情報保護學會論文誌
第15卷 第6號, 2005. 12

멀티캐스트 일괄 키 갱신 기법의 서버계산 비용 분석*

이 규 원,‡ 박 창 섭†
단국대학교

Analysis of Server's Computational Cost for Multicast Batch Rekeying Scheme*

Gyu-Won Lee,‡ Chang-Seop Park†
Dankook University

요 약

향후 다양한 인터넷 응용 프로그램들은 멀티캐스트 그룹 통신에 기반을 두게 될 것이며, 따라서 그룹 멤버들의 빈번한 가입과 탈퇴를 효율적으로 대처하기 위한 그룹키 관리기법이 요구된다. 본 논문에서는 기존의 개별 키 갱신기법들을 일괄 키 갱신 기법으로 확장하여 제안하고, 기존의 기법들과 제안된 기법을 키 서버에 의해 수행되는 암호화 및 일방향 해쉬 함수의 횟수 그리고 멀티캐스트 메시지의 크기 측면에서 성능을 비교 분석한다. 비교 분석에 있어서는, 다중 탈퇴자가 존재하는 상황에서 그들에 의해 초래되는 키 갱신 비용을 확률론적인 접근법을 기반으로 평균치를 계산하였다.

ABSTRACT

In the near future, various aplications on the Internet will be based on the multicast group communication, so that the efficient group key management is essential for managing the frequent group join and leave events. In this paper, we propose several batch rekeying schemes extended from conventional individual rekeying schemes, and analyze the efficiencies of them in terms of both the number of encryption and one-way hash function as well as multicast message size. Considering multiple member leaves, a probabilistic approach is need to compute the average computational amounts for rekeying.

Keywords : *one-way hash fuction, XOR, ECSM+*

Ⅰ. 서 론

향후 다양한 인터넷 응용 프로그램들은 음성 채팅, 원격 화상 회의, 유료 영상 서비스 등 그룹 통신을 위한 멀티캐스트에 기반을 두게 될 것이며, 그 서비스들은 점차 다양해질 것이다.[1] 안전한 그룹 통신을 위해서는 메시지의 기밀성(confidentiality), 인증(authentication), 무결성(integrity)등을 제공해야 하는데, 이를 위해서 그룹 멤버(group member)들과 키 서버(key server)만이 그룹키(group key)를 공유하게 된다. 그룹 멤버들의 탈퇴가 빈번하게 이루어지는 환경하에서 새로운 가입자는 이전의 그룹 통신에 접근할 수 없도록 하고(backward secrecy), 탈퇴자는 향후 그룹 통신을 이용할 수 없도록(forward secrecy) 하기 위해서, 그룹키 관리 시스템에서는 그룹키를 변경하고, 그 변경된 키 정보를 각

접수일 : 2005년 9월 7일 ; 채택일 : 2005년 11월 23일
* 본 연구는 2003학년도 한국과학재단 과제(R05-2003-000-11411-0)에 의해 지원되었습니다.
† 주저자 : csp0@dankook.ac.kr
‡ 교신저자 : cool2527@hanmail.net

멤버들에게 키 갱신 메시지(rekey message)를 통해서 전송해 주어야 한다.

그룹키 갱신기법은 두 가지 측면에서 다음과 같이 분류되어진다. 첫째는 그룹 멤버들이 전송되는 모든 키 갱신 메시지를 빠짐없이 수신해야 하느냐에 따른 분류이다. Stateful 키 갱신의 경우는 모든 그룹 멤버들이 그룹 멤버쉽(group membership)에 변화가 있을 때 마다 발생되는 모든 키 갱신 메시지의 수신을 요구하는 기법이다. 반면에, Stateless 키 갱신은 모든 키 갱신 메시지의 수신을 요구하고 있지는 않지만, 그룹에 참여할 수 있는 멤버들의 최대 규모가 사전에 정의가 되기 때문에 확장성 측면에서는 제한적인 기법이다. 둘째는 그룹 멤버의 탈퇴 요청이 있을 때마다 요청을 즉시 처리해주느냐에 따른 분류이다. 개별 키 갱신(individual rekeying)기법은 그룹 멤버쉽 변화에 따른 키 갱신 메시지를 즉시 처리해 주는 기법이고, 일괄 키 갱신(batch rekeying) 기법은 일정기간 동안 발생된 멤버쉽 변화를 수합하여 일괄적으로 처리해 주는 기법이다. 개별 키 갱신 기법의 경우는 키 트리상에서 인접한 위치에 있는 그룹 멤버들이 가입 또는 탈퇴를 한다면 키 서버 및 기존 그룹 멤버들은 그룹키 갱신에 중복적인 계산을 요구한다는 측면에서 비효율적이다. 또한, 그룹키 갱신 메시지와 응용 메시지의 전달이 병행해서 이루어지기 때문에, 메시지 전송지연에 따른 그룹키 불일치가 발생할 가능성이 높아 그룹 멤버들의 입장에서는 여러개의 이전 그룹키를 유지, 관리하고 있어야 하는 부담이 발생한다. 반면에, 일괄 키 갱신 기법은 일정기한 동안 탈퇴 요청과 가입 요청을 수집한 후에 새로운 키들을 생성하고, 키 갱신 메시지를 그룹 멤버들에게 멀티캐스트 한다는 측면에서 개별 키 갱신 기법에 비해 개선된 방법이라 할 수 있다.

본 논문에서는 기존에 제시된 개별 키 갱신기법[2, 3]과 Stateless 키 갱신기법[7]을 일괄 키 갱신기법으로 확장 전환시키고, 키 서버의 계산비용 측면에서의 성능평가를 시도한다. 가입에 따른 계산비용은 거의 모든 기법들이 대동소이하기 때문에, 여기서는 다중 탈퇴에 따른 경우에 국한시키고 각 기법들에 대한 키 서버의 계산비용을 확률론적인 접근방식을 통해 산출, 비교한다. 본 논문의 구성은 다음과 같다. 2장에서는 LKH[3, 10], ELKH[4], OFT[5, 9], EHBT[6], CSM[7], ELK[11], EMKM[12] 등의 관련 연구를 소개하고, 3장에서는 기존에 제시된 ELKH와 EHBT를 일괄키 갱신기법으로 접근하고, OFT를 일괄 키 갱신기법으로 확장한 EOFT와 Stateless 방식의 CSM을 Stateful 일괄 키 갱신 기법으로 전환한 ECSM과 ECSM+을 제안한다. 4장에서는 관련연구 및 본 논문에서 확장, 제안한 기법을 성능 분석하고 비교 평가한다. 그리고 5장에서 결론을 맺는다.

II. 관련 연구

가장 대표적인 개별키 갱신기법은 Wallner[3] 등에 의해 제안된 논리적인 키 계층(LKH : Logical Key Hierarchy)이다. 키 서버는 그룹키 갱신 및 분배를 위해 노드키(node key)들로 구성된 키 트리(key tree)를 유지한다. 트리에서 루트 노드는 그룹키(group key), 말단 노드는 개인키(personal key), 중간 노드를 보조키(auxiliary key)라 한다. 각 멤버는 자신에게 할당된 개인키로부터 루트의 그룹키에 이르는 경로상의 모든 키들이 할당된다. 말단 노드의 개수가 N인 이진 트리(binary tree)에서 각 멤버는 최대 $\log_2 N+1$개의 키를 저장한다. 만약 그룹에 탈퇴하는 멤버가 있다면, 탈퇴한 멤버의 부모 노드(parent node)로부터 루트 노드에 이르는 모든 키들은 전방기밀성(forward secrecy)을 유지하기 위하여 갱신되어야 하는데, 키 서버는 갱신된 노드키를 자식 노드(child node)의 키로 암호화한 키 갱신 메시지(rekeying message)를 생성하여 멀티캐스트 한다. 한 명의 멤버가 탈퇴할 경우 생성된 키 갱신 메시지에는 최대 $2 \cdot \log_2 N$개의 암호화 된 키를 포함한다. 또한, Steve[4] 등은 LKH를 일괄 키 갱신으로 확장한 ELKH를 제안하였다. 하지만, ELKH의 경우는 LKH를 단순 확장한 것이기 때문에 키 서버가 부담해야 할 계산량이 크다는 단점이 있다.

McGrew[5] 등에 의해 제안된 OFT(One-way Function Tree)에서는 키 갱신 메시지의 크기를 LKH에서의 $2 \cdot \log_2 N$을 $\log_2 N$으로 줄였다. 특히, 키 트리를 구성하는 각각의 노드키에 대해서 여기에 일방향 함수를 적용한 결과인 Blind 키가 정의된다. 각 멤버들은 자신의 말단 노드로부터 루트에 이르는 경로 상의 Unblind 키들을 알고 있고, 또한 해당 노드들의 형제 노드(sibiling node)에 해당하는 Blind 키들이 제공된다. 특정 말단 노드에 해당하는 멤버의 탈퇴시에는 해당 경로 상의 변경된 노드들의 Blind 키를 각 노드의 형제 노드의 Un-

blind 키로 암호화하여 전송하고, 기존 멤버들은 자신의 경로 상에 있는 노드의 Blind 키와 전달받은 형제 노드의 Blind 키를 혼합(mixing)하여 새로운 Unblind 키를 갱신한다. 이진 트리에 대해서 한 명의 멤버가 탈퇴할 경우 생성되는 키 갱신 메시지는 최대 $\log_2 N$개의 키를 포함한다.

Rafaeli[6] 등에 의해 제안된 EHBT(Efficient Hierarchical Binary Tree)는 Canetti[8]에 의해 제안된 해쉬 체인 기반의 개별 키 갱신기법의 확장이라 할 수 있다. 여기에서 그룹 멤버의 탈퇴로 인해 갱신되어질 경로상의 노드키들은 탈퇴한 멤버의 형제 노드의 키로부터 루트에 이르기까지 위로 발생되어진다. 이때, 키 서버는 경로상의 노드키들을 일방향 일방향 해쉬 함수와 XOR 연산을 하여 갱신하고, 갱신된 노드키들에 영향을 받는 그룹 멤버들에게는 그들의 공통된 상위 노드키로 암호화하여 전송한다. 이진 트리에 대해서 한명의 멤버가 탈퇴할 경우 생성된 멀티캐스트 메시지는 최대 $\log_2 N$의 크기를 갖는다.

Naor[7] 등에 의해서 제안된 CSM(Complete Subtree Method)은 Stateless 키 갱신기법 중 하나로써, 전체 그룹 멤버의 최대규모 N이 사전에 정의된 키 트리를 기반으로 다수의 멤버가 탈퇴시에 해당 말단 노드들을 제외한 subtree들을 구성하고, 해당 subtree의 루트키를 이용하여 새로운 그룹키를 갱신하는 방식이다. n명의 멤버가 동시에 탈퇴할 경우 생성되는 키 갱신 메시지는 최대 $n \cdot \log_2(N/n)$개의 키를 포함한다.

Perrig[11] 등에 의해 제안된 ELK(Efficient Large-Group Key)는 OFT와 유사한 기법으로 힌트(hint) 메시지의 개념을 도입하여 키 갱신 메시지의 길이를 줄이는 방안을 제안하였다. ELK에서는 새로운 멤버의 가입에 대해서는 키 갱신 메시지의 전송을 요구하지 않는다. 탈퇴의 경우에 키 갱신이 되어야 할 노드들은 각각 자식 노드의 키로 암호화해야 하므로 키 서버의 계산비용 측면에서는 LKH와 유사하거나 오히려 비효율적이다.

Ki[12] 등에 의해 제안된 EMKM(Efficient Multicast Key Management)은 CSM과 같이 Stateless Receiver들에 대한 키 갱신 기법 중 하나로써 멤버에 대한 개별키의 unicast 전송을 제외한 모든 키 갱신 메시지들이 암호화 없이 전송되는 기법이다. 즉, 키 갱신에 대한 모든 계산은 일방향 해쉬함수와 XOR만을 사용한다.

Ⅲ. 일괄 키 갱신기법 제안

이번 장에서는 기존에 제시된 ELKH와 EHBT를 일괄키 갱신기법으로 접근하고, OFT 개별키 갱신기법을 일괄 키 갱신기법으로 확장한 EOFT (Extended OFT)와 CSM을 Stateful 일괄 키 갱신기법으로 확장한 ECSM(Extended CSM)을 제안한다.

3.1 표기

본 논문에서의 표기는 다음과 같이 정의한다. N은 그룹 멤버의 수이고, 키 트리를 균형된 이진 트리로 가정한다. 이진 트리를 T와 같이 표시하고, 서브 트리(sub tree)를 $ST_k(k=1,2,3,..)$로 표시한다. 키 트리의 깊이는 $d = \log_2 N$이다. 키 트리를 구성하는 각 노드키들은 K_{ij}와 같이 나타내고, $i(i=$

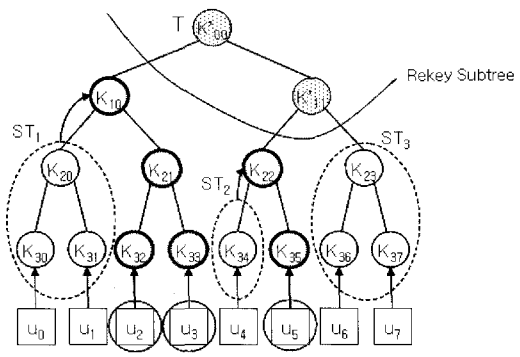


그림 1. 그룹 멤버 u2, u3, u5가 탈퇴할 때의 서브 트리의 이동

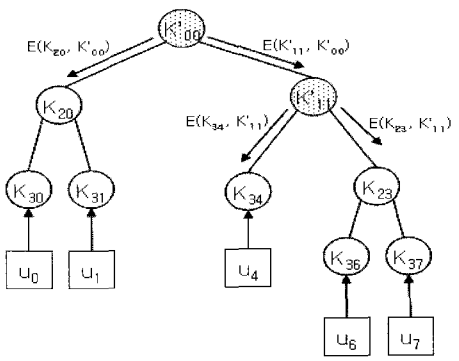


그림 2. 그룹 멤버 u2, u3, u5의 탈퇴 후 ELKH

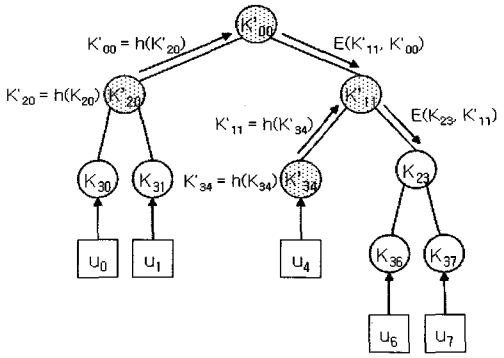


그림 3. 그룹 멤버 u2, u3, u5의 탈퇴 후 EHBT

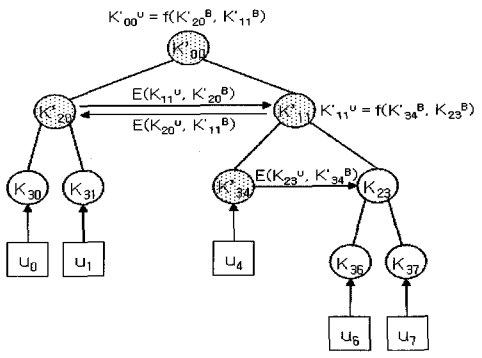


그림 4. 그룹 멤버 u2, u3, u5의 탈퇴 후 EOFT

0,1,2,..)는 트리의 깊이, $j(j=0,1,2,..)$는 해당 깊이에서 왼쪽에서부터의 노드키의 위치를 의미한다. K_{ij}는 갱신 이전의 키 값이고, 키 갱신이 발생한 노드들은 K'_{ij}와 같이 표시한다. k는 키 서버가 새롭게 생성하여 전송할 키 값이다. $h(\ )$는 일방향 해쉬 함수, $f(\ )$는 XOR과 같은 혼합 함수(mixing function), G(Generation)는 키의 생성, $E(K, m)$는 m을 대칭키 K로 암호화하는 것을 의미한다. 본 장에서는 그림 1과 같이 일괄적으로 u_2, u_3, u_5가 동시에 탈퇴할 경우를 가정한다. 그리고 ELKH, EHBT, EOFT, ECSM 방식 모두 탈퇴하지 않은 멤버들의 서브 트리 ST_1, ST_2, ST_3이 탈퇴한 멤버들의 부모 노드 자리로 이동하는 것을 고려한다. "키 갱신 서브 트리(rekey subtree)"는 그림 1에서와 같이 원래의 키 트리 중에서 갱신될 노드들만으로 구성된 서브 트리를 의미한다.

3.2 ELKH (Extended LKH)

LKH를 일괄 키 갱신 기법으로 확장한 ELKH는 그룹 멤버쉽 변화가 있을 때, "키 갱신 서브트리"에 속하는 노드들의 경우 좌·우 자식노드들의 키로 암호화하는 방식이다. 예를 들어 그림 2에서 멤버 u_2, u_3, u_5가 동시에 탈퇴 하였을 때 u_2, u_3, u_5의 경로상의 노드키 K_{ij}는 K'_{ij}로 변경해 주어야 한다. 그리고 갱신될 키 노드들은 각각 자식 노드의 키로 암호화되어 전송된다. 예를 들어 키 갱신 노드 중 하나인 K'_{00}은 자신의 자식 노드키 K'_{20}과 K'_{11}로 암호화하여 전송하고, K'_{11}은 K_{34}와 K'_{23}으로 암호화하여 전송한다. 탈퇴한 멤버의 노드키는 트리에서 삭제한다. 따라서 키 갱신 노드의 말단 노드를 제외한 노드들은 각각 두 번씩의 암호화 과정을 갖는다.

3.3 EHBT

EHBT는 탈퇴한 멤버들의 그룹인 subtree에서 root 노드의 형제노드는 삭제된 부모노드의 위치로 이동 후 한 번의 해쉬를 갖고, root에 이르는 경로상의 노드키들은 각각 한 번의 해쉬와 한 번의 암호화 과정을 갖는다. 예를 들어 그림 3에서와 같이 K_{20}은 부모 노드의 자리에 위치하고 자신의 키를 해쉬 적용하여 $K'_{20} = h(K_{20})$으로 갱신한다. 그리고 갱신된 키를 한 번의 해쉬를 더 적용하여 그룹키 $K'_{00} = h(K'_{20})$을 구한다. 탈퇴한 멤버의 형제 노드를 제외한 그룹의 다른 멤버들 u_4, u_6, u_7에게는 그들의 공유키 K'_{11}로 그룹키 K'_{00}을 $E(K'_{11}, K'_{00})$와 같이 암호화하여 전송한다. 따라서 탈퇴한 멤버의 형제 노드를 제외한 키 갱신 노드는 한 번의 해쉬와 암호화 과정을 갖는다.

3.4 EOFT (Extended OFT)

EOFT는 OFT 개별키 갱신기법을 일괄 키 갱신 기법으로 확장한 방식이다. 각 멤버들은 자신만의 Unblind 키 K_{ij}^{u}를 알고 있고, K_{ij}^{u}에 일방향 해쉬를 적용한 Blind 키 K_{ij}^{B}를 구할 수 있다. 멤버의 탈퇴로 인해 갱신되어야 할 키 노드들은 K_{ij}^{u}를 일방향 해쉬를 적용하여 새로운 Blind 키 K'^{B}_{ij}를 구하고 이 키를 자신의 형제 노드의 키로 암호화하여 전송하면 전송 받은 키 K_{ij}^{B}를 자신의 새로운 키 K'^{B}_{ij}를 XOR하여 자신의 부모 노드키를 구한다. 이러한 방식으로 root에 이르는 경로상의 노드키들은

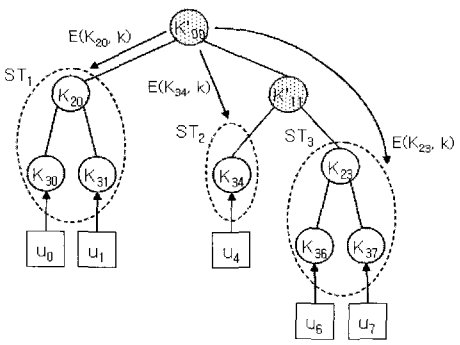


그림 5. 그룹 멤버 u2, u3, u5의 탈퇴 후 ECSM

자동으로 구해지고 결국 그룹키를 얻을 수 있다. 예를 들어 노드 K_{34}에 대하여 키 서버는 노드 K_{34}의 새로운 Blind 키 $K'_{34}{}^{B}$를 도출하고, 그 노드 형제 노드의 Unblind 키로 $E(K_{23}{}^{U}, K'_{34}{}^{B})$와 같이 암호화하여 전송한다. 그리고 노드 K_{20}, K_{11}도 마찬가지로 그들의 형제 노드인 Unblind 키로 암호화하여 전송한다. 그룹 멤버 u_0, u_1은 자신이 도출한 $K'_{20}{}^{B}$와 전송받은 $K'_{11}{}^{B}$를 혼합하여 새로운 그룹키 $K'_{00}{}^{U} = f(K'_{20}{}^{B}, K'_{11}{}^{B})$를 도출할 수 있다. 따라서 루트 노드를 제외한 키 갱신 노드들은 자신의 형제 노드로 각각 한 번의 암호화 과정을 갖는다.

3.5 ECSM (Extended CSM)

ECSM은 Stateless 방식의 CSM을 Stateful 일괄 키 갱신기법으로 확장한 방식이다. ECSM은 그룹 멤버쉽 변동이 있을 때 그룹 멤버들의 모든 키 갱신 정보를 구현하는 stateful방식이다. 이 방식은 확장성 측면에서 stateless방식에 비해 선호되는 방식이다. ECSM을 암호화에 의한 방법과 해쉬에 의한 방법 두 가지로 고려해 본다.

3.5.1 ECSM(암호화에 의한 방법)

그룹 멤버쉽 변동이 있을 때, 키 서버는 탈퇴한 멤버들을 제외한 subtree를 구성하고 각각의 subtree의 root 키로 새롭게 생성한 키 k를 암호화하여 전송하면, 멤버들은 새로운 키 k와 자신의 subtree에 해당하는 root 키를 일방향 해쉬 함수를 적용하여 새로운 보조키 및 그룹키를 얻는다. 예를 들어 그림 5에서처럼 키 서버는 탈퇴하지 않은 멤버들의 서브 트리 ST_1, ST_2, ST_3을 찾고, 새로운 키 k를 생성한다. 그리고 서브 트리 ST_1, ST_2, ST_3의 루트 키 K_{20}, K_{34}, K_{23}으로 새롭게 생성한 키 k를 암호화하여 전송한다. 키 서버의 계산 알고리즘을 나타내면 표 1과 같다. 반면에 그룹 멤버 u_4의 경우에는 자신의 키 K_{34}로 전달받은 메시지를 복호화하고, 기존의 보조키 K_{11}과 전달받은 k를 일방향 해쉬 함수를 적용하여 새로운 보조키 $K'_{11} = h(K_{11}, k)$을 도출한다. 그리고 기존의 그룹키 K_{00}과 k를 일방향 해쉬 함수를 적용하여 새로운 그룹키 $K'_{00} = h(K_{00}, k)$을 도출할 수 있다.

표 1. ECSM에 대한 키 서버의 계산 알고리즘

```
KS generates new key value k
Mark nodes of T(Tree) which should be
updated.
Partition T into several biggest
subtrees.
ST_l, l = 1,2,3,.. , whose node keys are
not to be updated.
Compute E(K_mn, k) for each ST_l, where K_mn
= root key of ST_l.
j ← n ;
For i = m-1 To 0
 j ← ⌊ j/2 ⌋ ;
 k'_ij ← h( k, k_ij ) ;
End For
```

3.5.2 ECSM+(해쉬에 의한 방법)

그룹 멤버쉽 변동이 있을 때, 키 서버는 탈퇴한 멤버들을 제외한 subtree를 구성하고 각각의 subtree의 root키와 임의의 랜덤 값(random number) r을 해쉬한 값에 새롭게 생성한 키 k를 XOR하여 전송한다. 예를 들어 그림 5에서와 같이 키 서버는 탈퇴하지 않은 멤버들의 서브 트리 ST_1, ST_2, ST_3를 찾고, 임의의 랜덤 값 r과 새로운 키 값 k를 생성한다. 그리고 그룹 멤버 u_0, u_1에 대하여 랜덤 값 r과 $h(K_{20}, r) \oplus k$를 멀티캐스트 메시지로 전송하면, 그룹 멤버 u_0, u_1은 그들의 공통키인 K_{20}과 전송받은 랜덤 값 r을 해쉬 적용하고, 전송받은 $h(k_{20}, r) \oplus k$에 그들이 계산한 $h(K_{20}, r)$을 XOR하여 새로운 키 값 k를 도출해낸다. 그룹 멤버는 도출한 키 k를 사용하여 기존의 보조키와 그룹키를 ECSM의 암호화에 의한 방법과 같은 방법으로 갱신할 수 있다. 결과적으로 암호화에 의한 방법과 비슷한 방법을 사용하지만 암호화가 전혀 사용되지 않고 단지 랜덤 값 r과 일방향 해쉬 함수 그리고 XOR만이 연산에 사용되어진다. 그럼에도 불구하고

안전성은 보장된다. 일방향 해쉬 함수의 다음과 같은 특성 때문이다.

특성 모든 $x \in X$, $y \in Y$에 대하여 y와 $z(=h(x, y))$가 주어졌을 때, $h(x', y) = z$와 같은 $x' \in X$를 찾는 것이 계산학적으로 불가능하다.

Ⅳ. 성능 분석 및 비교 평가

본 논문에서의 성능 평가는 탈퇴한 다수의 멤버들을 일괄적으로 처리하는 ELKH, EHBT, EOFT, ECSM에서의 키 서버 계산비용을 암호화 횟수의 평균값을 계산하여 비교 평가한다. 반면에 기존의 논문들에서는 멤버 탈퇴에 따라서 갱신되어야 할 노드의 수를 계산하지 않고, 서버의 계산비용을 제시하고 있다. 결과적으로 여러 기법들간의 정확한 비교, 분석이 명확하지 않았다. 또한 암호화 횟수를 전혀 사용하지 않는 EMKM과 본 논문에서 제안한 ECSM+를 해쉬와 XOR 관점으로 비교 분석하여 본다.

4.1 암호화 횟수에 의한 평균의 경우 분석

평균의 경우(average case) 각각의 노드키가 "키 갱신 서브 트리"에 포함될 확률을 기반으로 Steve[3] 등이 사용한 다음의 방식을 이용한다. 즉, 그림 6에서 키 트리의 레벨 0에 있는 노드를 루트 노드라 하고, 레벨 d에 있는 노드를 말단 노드라 한다. (단, $d = \log_2 N$, N : 말단 노드의 수). $T(v)$는 노드 v가 루트인 서브 트리이고, $T_1(v)$와 $T_2(v)$는 노드 v에 대한 자식의 서브 트리로 가정한다. $L(v)$를 $T(v)$의 말단 노드들의 집합이라 하면, 이진 트리의 경우 $L(v)$의 크기는 $N/2^x$이 된다. (단, $0 \le x \le d-1$). 그러나 본 논문에서는 Steve 등이 사용한 방식과 조금 다른 형태의 방법으로 평균을 분석한다. 즉, 서브 트리의 이동을 고려한 방법으로서 Steve 등이 사용한 방식에서는 노드 v가 "키 갱신 서브 트리"에 포함되지 않을 확률을 서브 트리 $T(v)$의 말단 노드들이 모두 제거되었을 경우로서 정의하지만, 본 논문에서는 노드 v가 "키 갱신 서브 트리"에 포함되지 않을 확률을 노드 v의 어느 한쪽 자식 서브 트리의 말단 노드들이 모두 제거되었을 경우로서 정의한다.

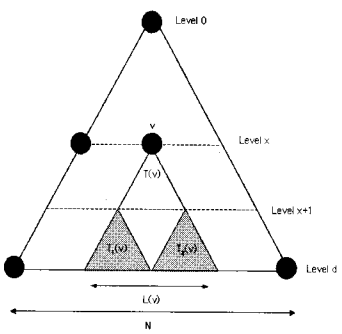


그림 6. 평균의 경우 분석

4.1.1 ELKH

ELKH 기법은 LKH를 단순히 일반화한 것이다. 즉, 이진 트리를 기반으로 한 LKH의 경우에 "키 갱신 서브 트리"에 속하는 한 개의 노드키 갱신을 위해서는 해당 노드의 두 개의 자식 노드키로 두 번의 암호화가 요구된다. v가 "키 갱신 서브 트리"에 속한다는 것은 서브 트리 $T(v)$의 $L(v)$에서 적어도 한 명이 탈퇴 한다는 것을 의미한다. 각각의 그룹 멤버가 탈퇴할 확률이 동일하다고 가정하면, N명의 멤버 중에서 L명이 탈퇴할 경우의 수는 $\binom{N}{L}$이 되고, $\binom{N-N/2^x}{L}$은 $T(v)$의 말단 노드 이외의 위치에서 탈퇴자가 발생하고 거기서 L명이 탈퇴할 경우의 수가 된다. 따라서 v가 "키 갱신 서브 트리"에 속할 확률은 $1-\binom{N-N/2^x}{L}/\binom{N}{L}$이 된다. 또한, 레벨 x에 존재하는 전체 가능한 서브 트리 $T(v)$의 개수는 2^x개이고, 이 서브 트리는 레벨 0부터 레벨 d-1까지에서 정의된다. 그러나 $T(v)$의 어느 한쪽 자식 $T_1(v)$의 모든 멤버들이 탈퇴한다면 노드 v는 삭제되고, $T_2(v)$의 루트 노드는 삭제된 노드 v의 위치로 올라간다. 따라서 노드 v는 "키 갱신 서브 트리"에 포함되지 않고, 결과적으로 대칭형 암호화는 생략된다. 즉, 레벨 x에서 $L \ge N/2^x+1$인 경우에 한해서 $T(v)$의 어느 한쪽 자식 서브 트리에 속하는 모든 멤버들이 탈퇴할 확률은 $\binom{N-N/2^{x+1}}{L-N/2^{x+1}}/\binom{N}{L}$이 된다. 따라서 ELKH의 평균은 식 (1)과 같다.

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

식 (1)에서 $L < N/2^x+1$인 경우에 $\binom{N-N/2^{x+1}}{L-N/2^{x+1}} / \binom{N}{L}$ 는 0으로 정의한다.

4.1.2 EHBT

EHBT 기법의 경우 "키 갱신 서브 트리"에 속하는 한 개의 노드키는 그 노드 한쪽 자식의 노드키에 일방향 해쉬함수를 적용함으로서 갱신 한다. 예를 들어 $T(v)$의 어느 한쪽 자식 서브 트리 $T_1(v)$에서 모두 탈퇴할 경우 노드 v는 삭제되고, $T_2(v)$의 루트 노드는 삭제된 노드 v의 위치로 올라간다. 따라서 삭제된 노드 v는 "키 갱신 서브 트리"에 속하지 않게 되고, $T(v)$의 자식 노드에서 탈퇴할 확률을 ELKH에서와 같이 제외하면 식 (2)와 같다.

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

4.1.3 EOFT

EOFT 기법에서는 그룹 멤버가 탈퇴할 경우 그룹키를 제외한 경로상의 노드키들은 각각 형제 노드의 키로 암호화하여 전송된다. 그룹키를 제외하므로 레벨 1에서부터 d-1까지의 서브 트리가 대상이 되고, 노드 v가 "키 갱신 서브 트리"에 속할 경우 한 번의 암호화가 발생한다. 그림 4의 경우 "키 갱신 서브 트리"는 짙은색으로 표시된 노드키들로 구성된다. 예를 들어 $T(v)$의 어느 한쪽 자식 서브 트리 $T_1(v)$ 모두에서 탈퇴가 발생하고, 다른 쪽 자식 서브 트리 $T_2(v)$의 루트 노드가 삭제된 노드 v의 위치로 올라갈 경우 $T_2(v)$의 루트 노드는 삭제된 노드 v의 형제 노드키로 암호화를 하게 된다. 그러나 $L(v)$에서 모든 멤버들이 탈퇴할 경우, 노드 v는 삭제되므로 "키 갱신 서브 트리"에 포함되지 않는다. 따라서 EOFT에서의 평균의 경우는 식 (3)과 같다.

$$E(L)=1\cdot\sum_{x=1}^{d-1}2^x\left(1-\frac{\binom{N-N/2^x}{L}}{\binom{N}{L}}-\frac{\binom{N-N/2^x}{L-N/2^x}}{\binom{N}{L}}\right) \quad (3)$$

4.1.4 ECSM

ECSM 기법은 다른 기법과 달리 그룹별 탈퇴를 고려한 방식이다. 이 방식은 이상의 방식과 다르게 노드 v가 "키 갱신 서브 트리"에 속할 확률로서 평균을 구하는 것이 아니라, 서브 트리 $T(v)$의 루트 노드에 대한 한쪽 자식의 서브 트리 $T_1(v)$에서 적어도 하나의 탈퇴가 발생하고, 동시에 다른 쪽 자식의 서브 트리 $T_2(v)$에서 탈퇴가 전혀 발생하지 않았을 경우에 한하여, 탈퇴가 전혀 발생하지 않은 서브트리 $T_2(v)$의 루트 키로 암호화 하는 방식이다. 따라서 $T_1(v)$에서 적어도 하나의 탈퇴가 발생할 확률 $1-\binom{N-N/2^{x+1}}{L}/\binom{N}{L}$와 $T_2(v)$에서 탈퇴가 전혀 발생하지 않을 확률 $\binom{N-N/2^{x+1}}{L}/\binom{N}{L}$을 구하여 곱하고 최종적으로 식 (4)를 도출할 수가 있다.

$$E(L)=1\cdot\sum_{x=0}^{d-1}2^x\left(\frac{\binom{N-N/2^{x+1}}{L}}{\binom{N}{L}}\right)\left(1-\frac{\binom{N-N/2^{x+1}}{L}}{\binom{N}{L}}\right) \quad (4)$$

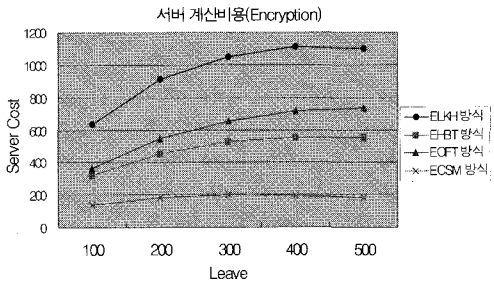


그림 7. 평균의 경우 서버 계산비용

그림 7의 그래프는 기존의 개별키 갱신기법을 일괄키 갱신 기법으로 확장한 ELKH, EHBT, EOFT, ECSM에 대한 서버 계산비용을 보이고 있다. 가로 축이 탈퇴자 수(Leave)이고, 세로 축이 암호화 회수(Server Cost)이다. 1024명의 그룹 멤버가 존재한다고 가정하고, 약 10%(102명)에서 50%(512명)의 멤버가 동시에 탈퇴할 경우에 대해서 측정하였다. 그래프에서 위에서부터 아래로 ELKH, EOFT, EHBT, ECSM이다. 그림 7에서 알 수 있듯이 일괄키 갱신기법으로 평균의 경우를 비교 분석한 결과 ECSM이 가장 좋은 성능을 보이고 있다.

4.2 해쉬에 의한 평균의 경우 분석

기존의 방식들은 대부분 그룹 멤버쉽 변동에 따른 그룹 키 갱신을 암호화에 기반을 두고 있다.

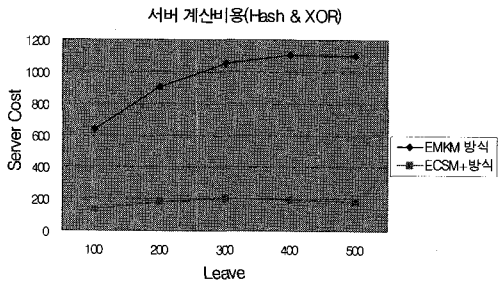


그림 8. 평균의 경우 서버 계산비용

ELKH, EHBT, EOFT, ECSM 모두 암호화 방식으로 그룹키를 갱신한다. 그러나 본 장에서 EMKM 방식과 같이 그룹 멤버의 탈퇴에 따른 그룹키 갱신을 암호화 방식이 아닌 해쉬와 XOR 방식으로 비교 분석해 본다. 일반적으로 일방향 해쉬와 XOR의 계산량은 전자서명 방식의 암호화에 비해 훨씬 적은 것으로 알려져 있다.

4.2.1 EMKM

EMKM 기법은 ELKH와 동일한 방법으로 평균을 구한다. 즉, 이진 트리를 기반으로 하였을 때 "키 갱신 서브 트리"에 속하는 한 개의 노드키 갱신을 위해서는 두 번의 해쉬와 두 번의 XOR가 요구된다. v가 "키 갱신 서브 트리"에 속할 확률에서 $T(v)$의 어느 한쪽 자식 서브 트리에 속하는 모든 멤버들이 탈퇴할 확률을 제외시킨다..

4.2.2 ECSM+

ECSM+ 기법은 ECSM과 동일한 방법으로 평균을 구한다. 즉, 이진 트리를 기반으로 하였을 때, 한 개의 노드키 갱신을 위해서는 한 번의 해쉬와 한 번의 XOR가 요구된다. $T_1(v)$에서 적어도 하나의 탈퇴가 발생할 확률과 $T_2(v)$에서 탈퇴가 전혀 발생하지 않을 확률을 곱하여 구한다.

그림 8의 그래프는 해쉬 기반의 EMKM 방식과 ECSM+ 방식에 대한 서버 계산비용을 보이고 있다. ECSM+ 방식의 경우 그룹 멤버의 탈퇴에 따른 그룹키 갱신을 위해 어떠한 암호화도 사용하지 않고, 단지 해쉬 및 XOR만으로 그룹 키를 갱신 할 수 있다. 그림에서 알 수 있듯이 해쉬 기반의 일괄키 갱신기법으로 평균의 경우를 비교 분석한 결과 ECSM+가 EMKM에 비해 좋은 성능을 보이고 있다.

표 2. 탈퇴시 멀티캐스트 키 갱신 메시지의 크기

기법	ELKH	EOFT	EHBT	EMKM	ECSM+
한 명의 그룹멤버 탈퇴시 메시지의 크기					
멀티캐스트	2dK	dK	dK	2dK	dK
L 명의 그룹멤버 탈퇴시 메시지의 크기					
멀티캐스트	2dLK	dLK	dLK	2dLK	(d~N/2)K

우리가 제안한 ECSM+ 기법은 그룹 멤버 탈퇴시 키 서버 및 그룹 멤버 모두 그룹키 갱신을 위해 어떠한 암호화 및 복호화도 필요로 하지 않는다. 단지 키 갱신을 위해 일방향 해쉬 함수 및 XOR 연산만을 사용한다. 표 2는 멀티캐스트 키 갱신 메시지에 대한 대략적인 수치를 나타낸다. ECSM+를 제외한 다른 기법들은 멀티캐스트 키 갱신 메시지가 그룹 멤버의 탈퇴에 따른 트리내의 키 갱신노드의 수와 관련이 있다. 그러나 ECSM+는 탈퇴하지 않은 멤버들에 대한 그룹의 수와 관련이 있다. 즉, ECSM+의 경우 한 명의 멤버가 탈퇴할 경우 d개의 멀티캐스트 키 갱신 메시지를 필요로 하고, L명이 탈퇴할 경우 d에서 최대 $N/2$개의 멀티 캐스트 키 갱신 메시지를 필요로 한다.

V. 결 론

본 논문에서는 기존에 제시되었던 개별키 갱신기법을 일괄 키 갱신기법으로 확장 전환시키고, Stateless 방식을 Stateful 방식으로 전환하였다. ELKH와 EHBT의 경우 기존에 제시되었던 LKH와 HBT를 일괄키 갱신기법으로 확장한 기법이고, EOFT와 ECSM의 경우도 OFT와 CSM을 일괄키 갱신기법으로 확장한 기법이다. 이러한 확장된 일괄키 갱신기법을 암호화에 의한 평균의 경우로서 비교 분석하였다. 그리고 암호화를 전혀 사용하지 않고 일방향 해쉬 함수 및 XOR 연산만을 사용하는 EMKM과 ECSM+를 평균의 경우로서 비교 분석하였다. 비교분석한 결과 ECSM이 암호화 측면에서, ECSM+가 해쉬 측면에서 키 서버의 계산 비용을 최소화함을 알 수 있었다. 또한 그룹 멤버 탈퇴의 경우 키 갱신 메시지의 크기가 기존의 기법과 비교하여 나쁘지 않았다. 따라서 본 논문에서는 일반적으로 암호화에 의한 방법보다 해쉬에 의한 방법이 계산 비용 측면에서 효율적인 것을 감안할 때, 일방향 해쉬 함수와 XOR 연산만을 사용하는 ECSM+ 방식을 통한 효

율적인 키 관리를 할 수 있다고 본다.

참 고 문 헌

[1] S. E. Deering, "Multicast routing in internetworks and extended LANs," *In Proceedings of ACM SIGCOMM*, pp. 55-64, 1988.

[2] C. K. Wong, M. Gouda, and S. S. Lam, "Secure group communications using key graphs," *In Proceedings of ACM SIGCOMM*, 1998.

[3] D. Wallner E. Harder, and R. Agee, "Key Management for Multicast : Issues and Architectures," IETF RFC 2627, 1999.

[4] X. S. Li, Y. R. Yang, M. G. Gouda, S. S. Lam, "Batch Rekeying for Secure Group Communications," WWW10, 2001.

[5] A. T. Sherman and D. A. McGrew, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees," IEEE Transactions on software engineering, Vol. 29, 2003.

[6] S. Rafaeli, L. Mathy, and D. Hutchison, "EHBT: An efficient protocol for group key management," Lecture Notes in Computer Science, 2233: 159-171, 2001.

[7] D. Naor, M. Naor, J. Lotspiech, "Revocation and Tracing Schemes for Stateless Receivers," Lecture Notes in Computer Science 2139, pp. 41-62, 2001.

[8] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas, "Multicast Security: A Taxonomy and Efficient Authentication," In Proc. of IEEE '99, vol.2, pp.708-716, March 1999.

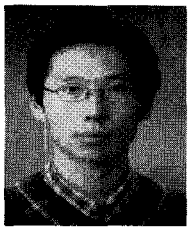
[9] A. T. Sherman and D. A. McGrew, "Key Establishment in Large Dynamic Groups Using One-Way Function Trees," Cryptographic Technologies Group TIS Labs at Network Associates, 1998.

[10] H. Harney and E. Harder, "Logical key hierarchy protocol," IETF Internet Draft, 1999.

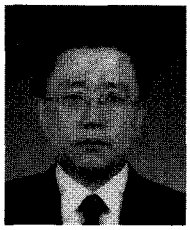
[11] A. Perrig, D. Song, J.D.Tygar, "ELK, a New Protocol for Efficient Large-Group Key Distribution," In 2001 IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 2001.

[12] Ju Hee Ki, Hyun-Jeong Kim, Dong Hoon Lee, Chang-Seop Park, "Efficient Multicast Key Management for Stateless Receivers," ICISC 2002: 497-509

〈著者紹介〉



이 규 원 (Gyu-won Lee) 준회원
2005년: 단국대학교 전자컴퓨터학부 졸업
2005년~현재: 단국대학교 전자계산학과 석사과정
〈관심분야〉 멀티캐스트 보안, 모바일 에드혹 네트워크 보안



박 창 섭 (Chang-seop Park) 정회원
1983년: 연세대학교 경제학과 졸업
1983년: 한국 IBM 근무
1990년: 미국 Lehigh Univ. 전자계산학 박사
1990년~현재: 단국대학교 전자컴퓨터학부 교수
〈관심분야〉 암호 프로토콜, 네트워크 보안