브로드캐스트 암호화에서의 효율적인 키 생성과 갱신 방법

이 덕 규[†] · 이 임 영[††]

요 약

브로드캐스트 암호화 기법은 공개된 네트워크 상에서 멀티미디어, 소프트웨어, 유료 TV 등의 디지털 정보들을 전송하는데 적용되고 있다. 브로드캐스트 암호화 기법에서는 중요한 것은 오직 사전에 허가된 사용자만이 디지털 정보를 얻을 수 있어야 한다는 것이다. 브로드캐스트 메시지가 전송되면 권한이 있는 사용자들은 자신이 사전에 부여받은 개인키를 이용하여 디지털 정보를 얻게 된다. 이와 같이 사용자가 디지털 정보를 획득하기 위해서는 브로드캐스터가 키를 생성하고 분배하는 과정이 필요하다. 또한 사용자가 탈퇴나 새로운 가입 시에 효율적인 키 갱신이 필요하게 된다. 이에 본 논문에서는 효율적인 키 생성과 분배, 키 갱신 방법에 대해 소개한다. 제안 방식은 두 가지 기법을 이용하는데, 하나는 서버가 사용자의 동의 없이 사용자를 예측하여 키를 생성하는 방법이고 다른 하나는 서버와 사용자가 서로 동의하여 키를 생성하는 방법이다. 두 제안 방식의 장점은 수신자는 하나의 비밀키를 이용하여 브로드캐스트되는 메시지를 복호화 할 수 있으며 후에 키가 갱신된다 하더라도 하나의 정보만을 이용하여 효과적으로 갱신이 가능하다.

A Efficient Key Generation and Renewal for Broadcast Encryption

Deok-Gyu Lee[†] · Im-Yeong Lee[††]

ABSTRACT

Broadcast encryption schemes are applied to transmit digital informations of multimedia, software, Pay-TV etc. in public network. Important thing is that only user who is permitted before only must be able to get digital information in broadcast encryption schemes. If broadcast message transfers, users who authority is get digital information to use private key given in the advance by oneself. Thus, user acquires message or session key to use key that broadcaster transmits, broadcaster need process that generation and distribution key in these process. Also, user secession new when join efficient key renewal need. In this paper, introduce about efficient key generation and distribution, key renewal method. Take advantage of two technique of proposal system. One is method that server creates key forecasting user without user's agreement, and another is method that server and user agree each other and create key. Advantage of two proposal system because uses a secret key broadcast message decryption do can and renewal is available effectively using one information whatever key renewal later.

키워드 : 브로드캐스트 암호화(Broadcast Encryption), 키 생성(Key Generation), 키 갱신(Key Renewal)

1. 서 론

최근 브로드캐스트 암호화 기법은 공개된 네트워크상에서 멀티미디어, 소프트웨어, 유료 TV 등의 디지털 정보들을 전송하는데 적용되고 있다.

키를 제공하는 방식 중에 하나인 공개키 방식은 세션키를 암호화하기 위한 그룹의 암호화키는 하나이고 이를 복호하기 위한 키는 여러 개의 무수히 많은 키를 이용함으로써 서버는 세션키를 암호화하고 각 사용자에게는 서로 다른 키를 이용하여 복호화 할 수 있도록 되어 있다[3, 4, 7, 8].

브로드캐스트 암호화 기법에서 중요한 것은 오직 사전에 허가받은 사용자만이 디지털 정보를 얻을 수 있어야 한다는 것이다. 브로드캐스트 메시지가 전달되면 권한이 있는 사용자들은 자신이 사전에 부여받은 개인키를 이용하여 디지털 정보를 얻게 된다. 브로드캐스트 암호화에 있어 가장 중요한 것은 키 생성, 분배, 갱신이다.

본 논문에서는 효율적인 키 생성과 분배, 키 갱신 방법에 대해 소개한다. 제안 방식의 두 가지 기법을 이용한다. 각각에 대해 살펴보면 우선 첫 번째는 서버가 사용자와 무관하게 사용자를 예측하여 키를 생성하는 방법이고, 다른 하나는 서버와 사용자가 서로 동의하여 키를 생성하는 방법이다. 두 제안 방식의 장점은 수신자는 하나의 비밀키를 이용하여 브로드캐스트 되는 메시지를 복호화 할 수 있으며 후에 키가 갱신된다 하더라도 하나의 정보만을 이용하여 효과적으로 갱신이 가능하다. 제안 방식에서는 빠른 키 갱신을 위한 키 갱신 인자를 첨가하고 이 인자를 통해 새로운 신규 가입자 혹은 탈퇴자가 발생하더라도 기존의 사용자에게 갱신 값을 제공함으로써 쉽게 키 갱신이 가능해지

※ 본 연구는 한국과학재단 목적기초연구(R05-2003-000-12019-0) 지원으로 수행되었음.
† 준 회 원 : 순천향대학교 대학원 전산학과
†† 종신회원 : 순천향대학교 정보기술공학부 교수
논문접수 : 2004년 1월 19일, 심사완료 : 2004년 2월 26일

도록 설계하였다.

본 논문은 브로드캐스트 암호화의 개요 및 기존 방식을 살펴 본 후, 제안 방식의 각 단계에 관하여 살펴본다. 또한 기존 방식과 제안 방식간의 비교 분석을 통하여 제안 방식에 대해 고찰하며, 마지막으로 결론을 맺도록 한다.

2. 브로드캐스트 암호화의 개요 및 기존 방식

2.1 브로드캐스트 암호화의 개요

브로드캐스트 암호화는 컨텐츠 제공자가 암호화된 형태로 수많은 정보를 전송하고 정당한 사용자만이 암호화된 정보를 복호화 할 수 있게 하는 많은 시나리오에 적용될 수 있다. 전형적인 예로, Pay-TV를 들 수 있으며 많은 브로드 캐스트 암호화 기법들이 제시되고 있다. 최초로 제시되었던 Cho외 2명이 제안한 브로드캐스트 암호화 기법은 다음과 같이 세 단계로 구성된다[1, 5, 6, 9-13]

- 컨텐츠 제공자 초기화 : 컨텐츠 제공자는 모든 사용자들에게 필요한 정보를 생성하고 이것을 초기 기록이라 명명한다.
- 사용자 초기화 : 개개인의 사용자들은 컨텐츠 제공자에게 등록하는 단계이다. 이 단계 후 사용자가 저장하는 정보는 사용자의 개인키(Personal Key)이고 정보제공자는 각 사용자 초기화 단계 후에 각 사용자에 대한 초기 기록을 갱신한다.
- 세션 송신 : 컨텐츠 데이터는 세션키로 암호화 되고 이 세션키는 세션이라 부르는 작은 부분들로 분할되서 전송된다. 이 때 각 세션은 각각 다른 부분 세션키로 암호화되어 전송되고 정당한 사용자들이 자기들의 개인키를 가지고 부분 세션키를 복호화 할 수 있게 함으로써 실제 데이터를 얻을 수 있게 해주는 세션키를 얻게 한다.

2.2 기존 방식

Narayanan[2]등이 2003년에 RSA 기반을 둔 악의적인 사용자의 추적능력을 가진 실용적인 유료 TV 스킴을 제안하였다. 악의적인 사용자를 추적하는 방식은 다음의 원리를 이용하여 제안하고 있다.

> *임의의 $s(<t)$ 벡터들의 선형 조합으로 주어진 n개의 $(t+1)$차 벡터 $X_1, X_2, \cdots, X_n$을 구성하면, 사용된 정확한 벡터들을 높은 확률로 알아 낼 수 있다.*

2.2.1 Narayanan 방식

m개의 채널을 브로드캐스트 하는 컨텐츠 제공자와 n명의 사용자가 있다고 가정한다. 프로토콜은 다음의 단계로 나뉘어져 있다. Setup, AddStream, AddUser, Subscribe, Unsubscribe, Broadcast, Receive 7개의 알고리즘으로 구성된다.

사용자의 채널 수신여부는 $m \times n$행렬인 Subsc로 나타내며 사용자 U_j가 S_i에 등록되어 있으면 $subsc[i,j]$는 1의 값을 가지고 등록되어 있지 않으면 0을 가지게 된다.

[Step 1] Setup : 컨텐츠 제공자는 다음과 같은 변수 값을 생성한다. $N = pq, R, d_r \le R\{1,2,\cdots\phi(N)\}$ 이때 $1 \le r \le 4+t$ 이고 p와 q는 큰 값의 소수이며, R은 랜덤값이다. p, q, d는 컨텐츠 제공자의 비밀키로 구성되며, 컨텐츠 제공자는 공개키 N을 공개한다.

[Step 2] AddStream : 시스템에 새로운 채널 스트림 S_i를 추가하기 위해 컨텐츠 제공자는 큰 위수를 갖는 임의의 $g_i \in Z_N^*$를 선택한다. $Subsc[i,j]$는 모든 j에 대해 0의 값을 가지도록 설정하고 g_i 값은 공개하지 않는다.

[Step 3] AddUser : 새로운 사용자 U_j 가입시키려면 컨텐츠 제공자는 $\sum_{r=1}^{t+4} e_{rj} d_r = R\Phi(N)+1$를 만족하는 $(e_{1j}, e_{2j}, \cdots, e_{(t+4)j})$을 선택한다. 이때 U_j는 비밀키를 안전한 메모리에 저장한 복호기(Set-Top Terminal)를 받게 되고 U_j의 비밀키는 $(e_{1j}, e_{2j}, \cdots, e_{(t+4)j})$이 된다.

[Step 4] Subscribe : 사용자 U_j가 서비스 S_i를 구독하면 컨텐츠 제공자는 사용자 U_j에게 g_i^{e1j}를 전송하고 $Subsc[i,j]$ 값을 1 변경한다.

[Step 5] Unsubscribe : 사용자 U_j가 서비스 S_i의 구독을 중지하면 컨텐츠 제공자는 $Subsc[i,j]=0$으로 설정한다. AddStream 알고리즘에서 했던 것과 같이 새로운 g_i 값을 선택하여 $Subsc[i,j]=1$인 모든 사용자들에게 g_i^{e1j}를 전송한다.

[Step 6] Broadcast : 메시지 M을 채널 스트림 S_i에 전송하려면 컨텐츠 제공자 $\Phi(N)$과 서로 소인 랜덤값 x를 선택한다. 암호화된 데이터 $C=(x, C_1, C_2, \cdots, C_{t+4})$를 전송하며, 이때 $C_1 = M^{d1} g_i^x$, $C_2 = M^{d2}$, $C_{t+4} = M^{dt+4}$이다.

[Step 7] Receive : 채널 스트림 S_i로 전송되는 암호화된 데이터 $C=(x, C_1, C_2, \cdots, C_{t+4})$로 복호화하기 위하여 사용자 U_j는 비밀키 $(e_{1j}, e_{2j}, \cdots, e_{(t+4)j})$를 이용하여 다음 $\left(\prod_{r=1}^{t+4} C_r^{e_r}\right) / g_i^{xe_{1j}}$을 계산한다. 사용자 U_j는 다음과 같은 과정을 거쳐 컨텐츠 데이터 M을 다음과 같이 복원한다.

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

Narayanan 스킴의 문제점은 하나의 채널당 $(x, C_1, C_2, \cdots, C_{t+4})$의 통신량이 필요하다는 것이다. 통신량은 채널의 개

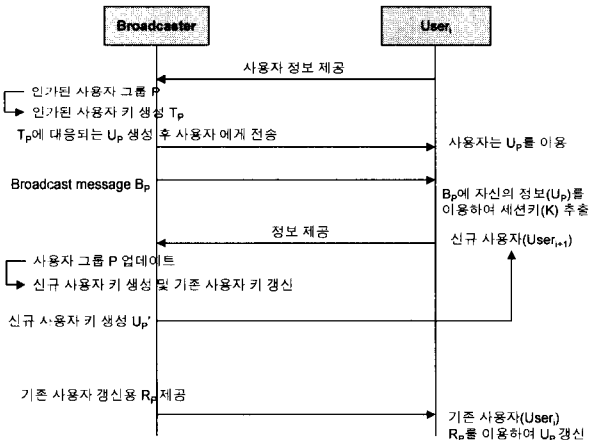
수와 연관되어 있기 때문에 채널이 늘어나면 통신량도 증가하는 문제가 발생할 수 있게 된다. 또한 컨텐츠 제공자는 배신자 U_j를 찾는다하더라도 U_j를 제외한 모든 가입자에게 다시 새로운 비밀키를 배포해야만 U_j를 탈퇴시킬 수 있다.

3. 제안 방식

기존 사용자의 탈퇴 혹은 신규 사용자의 가입에 따른 효율적인 키 갱신을 위해 다음과 같은 두 가지 방식을 적용하는 방식을 기술하고 이를 바탕으로 각각의 방식에 대해 제안한다. 첫 번째 방식은 서버가 사용자의 동의 없이 사용자를 예측하여 키를 생성하고 분배하여 암호화 통신이 이루어지는 과정이며, 두 번째 방식은 서버가 사용자의 동의를 받아야만 브로드캐스팅 암호화키를 생성할 수 있는 방식을 제안한다. 이와 같은 방식에 대하여 다음과 같이 모델을 제시할 수가 있다.

3.1 적용 방식

브로드캐스트 암호화는 다음과 같이 2가지 모델을 기반으로 할 수 있다. 적용모델간의 차이점이 있지만 각각에 대하여 살펴보면 첫 번째 모델은 (그림 1)과 같다.

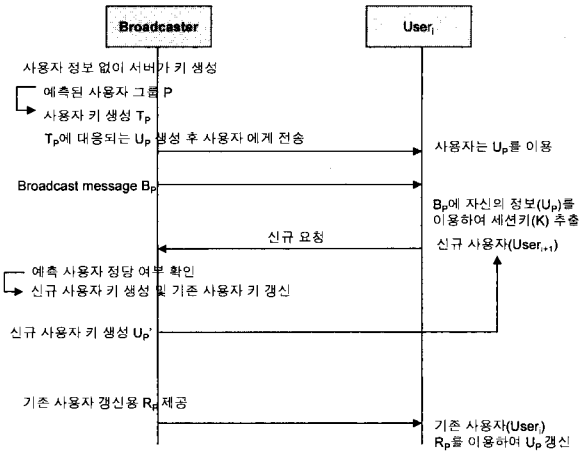


(그림 1) 사용자 정보 제공 방식

사용자와 서버간의 정보를 이용하여 키를 생성하고 분배하는 방식이다. 이는 전송되는 방식에서 차이가 존재할 뿐 제공되는 메시지가 이전의 사용 그룹에 의해 결정되는 점에서 기존 멀티캐스트 방식과 유사하다.

키 생성과정에서 사용자가 참여하여야 하므로 생성시간에 사용자의 참여 시간이 포함될 수 있다. 키 갱신과정에서도 기존 사용자의 탈퇴와 신규 과정에 따라 사용자의 최초 참여 때 부여받은 키가 키 갱신에 따른 소요시간이 많이 발생하게 된다.

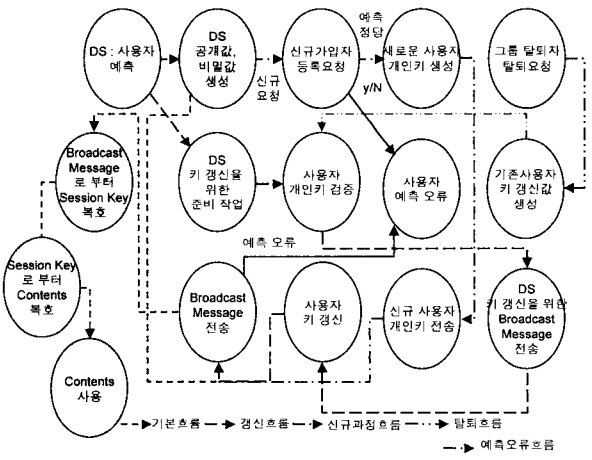
앞에서 살펴본 (그림 1)의 적용모델과는 다르게 서버가 키를 생성하는 방식으로 서버가 단독으로 참여할 사용자를 예측하여 키를 생성한다((그림 2) 참조). 이러한 방법은 사용자의 동의 없이 서버가 모든 사용자의 키를 생성하게 됨으로써 빠른 생성과 빠른 갱신이 가능하다. 하지만 서버가 악의적인 목적 혹은 서버가 공격의 대상이 되었을 경우 많은 취약점을 내포하고 있다.



(그림 2) 사용자 정보 제공없는 방식

3.2 제안 방식 개요

다음은 제안 방식의 전체적인 개요에 대하여 살펴본다. (그림 3)은 본 제안 방식에서 나타날 수 있는 시나리오를 구분한 것이다. 시나리오를 살펴보면 기본적인 흐름, 갱신흐름, 신규과정흐름, 탈퇴흐름, 사용자 예측 오류 흐름으로 볼 수 있다. 아래의 시나리오에 따라 제안 방식은 크게 세 부분으로 구분된다. 첫째 키 생성 및 분배 부분, 브로드캐스트 메시지 생성 부분, 마지막으로 키 갱신 부분으로 나뉠 수 있다. 두 가지 제안 방식은 아래의 전체흐름에 동일하게 적용될 수 있다. 초기 키 생성 및 분배부분에서만 서버 예측 혹은 사용자 동의에서만 다르게 진행되고 후의 흐름에서는 동일하게 진행된다.



(그림 3) 제안 방식 흐름에 따른 분류

또한 본 제안 방식 중 첫 번째 제안 방식은 다음과 같은 특징을 가지고 있다. 사용자의 개인키는 서버가 생성하며, 사용자 이외의 사람은 브로드캐스팅되는 메시지에 대해 복호할 수 없다. 신규가입자, 사용자 탈퇴 등에 따른 키 갱신이 용이하다. 두 번째 제안 방식의 특징은 다음과 같다. 사용자의 개인키는 사용자의 동의 과정을 거쳐 생성하며, 여러 사용자가 모였을 때 공개키를 생성하고 이를 통해 브로드캐스팅되는 메시지를 암호화하여 전송하게 된다. 이 제안 방식에 있어서도 신규 가입자 혹은 사용자 탈퇴는 용이하게 일어나게 되는데 사용자가 제공한 정보를 삭제함으로써 가능하다.

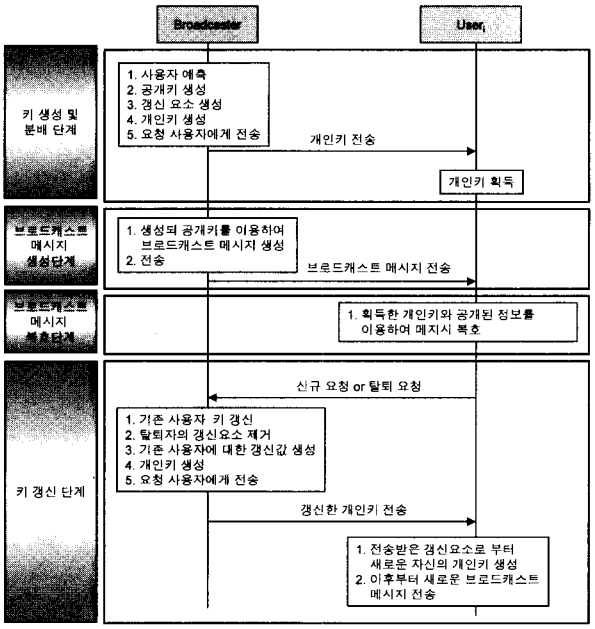
3.3 시스템 계수

다음은 본 방식에서 사용되는 시스템 계수를 기술한 것이다.

- p : 소수 $\geq 512bit$
- q : 소수 $\geq 160bit (q \mid p-1)$
- l : 개인키 생성을 위한 랜덤수
- e : 공개 암호화 키
- θ_i : 세션키
- $\gamma_i \in Z_p$: 랜덤수
  - $\Gamma = \gamma_1, \cdots, \gamma_k$
- M : 메시지
- S : 세션키
- k : 예측 사용자의 수
- i : 사용자($i = 1, \cdots, k$)
- j : 탈퇴자
- r_i : 랜덤 수 집합 $(r_i \in Z_p) \rightarrow (r_1, \cdots, r_k)$
- $h_i = g^{r_i}$
- $\langle y, h_1, \cdots, h_k \rangle$: 공개키
  - $y = \prod_{i=1}^{k} h_i^{a_i}$
  - a_i : 랜덤수 $(a_i \in Z_q)\{a_1, \cdots, a_k\}$
- a : 랜덤 요소 $(a \in Z_q)$
- C : 방송 메시지(Broadcast message)
  - $C = \langle M(or\ S)\, y^{aT}, h_1^a, \cdots, h_k^a \rangle = \langle B, H_1, \cdots, H_k \rangle$
  - $B = M(\text{or } S) y^a$
  - $H_i = \prod_{i=1}^{k} h_i^a$
  - T : 키 갱신을 위한 인자 $(t_1, \cdots, t_k \in Z_q)$
  - $T = t_1 \cdot \cdots \cdot t_k$
- o : 보안 파라메타
- b : 사용자가 생성한 공개정보 $(b \in Z_q)$
- Θ, U : 사용자가 등록에 참여하기 위한 사용자 정보
- ζ : 사용자가 임의로 선택한 값
- Ξ : 사용자의 ID를 보관한 값

3.4 제안 방식 1

첫 번째 제안 방식은 서버가 사용자를 예측하여 키를 생성하고 후에 분배하는 방식이다. 적용모델에서는 첫 번째 방식으로 사용자의 정보가 제공되지 않고 서버가 생성하는 방식이다. 본 논문에서는 제안하는 방식 중에서 키 갱신에 초점을 맞추어 좀 더 쉽고 효율적으로 갱신이 될 수 있도록 키 갱신요소를 삽입하는 형태로 제안하였다. 삽입되는 키 갱신요소를 통해 후에 탈퇴가 발생하는 경우 사용자의 갱신요소만을 삭제함으로써 전체적으로 효율적인 키 갱신이 되도록 하였다.



(그림 4) 제안 방식 1 흐름도

3.4.1 키 생성 및 분배 단계

키 생성은 서버의 담당이며, 개인키와 공개키를 생성하고 전달하기 위해 다음의 일련의 과정을 거친다.

[Step 1] 서버는 사용자 $(i=1, \cdots, k)$를 예측하여 만족하는 랜덤수 (r_i)를 생성한다.

[Step 2] 서버는 선택된 랜덤 수를 바탕으로 공개키 작성에 필요한 값을 생성한다.

$$h_i \equiv g^{r_i} \mod q$$

작성한 값에 해당하는 공개키 값을 생성하고 갱신을 위한 T 요소를 생성하여 공개한다.

$$\langle y, h_1, \cdots, h_k \rangle, \quad T = t_1 \cdot \cdots \cdot t_k$$

[Step 3] 서버는 생성된 값 h_i를 이용하여 공개키를 작성한 후 이를 바탕으로 개인키를 계산한다.

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

[Step 4] 서버는 생성된 정보 d_i를 사용자에게 전송한다.

$$d_i = \theta_i \cdot \gamma_i$$

[Step 5] 사용자는 전송받은 d_i에서 θ_i를 획득하게 된다. 이때, Γ가 공개되어 있다 하더라도 θ_i는 비밀로 유지될 수 있다.

$$\theta_i = \theta_i \cdot \gamma_i / \gamma_i$$

3.4.2 브로드캐스트 메시지 생성 단계

브로트캐스트 메시지를 전송하는데 있어 메시지를 암호화한 세션키를 암호화하여 전송할 수 있고 메시지 자체를 암호화하여 전송할 수 있다. 다음에서는 두 가지 모두를 고려하여 기술한다.

[Step 1] 서버는 다음과 같이 브로드캐스트 메시지를 작성하여 전송한다.

$$C = \langle M(or\ S) y^{aT}, h_1^a, \cdots, h_k^a \rangle = \langle B, H_1, \cdots, H_k \rangle$$

[Step 2] 전송받은 메시지는 개인키를 이용하여 메시지 M이나 세션키 S를 획득한다.

$$M(\text{or } S) = B / U^{\theta_i},\ U = \prod_{i=1}^{k} H_i^{\gamma_i}$$

$$U^{\theta_i} = \left(\prod_{i=1}^{K} H_i^{\gamma_i}\right)^{\theta_i} = \left(\prod_{i=1}^{k} g^{ar_i\gamma_i}\right)^{\theta_i} = \left(\prod_{i=1}^{k} g^{r_i\gamma_i}\right)^{\theta_i a}$$

$$= \left(\prod_{i=1}^{k} g^{r_i\alpha_i}\right)^{a} = \left(\prod_{i=1}^{k} h_i^{\alpha_i T}\right)^{a} = y^{aT}$$

$$M(\text{or } S) = M(\text{or } S) \cdot y^{aT} / y^{aT}$$

3.4.3 키 갱신 단계

사용자의 탈퇴 혹은 신규 가입자가 발생한 경우 다음과 같은 과정을 수행한다.

[Step 1] 사용자 j가 탈퇴를 요청

[Step 2] 서버는 기존 사용자의 개인키를 갱신하기 위해 갱신요소인 T에서 사용자 j의 갱신요소를 제거한다.

[Step 3] 서버는 탈퇴 사용자의 갱신요소를 제거한 후 개인키를 갱신하고 사용자에게 전송한다.

$$\theta_i \cdot \gamma_i \cdot t_j^{-1} = d_i'$$

[Step 4] 사용자는 갱신된 키를 이용하여 브로드캐스트 메시지를 전송받고 다음과 같이 암호화된 메시지를 복호화하여 메시지를 획득하게 된다.

$$(C = \langle B, H_1, \cdots, H_k \rangle) = \left(C = \langle M(\text{or } S) \cdot y^{aTt_j^{-1}}, h_1^a, \cdots, h_k^a \rangle\right)^{\theta_i}$$

$$M(\text{or } S) = B / U^{\theta_i t_j^{-1}},\ U = \prod_{i=1}^{k} H_i^{\gamma_i}$$

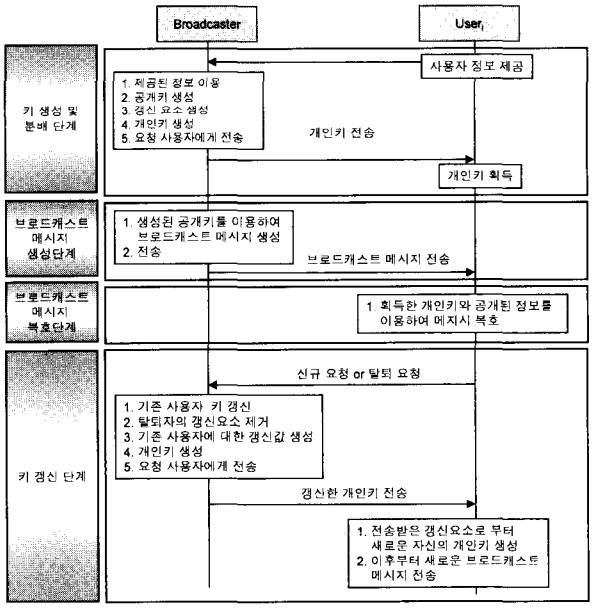
$$U^{\theta_i t_j^{-1}} = \left(\prod_{i=1}^{k} H_i^{\gamma_i}\right)^{\theta_i t_j^{-1}} = \left(\prod_{i=1}^{k} g^{ar_i\gamma_i}\right)^{\theta_i t_j^{-1}} = \left(\prod_{i=1}^{k} g^{r_i\gamma_i}\right)^{\theta_i a t_j^{-1}}$$

$$= \left(\prod_{i=1}^{k} g^{r_i\alpha_i}\right)^{at_j^{-1}} = \left(\prod_{i=1}^{k} h_i^{\alpha_i t_j}\right)^{a} = y^{at_j^{-1}}$$

$$M(or\ S) = M(or\ S) \cdot y^{at_j^{-1}} / y^{at_j^{-1}}$$

3.5 제안 방식 2

두 번째 제안 방식은 사용자의 수가 정해져 있고 사용자가 서버에게 정보를 제공하면 이를 바탕으로 사용자의 키를 생성하고 분배하는 방식이다. 적용모델에서 사용자의 정보가 제공된 후 서버가 이를 바탕으로 키 생성과 분해하는 방식으로 본 논문에서는 제안방식 1에서의 문제점인 사용자의 정보가 포함되지 않게 됨으로 인해 후에 문제가 발생할 수 있는 경우에 대해 보완하는 방식이다. 제안 방식 2는 제안 방식 1의 효율적인 키 갱신은 동일하게 적용하고 있다. 또한 사용자의 정보를 포함함으로써 사용자에 대한 인증과 함께 서버에 대한 인증과정도 한 번에 처리할 수 있도록 제안한 것이 특징이다.



(그림 5) 제안방식 2 흐름도

3.5.1 키 생성 및 분배 단계

키 생성은 서버의 담당이며, 개인키와 공개키를 생성하고 전달하기 위해 다음의 일련의 과정을 거친다.

[Step 1] 서버(데이터 제공자)는 사용자의 정보 획득을 위한 값을 생성하여 공개한다.

$$\gamma_i \in \Gamma,\ (\gamma_1, \cdots, \gamma_k)$$

[Step 2] 사용자는 공개된 정보 Γ과 자신의 ID를 이용하여 다음의 값을 계산한다.

$$ID_i = (\Xi_i)^{\gamma_i} (\mathrm{mod}\ n)$$

[Step 3] 사용자는 생성된 값을 이용하여 다음의 값을 계산한다.

$$\Xi_i \equiv (ID_i)^{1/\gamma_i} (\mathrm{mod}\ n)$$
$$U \equiv \Xi_i \cdot \zeta (\mathrm{mod}\ n)$$
$$\Theta \equiv \zeta^b (\mathrm{mod}\ n)$$

[Step 4] 사용자는 생성된 값(Θ, U)를 데이터제공자에게 전송한다.

[Step 5] 서버는 데이터는 제공받은(Θ, U)를 이용하여 사용자 정보 ID_i를 획득한다. Θ로부터 ζ을 추출하면 추출한 값을 이용하여 Ξ_i를 얻는다. 얻은 Ξ_i값을 이용하여 ID_i값을 획득한다.

$$\Theta \equiv \zeta^b (\mathrm{mod}\ n) = \zeta$$
$$U \equiv \Xi_i \cdot \zeta (\mathrm{mod}\ n) = \Xi_i$$
$$\Xi_i \equiv (ID_i)^{1/\gamma_i} (\mathrm{mod}\ n)$$
$$ID_i \equiv (\Xi_i)^{\gamma_i} (\mathrm{mod}\ n)$$

[Step 6] 서버는 사용자 i의 정보를 이용하여 ID_i열을 선택하고 다음을 계산한다.

$$h_i \equiv g^{r_i} \bmod q$$

작성한 값에 해당하는 공개키 열을 생성하고 공개하고 갱신을 위해 T 요소를 생성한다.

$$\langle y, h_1, \cdots, h_k \rangle, \quad T = t_1 \cdot \cdots \cdot t_k$$

[Step 7] 서버는 생성된 값 h_i를 이용하여 공개키를 작성한 후 이를 바탕으로 개인키를 계산한다.

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

[Step 8] 서버는 생성된 개인키 d_i를 사용자에게 전송한다.

$$d_i = \theta_i \cdot \gamma_i$$

[Step 9] 사용자는 전송받은 d_i에서 θ_i를 획득한다.

$$d_i = \theta_i \cdot \gamma_i / \gamma_i$$

3.5.2 브로드캐스트 메시지 생성 단계

브로트캐스트 메시지를 전송하는데 있어 메시지를 암호화한 세션키를 암호화하여 전송할 수 있고 메시지 자체를 암호화하여 전송할 수 있다. 다음에서는 두 가지 모두를 고려하여 기술한다.

[Step 1] 서버는 다음과 같은 브로드캐스트 메시지를 작성하여 전송한다.

$$C = \langle M(or\ S) y^{aT}, h_1^a, \cdots, h_k^a \rangle = \langle B, H_1, \cdots, H_k \rangle$$

[Step 2] 사용자는 전송받은 메시지는 개인키를 이용하여 메시지 M이나 세션키 S를 획득한다.

$$M(\mathrm{or}\ S) = B / U^{\theta_i},\ U = \prod_{i=1}^{K} H_j^{\gamma_i}$$
$$U^{\theta_i} = \left(\prod_{i=1}^{k} H_i^{\gamma_i} \right)^{\theta_i} = \left(\prod_{i=1}^{k} g^{ar_i\gamma_i} \right)^{\theta_i} = \left(\prod_{i=1}^{k} g^{r_i\theta_i} \right)^{\theta_i a}$$
$$= \left(\prod_{i=1}^{k} g^{r_i\alpha_i} \right)^{a} = \left(\prod_{i=1}^{k} h_i^{\alpha_i T} \right)^{a} = y^{aT}$$
$$M(\mathrm{or}\ S) = M(\mathrm{or}\ S) \cdot y^{aT} / y^{aT}$$

3.5.3 키 갱신 단계

사용자의 탈퇴 혹은 신규 가입자가 발생한 경우 다음과 같은 과정을 거친다.

[Step 1] 사용자 j가 탈퇴를 요청

[Step 2] 서버는 기존 사용자의 개인키를 갱신하기 위해 갱신요소인 T에서 사용자 j의 갱신요소를 제거한다.

[Step 3] 서버는 사용자의 갱신 요소를 제거한 후 개인키를 갱신하고 사용자에게 전송한다.

$$\theta_i \cdot \gamma_i \cdot t_j^{-1} = d_i'$$

[Step 4] 사용자는 기존 자신의 개인키에 대해 갱신 후 갱신된 키를 이용하여 브로드캐스트 메시지를 전송받고 다음과 같이 암호화된 메시지를 복호화하여 메시지를 획득하게 된다.

$$(C = \langle B, H_1, \cdots, H_k \rangle) = \left(C = \langle M(\mathrm{or}\ S) \cdot y^{aTt_j^{-1}}, h_1^a, \cdots, h_k^a \rangle \right)^{\theta_i}$$
$$M(\mathrm{or}\ S) = B / U^{\theta_i t_j^{-1}},\ U = \prod_{i=1}^{k} H_i^{\gamma_i}$$
$$U^{\theta_i t_j^{-1}} = \left(\prod_{i=1}^{k} H_i^{\gamma_i} \right)^{\theta_i t_j^{-1}} = \left(\prod_{i=1}^{k} g^{a r_i \gamma_i} \right)^{\theta_i t_j^{-1}} = \left(\prod_{i=1}^{k} g^{r_i^{\gamma_i}} \right)^{\theta_i a t_j^{-1}}$$
$$= \left(\prod_{i=1}^{k} g^{r_i\alpha_i} \right)^{a t_j^{-1}} = \left(\prod_{i=1}^{k} h_i^{\alpha_i t_j^{-1}} \right)^{a} = y^{a t_j^{-1}}$$
$$M(or\ S) = M(or\ S) \cdot y^{a t_j^{-1}} / y^{a t_j^{-1}}$$

4. 비교 분석

본 논문에서는 기존의 방식보다 효율적인 키 생성과 키 갱신을 위한 브로드캐스트 암호화 방식을 제안하였다. 본 제안 방식들의 안전성은 이산대수의 문제에 기반을 두고

〈표 1〉 기존 방식과 제안 방식 비교

분 석 / 방 식	사용자 참여	키 갱신	N회 탈퇴	키의 연속성	Traitor Tracing	초기 예측 오류에 따른 재연산	암호 블록길이
기존 KPS[3]	×	×	×	○	×	필 요	
Broadcast Encryption[1]	×	○	×	△	×	필 요	O(Z+1)
IKPS[4]	○	○	×	△	×	불필요	
Narayanan[2]	×	○	△	○	○	필 요	O(2z)×|p|
제안방식 1	×	○	○	○	×	불필요	O((n+1)/2)
제안방식 2	○	○	○	○	×	불필요	O(n)×|ID|

○ : 가능, × : 불가능, △ : 일부 제공

있다. 기존의 방식에 비해 사용자의 참여, 키 갱신, 사용자의 탈퇴 혹은 연산량에 있어 효율성을 나타내고 있다. 본 장에서는 제안 방식과 기존 방식을 비교하여 제안 방식의 효율성을 보인다.

4.1 사용자 참여

기존의 방식에서 보면 서버는 사용자의 참여 없이 사용자를 예측하여 키를 미리 생성하여 새로 가입하는 사용자에게 키를 제공하고 분배하는 형태를 가지고 있다. 이런 방식은 키를 생성하는 서버자체에 대하여 공격이 행하여 질 경우 서버가 생성한 키 전체에 대하여 악의적인 행위가 발생할 수 있다. 본 논문의 제안 방식 2에서는 이러한 문제점을 해결하고자 키를 생성하고 분배하기 이전에 사용자의 그룹을 먼저 형성하여 사용자의 정보를 이용함으로써 서버에 대한 공격을 대비하였다. 서버에 사용자의 정보가 전송될 때 사용자와 서버가 같은 정보가 다르게 함으로써 사용자들의 정보를 수집하여 키를 유도하지 못하도록 제안하였다. 하지만 제안 방식 1에서는 서버에 대한 공격보다는 빠른 키 생성 및 키 갱신에 주목적을 두고 있으므로 사용자 참여과정을 배제하였다.

4.2 키 갱신

기존 KPS(Key Predistribution Scheme)에서는 키가 생성되고 분배가 되어진 후 이를 이용하여 암호화하여 메시지를 전송하게 된다. 전송된 메시지를 사용자가 확인한 후 한 세션이 종료되면 키를 새로 생성하여 전송되거나, 키에 대하여 공격이 이루어진 경우 키를 갱신하지 아니하고 전체적으로 다시 생성하게 된다. 하지만 제안 방식들에서는 사용자의 가입 혹은 탈퇴가 발생하면 기존 사용자들의 키를 갱신하고 사용이 가능하다. 키 갱신은 초기 키 생성 시에 키 갱신 요소인 T인자를 삽입하게 된다. 차후 사용자 탈퇴/강제 탈퇴 등과 같은 상황이 발생되면 서버는 탈퇴자의 키 갱신 정보인 t_i^{-1}를 제공함으로써 사용자는 간단한 연산으로 키 갱신을 마치게 된다. 따라서 본 제안 방식들은 기존 방식보다 빠르고 효율적으로 키 갱신이 되도록 제안하였다.

4.3 초기 예측 오류에 따른 재연산

기존 방식과 제안 방식 I의 경우 서버가 시스템을 설정하고 관리해야한다. 만일 서버가 유동적인 사용자를 관리한다면 사용자에 대한 예측이 올바르게 이뤄져야한다. 그러므로 서버는 초기 예측에 대한 오류가 발생하였을 경우 재연산이나 혹은 추가 연산을 실시해야 한다. 하지만 기존 방식의 경우에서는 이러한 사용자 예측 오류에 대하여 수행할 수 있는 연산이 없다. 본 논문에서 제공하는 방식에서는 서버가 시스템을 설정하는데 사용자에 대한 예측 연산을 원활히 할 수 있도록 g^r과 같이 간편한 연산을 통해 이뤄질 수 있도록 제안하였다. 또한 랜덤한 수 r에 대해서는 Z_p상에서 생성하게 되며 r에 대해서 사전에 예측 사용자보다 많은 수를 만들면 해결할 수 있다. 제안 방식 2에서는 초기 사용자의 예측 없이 일정한 사용자의 그룹을 형성한 뒤에 키를 생성하고 분배하게 됨으로 초기 사용자의 예측 오류가 발생할 위험이 감소하게 된다. 하지만 2장의 적용 모델에서도 언급했던 바와 같이 사용자의 늦은 가입이나 늦은 통신은 전체적인 효율성을 감소시키는 원인이 될 수도 있다.

5. 결 론

브로드캐스트 암호화 기법은 공개된 네트워크상에서 멀티미디어, 소프트웨어, 유료 TV 등의 디지털 정보들을 전송하는데 적용되고 있다. 브로드캐스트 암호화 기법에서는 중요한 것은 오직 사전에 허가된 사용자만이 디지털 정보를 얻을 수 있어야 한다는 것이다. 브로드캐스트 메시지가 전송되면 권한이 있는 사용자들은 자신이 사전에 부여받은 개인키를 이용하여 디지털 정보를 얻게 된다. 이와 같이 사용자는 브로드캐스터가 전송하는 키를 이용하여 메시지나 세션키를 획득하게 되는데, 이러한 과정에서 브로드캐스터가 키를 생성하고 분배하는 과정이 필요하다. 또한 사용자가 탈퇴나 새로운 가입 시에 효율적인 키 갱신이 필요하게 된다.

인가된 사용자이외에는 브로드캐스트 되는 메시지에 대해 아무런 정보를 얻어낼 수 없으며, 인가된 사용자는 사전에 전송된 개인키를 이용하여 세션키를 취득할 수 있게 된

다. 본 논문은 사용자에게 개인키의 생성, 분배와 갱신에 이르는 방법을 제안한다. 제안 방식 1은 사용자의 동의 없이 서버가 사용자의 수를 예측하여 키를 생성하는 방법이고 제안 방식 2는 사용자의 동의하에 서버가 사용자로부터 정보를 받아 키를 생성하는 방식이다. 본 논문에서 제안 방식 1은 키 갱신에 초점을 두어 키 갱신 인자인 T요소를 통하여 효율적인 키 갱신이 이뤄지도록 하였으며, 제안 방식 2는 기존 사용자의 동의 없이 키 정보를 생성하므로 인해 서버가 노출될시 사용자의 키가 모두 노출된다는 취약점에 대해 사용자가 자신의 정보를 생성하여 서버에 제공하게 함으로써 서버에 대한 취약점을 보안하였다. 따라서 본 논문은 브로드캐스트 암호화를 위한 키 생성과 키 갱신에 효율적인 방안을 제시하였는데 각각의 제안 방식에 따라 음악, 방송과 같은 컨텐츠 분배에 있어 효율적인 제공 및 서비스 제공자의 컨텐츠 분배에 대한 효율적인 방식이 될 수 있을 것이다. 마지막으로 본 연구는 불법적인 컨텐츠 사용 및 사용자 키의 누출 시 불법적인 사용자에 대한 사용자 추적, 세션에 대해 각각의 키를 다시 생성하고 분배하는 것은 서버나 사용자에 많은 부담을 전가시킬 수 있는데 키 주기에 관한 연구를 통해 효율적인 키 관리를 이룰 수 있을 것이라 본다.

참 고 문 헌

[1] Amos Fiat and Moni Naor, "*Broadcast Encryption,*" Crypto'93, pp.480-491, 1993.

[2] A. Narayana, "*Practical Pay TV Schemes,*" to appear in the Proceedings of ACISP 03, July, 2003.

[3] C. Blundo, Luiz A. Frota Mattos and D. R. Stinson, "*Generalized Beimel-Chor schemes for Broadcast Encryption and Interactive Key Distribution,*" Theoretical Computer Science, Vol.200, pp.313-334, 1998.

[4] 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,*" In Advances in Cryptology-Crypro '96, Lecture Notes in Computer Science 1109, pp.387-400.

[5] Carlo Blundo and A. Cresti, "*Space Requirements for Broadcast Encryption,*" EUROCRYPT '94, LNCS 950, pp. 287-298, 1994.

[6] Donald Beaver and Nicol So, "*Global, Unpredictable Bit Generation Without Broadcast,*" EUROCRYPT '93, Volume 765 of Lecture Notes in Computer Science, Springer-Verlag, pp.424-434, pp.23-27, May, 1993.

[7] Dong Hun Lee, Hyun Jung Kim and Jong In Lim, "*Efficient Public-Key Traitor Tracing in Provably Secure Broadcast Encryption with Unlimited Revocation Capability,*" KoreaCrypto '02, 2003.

[8] D. Boneh and M. Franklin, "*AN Efficient Public Key Traitor Tracing Scheme,*" CRYPTO '99, LNCS 1666, pp. 338-353, 1999.

[9] Dani Halevy and Adi Shamir, "*The LSD Broadcast Encryption Scheme,*" Crypto '02, Lecture Notes in Computer Science, Vol.2442, pp.47-60, 2002.

[10] Ignacio Gracia, Sebastia Martin and Carles Padro, "*Improving the Trade-off Between Storage and Communication in Broadcast Encryption Schemes,*" 2001.

[11] Juan A. Garay, Jessica Staddon and Avishai Wool, "*Long-Lived Broadcast Encryption,*" In Crypto 2000, volume 1880 of Springer Lecture Notes in Computer Science, pp.333-352, 2000.

[12] Michel Abdalla, Yucal Shavitt and Avishai Wool, "*Towards Marking Broadcast Encryption Practical,*" IEEE/ACM Transactions on Networking, Vol.8, No.4, pp. 443-454, August, 2000.

[13] Yevgeniy Dodis and Nelly Fazio, "*Public Key Broadcast Encryption for Stateless Receivers,*" ACM Workshop on Digital Rights Management, 2002.

이 덕 규

e-mail : hbrhcdbr@sch.ac.kr

2001년 순천향대학교 컴퓨터공학과

2003년 순천향대학교 전산학과 석사

2003년~현재 순천향대학교 전산학과 박사과정

관심분야 : Broadcast Encryption, DRM, EKE

이 임 영

e-mail : imylee@sch.ac.kr

1981년 홍익대학교 전자공학과

1986년 오사카대학 통신공학전공 석사

1989년 오사카대학 통신공학전공 박사

1989년~1994년 한국전자통신연구원 선임연구원

1994년~현재 순천향대학교 정보기술공학부 부교수

관심분야 : 암호이론, 정보이론, 컴퓨터 보안