Efficient Broadcast Encryption Scheme with Log-Key Storage[†]

Yong Ho Hwang and Pil Joong Lee

Dept. of Electronic and Electrical Eng., POSTECH, Korea.
yhhwang@oberon.postech.ac.kr,pjl@postech.ac.kr

Abstract

In this paper, we present a broadcast encryption scheme with efficient transmission cost under the *log-key* restriction. Given n users and r revoked users, our scheme has the transmission cost of $O(r)$ and requires the storage of $O(\log n)$ keys at each receiver. These are optimal complexities in broadcast encryptions using one-way hash functions (or pseudo-random generators.) To achieve these complexities, the stratified subset difference (SSD) scheme and the $\overline{\mathsf{B1}}$ scheme were introduced by Goodrich *et al.* and Hwang *et al.* respectively. However, their schemes have the disadvantage that transmission cost increases linearly according to the number of stratifications. By assigning the related keys between stratifications, our scheme remedies the defect and achieves very efficient transmission cost even in an environment where the key storage is restricted. To the best of our knowledge, our scheme has the most efficient transmission cost in the existing schemes with *log-key* storage. In addition, our result is comparable to other schemes that allow a large key storage.

1 Introduction

Broadcast encryption is an encryption scheme that enables a center to securely distribute messages to a dynamically changing group of users over an insecure channel, where only predetermined users can obtain available information. The center should efficiently deliver information to the group of legitimate users and prevent the group of revoked users from decrypting transmitted messages. There are various practical applications such as pay-TV, multicast communication, satellite-based commerce, and distribution of copyrighted materials (CD/DVD, etc). In this area, an important requirement is for *stateless* receivers, which cannot update their original state, i.e., they are not capable of recording the past history of transmission and changing their state accordingly. Hence, each receiver must be able to decrypt the current transmission with only its initial configuration. Actually, in many practical environments most devices should be *stateless* since it is difficult to keep the receiver constantly on-line and it is very cumbersome for both the receiver and the center to keep the history of every transmission.

With the advent of mobile networks and other digital support services, the need to deliver multimedia data to user's handheld devices over a wireless network becomes more important. This situation is more intricate since handheld devices such as cellular phones and PDAs have only a small storage capability and low computing power. In addition, the bandwidth of wireless networks is narrower than that of wired networks. Therefore, we need an efficient broadcast encryption scheme to overcome these obstacles.

[†]This research was supported by University IT Research Center Project, the Brain Korea 21 Project, and grant No. R01-2005-000-10713-0 from the research program of KOSEF.

Related Works. The notion of broadcast encryption was first discussed by Berkovits [5]. Fiat and Naor [11] formalized the basic definitions and proposed a systematic paradigm. However, their scheme is difficult to apply to a practical system because it is highly complex. After the multicast scheme based on a *logical tree hierarchy* was independently introduced by Wallner *et al.*[21] and Wong *et al.*[22], various schemes [1, 2, 19, 15, 12] based on a tree structure were suggested. There are two approaches to construct an efficient tree-based scheme. One is a scheme based on sequential one-way hash functions (or pseudo-random generators)[19, 15, 12] and the other is based on the RSA accumulator [1, 2]. One-way hash function-based schemes have various trade-offs between $O(r)$ transmission cost and $O(\log n)$ key storage where n is the number of users and r is the number of revoked users. While RSA accumulator-based schemes can reduce key storage to $O(1)$, their transmission cost depends on n.[i] Moreover, these schemes require expensive computations such as modular exponentiation and prime number generation. We deal with one-way function based schemes in this paper.

In 2001, Naor *et al.*[19] introduced a *Subset-Cover* framework and designed two broadcast encryption schemes for stateless receivers under this framework. One is the CS (Complete Subtree) scheme which requires $O(r \log n/r)$ transmission cost and $O(\log n)$ key storage, and the other is the SD (Subset Difference) scheme which guarantees $2r - 1$ transmission cost and $O(\log n)$ key computation cost, while each user should store $O(\log^2 n)$ keys. The transmission cost of $O(r)$ and the key storage of $O(\log n)$ have been regarded as the optimal bounds of tree-based schemes, which use the key assignment technique of sequentially applying a one-way function (or a pseudo-random generator). Afterwards, a number of papers tried to reduce the storage size by sacrificing the transmission cost of the SD scheme. Halevy and Shamir [15] proposed the LSD (Layered Subset Difference) scheme that lowers the key storage to $O(\log^{1+\epsilon} n)$ while maintaining $O(r)$ transmission cost by labelling special layers in a binary tree. In addition, Goodrich *et al.* [12] presented the SSD (Stratified Subset Difference) scheme that can lower the transmission cost to $O(r)$ with $O(\log n)$ key storage by stratifying subtrees between special layers in a binary tree. The SSD scheme seems to be able to achieve the lower bounds of both the transmission cost and the key storage in tree-based schemes using one-way functions. However, the LSD scheme and the SSD scheme linearly increase the transmission cost according to the number of layers or stratified subtrees, although key storage does approach the $O(\log n)$ bound. Other interesting improvements were introduced in [4] and [18]. In [4] the key storage of the SD scheme and the LSD scheme were slightly reduced by the sequential key derivation method while maintaining their transmission costs. In [18] the system complexity was adjusted by a hybrid structure based on the CS, SD, and LSD schemes. Moreover, other variants related to broadcast encryption have been investigated in [8, 6, 13, 20, 10, 17].

Recently, new broadcast encryption schemes based on a hash-chain [16] were proposed which can reduce the transmission cost below r by exploiting the tradeoff between the transmission cost and the key storage. In doing so, however, too much secure memory must be sacrificed. For example, the transmission cost of these schemes is similar to that of the SD scheme when key storage is bounded as in the SD scheme. This approach seems useful in practical applications, since the storage size of user's devices, even in the case of cellular phones or PDAs, seems to no longer be a problem because storage devices have become larger and cheaper. However, to guarantee security, user keys must be securely stored in tamper-proof storage devices, which are still small and expensive. To solve this problem, Hwang *et al.* [14] introduced a compiler that made scalable broadcast encryption schemes by transforming ones that had impractical computation costs or key storage requirements when there are huge numbers of users. They applied a given broadcast encryption scheme to a relatively small

[i] Recently, an RSA accumulator-based scheme with transmission cost independent of n was accepted by Asiacrypt 2005[3]. However, this scheme also has the disadvantage that transmission cost linearly increases according to the number of stratifications, like the SSD scheme.

subset in a hierarchical and independent manner. Their compiler make the computation cost and the key storage reasonable by slightly increasing the transmission cost. However, their compiler also does not achieve $O(r)$ transmission cost when users are holding strictly resource-restricted devices.

In addition, Boneh *et al.* [7] introduced a *public key* broadcast encryption scheme with $O(1)$ for both the transmission cost and the private key. Their scheme requires $O(n)$ non-secure key storage and $O(n-r)$ computation cost. To achieve reasonable storage and computation cost, they constructed a general scheme divided into a number of subsets. This scheme has $O(\sqrt{n})$ transmission cost and $O(\sqrt{n})$ key storage. Consequently, their complexity is not independent of n.

Our Contribution. In this paper, we focus on stateless receivers which can store at most $O(\log n)$ keys since it is actually difficult to store much data in tamper-proof storage. We refer to this as the *log-key* restriction. We propose a new broadcast encryption scheme which satisfies $O(r)$ transmission cost and $O(\log n)$ key storage at a reasonable computation cost. Our scheme has the most efficient transmission cost under the *log-key* restriction. Table 1 shows the comparison between schemes with $O(\log n)$ key storage per user.

Table 1: Complexity of BE schemes with $O(\log n)$ key storage.

	Transmission cost	(Bound)	Key storage	Computation cost
CS [19]	$O(r \log n/r)$		$O(\log n)$	$O(\log \log n)$
SSD [12]	$O(r)$	$4kr$	$O(\log n)$	$O(n^{1/k})$
$\overline{\mathsf{B1}}$ [14]	$O(r)$	$2sr$	$O(\log n)$	$O(n^{1/s})$
Our scheme	$O(r)$	$2r$	$O(\log n)$	$O(n^{1/d})$

(k, s, and d are system parameters which mean the number of stratified subsets to obtain a reasonable computation cost.)

In [14], Hwang *et al.* introduced the B1 scheme with the computation cost proportional to n and transformed it to the $\overline{\mathsf{B1}}$ scheme, which has a practical computation cost and *log-key* storage, by their compiler. Our scheme is also based on the B1 scheme and extends it in a hierarchical manner to a scheme with at most $2r$ transmission cost under the *log-key* restriction. To achieve a transmission cost free of the level of stratification, our scheme additionally assigns the related keys between stratifications to the $\overline{\mathsf{B1}}$ scheme. There is a trade-off between the key storage and the computation cost in our scheme. Consequently, while our scheme reduces a upper bound of the transmission cost to $2r$, $(d + \frac{d+1}{2} \cdot \log n)$ key storage and $(d \cdot n^{1/d})$ computation cost are required.

Organization of The Paper. The remainder of this paper is organized as follows. In Section 2, we formalize a model for a broadcast encryption scheme based on a *Subset-Cover* framework. In Section 3, we first introduce our basic scheme and propose the complete scheme based on it. Then we discuss the performance and the properties of our scheme in detail and compare it with various broadcast encryption schemes in Section 4. Finally, we give concluding remarks in Section 5.

2 Model for Broadcast Encryption

We define a model for a broadcast encryption based on the *Subset-Cover* framework introduced by Naor *et al.*[19] since our scheme is also based on it.

2.1 Generic Model

In broadcast encryption the center (or the broadcaster) assigns secret keys to all users and broadcasts a encrypted message with the subset keys. Legitimate users can derive the subset keys from the assigned secret keys and decrypt the ciphertext with them. Let $\mathcal{N}$ be the set of all users, $\mathcal{R}$ the set of revoked users, and $\mathcal{N}\backslash\mathcal{R}$ the set of remaining users. We suppose that $|\mathcal{N}| = n$ and $|\mathcal{R}| = r$. A *broadcast encryption scheme* BE consists of 3 phases (Setup, Broadcast, Decryption):

- Setup: The center generates secret keys for each user and delivers them to each user over a secure channel.
- Broadcast: In this phase, the center broadcasts a message to users. Given $\mathcal{R}$, the center divides $\mathcal{N}\backslash\mathcal{R}$ into disjoint subsets $S_1, \ldots, S_m$ so that $\mathcal{N}\backslash\mathcal{R} = \bigcup_{i=1}^{m} S_i$, and computes a subset key sk_i for each subset S_i. At this time, sk_i is generated by a pre-defined algorithm. The center chooses a session key K at random and encrypts it m times with $sk_1, \ldots, sk_m$. In addition, an *"actual"* message M is encrypted with K. The center broadcasts a ciphertext $\langle \mathsf{Hdr}, \mathtt{Enc}_K(M) \rangle$ where

  $$\mathsf{Hdr} =< I_1, \ldots, I_m, E_{sk_1}(K), \ldots, E_{sk_m}(K) > .$$

  E:$\{0,1\}^l \rightarrow \{0,1\}^l$ and $\mathtt{Enc}$:$\{0,1\}^* \rightarrow \{0,1\}^*$ are symmetric encryptions where l is a security parameter and I_j is the information on the subset S_j. Generally, a fast encryption scheme such as a stream cipher is used for $\mathtt{Enc}$ to encrypt the *actual* message. We call Hdr a *Header* (or an *enabling block*).
- Decryption: After receiving the ciphertext, a user u first finds the subset S_i including him from I_i. A legitimate user can then generate a subset key sk_i from his secret keys. He decrypts $E_{sk_i}(K)$ with it and obtains the *actual* message M from K.

A legitimate user should be included in an arbitrary subset and be able to derive its subset key from his secret keys and the current transmission. In addition, even though all the revoked users collude with one another, it must be impossible for them to obtain any of the subset keys. The important factors for evaluating the broadcast encryption scheme are as follows.

- *Transmission cost* - the length of the Header for delivering the session key to users in $\mathcal{N}\backslash\mathcal{R}$. This depends on the number of subsets covering $\mathcal{N} \setminus \mathcal{R}$; namely, the number of partitions included in a Header.
- *Key storage* - the number of secret keys which each user should store in his secure device.
- *Computation cost* - the processing time to compute the subset key from the user's secret keys.

2.2 Adversarial Model

Our adversarial model follows the security model of Definition 10 in [19]. We briefly review their attack scenario. The attack game between the challenger and the adversary is as follows.

- **Setup**: The challenger runs the Setup algorithm and generates secret keys for all users.
- **Phase 1**: The adversary adaptively selects a set $\mathcal{R}$ of revoked users and obtains the secret keys of users in $\mathcal{R}$ from the challenger. He can get the encryption of message selected by himself when $\mathcal{R}$ is chosen. In addition, he can also create a ciphertext and see how any non-corrupted user decrypts it.

- **Challenge**: The adversary chooses a message M and a set $\mathcal{R}'$ including all the sets of revoked users selected in Phase 1. The challenger picks a random bit $b \in \{0,1\}$ and sets $C = \mathsf{Broadcast}(\mathcal{R}', M_b)$ where M_1 is M and M_0 is a random message of similar length. Then he sends it to the adversary.

- **Guess**: The adversary outputs a guess $b' \in \{0,1\}$.

We say that a broadcast encryption scheme is secure if for any polynomial time adversary, the probability that he distinguishes between M_0 and M_1 is negligible.

3 Proposed Scheme

In this section we propose an efficient broadcast encryption scheme with *log-key* storage. Our construction is based on the B1 scheme by Hwang *et al.* [14]. While the B1 scheme has at most $2r$ transmission cost and $O(\log n)$ key storage, its computation cost is proportional to n. To achieve a reasonable computation cost, in [14] the $\overline{\mathsf{B1}}$ scheme was constructed from the B1 scheme by their compiler. However, its transmission cost increases in proportion to the number of levels in the hierarchy. While our complete scheme has a similar structure to the B1 scheme, it achieves efficient transmission cost by the related keys between each level in the hierarchy. We first introduce the modified B1 scheme and construct an efficient broadcast encryption scheme from it.

3.1 Basic Scheme

In this section, we slightly modify the B1 scheme. Actually, this scheme is identical to the B1 scheme except for technique that the information I on the subset is represented and a user searches a subset including him. In the B1 scheme, a non-revoked user first finds two adjacent revoked users and should performs a binary search in an interval of two revoked users. In our scheme, a user can directly search his subset from the indexes and the direction of a hash chain

We define two one-way chains for users between u_i and u_j $(i \leq j)$ as $\mathcal{OC}_{i \to j}$ and $\mathcal{OC}_{i \leftarrow j}$. Let $f : \{0,1\}^l \to \{0,1\}^l$ be a one-way function. Then $\mathcal{OC}_{i \to j}$ is a one-way chain from i to j that, given a label $L_i \in_R \{0,1\}^l$ for u_i, ha the value $f^{j-i}(L_i)$. On the other hand, $\mathcal{OC}_{i \leftarrow j}$ is a one-way chain from j to i that, given a label $L_j \in_R \{0,1\}^l$ for u_j, has the value is $f^{j-i}(L_j)$. Our basic scheme is as follows.

- Setup: The center imagines the number line $\mathfrak{L}$ with n nodes where each node is numbered i $(i = 1, \ldots, n)$ with level order from left to right. Each user is assigned to each node. Let a user assigned to a node i be u_i. The center randomly selects a label $L_i \in \{0,1\}^l$ for each node i $(1 \leq i \leq n)$. We denote a set of users in an interval of i and j as $\mathcal{I}_{i;j}$. Then $f^{m-i}(L_i)$ and $f^{j-m}(L_j)$ are given to a user u_m in $\mathcal{I}_{i;j}$ as the secret key. If keys for $\mathcal{I}_{i;j}$ are assigned, $\mathcal{I}_{i;j}$ is divided into two intervals, $\mathcal{I}_{i;t}$ and $\mathcal{I}_{t+1;j}$, where $t = \lfloor \frac{i+j}{2} \rfloor$. Then secret keys for users in $\mathcal{I}_{i;t}$ and $\mathcal{I}_{t+1;j}$ are assigned by the same method. If $m < t$, it assigns only $f^{t-m}(L_t)$ to a user u_m for $\mathcal{I}_{i;t}$ since $f^{m-i}(L_i)$ can be used for both $\mathcal{I}_{i;j}$ and $\mathcal{I}_{i;t}$. If $m \geq t$, only $f^{m-t}(L_t)$ is assigned for $\mathcal{I}_{t+1;j}$. Therefore, one additional key is given to a user whenever a new interval is made. It starts from $\mathcal{I}_{1;n}$ and recursively repeats. Consequently, a user should store $1 + \log n$ keys in his secure storage. For example, assume that there are 16 users in total. Then the secret keys for u_7 are $f^6(L_1)$, $f^9(L_{16})$, $f(L_8)$, $f^2(L_5)$, and L_7 as shown in Figure 1.

- Broadcast: Given $\mathcal{R}$, the center first divides the number line $\mathfrak{L}$ into the intervals where each interval include one revoked user or successively revoked users. If a user u_t in $\mathcal{I}_{i;j}$ is revoked,

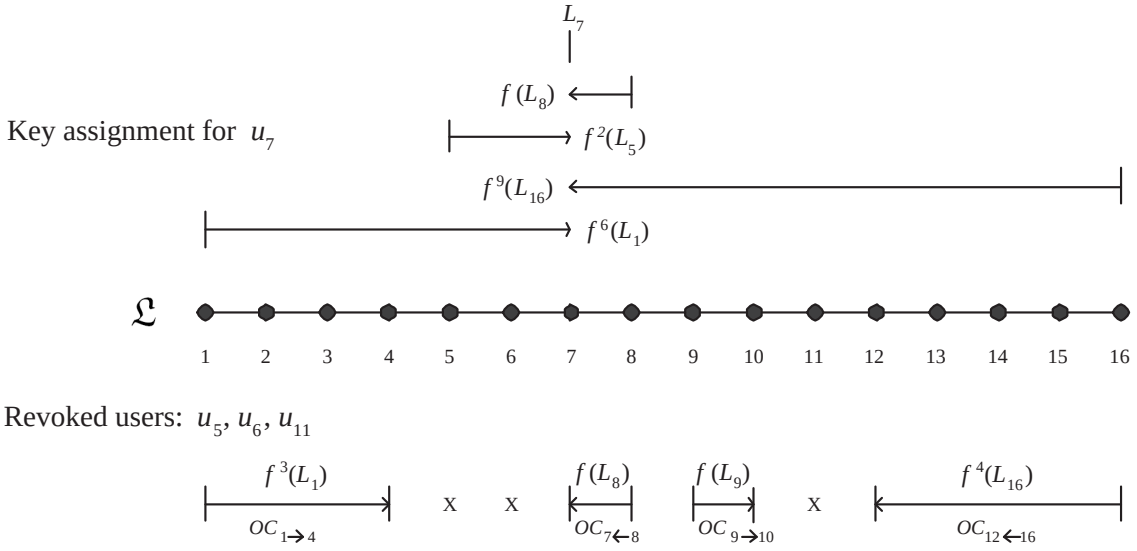

Figure 1: An example of the basic scheme for n=16.

non-revoked users in $\mathcal{I}_{i;j}$ are covered by two hash chains $\mathcal{OC}_{i\leftarrow t-1}$ and $\mathcal{OC}_{t+1\leftarrow j}$. Then, for users in $\mathcal{I}_{i;j}$, a session key K is encrypted with the chain values of $\mathcal{OC}_{i\rightarrow t-1}$ and $\mathcal{OC}_{t+1\leftarrow j}$, namely $f^{t-1-i}(L_i)$ and $f^{j-(t+1)}(L_j)$. Here, the subset information for two hash chains $\mathcal{OC}_{i\leftarrow t-1}$ and $\mathcal{OC}_{t+1\leftarrow j}$ can be $[+;i, t-1]$ and $[-;t+1, j]$.

- Decryption: After receiving the ciphertext, a user u_m first finds the subset including him from the subset information $[\pm;i, j]$ by checking whether $i \leq m \leq j$. If the direction of his subset is $+$, then he computes $\mathcal{OC}_{i\rightarrow j}$ by $f^{j-m}(f^{m-i}(L_i))$. Otherwise, he computes $\mathcal{OC}_{i\leftarrow j}$ by $f^{m-i}(f^{j-m}(L_j))$.

In Figure 1, if u_5, u_6 and u_{11} are revoked, the session key is encrypted with $f^3(L_1)$, $f(L_8)$, $f(L_9)$, and $f^4(L_{16})$ respectively. The scheme requires at most $2r$ transmission cost because at most two ciphertexts for one revoked user are generated. Its security is provided under the pseudo-randomness of f [14].

3.2 Complete Scheme

The basic scheme is not reasonable for practical applications because it has a computation cost proportional to n, though it satisfies the *log-key* restriction and $2r$ bound of the transmission cost. We extend the basic scheme to a hierarchical structure similar to the generic transformation of [14]. Actually, in all the schemes with hierarchical structure for efficient trade-offs among the transmission cost, the key storage and the computation cost, the transmission cost increases linearly according to the number of hierarchies (or stratifications).[ii]

However, our construction can maintain the $2r$ bound of the transmission cost while satisfying the reasonable computation cost and log-key storage requirements. Our scheme achieves it from additional keys and computation cost proportional to the number of the levels in the hierarchy. In addition, our scheme has a trade-off between the key storage and the computation cost under a reasonable bound. The complete scheme is as follows.

[ii]For example, the LSD scheme, the SSD scheme, and the $\overline{\text{B1}}$ scheme have a transmission cost proportional to the number of layers, stratifications, and the levels in the hierarchy respectively.

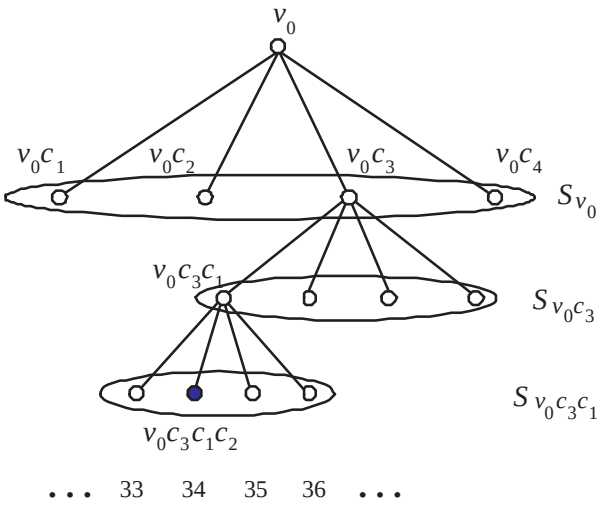

Figure 2: An example of the complete scheme for n=64.

- Setup: We assume that n is a^d. The center imagines a-ary tree T_a with a depth d and assigns one user to each leaf. Then, each leaf in T_a is numbered i ($i = 1, \ldots, n$) with level ordered from left to right. Let a root of T_a denote v_0 and i-th child of a node v denote vc_i. In addition, we call a set of children of a node v a sibling set S_v. In an example of Figure 2, node 34 is represented as $v_0c_3c_1c_2$ and $S_{v_0c_3c_1}$ is $\{33, 34, 35, 36\}$.

  Let T_v be a subtree rooted at a node v of T_a. The center randomly selects each label L_i for each node i in T_a. Then it generates keys for S_v by Setup of the basic scheme. Keys for vc_t in S_v are given to users assigned to leaves of T_{vc_t}. In consequence, a user assigned to $v_0c_{t_1} \cdots c_{t_d}$ has keys for S_{v_0}, $S_{v_0c_{t_1}}$, $\ldots$, $S_{v_0c_{t_1}\cdots c_{t_d}}$. Then, in Figure 2, a user u_{34} has secret keys, $f^2(L_{v_0c_1})$, $f(L_{v_0c_4})$, $L_{v_0c_3}$ $L_{v_0c_3c_1}$, $f^3(L_{v_0c_3c_4})$, $f(L_{v_0c_3c_2})$ $f(L_{33})$, $f^2(L_{36})$, L_{34}. This assignment is actually identical to that by the compiler introduced in [14].

  In our scheme, to eliminate the transmission cost of the hierarchical structure, users receives additional keys. Let $g : \{0,1\}^l \to \{0,1\}^l$ be a different one-way function with f. Let $f(f(L))$ denote $f \circ f(L)$, and $g \circ f^k(L)$ denote $g_k(L)$. Then $g \circ f^x \circ g \circ f^y(L)$ can be represented as $g_x \circ g_y(L)$.

  When generating keys for an interval $\mathcal{I}_{vc_i;vc_j}$ of S_v, the center additionally computes the following labels for a user $vc_{t_1} \cdots c_{t_k}$; For $1 < h < k$,

$$L^{vc_i}_{vc_{t_1}\ldots c_{t_h}c_1} = f \circ g_{t_h-1-1}(L^{vc_i}_{vc_{t_1}\cdots c_{t_{h-1}}c_1}) \tag{1}$$

$$L^{vc_j}_{vc_{t_1}\ldots c_{t_h}c_a} = f \circ g_{a-t_h-1}(L^{vc_i}_{vc_{t_1}\cdots c_{t_{h-1}}c_a}). \tag{2}$$

  A label $L^{vc_i}_w$ means the label for a node w generated by L_{vc_i}. Secret keys, $f^{t_{h+1}-1}(L^{vc_i}_{vc_{t_1}\ldots c_{t_h}c_1})$ and $f^{a-t_{h+1}}(L^{vc_j}_{vc_{t_1}\ldots c_{t_h}c_a})$ for $1 < h < k$, are additionally given to a user $vc_{t_1} \ldots c_{t_k}$. Here, $g_{-1}(L_i) = g \circ f^{-1}(L_i)$. For example, if $t_h = 1$, $L^{vc_i}_{vc_{t_1}\ldots c_{t_h}c_1} = f \circ g_{-1}(L^{vc_i}_{vc_{t_1}\ldots c_{t_{h-1}}c_1}) = f \circ g_{-1} \circ f \circ g_{t_{h-1}-1-1}(L^{vc_i}_{vc_{t_1}\ldots c_{t_{h-2}}c_1}) = f \circ g \circ g_{t_{h-1}-1-1}(L^{vc_i}_{vc_{t_1}\ldots c_{t_{h-2}}c_1})$. In addition, if $t_1 = i$ or j, the above additional keys are not given and only L_{vc_i} or L_{vc_j} is given. Consequently, $1 + k \cdot \log a$ keys are given

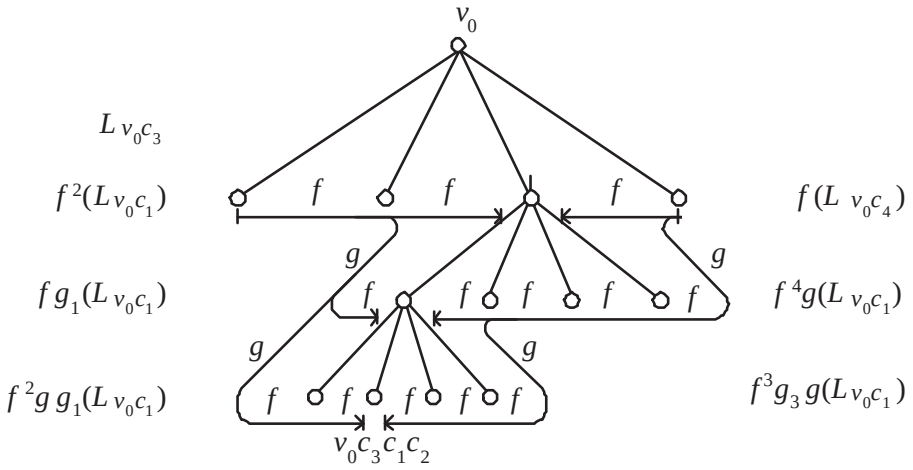

Figure 3: Key assignment to u_{34} for S_{v_0}.

to a user for S_v. A user $v_0c_{t_1}\dots c_{t_d}$ in T_a has all secret keys for $S_{v_0c_{t_1}}, S_{v_0c_{t_1}c_{t_2}}, \dots, S_{v_0c_{t_1}\dots c_{t_d}}$. Therefore, the number of secret keys for a user is $d + \frac{(d+1)}{2} \cdot \log n$ in total;

$$\sum_{h=1}^{d} 1 + h \log a = d + \log a \cdot \sum_{h=1}^{d} h = d + \frac{d^2+d}{2} \cdot (\log a) = d + \frac{(d+1)}{2} \cdot \log n.$$

In the example of Figure 3, the secret keys for u_{34} are given as follows since $L^{v_0c_1}_{v_0c_3c_1} = f \circ g_1(L_{v_0c_1})$, and $L^{v_0c_1}_{v_0c_3c_1c_1} = f \circ g \circ g_1(L_{v_0c_1})$ by (1). Other labels can be easily computed by (2).

$$\begin{array}{lcl} S_{v_0} & : & f^2(L_{v_0c_1}),\ f \circ g_1(L_{v_0c_1}),\ f^2 \circ g \circ g_1(L_{v_0c_1}) \\ & & f(L_{v_0c_4}),\ f^4 \circ g(L_{v_0c_4}),\ f^3 \circ g_3 \circ g(L_{v_0c_4}),\ L_{v_0c_3} \\ S_{v_0c_3} & : & L_{v_0c_3c_1},\ f^3(L_{v_0c_3c_4}),\ f^3 \circ g_2(L_{v_0c_3c_4}),\ f(L_{v_0c_3c_2}),\ f^3 \circ g(L_{v_0c_3c_2}) \\ S_{v_0c_3c_1} & : & f(L_{33}),\ f^2(L_{36}),\ L_{34} \end{array}$$

A user has 15 secret keys in total because $d = 3$ and $n = 2^6$.

- Broadcast: The center imagines the number line $\mathfrak{L}$ composed by leaves of T_a. Given $\mathcal{R}$, the center makes the hash chains in the form of $\mathcal{OC}_{i\to j}$ or $\mathcal{OC}_{i\leftarrow j}$ which cover $\mathfrak{L}$ as in Broadcast of the basic scheme. If a least common ancestor of nodes from i to j is v, we denote this chain as $\mathcal{OC}^v_{i\to j(\text{ or } i\leftarrow j)}$. Let a node, a child of v and an ancestor of i, be v_i and a parent of j be p_j. And if a node w is p_wc_m, $s(w)$ means m. First, we consider $\mathcal{OC}^v_{i\to j}$. The chain value of $\mathcal{OC}^v_{i\to j}$ is computed by the following process.

  1. If i and j are siblings (namely, v is a parent of i and j), then the chain value of $\mathcal{OC}^v_{i\to j}$ equals that of $\mathcal{OC}_{i\to j}$ in the basic scheme.
  2. Else if j is the rightmost leaf in a subtree T_b of T_v.
     - If v_i and b are siblings, then the chain value of $\mathcal{OC}^v_{i\to j}$ is $f^{s(b)-s(v_i)}(L_{v_i})$.
     - Otherwise, the chain value of $\mathcal{OC}^v_{i\to j}$ is $f^{s(b)-1}(L^{v_i}_{p_bc_1})$.
  3. Otherwise, the chain value of $\mathcal{OC}^v_{i\to j}$ is $f^{s(j)-1}(L^{v_i}_{p_jc_1})$.

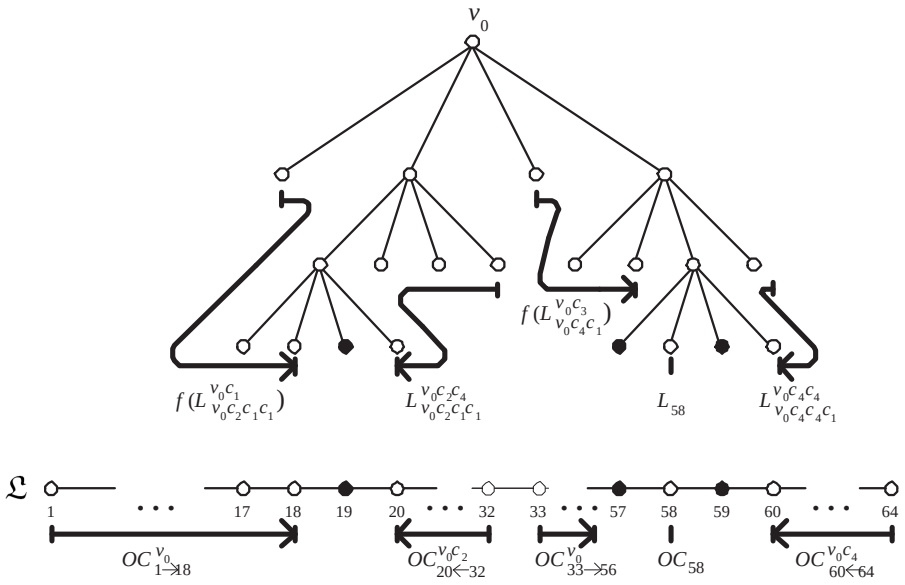

Figure 4: Revocation in the complete scheme.

The chain value of $\mathcal{OC}^v_{i\leftarrow j}$ is generated by the opposite operation with the above process. Consequently, our scheme has the same transmission cost as the basic scheme.

In Figure 4, we assume that three users u_{19}, u_{57}, and u_{59} are revoked. Then the following one-way chains are generated:

$$\mathcal{OC}^{v_0}_{1\to 18}, \mathcal{OC}^{v_0c_2}_{20\leftarrow 32}, \mathcal{OC}^{v_0}_{33\to 56}, \mathcal{OC}_{58\leftarrow 58}, \text{ and } \mathcal{OC}^{v_0c_4}_{60\leftarrow 64}.$$

For them, the chain values are $f(L^{v_0c_1}_{v_0c_2c_1c_1})$, $L^{v_0c_2c_4}_{v_0c_2c_1c_4}$, $f(L^{v_0c_3}_{v_0c_4c_1})$, L_{58}, and $L^{v_0c_4c_4}_{v_0c_4c_3c_4}$, respectively. For a specific example of a chain value, consider $\mathcal{OC}^{v_0}_{33\to 56}$. A least common ancestor of 33 and 56 is v_0 and 56 is the rightmost leaf of $T_{v_0c_4c_2}$. This chain value is $f(L^{v_0c_3}_{v_0c_4c_1})$ since v_0c_3 and $v_0c_4c_2$ are not siblings, where $L^{v_0c_3}_{v_0c_4c_1} = f \circ g(L_{v_0c_3})$.

- Decryption: After receiving the ciphertext, a user u_w finds his subset from $[\pm; i, j]$. If $i \le w \le j$, u_w is included in the subset $[\pm; i, j]$. Suppose that the direction is $+$.

  1. If i and j are siblings, then he computes a subset key $f^{j-i}(L_i)$ by $f^{j-w} \circ f^{w-i}(L_i)$ from his secret key $f^{w-i}(L_i)$.
  2. Else if j is the rightmost leaf in a subtree T_b of T_v,
     - If v_i and b are siblings, then he computes the chain value $f^{s(b)-s(v_i)}(L_{v_i})$ for $\mathcal{OC}^v_{i\to j}$ by iteratively operating the f function with his secret key.
     - Otherwise, he computes the chain value $f^{s(b)-1}(L^{v_i}_{p_bc_1})$ for $\mathcal{OC}^v_{i\to j}$ using g and f with his secret key.
  3. Otherwise, he computes the chain value $f^{s(j)-1}(L^{v_i}_{p_jc_1})$ for $\mathcal{OC}^v_{i\to j}$ using g and f with his secret key.

If the direction is $-$, then it performs the above method in the opposite direction. For example, u_{34} is included in $\mathcal{OC}^{v_0}_{33\to 56}$. A least common ancestor of nodes from 33 to 56 is v_0 and an

ancestor of 33 in the children of v_0 is v_0c_3. Because 56 is a rightmost leaf of $T_{v_0c_4c_2}$, a chain value for $\mathcal{OC}^{v_0}_{33\to 56}$ is $f(L^{v_0c_3}_{v_0c_4c_1})$. User u_{34} is also a descendent of v_0c_3, so he has $L_{v_0c_3}$ as his secret key. Therefore, he can obtain a subset key $f(L^{v_0c_3}_{v_0c_4c_1})$ by $f^2 \circ g(L_{v_0c_c})$. Because a revoked user u_{57} has secret keys $f^2(L^{v_0c_3}_{v_0c_4c_1})$ and $L^{v_0c_3}_{v_0c_4c_3c1}$ $(=f \circ g \circ f(L^{v_0c_3}_{v_0c_4c_1}))$ generated by $L_{v_0c_3}$, he cannot obtain the subset key without inverting f or $f \circ g$.

Efficiency. Transmission cost of the complete scheme is less than $2r$ because at most two ciphertexts per revoked user are generated, as in the basic scheme. To generate a subset key, a user needs at most $d \cdot a$ computation cost. In addition, a user stores $d + \frac{d+1}{2} \cdot \log n$ keys as shown above. Our scheme achieve the efficient transmission cost from a trade-off with the computation cost and the key storage by the number of stratification in hierarchical structure.

Security. The security of our scheme is provided under the pseudo-randomness of f and g. Actually, because all secret keys given to users are generated by one-way chains, excluded users (i.e. revoked users) by one-way chains cannot compute any subset key without inverting the given one-way functions f and g. However, a more formal security analysis is needed. We show that our scheme is resilient to collusion of any set of revoked users.

Lemma 1 *The key assignment of the complete scheme satisfies the key-indistinguishability property under the pseudo-randomness of two functions f and g.*

Proof. Let $f \circ g$ define a function $h : \{0,1\}^l \to \{0,1\}^l$. If an adversary $\mathcal{A}$ can break the key-indistinguishability property of our scheme, we show that the pseudo-randomness of f and h is also broken by simulating $\mathcal{A}$. We assume that our scheme is defined by a collection of subsets $S_1, \ldots, S_w$. For any $1 \le i \le w$, let $S_{i_1}, \ldots, S_{i_t}$ be all the subsets that are contained in S_i and $sk_{i_1}, \ldots, sk_{i_t}$ be their corresponding keys. An adversary $\mathcal{A}$ attempts to distinguish the keys $sk_{i_1}, \ldots, sk_{i_t}$ from the random keys $rk_{i_1}, \ldots, rk_{i_t}$. Consider a feasible adversary $\mathcal{A}$ that

1. Selects i, $1 \le i \le w$
2. Receives the secret keys K_u's for all $u \in \mathcal{N} \setminus S_i$

We denote the probability that $\mathcal{A}$ distinguishes the key from the random key by ε as follows.

$$|\Pr[\mathcal{A} \text{ outputs } i|sk_i] - \Pr[\mathcal{A} \text{ outputs } i|rk_i]| \le \varepsilon.$$

If an adversary $\mathcal{A}$ can distinguish the key from the random key, we can break the pseudo-randomness of f or h, since K_u includes an output of the function f or h on the key. Hence, if the pseudo-randomness of two one-way functions f and h is guaranteed, ε is negligible.

Also, let P_{i_j} be the probability that given the subset keys contained in S_i, $\mathcal{A}$ outputs i, where the first j keys are the true keys and the remaining $t-j$ keys are the random keys. Namely,

$$P_{i_j} = \Pr[\mathcal{A} \text{ outputs } i|sk_{i_1}, \ldots, sk_{i_j}, rk_{i_{j+1}}, \ldots, rk_{i_t}].$$

Then we can obtain the following equation by the standard hybrid argument, since $|P_{i_j} - P_{i_{j+1}}| < \varepsilon$ for $1 \le j < t$.

$$|\Pr[\mathcal{A} \text{ outputs } i|sk_{i_1}, \ldots, sk_{i_t}] - \Pr[\mathcal{A} \text{ outputs } i|rk_{i_1}, \ldots, rk_{i_t}]| \le t \cdot \varepsilon.$$

In consequence, our scheme satisfies the key-indistinguishability property under the pseudo-randomness of given functions f and g. □

In addition, Naor *et al.* showed that the key-indistinguishability property is sufficient for a scheme in the subset-cover framework to be secure in the adversarial model of Section 2.2 [19]. By Lemma 1 and Theorem 11 of [19], the security of the complete scheme is provided.

4 Discussions

We analyze the complexities of various broadcast encryption schemes in this section. While the SD scheme needs at most $2r$ transmission cost, $O(\log^2 n)$ key storage is required. The LSD scheme, the SSD scheme, the π scheme, and the $\overline{\text{B1}}$ scheme have trade-offs among the transmission cost, the computation cost and the key storage. Their complexities change depending on the system parameters that define the degree of stratification. Table 2 shows the comparison between our scheme and other efficient schemes. In the transmission cost column of Table 2, '$\leq$' means a upper bound of the transmission cost.

Table 2: Complexity of efficient BE schemes for $n = 10^8$

		Transmission cost	Key storage	Computation cost
SD	[19]	$\leq 2r$	368 (5.74 Kbyte)	27
Basic LSD	[15]	$\leq 4r$	143 (2.24 Kbyte)	27
SSD	[12]	$\leq 8r$	213 (3.33 Kbyte)	100
(1,100)-π_1	[16]	$\leq 2r + 10^6$	5274 (82.4 Kbyte)	100
$\overline{\text{B1}}$	[14]	$\leq 8r$	27 (0.432 Kbyte)	100
Our scheme		$\leq 2r$	129 (2.06 Kbyte)	80

We assume that the size of keys is 128 bits and n is 10^8 for a practical instance. While the computation cost of the SD scheme and the LSD scheme is fixed to $O(\log n)$, that of other schemes varies with the system parameters. Hence, we bound the computation cost to 100. This computation cost is reasonable even for low-power devices. If the computation cost of the SSD scheme and the $\overline{\text{B1}}$ scheme is bounded to 100, their system parameters d and k are 4. Therefore, their transmission cost is $2 \cdot (4r)$.[iii] In addition, we compare other schemes to the Basic LSD with $k = 2$ because the LSD scheme satisfies the most efficient transmission cost when having two layers.

The SSD scheme and the $\overline{\text{B1}}$ scheme have high transmission cost proportional to the parameters d and k, and the (1,100)-π_1 scheme does not have a good transmission cost where the revocation rate is very small (i.e less than 1%). However, our scheme maintains a low transmission cost regardless of the parameter and revocation rate. For our scheme, we consider the case of $a = 10$ and $d = 8$ to achieve a reasonable computation cost. At this time, the computation cost of our scheme is less than 80. As shown in Table 2, our scheme has the most efficient transmission cost under the reasonable computation cost and *log-key* restriction.

Because our scheme possesses low transmission cost and small storage size, it can be efficiently

[iii] The upper bound of the transmission cost of the SSD scheme should be $16r$ from Table 1 when $k = 4$. However, its transmission cost is actually similar to that of the $\overline{\text{B1}}$ scheme. Hence we regard a upper bound of its transmission cost as $2kr$.

used where the computation and the storage are restricted as in a handheld device, or where the transmission is expensive as in a set-top box and CD/DVD. In addition, when a group of malicious users (called traitors) combines their secret keys to produce a pirate decode, the center can trace at least one of the traitors given access to this decoder by a subset tracing procedure introduced in [19] since our scheme is based on a subset-cover framework.

Our scheme is also suitable for broadcast encryption over wireless networks. In a wireless network, the target of messages is a handheld device with small memory and low computing power. Moreover, the bandwidth of wireless networks is narrower than that of wired networks. Therefore, our scheme is of great use for broadcast encryption scheme over wireless networks.

In addition, the key assignment technique used to construct our scheme can be applied to the schemes with a hierarchical structure such as the SSD scheme [12] and the $\overline{\mathsf{B2}}$ scheme [14]. The transmission cost of the modified schemes would be independent of the number of levels in hierarchy.

5 Concluding Remarks

We have presented a communication-efficient broadcast encryption scheme under the *log-key* restriction. In many practical applications, the systems should be efficiently able to deal with a very large group of users having a wide variety of devices. Our scheme can provide an efficient transmission cost under a reasonable computation cost for a large number of users by requiring key storage proportional to the log of the number of users. It is also a good solution for systems that rely on devices with limited secure storage.

References

[1] T. Asano, "A Revocationn Scheme with Minimal Storage at Receivers," *Advances in Cryptology - ASIACRYPT'02*, LNCS vol. 2501, pp. 433-450, 2002.

[2] T. Asano and K. Kamio, "A Tree Based One-Key Broadcast Encryption Scheme with Low Computational Overhead," *Information Secrutiy and Privacy - ACISP'05*, LNCS vol. 3574, pp. 89-100, 2005.

[3] N. Attrapadung and H. Imai, "Graph-Decomposition-Based Framework for Subset-Cover Broadcast Encryption and Efficient Instantiations," *Advances in Cryptology - ASIACRYPT'05*, (to appear).

[4] N. Attrapadung, K. Kobara, and H. Imai, "Sequential Key Derivation Patterns for Broadcast Encryption and Key Predistribution Schemes," *Advances in Cryptology - ASIACRYPT'03*, LNCS vol. 2894, pp. 374-391, 2003.

[5] S. Berkovits, "How to broadcast a secret," *Advances in Cryptology - EUROCRYPT'91*, LNCS vol. 547, pp. 535-541, 1991.

[6] C. Blundo, L. A. Frota, and D. R. Stinson, "Trade-off between Communication and Storage in Unconditionally Secure Schemes for Broadcast Encryption and Interactive Key Distribution," *Advances in Cryptology - CRYPTO'96*, LNCS vol.1109, pp. 387-400, 1996.

[7] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," *Advances in Cryptology - CRYPTO'05*, LNCS vol.3621, pp. 258-275, 2005.

[8] B. Chor, A. Fiat, and M. Naor, "Tracing traitor," *Advances in Cryptology - CRYPTO'94*, LNCS vol. 839, pp. 257-270, 1994.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms," *MIT Press*, 2001.

[10] Y. Dodis and N. Fazio, "Public Key Trace and Revoke Scheme Secure Against Adapitive Chosen Ciphertext Attack," *Public Key Cryptography - PKC'03*, LNCS vol. 2567, pp. 100-115, 2003.

[11] A. Fiat and M. Naor, "Boradcast Encryption," *Advances in Cryptology - CRYPTO'93*, LNCS vol. 773, pp. 480-491, 1993.

[12] M. T. Goodrich, J. Z. Sun, and R. Tamassia, "Efficient Tree-Based Revocation in Groups of Low-State Devices," *Advances in Cryptology - CRYPTO'04*, LNCS vol. 3152, pp. 511-527, 2004.

[13] E. Gafni, J. Staddon, and Y. L. Yin, "Efficient Methods for Intergrating Traceability and Broadcast Encryption," *Advances in Cryptology - CRYPTO'99*, LNCS vol. 1666, pp. 372-387, 1999.

[14] J. Y. Hwang, D. H. Lee, and J. Lim, "Generic Transformation for Scalable Broadcast Encryption Scheme," *Advances in Cryptology - CRYPTO'05*, LNCS vol. 3621, pp. 276-292, 2005.

[15] D. Halevy and A. Shamir, "The LSD broadcast encryption scheme," *Advances in Cryptology - CRYPTO'02*, LNCS vol. 2442, pp. 47-60, 2002.

[16] N.-S. Jho, J. Y. Hwang, J. H. Cheon, M.-H. Kim, D. H. Lee, and E. S. Yoo, "One-Way Chain Based Broadcast Encryption Schemes," *Advances in Cryptology - EUROCRYPT'05*, LNCS vol. 3494, pp. 559-574, 2005.

[17] C. H. Kim, Y. H. Hwang, and P. J. Lee, "An Efficient Public Key Trace and Revoke Scheme Secure against Adaptive Chosen Ciphertext Attack," *Advances in Cryptology - ASIACRYPT'03*, LNCS vol. 2894, pp. 359-373, 2003.

[18] M. Mihaljevic, "Key Management Schemes for Stateless Receivers Based on Time Varying Heterogeneous Logical Key Hierarchy," *Advances in Cryptology - ASIACRYPT'03*, LNCS vol. 2894, pp. 137-154, 2003.

[19] D. Naor, M. Naor, and J. Lostpiech, "Revocation and tracing schemes for stateless receivers," *Advances in Cryptology - CRYPTO'01*, LNCS vol. 2139, pp. 41-62, 2001.

[20] M. Naor and B. Pinkas, "Efficient Trace and Revoke Schemes," *Financial Cryptography'00*, LNCS vol. 1962, pp. 1-20, 2000.

[21] D. M. Wallner, E. J. Harder, and R. C. Agee, "Key management for multicast: Issues and Architectures," *IETF Network Working Group*, RFC 2627, 1999.

[22] C. K. Wong, M. Gouda, and S. Lam, "Secure group communications using key graphs," *ACM SIGCOMM'98*, pp. 68-79, 1998.