One-way Chain Based Broadcast Encryption Schemes*

Nam-Su Jho[1], Jung Yeon Hwang[2]**, Jung Hee Cheon[1], Myung-Hwan Kim[1], Dong Hoon Lee[2], and Eun Sun Yoo[1]

[1] ISaC and Department of Mathematical Sciences, Seoul National University, Seoul 151-747, Korea
{drake,jhcheon,mhkim,eunsun}@math.snu.ac.kr
[2] Graduate School of Information Security CIST, Korea University, Seoul 136-701, Korea
{videmot,donghoonlee}@korea.ac.kr

Abstract. We propose a new broadcast encryption scheme based on the idea of 'one key per each punctured interval'. Let r be the number of revoked users. In our scheme with p-punctured c-intervals, the transmission overhead is roughly $\frac{r}{p+1}$ as r grows. Our scheme is very flexible with two parameters p and c. We may take p as large as possible if a user device allows a large key storage, and set c as small as possible if the storage size and the computing power is limited. As variants of the proposed scheme, we further study a combination of a one-way chain and a hierarchical ring. This combination provides a fine-grained trade-off between user storage and transmission overhead. As one specific instance, the combination includes the subset difference (SD) scheme which is considered the most efficient one in the literature.

1 Introduction

Broadcast encryption (BE) is a cryptographic method for a center to efficiently broadcast digital contents to a large set of users so that only non-revoked users can decrypt the contents. In broadcast encryption, the center distributes to each user u the set K_u of keys, called the *user key set* of u, in the system setup stage. We assume that the user keys are not updated afterwards, that is, user keys are *stateless*. A *session* is a time interval during which only one encrypted message (digital contents) is broadcasted. The *session key*, say SK, is the key used to encrypt the contents of the session. In order to broadcast a message M, the center encrypts M using the session key SK and broadcasts the encrypted message together with a *header*, which contains encryptions of SK and the information for non-revoked users to recover SK. In other words, the center broadcasts

$$\langle \text{header} ; E_{SK}(M) \rangle,$$

where $E_{SK}(M)$ is a symmetric encryption of M by SK. Then, every non-revoked user u computes $F(K_u, \text{header}) = SK$ and decrypts $E_{SK}(M)$ with SK, where

* This work was supported by Samsung Advanced Institute of Technology.
** Corresponding author.

F is a predefined algorithm. But for any revoked user v, $F(K_v, \text{header})$ should not render SK. Furthermore, there should be no polynomial time algorithm that outputs SK even with all the revoked user keys and the header as input.

The length of the header, the computing time of F and the size of a user key are called the *transmission overhead*, the *computation cost* and the *storage size*, respectively. The main issue of broadcast encryption is to minimize the transmission overhead with practical computation cost and storage size.

The notion of broadcast encryption was first introduced by Berkovits [2] in 1991 using polynomial interpolation and vector based secret sharing. Fiat and Naor [7] in 1993 suggested a formal definition of broadcast encryption and proposed a systematic method of broadcast encryption. The polynomial interpolation method was improved by Naor and Pinkas [14] in 2000 to allow multiple usage. The first practical broadcast encryption scheme was proposed in 2001 by Naor *et al.* [13], called the *Subset Difference* (SD) method. This was improved by Halevi and Shamir [11] in 2002 by adopting the notion of layers and thereby the improved scheme is called the *Layered Subset Difference* (LSD) method. Both SD and LSD are based on tree structure and they are the best known broadcast schemes up to now. To be more precise, let N be the total number of users and r be the number of revoked users. The SD scheme requires $2r$ transmission overhead and $O(\log^2 N)$ storage size for each user. The computation cost is only $O(\log N)$ computations of one-way permutations. The LSD scheme reduces the storage size to $O(\log^{3/2} N)$ while keeping the computation cost same. But the transmission overhead increases to $4r$ in LSD. For other interesting recent articles on broadcast encryption, we refer the readers [3, 8].

Our Contribution In this paper, we propose a new broadcast encryption scheme based on the idea of "one key per each punctured interval". It has been a general belief that at least one key per each revoked user should be included in the overhead and hence r seems to be the lower bound of the transmission overhead in any broadcast encryption scheme with reasonable computation cost and storage size. In our scheme with p-punctured c-intervals, however, the transmission overhead is about $\frac{r}{p+1} + \frac{N-r}{c}$ which breaks the barrier of r, for the first time under our knowledge if r is not too small, even when $p = 1$, where c is a predetermined constant and r is not too small. Although we set $c = 100$ or 1000 for comparison purpose here, we can choose any c that is suitable for other purposes. The computation cost is very cheap with only $c - 1$ computations of one-way permutations. The storage size is $O(c^{p+1})$, which is practical for most user devices if p is small. Our scheme is very flexible with two parameters p and c. If a user device allows a large key storage like set-top boxes and mobile devices, then we may take p as large as possible to reduce the transmission overhead, which is much more expensive. If a user device has limited storage and computing power like smart cards and sensors, then we may set c as small as possible. Another remarkable feature of our scheme is that it does not have to preset the total number of users - any number of additional users can join at any time, which is not possible in tree based schemes.

Our idea is to put all the users on a straight line and divide the line into subintervals of length at most c beginning and ending with non-revoked users containing p or less revoked users in between. Then, to each of such intervals, the center assigns just one key, which can be derived by all non-revoked users in the interval, for decrypting the session key. For practical purpose, we introduce a layered variance of our scheme to improve the efficiency for very small r. Compared with SD and LSD, the variance beats them in the transmission overhead. As for the the storage size, ours is better than SD when $p = 0$ and a little bit worse when $p \geq 1$.

Furthermore, we study a combination of a one-way chain and a hierarchical ring to provide a trade-off between transmission overhead and keys storage size per user. As a building block we first present a simple ring structure of which transmission overhead is proportional to the number r of revoked users while each user stores N keys. Then, by transforming the simple ring structure to a hierarchical ring one recursively, we extend the basic scheme to more generalized revocation schemes. Interestingly, our specific example of one extreme side is structurally equivalent to SD [13] with $1+\frac{1}{2}(\log^2 n+\log n)$ storage size and $2r-1$ transmission overhead.

Organization The rest of this paper is organized as follows: In section 2, we propose revocation schemes with p-punctured intervals together with efficiency and security analysis and introduce layers to our scheme. In Section 3, we propose two revocation schemes based on a ring structure: The first one is a basic scheme with r transmission overhead and the second one is an extension to reduce the transmission overhead below r. We generalize the schemes of a simple ring structure to a *hierarchical* ring structure. In Section 4, we compare our schemes with SD and LSD schemes, and give concluding remarks in Section 5.

2 The Punctured Interval Scheme π

A broadcast encryption scheme involves the center (the message sender) and the set of users (the receivers). Our revocation method is based on a so-called *Subset-Cover* framework proposed by Naor et. al [13]. It consists of three phases as follows:

- The initialization phase: the center provides each user with his/her secret keys that will be used when computing his/her partition key K.
- The broadcast phase: when the center wants to transmit a message M, it partitions the set of all privileged users into disjoint subsets $S_1, ..., S_m$ and computes the partition key K_i corresponding to each S_i, and then broadcasts

  $$\langle \text{info}_1, \text{info}_2, \ldots, \text{info}_m \,;\, \mathcal{E}_{K_1}(SK), \mathcal{E}_{K_2}(SK), \ldots, \mathcal{E}_{K_m}(SK) \,;\, E_{SK}(M) \rangle,$$

  where info_i is the information on S_i, SK is the session key, and $\mathcal{E}, E$ are symmetric encryption functions. The info_i's and $\mathcal{E}_{K_i}(SK)$'s together form the header.

- The decryption phase: each user first finds the partition S_i where he/she belongs from info_i's, computes the partition key K_i using his/her secret keys, and then decrypts SK and M in order.

2.1 Punctured Intervals

Assume that L be a straight line with N dots (users) on it, where N is the number of total users. In our scheme, each user is indexed by an integer $k \in [1, N]$ and he/she is represented by the k-th dot, denoted by u_k, in the line L. Let $p \geq 0$ and $c > 0$ be integers. By a *p-punctured c-interval* we mean a subset of L which contains c or less consecutive users starting from and ending at non-revoked users and containing p or less revoked users. Let $\mathcal{S}_{(p\,;\,c)}$ be the set of all p-punctured c-intervals.

In each session, the p-punctured c-intervals are to be determined under the following rule:

- The first p-punctured c-interval starts from the leftmost non-revoked user, and each of the following starts from the first non-revoked user after the last non-revoked user of the previous.
- Each p-punctured c-interval contains the maximal possible number of users.

Fig.1 illustrates how to make p-punctured c-intervals with an example when $p = 1, c = 6$:

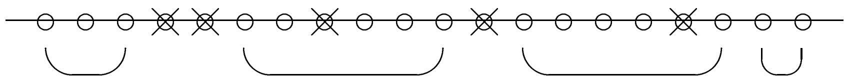


Fig. 1. 1-punctured 6-intervals

The p-punctured c-interval starting from u_i and ending at u_j with $u_{x_1}, \ldots, u_{x_q}$ revoked users is denoted by $P_{i,j;x_1,\ldots,x_q}$ or $P_{i,j;X}$ in short for $X = \{x_1, \ldots, x_q\}$, where $1 \leq j - i + 1 \leq c$, $0 \leq q \leq p$, and $i < x_1 < \cdots < x_q < j$ if there are revoked users.

2.2 Punctured Interval Scheme $(p\,;c)$-π

In this subsection, we propose the punctured interval broadcast encryption scheme $(p\,;c)$-π (PI - Punctured Interval). We assign just one key to each p-punctured c-interval, which can be easily derived by all non-revoked users in that interval, and construct key chains using one-way permutations in order to reduce the storage size.

Key Generation Let $h_t : \{0,1\}^\ell \to \{0,1\}^\ell$ be one-way permutations for $t = 0, 1, \ldots, p$, where ℓ is the key length. To assign one key to each p-punctured interval, we randomly choose N keys $K_{1,1}, K_{2,2}, \ldots, K_{N,N}$ to be given to $u_1, \ldots, u_N$, respectively. From each $K_{i,i}$ the center constructs the one-way key chains under the following rule : For any possible p-punctured c-interval P starting from u_i given,

- The one-way key chain consists only of the keys of all non-revoked users in P. There are no keys of the revoked users in the chain.
- For any non-revoked user $u_k \in P$, if the next user $u_{k+1} \in P$ is also non-revoked, then just apply h_0 to the key of u_k to obtain the key of u_{k+1}.
- If the next t users are revoked and the user $u_{k+t+1} \in P$ is non-revoked, then apply h_t to the key of u_k to obtain the key of u_{k+t+1}, where $1 \le t \le p$.

The following example illustrates how to construct the key chain of a given punctured interval (with $p = 10, c = 20$) :

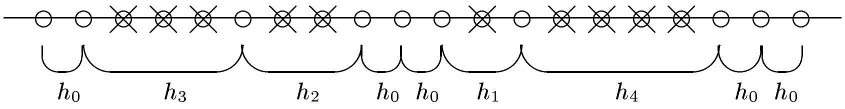


Fig. 2. The key chain of a 10-punctured 20-interval

In the key chain of $P = P_{i,j;x_1,\ldots,x_q}$, the key of a non-revoked user $u_k \in P$ is denoted by $K_{i,k;x_1,\ldots,x_t}$, where $i < x_1 < \cdots < x_t < k < x_{t+1} < \cdots < x_q$ and $0 \le t \le q \le p$. For examples,

$$K_{5,11} = h_0^6(K_{5,5})\,;\; K_{5,11;7} = h_0^3 h_1 h_0(K_{5,5})\,;\; K_{4,11;5,6,7,9,10} = h_2 h_3(K_{4,4})\,;$$
$$K_{3,11;4,5,7,8} = h_0^2 h_2^2(K_{3,3})\,;\; K_{3,11;4,5,6,7,9} = h_0 h_1 h_4(K_{3,3})\,;\; \ldots.$$

The center assigns these keys to users so that the user u_k receives $K_{k,k}$ and all possible $K_{i,k;x_1,\ldots,x_t}$'s, where $i < x_1 < x_2 < \cdots < x_t < k$ with $0 \le t \le p$ and $2 \le k - i + 1 \le c$. Fig. 3 describes the key assignment in the scheme $(3\,;5)$-π for u_5 :

Encryption For each session, the center divides L into disjoint p-punctured c-intervals $P_1, \ldots, P_m \in \mathcal{S}_{(p\,;\,c)}$, whose union covers all the non-revoked users. Let $P = P_{i,j;x_1,\ldots,x_q}$ be one of P_μ's. The last key $K_{i,j;x_1,\ldots,x_q}$ of the key chain corresponding to P is called the *interval key* of P. Let's denote the interval key of P_μ by K_μ for each $\mu = 1, 2 \ldots, m$, just for convenience. Then the center broadcasts :

$$\langle\, \mathrm{info}_1, \mathrm{info}_2, \ldots, \mathrm{info}_m\,;\, \mathcal{E}_{K_1}(SK), \mathcal{E}_{K_2}(SK), \ldots, \mathcal{E}_{K_m}(SK)\,;\; E_{SK}(M)\,\rangle,$$

where info_μ is information on P_μ, the μ-th interval starting from u_{i_μ} and ending at u_{j_μ} with q_μ revoked users. For each μ, info_μ consists of $i_\mu, \ell_\mu, \ell_{\mu,1}, \ldots, \ell_{\mu,q_\mu}$,

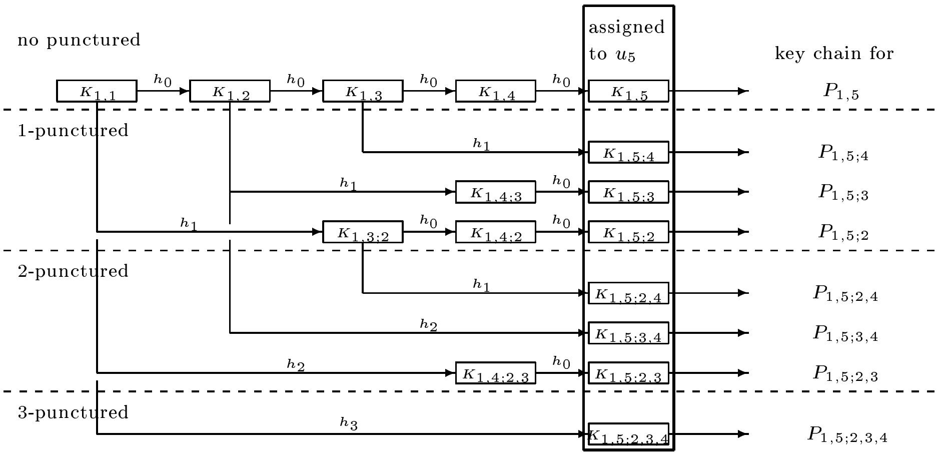


Fig. 3. One-way key chains starting from $K_{1,1}$, where $c = 5$

where $\ell_\mu = j_\mu - i_\mu + 1$ and $\ell_{\mu,1}, \ldots, \ell_{\mu,q_\mu}$ are the distances from u_{i_μ} to the first, $\ldots$, to the last revoked users of P_μ, respectively. The starting position i_μ can be represented by $\log N$ bits and the ℓ's are at most $\log c$ bits. So the size of all info's is $m(\log N + p \log c)$, which will be ignored when computing the transmission overhead because it is negligible compared to the size of all $\mathcal{E}_K(SK)$'s.

Decryption Receiving the encrypted message, each non-revoked user u_k first locates the punctured interval that he/she belongs using the info's. Let the punctured interval be $P_{i,j;x_1,\ldots,x_q}$, where $i \le k \le j, k \ne x_1, \ldots, x_q$. Then u_k can find $K_{i,j;x_1,\ldots,x_q}$ as follows:

- Find t for which $x_t < k < x_{t+1}$, where $0 \le t \le q$. Here, $t = 0$ and $t = q$ mean that there is no revoked user before and after u_k, respectively.
- Choose $K_{i,k;x_1,\ldots,x_t}$ from the assigned user keys.
- Starting from $K_{i,k;x_1,\ldots,x_t}$, apply one-way permutation h_i's under the rule described in Key Generation until the second subscript reaches to j.
- The resulting key is then $K_{i,j;x_1,\ldots,x_q}$.

With this, u_k decrypts $E_{K_{i,j;x_1,\ldots,x_q}}(SK)$ and $E_{SK}(M)$ to obtain the session key SK and the message M, respectively, in order.

2.3 Efficiency

We analyze efficiency - the transmission overhead (TO), the computation cost (CC) and the storage size (SS) - of the scheme $(p\,;c)$-π.

The transmission overhead of the scheme $(p\,;c)$-π is

$$\mathrm{TO}_{(p\,;\,c)}(N, r) = \left\lfloor \frac{r}{p+1} \right\rfloor + \left\lceil \frac{N - (p+2)\lfloor r/(p+1) \rfloor}{c} \right\rceil, \qquad (1)$$

where N and r are the total number and revoked users, respectively. Especially,

$$\mathrm{TO}_{(1\,;\,c)}(N,r) = \lfloor r/2 \rfloor + \left\lceil \frac{N - 3\lfloor r/2 \rfloor}{c} \right\rceil .$$

This occurs when $\circ \times \times$ is repeated from the leftmost user and then the remaining privileged users are on the right, where $\circ$ and $\times$ denote a privileged user and a revoked user, respectively. It is not hard to prove (1), but we omit the proof because it is long and tedious.

It is trivial that the computation cost $CC_{(p;c)}$ is at most $c-1$ computations of one-way permutations, which is independent of N and r. The storage size of each user can be easily computed as follows:

$$SS_{(p\,;\,c)} = \sum_{k=0}^{p} \left(\frac{1}{(k+1)!} \prod_{i=1}^{k+1} (c-i) \right) + 1,$$

which is also independent of N and r.

2.4 Security

Note that even a non-revoked user cannot compute the interval keys of the other punctured intervals. Those who do not belong to any punctured interval are the revoked ones and they can never access to the session key. Neither those revoked users who belong to punctured intervals can access to their interval keys because they cannot invert the one-way permutations.

The only way to compute the interval key $K_{i,j;x_1,\ldots,x_q}$ of $P_{i,j;x_1,\ldots,x_q}$ is to obtain one of the keys in the key chain. However, no revoked user is assigned a key in the key chain and hence they cannot compute the interval key even though they all collude. Furthermore, the interval keys of previous sessions when the user was not revoked do not help at all in the present session, in which he/she is revoked, because the revocation of him/her results in a totally new key chain.

2.5 Layered Punctured Interval Scheme

The scheme $(p\,;c)$-π is less efficient than SD when r is small. This is mainly because of long intervals consisting of non-revoked users which require several keys while covering no revoked users at all. To deal with this case, we introduce another set of user keys, each of which covers a long interval. To reduce the number of keys, we restrict the starting points of long intervals to some special *nodes* (users) on the line such that the distance between every neighboring nodes, called *node-distance* is c. This process can be repeated by $d-1$ more times taking special nodes with node distances are $c^2, c^3, \ldots, c^{d-1}$ or c^d, respectively, for a positive integer d. We call this scheme by *d-layered p-punctured c-interval* scheme or the $(p\,;c)$-π_d scheme.

Layered Structure As in the $(p\,;c)$-π scheme, the set of all N users are arranged on a long line L. Given a positive integer d ($< \log_c N - 1$), we consider d layers above the line L. The first layer L_1 consists of $N_1 = \lceil \frac{N}{c} \rceil - 1$ users $u_1, u_{c+1}, \ldots, u_{(N_1-1)c+1}$. Inductively, the t-th layer L_t consists of $N_t = \lceil \frac{N_{t-1}}{c} \rceil - 1$ users $u_1, u_{c^t+1}, \ldots, u_{(N_t-1)c^t+1}$ for $1 < t \leq d$. We define layered intervals of length c^t in the layer L_t by

$$LP_i^{(t)} = \{u_k | (i-1)c^t + 1 \leq k \leq ic^t\}. \tag{2}$$

Key Assignment First, the center assigns a random key $LK_i^{(t)}$ to $LP_i^{(t)}$ for each i and gives it to all members of $LP_i^{(t)}$. Next, it constructs a one-way key chain starting from $LK_i^{(t)}$. Let $g_1, \ldots, g_d : \{0,1\}^\ell \to \{0,1\}^\ell$ be one-way permutations and $h = h_0$ in $(p\,;c)$-π. Given k with $ic^t \leq k \leq (i+c-1)c^t$, $LK_{i,k}^{(t)}$ is defined by

$$LK_{i,k}^{(t)} = h^{e_0} g_1^{e_1} \cdots g_t^{e_t} (LK_i^{(t)}) \tag{3}$$

where $k - ic^t = e_t c^t + e_{t-1} c^{t-1} + \cdots + e_1 t + e_0$ $(0 \leq e_i < c)$ is a c-ary expansion of $k - ic^t$.

Let us consider the layered keys for the user u_k in the t-th layer. Assume $k = e_t c^t + \cdots + e_1 c + e_0$ for $0 \leq e_0, e_1, \ldots, e_{t-1} < c$ and $e_t \geq 0$. Then the center takes j with $e_t + 1 - (c-1) \leq j \leq e_t + 1$ and gives to the user u_k all the user keys $LK_{j;k_\tau}$ where $k_0 = e_0$ and $k_\tau = \lfloor (\frac{k}{c^\tau} + 1) \rfloor c^\tau$ for $1 \leq \tau \leq t$.

The center assigns these keys to the user u_k along with the interval keys for the scheme $(p\,;c)$-π. Hence the total number of keys for each user is

$$SS_{(p\,;\,c)} + \sum_{t=1}^{d} \{(c-1)(t+1)+1\} = SS_{(p\,;\,c)} + \frac{cd(d+3) - d(d+1)}{2}.$$

Encryption/Decryption If there is no layered interval consisting of all non-revoked users, the center encrypts the session key just as in the scheme $(p\,;c)$-π. Otherwise, we can save the transmission overhead by using layered keys. First the center marks all the layered intervals at each layer which has at least one revoked user as revoked intervals. Next, it finds the leftmost non-revoked interval, say $LP_i^{(d)}$, in the d-th layer. Then the session key is encrypted by $LK_{i,k}^{(d)}$, where u_{k+1} is the first revoked user after u_{ic^d} with $k \leq (i+c)c^d$. The center then marks all the users from $u_{(i-1)c^t+1}$ to u_k and the layered intervals containing at least one of them revoked. This process is repeated for the next non-revoked interval. If there is no non-revoked interval in the d-th layer, go to $(d-1)$-st layer and repeat the same procedure and so on. Finally, if all layered intervals at each layer are revoked, then the scheme $(p\,;c)$-π is applied for the remaining non-revoked users.

Note that each non-revoked user u_k can decrypt the session key by an interval key of $(p\,;c)$-π or a layered key. In order to obtain the key (to decrypt the session key) it costs at most $c-1$ and $t(c-1)$ computations of one-way permutations, respectively. Hence the computation cost is at most $d(c-1)$ computations of one-way permutations.

Transmission Overhead First we estimate the transmission overhead for $(p\,;c)$-π_1. If there is no revoked user, then $\lceil \frac{N}{c^2} \rceil$ layered intervals cover entire straight line L. By inserting one revoked user to an interval, the interval is divided to at most 3 intervals including punctured or long intervals. So the transmission overhead is at most $\lceil \frac{N}{c^2} \rceil + 2r$. Trivially, the transmission overhead of this scheme cannot be larger than that of punctured interval scheme. So we can conclude that the transmission overhead is at most $\lceil \frac{N}{c^2} \rceil + 2r$ for $r \le \frac{2}{3}\frac{N(c-1)}{c(c+1)}$ or $\lceil \frac{r}{2} \rceil + \left\lceil \frac{N-\lceil 3r/2 \rceil}{c} \right\rceil$ otherwise. The transmission overhead of $(p\,;c)$-π_d for $d \ge 2$ can be similarly estimated.

3 Revocation Scheme with a Ring Structure

In this section, we present revocation schemes using combination of (punctured) one-way chains and ring structures. We can further reduce user key storage using a hierarchical ring structure. In the following schemes we use relatively simple key assignment applying one permutation to a random label instead of several one-way permutations.

3.1 Revocation Scheme with a Simple Ring Structure

Initially we assume that N nodes (users) are arranged on a ring in a clockwise direction. We denote by u_i the i-th node from the initial node in a clockwise direction and identify u_i and u_j if and only if $i \equiv j \pmod N$. For two nodes u_i and u_j, we set $S_{[i,j]} = \{u_i, u_{i+1}, \cdots, u_{j-1}, u_j\}$. Let $\mathcal{C}_{[i,j]}$ denote a *one-way chain* consisting of all users in $S_{[i,j]}$ that starts from u_i and ends at u_j. For a given one-way permutation $h : \{0,1\}^\ell \to \{0,1\}^\ell$ and an input value $\mathsf{sd} \in \{0,1\}^\ell$, the *chain-value* of $\mathcal{C}_{[i,j]}$ is defined by the value $h^k(\mathsf{sd})$, which is computed by applying h to sd iteratively $k(=j-i+1 \mod N)$ times.

Key Assignment First, to each node u_i on the ring, a random and independent label $\mathsf{L}_i \in \{0,1\}^\ell$ is selected and assigned. Then the center computes $h^k(\mathsf{L}_i)$, $(1 \le k \le N-1)$ and assigns $h^k(\mathsf{L}_i)$ to each node u_{i+k}. Finally the center provides the user u_t with a set of N keys,

$$\{\mathsf{GSK}_0, h^1(\mathsf{L}_{t-1}), h^2(\mathsf{L}_{t-2}), \cdots, h^{N-1}(\mathsf{L}_{t-(N-1)=t+1 \mod N})\},$$

where GSK_0 is an initial group session key.

Encryption For a given set of revoked users $R = \{u_{i_1}, u_{i_2}, \cdots, u_{i_r}\}$, the center partitions the remaining legal users into r subsets $S_{[i_1+1, i_2-1]}$, $S_{[i_2+1, i_3-1]}$, $\cdots$, $S_{[i_{r-1}+1, i_r-1]}$, $S_{[i_r+1, i_1-1]}$. If u_{i_k} and $u_{i_{k+1}}$ in R are adjacent on the ring, then there is no privileged user between u_{i_k} and $u_{i_{k+1}}$ and the subset $S_{[i_k+1, i_{k+1}-1]}$ is empty. For example, if four users u_3,u_6,u_7,u_{11} are revoked, the set of remaining privileged users is partitioned into 3 non-empty subsets $S_{[4,5]}$, $S_{[8,11]}$ and $S_{[12,2]}$ (see Fig. 4). For each non-empty subset $S_{[i,j]}$, the center assigns a one-way chain $C_{[i,j]}$ and computes its chain-value $h^k(\mathsf{L}_{i-1})$ where $k{=}j-i+1 \mod N$. Because there are r arcs in the worst case and a one-way chain is assigned to each arc, the transmission overhead is r.

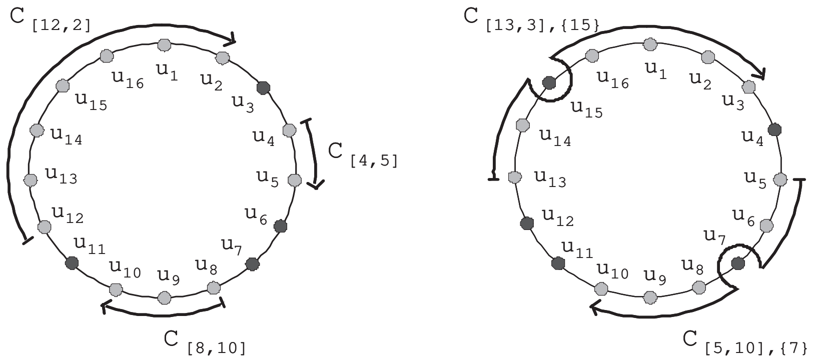
Decryption Upon receipt of a broadcast message, each privileged user u_t finds a subset $S_{[i,j]}$ including u_t from the information of indices of revoked users. We note that the subset including u_t is unique. To find the subset $S_{[i,j]}$, u_t performs a binary search on the sequence of indices of the revoked users. Then, by using a value $\mathsf{KV}{=}h^{(t-i+1) \mod N}(\mathsf{L}_{i-1})$ given initially, u_t computes a key $h^{j-i+1 \mod N}(\mathsf{L}_{i-1})$ by applying h to KV ($j-t \mod N$) times. Each user should compute function h, in the worst case, $N-1$ times.

Basically revoked users are cannot obtain useful any information to decrypt the encrypted session key because of one-wayness of h. However, we should show that the basic scheme is resilient to collusion of any set of revoked users. We can show that the security of the basic scheme is as strong as that of the SD method in [13] by using the following lemma.

Lemma 1. *The above key assignment satisfies the* key-indistinguishability *condition [13] under the pseudo-randomness of a given function h.*

By using a standard hybrid argument on the length of one-way chains we can prove the lemma under the pseudo-randomness of a given function h as in [13]. We omit the proof here.

Fig. 4. Revocation in $\mathsf{OFBE}(N{:}1)$ and $\mathsf{OFBE}(N{:}2)$ for $N{=}16$



Now we present a method to further reduce the number of transmission messages in the basic scheme below r. The basic idea is to cover several subsets of privileged users separated by revoked users in the basic scheme by using only one key. This makes the number of messages transmitted drastically goes below

r. For a set of users D, let $S_{[i,j],D}$ denote a difference set $S_{[i,j]} \backslash D$, i.e., $\{u \mid u{\in}S_{[i,j]}$ and $u{\notin}D\}$. We define a 'jumping' one-way chain $\mathcal{C}_{[i,j],D}$ for $S_{[i,j],D}$ such that, for two nodes u_i and u_j ($u_i{\neq}u_j$) and a subset $D{=}\{u_{k_1},...,u_{k_s}\}$ of $S_{[i,j]}$, it starts from u_i, proceeds in a clockwise direction, but jumps over the nodes $u_{k_1},...,u_{k_s}$, and ends at u_j. First we concentrate on the case that D contains only one revoked user.

Key Assignment To provide a user with an initial key-set, the center performs the followings : First the center performs the key assignment in the basic scheme. Next, for every pair of (potential revoked) users u_i and u_j not adjacent each other on the ring, the center additionally chooses a random and independent label $\mathsf{L}_{i,j}$ ($1{\leq}i{<}j{\leq}N$). We exclude the cases that u_i and u_j are adjacent on the ring since those cases are covered by one-way chains of the basic scheme. Then the center computes chain-values $h^k(\mathsf{L}_{i,j})$, ($1{\leq}k{\leq}N-1$) and assigns it to each node u_{i+k}, ($1{\leq}k{\leq}N-1$) except u_i and u_j. In this case, the total number of the keys which a user should store is $N{+}\binom{N-1}{2}{-}(N-2) = \binom{N-1}{2}{+}2 = \frac{(N-1)(N-2)}{2}{+}2$ since the number of keys assigned by the basic scheme is N, the number of cases choosing two different users in the remaining $N-1$ users is $\binom{N-1}{2}$, and the number of cases covered by keys assigned in the basic scheme is $N-2$.

Encryption To revoke users, the center constructs one-way chains as follows. Starting from any remaining user in a clockwise direction on the ring, the first one-way chain proceeds until it meets the first revoked user u_{i_1}. If the next revoked users u_{i_2} in a clockwise direction is adjacent to u_{i_1}, the chain ends at u_{i_1-1}, just before u_{i_1}. Otherwise the chain jumps over u_{i_1} and continues until it meets u_{i_2}, and ends at u_{i_2-1}, just before u_{i_2}. This process is repeated until all remaining users are covered by one-way chains. For example as in Fig. 4, suppose that users u_4, u_7, u_{11}, u_{12}, and u_{15} are to be revoked. Starting from u_5, the first one-way chain proceeds in clockwise directions, jumps u_7, and ends at u_{10}. The second one-way chain starts at u_{13}, jumps over u_{15}, and ends at u_3. Hence all remaining users are covered by two chains in this example.

By applying the method for r revoked users $\{u_{i_1},...,u_{i_r}\}$, the center broadcasts at most $\lfloor\frac{1}{2}\cdot r\rfloor{+}1$ encrypted keys where $\lfloor\cdot\rfloor$ is a floor function. If r is even, $\frac{1}{2}\cdot r$ one-way chains sufficiently cover the remaining privileged users at worst case. If r is odd then we need to cover a last subset $S_{[u_{i_r+1},u_{i_1-1}]}$, which is not covered by directly assigning 'jumping' one-way chain. In this case, we use a one-way chain $\mathcal{C}_{[i_r+1,i_1-1]}$ of the basic scheme.

Extension The above cover strategy can be generalized, i.e., naturally extended to cover k subsets by only one key. We denote this method by $\mathsf{OFBE}(N{:}k)$ where N is the number of users.

In $\mathsf{OFBE}(N{:}k)$, the cover strategy is similar to that of $\mathsf{OFBE}(N{:}2)$. Starting from any remaining user u in a clockwise direction on the ring, the center jumps $k-1$ revoked users until it meets the k-th revoked user u_{i_k}. Next, the center goes back in counterclockwise direction to find the first remaining user u'. The first one-way chain starts from u and ends at u', jumping revoked users between

u and u' in a clockwise direction. This process is continued until all remaining users are covered. All remaining users are covered by at most $\lfloor \frac{1}{k} \cdot r \rfloor + 1$ chains.

To assign an initial key-set to each user, the center generates a label L_A for every subset $A = \{u_{i_1},...,u_{i_k}\}$ with k users. Then the center computes $h^t(\mathsf{L}_A)$, $(1 \leq t \leq N-1)$ and gives it to node u_{i_1+t} not in A. To avoid double key assignment, we exclude the cases which can be covered by a key generated in $\mathsf{OFBE}(N{:}j)$ where $0<j<k$. Hence the number of possible selections is $\mathsf{Num}(N,k) = \binom{N-1}{k} - \binom{N-1}{k-1} + 1$ where $\binom{N-1}{0}=0$. In $\mathsf{OFBE}(N{:}k)$, a key assignment method is performed recursively. Hence the number of keys which a user should store in $\mathsf{OFBE}(N{:}k)$ is $\#\mathsf{OFBE}(N{:}k) = \#\mathsf{OFBE}(N{:}k-1) + \mathsf{Num}(N,k) = \binom{N-1}{k} + k$ where $\#\mathsf{OFBE}(N{:}j)$ is the number of keys for each user in $\mathsf{OFBE}(N{:}j)$.

3.2 Revocation Scheme with a Hierarchical Ring Structure

In this subsection we generalize the basic scheme $\mathsf{OFBE}(N{:}1)$ by extending a simple ring structure to a hierarchical one. A *hierarchical ring* R is recursively defined such that there is an outmost ring R_0 in level 1 having nodes on the ring, each node on R_0 has children arranged on a ring in level 2, and each node in level 2 again has children on a ring in level 3, and so on. A node which does not have any child is called a *leaf*. The *depth* of a hierarchical ring is the highest level of a leaf in the ring. A hierarchical ring R is called w*-ary* if each node in R has w children.

Underlying Scheme of Our Generalization. For our generalization, we first describe an underlying revocation scheme with a relatively simple hierarchical ring R with depth 2 such that there are two nodes u_1 and u_2 in level 0 (i.e., on the outmost ring). Each u_b for $b \in \{1,2\}$ has m children $\{u_{b.1},..,u_{b.m}\}$ on a sub-ring R_b, where $u_{b.j}$ denotes the j-th user from $u_{b.1}$ on ring R_b. Hence the number of users in R are $2 \cdot m$. We denote this scheme by $\mathsf{HOC}(2,m)$.

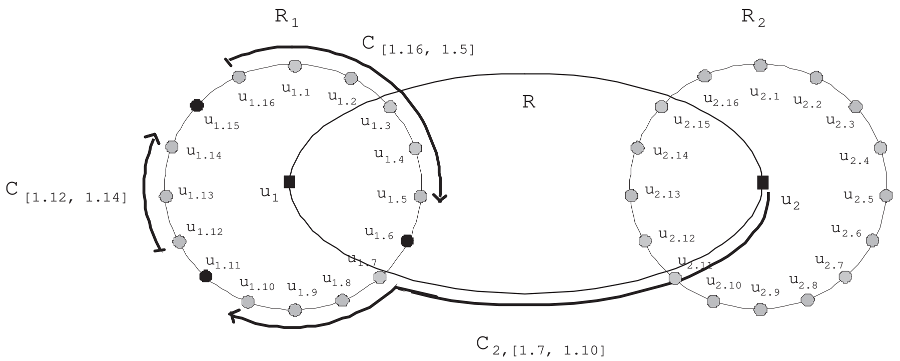
Let $S_{\bar{b},[b.i,b.j]}$ denote a union set $\mathsf{R}_{\bar{b}} \cup S_{[b.i,b.j]}$, i.e., $\{u | u \in S_{[b.i,b.j]}$ and $u \in \mathsf{R}_{\bar{b}}\}$ where $\bar{b}=2^{(b \mod 2)}$. We define a simple 'hierarchical' one-way chain $\mathcal{C}_{\bar{b},[b.i,b.j]}$ for $S_{\bar{b},[b.i,b.j]}$ such that it starts from a node $u_{\bar{b}}$, goes through children $u_{b.i}, u_{b.i+1}, ..., u_{b.j-1}$ of u_b in a clockwise direction and ends at $u_{b.j}$ on R_b. $\mathcal{C}_{\bar{b},[b.i,b.j]}$ is used to cover users on two separated rings by one key, i.e., all users on $\mathsf{R}_{\bar{b}}$ and some users on R_b. For example as in Fig. 5, suppose that $u_{1.6}$, $u_{1.11}$, and $u_{1.15}$ on R_1 are to be revoked. the center partitions the remaining privileged users into disjoint subsets $\mathsf{R}_2 \cup S_{[1.7,1.10]}$, $S_{[1.12,1.14]}$ and $S_{[1.16,1.5]}$. Then the center assigns one-way chains $\mathcal{C}_{2,[1.7,1.10]}$, $\mathcal{C}_{[1.12,1.14]}$, $\mathcal{C}_{[1.16,1.5]}$ to those subsets, respectively.

Key Assignment To exploit the previous hierarchical cover strategy using two types of one-way chains, two types of labels are selected by the center initially: One is a label used in the basic scheme. The other is a label L_b associated to a node v_b, which is used to compute a chain-value corresponding to a hierarchical one-way chain. In this case, we use a cryptographic pseudo-random sequence generator $G{:}\{0,1\}^\ell \rightarrow \{0,1\}^{m\ell}$. We denote by $G(\mathsf{L})_i$ the i-th output block of length ℓ of G on L. A similar function which triples an input is used in the SD scheme [13] to achieve a similar purpose.

The center provides a user $u_{b.t}$ with n keys by using the following key assignment method: First the center performs the key assignment of the basic scheme to cover only privileged users in one sub-ring of m users. That is, for each $b \in \{1,2\}$ and $u_{b.i} \in \mathsf{R}_b$, a label $\mathsf{L}_{b.i} \in \{0,1\}^\kappa$ is selected. Then the center computes $h^k(\mathsf{L}_{b.i})$, $(1 \le k \le m-1)$ and assigns it to $u_{b.i+k}$. Next, for each $b \in \{1,2\}$, a label $\mathsf{L}_b \in \{0,1\}^\ell$ is selected. Then the center computes $h^k(G(h(\mathsf{L}_b))_i)$, $(1 \le k \le m-1)$ and assigns it to $u_{\bar{b}.i+k}$. Finally the center provides $u_{b.t}$ with a set of the following keys;

$$\{\ \mathsf{GSK}_0,\ h(\mathsf{L}_{b.j-1}), h^2(\mathsf{L}_{b.j-2}), \cdots, h^{m-1}(\mathsf{L}_{b.(j-(m-1)=j+1 \mod m)}),$$
$$h(\mathsf{L}_{\bar{b}}),\ h(G(h(\mathsf{L}_b))_{j-1}), h^2(G(h(\mathsf{L}_b))_{j-2}), \cdots, h^{m-1}(G(h(\mathsf{L}_b))_{j-(m-1)=j+1 \mod m})\}$$

Fig. 5. Revocation in $\mathsf{HOC}(2,m)$



Encryption To revoke users, the center covers the remaining legal users as follows: If all revoked users are included in only one sub-ring R_b, the center generates one-way chains $\mathcal{C}_{\bar{b},[b.i_1+1,b.i_2-1]}$ and $\{\mathcal{C}_{[b.i_2+1,b.i_3-1]}, \cdots, \mathcal{C}_{[b.i_{r_b}+1,b.i_1-1]}\}$. Otherwise, the center performs the revocation method of the basic scheme on each sub-ring. Then the center encrypts a group session key GSK by using chain-values as keys and broadcasts the ciphertexts. In particular, for a chain-value of the hierarchical one-way chain $\mathcal{C}_{\bar{b},[b.i,b.j]}$, the center computes $h^{j-i+1 \mod m}(G(h(\mathsf{L}_b))_{i-1})$. The number of messages to be transmitted is at most r as in the basic scheme.

Security The security of this scheme depends on the association of the securities of two functions, h and G. However, without considering the association, we can use one function instead of two independent functions. In this case, for the expansion of labels, we can define the output of h as the first output block of G, namely, $h(\mathsf{L}) = G(\mathsf{L})_1$ for $\mathsf{L} \in \{0,1\}^\ell$. Using the similar idea in [13] we can prove the security under the pseudo-randomness of G.

Generalization. Naturally, we can extend the previous $\mathsf{HOC}(2, m)$ by allowing more levels and more nodes in each level. In general schemes the center uses a hierarchical one-way chain traversing nodes in many levels, which starts from a node of level d and goes through some nodes of the same level in a clockwise direction and ends at u_j, and comes down to a children node of u_{j+1} in level $d+1$ and iterates this process. For space limitation, we omit the concrete explanation of the general scheme.

One important thing is that there is a trade-off relation between transmission overhead and keys stored at a user. Using the deepest hierarchical ring structure such as a complete binary ring of depth $\log_2 n$, we gain reduction in user storage up to $\frac{\log_2^2 n+\log_2 n}{2}+1$ while the transmission overhead increases up to $2r$. Interestingly, the revocation scheme for a binary (2-way) ring is structurally equivalent to SD based on a binary tree proposed by Naor et. al [13]. The forms of subsets to be covered in both schemes are equivalent, since a binary ring is structurally equivalent to a binary tree.

To further reduce the transmission overhead, we can use 'jumping' hierarchical one-way chains which jump potential revoked users and cover several subsets of privileged users separated by revoked users at a time. Our generalization provides a generic method for any application in terms of setting specific values for parameters to achieve a desirable trade-off.

3.3 Specific Revocation Scheme

In this section we present a specific revocation scheme where the transmission overhead is less than r while storage requirement per user is reasonably low. First we notice that different $\mathsf{OFBE}(N{:}k)$ schemes can be independently combined with revocation schemes using hierarchical one-way chains. In particular $\mathsf{OFBE}(m{:}2)$ can be directly applied to each sub-ring R_b of $\mathsf{HOC}(2, m)$, which contains $m=\frac{N}{2}$ users. In this case, the number of keys which a user should store reduces from $\frac{N^2-3N+6}{2}$ to $\frac{m^2-m+6}{2}$ while the transmission overhead is still at most $\lfloor\frac{1}{2}\cdot r\rfloor+1$. We denote this method by $\mathsf{HOC}(2,[m{:}2])$.

Lemma 2. *In HOC(2,[m:2]), the number of keys stored by a user is* $\frac{m^2-m+6}{2}$ *and the transmission overhead is at most* $\lfloor\frac{1}{2}\cdot r\rfloor+1$*, for* $N = 2m$.

Though, in $\mathsf{HOC}(2,[m{:}2])$, the number of keys storage per user is reduced while the number of transmission messages is still at most $\lfloor\frac{1}{2}\cdot r\rfloor+1$, the method is still not applicable for a large number of group users. To reduce user storage further with slight increase in transmission complexity, we can use divisional approach as follows: First partition a group of N $(=2m{\cdot}s)$ users into s subgroups and then apply $\mathsf{HOC}(2,[m{:}2])$ to each sub-group of $2m$ users where m is a predetermined constant. We denote this method by $\mathsf{HOC}(2,[m{:}2])_{\mathsf{p}}$. In this case, the transmission overhead is $\lfloor\frac{1}{2}\cdot r\rfloor+\frac{N}{2m}$ at worst case and each user should store $\frac{m^2-m+6}{2}$ keys.

4 Comparison

Table 1 shows the complexity of the storage sizes, the transmission overhead and the computation costs of our schemes, SD and LSD when $N = 10^8$ and r is $0.1, 0.5, 1, 5, 10$ and 20% of N. In the table, we assume that the size of a user key is 128 bits, which is considered reasonably secure, currently.

Table 1. Examples when $N = 10^8$

Scheme	Storage	TO (Mbits)						CC
r revoked	(KBytes)	0.1%	0.5%	1%	5%	10%	20%	
$(0\,;100)$-π	1.60	141	191	253	755	1380	2640	100
$(1\,;100)$-π	79.2	134	159	190	438	749	1370	100
$(0\,;100)$-π_1	4.80	26.9	129	253	755	1380	2640	198
$(1\,;100)$-π_1	82.4	26.9	129	190	438	749	1370	198
$\mathsf{HOC}(2,[100:2])_{\mathsf{p}}$	79.2	70.4	96	128	384	704	1344	100
$\mathsf{HOC}(2,[50:2])_{\mathsf{p}}$	19.6	134.4	160	192	448	768	1408	50
SD	5.8	25.6	128	256	1280	2560	5120	27
LSD	2.3	51.2	256	512	2560	5120	10240	27

Figure 6 shows the comparison of the *worst-case* transmission overheads by graphs when the revocation rate ranges from 0% to 3%. Among the graphs, the dotted line represents the transmission overhead of the scheme $(1\,;100)$-π_1. The dotted graph is very close to that of SD for small r. It has steeper slope than the graph of $(1\,;100)$-π, but a lower y-intercept at $\lceil \frac{N}{c} \rceil$. As we mentioned above, the layered π scheme improves the transmission overhead when the revocation rate is small. For large r, it has the same transmission overhead as that of the scheme $(p\,;c)$-π.

The transmission overhead of $\mathsf{HOC}(2,[50:2])_{\mathsf{p}}$ or $\mathsf{HOC}(2,[100:2])_{\mathsf{p}}$ is relatively larger than that of SD at initial interval where the number r of revoked users is smaller than 0.75 % of the total users. But, except this interval, the transmission overhead of $\mathsf{HOC}(2,[50:2])_{\mathsf{p}}$ becomes, at worst case, about $\frac{1}{3.5}$ of the transmission overhead of SD. This should be a good trade-off in most applications since the number of initial messages is relatively small. The number of keys per user in $\mathsf{HOC}(2,[50:2])_{\mathsf{p}}$ is about 3.5 times as many as that of SD as the number of revoked users increases. But this difference may be acceptable in many practical applications.

5 Conclusion

In this paper, we proposed efficient broadcast encryption schemes based on linear and circular structures. Introducing the idea of punctured one-way chain to

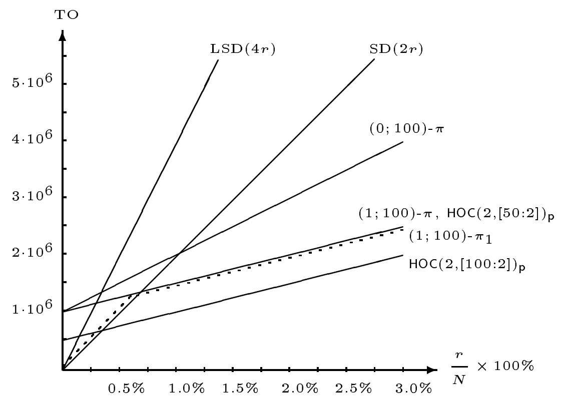


Fig. 6. TO for $N = 1 \cdot 10^8$ in the worst case

these strucutres, we could reduce the transmission overhead below r. Particularly, our specific schemes have about 1/3 transmission overhead than SD while maintaining the computation cost and the storage size in a reasonable bound.

Moreover our methods provide many flexibility on the system efficiency. The system can be optimized to have best efficiency for any of the three parameters of broadcast encryption the transmission overhead, the computation cost and the storage size.

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. D. Boneh and A. Silverberg, *Applications of Multilinear Forms to Cryptography*, Contemporary Mathematics 324, American Mathematical Society, pp.71-90.
4. B. Chor, A. Fiat and M. Noar, *Tracing Traitors*, Advances in Cryptology CRYPTO'94, Lecture Notes in Computer Science 839, pp. 257-270.
5. G. Chick and S. Tavares, *Flexible access control with master keys*, Advances in Cryptology - Crypto'89, Lecture Notes in Computer Science, pp.316-322.
6. 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.
7. A. Fiat and M. Naor, *Broadcast Encryption*, Advances in Cryptology - Crypto'93, Lecture Notes in Computer Science 773, pp.480-491.
8. 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.
9. 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.

10. E. Gafni, J. Staddon and Y.L. Yin, *Efficient Methods for Integrating Traceability and Broadcast Encryption*, Advances in Cryptology - CRYPTO'99, Lecture Notes in Computer Science 1666, pp.372-387.
11. D. Halevi and A. Shamir, *The LSD Broadcast Encryption Scheme*, Advances in Crytology - Crypto'02, Lecture Notes in Computer Science 2442, pp.47-60.
12. 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.
13. 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.
14. M. Naor and B. Pinkas, *Efficient Trace and Revoke Schemes*, Financial Cryptography'00, Lecture Notes in Computer Science.
15. C.K. Wong, M. Gouda and S.S. Lam, *Secure Group Communication using Key Graphs*, ACM SIGGCOM'98 ACM.
16. M. Luby and J. Staddon, *Combinatorial Bounds for Broadcast Encryption*, Advances in Cryptology - Eurocrypt'98, Lecture Notes in Computer Science 1403, pp.512-526.