and the computation overhead are three most important parameters determining the efficiency of the scheme.

3 Basic scheme

Let n be the number of total users. Given a system parameter α satisfying $1 < \alpha < 2$, define a sequence of positive integers d_i by the recurrence relation:

$$d_1 = 1 \quad \text{and} \quad d_{i+1} = \lfloor \alpha(d_i + 1) \rfloor,$$

where $\lfloor x \rfloor$ denotes the largest integer not exceeding x. Let w be the minimal integer such that $n \leq (\alpha + 1)(d_w + 1) + 1$. Then we take random w polynomials $f_{d_1}, f_{d_2}, \ldots, f_{d_w}$, where f_{d_i} is a polynomial of degree d_i. Let r be the number of revoked users. If $d_{i-1} < r \leq d_i$ for some $1 \leq i \leq w$ (we make a convention that $d_0 = 0$ for convenience), then the polynomial of degree d_i is to be used for revocation. For the case when $r = 0$ or $d_w < r \leq n$, see below.

In the initialization step, the center chooses an elliptic curve E over $\mathbb{F}_p$ and selects a point P in E which is of order q, where p and q are 160 bit primes. (As a matter of fact, any abelian group over which the computational Diffie-Hallman problem is hard will do.) Then it chooses a system parameter α satisfying $1 < \alpha < 2$, and selects a non-secret identifier $I_u (= I) \in \mathbb{Z}_q$ to be given to each user u. Then, the center chooses random polynomials f_{d_i} of degree d_i corresponding to (d_i+1)-out-of-n threshold secret sharing schemes for $i = 1, 2, \ldots, w$, where

$d_{i+1} = \lfloor \alpha(d_i + 1) \rfloor$ over $\mathbb{Z}_q$ for $i = 1, 2, \ldots, w-1$. Each user u receives his/her user-key

$$K_u = \langle\ I, f_{d_1}(I), f_{d_2}(I), \ldots, f_{d_w}(I), \phi, \psi_u\ \rangle$$

and the system information E, P, p and q over a private channel from the center, where ϕ is the key which will be used when there is no revocation and ψ_u is the key which will be used when of $r > d_w$.

Let $u_1, u_2, \ldots, u_r$ be the revoked users, where $d_{i-1} < r \leq d_i$. For revocation, the center first learns the identifiers $I_a = I_{u_a}$ $(a = 1, 2, \ldots, r)$ of the revoked users. For a random $s \in \mathbb{Z}_q$, it takes $k = f_{d_i}(0)sP$ as a new session key that should be unknown to the revoked users. The center chooses distinct $d_i - r$ random points, say $I_{r+1}, I_{r+2}, \ldots, I_{d_i} \in \mathbb{Z}_q$, which are distinct from the identifiers of all users, and broadcasts

$$\langle\ index,\ Q,\ (I_1, f_{d_i}(I_1)Q),\ \ldots,\ (I_r, f_{d_i}(I_r)Q),\ (I_{r+1}, f_{d_i}(I_{r+1})Q),\ \ldots,\ (I_{d_i}, f_{d_i}(I_{d_i})Q)\ \rangle,$$

where $Q = sP$ and $index$ is the indicator of the polynomial used in the session. Each privileged user u can compute $f_{d_i}(I)Q$ by his/her own secret key $f_{d_i}(I)$ and therefore can recover the session key $f_{d_i}(0)Q$. More precisely, the user u can compute $f_{d_i}(0)Q$ from the values

$$(I_0, f_{d_i}(I_0)Q),\ (I_1, f_{d_i}(I_1)Q),\ \ldots,\ (I_r, f_{d_i}(I_r)Q),\ (I_{r+1}, f_{d_i}(I_{r+1})Q),\ \ldots,\ (I_{d_i}, f_{d_i}(I_{d_i})Q),$$

where $I_0 = I$, as follows:

$$f_{d_i}(0)Q = \left(\sum_{a=0}^{d_i} \lambda_a f_{d_i}(I_a) \right) Q = \sum_{a=0}^{d_i} \lambda_a \left(f_{d_i}(I_a)Q \right),$$

where

$$\lambda_a = \prod_{b \neq a} \frac{I_b}{I_b - I_a}. \tag{1}$$

If there is no revoked user, i.e., $r = 0$, then the center encrypts the session key k by symmetric encryption scheme with the key ϕ and broadcasts

$$\langle\ index, E_\phi(k)\ \rangle.$$

So only one encrypted session key is required. If $r > d_w$, then the center encrypts the session key k with each non-revoked user's private key ψ_u and broadcasts

$$\langle\ index, E_{\psi_{u_1}}(k), E_{\psi_{u_2}}(k), \ldots, E_{\psi_{u_{n-r}}}(k)\ \rangle,$$

where $u_1, \ldots, u_{n-r}$ are non-revoked users. This requires $n - r$ encrypted session keys.

Performance Each user stores $w+2$ private keys, the identifier I and the system parameters E, P, p and q.

If $d_{i-1} < r \leq d_i$ for $1 \leq i \leq w$, then the transmitted data consists of an *index*, one base point, and d_i points in $E(\mathbb{F}_p)$. Since $d_i \leq \alpha(d_{i-1}+1) \leq \alpha r$, it is at most $\alpha r + 1$ elements in $E(\mathbb{F}_p)$. If $r = 0$, then the header contains only one encryption. If $r > d_w$, then the header contains $n - r$ encryptions, which is also bounded by $\alpha r + 1$ since $n - r \leq n - (d_w + 1) \leq \alpha r + 1$. In any cases, the transmission overhead is at most $\alpha r + 1$ points ignoring the *index*.

The computation overhead is $d_i + 1 (\leq \alpha r + 1)$ scalar multiplications in $E(\mathbb{F}_p)$ if $d_{i-1} < r \leq d_i$. If $r > d_w$, then it is $n - r (\leq \alpha r + 1)$ symmetric encryptions. Since one symmetric encryption is faster than a scalar multiplication in $E(\mathbb{F}_p)$, the computation overhead is bounded by $(\alpha r + 1)$ scalar multiplications in $E(\mathbb{F}_p)$. We can save the computation using a simultaneous scalar multiplication method, which will be discussed in Section 5.

Security If $r = 0$ or $r > d_w$, then the scheme is secure since the session key is encrypted by a symmetric encryption algorithm using the key ϕ or private keys ψ_u's of non-revoked users. If $d_{i-1} < r \leq d_i$, the session key $f_{d_i}(0)Q$ is secure against the coalition of the r revoked users. This is because every revoked user has only d_i values and cannot gain any further information. A more detailed security proof is given in Appendix.

4 Extended scheme

Although the basic scheme is quite efficient when the number n of users is small, the scheme is not usable in practice when n is very large because the polynomial degree grows too big and so does the computation overhead. This problem, however, can be resolved easily by partitioning the users into small partitions and applying the basic scheme to each partition. Despite the number of revoked users in each partition can vary dynamically, we can handle the dynamics properly with the polynomials chosen in the basic scheme.

In the initialization step, the center divides n users into m partitions of size D. Actually, D and m are adjusted by the system parameter α in order to make the computation overhead reasonable. (This will be discussed in Section 5.) Because each user should belong to one and only one partition, we can apply the basic scheme to each partition. The center generates random polynomials

$$f_{1d_i}, f_{2d_i}, \ldots, f_{md_i} \quad \text{for } i = 1, 2, \ldots, w,$$

where f_{jd_i} is a polynomial of degree d_i over $\mathbb{Z}_q$ assigned to the j-th partition. Then it provides each user u in the j-th partition, via a private channel, his/her user-key

$$K_u = \langle\ I,\ f_{jd_1}(I),\ f_{jd_2}(I),\ \ldots,\ f_{jd_w}(I), \phi_j, \psi_u\ \rangle,$$

where I is the identifier of u, j is the partition number of the partition containing u, and ϕ_j, ψ_u are the keys corresponding to ϕ, ψ_u, respectively, in the basic scheme.

For revocation, the center first learns the identifiers of r revoked users and the partitions they belong to. Let $u_{j1}, u_{j2}, \ldots, u_{jr_j}$ be the revoked users in the j-th partition, where

$$r = r_1 + r_2 + \cdots + r_m.$$

If $d_{i-1} < r_j \leq d_i$, then the polynomial f_{jd_i} will be used for revocation in the j-th partition. Let's denote this d_i(depending on r_j) by t_j for convenience in the following. The center then chooses $t_j - r_j$ random distinct points, say $I_{j,r_j+1}, I_{j,r_j+2}, \ldots, I_{jt_j}$, from $\mathbb{Z}_q$, which are distinct from the identifiers of the users in the j-th partition, and broadcasts the following:

$$\langle\ \ldots,\ par(j),\ ind(j),\ Q_j,\ (I_{j1}, f_{jt_j}(I_{j1})Q_j), \ldots, (I_{jr_j},\ f_{jt_j}(I_{jr_j})Q_j),$$
$$(I_{j,r_j+1},\ f_{jt_j}(I_{j,r_j+1})Q_j),\ \ \ldots\ , (I_{jt_j},\ f_{jt_j}(I_{jt_j})Q_j),\ \ \ldots\ \rangle\,,$$

where $par(j)$ is the indicator of the j-th partition, $ind(j) = t_j$, $Q_j = s_jP$, s_j is a random number of the j-th partition and I_{ja} is the identifier of u_{ja} for $1 \leq a \leq r_j$. Each non-revoked user u in the j-th partition can find which part of the header is for his/her partition from the partition indicator $par(j)$, and then he/she can compute the session key $k = f_{jt_j}(0)Q_j$ as in the basic scheme. If $r_j = 0$, then the center broadcasts

$$\langle\ \ldots, par(j), ind(j), E_{\phi_j}(k), \ldots\ \rangle.$$

If $r_j > d_w$, then $D - r_j < \alpha r_j$. So the center sends following:

$$\langle\ \ldots, par(j), ind(j), E_{\psi_{u_1}}(k), E_{\psi_{u_2}}(k), \ldots, E_{\psi_{u_{D-r_j}}}(k), \ldots\ \rangle,$$

where $u_1, u_2, \ldots, u_{D-r_j}$ are non-revoked users and $E_{\psi_u}(k)$ is a symmetric encryption of the session key k by ψ_u.

In the above, if $f_{jt_j}(0)Q_j$'s are distinct, then we have to insert more information in the header for all privileged users in each partition to compute the session key k. To avoid this, we make all $f_{jt_j}(0)Q_j$'s be equal as follows:

Choose a random $s \in \mathbb{Z}_q$ and let

$$\widehat{s} = f_{1t_1}(0)f_{2t_2}(0)\cdots f_{mt_m}(0)s.$$

For $\widehat{s}$, define

$$s_j = \frac{\widehat{s}}{f_{jt_j}(0)} \pmod q \quad \text{for all } j = 1, 2, \ldots, m.$$

Then we have

$$k = f_{1t_1}(0)Q_1 = f_{2t_2}(0)Q_2 = \cdots = f_{mt_m}(0)Q_m. \tag{2}$$

5 Analysis

In the basic scheme, given α and w, the maximum possible value of n is $\lfloor (d_w + 1)(\alpha + 1) \rfloor + 1$. Since the performance of the basic scheme depends only on α and w, we take D to be $\lfloor (d_w + 1)(\alpha + 1) \rfloor + 1$ in the extended scheme.

Storage Overhead The storage overhead for each user is $w + 2$ elements in $\mathbb{F}_p$ as in the basic scheme. We estimate w in terms of D. Recall that d_i's are determined from recurrence relations: $d_1 = 1$ and $d_{i+1} = \lfloor \alpha(d_i + 1) \rfloor$. Since $d_{i+1} > \alpha(d_i + 1) - 1$, we have

$$d_w + 1 > \alpha(d_{w-1} + 1) > \cdots > \alpha^{w-1}(d_1 + 1) = 2\alpha^{w-1}.$$

Since $D > (\alpha + 1)(d_{w-1} + 1)$, we obtain $D > 2(\alpha + 1)\alpha^{w-2} > 2\alpha^{w-1}$ and hence $w < \log D / \log \alpha$ (we assume that $1 < \alpha < 2$).

Transmission Overhead Since the size of $par(j)$ and $ind(j)$ are negligible, the transmission overhead for the j-th partition is $\alpha r_j + 1$ elements of $E(\mathbb{F}_p)$ in the worst case. So the total message overhead is at most

$$\sum_{j=1}^{m} (\alpha r_j + 1) = \sum_{j=1}^{m} \alpha r_j + m = \alpha r + m.$$

Computation Overhead Most computation of this scheme consists of scalar multiplications in $E(\mathbb{F}_p)$. To speed up this, one can use a simultaneous scalar multiplication method [**?**]. If one computes c scalar multiplications using Non-Adjacent Form (NAF), it takes $c \log q$ (doublings) + $(c/3) \log q$ (additions) on the average. Using the standard simultaneous scalar multiplication method, it takes $\log q$ (doublings)+$(c/3) \log q$ (additions) for c scalar multiplications, which amounts about $(3 + c)/4$ scalar multiplications.

The computation overhead for each user is bounded by $d_w + 1$ scalar multiplications, as in the basic scheme. Using the standard simultaneous multiplication method, it is reduced to $(d_w/4) + 1$ scalar multiplications. If the revoked users are assumed to be uniformly distributed over all partitions, however, there are r/m revoked users in each partition on the average. In this case, the computation overhead is about $(r/4m) + 1$ scalar multiplications.

Optimization In the extended scheme, we first fix a constant α with $1 < \alpha < 2$ and an upper bound M of the computation overhead (the number of scalar multiplications in $E(\mathbb{F}_p)$). And then, in order to optimize the scheme's efficiency, we choose the other system parameters as follows:

- Compute d_i's, where $d_1 = 1$ and $d_{i+1} = \lfloor \alpha(d_i + 1) \rfloor$
- Find the maximum w such that $(d_w/4) + 1 \leq M$
- Compute $D = \lfloor (d_w + 1)(\alpha + 1) \rfloor + 1$ (with this D, we can optimize the upper bound of w with $(\log D - 1)/ \log \alpha$
- Divide all users into $m = \lceil n/D \rceil$ partitions of size D

The extended scheme with these optimized parameters is summarized in Fig. 1.

Parameters
- n : the number of total users in the system
- p, q : 160-bit primes
- E : an elliptic curve over $\mathbb{Z}_p$
- P : an element of $E(\mathbb{F}_p)$ of order q
- M : an upper bound of the computation overhead

Setup (partitioning)
- Pick $\alpha (1 < \alpha < 2)$ and compute d_i's, where $d_1 = 1$ and $d_{i+1} = \lfloor \alpha(d_i + 1) \rfloor$
- Find the maximum w satisfying $d_w \leq 4(M-1)$
- Compute $D = \lfloor (d_w + 1)(\alpha + 1) \rfloor + 1$
- Divide all users into $m = \lceil n/D \rceil$ partitions of size D

Key Generation of the j-th partition for $j = 1, 2, \ldots, m$
- Assign the identifier I to each user u
- Make w random polynomials $f_{jd_1}, f_{jd_2}, \ldots, f_{jd_w}$ over $\mathbb{Z}_p$, where $\deg(f_{jd_i}) = d_i$
- Give the set $K_u = \{I, f_{jd_1}(I), \ldots, f_{jd_w}(I), \phi, \psi_u\}$ to each user u

Encryption and Decryption

∘ **Case 1** $1 \leq r_j \leq d_w$**:**
- Assume that there are r_j revoked users $u_{j1}, u_{j2}, \ldots, u_{jr_j}$ in the j-th partition
- For each j find d_i satisfying $d_{i-1} < r_j \leq d_i$ (convention : $d_0 = 0$) and let $t_j = d_i$
- Choose $s \in \mathbb{Z}_q$ randomly and compute $\hat{s} = f_{1t_1}(0) f_{2t_2}(0) \cdots f_{mt_m}(0) s$
- For $\hat{s}$, define $s_j = \hat{s} / f_{jt_j}(0) \pmod q$ for each $j = 1, 2, \ldots, m$
- Compute $Q_j = s_j P$
- Select distinct I_{ja}'s which are different from the identifiers assigned to the users in the j-th partition for $a = r_j + 1, r_j + 2, \ldots, t_j$
- Broadcast the following message:

$$\langle\ \ldots, par(j), ind(j), Q_j, (I_{j1}, f_{jt_j}(I_{j1})Q_j), (I_{j2}, f_{jt_j}(I_{j2})Q_j), \ldots,$$
$$(I_{jr_j}, f_{jt_j}(I_{jr_j})Q_j), (I_{j,r_j+1}, f_{jt_j}(I_{j,r_j+1})Q_j), \ldots, (I_{jt_j}, f_{jt_j}(I_{jt_j})Q_j), \ldots\ \rangle$$

- Assume that u with identifier $I = I_0$ is a privileged user in the j-th partition
- From the message, find out t_j
- Compute $\lambda_{ja} = \prod_{b \neq a} \frac{I_{jb}}{I_{jb} - I_{ja}}$ and the session key k as follows:

$$k = f_{jt_j}(0)Q = \sum_{a=0}^{t_j} \lambda_{ja} f_{jt_j}(I_{ja})\ \ Q = \sum_{a=0}^{t_j} \lambda_{ja}\ \ f_{jt_j}(I_{ja})Q$$

∘ **Case 2** $r_j = 0$**:**
- In this case, the center transmits $\langle\ \ldots, par(j), ind(j), E_{\phi_j}(k), \ldots\ \rangle$, where $E_{\phi_j}(k)$ is a symmetric encryption of the session key k by ϕ_j
- Then all users in the j-th partition decrypt by ϕ_j

∘ **Case 3** $r_j > d_w$**:**
- Let $u_1, u_2, \ldots, u_{D-r_j}$ be the non-revoked users. Then the center broadcasts

$$\langle\ \ldots, par(j), ind(j), E_{\psi_{u_1}}(k), E_{\psi_{u_2}}(k), \ldots, E_{\psi_{u_{D-r_j}}}(k), \ldots\ \rangle$$

  where $E_{\psi_u}(k)$ is a symmetric encryption of the session key k by ψ_u
- Then each privileged user u in the j-th partition decrypts by ψ_u

Fig. 1. Extended Scheme

6 Comparison

Table 1 provides the values of d_i's for some α's such that $1 < \alpha < 2$. In the following, because the computation overhead depends on the degrees of polynomials, we set 40 as an upper bound of the degrees. Interpolating a polynomial of degree 40 requires about 11 scalar multiplications in $E(\mathbb{F}_p)$, which, we believe, is reasonable in practice. The table also provides the value of D, the size of one partition, which is determined from given α and d_w. With these values, the transmission overhead is less than or equals to $\alpha r + 1$ for any number r of the revoked users in one partition.

Table 1. The values of d_i's

α	d_i's														size of D
3/2	1	3	6	10	16	25	39								101
4/3	1	2	4	6	9	13	18	25	34						82
$\sqrt[4]{2}$	1	2	3	4	5	7	9	11	14	17	21	26	32	39	88

In Table 2, we compare our extended scheme with SD [**?**] and LSD [**?**], which are regarded as the best known broadcast encryption schemes. The threshold column shows that our scheme has the smaller message length than SD and LSD except when r/n is very small. For example, our scheme with $\alpha = \sqrt[4]{2}$ has smaller message length than SD when the number of revoked users is larger than 1.41 % of the total users. Fig. 2 depicts the transmission overhead (TO) of each scheme with respect to $100r/n(\%)$.

Table 2. Efficiency Comparison

Scheme	Key Storage	# of Computations	Message Length	Threshold(vs SD)
SD [**?**]	$log^2 n$	$\log n$ hashes	$2r - 1$	-
LSD [**?**]	$log^{3/2} n$	$\log n$ hashes	$4r$	-
Ours	$\lfloor \frac{(\log D - 1)}{\log \alpha} \rfloor + 2$	$(d_w/4) + 1$ mul. in E	$\alpha r + n/D$	-
α=3/2	9	10.75 mul. in E	$(3r/2) + (n/101)$	$r > 1.99n/100$
α=4/3	11	9.5 mul. in E	$(4r/3) + (n/82)$	$r > 1.83n/100$
$\alpha = \sqrt[4]{2}$	16	10.75 mul. in E	$\sqrt[4]{2}\, r + (n/88)$	$r > 1.41n/100$

¿From this comparison, we can see that our scheme always has the smaller key storage as well than SD and LSD. The computation overhead, however, is worse. But even so, our scheme is still practical for systems with some computing power. Furthermore, considering the fact that the transmission cost is much more expensive than the computation cost in practice, this is a desirable trade off. In addition, if the revoked users are assumed to be uniformly distributed

in each partition, the computation overhead $O(d_w)$ is reduced to $(r/4m) + 1$, which is 2.26, 2.03, and 2.1 scalar multiplications in $E(\mathbb{F}_p)$ when $r \leq 0.05n$ and $\alpha = 3/2, 4/3, \sqrt[4]{2}$, respectively.

Fig. 2. Comparison of Transmission Overhead for n = 100,000,000

7 Security proof

Naor and Pinkas [**?**] proved that their NP scheme is secure against coalitions of up to d revoked users under the computational Diffie-Hellman (CDH) assumption, where d is the degree of a polynomial predetermined. We generalize their proof to our basic scheme (with multiple polynomials) in the following lemma.

Lemma. *The basic scheme is secure against coalition of r revoked users under the CDH assumption for any r.*

Proof. See Appendix. □

And then we prove the extended scheme (with multiple partitions) is also secure under the same assumption in the following theorem.

Theorem. *The extended scheme is secure against coalition of total r revoked users, where $r_1, r_2, \ldots, r_m$ are the numbers of revoked users in partitions*

$$G_1, G_2, \ldots, G_m,$$

respectively, such that

$$r = r_1 + r_2 + \cdots + r_m.$$

Proof. See Appendix. □

Observe that if different $f_{jt_j}(0)Q_j$'s - called the *partition keys* - are used for different partitions in the extended scheme, the security proof is an immediate consequence of the lemma above. But this increases the transmission overhead by $O(m)$. In order to avoid this, the extended scheme makes all the partition keys be the same. The main part of the proof is, in fact, the part proving that the same partition key does not harm to the security of the scheme.

The authors expect that our proof is useful for security proofs in similar situations, that is, our security proof works for any broadcast encryption scheme with multiple partitions using the same partition key.

8 Conclusion

In this paper, we introduced two simple ideas, multiple interpolations and multiple partitions, to the NP scheme to obtain an efficient broadcast encryption scheme for a large number of stateless receivers. Our scheme fully satisfies the general requirements of broadcast encryption (even the log-key restriction, the notion of which was introduced in this year's Crypto) and the scheme's efficiency is comparable to the most efficient broadcast encryption schemes known. Our scheme, in fact, is better in the storage overhead and in the transmission overhead. It costs more computations but the computation overhead is reasonable for systems with moderate computing power. Our scheme is applicable to any abelian group over which the CDH is hard. Moreover, our scheme has another advantage: later entry of new users is very easy and cheap because we can simply add new partitions at anytime. We expect that this advantage is exploited in many applications in practice. We provide a detailed security proof in Appendix.

References

1. J. Anzai, N. Matsuzaki and T. Matsumoto, *A quick key distribution scheme with "Entity Revocation"*, Advances in Cryptology - Asiacrypt'99, Lecture Notes in Computer Science 1716, pp.333-347.

2. S. Berkovits, *How to Broadcast a secret*, Advances in Cryptology - Eurocrypt'91, Lecture Notes in Computer Science 547, pp.536-541.
3. G. Chick and S. Tavares, *Flexible access control with master keys*, Advances in Cryptology - Crypto'89, Lecture Notes in Computer Science, pp.316-322.
4. P. D'Aroco and D.R. Stinson, *Fault Tolerant and Distributed Broadcast Encrytion*, CT - RSA'03, Lecture Notes in Computer Science 2612, pp.263-280.
5. A. Fiat and M. Naor, *Broadcast Encryption*, Advances in Cryptology - Crypto'93, Lecture Notes in Computer Science 773, pp.480-491.
6. M.T. Goodrich, J.Z. Sun and R. Tamassia, *Efficient Tree-Based Revocation in Groups of Low-State Devices*, Advances in Cryptology - Crypto'04, Lecture Notes in Computer Science 3152, pp.511-527.
7. J. Garay, J. Staddon and A. Wool, *Long-Lived Broadcast Encryption*, Advances in Cryptology - Crypto'00, Lecture Notes in Computer Science 1880, pp.333-352.
8. D. Halevi and A. Shamir, *The LSD Broadcast Encryption Scheme*, Advances in Crytology - Crypto'02, Lecture Notes in Computer Science 2442, pp.47-60.
9. R. Kumar, S. Rajagopalan and A. Sahai, *Coding Constructions for blacklisting problems without Computational Assumptions*, Advances in Cryptology - Crypto'99, Lecture Notes in Computer Science 1666, pp.609-623.
10. B. Möller, *Algorithms for Multi-exponentiation*, Selected Areas in Cryptography - SAC'01. Lecture Notes in Computer Science 2259, pp.165-180.
11. D. Naor, M. Naor and J. Lotspiech, *Revocation and Tracing Schemes for Stateless Receivers*, Advances in Cryptology - Crypto'01, Lecture Notes in Computer Science 2139, pp.41-62.
12. M. Naor and B. Pinkas, *Efficient Trace and Revoke Schemes*, Financial Cryptography'00, Lecture Notes in Computer Science.
13. C.K. Wong, M. Gouda and S.S. Lam, *Secure Group Communication using Key Graphs*, ACM SIGGCOM'98 ACM.
14. M. Luby and J. Staddon, *Combinatorial Bounds for Broadcast Encryption*, Advances in Cryptology Eurocrypt'98, Lecture Notes in Computer Science 1403, pp.512-526.
15. A. Shamir, *How to Share a Secret*, Comm. ACM 22, pp.612-613.

Appendix: Security Proof

Naor and Pinkas [**?**] proved that the NP scheme is secure against coalitions of up to t revoked users under the computational Diffie-Hellman assumption.

Computational Diffie-Hellman Assumption: For a cyclic group G in which DLP is hard, let $g \in G$ be a generator of G. Then there is no efficient polynomial time algorithm that can compute g^{xy} from $\langle g, g^x, g^y \rangle$, where x and y are random integers in the interval $[1, |G|]$. In an elliptic curve, computational Diffie-Hellman assumption means that there is no efficient algorithm that can compute xyP from $\langle P, xP, yP \rangle$.

We will prove that our scheme is also secure under the same assumption.

Lemma. *The basic scheme is secure against coalition of r revoked users under CDH assumption for any r.*

Proof. Let r be the number of revoked users. When $r = 0$ or $r > d_w$, then the security of the scheme is guaranteed by the security of the symmetric encryption algorithm used. So we may assume that $r \in (d_{i-1}, d_i]$ for some $i = 1, 2, \ldots, w$, where $d_0 := 0$. We first consider the case of one time revocation. Since $r \in (d_{i-1}, d_i]$, the polynomial f_{d_i} of degree d_i is used for revocation. Points on other polynomials are useless to compute the session key since all polynomials are chosen randomly. Therefore, for revoked users, computing the session key is equivalent to finding polynomial of degree d_i with only d_i points, which is impossible by [**?**].

Next we consider the case of repeated revocations. Let r and r' be the number of revoked users in current session and previous session, respectively. If $r \in (d_{i-1}, d_i]$ but $r' \notin (d_{i-1}, d_i]$, then the scheme is secure because the current polynomial f_{d_i} is different from previous ones. We now assume that $r, r' \in (d_{i-1}, d_i]$. Then the same polynomial f_{d_i} of degree d_i is used in two different sessions, say the first two sessions. Assume also that a user u was not revoked in the first session but is revoked in the second session. Let the revoked users in the second session be $u_1 (= u), u_2, \ldots, u_r$. At least the following information are available to u with coalition of r revoked users:

$$\begin{gathered} s^{(1)}P,\ f_{d_i}(I_1)s^{(1)}P,\ \ldots,\ f_{d_i}(I_r)s^{(1)}P,\ f_{d_i}(I'_{r+1})s^{(1)}P,\ \ldots, f_{d_i}(I'_{d_i})s^{(1)}P, \\ f_{d_i}(0)s^{(1)}P,\ s^{(2)}P,\ f_{d_i}(I_1)s^{(2)}P,\ \ldots,\ f_{d_i}(I_r)s^{(2)}P,\ f_{d_i}(I_{r+1})s^{(2)}P,\ \ldots, \\ f_{d_i}(I_{d_i})s^{(2)}P,\ I_1,\ I_2,\ \ldots,\ I_r,\ f_{d_i}(I_1),\ f_{d_i}(I_2),\ \ldots,\ f_{d_i}(I_r), \end{gathered}$$

where I_a is the identifier of u_a for each $a = 1, 2, \ldots, r$; I'_j's and I_j's for $j = r+1, \ldots, d_i$ are random and distinct from I_a's; and $s^{(b)}$ is a random number chosen in the b-th session for each b = 1,2.

Although u doesn't know the value of the session key in session 2, u knows that the session key is of the form $f_{d_i}(0)s^{(2)}P$. In the following, we prove that u cannot find the value of $f_{d_i}(0)s^{(2)}P$. Suppose that the scheme is not secure, that is, there is an efficient algorithm O with input

$$\begin{gathered} Input = \langle\ s^{(1)}P,\ f_{d_i}(I_1)s^{(1)}P,\ \ldots,\ f_{d_i}(I_r)s^{(1)}P,\ f_{d_i}(I'_{r+1})s^{(1)}P,\ \ldots, \\ f_{d_i}(I'_{d_i})s^{(1)}P,\ f_{d_i}(0)s^{(1)}P,\ s^{(2)}P,\ f_{d_i}(I_1)s^{(2)}P,\ \ldots, \\ f_{d_i}(I_r)s^{(2)}P,\ f_{d_i}(I_{r+1})s^{(2)}P,\ \ldots, f_{d_i}(I_{d_i})s^{(2)}P, \\ I_1,\ I_2,\ \ldots,\ I_r,\ f_{d_i}(I_1),\ f_{d_i}(I_2),\ \ldots,\ f_{d_i}(I_r)\ \rangle, \end{gathered}$$

which can compute $f_{d_i}(0)s^{(2)}P$. Let

$$O(\ Input\ )\ =\ f_{d_i}(0)s^{(2)}P$$

Then from the algorithm O, one can derive an efficient algorithm O' that can compute g^{xy} from arbitrarily given inputs g, g^x and g^y (see [**?**] for details). This, however, implies that u can solve the computational Diffie-Hellman problem. So the scheme is secure against coalition of r revoked users under CDH assumption. □

Theorem. *The extended scheme is secure against coalition of total r revoked users, where $r_1, r_2, \ldots, r_m$ are the numbers of revoked users in partitions*

$$G_1, G_2, \ldots, G_m,$$

respectively, such that

$$r = r_1 + r_2 + \cdots + r_m.$$

Proof. We first show that the session key cannot be recovered from given secret shares for the revoked users. Since polynomials in this scheme are all chosen randomly, secret shares (points on polynomials) in one partition are useless in guessing the polynomials in the other partitions. So, the security of the whole scheme depends on the security of the session key of each partition. For all $j = 1,2,\ldots,m$, we define t_j as d_i satisfying $d_{i-1} < r_j \leq d_i$. Since the case of $r_j = 0$ or $r_j > d_w$ can be proved trivially as in the previous lemma, we may assume that $0 < r_j \leq d_w$.

We now prove that using the same partition key for all partitions is also secure. In the extended scheme, non-revoked users in all partitions can compute the same partition key

$$k = f_{1t_1}(0)s_1P = f_{2t_2}(0)s_2P = \cdots = f_{mt_m}(0)s_mP$$

as in (2). This holds for all sessions. This, however, does not cause any weakness. In order to show this, we first consider the case of one time revocation with two partitions. In this case, the coalition of $r = r_1 + r_2$ revoked users know the values of s_1P, s_2P, the secret shares of r users, and the fact that $f_{1t_1}(0)s_1P = f_{2t_2}(0)s_2P$. Then,

$$f_{1t_1}(0)s_1P = \lambda_u f_{1t_1}(I_u)s_1P + \sum_{a=1}^{t_1} \lambda_{1a} f_{1t_1}(I_{1a})s_1P,$$

$$f_{2t_2}(0)s_2P = \lambda_v f_{2t_2}(I_v)s_2P + \sum_{a=1}^{t_2} \lambda_{2a} f_{2t_2}(I_{2a})s_2P,$$

where u and v are non-revoked users chosen randomly from the first and the second partitions, respectively, and λ_u and λ_v are the constants obtained from the formula (1). So the attackers, the r revoked users, can make the following equation:

$$A + \alpha = B + \beta,$$

where

$$A = \lambda_u f_{1t_1}(I_u)s_1P, \quad B = \lambda_v f_{2t_2}(I_v)s_2P,$$

$$\alpha = \sum_{a=1}^{t_1} \lambda_{1a} f_{1t_1}(I_{1a})s_1P, \quad \beta = \sum_{a=1}^{t_2} \lambda_{2a} f_{2t_2}(I_{2a})s_2P.$$

Here, α and β are known to the attackers, but A and B are not. Although there are many pairs of (A,B) satisfying the above equation, it is impossible for the attackers to find the right values of A and B, which render the correct session key. When there are three partitions, the attackers may obtain

$$A + \alpha = B + \beta = C + \gamma.$$

But it is still impossible for them to find the correct session key. It is obvious from mathematical induction that increasing the number of partitions does not do any harm to the security of the scheme.

Next, let's consider the case of repeated revocations. Assume that there are two partitions. We may assume that for each $j = 1,2$, the degree t_j of the polynomial corresponding to the number $r_j^{(b)}$ of the revoked users in the j-th partition dose not change in the b-th session for $b = 1, 2$. Assume further that in the first session, some of the r revoked users in the second session were not revoked so that they know the first session key. After receiving the header of the second session, those r revoked users know

$$s_1^{(1)}P, \quad s_2^{(1)}P, \quad f_{1t_1}(0)s_1^{(1)}P \;=\; f_{2t_2}(0)s_2^{(1)}P$$

and the private keys of their own, where $s_j^{(b)}$ denotes a random number used in the j-th partition in the b-th session. To break the scheme, they must obtain the value of

$$f_{1t_1}(0)s_1^{(2)}P = f_{2t_2}(0)s_2^{(2)}P.$$

As above, the attackers can set up the following equations:

$$A + \alpha = B + \beta \tag{3}$$

$$\tau A + \alpha' = \tau B + \beta',$$

where $\tau \in \mathbb{Z}_q$ satisfying $\tau s_1^{(1)}P = s_1^{(2)}P$, $\tau s_2^{(1)}P = s_2^{(2)}P$ and

$$\alpha' = \sum_{a=1}^{t_1} \lambda_{1a}^{(2)} f_{1t_1}(I_{1a}^{(2)})s_1^{(2)}P, \quad \beta' = \sum_{a=1}^{t_2} \lambda_{2a}^{(2)} f_{2t_2}(I_{2a}^{(2)})s_2^{(2)}P,$$

where $I_{ja}^{(b)}$ is the identifier of a user $u_{ja}^{(b)}$, a revoked user in the j-th partition in the b-th session for $1 \le a \le r_j^{(b)}$. Since α, α', β and β' satisfy

$$\tau A + \tau\alpha = \tau B + \tau\beta \quad \text{and} \quad (\tau\alpha) - (\alpha') = (\tau\beta) - (\beta'),$$

we obtain

$$\beta' - \alpha' = \tau(\beta - \alpha). \tag{4}$$

Suppose that there is an efficient algorithm O that can compute τA, which renders the second session key. In other words, let

$$O(A, \alpha, B, \beta, \alpha', \beta') = \tau A,$$

where $A, B, \alpha, \beta, \alpha', \beta'$ satisfy the conditions (**??**) and (**??**). Then, one can derive an efficient algorithm O' that can compute xyP with the input data P, xP, yP as follows:

$$O'(P, xP, yP) = O(\ xP,\ P,\ (x - \xi + 1)P,\ \xi P,\ \eta P,\ (y(\xi - 1) + \eta)P\ ),$$

where ξ and η are random. From the conditions (**??**) and (**??**), one can easily obtain that $\tau = y$ and $O'(P, xP, yP) = xyP$.

Since this contradicts to the CDH assumption, we may conclude that there is no efficient algorithm that can compute the second session key. For three partitions, we can prove using the algorithm O defined by

$$O(A, \alpha, B, \beta, C, \gamma, \alpha', \beta', \gamma') = \tau A.$$

where $A, B, C, \alpha, \beta, \gamma, \alpha', \beta', \gamma'$ satisfy the conditions

$$A + \alpha = B + \beta = C + \gamma$$

$$\tau(\beta - \alpha) = \beta' - \alpha' \ , \ \tau(\gamma - \alpha) = \gamma' - \alpha'.$$

By the same argument, we can derive from O an efficient algorithm O' that can solve the CDH, which proves the security of the scheme. So, the security of the extended scheme with two sessions follows from mathematical induction on the number of partitions.

Assume that the extended scheme with m partitions is secure in the first $\ell(\geq 2)$ sessions for any m. We now suppose that if the attackers know the first ℓ session keys, then session $(\ell + 1)$ is not secure. Under this supposition, we prove that the second session is not secure if the attackers know the first session key, which is a contradiction, as follows:

¿From the first session, the attackers can set up the equation

$$A_1 + \alpha_1 = A_2 + \alpha_2 = \cdots = A_m + \alpha_m \tag{5}$$

and they know that the second session key is

$$\tau A_1 + \alpha_1' = \tau A_2 + \alpha_2' = \cdots = \tau A_m + \alpha_m'$$

for some τ. From (**??**) the attackers can compute

$$\tau^{(b)} A_1, \ \tau^{(b)} A_2, \ \ldots, \ \tau^{(b)} A_m, \ \alpha_1^{(b)}, \ \alpha_2^{(b)}, \ \ldots, \ \alpha_m^{(b)}$$

satisfying

$$\alpha_1^{(b)} = \tau^{(b)}\alpha_1 + t^{(b)}, \ \alpha_2^{(b)} = \tau^{(b)}\alpha_2 + t^{(b)}, \ \ldots, \ \alpha_m^{(b)} = \tau^{(b)}\alpha_m + t^{(b)}$$

from randomly chosen distinct $\tau^{(b)}$ and $t^{(b)}$. Then it is easy to check that

$$\tau^{(b)} A_1 + \alpha_1^{(b)} = \tau^{(b)} A_2 + \alpha_2^{(b)} = \cdots = \tau^{(b)} A_m + \alpha_m^{(b)} \tag{6}$$

and

$$\alpha_2^{(b)} - \alpha_1^{(b)} = \tau^{(b)}(\alpha_2 - \alpha_1), \ldots, \ \alpha_m^{(b)} - \alpha_1^{(b)} = \tau^{(b)}(\alpha_m - \alpha_1)$$

for $b = 1,2,\ldots,\ell-1$. With these $\ell-1$ equations in (**??**) together with the equation (**??**), the attackers can find the second session key (regarded as the $(\ell + 1)$-st session) by the supposition.

Therefore, we may conclude that the extended scheme with any number of partitions is secure in any number of repeated sessions. □

We expect that our security proof may be applied to systems using partitions. In particular, when a broadcast encryption, which is efficient for a small set of users, is applied to many disjoint such sets, our security proof may help reducing the transmission overhead further securely.