Generic Transformation for Scalable Broadcast Encryption Schemes*

Jung Yeon Hwang, Dong Hoon Lee, and Jongin Lim

Graduate School of Information Security CIST,
Korea University, Seoul 136-701, Korea
{videmot,donghlee,jilim}@korea.ac.kr

Abstract. Broadcast encryption schemes allow a message sender to broadcast an encrypted data so that only legitimate receivers decrypt it. Because of the intrinsic nature of one-to-many communication in broadcasting, transmission length may be of major concern. Several broadcast encryption schemes with good transmission overhead have been proposed. But, these broadcast encryption schemes are not practical since they are greatly sacrificing performance of other efficiency parameters to achieve good performance in transmission length.
In this paper we study a generic transformation method which transforms any broadcast encryption scheme to one suited to desired application environments while preserving security. Our transformation reduces computation overhead and/or user storage by slightly increasing transmission overhead of a given broadcast encryption scheme. We provide two transformed instances. The first instance is comparable to the results of the "stratified subset difference (SSD)" technique by Goodrich et al. and firstly achieves $\mathcal{O}(\log n)$ storage, $\mathcal{O}(\log n)$ computation, and $\mathcal{O}(\frac{\log n}{\log \log n} r)$ transmission, at the same time, where n is the number of users and r is the number of revoked users. The second instance outperforms the "one-way chain based broadcast encryption" of Jho et al., which is the best known scheme achieving less than r transmission length with reasonable communication and storage overhead.

1 Intoduction

In recent years broadcast encryption schemes have been intensively studied for lots of applications such as satellite-based commerce, multicast communication, secure distribution of copyright-protected material and DRM(Digital Rights Management), etc. Broadcast encryption (BE) schemes are one-to-many communication methods in which a message sender can broadcast an encrypted data to a group of users over an insecure channel so that only legitimate receivers decrypt it. Especially, a *stateless* BE scheme has a useful property that any legitimate receiver with its initial set-up can obtain the current group session key

* This research was supported by the MIC(Ministry of Information and Communication), Korea, under the ITRC(Information Technology Research Center) support program supervised by the IITA(Institute of Information Technology Assessment).

only from the current transmission without the history of past transmissions. One of main security concerns in the stateless broadcast encryption schemes is how to efficiently exclude illegal (revoked) users from a privileged set, that is, how to ensure that only legal users decrypt a encrypted broadcast message.

Various BE schemes have been designed to improve efficiency. Efficiency of BE schemes is mainly measured by three parameters: the length of transmission messages, user storage, and computational overhead at a user device. The ultimate goal would be to achieve the best efficiency of all three parameters simultaneously. But it seems, to date, that there exists no BE scheme achieving this goal. As an alternative treatment, a trade-off between the parameters has been considered. In fact, schemes with a various efficiency trade-off fit into many real applications and moreover support the creation of potential application scenarios. Since a message sender in BE schemes broadcasts a message to possible huge number of users, efficiency in transmission overhead has been considered as a critical measure by service providers. Therefore, reducing storage or computation overhead without greatly sacrificing transmission overhead is important.

In most practical applications of BE, a group of users may be quite huge and BE schemes should basically provide scalability, i.e., suitability for a large number of group users. But, unfortunately, most of transmission-efficient BE schemes are not scalable since they requires large storage or computation at a user device. Especially, these schemes are not suitable to wireless networks where users are holding strictly resource-restricted mobile devices.

OUR CONTRIBUTIONS. In the paper we study a modular approach to transform an arbitrary BE scheme to a scalable one efficiently while preserving the security of the underlying scheme. We construct a compiler of which resulting scheme, for a large number of group users, maintains transmission overhead of the original scheme asymptotically but gains reduction in users storage and/or computation overhead. Hence, by applying our compiler to a known transmission-efficient BE scheme which is impractical due to large computation or user storage for keys, we can inexpensively construct an efficient and scalable solution regardless of the structure of the underlying BE scheme.

To illustrate our transformation, we concretely present two compiled instances which provide a good performance in various aspects, in fact, outperform the previously known schemes:

- Goodrich et al. [9] proposed the stratified subset difference (SSD) method, which achieves $\mathcal{O}(r)$ transmission and $\mathcal{O}(n^{\frac{1}{d}})$ computation and $\mathcal{O}(\log n)$ storage overhead per user, where n is the number of users, r is the number of revoked users, and d is a predefined constant. This is the best scheme achieving $\mathcal{O}(r)$ transmission and $\mathcal{O}(\log n)$ storage overhead simultaneously. But under $\mathcal{O}(\log n)$ computation restriction, the scheme needs $\mathcal{O}(\frac{1}{\log(\log n)}\log^2 n)$ storage, which is closer to $\mathcal{O}(\log^2 n)$ storage overhead per user. This should be undesirable in memory-constrained environments. Our first example is a BE scheme which achieves $\mathcal{O}(\frac{\log n}{\log(\log n)}r)$ transmission, $\mathcal{O}(\log n)$ computation overhead, and $\mathcal{O}(\log n)$ (precisely $\log n + 1$) user storage, at the same time.

- Very recently, Jho et al. [14] proposed the "one-way chain based broadcast encryption schemes" of which one is the best scheme achieving less than r transmission messages with user computation overhead proportional to n at the worst case. But their scheme is still considered non-scalable because of excessive storage requirement, i.e., for a predetermined constant k, $\binom{n-1}{k}$ keys storage at a user device. The second example is a BE scheme in which the number of transmission messages is less than r only except for a small number of revoked users, i.e., 0.75 % of n, while user storage and computation overhead are in a reasonable bound.

RELATED WORK. Since the first formal work of BE by Fiat and Naor [8], many researches [12] have been done to improve the efficiency in various aspects by using various trade-off methods and design approaches, i.e., combinatorial designs, logical key trees, algebraic approaches such as secret sharing, multi-linear mapping, and cryptographic tools such one-way accumulator.

Some BE schemes based on combinatorial design are suggested to provide information-theoretical security [10, 11, 17, 18].

Based on a logical key tree structure, a number of broadcast encryption schemes [20, 19, 1, 2, 16, 13, 9] have been suggested. Significant works among them are the Subset Difference (SD) scheme [16] by Naor et al. and its improvement, the layered SD scheme [13] by Halevi and Shamir. These schemes achieve $\mathcal{O}(r)$ transmission complexity while maintaining $\mathcal{O}(\log n)$ computation overhead and $\mathcal{O}(\log^2 n)$ key storage per user. Recently Goodrich et al. [9] firstly proposed the stratified subset difference (SSD) method which satisfies $\mathcal{O}(\log n)$ keys storage per user (this is called the *log-key* restriction) and $\mathcal{O}(r)$ transmission overhead simultaneously but requires $\mathcal{O}(n^{\frac{1}{d}})$ computation overhead where d is a predetermined constant. Their security depends on the existence of pseudo-random sequence number generator.

To achieve more efficient transmission overhead, some schemes have used algebraic properties such as secret-sharing [15, 3]. But these schemes have to broadcast at least r transmission messages in order to expose the shares of revoked users. Recently, a notable work based on a one-way accumulator was suggested by Attrapadung et al. to achieve $\mathcal{O}(1)$ transmission complexity [2]. Their method uses a trade-off between security against collusion and non-secret storage size. However, despite of constant transmission complexity, their scheme is considered as impractical in the case of large number of users because of massive requirement in non-secret keys and computation cost at user side. Boneh and Silverberg [6] proposed a zero-message BE scheme which requires only constant amount of non-secret storage by using n-linear maps of which construction seems to be very difficult for $n > 2$. Very recently, Boneh et. al. [5] proposed a (public-key) BE scheme using bilinear maps where transmission length is $\mathcal{O}(\sqrt{n})$, user key storage is a constant size and computation overhead is $\mathcal{O}(\sqrt{n})$. Security of their scheme is based on the so-called *Bilinear Diffie-Hellman Exponent* assumption.

Organization. The rest of this paper is organized as follows. We review and define some notions of broadcast encryption in Section 2 and construct our compiler and analyze its efficiency in Section 3. We illustrate two compiled instances of our compiler in Section 4. We compare the resulting schemes with the SD [16], LSD [13], SSD [9], one-way chain based BE [14] schemes in Section 5. Finally, we conclude with some remarks on other issues in Section 6.

2 Broadcast Encryption

In this section we briefly review and define the notion of broadcast encryption. Generally BE schemes are classified into two types: symmetric key and public key based BE schemes. In the symmetric key setting, the only trusted group center GC can generate a broadcast message to users while, in the public key setting, any users are allowed to broadcast a message. We denote by $\mathcal{U}$ the set of users and by $\mathcal{R} \subset \mathcal{U}$ the set of revoked users. The following is a formal definition of a symmetric key based BE scheme.

Broadcast Encryption Scheme. A BE scheme B is a triple of polynomial-time algorithms (SetUp, BEnc, Dec), i.e., setup, broadcast encryption, and decryption:

- SetUp, the randomized algorithm takes as input a security parameter 1^λ and user set $\mathcal{U}$. It generates private information SKEY_u for user $u \in \mathcal{U}$. Private information of group center GC is defined as the set $\mathsf{SKEY}_{\mathcal{U}}$ of private information of all users.

- BEnc, the randomized algorithm takes as input a security parameter 1^λ, private information $\mathsf{SKEY}_{\mathcal{U}}$ of GC, a set $\mathcal{R}$ of revoked users, and a message M to be broadcast. It first generates a session key GSK and outputs $(Hdr_{\mathcal{R}}, C_{\mathsf{GSK},M})$ where a header Hdr is information for a privileged user to compute GSK and $C_{\mathsf{GSK},M}$ is a ciphertext of M encrypted under the symmetric key GSK.
  Broadcast message consists of $[\mathcal{R}, Hdr_{\mathcal{R}}, C_{\mathsf{GSK},M}]$. The pair $(\mathcal{R}, Hdr_{\mathcal{R}})$ and $C_{\mathsf{GSK},M}$ are often called the full header and the body, respectively.

- Dec, the (deterministic) algorithm takes as input a user index ind_u, private information SKEY_u of u, the set of revoked users $\mathcal{R}$, and a header $Hdr_{\mathcal{R}}$. If $u \in \mathcal{U} \backslash \mathcal{R}$ then it outputs the session key GSK.

In public key broadcast encryption, the setup algorithm additionally generates the public keys $\mathsf{PK}_{\mathcal{U}}$ of users and $\mathsf{PK}_{\mathcal{U}}$ instead of the private information $\mathsf{SKEY}_{\mathcal{U}}$ of GC is taken as input in the algorithms BEnc and Dec.

Input terms in the above description may be extended by allowing additional input terms such as a revocation threshold value, i.e., the maximum number of users that can be revoked.

In [16] Naor et. al. presented the so-called *Subset-Cover* framework. The idea of this abstract method is to define a specific subset and associate each subset with a (subset) key SK, which is made available only to the users of the given subset. To cover the set $\mathcal{U}\backslash\mathcal{R}$ of privileged users, $\mathcal{U}\backslash\mathcal{R}$ are partitioned into collection of such pre-defined subsets and the (subset) keys SK_i associated to the subsets are used to encrypt a session key GSK. In this case the header consists of ciphertexts of GSK, i.e., $Hdr_{\mathcal{R}}{=}[\mathcal{E}_{SK_1}(\mathsf{GSK}), ..., \mathcal{E}_{SK_t}(\mathsf{GSK})]$ where $\mathcal{E}$ is a symmetric encryption scheme.

EFFICIENCY. Let n and r be the numbers of total users and revoked users for a given BE scheme B, respectively. Efficiency of BE schemes is mainly measured by three parameters: transmission overhead, user storage, and computational overhead.

- $TO_{\mathsf{B}}(r, n)$: Transmission overhead is defined as the total length (number of bits) of a header in a broadcast message transmitted. We exclude the information of indices of revoked users and the body from the transmission overhead since the information are equivalently needed for all BE schemes.
- $SO_{\mathsf{B}}(n)$: User storage overhead is defined as the maximum number of private keys initially given to a user.
- $CO_{\mathsf{B}}(n)$: Computational overhead is defined as the maximum number of basic computation done by a user device.

SECURITY. Basically a BE scheme should provide resiliency to collusion of any set of revoked users. According to the capabilities of an adversary and security goal, we can formally define several types of the security notion of broadcast encryption. Here we briefly present the so-called $\mathsf{CCA1}$-security [4] (chosen ciphertext security in the pre-processing mode [7]) of broadcast encryption, which is believed to be sufficient for most applications. Especially we note that the Subset-Cover framework of [16] in which *computationally independent* keys are used as a message encryption key, is suitable to this notion.

To measure the $\mathsf{CCA1}$-security of a BE scheme B we consider the following game between an adversary $\mathcal{A}$ and a challenger which models adaptive adversarial actions, user corruption and chosen ciphertext attack, etc:

- **Setup**. The challenger runs $\mathsf{SetUp}(1^\lambda, \mathcal{U})$ algorithm and generates private information of users $u \in \mathcal{U}$.
- **Adversarial Action**. $\mathcal{A}$ corrupts any user u' to obtain private information $\mathsf{SKEY}_{u'}$ and asks to any (non-corrupted) user to decrypt a ciphertext C created by $\mathcal{A}$. $\mathcal{A}$ also gets the encryption of a message M selected by itself when it chooses a set $\mathcal{R}$ of revoked users.
- **Challenge**. As a challenge, $\mathcal{A}$ outputs a message CM and a set $\mathcal{R}'$ of revoked users including all ones corrupted by $\mathcal{A}$. The challenger selects a random bit $b{\in}\{0, 1\}$. If $b{=}1$ the challenger runs BEnc with $\mathcal{R}'$ to obtain $C{=}(Hdr_{\mathcal{R}'}$, $C_{\mathsf{GSK},CM})$. Otherwise it computes $C{=}(Hdr'_{\mathcal{R}'}$, $C_{\mathsf{GSK}',RM})$ where RM is a random message whose length is similar to that of the message CM. Then it gives C to $\mathcal{A}$.

- **Guess**. $\mathcal{A}$ outputs its guess $b' \in \{0,1\}$.

Let CGues denote the event that the adversary correctly guesses the bit b in the above game. The advantage of an adversary $\mathcal{A}$ is defined as $\mathsf{Adv}_{\mathcal{A},\mathsf{B}}(\lambda) = |2 \cdot Pr[\mathsf{CGues}] - 1|$ where $Pr[\mathsf{CGues}]$ is the probability of CGues. We say that a BE scheme B is *CCA1-secure* if for any probabilistic polynomial time adversary $\mathcal{A}$, the advantage $\mathsf{Adv}_{\mathcal{A},\mathsf{B}}(\lambda)$ is negligible.

3 Generic Transformation for Scalable Broadcast Encryption

In this section we present a compiler transforming a broadcast encryption scheme impractical due to computation overhead or user storage for huge number of users to a scalable one. We assume that the number of group users is denoted by $n = w^s$. The variables w and s are to be defined to reduce user storage or computation overhead in advance.

We first provide an overview of our construction intuitively. The main idea of our method is to apply a given broadcast encryption scheme B to a relatively small subset in a hierarchical and independent manner. To implement such a concept, we use a complete w-ary tree with height s, where each user is associated with a leaf. In the tree the root is labeled with a special symbol $b_0 = e$. If a node at depth less than s is labeled with β then its b_i-th child is labeled with βb_i where $b_i \in \{1, \ldots, w\}$. That is, $v_{b_0 b_1 \cdots b_{k-1}}$ is a node in level k where $b_0 b_1 \cdots b_{k-1}$ is the concatenation of all indices on the path from the root to the node. Let *sibling set* $S_{b_0 b_1 \cdots b_j}$ be a set of nodes with a same parent $v_{b_0 b_1 \cdots b_j}$ in the tree. The BE scheme B is applied to each sibling set $S_{b_0 b_1 \cdots b_j}$ independently, as if nodes in $S_{b_0 b_1 \cdots b_j}$ are users for an independent BE scheme. To revoke a user, by considering all nodes on the path from the revoked leaf (i.e., user) to the root as revoked nodes, we independently apply the revocation method of B to each sibling set including a node along in the path from the root to the revoked leaf.

3.1 Our Compiler

Given any BE scheme $\mathsf{B} = (\mathsf{SetUp}, \mathsf{BEnc}, \mathsf{Dec})$, our compiler constructs a BE scheme $\overline{\mathsf{B}} = (\overline{\mathsf{SetUp}}, \overline{\mathsf{BEnc}}, \overline{\mathsf{Dec}})$ as follows:

- $\overline{\mathsf{SetUp}}$: For given security parameter 1^λ and a set $\mathcal{U}$ of group users, the algorithm performs the following:
  - First $\overline{\mathsf{SetUp}}$ makes a complete w-ary tree $\mathcal{T}_{|w|}$ in which each leaf is associated to each user. Next, (if necessary) $\overline{\mathsf{SetUp}}$ constructs a user structure for each sibling set in $\mathcal{T}_{|w|}$ according to B.
  - Independently running SetUp of B on each sibling set $S_{b_0 b_1 \cdots b_j}$, ($0 \leq j \leq s$-1), $\overline{\mathsf{SetUp}}$ assigns keys to each node (including an interior node). For distinction we denote the BE scheme B and its SetUp applied to $S_{b_0 b_1 \cdots b_j}$ by $\mathsf{B}_{b_0 b_1 \cdots b_j}$ and $\mathsf{B}.\mathsf{SetUp}_{b_0 b_1 \cdots b_j}$, respectively. That is, each node (which

is not actually a user in the tree) in $S_{b_0 b_1 \cdots b_j}$ is assigned user keys by $\mathsf{B}_{b_0 b_1 \cdots b_j}$. Let $K_{b_0 b_1 \cdots b_j b_{j+1}}$ be the set of keys assigned to a node $v_{b_0 b_1 \cdots b_j b_{j+1}}$ in $S_{b_0 b_1 \cdots b_j}$. $\overline{\mathsf{SetUp}}$ then provides each leaf $v_{b_0 b_1 \cdots b_s}$ (i.e., user) with a set

$$UK_{v_{b_0 b_1 \cdots b_s}} = K_{b_0} \cup K_{b_0 b_1} \cup \cdots \cup K_{b_0 b_1 \cdots b_s},$$

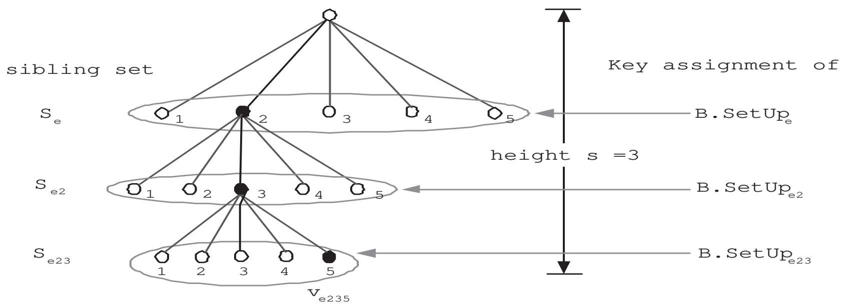
where K_{b_0} is a singleton set of an initial session key.

- $\overline{\mathsf{BEnc}}$: For given message M and a set $\mathcal{R}$ of r revoke users, it performs the followings to generate a broadcast message: it first makes the Steiner Tree $\mathcal{ST}$ induced by $\mathcal{R}$, that is, the minimal subtree of $\mathcal{T}_{|w|}$ which connects the root of $\mathcal{T}_{|w|}$ to all leaves in $\mathcal{R}$. Starting from $\mathcal{ST}$ as an initial tree, it recursively removes leaves from $\mathcal{ST}$ until $\mathcal{ST}$ becomes a single node.
  1. Find a sibling set S consisting of leaves of $\mathcal{ST}$.
  2. If $|S|=w$, then it removes from $\mathcal{ST}$ all leaves in S and makes their parent node a leaf.
  3. Otherwise, it applies revocation method of BEnc to S and generates ciphertexts of a group session key. Then it removes all leaves in S from $\mathcal{ST}$ and makes their parent node a leaf.

- $\overline{\mathsf{Dec}}$: For given legal user $v_{b_0 b_1 \cdots b_s} \in \mathcal{U} \backslash \mathcal{R}$, it first finds the user's ancestor $v_{b_0 b_1 \cdots b_t}$ in the lowest level such that $v_{b_0 b_1 \cdots b_t c_{t+1} \cdots c_s}$ is a revoked user. To decrypt a group session key, it uses a key assinged to revoke a node $v_{b_0 b_1 \cdots b_t c_{t+1} \cdots c_s}$ from $S_{b_0 b_1 \cdots b_t}$.

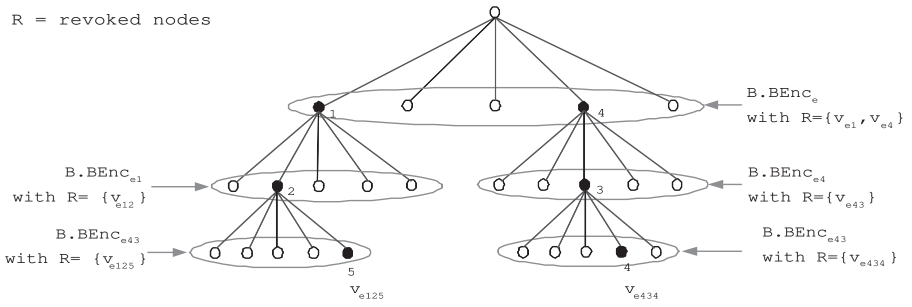
As an example shown in Figure 1, we consider a complete 5-ary tree with height 3 for a set of 125 users $\mathcal{U}=\{u_1, \cdots, u_{125}\}$. A leaf v_{e235}, which is associated with user u_{40}, receives a set of keys $UK_{v_{e235}}=K_e \cup K_{e2} \cup K_{e23} \cup K_{e235}$ where K_e is a singleton set of an initial group session key, K_{e2} is a set of keys assigned to a node v_{e2} in sibling set S_{e2} by $\mathsf{B.SetUp}_{e2}$, K_{e23} is a set of keys assigned to a node $v_{\varepsilon 23}$ in sibling set S_{e23} by $\mathsf{B.SetUp}_{e23}$ and K_{e235} is a set of keys assigned to a node v_{e235} in the sibling set S_{e235} by $\mathsf{B.SetUp}_{e235}$, as in Figure 1.

Fig. 1. Key assignment in our compiler : a complete 5-ary tree



To revoke $\{v_{e125}, v_{e434}\}$, as in Figure 2, consider the minimal subtree $\mathcal{ST}$ which connects the root to the leaves v_{e125} and v_{e434}. Taking all nodes with a same parent in $\mathcal{ST}$ revoked in their sibling set S_α, we apply revocation method of B_α to the sibling set S_α. Revocation methods of B_{e12}, B_{e43}, B_{e1}, B_{e4}, B_e are sequentially applied to the sibling sets S_{e12}, S_{e43}, S_{e1}, S_{e4}, S_e in a bottom-up manner, respectively.

Fig. 2. Revocation in our compiler



In the construction of our compiler, a single broadcast encryption scheme are independently applied to each sibling set in $\mathcal{T}_{|w|}$. But the construction allows that different broadcast encryption schemes are applied to different sibling sets, in order to provide flexibility depending on the resource restriction of client devices. We observe that nodes in the higher level (i.e., closer to the root) become useless more quickly as revoked users are uniformly distributed. Utilizing this observation, we can use a BE scheme assigning less keys per node at a higher level, which will increase the number of transmission messages slightly during initial period. This must be a good trade-off because the initial transmission overhead is relatively small.

Basically the security of our modular method is based on the security of a given BE scheme and the independence usage of the scheme. By using a standard hybrid argument, we can prove the following lemma. The proof will appear in the full version of the paper.

Lemma 1. *The compiled scheme preserves the security of the underlying broadcast encryption scheme.*

3.2 Performance Analysis

We analyze efficiency of the presented compiler with respect to three efficiency parameters: transmission overhead, user storage overhead, computational overhead at a user device.

User Storage Overhead. In a compiled BE scheme, the number of keys that a user should store is $|UK_{v_{b_0 b_1 b_2 \cdots b_s}}| = |K_{b_0}| + |K_{b_0 b_1}| + \cdots + |K_{b_0 b_1 \cdots b_s}| = 1 + s \cdot SO_{\mathsf{B}}(n^{1/s})$. BE schemes satisfying $\mathcal{O}(\log n)$ storage restriction have been considered important [9] since they are well suited to low-memory devices in wireless mobile networks. We note that the compiled BE scheme $\bar{\mathsf{B}}$ preserves $\mathcal{O}(\log n)$ key restriction of the underlying BE scheme B. Concretely, $SO_{\bar{\mathsf{B}}}(n)$ is $\mathcal{O}(\log n)$ since $1 + s \cdot SO_{\mathsf{B}}(n^{1/s}) \leq 1 + s \cdot (c \cdot \log_w n^{\frac{1}{s}} + 1) = 1 + (c+1) \cdot \log_w n \leq 1 + (c+1) \cdot \log_2 n$ where c is a constant factor. If storage size in the underlying scheme is less than $\log_w n$ such as a constant then storage size in the compiled scheme increases up to $\log_w n$ which is still satisfying $\mathcal{O}(\log n)$ storage restriction.

Computation Overhead. In a compiled BE scheme, the maximum number of the basic operations which a user should perform is $CO_{\mathsf{B}}(n^{1/s}) (= CO_{\mathsf{B}}(n^{\frac{1}{\log_w n}}))$ since the size of each sibling set at each level is $n^{1/s}$. If $s = \frac{\log_w n}{\log_w(\log_w n)}$ then $n^{1/s} = \log_w n$. If t different BE schemes B_i $(1 \leq i \leq t)$ are used for sibling sets in the setup algorithm then $CO_{\overline{\mathsf{B}}}(n) =$ Max $\{CO_{\mathsf{B}_i}(n^{1/s}) | 1 \leq i \leq t\}$.

Transmission Overhead. Generally it is not easy to analyze the asymptotic behavior of transmission overhead in compiled BE schemes since BE schemes show various transmission overhead. However we assume that transmission overhead in a given BE scheme is monotone increasing (possibly non-decreasing) as the number of revoked users increases. In this case, transmission overhead $TO_{\overline{\mathsf{B}}}(r,n)$ in a compiled BE scheme is upper-bounded by $s \cdot TO_{\mathsf{B}}(r, n^{1/s})$.

In particular, if a given BE scheme satisfies the Subset-Cover framework we can concretely show that $TO_{\overline{\mathsf{B}}}(r,n)$ is recursively described as follows:

$$\begin{array}{ll} r(s-i-1)TO_{\mathsf{B}}(1,\omega) + (r \bmod \omega^i) TO_{\mathsf{B}}(1 + \lceil \frac{r-\omega^i}{\omega^i} \rceil, \omega) & \text{if } \omega^i \leq r < \omega^i \gamma, \\ \qquad + (\omega^i - (r \bmod \omega^i)) TO_{\mathsf{B}}(\lceil \frac{r-\omega^i}{\omega^i} \rceil, \omega) & \\ r(s-i-1)TO_{\mathsf{B}}(1,\omega) + (\omega^{i+1} - r) & \text{if } \omega^i \gamma \leq r < \omega^{i+1}. \end{array}$$

where $\omega = n^{1/s}$ and γ is a number such that the maximum number of transmission ciphertexts in B for γ revoked users is $n - \gamma$. The concrete analysis appears in Appendix.

4 Compiled Instances

We apply our compiler to several transmission-efficient schemes, which have inefficiency in computational overhead or user keys storage for huge number of users, to gain scalable and efficient BE schemes. The transformation provides reduction in user storage and/or computation overhead by slightly increasing transmission overhead of a given BE scheme.

4.1 Broadcast Encryption Scheme for User Devices with Low-Resource

In this section we present a BE scheme which achieves $\mathcal{O}(\log n)$ user storage, $\mathcal{O}(\log n)$ computation overhead, and $\mathcal{O}(\frac{\log n}{\log(\log n)}r)$ transmission overhead at the same time. To achieve this goal, we first construct a BE scheme $\mathsf{B1}$ which requires $2r$ transmission messages and only $1+\log_2 n$ key storage per user, but n operations per user. Next, by applying the compiler to $\mathsf{B1}$, we gain the desired scheme.

Broadcast Encryption Scheme $\mathsf{B1}$. As a structure of $\mathsf{B1}$ scheme we consider a segment of the number line $\mathcal{L}$ where numbers are linearly ordered by their magnitude. For any points i and j $(\geq i)$, we denote the set $\{k|i \leq k \leq j\}$, called as a closed interval, by $S_{[i,j]}$. For example, $S_{[2,6]}$=$\{2,3,4,5,6\}$.

We define two *one-way chains*, $\mathcal{C}^{+}_{[i,j]}$ and $\mathcal{C}^{-}_{[i,j]}$ associated with $S_{[i,j]}$, and, for a given function F: $\{0,1\}^{\ell} \rightarrow \{0,1\}^{\ell}$, *chain-values* corresponding to them as follows:

- $\mathcal{C}^{+}_{[i,j]}$ is a one-way chain such that starts from i and positively goes through $i+1,\cdots, j-1$ and then ends at j. For a given $\mathsf{sd}_i \in \{0,1\}^{\ell}$, the chain-value of $\mathcal{C}^{+}_{[i,j]}$ is $F^{|j-i|}(\mathsf{sd}_i)$.
- $\mathcal{C}^{-}_{[i,j]}$ is a one-way chain such that starts from j and negatively goes through $j-1,\cdots, i+1$ and then ends at i. For a given $\mathsf{sd}_j \in \{0,1\}^{\ell}$, the chain-value of $\mathcal{C}^{-}_{[i,j]}$ is $F^{|j-i|}(\mathsf{sd}_j)$.

$F^{d}(\mathsf{sd})$ is computed by repeatedly applying the function F to sd d times.

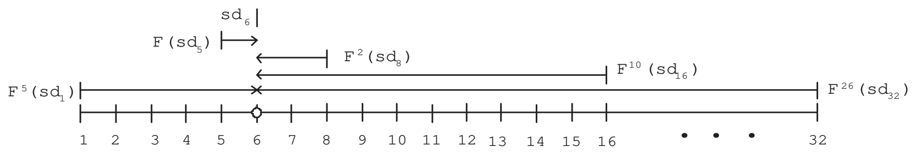
SetUp. For a given security parameter 1^{λ} and a set $\mathcal{U}$ of users, the algorithm SetUp performs the following: First it arranges all users in $\mathcal{U}$ on a segment of the number line $\mathcal{L}$ linearly by the magnitude. A point i in $\mathcal{L}$ is associated with a user u_i. Next, to give a user a set of private keys, it executes the following key assignment.

> Starting from $S_{[1,n]}$ as an initial closed interval SetUp performs the following recursively: For a given closed interval $S_{[i,j]}$ for $1 \leq i < j \leq n$, SetUp selects random and independent labels sd_i and sd_j, and assigns these to users u_i and u_j. SetUp computes chain-values by consecutively applying F to labels sd_i and sd_j, respectively. Then SetUp assigns $F^{k-i}(\mathsf{sd}_i)$ and $F^{j-k}(\mathsf{sd}_j)$ to a user u_k. Next SetUp divides the closed interval $S_{[i,j]}$ to get two sub-intervals $S_{[i,m]}$ and $S_{[m+1,j]}$ where m=$\frac{i+j-1}{2}$. While a sub-interval is not a singleton, SetUp applies the above assignment method to the sub-intervals repeatedly. The label sd_i (sd_j), which is assigned to the previous closed interval, is reused in a sub-interval $S_{[i,m]}$ $(S_{[m+1,j]})$ and label sd_m (sd_{m+1}) is newly selected and assigned to a user $u_m(u_{m+1}$, respectively).
>
> By using the above method, SetUp provides a user with $1+\log_2 n$ keys since $1+\log_2 n$ closed intervals including the user are gained from the above binary division and, for each interval, only one key value is newly assigned

to the user.

For an example, for $\mathcal{U}=\{u_1,\cdots, u_{32}\}$, SetUp provides a user u_6 with 6 (=$1+\log_2 2^5$) keys, i.e., chain-values $F^5(\mathsf{sd}_1)$, $F^{26}(\mathsf{sd}_{32})$, $F^{10}(\mathsf{sd}_{16})$, $F^2(\mathsf{sd}_8)$, $F(\mathsf{sd}_5)$, and sd_6 associated to 4 closed intervals, $S_{[1,32]}$, $S_{[1,16]}$, $S_{[1,8]}$, $S_{[5,8]}$ and $S_{[5,6]}$, as in Figure 3.

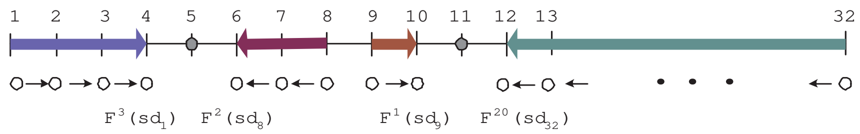
Fig. 3. Key assignment to u_6 in $\mathsf{B1}$



$\mathsf{Broadcast\ Encryption}$. The revocation method of $\mathsf{B1}$ is based on the following singleton revocation: For a given closed interval $S_{[i,j]}$ of $\mathcal{L}$, to revoke a user u_t, that is, a point $t{\in}S_{[i,j]}$, the remaining users are covered by two one-way chains $\mathcal{C}^+_{[i,t-1]}$ and $\mathcal{C}^-_{[t+1,j]}$, which proceed from each end point toward opposite directions. The use of these two chains obviously excludes a point t in a subset $S_{[i,j]}$. The keys associated with $\mathcal{C}^+_{[i,t-1]}$ and $\mathcal{C}^-_{[t+1,j]}$ are $F^{t-i}(\mathsf{sd}_i)$ and $F^{j-t}(\mathsf{sd}_j)$, respectively.

For given r revoked users, BEnc applies the above single revocation method to each disjoint sub-interval including one with one revoked user. In order to apply the method systematically BEnc uses a binary division. That is, for a given set of revoked points $\mathcal{R}=\{u_{i_1},...,u_{i_r}\}$, BEnc finds a division point d_i firstly separating each pair of consecutive revoked nodes u_{i_j} and u_{i_j+1} by performing a binary search on $\mathcal{L}$. BEnc then partitions $\mathcal{L}$ so that $\mathcal{L} = S_{[d_0,d_1]}{\cup}S_{[d_1+1,d_2]}{\cup}\cdots\cup S_{[d_{r-1}+1,d_r]}$ where $i_j{\in}S_{[d_{j-1},d_j]}$, d_0=1 and d_r=n. Finally BEnc covers each subset by using the above single revocation method.

Fig. 4. Revocation in $\mathsf{B1}$



For example, as shown in Figure 4, for $\mathcal{U}=\{u_1,\cdots,u_{32}\}$ and $\mathcal{R}=\{u_5,u_{11}\}$, the set $\mathcal{U}\backslash\mathcal{R}$ of remaining users is partitioned as follows:

$$\mathcal{L}\backslash\{5,11\}=S_{[1,8]}\cup S_{[9,32]}=(\mathcal{C}^{+}_{[1,4]}\cup\mathcal{C}^{-}_{[6,8]})\cup(\mathcal{C}^{+}_{[9,10]}\cup\mathcal{C}^{-}_{[12,32]}).$$

Then four keys $F^3(\mathsf{sd}_1)$, $F^2(\mathsf{sd}_6)$, $F^1(\mathsf{sd}_9)$, and $F^{20}(\mathsf{sd}_{32})$ are assigned to four one-way chains, $\mathcal{C}^{+}_{[1,4]}$, $\mathcal{C}^{-}_{[6,8]}$, $\mathcal{C}^{+}_{[9,10]}$, and $\mathcal{C}^{-}_{[12,32]}$, respectively.

After construction of cover sets, BEnc applies another one-way function F' to the chain-values and then uses the resulting values as keys to encrypt a group session key.

Decryption. For given legal user $u_k \in \mathcal{U}\backslash\mathcal{R}$, the decryption algorithm Dec first finds two consecutive revoked users u_{i_j} and $u_{i_{j+1}}$ such that $k\in S_{[i_j,i_{j+1}]}$. Next, by using a binary search, Dec finds the division point d which firstly separates two points i_j and i_{j+1}. If $d \geq k$ then it computes $F^{k-i_j}(F^{d-k}(\mathsf{sd}_d))=F^{d-i_j}(\mathsf{sd}_d)$. Otherwise, it computes $F^{i_{j+1}-k}(F^{k-d-1}(\mathsf{sd}_{d+1}))=F^{i_{j+1}-d-1}(\mathsf{sd}_{d+1})$.

Security. We can easily show the correctness of B1 that every privileged user can decrypt an encrypted group session key. Revoked users are excluded by one-wayness of one-way chain and so cannot obtain useful information to decrypt an encrypted group session key. Formally, we show that B1 scheme is resilient to collusion of any set of revoked users by using the following lemma and the similar idea in [16]. In the lemma we assume that F and F' are pseudo random permutations in the sense that no probabilistic polynomial-time adversary can distinguish the output of F (and F') on a randomly selected input from a truly random string of similar length with non-negligible probability.

Lemma 2. *The above key assignment satisfies the key-indistinguishability condition [16] under the pseudo-randomness of given functions F and F'.*

We can prove the lemma by using a hybrid argument on the length of one-way chains, i.e., showing that the gap between true randomness and pseudo-randomness is negligible.

Efficiency. In the presented scheme, at most two ciphertexts of a group session key per revoked user are generated. Hence the number of total ciphertexts consisting of a header Hdr is at most $2{\cdot}r$ for r revoked users. But computation overhead is proportional to n.

When we apply the compiler to B1 scheme, in the resulting scheme $\overline{\mathsf{B1}}$, the compiled BE scheme $\overline{\mathsf{B1}}$ satisfies $\mathcal{O}(\log n)$ key restriction since user keys storage in the original BE scheme B1 is $1 + \log_2 n$. However, we can show that user storage overhead does not change, i.e., $1 + \log_2 n$ since one private node key assigned to the parent node of a given node can be deleted and so $1 - s + s{\cdot}(\log_2 n^{\frac{1}{s}}+1) = 1 + \log_2 n$.

Computation overhead is reduced to $\mathcal{O}(n^{\frac{1}{s}})$ for $s=\log_w n$. If we choose the variables $w=n^{\frac{\log_2(\log_2 n)}{\log_2 n}}$ and $s=\frac{\log_2 n}{\log_2(\log_2 n)}$ then we also reduce $\mathcal{O}(n^{\frac{1}{s}})$ computation overhead to $\mathcal{O}(\log n)$. However transmission overhead slightly increases by at most a factor of s from $2r$. More precisely, transmission overhead is described by the recursive formula in Session 3.2 since $\mathsf{B1}$ satisfies the Subset-Cover framework.

Remark. Based on a similar approach using one-way chains, Goodrich et al. [9] presented the SSD (stratified subset difference) scheme for low-memory devices. But, unlike the work in [9], our method does not use a tree structure. This eliminates the cost for traversing internal nodes in the tree, which causes increase in computation overhead. In addition, with respect to efficiency, the SSD scheme achieves $\mathcal{O}(\log n)$, more precisely $2d \log_2 n$, user storage overhead and $\mathcal{O}(r)$ transmission overhead, but $\mathcal{O}(n^{\frac{1}{d}})$ computation overhead where d is a predetermined constant. When $\mathcal{O}(\log n)$ computation restriction is strictly required, the constant d should be as large as $\frac{\log_2 n}{\log_2(\log_2 n)}$ and user storage overhead also becomes, rather than $\mathcal{O}(\log n)$, closer to $\mathcal{O}(\log^2 n)$, which is relatively heavy and so undesirable in memory-constrained environments.

4.2 Transmission-Efficient Broadcast Encryption Schemes

In this section, to construct a scalable transmission-efficient BE schemes, we further apply our compiler to a previously known transmission-efficient BE scheme, but inefficient in computation cost and user storage size for a huge number of users.

Recently, Jho et al. [14] have presented BE schemes where the number of transmission messages is less than the number of revoked users r, i.e., $\frac{1}{k} \cdot r$ for a predetermined constant k. To bring the number of transmission messages down, they used a fine strategy to cover several subsets of privileged users by using only one key. Their basic scheme requires $\mathcal{O}(n^k)$ user storage overhead and $\mathcal{O}(n)$ computation overhead. To reduce storage and computation overhead further, they presented interval or partition-based construction to deal with relatively small number of users. Unfortunately, in their methods, user storage overhead is still heavy or initial transmission length is relatively large.

By applying our compiler to their schemes, we construct a scalable BE scheme $\overline{\mathsf{B2}}$, which has $\frac{1}{2} \cdot r$ transmission messages (except only for a small number of revoked users) with a reasonable user storage and computation overhead. As the underlying schemes for our compiler, we apply two different BE schemes in [14] at different depth of a w-ary tree. One is the BE scheme using simple one-way ring where the number of transmission messages is r. This scheme is applied to every sibling set not in the bottom level. The other is the BE scheme based on a so-called $\mathsf{HOC}(2,[m,2])$, which is a combination of $\mathsf{HOC}(2{:}m)$ with simple hierarchical ring with depth 2 and $\mathsf{OFBE}(m{:}2)$ using *1-jump* one-way chain. $\mathsf{HOC}(2,[m,2])$ has $\lfloor \frac{1}{2} r \rfloor + 1$ transmission ciphertexts and relatively low user storage compared to that of the one-way chain-based scheme. This scheme is applied to every sibling set in the bottom level, i.e., sibling set consisting of leaves in the tree. The efficiency for $\overline{\mathsf{B2}}$ is as shown in Table 1.

Table 1. Efficiency of $\overline{\mathsf{B2}}$

	$TO_{\overline{\mathsf{B2}}}(r, n)$	$SO_{\overline{\mathsf{B2}}}(n)$	$CO_{\overline{\mathsf{B2}}}(n)$
$\overline{\mathsf{B2}}$	$\leq \frac{1}{2}r+n^{\frac{s-1}{s}}$	$(s-1)n^{1/s}+\frac{(n^{1/s})^2-2n^{1/s}+24}{8}-s$	$\mathcal{O}(n^{\frac{1}{s}})$

We note that, for $w(=n^{1/s})=100$, $n^{(s-1)/s}$ is less than 1% of n. The compiled scheme $\overline{\mathsf{B2}}$ provides similar (or less) transmission overhead, compared to the schemes in [14] while gains reasonably low user storage and computation overhead. For comparison between the schemes, refer to Session 5.

Similarly applying our compiler to other BE schemes such as BE schemes based on a secret sharing [3, 15], one-way accmulators [2], or complicated operations etc., gives scalable transformations of these BE schemes under different security assumptions in information theoretical or computational aspects.

5 Efficiency Comparison between Proposed Schemes

In this section we compare the efficiency between our compiled BE schemes with SD [16], LSD [13], SSD [9], (1,100)-π_1 [14] schemes. In the following we assume that the size of a key is 128 bits, which is considered reasonably secure currently. The number of computations means the number of basic operations needed to compute a key encrypting a group session key.

For a specific example, we consider the case of n=10^8 users and w=100. As we show in Figure 5, the number of transmission ciphertexts of $\overline{\mathsf{B2}}$ is similar to that of the SD scheme at initial interval where the number r of revoked users is smaller than 0.75 % of the total users. But, except this interval, the number of transmission messages of $\overline{\mathsf{B2}}$ becomes, at worst case, about $\frac{1}{4}$ of the number of transmission messages of the SD scheme. The number of keys stored by a user in $\overline{\mathsf{B2}}$ is about 4 times as many as that of the SD scheme. But this difference is acceptable in many applications.

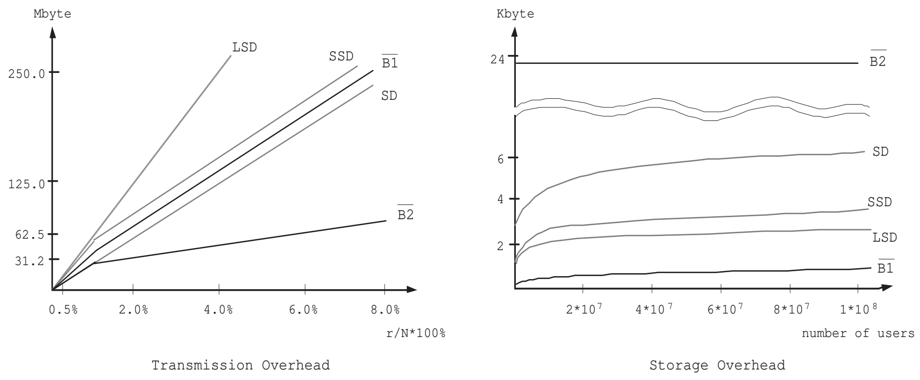
In particular, $\overline{\mathsf{B1}}$ satisfies log-key restriction strictly, and suitable to low-memory applications where the memory is less 1 Kbyte such as a smart card. This allows a message sender to revoke any r users with transmission overhead being similar to that of the SD scheme [16].

Table 2. Comparison between $\overline{\mathsf{B1}}$, $\overline{\mathsf{B2}}$, SD[16], LSD[13], and SSD[9] for n=10^8

Scheme	Transmission Overhead	User Storage Overhead	Computation Overhead
SD [16]	$\leq 2r-1$	368 (5.74$Kbyte$)	27
LSD [13]	$\leq 4r$	143 (2.24$Kbyte$)	27
SSD [9]	$\leq 2sr(s=4)$	213 (3.33$Kbyte$)	100
(1,100)-π_1 [14]	$\leq 2r$+0.01n	5274 (82.4$Kbyte$)	100
$\overline{\mathsf{B1}}$	$\leq 2r$+0.01n	27 (0.422$Kbyte$)	100
$\overline{\mathsf{B2}}$	$\leq 0.5r$+0.01n	1528 (23.875$Kbyte$)	100

In Table 2. "$\leq$" in the first column means upper-bound of the number of transmission ciphertexts of a group session key. Since the original BE schemes $\mathsf{B1}$

Fig. 5. Transmission and storage overhead for $n{=}10^8$ for the worst case



and $\mathsf{B2}$ are defined in Subset-Cover framework transmission overheads in the compiled schemes $\overline{\mathsf{B1}}$ and $\overline{\mathsf{B2}}$ are described by the recursive formula in Section 3.2. More concretely, if $10^{-4}n \leq r \leq 10^{-2}n$ then $TO_{\overline{\mathsf{B1}}}(r,n) \leq 4r{+}10^{-4}n$ and $TO_{\overline{\mathsf{B2}}}(r,n) \leq r{+}10^{-4}n$. Else $10^{-6}n \leq r \leq 10^{-4}n$ then $TO_{\overline{\mathsf{B1}}}(r,n) \leq 6r{+}10^{-4}n$ and $TO_{\overline{\mathsf{B2}}}(r,n) \leq 1.5r{+}10^{-4}n$.

6 Conclusion

We have presented a modular method transforming broadcast encryption schemes, which are impractical due to computation complexity or user keys storage for huge number of users, to scalable ones. As concrete examples, we have presented some compiled instances: The first is a BE scheme achieving $\mathcal{O}(\log n)$ user storage, $\mathcal{O}(\log n)$ computation overhead, and $\mathcal{O}(\frac{\log n}{\log\log n} \cdot r)$ transmission overhead at the same time. The second is a transmission-efficient BE scheme with a reasonably low user storage and computation overhead.

For all schemes based on the Subset-Cover framework, our compiler provides a traitor tracing method by using a similar method in [16]. Further study would be a method to apply our modular approach to other traitor tracing methods.

Acknowledgments

The authors would like to thank the anonymous reviewers of CRYPTO 2005 for giving helpful comments.

References

1. T. Asano, *A Revocation Scheme with Minimal Storage at Receivers*, In Advances in Cryptology-Asiacrypt 2002, Springer-Verlag, LNCS vol. 2501, pp.433-450, 2002.

2. N. Attrapadung, K. Kobara and H. Imai, *Broadcast Encryption with Short Keys and Transmissions*, ACM Workshop On Digital Rights Management 2003, pp.55-66, 2003.
3. J. Anzai, N. Matsuzaki and T. Matsumoto, *Quick Group Key Distribution Scheme with Entity Revocation*, In Advances in Cryptology-Asiacrypt 1999, Springer-Verlag, LNCS vol. 1716, pp.333-347, 1999.
4. M. Bellare, A. Desai, E.Jokipii and P. Rogaway, *A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation*, In Proceedings of the 38th Annual Symposium on Foundations of Computer Science - FOCS'97, pp.394-403, 1997.
5. D. Boneh, C. Gentry and B. Waters, *Collusion Resistant Broadcst Encryption With Short Ciphertexts and Private Keys*, Available from http://eprint.iacr.org 2005. To appear in CRYPTO 2005.
6. D. Boneh and A. Silverberg, *Applications of Multilinear Forms to Cryptography*, Available from http://eprint.iacr.org 2002.
7. D. Dolev, C. Dwork, M. Naor, Pinkas, *Nonmalleable Cryptography,*, SIAM Journal on Discrete Mathematics, 30(2), pp.391-437, 2000.
8. A. Fiat and M. Naor, *Broadcast Encryption*, In Advances in Cryptology-CRYPTO 1994, Springer-Verlag, LNCS vol. 773, pp.480-491, 1994.
9. M. T. Goodrich, J. Z. Sun, and R. Tamassia, *Efficient Tree-Based Revocation in Groups of Low-State Devices*, In Advances in Cryptology-CRYPTO 2004, Springer-Verlag, LNCS vol. 3152, pp.511-527, 2004.
10. J. A. Garay, J. Staddon, and A. Wool, *Long-Lived Broadcast Encryption*, In Advances in Cryptology-CRYPTO 2000, Springer-Verlag, LNCS vol. 1880, pp.333-352, 2000.
11. E. Gafni, J. Staddon, and Y. L. Yin, *Efficient Methods for Integrating Traceability and Broadcast Encryption*, In Advances in Cryptology-CRYPTO 1999, Springer-Verlag, LNCS vol. 1666, pp.372-387, 1999.
12. J. Horwitz, *A Survey of Broadcast Encryption*, Manuscript, 2003.
13. D. Halevy and A. Shamir, *The LSD Broadcast Encryption Scheme*, In Advances in Cryptology-CRYPTO 2002, Springer-Verlag, LNCS vol. 2442, pp.41-60, 2002.
14. N.-S. Jho, J. Y. Hwang, J. H. Cheon, M. Kim, D. H. Lee and E. S. Yoo, *One-way chain Based Broadcast Encryption Scheme*, In Advances in Cryptology-Eurocrypt 2005, Springer, LNCS vol. 3494, pp.559-574, 2005.
15. M. Naor and B. Pinkas, *Efficient Trace and Revoke Scheme*, Financial Cryptography FC 2000, Springer-Verlag, LNCS vol. 1962, pp.1-20, 2000.
16. D. Naor, M. Naor, and J. Lotspiech, *Revocation and Tracing Schemes for Stateless Receivers*, In Advances in Cryptology-CRYPTO 2001, Springer-Verlag, LNCS vol. 2139, pp.41-62, 2001.
17. D. R. Stinson and T. V. Trung, *Some New Results on Key Distribution Patterns and Broadcast Encryption*, Designs, Codes and Cryptography, vol 14., no. 3, pp.261-279, 1998.
18. D. R. Stinson and R. Wei, *Combinatorial Properties and Constructions of Traceability Schemes and Frameproof Codes*, SIAM Journal on Discrete Mathematics, vol 11., no. 1, pp.41-53, 1998.
19. D. M. Wallner, E. G. Harder, and R. C. Agee, *Key Agreement for Multicast :Issues and Architecture*, In internet draft draft-waller-key-arch-01.txt, Sep, 1998.
20. C. K. Wong and S. S. Lam, *Digital Signatures for Flows and Multicasts*, IEEE/ACM Transactions on Networking, vol. 7, no. 4: pp. 502-513, 1999.

A Analysis of Transmission Efficiency of Our Compiler

Let $\omega{=}n^{1/s}$ and γ be a number satisfying $TO_B(\gamma,\omega){=}\omega{-}\gamma$. To analyze transmission efficiency, we use the following observations: The worst case occurs when revoked users have the least number of common ancestors. If there is no revoked user, then GC uses an initial group session key to cover all users and hence there is no transmission messages (ciphertext of the group session key). If $r = 1$, we obtain the formula (1) since there is one revoked node in each level and so total s sibling sets to be covered, and $TO_B(1,\omega)$ transmission messages for each sibling set are required. If $2 \leq r < \omega$, then there is r revoked nodes in each level and total $r(s-1)$ sibling sets should be covered. In this case, if $\gamma < r$, then $\omega - r$ transmission messages are transmitted for the first level. Therefore, we obtain the formula (2) and (3) for $2 \leq r < \omega$. If $\omega \leq r < \omega\gamma$, then we do not need to consider nodes in the first level since all nodes in the first level are revoked. In level 2, $(r \mod \omega)$ sibling sets have $1+\lceil\frac{r-\omega}{\omega}\rceil$ revoked nodes and $\omega-(r \mod \omega)$ sibling sets have $\lceil\frac{r-\omega}{\omega}\rceil$ revoked node. In level j $(3 \leq j \leq s)$, $r(s-2)$ messages should be transmitted to cover $r(s-2)$ sibling sets since one revoked node exists in r sibling sets in level j $(1 \leq j \leq s-2)$. Hence we obtain the formula (4).

Now we can easily generalize the formula (3) and (4) to the formula (5) and (6), and again the formula (5) and (6) to get the formula (7), (8), (9) inductively.

$$
\begin{aligned}
r = 1,&\quad sTO_B(1,\omega) && (1)\\
2 \leq r < \gamma,&\quad r(s-1)TO_B(1,\omega) + TO_B(r,\omega) && (2)\\
\gamma \leq r < \omega,&\quad r(s-1)TO_B(1,\omega) + (\omega - r) && (3)\\
\omega \leq r < \omega\gamma,&\quad r(s-2)TO_B(1,\omega) + (r \bmod \omega)TO_B(1+\lceil\tfrac{r-\omega}{\omega}\rceil,\omega)\\
&\qquad + (\omega - (r \bmod \omega))TO_B(\lceil\tfrac{r-\omega}{\omega}\rceil,\omega) && (4)\\
\vdots\qquad&\qquad\qquad\vdots\\
\omega^{i-1}\gamma \leq r < \omega^i,&\quad r(s-i)TO_B(1,\omega) + (\omega^i - r) && (5)\\
\omega^i \leq r < \omega^i\gamma,&\quad r(s-i-1)TO_B(1,\omega)\\
&\qquad + (r \bmod \omega^i)TO_B(1+\lceil\tfrac{r-\omega^i}{\omega^i}\rceil,\omega)\\
&\qquad + (\omega^i-(r \bmod \omega^i))TO_B(\lceil\tfrac{r-\omega^i}{\omega^i}\rceil,\omega) && (6)\\
\vdots\qquad&\qquad\qquad\vdots\\
\omega^{s-2}\gamma \leq r < \omega^{s-1},&\quad rTO_B(1,\omega) + (\omega^{s-1} - r) && (7)\\
\omega^{s-1} \leq r < \omega^{s-1}\gamma,&\quad (r \bmod \omega^{s-1})TO_B(1+\lceil\tfrac{r-\omega^{s-1}}{\omega^{s-1}}\rceil,\omega)\\
&\qquad + (\omega^{s-1}-(r \bmod \omega^{s-1}))TO_B(\lceil\tfrac{r-\omega^{s-1}}{\omega^{s-1}}\rceil,\omega) && (8)\\
\omega^{s-1}\gamma \leq r < n,&\quad n - r && (9)
\end{aligned}
$$