멀티캐스트 키 관리 기법을 적용한 브로드캐스트 암호화에 관한 연구

이덕규° 이임영
순천향대학교 정보기술공학부
{hbrhcdbr°, imylee}@sch.ac.kr

A Strudy on Broadcast Encryption Using Multicast Key Management

DeokGyu Lee° ImYeong Lee
Division of Information Technology Engineering, Soonchunhyang University

요 약

브로드캐스트 암호화 기법은 공개된 네트워크 상에서 멀티미디어, 소프트웨어, 유료 TV 등의 디지털 정보들을 전송하는데 적용되고 있다. 브로드캐스트 암호화 기법에서 중요한 것은 오직 사전에 허가된 사용자만이 디지털 정보를 얻을 수 있어야 한다는 것이다. 브로드캐스트 메시지가 전송되면 권한이 있는 사용자들은 자신이 사전에 부여받은 개인키를 이용하여 먼저 세션키를 복호화하고 이 세션키를 통하여 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 가입시에 효율적인 키 갱신이 필요하게 된다. 이에 본 논문에서는 기존에 서버가 단독으로 사용자를 예측하여 사용자에게 키를 분배하는 것이 아니고 초기 중심 서버가 키를 생성한 후 하부의 서버에 권한위임을 하면 하부 서버는 다시 사용자에게 키를 배포하는 방식으로 키를 생성 분배하도록 한다.

본 제안 방식은 기존의 서버 중심의 단독 키 분배보다 보다 효율적인 키 생성과 분배, 키 갱신을 하도록 제안하였다.

1. 서 론

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

키를 제공하는 방식 중에 하나인 공개키 방식은 세션키를 암호화하기 위한 그룹의 암호화키는 하나이고 이를 복호하기위한 키는 여러개의 무수히 많은 키를 이용함으로써 서버는 세션키를 암호화하고 각 사용자에게는 서로 다른 키를 이용하여 복호화 할수 있도록 되어 있다.

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

제안 방식에서는 빠른 키 갱신을 위한 키 갱신 인자를 첨가하고 이 인자를 통해 새로운 신규 가입자 혹은 탈퇴자가 발생하더라도 기존의 사용자에게 갱신값을 제공함으로써 쉽게 키 갱신이 간능해지도록 설계하였다. 또한 기존 서버가 단독으로 키를 생성하는 방식이 아니라 서버는 하부의 서버 키만 생성하고 다시 하부 서버가 사용자에게 키를 생성하는 방식을 제안함으로써 기존 방식보다 효율적으로 키 관리를 할 수 있도록 제안하였다.

본 논문은 Broadcast Encryption의 개요 중에서 적용방식에 대해 간략히 설명하고 제안방식의 각 단계에 관하여 살펴본다. 각 단계에 관한 프로토콜을 살펴본 마지막으로 결론으로써 끝을 맺도록 한다.

2. Broadcast Encryption 개요

2.1 적용 모델

브로드캐스트 암호화는 다음과 같이 2가지 모델을 기반으로 할 수 있다. 적용모델간의 차이점이 있지만 각각에 대하여 살펴보면 다음과 같다.

첫 번째 방식을 살펴보면, 사용자와 서버간의 정보를 이용하여 키를 생성/분배하는 방식이다. 다음은 기존의 멀티캐스트 방식과 유사하다. 이는 전송되는 방식에서 차이가 존재할 뿐 제공되는 메시지가 이전의 사용 그룹에 의해 결정되는 점에서 유사하다.

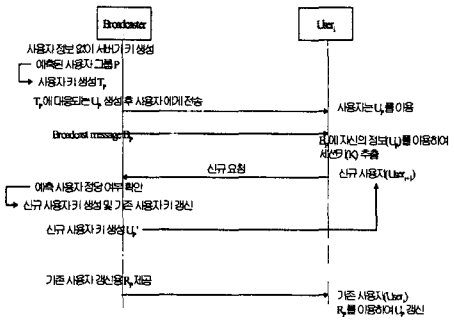

그림 1 적용 모델 2

키 생성과정에서 사용자가 참여하여야 하므로 생성시간에 사용자의 참여 시간이 포함될 수 있다. 키 갱신과정에서도 기존 사용자의 탈퇴/신규 사용자의 참여 시 키 갱신에 따른 소요시간이 많이 발생하게 된다.

위 방식과 다르게 서버가 키를 생성하는 방식으로 두 번째 적용 모델을 살펴볼 수 있다.

서버가 단독으로 참여할 사용자를 예측하여 키를 생성한다. 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.

하지만 두 방식 모두 서버가 사용자의 키를 모두 단독으로 생성하여 서버의 부담이 크다는 문제점을 가지고 있으며 서버

가 공격당하였을 경우 모든 키가 노출된다는 취약점을 가지고 있다. 이에 본 논문에서는 이러한 구조를 벗어나 서버가 하부 서버에 키를 생성/분배하고 다시 하부 서버가 사용자의 키를 생성/분배하는 방식을 제안한다.

3. 제안 방식

기존 서버가 단독으로 사용자의 키를 생성하고 분배하는 방식을 새로이 중앙 서버로부터 하부 서버로 키를 생성하고 다시 사용자에게 키를 생성하는 효율적인 키 관리 방식을 제안한다.

3.1 제안방식 개요

다음은 제안방식의 전체적인 개요에 대하여 살펴본다.

다음 그림은 본 제안방식에서의 전체적인 도식을 표현 한 것이다. 다음의 그림을 살펴보면 기존 멀티캐스트 키 관리 구조를 갖는 브로트캐스트 암호화 방법으로 중심 브로드캐스터가 하부 서버(라우터)에게 키를 생성하고 위임하면 다시 각 하부 서버는 키를 생성하여 하부 혹은 사용자에게 키를 분배하는 방식이다.

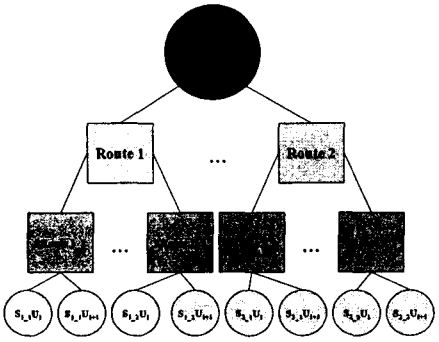

그림 2 제안 방식 전체 그림

본 제안 방식은 상위 정보를 포함하여 하부의 키를 생성하게 되는데 이러한 이유는 각 하부 서버는 자신이 생성한 키의 정당성을 보장받기 위해서이다. 또한 본 제안방식은 사용자가 탈퇴한다 하더라도 기존의 방식과 같이 전체의 키를 갱신할 필요없이 사용자가 속한 그룹의 전체 키만 갱신할 수 있기 때문에 키 갱신이 용이하다.

3.2 시스템 계수

다음은 본 방식에서 사용되는 시스템 계수를 기술한 것이다.

- p : 소수 ≥512 bit
- q : 소수 ≥160 bit (q | p−1)
- I : 개인키 생성을 위한 수
- e : 공개 암호화 키
- $d_1, \ldots, d_k$: 개별 복호화키 리스트
- M : 메시지 · S : 세션 키 · k : 사용자
- r_i : 랜덤 수 집합($r_i \in Z_p$) ($r_1, \ldots, r_k$)
- $h_i = g^{ri}$ · <y, $h_1, \ldots, h_k$> : 공개키
- $y = \prod h_i^{\,i}$ · $z = \prod h_i^{\,ri}$
- u_i : 랜덤수 ($u_i \in Z_q$) {$u_1, \ldots, u_k$}
- $d_i = B_i \cdot v_i^{(i)}$ ($v^{(i)} \in$:) · $i = v_1, \ldots, v_k$
- a: 랜덤 요소($a \in Z_q$)
- C: 방송 메시지(Broadcast message)
- $C = \langle M(\text{or } S)y^{aT}, h_1^a, \ldots, h_k^a \rangle = \langle B, H_1, \ldots, H_k \rangle$
- $B = M(\text{or } S)\, y^{aT}$ · $H_i = \prod h_i^a$
- T : 키 갱신을 위한 인자 ($t_1, \ldots, t_k \in Z_q$). $T = t_1 \cdot \ldots \cdot t_k$

3.3 프로토콜

(1) 키 생성 및 분배 단계

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

Step 1. 서버는 하부 서버들을 예측하여 이를 바탕으로 열을 랜덤하게 선택한다.

$i = 1, \ldots, k$ 예측 ⇒ r_i열 선택

Step 2. 이 선택된 랜덤열을 바탕으로 공개키 작성에 필요한 값을 생성한다.

$h_i = g^{ri} \bmod q$ 계산

공개키 <y, $h_1, \ldots, h_k$>

갱신을 위해 T생성 : $T = t_1 \cdot \ldots \cdot t_k$

Step 3. 생성된 값 h를 이용하여 공개키를 작성한 후 이를 바탕으로 개인키를 계산한다.

$B_i = (\sum r_i u_i t_i)/(\sum r_i v_i) \bmod q$

Step 4. 생성된 개인키 d_i를 하부 서버에게 전송한다.

$d_i = B_i v_i$

Step 5. 하부 서버는 전송받은 d_i에서 위임 키 생성을 위해 B_i를 획득한다.

$d_i = B_i v_i \,/\, v_i$

Step 6. 하부 서버는 사용자에게 전송할 키를 생성하기 위해 Step 1과 동일한 방식을 진행한다.

Step 7. 선택된 랜덤열과 자신의 개인키를 방탕으로 공개키 작성에 필요한 값을 생성한다.

$di = \Xi$, $O_i = g^{ri} \bmod q$ 계산

공개키 <z, $O_1, \ldots, O_k$>

갱신을 위해 T생성 : $T = t_1 \cdot \ldots \cdot t_k$

Step 8. 생성된 값 o를 이용하여 공개키를 작성한 후 이를 바탕으로 개인키를 계산한다.

$OB_i = (\sum \Xi r_i u_i t_i)/(\sum r_i v_i) \bmod q$

Step 9. 생성된 개인키를 Di를 사용자에게 전송한다.

$D_i = OB_i v_i$

Step 10. 사용자는 Di에서 개인키 OB_i를 획득한다.

$OB_i = OB_i v_i \,/\, v_i$

(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. 메시지 M 혹은 세션키 S를 암호화하여 계산하다.

Step 2. 랜덤 요소 a를 선택하고 키 갱신 요소 T를 연산하여 랜덤요소와 갱신요소를 같이 메시지 작성에 사용한다.

Step 3-1. 브로드캐스트 메시지를 작성하여 하부 서버에 전송한다.

$C = \langle M(\text{or } S)y^{aT}, h_1^a, ., h_k^a \rangle$

Step 3-2. 전송받은 하부 서버는 직접 자신의 메시지를 획득하던가(Step 4.) 하부 사용자에게 새로운 메시지 전송을 위해서 브로드캐스트 메시지를 작성한다.

$C' = \langle (z^{T v_i^{-1}})^{-1}, M(\text{or } S)y^{aT}, h_1^a, ., h_k^a \rangle$

Step 3-3. 하부 서버 단독으로 브로드캐스트 메시지를 작성하여 전송한다.

$CH = \langle M(\text{or } S)z^{aT}, o_1^a, ., o_k^a \rangle$

Step 4-1. 전송받은 하부 서버는 메시지는 개인키를 이용하여 메시지 M이나 세션키 S를 획득한다.

브로드캐스터 밑의 서버: $M(\text{or } S) = B/U^{i}$, $U = \prod H_j^{vj}$

$U^{ti} = (\prod H_j^{vj})^{ti} = (\prod g^{arjvj})^{ti} = (g^{rjvj})^{ati} = (h_j^{riT})^a = y^{aT}$

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

Step 4-2. 최종 사용자는 브로드캐스터가 발송한 브로드캐스트 메시지 C'를 하부 서버로부터 획득하면 자신의 개인키를 이용하여 메시지 M이나 세션키 S를 획득한다.

서버가 하부 서버네 전송하는 메시지 복호화: B_i 획득과정

$B_i=(z^{T\cdot i\ -1})^{-1\cdot o\cdot i}=(g^{ri-1\ \cdot\ i-1ti-1\ -1\cdot i-1})^{o\cdot i}=(g^{ri-1\cdot\ \cdot i-1ti-1\ -1\cdot i-1})^{ri\cdot iti\cdot i\cdot i/ri\cdot i}$

$M(or\ S)=B/U^{i}$, $U=\prod H_j^{\cdot ij}$

$U^{i}=(\prod H_i^{\cdot ij})^{i}=(\prod g^{ar/\cdot ij})^{i}=(g^{ri\cdot ij})^{i\cdot ia}=(g^{ri\ i})^{a}=(h_i^{\cdot iT})^{a}=y^{aT}$

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

Step 4-3. 하부 서버가 자신의 그룹에 대해 브로드캐스트 메시지를 발송한 경우 자신의 키를 이용하여 메시지 M이나 세션키 S를 획득한다.

브로드캐스터 밑의 서버: $M(or\ S)=B/U^{o\cdot i}$, $U=\prod O_i^{\cdot ij}$

$U^{o\cdot i}=(\prod O_i^{\cdot ij})^{i}=(\prod g^{a\ ri\cdot ij})^{i}=(g^{\cdot ri\cdot ij})^{i\cdot ia}=(g^{ri\ i})^{a}=(O_i^{\cdot iT})^{a}=z^{aT}$

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

(3) 키 갱신 단계(브로드캐스터에서의 갱신 과정)

하부 서버의 탈퇴 혹은 신규 가입자가 발생한 경우 다음과 같이 브로트캐스터와 하부 서버에서 키 갱신 과정을 거친다.

Step 1. 하부 서버 i가 탈퇴를 요청

Step 2. 서버는 기존 하부 서버의 개인키를 갱신하기 위해 갱신요소인 T에서 사용자 i의 갱신요소를 제거한다.

Step 3. 제거한 후 개인키를 갱신하고 하부서버에게 전송한다.

$B_i\cdot v^{(i)}\cdot t_i^{-1}=d_i'$

Step 4. 갱신된 키를 이용하여 하부 서버들은 브로드캐스트 메시지를 전송받고 다음과 같이 암호화된 메시지를 복호화하여 메시지를 획득하게 된다.

$(C=<B, H_1,\cdots,H_k>)=(C=<M(or\ S)\cdot y^{aTti-1},h_1^{a},\cdots,h_k^{a}>)^{ti}$ 계산

$M(or\ S)=B/U^{\cdot iti-1}$, $U=\prod H_j^{\cdot ij}$

$U^{\cdot iti-1}=(\prod H_j^{\cdot ij})^{\cdot iti-1}=(\prod g^{ar/\cdot ij})^{\cdot iti-1}=(g^{ri\cdot ij})^{\cdot iati-1}=(g^{ri\cdot iti})^{ati-1}=(H_j^{\cdot iti})^{a}=y^{aTti-1}$

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

(4) 키 갱신 단계(하부 서버에서의 갱신 과정)

하부 서버의 탈퇴 혹은 신규 가입자가 발생한 경우 다음과 같이 브로트캐스터와 하부 서버에서 키 갱신 과정을 거친다.

Step 1. 사용자 i가 탈퇴를 요청

Step 2. 서버는 기존 사용자의 개인키를 갱신하기 위해 갱신요소인 T에서 사용자 i의 갱신요소를 제거한다.

Step 3. 제거한 후 개인키를 갱신하고 사용자에게 전송한다.

$OB_i\cdot v^{(i)}\cdot t_i^{-1}=D_i'$

Step 4. 갱신된 키를 이용하여 사용자들은 브로드캐스트 메시지를 전송받고 다음과 같이 암호화된 메시지를 복호화하여 메시지를 획득하게 된다.

$(C=<B, H_1,\cdots,H_k>)=(C=<M(or\ S)\cdot y^{aTti-1},h_1^{a},\cdots,h_k^{a}>)^{Oti}$ 계산

$M(or\ S)=B/U^{Oti\cdot ti-1}$, $U=\prod H_j^{\cdot ij}$

$U^{Oti\cdot ti-1}=(\prod H_j^{\cdot ij})^{O\cdot iti-1}=(\prod g^{ar/\cdot ij})^{O\cdot iti-1}=(g^{ri\cdot ij})^{O\cdot iati-1}=(g^{ri\cdot iti})^{ati-1}=(H_j^{\cdot iti})^{a}=y^{aTti-1}$

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

4. 제안방식 고찰

본 방식은 다음과 같은 특징을 갖도록 제안하였다. 하부의 사용자 탈퇴가 전체 키 구조에 영향을 주지 않는다는 것이다. 이것은 하부 사용자가 탈퇴하더라도 전체키를 갱신하거나 교체되는 일은 방생되지 않는다는 것이다. 또한 서버가 추가되더라도 사전에 예측한 키로부터 키를 생성하기 때문에 다른 서버나 사용자에게 영향을 미치지 않는다.

본 제안 방식은 여러 응용 서비스에 이용될 수 있을 것이다. 현재의 멀티미디어 서비스를 제공할 경우 한 채널을 통해 음성, 영상, 동영상 등을 제공하였지만 제안 방식의 경우 각각의 채널을 통해 각각의 서비스가 가능할 것이다. 이것은 동일 서비스에 동일키를 적용하는 것이 유용할 것이다. 방송국과 같은 전체 서비스를 하는 곳에서 유용하게 사용될 수 있을 것이다.

마지막으로 키 갱신이 쉽게 이뤄질 수 있을 것이다. 기존의 방식처럼 서버가 단독으로 사용자 전체를 관리하는 것이 아니고 각 하부의 키만 관리하기 때문에 키 관리가 용이하게 될 것이며, 각 서버가 사용자를 담당하고 키를 갱신함으로써 기존 방식에 비해 빠른 키 갱신을 이룰 수 있을 것이다.

5. 결론

브로드캐스트 암호화는 공개된 네트워크 상에서 인가된 사용자에게만 컨텐츠를 제공하는데 사용한다. 인가된 사용자이외에는 브로드캐스트되는 메시지에 대해 아무런 정보를 얻어낼 수 없으며, 인가된 사용자는 사전에 전송된 개인키를 이용하여 세션키를 취득할 수 있게 된다.

본 논문은 기존 서버를 통한 키의 생성이 아니 각 하부 그룹으로 묶은 다음 키를 생성하는 방식을 제안하였다. 제안 방식은 키 생성에서 서버의 부담을 줄였을 뿐만 아니라 키 갱신에서도 서버가 전체적으로 갱신하는 방식이 아닌 각 하부 구조가 키 갱신을 이루도록 하여 빠른 키 갱신을 제안하고 있다. 이후 연구는 최초 키를 받은 서버를 벗어나 다른 서버로 이동하였을 경우 사용자 새로운 키 분배 없이 이전 키의 계속적인 사용에 관한 연구가 필요하리라 본다.

참고 문헌

[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 Capability", KoreaCrypto 02', 2003