멀티캐스트 키 관리 시스템에서 서버 계산 비용 향상을 위한 해쉬 일괄 키 갱신 기법

이규원*, 이수연**, 박창섭*
*단국대학교 전자계산학과
**백석대학 컴퓨터학부
*e-mail:cool2527@cs.dankook.ac.kr

Hash Batch Rekeying Scheme for Reducing the Server Cost in Multicast Key Management System

Gyu-Won Lee*, Su-Youn Lee**, Chang-Seop Park*
*Dept of Computer Science, Dankook University
**Dept of Computer, Baekseok College

요 약

많은 인터넷 응용 프로그램들은 멀티캐스트 기반의 그룹 통신에 기반을 두고 있으며, 사용자들의 빈번한 가입과 탈퇴에 따른 효율적인 키 관리를 필요로 하게 된다. 본 논문에서는 기존의 개별 키 갱신 및 일괄 키 갱신 기법보다 서버의 계산 비용을 줄이는 일방향 해쉬 함수와 그룹별 해쉬 함수를 적용한 일괄 키 갱신 기법을 제안한다. 제안 방식과 기존 방식의 average case 값을 계산하여 서버의 계산 비용을 분석한 결과 제안한 방식이 효율적임을 나타내었다.

1. 서론

최근 많은 응용 프로그램들은 음성 채팅, 원격 화상 회의, 유료 영상 서비스 등 그룹 통신을 위한 멀티캐스트에 기반을 두고 있으며, 그 서비스들은 점차 다양해지고 있다. 안전한 그룹 통신을 위해서는 메시지의 기밀성, 인증, 비밀성을 제공해야 하는데, 이를 위해서 그룹 사용자들과 키 서버(key server)만이 그룹 키(group key)를 공유하게 된다. 그룹 사용자들의 가입과 탈퇴가 빈번하게 이루어지는 통신상에서 새로운 가입자는 가입 이전의 그룹 통신에 접근할 수 없도록 하고(backward access control), 탈퇴자는 향후 그룹 통신을 이용할 수 없도록(forward access control) 하기 위해서 그룹 키 관리 시스템에서는 그룹 사용자들에게 그룹 키를 변경해서 각 사용자들에게 키 갱신 메시지(rekey message)를 전송해 주어야 한다.

개별 키 갱신 기법(individual rekeying)의 경우 키 갱신 메시지는 인증을 목적으로 서명을 이용하나 서명 연산이 계산적으로 복잡하므로 서버의 계산 비용 낭비를 초래할 수 있다. 이러한 서버의 계산 비용을 줄이기 위한 방법으로 일괄 키 갱신(batch rekeying) 기법이 소개되었다. 일괄 키 갱신 기법은 일정기한 동안 탈퇴 요청과 가입 요청을 수집한 후에 일정기한이 지나고 나서 새로운 키들을 생성하고, 키 갱신 메시지를 생성한 후 그룹의 사용자들에게 멀티캐스트 한다는 측면에서 개별 키 갱신 기법에 비해 보다 나은 방법이라 할 수 있다. 그러나 일괄 키 갱신 기법에서 rekey subtree(키 갱신 노드들의 구성)의 각 노드는 그 노드의 자식노드의 키 값으로 암호화해서 전달하기 때문에 rekey subtree의 각 노드는 두 번의(degree=2) 암호화 과정이 필요하게 되어 서버의 계산 비용이 커진다. 따라서 본 논문에서는 서버의 계산 비용을 줄이는 해쉬 일괄 키 갱신 기법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 멀티캐스트 키 관리를 소개한다. 3장에서는 기존의 키 갱신 기법들을 소개하고, 4장에서는 해쉬 일괄 키 갱신 기법

을 제안한다. 5장에서는 제안된 해쉬 일괄 키 갱신 기법과 기존 기법의 서버 계산 비용을 비교 분석한다.

2. 멀티캐스트 키 관리

그룹 키 관리 시스템은 등록(registration), 키 관리(key management), 키 갱신 메시지 전송(rekey sender)의 세 가지 기능으로 분류된다. 사용자가 그룹 생성 또는 가입을 원할 경우 등록은 인증 과정을 통해 이루어진다. 사용자가 인증 되어 그룹의 사용자로 등록되면, 새로운 사용자에 대한 식별자(identifier)와 개별 키(personal key)를 사용자에게 전달하게 된다. 이와 동시에 키 관리 부분에 새로운 사용자의 리스트와 개별 키를 전달하여 사용자들의 키를 관리한다. 인증된 사용자의 가입이나 탈퇴 요청이 있을 경우에 사용자의 개인키로 암호화해서 키 관리 부분에 보내게 되면, 키 관리 부분은 그 요청이 개인키로 맞게 암호화 되었는지 확인하여 요청을 받아들인다. 그 후, 키 갱신 메시지 전송은 키 갱신 메시지를 모든 그룹 사용자들에게 전달하여 그룹 사용자들만이 서비스 받을 수 있도록 한다.

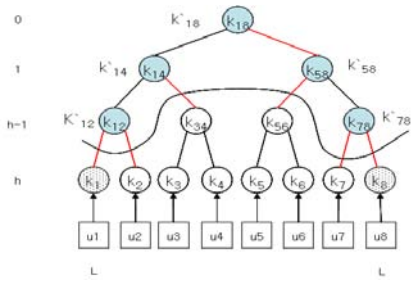
3. 기존의 키 갱신 기법

3.1 개별 키 갱신 기법

이 기법은 1998년 D.Wallner에 의해서 제안되었다[1]. 사용자가 그룹에서 탈퇴를 원하거나, 새로운 사용자가 가입을 원할 때 가능한 빨리 요청을 처리해 주어야 한다. 그러므로 키 서버는 사용자의 가입이나 탈퇴 요청을 받아들인 즉시 처리해주고 사용자가 공유하고 있던 키를 변경하는 것을 개별 키 갱신이라고 한다. 그러나 개별 키 갱신은 두 가지 이유로 인해서 비효율적이다. 먼저 키 갱신 메시지는 등록시 인증 기법으로 전자서명(digital signature)을 이용한다. 서명 연산은 계산적으로 복잡하기 때문에 만약 매 시간 하나의 요청(가입/탈퇴)이 발생하여 키 서버가 새로운 키를 생성하고, 그것을 서명을 이용해서 키 갱신 메시지를 전송하게 된다면, 요청이 빈번하게 일어나는 경우 키 서버의 서명 연산에 대한 계산 비용이 높아지게 된다. 두 번째, 키와 메시지간의 동기화 문제이다. 만약 두 개의 탈퇴 요청이 연달아 일어나는 경우 첫 번째 키 갱신 메시지가 전달된 상황에서 보조키와 그룹 키를 전혀 사용해보지도 못하고 두 번째 키 갱신 메시지를 전달해야 한다. 이것은 서버 계산 비용의 낭비를 초래하게 되는 문제점이 있다.

3.2 일괄 키 갱신 기법

개별 키 갱신의 문제점을 보완하기 위해서 2001년 Simon S.Lam가 일괄 키 갱신을 고안했다[2]. 이 기법은 일정기한(rekey interval)동안 탈퇴 요청과 가입 요청을 수집한 후에 그 일정기한이 지나고 나서 새로운 키들을 생성하고, 키 갱신 메시지를 만들어 그룹의 사용자들에게 전송한다. [그림 1]과 같이 사용자 u1과 u8이 탈퇴할 경우 탈퇴 노드부터 루트 노드까지의 모든 경로상의 노드를 갱신해야 하며, 이 갱신된 노드들을 구성하는 트리를 rekey subtree라 한다. rekey subtree의 모든 키 노드에 대해서 자식의 키로 암호화하여 할당함으로써 모든 rekey subtree안의 키 노드에 대해서 두 번씩의 암호화가 필요하다.



〔그림 1〕 일괄 키 갱신

4. 제안된 해쉬 일괄 키 갱신 기법

일괄 키 갱신 기법에서 rekey subtree의 각 노드는 그 노드의 자식노드의 키 값으로 암호화해서 전달한다. 그러므로 rekey subtree의 각 노드는 두 번씩의 암호화 과정이 필요하다. rekey subtree의 크기가 매우 클 경우, 각 노드에 대해 두 번씩의 암호화가 서버의 계산 비용을 높인다. 따라서 MDC(Manipulation Detection Code) 해쉬 함수의 일방향 해쉬 함수를 적용해서 rekey subtree의 말단 노드들을 제외한 나머지 노드들은 한 번의 암호화만 적용되는 기법과 MAC(Message Authentication Code) 해쉬 함수를 적용해서 그룹별 암호화하는 기법을 제안한다. 본 논문에서는 일괄적으로 탈퇴(leave)가 발생하는 조건만을 다룬다.

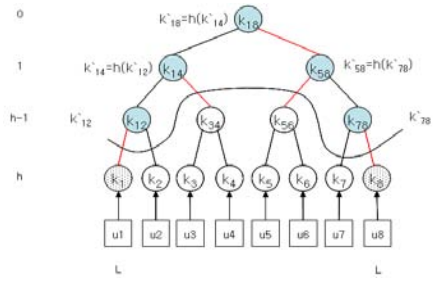
4.1 암호화 해쉬 함수

암호학적 해쉬 함수에는 MAC와 MDC 해쉬 함수가 있고, MDC 해쉬 함수에는 일방향 해쉬 함수와 충돌회피 해쉬 함수가 있다. MAC 해쉬 함수에서는 입력자료 m과 비밀키 k가 해쉬 값 h(k, m)를 생성하는데 사용되는 반면에, MDC 해쉬 함수에서는 해쉬 값 h(m) 생성에 단지 입력자료 m만이 사용되는 특성이 있다. 이러한 특

성을 이용하여 본 논문에서는 MAC 해쉬 함수 및 MDC 해쉬 함수의 일방향 해쉬 함수를 적용한다.

4.2 일방향 해쉬 일괄 키 갱신 기법 제안

일괄 키 갱신 기법의 서버 계산비용을 줄이기 위해 MDC 해쉬 함수의 일방향 해쉬 함수를 적용하면 rekey subtree의 각 노드들은 한 번씩의 암호화만 하면 된다.

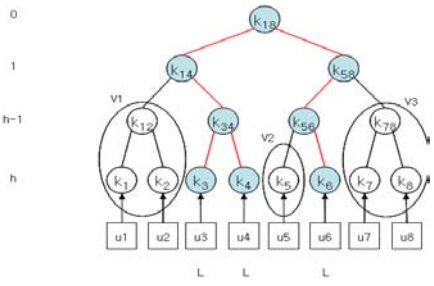


〔그림 2〕 일방향 해쉬 일괄 키 갱신

[그림 2]에서처럼 사용자 u1이 탈퇴했을 때 k_{12}를 새로운 키 값 $k`_{12}$로 바꾼 후 $k_2$로 암호화 하여 u2에게 전송한다. 그러면, $k`_{12}$를 전달받은 u2는 $k`_{12}$에 한 번의 해쉬 함수를 적용시켜 사용자 u2, u4가 공유한 보조키 값 $h(k`_{12})$을 얻고, 또 이 $h(k`_{12})$에 한 번의 해쉬를 더 적용해서 그룹 키인 $h(h(k`_{12}))$를 얻는다. 사용자 u3, u4의 경우 그들이 공유한 보조키 k_{34}로 $k`_{14}$을 암호화하여 전송한다. 또한 사용자 u8이 탈퇴할 경우 $k`_{78}$을 k_7로 암호화하여 u7에게 전송하고, 사용자 u5, u6의 경우 그들이 공유한 보조키 k_{56}으로 $k`_{58}$을 암호화하여 전송한다. 마지막으로 $k`_{18}$을 $k`_{58}$로 암호화한다. 이와 같이 rekey subtree의 노드들은 모두 자식 노드로 한 번씩의 암호화 과정을 갖는다. 일괄 키 갱신에서 rekey subtree의 각 노드에 대해서 두 번씩의 암호화 과정을 거치는 경우에 비해 키 서버의 계산 비용이 낮아진다는 것을 알 수 있다.

4.3 그룹 해쉬 일괄 키 갱신 기법 제안

다음으로 MAC 해쉬 함수를 적용한 그룹 해쉬 일괄 키 기법을 제안한다.



〔그림 3〕 그룹 해쉬 일괄 키 갱신

[그림 3]와 같이 사용자 u3, u4, u6가 탈퇴한 경우 V1, V2, V3의 각 사용자 그룹들은 새로운 키 값(rekey)을 전달 받는다. 예를 들어 V1의 사용자 u1, u2의 경우 그들의 공통키인 k_{12}로 새로운 키 값을 암호화한 값 $Ek_{12}(k)$을 전송 받는다. 그러면 사용자 u1, u2는 그들의 공통키인 k_{12}로 복호화($D(Ek_{12}(k))$)하여 k값을 얻는다. 이후 k_{14}와 k값을 해쉬 적용하여 $k`_{14}=h(k_{14}, k)$을 얻고, $k_{18}$과 k값을 해쉬 적용하여 그룹 키인 $k`_{18}$을 얻어낸다. V2, V3역시 같은 과정을 갖는다. k_{18}의 경우 왼쪽과 오른쪽 자식의 서브트리에서 적어도 한 개 이상의 탈퇴가 발생했음을 알 수 있다. 또한 V1, V2, V3의 경우 어떠한 탈퇴도 발생하지 않았음을 알 수 있다. 하지만 k_{14}, k_{56}, k_{58}과 같이 한 쪽은 탈퇴가 발생하고, 다른 한 쪽은 탈퇴가 발생하지 않을 경우 탈퇴가 일어나지 않은 노드에 대해 한 번의 암호화를 하게 된다. 이와 같이 그룹별로 암호화를 하기 때문에 기존의 방식보다 암호화 횟수가 줄어든다. 성능 분석 비교를 통하여 이를 나타내었다.

5. 성능 분석 비교

5.1 Average case 분석

서버의 계산 비용(server cost)은 rekey subtree에 속하는 노드의 수와 탈퇴가 발생하지 않는 그룹의 수와 자식 노드의 수에 달려있다. 그래서 average case 분석을 위한 방법으로 각 노드가 rekey subtree에 속할 확률과 탈퇴가 발생하지 않는 그룹에 속할 확률, 그리고 예상되는 자식노드 수를 고려해 본다. 세 가지 경우에 대해서 분석하였다.

(1) 일괄 키 갱신

키 트리의 레벨 0에 있는 노드를 루트라 하고, 레벨 h에 있는 노드를 말단 노드라 하자. (단, $h = \log_2 N$) T(v)는 v노드가 루트인 subtree이고, L(v)를 T(v)의 말단 노드들의 집합이라 하면 L(v)의 사이즈는 $N/2^k$이 된다. (단, $0 \le k \le h-1$)

v가 rekey subtree에 속한다는 것은 L(v)에 적어도 하나의 탈퇴가 존재한다는 걸 의미한다. 현재 모든 사용자가 탈퇴할 확률이 같다면, N명의 사용자 중에서 L명이 탈퇴할 확률은 $\binom{N}{L}$이 되고, $\binom{N-N/2^k}{N}$이 L(v)에 속하지 않을 확률이 된다. 따라서 v가 rekey subtree에 속할 확률은 1-

$\frac{\binom{N-N/2^k}{N}}{\binom{N}{L}}$ 이 된다. 이때, 레벨 0부터 레벨 h-1까지의 각 노드가 변경이 된다면, 그 노드들은 두 번의 암호화를 필요하게 된다. 그러므로 사용자의 탈퇴(L)에 따른 서버의 계산 비용 즉, 암호화(E) 횟수 값 E(L)은 (식1)과 같다.

$$E(L) = 2 \cdot \sum_{k=0}^{h-1} \cdot 2^k \cdot \left(1 - \frac{\binom{N-N/2^k}{N}}{\binom{N}{L}}\right) \quad (식1)$$

(2) 일방향 해쉬 일괄 키 갱신

일방향 해쉬 일괄 키 갱신의 경우 rekey subtree의 각 노드들은 한 번의 암호화를 필요하게 된다. 그러므로 서버의 계산 비용의 값은 (식2)와 같다.

$$E(L) = 1 \cdot \sum_{k=0}^{h-1} \cdot 2^k \cdot \left(1 - \frac{\binom{N-N/2^k}{N}}{\binom{N}{L}}\right) \quad (식2)$$

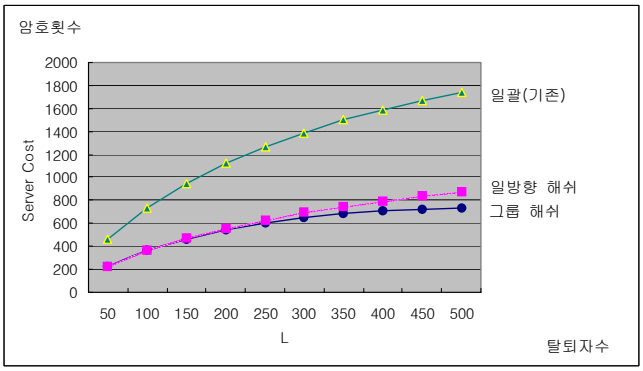
(3) 그룹 해쉬 일괄 키 갱신

트리에 대해 모두 탈퇴가 발생할 경우와 모두 탈퇴가 발생하지 않을 경우를 구한 후 1에서 빼주면 암호화한 개수가 나온다. 따라서 서버의 계산 비용 E(L)의 값을 구하면 (식3)과 같다.

$$E(L) = 1 \cdot \sum_{k=0}^{h-1} \cdot 2^k \cdot \left(1 - \frac{\binom{L}{N/2^k}}{\binom{N}{N/2^k}} + \frac{\binom{N-N/2^k}{L}}{\binom{N}{L}}\right) \quad (식3)$$

5.2 서버 계산 비용 비교

각 키 갱신 기법의 서버 계산 비용을 그룹의 사용자가 1024명 일 때, 가입과 탈퇴자가 1명부터 500명까지를 범위로 하여 그래프로 나타내어 비교한다.



[그림 4] 평균의 경우 서버 계산 비용

[그림 4]와 같이 일괄 키 갱신 기법의 암호화 횟수와 비교하여 일방향 해쉬 일괄 키 갱신 기법의 암호화 횟수와 그룹 해쉬 일괄 키 갱신 기법의 암호화 횟수가 절반 가까이 줄어든다는 것을 알 수 있다. 이와 같이, 기존의 일괄 키 갱신 기법보다 본 논문에서 제안한 해쉬 일괄 키 갱신 기법이 보다 효율적임을 알 수 있다.

6. 결론

본 논문에서는 그룹 기반의 멀티캐스트 환경에서 효율적인 키 관리를 위해 해쉬 함수를 적용한 키 갱신 기법을 제안하였다. 기존의 일괄 키 갱신 기법에 MDC 해쉬 함수의 일방향 해쉬 함수를 적용하였고, MAC 해쉬 함수를 적용하여 그룹 해쉬 일괄 키 갱신 기법을 제안함으로써 서버의 계산 비용을 줄이는 목적을 달성했다. 이와 같이, 기존의 일괄 키 갱신 기법보다 본 논문에서 제안한 해쉬 일괄 키 갱신 기법이 서버의 계산 비용을 최소화하고 멀티캐스트 키 관리에 응용될 수 있다고 본다.

참고문헌

[1] D. Wallner E. Harder, and Ryan Agee. Key Management for Multicast : Issues and Architectures, INTERNET-DRAFT

[2] Batch Rekeying for Secure Group Communications

[3] A. Ballardie. Scalable Multicast Key Distribution, RFC 1949

[4] D. Balenson, D. McGrew, and A. Sherman, Key Managemnt for Large Dynamic Groups : One-way Function Trees and Amortized Initialization, INTERNET-DRAFT

[5] S. Setia, S. Koussih, S. jajodia, and E. Harder, "Kronos : A scalable group re-keying approach for secure multicast", in Proceedings of IEEE Symposium on Security and Privacy

[6] Chung Kei Wong and Simon S. Lam. Keyston : a group key management system. In International Conference on Telecommunications, Acapulco, Mexico

[7] X. S. Li, Y. R. Yang, M. G. Gouda, and S. S. Lam, Batch rekeying for secure group communications, in Proceedings of Tenth International World Wide Web Conference