Broadcast Encryption Using Efficient Key Distribution and Renewal for Ubiquitous Environments

Deok-Gyu Lee, Jang-Su Park, and Im-Yeong Lee

Division of Information Technology Engineering, Soonchunhyang University,
# 646, Eupnae-ri, Shinchang-myun, Asan-si, Choongchungnam-do, Korea
{hbrhcdbr, pjswise, imylee}@sch.ac.kr
http://sec-cse.sch.ac.kr

Abstract. The method of broadcast encryption has been applied to the transmission of digital information such as multimedia, software, and paid TV on the open network. In this broadcast encryption method, only previously au-thorized users can gain access to digital in-formation. When broadcast message is transmitted, authorized users can first decode the session key using the previously given private key and get digital information using this ses-sion key. This way, users retrieve a message or a session key using the key transmitted by broadcasters. For their part, broadcasters need to generate and distribute keys. Broadcast-ers should also carry out efficient key renewal when users subscribe or unsubscribe. This paper introduces how to generate and distribute key efficiently and how key renewal works. The proposal uses two methods: (1) the server generates keys without the consent of us-ers by anticipating users, and; (2) the server and users generate keys by mutual agreement. The advantage of the two proposed methods is that the receiver can decode broadcast message using a secret key. Even if the key is re-newed later, the user can efficiently renew using only a single set of information.

1 Introduction

The broadcast encryption method has been recently applied to the transmission of digital information such as multimedia, software, pay TV, etc. As one of the key pro-viding methods, the public key method uses a single group key to encode the session key and an infinite number of keys for decoding. As such, the server encodes the session key and enables each user to decode it using different keys. In the broadcast encryption method, only previously authorized users can gain access to digital information. When broadcast message is transmitted, authorized users can first decode the session key using the previously given private key and get digital information using this session key. In short, broadcast encryption involves generating, distributing, and renewing keys. This paper introduces the method of generating, distributing, and renewing keys efficiently. The proposal uses 2 methods: (1) the server generates keys without the consent of users by anticipating users, and; (2) the server and users generate keys by mutual agreement. The advent- age of the two proposed schemes is that the receiver can decode broadcast message using a secret key. Even if the key is renewed later, the user can efficiently renew using only a

X. Zhou et al. (Eds.): EUC Workshops 2006, LNCS 4097, pp. 721 – 730, 2006.

single set of information. In the proposed methods, key renewal factor is added for fast key renewal. This allows easy key renewal and provides users with renewal values even in case of new subscription or withdrawal. This paper briefly introduces application methods in broadcast encryption, goes through the existing methods, and discusses each stage of the proposed methods. Like-wise, the protocols of each stage are explained. Proposed methods are also re-viewed through comparison analysis between the existing methods and the proposed methods. Finally, the conclusion is presented.

2 Overview of Broadcast Encryption

Broadcast encryption is one of protocols to communicate between one sender and a lot of receivers. A sender who has one encoded key can broadcast the encoded message. Only privileged receivers can get a message with a key to decode the message. Broadcast encryption can be applied for many kinds of scenarios where content providers send out many kinds of encoded information and only the privileged users can decode such information. One example can be Pay-TV which has provided many kinds of broadcast encryption technique. The first suggested method for broad-cast encryption presented by Cho and two other colleagues is composed into three steps.

Initiation by a content provider: A content provider will generate necessary information for all users, which step is called as initial information.

User information initiation: This is a step for a user to register his or her personal information to the content provider. After this step, the user's information will be stored as his or her personal key. The content provider will renew the user's initial information after the user initiation step.

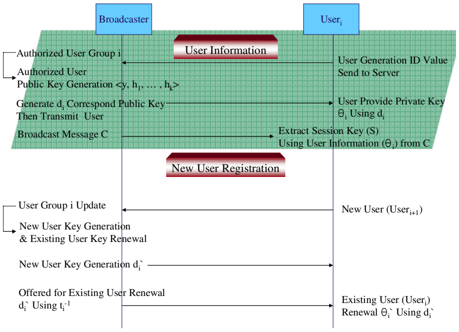

Fig. 1. Application Method 1

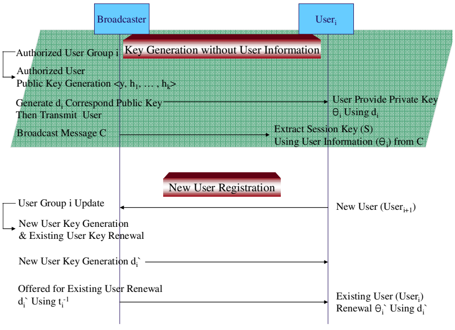

Fig. 2. Application Method 2

Session Transmission: Content data will be encoded as session keys. These session keys will be transmitted by divided into small parts called session. Each session will be transmitted as an encoded form into different session part. Privileged users will get session keys to get real data by decoding part of session keys using his or her personal keys.

2.1 Application Methods

Broadcast encryption is based on two models. Although there are some differences between the applied models, each of them will be discussed. To begin with, the first model is shown in the figure below:

This method involves generating/distributing keys using information between the user and server. This is similar to the existing multicast method, since the message pro-vided is determined by the previous user group. The only difference lies in the transmitting method. The user participation time may be included in the key generating time, since it requires user participation in the process of key generation. Unlike the abovementioned method, the server in the second applied model generates keys. The server generates keys by anticipating user participation at its own discretion. This method enables quick creation and renewal since the server generates all users' keys without their consent. In case the server becomes the target of attacks or other vicious purposes, however, it becomes very vulnerable.

3 Conventional Scheme – Narayanan

3.1 Protocol of the Narayanan Scheme

Assume one contents provider broadcasting in m number of channels and n number of users. Protocol is divided into seven algorithms such as Setup, AddStream, AddUser, Broadcast, Receive, Subscribe, and Unsubscribe. Whether or not users receive channels can be displayed with Subsc and a $m \times n$ matrix. If userU_j is registered at S_i, the value of $Subsc[i, j]$ is 1.Otherwise, if the user is not registered, the value is 0.

Setup: The contents provider generates the following variables: When $N = pq, R, d_r \leq R\{1,2,\ldots,\varphi(N)\}$, $1 \leq r \leq 4+t$. P and q are larger prime numbers, and R is a random value. p , q , and d are composed as secret keys of the contents provider. In turn, the contents provider opens the public key (N).

AddStream: The contents provider randomly choose $g_i \in Z_{N^*}$ to add new channel stream S_i to the system and sets up $Subsc[i, j]$ to set all j to 0; thus preventing the opening of the g_i value.

AddUser: The contents provider chooses $(e_{1j}, e_{2j}, \ldots, e_{(t+4)j})$, which satisfies$\sum_{r=1}^{t+4} e_{rj} d_r = R\Phi(N) + 1$. At this time, U_j receives the decoding device (Set-Top Terminal) that stored the secret key in the safe memory. The secret key of U_j will be$(e_{1j}, e_{2j}, \ldots, e_{(t+4)j})$.

Subscribe: When user U_j subscribes to service S_i, the contents provider transmits g_i^{e1j} to U_j and changes the $Subsc\ [i,\ j]$ value to 1.

Unsubscribe: When user U_j unsubscribe to S_i , the contents provider sets $Subsc\ [i,\ j] = 0$. Similar to the AddStream algorithm, the contents provider chooses a new g_i value and transmits g_i^{e1j} to all users who have the value $Subsc\ [i,\ j] = 1$.

***Broadcast*:** To transmit message M to channel stream S_i, the contents provider randomly chooses value x as a value smaller than Φ (N) and transmits encrypted data $C = (x, C_1, C_2, \ldots, C_{t+4})$ as $C_1 = M^{d1} g_i^x, C_2 = M^{d2}, C_{t+4} = M^{dt+4}$.

Receive: User U_j determines $\left(\prod_{r=1}^{t+4} C_r^{e_{rj}}\right) / g_i^{xe_{1j}}$ using secret key $(e_{1j}, e_{2j}, \ldots, e_{(t+4)j})$ to decode encrypted data $C = (x, C_1, C_2, \ldots, C_{t+4})$, which is transmitted to channel stream S_i. User U_j restores contents data M by going through this process.

$$\left(\prod_{r=1}^{t+4} C_r^{e_{rj}}\right) / g_i^{xe_{1j}} = M^{R\Phi(N)+1} = M$$

Problems of the Narayanan scheme

The Narayanan scheme requires the traffic of $(x, C_1, C_2, \ldots, C_{t+4})$ per channel. Since traffic is related to the number of channels, increasing number of channels can also cause heavier traffic. In addition, despite managing to find traitor U_j, the contents provider has to distribute a new secret key to all subscribers again except U_j to disqualify U_j.

4 Proposed Scheme

4.1 Overview of Proposed Schemes

This section presents an overview of the proposed methods. This proposed scheme is a classification of scenarios that can occur using the proposed methods. The scenario is composed of the basic flow, renewal flow, new process flow, leaving flow, and flow of false user anticipation. The proposal can be classified into three large parts depending on the scenario: key generation and distribution, broadcast message generation, and key renewal. Similarly, two proposed methods can be applied to the entire flow. Differences are only found in the initial key generation and distribution part through server anticipation and users; the rest proceeds in the same manner.

In addition, the first method in the proposal has the following features: (1) the user's private key is generated by the server; (2) persons other than the user cannot decode the broadcasting message, and; (3) renewing keys is easy, which is important when new subscribers subscribe and existing users unsubscribe. On the other hand, in the second method, the user's private key is generated only upon obtaining the user's consent. When many users gather, the server generates a public key. Through the public key, the encrypted broadcasting message is transmitted. Likewise, subscribing and unsubscribing can take place easily by deleting the information provided by the user.

4.2 System Coefficient

The following is a description of the system coefficient used in this method:

p : Prime number ($\geq 512bit$) q : Prime number ($\geq 160bit(q|p-1)$)

n : Prime number ($n = pq$) l : Number for Personal Key Generation

e : Public Encryption Key M : Message

S : Session Key $d_1, d_2, \cdots, d_k$: List of Personal Decryption Key

$d_i = \theta_i \cdot \gamma_i : (\gamma_i \in \Gamma)$ k : Number of Prediction User

$\Gamma = \gamma_1, \ldots, \gamma_k (\Gamma \in Z_q)$ Θ_i, U_i : user Registered by User Information

i : User $(i = 1, \ldots, k)$ j : Withdrawal user

z : User in Broadcast Group α_i : Random Number $(\alpha_i \in Z_q)\{\alpha_1, \ldots, \alpha_k\}$

b : Server's generated public information

r_i : Set of Random Number ($r_i \in Z_p$): $(r_1, r_2, \cdots, r_k)$

$h_i = g^{\gamma_i} \bmod n$ a : Random Element $(a \in Z_q)$

C : Broadcast message $\langle y, h_1, h_2, \cdots, h_k \rangle$: Public Key:

$y = \prod_{i=1}^{k} h_i^{\alpha_i} \bmod n$ $C = \langle M(orS) y^{aT}, h_1^{ar_1}, \cdots, h_k^{ar_k} \rangle = \langle B, H_1, \cdots, H_k \rangle$

$B = M(orS) y^{aT} \bmod n$ $H_i = h_i^{\alpha} \bmod n$

t_i : Element for Key Renewal $(t_1, \ldots, t_k \in Z_q)$ $T = t_1 \cdot \cdots \cdot t_k (\bmod n)$

o : Security Parameter CT : Encryption Message by Session Key $CT = E_S(M)$

ζ_i : User is random choose value Ξ_i : Stored User of ID_i

4.3 Key Distribution Scheme on Server Prediction

The first proposed scheme the server predicts the user and key generates and it is a scheme which after it distributes. Information of the user is not provided in the first scheme from the application model not to be, it is a scheme where the server generates. From the dissertation which it sees from in the scheme which it proposes focus the loach petty it will more be easy in key renewal and the possibility efficiently becoming the renewal in order to be, it proposed in the form which inserts a key renewal element. The key renewal element which is inserted it led and to eliminate only the renewal element of the case user where in order for the secession occurs after with whole to become the efficient key renewal.

4.3.1 Key Generation and Distribution Stage

Key generation is processed by the server. The generation and transmission of the private and public keys will go through the following process:

Step 1. The server anticipates users($i = 1, 2, \cdots, k$) and randomly chooses string(γ_i) accordingly. Based on this chosen string, the server generates the values

$h_i = g^{r_i} \bmod q(i=1,\ldots,k)$ required to produce the public key and key renewal element t_i and generated for renewal.

$$\langle y, h_1, h_2, \cdots, h_k \rangle,\ T = t_1 \cdot \ldots \cdot t_k (\bmod n)$$

Step 2. The server produces θ_i using the public key and t_i for renewal. The server transmits the generated private key $d_i = \theta_i \cdot \gamma_i \cdot t_z (\bmod n)$ to user and include element t_z of user z .

$$\theta_i = \left(\prod_{i=1}^{k} \gamma_i a_i t_i\right) \Big/ \left(\prod_{i=1}^{k} \gamma_i r_i\right) \bmod n$$

Step 3. The user acquires θ_i from the received d_i . The d_i personal key θ_i as different expression the Γ is opened to the public but the θ_i it could be maintained with secret.

$$d_i / \gamma_i = (\theta_i \cdot \gamma_i \cdot t_z) / \gamma_i (\bmod n),\ \theta_i' = \left(\prod_{i=1}^{k} (\alpha_i \cdot t_i) \Big/ r_i\right) \cdot t_z (\bmod n)$$

4.3.2 Broadcast Message Generation Stage

Broadcast messages can be transmitted by encrypting the session key S with the encrypted message M and encrypting the message itself. Both methods are described as follows:

Step 1. The server produces and transmit $\langle CT \| C \rangle$ using the broadcast message $C = \left\langle S \cdot y^{aT}, h_1^{ar_1}, \ldots, h_k^{ar_k} \right\rangle = \langle B, H_1, \ldots, H_k \rangle$.

Step 2. User received message $CT = E_S(M)$ and C acquires message M or session key S using the private key.

$$S' = B / U^{\theta_i},\ U = \prod_{i=1}^{k} H_i^{r_i} (\bmod n)$$

$$U^{\theta_i'} = \left(\prod_{i=1}^{k} H_i^{r_i}\right)^{\theta_i'} = \left(\prod_{i=1}^{k} g^{ar_i\gamma_i}\right)^{\theta_i'} = \left(\prod_{i=1}^{k} g^{r_i\gamma_i}\right)^{a\theta_i'} = \left(\prod_{i=1}^{k} g^{\gamma_i\alpha_i t_i}\right)^{a} = \left(\prod_{i=1}^{k} h_i^{\alpha_i t_i}\right)^{a} = y^{aT} (\bmod n)$$

$$S' = \left(S \cdot y^{aT}\right) / y^{aT} (\bmod n),\ M' = D_{S'}(E_S(M))$$

4.3.3 Key Renewal Stage

In case of existing users who unsubscribe or new users who subscribe, the following process is carried out:

Step 1. User j requests for withdrawal.

Step 2. The server removes j 's renewal factor from renewal factor t_j using t_j^{-1} to update existing users' private keys and renewal element T update. After removal, the server renews private keys d_i' and re-transmits them to users. $\theta_i \cdot t_z \cdot t_j^{-1}$ acquired to transmitted d_i' .

$$T'=t_1\cdot\ldots\cdot t_k\cdot t_j^{-1}(\bmod n),\ \theta_i\cdot\gamma_i\cdot t_z\cdot t_j^{-1}=d_i'$$

$$d_i'/\gamma_i=\left(\theta_i\cdot\gamma_i\cdot t_z\cdot t_j^{-1}\right)/\gamma_i=\theta_i\cdot t_z\cdot t_j^{-1}(\bmod n),\ \theta_i''=\left(\prod_{i=1}^{k}(\alpha_i\cdot t_i)\Big/r_i\right)\cdot t_z\cdot t_j^{-1}(\bmod n)$$

Step 3. Users get broadcast message $C=\left\langle S\cdot y^{aTt_j^{-1}},h_1^{ar_1},\ldots,h_k^{ar_k}\right\rangle=\langle B',H_1,\ldots,H_k\rangle$ using the renewed keys and acquire message M by decoding the encrypted message C as follows:

$$CT=E_S(M),\ S'=B'/U^{\theta_i''},\ U=\prod_{i=1}^{k}H_i^{r_i}(\bmod n)$$

$$U^{\theta_i''}=\left(\prod_{i=1}^{k}H_i^{r_i}\right)^{\theta_i''}=\left(\prod_{i=1}^{k}g^{ar_i\gamma_i}\right)^{\theta_i''}=\left(\prod_{i=1}^{k}g^{r_i\gamma_i}\right)^{a\theta_i''}=\left(\prod_{i=1}^{k}g^{\gamma_i\alpha_i t_i}\right)^{a}=\left(\prod_{i=1}^{k}h_i^{\alpha_i t_i}\right)^{a}=y^{aT}(\bmod n)$$

$$S'=\left(S\cdot y^{aT'}\right)/y^{aT}(\bmod n),\ M'=D_{S'}(E_S(M))$$

4.4 Key Distribution Scheme Using User Information

Second proposal scheme possibility of the user to come being to decide, when the user provides information to server, with the character which will encryption key of the user creates and it is a scheme which it distributes. After information of the user is provided from the application model, with the character where the server will encryption key creation and the server predicts the user from the dissertation which it sees in the scheme which it distributes and user information which is a problem point of the scheme which creates a key is not contained not to be, it becomes and with to be caused by against the case the problem will be able to occur after it is a scheme which it complements. The proposal scheme which it sees efficient key renewal of section 4.3 it was identical and is was applying, it contains information of the user with it confronted to the user, with it confronted to a server together, also the process will be able to control in one time in order, the fact that it proposes is feature.

4.4.1 Key Generation and Distribution Stage

Key generation is processed by the server. The generation and transmission of private and public keys will go through the following process:

Step 1. The server generates value the set Γ to acquire and open user information.

Step 2. The user uses gathering information Γ which and is opened to the public ID_i value of the oneself is contained $\Xi_i=(ID_i)^{1/\gamma_i}\bmod n$ where it calculates. This time γ_i value which is selected the server is transmitted at the value which is identical with the number which is allocated to the user. Also it created to use, it calculates Ξ_i value and $U_i=\Xi_i\cdot\zeta\bmod p$, the Θ_i,U_i which generates data provider and it transmits.

Step 3. The server from the user it uses Θ_i,U_i, it is provided and it acquires user information ID_i'. Namely, it extracts $(\theta_i)^{1/b}=\left(\zeta^b\right)^{1/b}\bmod p=\zeta'$ from θ_i, it extracts and ζ' uses $(U_i/\zeta')=(\Xi_i\cdot\zeta)/\zeta'\bmod p=\Xi_i'$ acquires. Its uses Ξ_i' and it calculates

$(\Xi_i')^{\gamma_i} = \left(ID_i^{1/\gamma_i}\right)^{\gamma_i} \bmod p = ID_i'$, ID_i acquires keeps with γ_i it is provided to the user together, from information which is kept $h_i = g^{\gamma_i} \bmod q$ calculates. Opening to the public which corresponds to the value which it draws up key it generates and it opens to the public, the server when the personal key d_i which creates operation doing in the user, contains the renewal elements t_z of user and z transmits.

$$\langle y, h_1, h_2, \cdots, h_k \rangle,\ T = t_1 \cdot \ldots \cdot t_k (\bmod n)$$

$$\theta_i = \left(\prod_{i=1}^{k} \gamma_i a_i t_i\right) \Big/ \left(\prod_{i=1}^{k} \gamma_i r_i\right) \bmod n,\ d_i = \theta_i \cdot \gamma_i \cdot t_z (\bmod n)$$

Step 4. User acquires θ_i' from the transmitted d_i.

$$d_i / \gamma_i = (\theta_i \cdot \gamma_i \cdot t_z) / \gamma_i (\bmod n),\ \theta_i' = \left(\prod_{i=1}^{k} (\alpha_i \cdot t_i) \Big/ r_i\right) \cdot t_z (\bmod n)$$

4.4.2 Broadcast Message Generation Stage

Broadcast messages can be transmitted by encrypting the session key S with the encrypted message M and encrypting the message itself. Both methods are described as follows:

Step 1. The server produces and transmit $\langle CT \| C \rangle$ using the broadcast message $C = \langle S \cdot y^{aT}, h_1^{ar_1}, \ldots, h_k^{ar_k} \rangle = \langle B, H_1, \ldots, H_k \rangle$.

Step 2. User received message $CT = E_S(M)$ and C acquires message M or session key S using the private key.

$$S' = B / U^{\theta_i},\ U = \prod_{i=1}^{k} H_i^{r_i} (\bmod n)$$

$$U^{\theta_i'} = \left(\prod_{i=1}^{k} H_i^{r_i}\right)^{\theta_i'} = \left(\prod_{i=1}^{k} g^{ar_i\gamma_i}\right)^{\theta_i'} = \left(\prod_{i=1}^{k} g^{r_i\gamma_i}\right)^{a\theta_i'} = \left(\prod_{i=1}^{k} g^{\gamma_i\alpha_i t_i}\right)^{a} = \left(\prod_{i=1}^{k} h_i^{\alpha_i t_i}\right)^{a} = y^{aT} (\bmod n)$$

$$S' = \left(S \cdot y^{aT}\right) / y^{aT} (\bmod q),\ M' = D_{S'}(E_S(M))$$

4.4.3 Key Renewal Stage

The broadcast encryption the personal key which is provided to the user who participates to a distribution/transmission, server broadcast message it creates and it transmits. With this same process it transmits a message in the whole broadcast encryption participant. But this time, secession of the user the case where the new number occurs or, the next process it leads and it renews is renewed newly the key which and it uses broadcast message it generates and it is transmitted and user key.

Step 1. User j requests for withdrawal.

Step 2. The server removes j's renewal factor from renewal factor t_j using t_j^{-1} to update existing users' private keys and renewal element T update.

$$T' = t_1 \cdot \ldots \cdot t_k \cdot t_j^{-1} (\bmod n),\ \theta_i \cdot \gamma_i \cdot t_z \cdot t_j^{-1} = d_i'$$

<u>Step 3.</u> After removal, the server renews private keys d_i' and re-transmits them to users. $\theta_i \cdot t_z \cdot t_j^{-1}$ acquired to transmitted d_i' . Users get broadcast message $C = \left\langle S \cdot y^{aTt_j^{-1}}, h_1^{ar_1}, \ldots, h_k^{ar_k} \right\rangle = \langle B', H_1, \ldots, H_k \rangle$ using the renewed keys and acquire message M by decoding the encrypted message C as follows:

$$d_i'/\gamma_i = \left(\theta_i \cdot \gamma_i \cdot t_z \cdot t_j^{-1}\right)/\gamma_i = \theta_i \cdot t_z \cdot t_j^{-1} (\bmod n),\ \theta_i'' = \left(\prod_{i=1}^{k} (\alpha_i \cdot t_i) \Big/ r_i \right) \cdot t_z \cdot t_j^{-1} (\bmod n)$$

$$CT = E_S(M),\ S' = B' / U^{\theta_i''},\ U = \prod_{i=1}^{k} H_i^{r_i} (\bmod n)$$

$$U^{\theta_i''} = \left(\prod_{i=1}^{k} H_i^{r_i} \right)^{\theta_i''} = \left(\prod_{i=1}^{k} g^{ar_i\gamma_i} \right)^{\theta_i''} = \left(\prod_{i=1}^{k} g^{r_i\gamma_i} \right)^{a\theta_i''} = \left(\prod_{i=1}^{k} g^{\gamma_i\alpha_i t_i} \right)^{a} = \left(\prod_{i=1}^{k} h_i^{\alpha_i t_i} \right)^{a} = y^{aT} (\bmod n)$$

$$S' = \left(S \cdot y^{aT'}\right) / y^{aT} (\bmod n),\ M' = D_{S'}(E_S(M))$$

5 Comparison Analysis Between the Conventional Scheme and Proposed Scheme

This paper proposes the broadcast encryption method, which is more efficient than the existing method in generating and renewing keys. The stability of the proposed method is based on discrete algebra issue. Compared to the existing method, the proposed method achieves efficiency in user participation, key renewal, user withdrawal, or operating amount. In this section, the efficiency of the proposed method is presented vis-à-vis the existing method.

***User participation*:** In the existing method, the server anticipates users, generates keys in advance without user participation, and provides and distributes them to new users who subscribe. In this method, when an attack is made on the server itself, all keys created by the server can be affected.

***Key renewal*:** In the existing Key Pre-distribution Scheme (KPS), message is transmitted as encrypted using this scheme after the key is generated and distributed. When the session is closed after the user checks the transmitted message, a key is newly produced and transmitted. If an attack is made on the key, all keys will be re-generated instead of merely renewing them. In the proposed method, however, keys are ready to use after renewing the existing users' keys in case of subscription or withdrawal.

***Re-operation due to false prediction error*:** In the existing method and the proposed method - I, the server should set up and control the system. If the server controls flexible users, the anticipation of users should be carried out correctly. Therefore, the server should implement re-operation or additional operation in case initial anticipation fails. In the existing method, however, there is no such operation in case of failure of user anticipation. In the proposed method, user anticipation can be achieved smoothly through a simple operation like g^r when the server configures the system. Likewise, random number r can be generated on Z_p . Problems can also be solved by giving numbers larger than the expected number of users in advance.

6 Conclusion

Broadcast encryption is used to provide contents only for authorized users on the open network. Except author-ized users, nobody can obtain messages from the broadcast message; authorized users can obtain the session key, with the private key transmitted in advance. This paper proposes the method of generation, distribution, and renewal of private key and suggests an easier way of renewing after users' requests for withdrawl or process of the server's withdrawal for existing users. Further studies on user tracing and key cycling are recommended.

References

1. Amos Fiat, and Moni Naor, "Broadcast Encryption", Crypto'93, LNCS 773, 480-491
2. C. Blundo, Luiz A. Frota Mattos, D.R. Stinson, "Generalized Beimel-Chor schemes for Broadcast Enryption and Interactive Key Distribution", Crypto'96, LNCS 1109
3. Carlo Blundo, Luiz A. Frota Mattos, and Douglas R. Stinson, " Trade-offs Between Communication and Storage in Unconditionally Secure Schemes for Broadcast Encryption and Interactive Key Distribution", Crypto 98
4. Juan A. Garay, Jessica Staddon, and Avishai Wool, "Long-Lived Broadcast Encryption", Crypto'00, LNCS 1880, 333-352
5. Ignacio Gracia, Sebastia Martin, and Carles Padro, "Improving the Trade-off Between Storage and Communication in Broadcast Encryption Schemes", 2001
6. Dani Halevy, and Adi Shamir, "The LSD Broadcast Encryption Scheme," Crypto'02, LNCS 2442, 47-60
7. Yevgeniy Dodis and Nelly Fazio, "Public Key Broadcast Encryption for Stateless Receivers", DRM2002, 2002. 11. 18
8. Donald Beaver, and Nicol So, "Global, Unpredictable Bit Generation Without Broadcast," 1993
9. Michel Abdalla, Yucal Shavitt, And Avishai Wool, "Towards Marking Broadcast Encryption Practical", FC'99, LNCS 1648
10. Dong Hun Lee, Hyun Jung Kim, and Jong In Lim, "Efficient Public-Key Traitor Tracing in Provably Secure Broadcast Encryption with Unlimited Revocation