Special English Edition of Journal of KIISC, Vol. 18 No.6 (B), December 2008

Public key broadcast encryption scheme using new converting method

(Invited paper)

Nam-Su Jho[1], *Eun Sun Yoo*[2] *and Man Young Rhee*[3]
[1]Cryptography Research Team, ETRI, Daejeon, Korea
[2]Security Consulting Business Team, Samsung SDS, Seoul, Korea
[3]Endowed Chair Professor, Kyung Hee University, Suwon, Korea

Abstract

Broadcast encryption is a cryptographical primitive which is designed for a content provider to distribute contents to only privileged qualifying users through an insecure channel. Anyone who knows public keys can distribute contents by means of public key broadcast encryption whose technique can also be applicable to many other applications. In order to design public key broadcast encryption scheme, it should devise some methods that convert a broadcast encryption scheme based on symmetric key cryptosystem to a public key broadcast encryption. Up to this point, broadcast encryption scheme on trial for converting from symmetric key setting to asymmetric public key setting has been attempted by employing the *Hierarchical Identity Based Encryption* (HIBE) technique. However, this converting method is not optimal because some of the properties of HIBE are not quite fitting for public key broadcast schemes.

In this paper, we proposed new converting method and an efficient public key broadcast encryption scheme Pub-PI which is obtained by adapting the new converting method to the PI scheme [10]. The transmission overhead of the Pub-PI is approximately $3r$, where r is the number of revoked users. The storage size of Pub-PI is $O(c^2)$, where c is a system parameter of PI and the computation cost is 2 *pairing computations*.

Key words : broadcast encryption, public-key BE, HIBE

I. Introduction

Broadcast Encryption was designed for a content center to distribute some contents to only privileged users through an insecure channel. Even though, anyone can get encrypted contents, privileged users can only decrypt and obtain original contents. Moreover, in a broadcast encryption, it is assumed that set of privileged users changes drastically. Nowadays, there are many applications such as distribution of digital contents through the internet, satellite TV, pay-TV, contents protection of CD or DVD and so on.

In general, most of broadcast encryption systems is assumed to be *stateless receiver*. It means that each user's secret key should not be updated by the center at the initial stage. This assuption makes sense because each user is not always on-line and users' keys might be stored in the temper-resistant device. A *session* is defined as a time interval during which only one message or digital contents is transmitted. Regardless either revocation or not, each user's state should not be changeable during one session. In each session, digital contents are encrypted by a *session key*, *SK*. Thus broadcast encryption can execute in a efficient way so that the center sends safely the session key *SK* to only non-revoked users.

A broadcast encryption system is summarized as follows: In order to distribute a content M, the center first encrypts M using the session key *SK* and then broadcasts the encrypted message together with *header*, which consists of encrypted session keys and some information so that only non-revoked users can decrypt and get *SK*. Every non-revoked user decrypts *SK* from the header of the transmitted message using its own secret key and then recovers M using *SK*, whereas any revoked user cannot obtain *SK*. In particular, broadcast encryption systems should be secure against collusion attack, that is, even when revoked users collude they can not obtain *SK*. When any possible collusion occurs in t revoked users, it is impossible to know *SK*. This is called t-resilient. A

Manuscript received September 30, 2008.
Manuscript revised November 7, 2008.
1) Nam-Su Jho, ETRI, nsjho@etri.re.kr
2) Eun Sun Yoo, Samsung SDS, eunsun.yoo@samsung.com
3) Man Young Rhee, Kyung Hee University, myrhee@tsp.snu.ac.kr

broadcast encryption system should satisfy r-resilient where r is the number of all revoked users in each session.

Discuss on the efficiency of broadcast encryption systems, the following three factors are considered, the *transmission overhead*, the *storage size* and the *computation cost* . These factors are closely related to the length of the header, the size of a user's secret keys and the computing time to decrypt SK from the given header, respectively. Especially, it is most important to minimize the transmission overhead as large as possible for maintaining proper computation cost and storage size.

In 1991, Berkovits [1] introduced the first broadcast encryption scheme. Since then, broadcast encryption schemes have been proposed. At the beginning stage, various broadcast encryption schemes were suggested, but tree based schemes became the main stream in broadcast encryption after Naor *et al.* [11] proposed the *Complete Subtree*(CS) method and the *Subset Difference*(SD) method based on tree structure in 2001. The *Layered Subset Difference*(LSD) method is a improved version of the SD method suggested by Halevy and Shamir [8] in 2002. In 2005 Jho *et al.* [10] introduced new approach, that is, *Punctured Interval*(PI) method based on linear structure which is very efficient broadcast encryption scheme reducing the transmission overhead remarkably. These CS, SD, LSD and PI are similar groups of subset cover system. In subset cover system, the center in advance defines subsets and its corresponding subset keys. Users are assigned some user keys from the center so that only users contained in a subset are able to compute the subset key corresponding to the subset. Then in each session the center divides the set of non-revoked users into several subsets and encrypts the session key by subset keys corresponding to each subset. Because each non-revoked user belongs to at least one subset, he/she can decrypt the message using the corresponding subset key.

These broadcast encryption schemes using subset cover method are basically based on symmetric key cryptosystem. In these schemes, only the trusted third party, typically the center, can encrypt and broadcast contents to users because the center only knows all subset keys. If anyone desires to distribute some contents, the individual need to know all subset keys. Hence in the case that the center and content provider are not same, either the content provider gives the full contents to the center or the center gives all secret subset keys to the provider. It is one of the biggest problems of symmetric key broadcast encryption. On the other side, broadcast encryptions based on public key cryptosystem enable anyone to broadcast some contents to specific users. Recently, as applications of broadcast encryption are complicated, public key broadcast encryptions become more important.

In order to design public key broadcast encryption scheme, it needs to consider methods that convert a broadcast encryption scheme based on symmetric key cryptosystem to a public key broadcast encryption. The simplest way is to convert each subset keys into a pair of public and secret keys. But this conversion process has a serious shortcoming because the size of public keys might be very long. Naor *et al.* [11] already mentioned about this problem and suggested to use of *Identity-Based Cryptography*(IBE) in order to solve the problem. Since in the CS method each subset keys is chosen independently, IBE can be applied to the CS method. However, IBE is not adequate for the SD/LSD methods because each user should store a too many secret keys. In 2002 Dodis and Fazio proposed a method to overcome this shortcoming, which utilizes the notion of *Hierarchical Identity-Based Encyption*(HIBE) in order to reform SD/LSD to the public key setting [5].

Our Main Result: According to the method of Dodis and Fazio, HIBE is directly applicable to the SD/LSD methods. So the transmission overhead of public key SD/LSD schemes are about k times bigger than the original SD/LSD, where k denotes the length of encryption of a HIBE scheme. On the other hand, k represents the depth in the hierarchy in HIBE of Gentry and Silverberg [7]. This method can also be applied to another subset cover system PI in order to meet the requirement toward the public key PI scheme. However this combination leaves much room for improvement since some properties of HIBE are not necessary for public key broadcast schemes.

We propose an efficient public key broadcast encryption scheme Pub-PI which is the extension to the public key setting of PI scheme in the symmetric key setting. This Pub-PI scheme increases the efficiency by removing unnecessary properties of HIBE when HIBE is used in broadcast encryption schemes. The transmission overhead of the Pub-PI is smaller than that of the public key PI obtained by the method of Dodis and Fazio as well as the previous public key SD/LSD. In fact, the transmission overhead of the Pub-PI is approximately $3r$, where r is the number of revoked users. The storage size of Pub-PI is $O(c^2)$, where c is a system parameter of PI and the computation cost is 2 *pairing computations*. This result is obtained by using only basic c-intervals in PI scheme. Actually, the method to design Pub-PI can be applied to any punctured intervals. Using punctured intervals, public key PI schemes with the reduced transmission overhead are realized at the cost of the storage size. That is, there is a trade-off between the transmission overhead and the storage size in our public key PI, such as in the original PI.

II. Preliminaries

1. Hierarchical ID-based encryption

Hierarchical Identity Based Encryption(HIBE) is an extension of Identity Based Encryption(IBE). IBE is a

public key cryptosystem used for identification of users in cooperation with public keys. Consequently, the authentication of public keys is not required and the size of public keys for each user can be reduced. In HIBE each user is included in a hierarchical tree structure and receives hierarchical identity which contains the information about his/her position in the hierarchical structure. Users at a level in the hierarchy can decrypt the message which is encrypted using hierarchical identities of users under the same level. Usually, HIBE has the root PKG as well as many additional lower-level PKGs to execute some load of root PKG. The root PKG generates only private keys of users in the next level; and other private keys are generated by lower-level PKGs. In fact, private keys of users are generated by their parent. So HIBE consists of the following five polynomial-time algorithms, (Root Setup, Lower-level Setup, Extract, Encrypt, Decrypt).

- Root Setup is a probabilistic algorithm which initialize the global parameters of the system by the Root PKG. Given a security parameter 1^{λ}, Root Setup generates system parameters params and a secret key master-key. Then, the PKG publishes params as the global public key and keeps master-key secret.
- Lower-level Setup is a probabilistic algorithm executed by Lower-level PKGs to generate some secret information which may be used by Extract. Given an hierarchical identity HID = (ID_1, … , ID_t), (t > 0) and the corresponding secret key, then Lower-level Setup generates secret information, which is called Lower-level Secret.
- Extract is a (possibly) probabilistic algorithm used by Lower-level PKGs to derive private keys of their children. In other words, the Lower-level PKG with HID = (ID_1, … , ID_t) generates the private key corresponding to any of his children by using HID = (ID_1, … , ID_t, ID_{t+1}), params, Lower-level Secret and returns the private key corresponding to HID = (ID_1, … , ID_t, ID_{t+1}).
- Encrypt is a probabilistic algorithm which sends a message M securely to the user with hierarchical identifier HID and all his/her ancestor. Encrypt takes input params, HID and M and generates a ciphertext C.
- Decrypt is a deterministic algorithm used to recover the message M from a ciphertext C. Decrypt takes input params, HID, C and the private key d corresponding to HID and recovers M.

We present two HIBE algorithms proposed by Gentry and Silverberg [7] in 2002 and by Boneh, Boyen and Goh [2] in 2005.

HIBE - Gentry and Silverberg

- Root Setup: Given a security parameter 1^{λ}, Root Setup generates two cyclic groups $\mathbf{G}_1$, $\mathbf{G}_2$ of a large prime order q and defines a bilinear map $\hat{e}: \mathbf{G}_1 \times \mathbf{G}_1 \rightarrow \mathbf{G}_2$. Namely, for any $P, Q \in \mathbf{G}_1$ and $a, b \in \mathbf{Z}_q$, $\hat{e}(aP, bQ) = \hat{e}(bP, aQ) = \hat{e}(P, Q)^{ab}$. For randomly chosen $P_0 \in \mathbf{G}_1$ of order q and $s_0 \in \mathbf{Z}_q$, output params = ($\mathbf{G}_1, \mathbf{G}_2, \hat{e}, P_0, Q_0, H_1, H_2$), master key = s_0. Here $Q_0 = s_0 P_0$ and $H_1 : \{0,1\}^* \rightarrow \mathbf{G}_1$ and $H_2 : \mathbf{G}_2 \rightarrow \{0,1\}^n$ are cryptographic hash functions, modeled as random oracles (i.e., they output a truly random string on every input), and n is the length of the message encrypted.
- Lower-level Setup: Each user at level $t \geq 1$ picks a random local secret $s_t \in \mathbf{Z}_q$ and keeps it secret. Recall that the root has secret s_0.
- Extract: Assume that every user (ID_1, … , ID_t) at level $t \geq 0$ has a secret point $S_t \in \mathbf{G}_1$ and t-1 'translation points' Q_1, … , $Q_{t-1} \in \mathbf{G}_1$ (notice, Q_0 is in the public key) and we assume that S_0 given to the root is the identity of $\mathbf{G}_1$. Recursively, to assign the secret key to its child (ID_1, … , ID_{t+1}), the parent (ID_1, … , ID_t) computes $P_{t+1} = H_1(ID_1, \ldots, ID_{t+1}) \in \mathbf{G}_1$, picks a random $s_t \in \mathbf{Z}_q$, sets the child's secret point $S_{t+1} = S_t + s_t P_{t+1}$, the child's final translation point $Q_t = s_t P_0$, and sends to the child the values S_{t+1}, Q_t together with its own t-1 translation points Q_1, … , Q_{t-1}. Unwrapping the notation, the child's secret key is ($S_{t+1} = \sum_{i=1}^{t+1} s_i P_i, Q_1 = s_1 P_0, \ldots, Q_t = s_t P_0$).
- Encrypt: To encrypt a message $M \in \{0,1\}^n$ for (ID_1, … , ID_t) using the public value Q_0, compute $P_i = H(ID_1, \ldots, ID_i) \in \mathbf{G}_1$ for all $1 \leq i \leq t$, choose a random $r \in \mathbf{Z}_q$, set $g = \hat{e}(Q_0, rP_1) \in \mathbf{G}_2$ and return $C = [rP_0, M \oplus H_2(g), rP_2, \ldots, rP_t]$. Since the user ($ID_1$, … , ID_t) is unable to decrypt the message using its 'translated' secret point S_{t+1}, additional values rP_2, …, rP_t should be given to the ciphertext. Combining them with secret translation points Q_1, …, Q_{t-1}, the message M is recovered by the following decrypt process.

- Decrypt: To decrypt $C = [U_0, V, U_2, \ldots, U_t]$ using S_t and $Q_1, \ldots, Q_{t-1}$, set $f_0 = \hat{e}(U_0, S_t), f_i = \hat{e}(Q_{i-1}, U_i)$ for $2 \le i \le t$ and output $M = V \oplus H_2(f_0 / (f_2 \cdot f_3 \cdots f_t))$. To see the correctness of the decryption, notice that:

$$f_0 = \hat{e}(U_0, S_t) = \hat{e}(rP_0, \sum_{i=1}^{t} s_{i-1}P_i) = \prod_{i=1}^{t} \hat{e}(rP_0, s_{i-1}P_i)$$

HIBE – Boneh, Boyen and Goh

- Root Setup: Given a security parameter 1^λ and a system parameter ℓ (this indicates the maximum depth of HIBE), Root Setup generates two cyclic groups $\mathbf{G}_1$, $\mathbf{G}_2$ of a large prime order q and defines a bilinear map $\hat{e}: \mathbf{G}_1 \times \mathbf{G}_1 \to \mathbf{G}_2$. For randomly chosen $P \in \mathbf{G}_1$ of order q and $s \in \mathbf{Z}_q$, outputs params = $(\mathbf{G}_1, \mathbf{G}_2, \hat{e}, P, P_1, P_2, P_3, Q_1, \ldots, Q_\ell)$, master key = sP_2. Here $P_1 = sP$ and $P_2, P_3, Q_1, \ldots, Q_\ell$ are randomly chosen elements in $\mathbf{G}_1$.

- Lower-level Setup and Extract: To generate a private key d_{ID} for an identity ID = $(I_1, \ldots, I_k) \in (\mathbf{Z}_q^*)^k$ of depth $k \le \ell$, using the master secret, pick a random $r \in \mathbf{Z}_q$ and output $d_{\mathsf{ID}} = (sP_2 + r \cdot (I_1Q_1 + \ldots + I_kQ_k + P_3), rP, rQ_{k+1}, \ldots, rQ_\ell) \in \mathbf{G}_1^{2+\ell-k}$. Note that d_{ID} becomes shorter as the depth of ID increases. The private key for ID can be generated incrementally, given a private key for the parent identity $\mathsf{ID}_{k-1} = (I_1, \ldots, I_{k-1}) \in (\mathbf{Z}_q^*)^{k-1}$, as required. Indeed, let $d_{\mathsf{ID}|k-1} = (sP_2 + r' \cdot (I_1Q_1 + \ldots + I_{k-1}Q_{k-1}), r'P, r'Q_k, \ldots, r'Q_\ell) = (a_0, a_1, b_k, \ldots, b_\ell)$ be the private key for ID_{k-1}. To generate d_{ID}, pick a random $t \in \mathbf{Z}_q$ and output $d_{\mathsf{ID}} = (a_0 + I_kb_k + t \cdot (I_1Q_1 + \ldots + I_kQ_k + P_3), a_1 + tP, b_{k+1} + tQ_{k+1}, \ldots, b_\ell + tQ_\ell)$. This private key is a properly distributed private key for ID = $(I_1, \ldots, I_k)$ for $r = r' + t \in \mathbf{Z}_q$.

- Encrypt: To encrypt a message $M \in \mathbf{G}_2$ under the public key ID = $(I_1, \ldots, I_k) \in (\mathbf{Z}_q^*)^k$, pick a random number $v \in \mathbf{Z}_q$ and output C = $(\hat{e}(P_1, P_2)^v \cdot M, vP, v(I_1Q_1 + \ldots + I_kQ_k + P_3)) \in \mathbf{G}_2 \times \mathbf{G}_1^2$.

- Decrypt: Consider an identity ID = $(I_1, \ldots, I_k)$. To decrypt a given ciphertext C = (A,B,C) using the private key $d_{\mathsf{ID}} = (a_0, a_1, b_{k+1}, \ldots, b_\ell)$, output $A \cdot \hat{e}(a_1, C) / \hat{e}(B, a_0) = M$. Indeed, for a valid ciphertext, we have

$$\frac{\hat{e}(a_1, C)}{\hat{e}(B, a_0)} = \frac{\hat{e}(rP, v(I_1Q_1 + \ldots + I_kQ_k + P_3))}{\hat{e}(vP, sP_2 + r \cdot (I_1Q_1 + \ldots + I_kQ_k + P_3))}$$

$$= \frac{1}{\hat{e}(P, P_2)^{vs}} = \frac{1}{\hat{e}(P_1, P_2)^v}$$

2. Punctured interval (PI) scheme

The punctured interval scheme is a broadcast encryption scheme using the subset cover method. Main difference between the PI and other broadcast encryptions using subset cover method is that the PI uses linear structure (i.e. all users are located on a straight line) unlike CS, SD, LSD using tree structure. In PI scheme, subset covering non-revoked users is the p-punctured c-interval which is a set of at most c consecutive users containing at most p revoked users, where $p \ge 0$ and $c > 0$. The first user and the last user of one punctured interval must be non-revoked users. The p-punctured c-interval starting from u_i and ending at u_j with q revoked users $u_{x_1}, \ldots, u_{x_q}$ is denoted by $P_{i,j;x_1,\ldots,x_q}$. Each p-punctured c-interval has one interval key (in fact, $K_{i,j;x_1,\ldots,x_q}$ is the interval key corresponding to $P_{i,j;x_1,\ldots,x_q}$), and every user in the punctured interval can obtain the corresponding interval key using his/her own user key. User keys are generated and distributed by the following method:

Key Generation and Distribution

Let $h_t : \{0,1\}^\ell \to \{0,1\}^\ell$ be one-way permutations for $t=0,1, \ldots, p$ where ℓ denotes the length of encryption key. In order to assign one key to each p-punctured c-interval, we should randomly choose N keys $K_{1,1}, K_{2,2}, \ldots, K_{N,N}$ to the corresponding users $u_1, \ldots, u_N$. 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,

- 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 key 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 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 $i < x_1 < x_2 < \ldots < x_t < k$ with $0 \le t \le p$ and $2 \le k-i+1 \le c$

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, under the rule described above. Let $P = P_{i,j;x_1,\ldots,x_q}$ be one of P_μ's (μ=1, …,m). 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 μ = 1, 2, …, m, just for a matter of convenience. Then the center broadcasts:

$$\langle info_1, info_2, \ldots, info_m; E_{K_1}(SK), E_{K_2}(SK), \ldots, E_{K_m}(SK); E_{SK}(M) \rangle$$

where $info_\mu$ is information of P_μ, the starting point u_{i_μ}, the end point u_{j_μ} and q_μ revoked users.

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 $i \le k \le j$ and $k \neq x_1, \ldots, x_q$. 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 the above process, 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.

III. Our Scheme (Pub-PI)

For simplicity of explanation, we will discuss only basic c-intervals (which is a set of at most c consecutive users containing no revoked user) and denote the c-interval starting from u_i and ending u_j by $P_{i,j}$.

- Root Setup: Given a security parameter 1^λ, Root Setup generates two cyclic groups $\mathbf{G}_1$, $\mathbf{G}_2$ of a large prime order q and defines a bilinear map $\hat{e}: \mathbf{G}_1 \times \mathbf{G}_1 \to \mathbf{G}_2$. Namely, for any $P, Q \in \mathbf{G}_1$ and $a, b \in \mathbf{Z}_q$, $\hat{e}(aP, bQ) = \hat{e}(bP, aQ) = \hat{e}(P,Q)^{ab}$. Assume that R_0 be a randomly chosen generator of $\mathbf{G}_1$. Place N users on a straight line L and index those users by integers in $[1,N]$ so that the numbering has increasing order. For example, the left most user is indexed by 1 and the right most user is indexed by N. And the unique identity ID_i is given to each user u_i. For randomly chosen $s_0 \in \mathbf{Z}_q$, Q_0 is defined by $s_0 R_0$.

  The output of Root Setup is params = $(\mathbf{G}_1, \mathbf{G}_2, \hat{e}, R_0, Q_0, H_1, H_2, L, c)$, master key = s_0. Here $H_1 : \{0,1\}^* \to \mathbf{G}_1$ and $H_2 : \mathbf{G}_2 \to \{0,1\}^n$ are cryptographic hash functions, where n is the length of the encrypted message and c is a system parameter. In fact, c is a constant which represents the maximum length of key chains.

- Key Generation: For each $i (1 \le i \le N)$, the key chain starting at i-th user with length c is generated by the following method: The first key in the key chain, that is, the key corresponding to $P_{i,i}$ is a pair $(K_{i,i} = s_0 R_{i,i} + s_{i,i} R_{i,i}, Q_{i,i} = s_{i,i} R_0)$, where $R_{i,i} = H_1(\mathsf{ID}_i)$ and $s_{i,i} \in \mathbf{Z}_q$ is randomly chosen. The next key is $(K_{i,i+1} = s_0 R_{i,i} + s_{i,i+1} R_{i,i+1}, Q_{i,i+1} = s_{i,i+1} R_0)$, for $R_{i,i+1} = H_1(\mathsf{ID}_i, \mathsf{ID}_{i+1})$ and randomly chosen $s_{i,i+1} \in \mathbf{Z}_q$. Generally, the j-th key in the key chain is $(K_{i,j} = s_0 R_{i,i} + s_{i,j} R_{i,j}, Q_{i,j} = s_{i,j} R_0)$, where $R_{i,j} = H_1(\mathsf{ID}_i, \ldots, \mathsf{ID}_j)$ and $s_{i,j} \in \mathbf{Z}_q$ is randomly chosen. At last, each user u_t receives all possible keys $(K_{i,k}, Q_{i,k})$ for $i \le t \le k \le i+c$.

- Encrypt: For each session, the sender divides L into disjoint c-intervals $P_1, \ldots, P_m$, whose union covers all the non-revoked users. The broadcasting header corresponding to $P_\mu = P_{i,j} (1 \le \mu \le m)$ is: $\mathsf{HDR}_\mu = [SK \oplus H_2(g_\mu), r_\mu R_0, r_\mu R_{i,j}]$, where $g_\mu = \hat{e}(Q_{i,i}, r_\mu R_{i,i}) = \hat{e}(R_0, R_{i,i})^{r_\mu s_0}$ for randomly chosen r_μ.

- Decrypt: Note that every user contained in $P_{i,j}$ knows $(K_{i,j}, Q_{i,j})$. With the encrypted message $\mathsf{HDR}_\mu = [SK \oplus H_2(g_\mu), r_\mu R_0, r_\mu R_{i,j}] = [V, U_1, U_2]$, each user in $P_{i,j}$ can decrypt the session key SK as follows: $SK = V \oplus H_2(g_\mu) = V \oplus H_2(\hat{e}(U_1, K_{i,j}) / \hat{e}(Q_{i,j}, U_2))$.

  Remark that

$$\frac{\hat{e}(U_1, K_{i,j})}{\hat{e}(Q_{i,j}, U_2)} = \frac{\hat{e}(r_\mu R_0, s_0 R_{i,i} + s_{i,j} R_{i,j})}{\hat{e}(s_{i,j} R_0, r_\mu R_{i,j})} = \frac{\hat{e}(r_\mu R_0, s_0 R_{i,i}) \cdot \hat{e}(r_\mu R_0, s_{i,j} R_{i,j})}{\hat{e}(s_{i,j} R_0, r_\mu R_{i,j})} = \hat{e}(R_0, R_{i,i})^{r_\mu s_0}$$

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, under the rule described above. Let $P = P_{i,j;x_1,\ldots,x_q}$ be one of P_μ's (μ=1, ...,m). 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 μ = 1, 2, ..., m, just for a matter of convenience. Then the center broadcasts:

$$\langle info_1, info_2, \ldots, info_m; E_{K_1}(SK), E_{K_2}(SK), \ldots, E_{K_m}(SK); E_{SK}(M) \rangle$$

where $info_\mu$ is information of P_μ, the starting point u_{i_μ}, the end point u_{j_μ} and q_μ revoked users.

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 $i \le k \le j$ and $k \neq x_1, \ldots, x_q$. 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 the above process, 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.

III. Our Scheme (Pub-PI)

For simplicity of explanation, we will discuss only basic c-intervals (which is a set of at most c consecutive users containing no revoked user) and denote the c-interval starting from u_i and ending u_j by $P_{i,j}$.

- Root Setup: Given a security parameter 1^λ, Root Setup generates two cyclic groups $\mathbf{G}_1$, $\mathbf{G}_2$ of a large prime order q and defines a bilinear map $\hat{e}: \mathbf{G}_1 \times \mathbf{G}_1 \to \mathbf{G}_2$. Namely, for any P, $Q \in \mathbf{G}_1$ and $a, b \in \mathbf{Z}_q$, $\hat{e}(aP, bQ) = \hat{e}(bP, aQ) = \hat{e}(P,Q)^{ab}$. Assume that R_0 be a randomly chosen generator of $\mathbf{G}_1$. Place N users on a straight line L and index those users by integers in [1,N] so that the numbering has increasing order. For example, the left most user is indexed by 1 and the right most user is indexed by N. And the unique identity ID_i is given to each user u_i. For randomly chosen $s_0 \in \mathbf{Z}_q$, Q_0 is defined by $s_0 R_0$.

  The output of Root Setup is params = $(\mathbf{G}_1, \mathbf{G}_2, \hat{e}, R_0, Q_0, H_1, H_2, L, c)$, master key = s_0. Here $H_1 : \{0,1\}^* \to \mathbf{G}_1$ and $H_2 : \mathbf{G}_2 \to \{0,1\}^n$ are cryptographic hash functions, where n is the length of the encrypted message and c is a system parameter. In fact, c is a constant which represents the maximum length of key chains.

- Key Generation: For each $i(1 \le i \le N)$, the key chain starting at i-th user with length c is generated by the following method: The first key in the key chain, that is, the key corresponding to $P_{i,i}$ is a pair $(K_{i,i} = s_0 R_{i,i} + s_{i,i} R_{i,i}, Q_{i,i} = s_{i,i} R_0)$, where $R_{i,i} = H_1(\mathsf{ID}_i)$ and $s_{i,i} \in \mathbf{Z}_q$ is randomly chosen. The next key is $(K_{i,i+1} = s_0 R_{i,i} + s_{i,i+1} R_{i,i+1}, Q_{i,i+1} = s_{i,i+1} R_0)$, for $R_{i,i+1} = H_1(\mathsf{ID}_i, \mathsf{ID}_{i+1})$ and randomly chosen $s_{i,i+1} \in \mathbf{Z}_q$. Generally, the j-th key in the key chain is $(K_{i,j} = s_0 R_{i,i} + s_{i,j} R_{i,j}, Q_{i,j} = s_{i,j} R_0)$, where $R_{i,j} = H_1(\mathsf{ID}_i, \ldots, \mathsf{ID}_j)$ and $s_{i,j} \in \mathbf{Z}_q$ is randomly chosen. At last, each user u_t receives all possible keys $(K_{i,k}, Q_{i,k})$ for $i \le t \le k \le i+c$.

- Encrypt: For each session, the sender divides L into disjoint c-intervals $P_1, \ldots, P_m$, whose union covers all the non-revoked users. The broadcasting header corresponding to $P_\mu = P_{i,j} (1 \le \mu \le m)$ is: $\mathsf{HDR}_\mu = [SK \oplus H_2(g_\mu), r_\mu R_0, r_\mu R_{i,j}]$, where $g_\mu = \hat{e}(Q_{i,i}, r_\mu R_{i,i}) = \hat{e}(R_0, R_{i,i})^{r_\mu s_0}$ for randomly chosen r_μ.

- Decrypt: Note that every user contained in $P_{i,j}$ knows $(K_{i,j}, Q_{i,j})$. With the encrypted message $\mathsf{HDR}_\mu = [SK \oplus H_2(g_\mu), r_\mu R_0, r_\mu R_{i,j}] = [V, U_1, U_2]$, each user in $P_{i,j}$ can decrypt the session key SK as follows: $SK = V \oplus H_2(g_\mu) = V \oplus H_2(\hat{e}(U_1, K_{i,j}) / \hat{e}(Q_{i,j}, U_2))$.

  Remark that

$$\frac{\hat{e}(U_1, K_{i,j})}{\hat{e}(Q_{i,j}, U_2)} = \frac{\hat{e}(r_\mu R_0, s_0 R_{i,i} + s_{i,j} R_{i,j})}{\hat{e}(s_{i,j} R_0, r_\mu R_{i,j})} = \frac{\hat{e}(r_\mu R_0, s_0 R_{i,i}) \cdot \hat{e}(r_\mu R_0, s_{i,j} R_{i,j})}{\hat{e}(s_{i,j} R_0, r_\mu R_{i,j})} = \hat{e}(R_0, R_{i,i})^{r_\mu s_0}$$

for small r and $TO_{(1\text{-punctured})} \approx 3r/2$ as r grows. The storage size becomes $SS_{(1\text{-punctured})} = O(c^3)$ and the computation cost is still 2 *pairing computations*.

2. Comparison

Previous work [5] Dodis and Fazio first proposed the method to extend a broadcast encryption system with related key structure such as the SD/LSD to a public key broadcast encryption in 2002. The efficiency of public key extension using their method is directly affected by the efficiency of HIBE since the method is the simple combination of BE and HIBE.

So in the case to use HIBE of Gentry and Silverberg in order to obtain the public key SD, the transmission overhead , the storage size and computation cost are $(2r\text{-}1)\log N$, $O(\log^3 N)$ and 2 pairing computations, respectively. It is because that the ciphertext for one subset among $2r$-1 subsets in SD has at most $\log N$ length and the number of secret keys stored by each user is $\log N$ times of number of user keys in SD. The pubic key size is $O(1)$. In the case to use HIBE of Boneh *et al.*, the transmission overhead is $3(2r\text{-}1)$. The storage size and the computation cost is same to the above case. However the public key size increases to $O(\log N)$.

In HIBE, each user in has different secret keys from other users and users with different keys can decrypt the same ciphertext. But this property of HIBE is not a requirement of broadcast encryption. Because of this point, our extension method uses the modified HIBE removing this property to improve the efficiency of resulting public key BE system. As a result, the transmission overhead, the storage size, the computation cost and the public key size of Pub-PI are $3\cdot\left(r+\lceil (N-2r)/c \rceil\right)$, $c(c+1)$, 2 *pairing computations* and $O(1)$, respectively.

Public Key Extension of PI by [5] Since PI scheme is much better than SD scheme in transmission overhead, we can easily guess that if the method in [5] can be applied to PI, the public key PI using [5] might have more efficient transmission overhead than the known public key SD. But so far there is no result applied the method to PI. In this subsection we explain public key extensions of PI using the method of Dodis and Fazio. In fact, these public key PI scheme don't have better efficiency than Pub-PI.

First we can define the hierarchical identifier HID assigning to each basic c-interval $P_{i,j}$ as follows: $\mathsf{HID}(P_{i,j})=(\mathsf{ID}_i, \mathsf{ID}_j)$, where ID_i is the identity of i-th user. In initialization step, the center runs the Setup algorithm of a HIBE and publishes param. Then it makes public keys and corresponding secret keys for each interval $P_{i,j}$ and sends the secret keys to each user in secure channel.

The key $K_{i,j}$ relative to a given interval $P_{i,j}$ is extracted using the following method:

$$K_{i,i} \leftarrow \mathsf{Extract}(\mathsf{param}, \mathsf{HID}(P_{i,i}), \mathsf{master\text{-}key})$$
$$K_{i,i+1} \leftarrow \mathsf{Extract}(\mathsf{param}, \mathsf{HID}(P_{i,i+1}), K_{i,i})$$
$$\vdots$$
$$K_{i,j} \leftarrow \mathsf{Extract}(\mathsf{param}, \mathsf{HID}(P_{i,j}), K_{i,j-1})$$

For $i \le t \le j$, each key $K_{i,t}$ generated by above method is assigned to user u_t as the secret key of u_t. Note that u_t already known $K_{i,t}$ also knows the next user's key $K_{i,t+1}$, but cannot obtain other keys $K_{i,k}(t+2 \le k \le j)$. Although u_t cannot know the encrypted key $K_{i,j}$ used in order to transmit a message to users in $P_{i,j}$, he/she can decrypt the message by HIBE's property. The following two concrete instantiations are obtained by HIBE schemes of [2] and [7].

PI using HIBE of Gentry and Silverberg [7]: In HIBE of Gentry and Silverberg, each user u_t in level t keeps $(t+1)$ secret keys and the ciphertext for u_t has the length $t+2$. When this HIBE is applied to the PI using c-interval, the number of secret keys of each user is maximum c times of the number of user keys in the PI, since for each chain in the PI all users located in the chain should store maximum c secret keys. Hence the storage size is $O(c^2)$. And the length of the ciphertext for one interval is at most $c+2$ because a message is encrypted by the secret key of last user in the c-interval. So total transmission overhead is about $c(\ r+\lceil (N-2r)/c \rceil\ )$ because the number of intervals is $r+\lceil (N-2r)/c \rceil$. The computation cost is at most c *pairing computations* because the decryption process for each user is exactly same to the decryption in HIBE. The public key size is still $O(1)$ as the HIBE.

PI using HIBE of Boneh *et al.*[2]: In HIBE of Boneh *et al.*, each user in level t stores ℓ -t+2 secret keys where ℓ is maximum depth in the HIBE. And the ciphertext contains 3 elements and decryption takes 2 *pairings computations* regardless of level. So in public key PI using this HIBE, the storage size is $O(c^2)$ and the transmission overhead is $3(\ r+\lceil (N-2r)/c \rceil\ )$ and the computation cost is 2 *pairing computations*. However, the public key size increases to $O(r)$.

V. Conclusion

In order to design public key broadcast encryption schemes, it should consider methods that convert broadcast encryption scheme based on the symmetric key cryptosystem to the public key system. Until now, several trials extending of symmetric key setting broadcast encryption to public key setting were executed by directly applying HIBE to broadcast encryption system. However, this extension may be the best way, since some properties of HIBE are not proper to public key broadcast schemes. In this paper, we proposed an efficient public key

broadcast encryption scheme Pub-PI which is obtained by adapting the concept of HIBE, but removing unnecessary properties of HIBE we can reduce the transmission overhead further. The transmission overhead of the Pub-PI is smaller than that of the public key PI obtained by the method of Dodis and Fazio as well as the public key SD/LSD scheme. In fact, the transmission overhead of the Pub-PI is approximately $3r$, where r is the number of revoked users. The storage size of Pub-PI is $O(c^2)$, where c is a system parameter of PI and the computation cost is 2 *pairing computations*. This result is obtained using only basic c-intervals in PI scheme. Actually, the method for designing Pub-PI can be applied to any punctured intervals. Using punctured intervals, public key PI scheme with the reduced transmission overhead will be realized at the cost of the storage size.

References

[1] Berkovits, S.: 'How to broadcast a secret', In Advances in Cryptology – Eurocrypt'91, LNCS vol. 547, 1991, pp. 536-541.

[2] Boneh, D., Boyen, X., and Goh, E.: 'Hierarchical identity based encryption with constant size ciphertext', In Advances in Cryptology – Eurocrypt'05, LNCS vol. 3494, 2005, pp. 440-456.

[3] Boneh, D., and Franklin, M.: 'Identity-based encryption from the Weil pairing', In Advances in Cryptology – Crypto'01, LNCS vol. 2139, 2001, pp. 213-229.

[4] Boneh, D., Gentry, C., and Waters, B.: 'Collusion resistant broadcast encryption with short ciphertexts and private keys'. In Advances in Cryptology – Crypto'05, LNCS vol. 3621, 2005, pp. 258-275.

[5] Dodis, Y., and Fazio, N.: 'Public key broadcast encryption for stateless receivers'. Proc. of the Digital Right Management Workshop'02, LNCS vol. 2696, 2002, pp. 61-80.

[6] Fiat, A., and Naor, M.: 'Broadcast encryption', In Advances in Cryptology – Crypto'93, LNCS vol. 773, 1993, pp. 480-491.

[7] Gentry, C., and Silverberg, A.: 'Hierarchical ID-based cryptography', In Advances in Cryptology – Asiacrypt'02, LNCS vol. 2501, 2002, pp. 548-566.

[8] Halevi, D., and Shamir, A.: 'The LSD broadcast encryption scheme', In Advances in Cryptology – Crypto'02, LNCS vol. 2442, 2002, pp. 47-60.

[9] Jho, N.-S., Cheon, J.H., Kim, M.-H., and Yoo, E.S.: 'Broadcast encryption π', http://eprint.iacr.org/2005/073, 2005.

[10] Jho, N.-S., Hwang, J.Y., Cheon, J.H., Kim, M.-H., Lee, D.H., and Yoo., E.S.: 'One-way chain based broadcast encryption schemes', In Advances in Cryptology – Eurocrypt'05, LNCS vol. 3494, 2005, pp. 559-574.

[11] Naor, D., Naor, M., and Lotspiech, J.: 'Revocation and tracing schemes for stateless receivers', In Advances in Cryptology – Crypto'01, LNCS vol. 2139, 2001, pp. 41-62.

[12] Naor, M., and Pinkas, B.: 'Efficient trace and revoke schemes', Proc. of Financial cryptography'00, LNCS vol. 1962, 2000, pp. 1-20.

Authors

Nam-Su Jho received the B.S. degree in mathematics from Korea Advanced Institute of Science and Technology, Daejeon, Korea, in 1999 and the Ph.D. degree in mathematics from Seoul National University, Seoul, Korea, in 2007.
Since 2007, he has been with the Electronics and Telecommunications Research Institute as a senior member of engineering staff. His research interests include cryptography and information theory

Eun Sun Yoo received the B.S. degree in mathematics education at Korea University in 2001 and Ph.D. degree in mathematics at Seoul National University in 2007.
Currently, she is with Security Consulting Business Team at Samsung SDS where she joined from 2007. Her research interests include cryptography and information theory.

Man Young Rhee is an Endowed Chair Professor at Kyung Hee University and holds over 45 years of research and teaching experience in the field of communication technologies, coding theory, cryptography, and information security. His career in academia includes professorships at Hanyang University (also held the position of Vice President at the university), Virginia Tech, Seoul National University, and the University of Tokyo. Dr. Rhee has held a number of high level positions in both government and corporate sectors: President of Samsung Semiconductor Communications, President of Korea Telecommunications Company, Chairman of the Korea Information Security Agency at the Ministry of Information and Communication, President of the Korea Institute of Information Security & Cryptology, and Vice President of the Agency for Defense Development at the Ministry of National Defense. He is a Member of the National Academy of Sciences, Senior Fellow at the Korea Academy of Science and Technology, and an Honorary Member of the National Academy of Engineering of Korea. His awards include the "Dongbaek" Order of National Service Merit and the "Mugunghwa" Order of National Service Merit, the highest grade honor for a scientist in Korea, NAS Prize, the National Academy of Sciences, NAEK Grand Prize, the National Academy of Engineering of Korea and Information Security Grand Prize, KIISC. He published five books with John Wiley, McGraw-Hill and Prentice Hall. Dr. Rhee has a B.S. in Electrical Engineering from Seoul National University, and an M.S. in Electrical Engineering and a Ph.D. from the University of Colorado.