論文
5-9-6

난수를 사용한 효율적인 일괄 rekeying 기법
Efficient Batch Rekeying Scheme using Random Number

정종인(Joung-in Chung)

요 약

멤버쉽의 변화는 그룹키 관리의 확장성 문제와 밀접한 관계가 있다. 그룹의 멤버가 가입하거나 탈퇴하면 새로운 그룹키를 생성하여 그룹의 나머지 모든 멤버들에게 전달되어야 한다. 그룹키의 변경은 그룹 제어기의 주도로 수행된다. 새로운 키를 생성하여 분배하는 멀티캐스트 그룹키 관리에서 제어기와 멤버가 저장하는 키의 수, rekeying할 때마다 제어기가 전달하는 메시지의 수, 초기단계에서 제어기에 의해 전달되는 키의 수, 일괄 rekeying시 전달하는 메시지의 수는 그룹 키 관리기법을 평가하는 중요한 기준들이다. 일괄 rekeying은 순차적으로 개별 rekeying하는 것보다 rekeying에 대한 메시지의 수와 연산비용을 줄일 수 있다. 본 논문에서는 난수를 사용하는 Pegueroles의 그룹키 관리기법에 적용할 수 있는 일괄 rekeying 방법을 제안하고, 그 방법이 적용된 Pegueroles기법은 효율적인 그룹키 관리 기법임을 보인다.

ABSTRACT

Membership changing is deeply associated with scalability problem for group key management. If members of the group join or leave, new group key has to be generated and distributed to all remaining members of group. Group key changing is performed by a group controller. The following parameters are important evaluating criteria of multicast key management scheme that generate and deliver new keys for rekeying: the number of keys stored by both controller and member, messages to deliver, keys to initially be delivered by controller, blocking capability of collusion attacks, messages to deliver at the batch rekeyng. Batch rekeying can reduce messages to deliver and operation costs of generation for message rather than removing members sequentially in fashion one after another. We propose efficient batch rekeying scheme applicable to Pegueroles scheme using random number and prove to be a efficient group key scheme for enhanced Pegueroles model with batch rekeying capability.

논문접수 : 2004. 9. 8.
심사완료 : 2004.10.25.

1) 정회원 : 공주대학교 컴퓨터교육과

1. 서론

미래의 네트워크의 사용자들은 비디오, 오디오, 데이터를 포함하는 멀티미디어 응용을 통신의 많은 부분으로 사용할 것이다. 이러한 사용자들은 멀티캐스트 그룹으로 관리된다. 멀티캐스트 통신을 안전하게 유지하기 위해서는 그룹 멤버들이 공유하는 그룹키가 필요하다. 안전한 멀티캐스트 그룹은 키 서버에 의해 관리된다. 이러한 키 서버를 그룹제어기라 한다. 멀티캐스트 그룹에 가입하기 위하여 클라이언트는 그룹 제어기에게 그룹에 접근을 요구하여야 한다.

그룹 멤버쉽의 변화가 있을 때마다 멤버가 가지고 있는 키를 변경하여야 FS(Forward Secrecy)와 BS(Backward Secrecy)가 보장된다. 이와 같이 키의 변경을 rekeying이라 한다. FS는 어떤 멤버가 그룹을 떠나면 그 이후에 이루어지고 있는 그룹에 대한 정보를 얻을 수 없는 것을 의미한다. BS는 새로운 멤버가 그룹에 가입할 때 이전에 이루어진 그룹에 대한 정보를 얻을 수 없는 것을 말한다[1].

멤버의 가입과 탈퇴는 멀티캐스트 키 관리의 확장성(scalability) 문제와 밀접한 관계가 있다. 1개의 그룹키를 공유하며 N개의 멤버로 구성되는 멀티캐스트 그룹을 생각하여 보자. 그룹에 멤버가 가입하면 BS을 보장하기 위하여 기존의 그룹키를 변경하여야 하며, 멤버가 탈퇴한다면 FS을 보장하기 위하여 그룹키를 변경하여 그룹의 나머지 모든 N-1개의 멤버에게 전달되어야 한다. 그러나 새로운 그룹키를 나머지 모든 멤버들에게 안전하게 보내는 것은 간단한 문제가 아니다. 아주 간단한 해결책은 제어기와 나머지 멤버들간에 유일한 키를 가지고 유니캐스트 통신을 하는 것이다. 이 방법은 간단하지만 N-1개의 유니캐스트 연결과 N개의 비밀키를 요구하기 때문에 확장성이 좋지 않다.

확장성을 제공하기 위하여 KEK(Key Encryption Key)의 논리적인 이진트리를 사용한 연구가 많이 수행되어 왔다[2-4]. LKH기법은 KEK의 이진 키트리를 사용한 방법으로서 그룹키를 변경하기 위한 메시지의 수가 트리의 깊이($\log_2 N$)에 비례하므로 매우 효율적이다. Chang[4]은 확장성을 제공하기 위하여 이진 트리를 사용하며, 제어기가 관리하는 키의 수가 $2\log_2 N+1$을 갖는 효율적인 그룹키 관리기법을 제안하였다. J. Pegueroles[8-11]은 의사 임의(pseudo random) 함수를 바탕으로 한 그룹키 관리기법을 제안하였다. 이 기법은 rekeying시 KEK을 전달하는 기존의 방법과 달리 KEK을 변경하기 위하여 필요한 정보를 전달한다.

그룹의 크기가 크고 탈퇴가 빈번할 경우, 각 멤버를 제거할 때마다 새로운 그룹키를 변경하고 분배하는 것은 많은 연산을 요구한다. 대부분의 응용에서는 멤버탈퇴 요구가 있을 때 즉시 처리할 필요가 없이 주기적으로 탈퇴할 멤버를 모아서 동시에 제거한다[5-7].

Rekeying을 수행할 때 고려하여야 할 가장 중요한 요소는 rekeying하기 위하여 제어기와 멤버가 저장하는 키의 수, rekeying할 때마다 제어기가 전달하는 메시지의 수, 초기단계에 제어기에 의해 전달되는 키의 수, 일괄 rekeying시 전달하는 메시지의 수 등이다.

본 논문에서는 Pegueroles의 기법에 부울함수의 간소화 개념을 적용한 효율적인 일괄 rekeying 기법을 제안한다. 이 일괄 rekeying 기법을 사용한 Pegueroles의 그룹키 관리기법은 기존의 이진키 트리를 사용한 그룹키 관리기법보다 효율적이다.

2. 관련 연구

2.1 LKH

규모가 큰 그룹에서 제어기가 모든 멤버들에게 새로운 키를 일대일로 전송하는 것보다 효율적으로 그룹키를 변경하기 위한 여러 가지 방법들이 연구되어 왔다. 그 중에서 현재 가장 널리 사용되고 있는 방법이 LKH이며 1998년 Wong[3]에서 처음 소개되었으며 Chang[4]에

서는 차수가 2인 이진트리를 사용하였다. LKH의 중요한 개선점은 키 변경시 전달하여야 할 메시지의 복잡도가 O(N)에서 $O(\log_2 N)$으로 줄이는 것인데, 이것은 키 트리가 균형이 잡혀 있다는 가정하에 이루어진다.

키 트리는 키 암호화 방법에 따라 SEK(Session Encryption Key)과 KEK(Key Encryption Key)으로 나눌 수 있다. SEK은 그룹 멤버들에게 멀티캐스트하는 실제적인 데이터, 예를 들면 비디오 컨퍼런스 세션에서의 비디오 스트림을 암호화는 데 사용된다. KEK은 멤버가 SEK을 암호화하는데 사용된다. 보통 KEK은 논리적인 이진트리 구조로 구성된다.

이진트리의 각 노드의 위치를 (레벨 번호, 각 레벨에서의 위치)로 정한다. 예를 들면, 레벨 0의 루트노드는 (0,0), 레벨 1의 노드는 (1,0)과 (1,1), 레벨 2의 노드는 (2,0), (2,1), (2,2)와 (2,3)이다. 그림 1은 8개의 멤버(M_0 ~ M_7)와 중앙집중식의 그룹제어기를 갖는 이진 키 트리의 예를 나타낸 것이다.

트리에서 사각형노드는 그룹멤버를 나타내고, 원형노드는 키를 나타낸다. 원형노드 (X,Y)의 키를 $K_{(X,Y)}$로 표기한다. 트리는 15개의 노드를 가지고 있으며 각 노드에는 한 개의 KEK이 할당되어 있고, 단말노드(레벨 3의 노드)는 멤버에 일대일로 대응되며 그 노드에 있는 키는 멤버의 개인키이다. 루트, $K_{(0,0)}$는 모든 멤버에게 공개되고, 나머지 키는 그 키의 자식노드의 멤버들에게만 공개된다. 예를 들면 $K_{(2,0)}$은 단말노드 (3,0)과 (3,1)의 멤버들에게만 공개되며, $K_{(1,1)}$는 노드 (3,4)부터 (3,7)까지의 멤버에게만 공개된다.

그룹제어기는 키 트리의 각 노드에 해당하는 모든 KEK을 저장하여야 한다. SEK이 변경될 때 새로운 SEK를 갖기 위하여 각 멤버는 그룹제어기가 가지고 있는 KEK의 부분집합을 저장하여야 한다. 각 멤버는 자신의 단말노드로부터 루트에 이르는 경로상의 키들을 소유하며, 루트는 그룹키이다. 예를 들면, 단말 노드 (3,3)에 있는 멤버 M_3은 키 $K_{(3,3)}$, $K_{(2,1)}$, $K_{(1,0)}$, $K_{(0,0)}$을 저장한다.

그림 1에서 새로운 멤버(M_3)가 그룹에 가입하기 위해서는 안전한 유니캐스트채널을 사용하여 그룹제어기에 접속한다. 제어기는 개인키 $K_{(3,3)}$을 만들어 그 멤버에게 전달한다. 그 후에 제어기는 단말노드부터 루트까지의 경로에 있는 모든 KEK를 변경하여 해당 멤버에게 변경된 키를 전달해야 한다. 제어기는 유니캐스트 채널을 통하여 모든 변경된 키를 가지는 2개의 메시지를 멤버 M_2와 M_3에게 한 개씩 전달한다. 그 후에 $\{K_{(1,0)}, K_{(0,0)}\}_{K(2,0)}$를 멀티캐스트하면 M_0과 M_1만이 그것을 복호화할 수 있다. 여기서 $\{L\}_M$은 문자열 L을 키 M에 의해 암호화하여 그룹 전체 멤버들에게 보낸다는 의미이다. 마지막으로 $\{K_{(0,0)}\}_{K(1,1)}$를 멀티캐스트하면 M_4 ~ M_7이 복호화할 수 있다.

M_3이 그룹을 떠난다고 생각하자. M_3이 저장하고 있었던 키를 변경하여야 한다. 변경된 키는 그것의 형제 노드의 키에 의해 암호화하여 나머지 그룹멤버에게 멀티캐스트한다. 즉, M_2의 개인키를 사용하여 유니캐스트 채널을 통하여 M_2에게 변경된 키의 모든 집합을 보낸다. 그 후에 멀티캐스트 메시지 $\{K_{(1,0)}, K_{(0,0)}\}_{K(2,0)}$를 만들어 보낸다. M_1과 M_2는 그 메시지를 해독할 수 있다. 마지막으로 메시지 $\{K_{(0,0)}\}_{K(1,1)}$를 멀티캐스트한다. 이것은 M_4부터 M_7까지의 모든 멤버가 해독할 수 있다. 이 시점에 M_3은 그룹멤버에 있었을 때 알고 있었던 모든 키가 변경되기 때문에 앞으로의 그룹통신에서 배제가 되므로 FS가 보장된다.

LKH는 $O(\log_2 N)$개의 메시지를 사용하여 키를 변경할 수 있으며 그룹 제어기는 트리를 유지하기 위하여 2N개의 키를 저장하여야 한다.

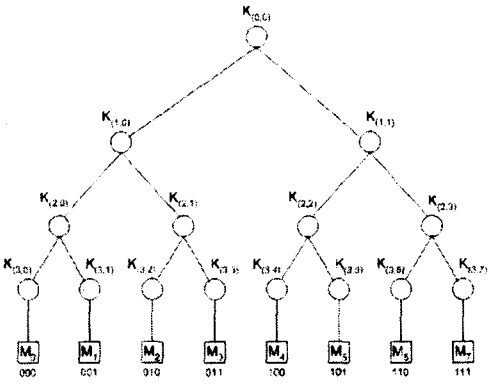

그림 1. 8개의 멤버를 가진 이진 키 트리

2.2 Chang의 기법

그룹의 각 멤버에는 n비트 길이의 2진수 문자열을 가진 사용자 ID(UID)가 부여된다. UID는 $x_{n-1}x_{n-2}\cdots x_0$으로 표현되어지며 각 x_i는 0이나 1이다. UID의 길이는 그룹 크기에 의존한다.

UID $x_{n-1}x_{n-2}\cdots x_0$인 멤버가 그룹에 가입하기 위하여 멤버는 제어기로부터 n개의 보조키 세트 K_{n-1}, K_{n-2}, $\cdots K_0$를 받는다. 여기서 K_i는 x_i가 1이면 k_i, 0이면 $\overline{k}_i$로 표현된다. 보조키는 그룹키를 안전한 방법으로 갱신하는데 사용된다. k_i와 $\overline{k}_i$는 서로 보수키(complement key)이며, 수치적인 보수관계가 아니라 서로 관계가 없는 키라는 의미이다. 제어기는 모든 보조키 $\{k_0, \overline{k}_0, k_1, \overline{k}_1, \cdots, k_{n-1}, \overline{k}_{n-1}\}$를 관리한다.

그림 2에서 그룹의 크기가 8인 경우에 사각형 단말 노드는 3비트의 UID를 가지는 그룹 멤버를 나타내고, 트리의 원형 노드는 키를 나타낸다. 각 멤버는 단말 노드에서 루트 노드까지의 경로의 키를 보관한다. 예를 들면, 멤버 M_5(UID 101)은 그룹키 GK와 보조키 k_2, $\overline{k}_1$, k_0를 보관한다.

Rekeying의 주기를 라운드(round)라 하며 rekeying은 매 주기마다 불연속적으로 일어나므로 그룹키와 보조키를 각각 GK(r)과 $k_i(r)$, $\overline{k}_i(r)$로 표현한다. 여기서 r은 현재의 라운드를 나타낸다. 그룹의 한 멤버가 탈퇴할 때마다 새로운 그룹키가 그룹을 탈퇴하는 멤버를 제외한 모든 멤버들에게 분배되어야 한다. 현재의 그룹키 GK(r)을 변경하기 위하여 제어기는 새로운 그룹키 GK(r+1)를 계산한다. GK(r+1)는 제거되는 멤버의 보수키를 사용하여 암호화된다. 예를 들면, 제거되는 멤버의 UID가 101이라면 그 멤버는 보조키 k_2, $\overline{k}_1$, k_0을 가지고 있다. 그러므로 GK(r+1)는 그 멤버의 보수키 $\overline{k}_2$, k_1, $\overline{k}_0$에 의해 각각 암호화된 3개의 메시지 $\{GK(r+1)\}_{\overline{k}_0}$, $\{GK(r+1)\}_{k_1}$, $\{GK(r+1)\}_{\overline{k}_2}$를 만들어 그룹의 모든 멤버들에게 분배한다. 제거될 멤버도 암호화된 모든 메시지를 수신하지만 그 메시지들은 그 멤버가 가지고 있지 않은 키로 암호화되어 있기 때문에 복호화할 수 없다. 그러나 그룹의 나머지 멤버들의 UID는 제거될 멤버의 UID와 적어도 1비트 이상 다르므로 그 멤버들은 암호화된 메시지 중의 적어도 1개 이상을 복호화할 수 있다.

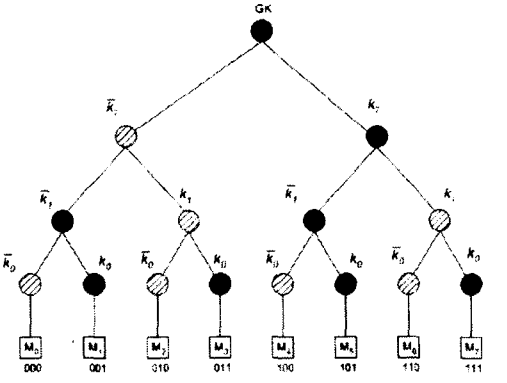

그림 2. 한 개의 멤버 M_5가 탈퇴하는 예

그림 2에서 멤버 M_5가 제거될 때 짙은 원형 노드는 그룹에서 제거되는 멤버 M_5가 가지고 있는 보조키를 나타낸다. 빗금친 원형노드는 M_5가 가지고 있지 않는 보조키 즉, M_5의 보수

키를 나타낸다. M_5에서 루트까지의 경로의 모든 노드는 짙은 원형 노드이지만 그 외의 멤버에서 루트까지의 경로에는 적어도 1개 이상의 빗금친 노드가 존재한다. 그러므로 M_5를 제외한 모든 멤버는 M_5가 가지고 있지 않은 보조키에 의해 암호화된 1개 이상의 메시지를 복호화할 수 있다. Chang의 기법을 분석하면 제어기에 의해 관리되는 키의 수는 $2\log_2 N+1$개이며, 1개의 멤버가 제거된 후에 그룹키를 갱신하기 위하여 보내는 메시지는 $\log_2 N$개이다.

그룹멤버쉽의 변화가 있을 때마다 rekeying을 하면 키를 변경하여야 하므로 많은 비용이 들게 된다. 일괄 rekeying은 개별적인 rekeying보다 더 적은 수의 메시지를 요구한다[4].

2.3 Pegueroles의 기법

그룹 제어기가 저장하여야 하는 키의 수를 감소하기 위하여 rekeying이 필요할 때 제어기는 변경된 키를 보내는 것이 아니라 키를 변경하기 위하여 필요한 정보를 보낸다[8-11]. LKH를 변형하여 단지 제어기만이 알고 있는 의사 임의(pseudo random) 함수에 의해 노드의 키를 만듦으로서 제어기가 저장할 키의 수를 줄일 수 있다.

Rekeying은 키 분배의 초기화, 메시지의 분배 및 키의 재계산 단계로 구성된다. 초기화 단계에서 제어기는 그룹 멤버들에게 전달할 키를 생성한다. 제어기는 그림 1의 노드 (i,j)의 키를 식 (1)에 따라 생성하여 이진 트리를 구성하면 그림 3과 같다.

$$K_{2^i+j}=F_{r1}(2^i+j)\oplus r \qquad (1)$$

F_{r1}은 random seed r_1을 갖는 의사 임의함수이며, 2^i+j를 파라메터로 사용하여 각 노드마다 다른 임의 난수를 생성한다. r은 키를 변경하는데 필요한 다른 난수(random number)이며 $\oplus$는 XOR 함수이다. LKH처럼 단말노드에 있는 각 멤버는 자신의 단말노드로 부터 루트에 이르는 경로상의 키들을 소유한다.

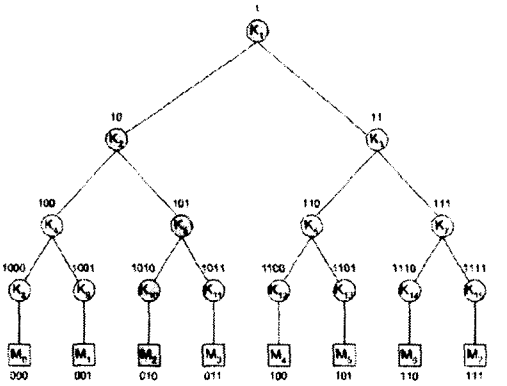

그림 3. 8개의 멤버를 가진 이진 키 트리

멤버의 집합, $M=\{M_0, M_1, \cdots, M_{N-1}\}$이며 $N=2^n$이다. 키 트리에서 그룹의 각 멤버에는 n비트 길이의 2진수 문자열을 가진 멤버 ID(UID)가 부여된다. UID는 $x_{n-1}x_{n-2}\cdots x_0$으로 표현되어진다. 원형노드는 키, 사각형노드는 멤버를 나타낸다. 멤버가 N개이면 키 노드(원형노드)는 2N-1개이다. 루트 노드의 키 ID(KID)는 1, 그의 왼쪽 자식노드의 KID는 $(10)_2$, 오른쪽 자식노드의 KID는 $(11)_2$이다. 어떤 노드의 왼쪽 자식노드의 KID는 부모노드의 KID의 하위비트에 0을, 오른쪽 노드의 KID는 1을 추가하면 된다. 레벨 0에 있는 루트 노드의 KID는 1비트, 레벨 1에 있는 노드의 KID는 2비트 등으로 표현되며, 일반적으로 레벨 i의 KID의 비트는 i+1비트로 표현된다. KID를 멤버의 UID의 기준으로 표현하면, UID가 $x_{n-1}x_{n-2}\cdots x_0$이면 각 멤버에 연결된 트리의 최하위 레벨의 원형노드의 KID는 $1x_{n-1}x_{n-2}\cdots x_0$이며, KID $1x_{n-1}x_{n-2}\cdots x_1 0$과 $1x_{n-1}x_{n-2}\cdots x_1 1$의 부모노드의 KID는 $1x_{n-1}x_{n-2}\cdots x_1$이다. 일반적으로 KID $1x_{n-1}x_{n-2}\cdots x_i 0$과 $1x_{n-1}x_{n-2}\cdots x_i 1$의 부모노드의 KID는 $1x_{n-1}x_{n-2}\cdots x_i$이다.

메시지의 분배단계는 새로운 공유 그룹키가 필요할 때 마다 일어난다. 제어기는 새로운 키를 재 계산하기 위하여 변경된 키를 보내는 것이 아니라 키를 변경하기 위하여 필요한 메시

지를 보낸다. 메시지 P를 식 (2)와 같이 계산하여 멀티캐스트 채널을 통하여 멤버에게 분배한다.

$$P= r_2 \prod_{i \subset S} K_i + (r \oplus r') \quad (2)$$

r_2는 F_{r1}이 아닌 다른 의사 임의 함수의 결과이며 그룹 멤버의 공모공격을 피하는 데 사용된다. $\prod_{i \subset S} K_i$는 키의 곱이며 S는 탈퇴하는 멤버로부터 루트노드까지의 경로의 모든 형제 노드의 부분집합이다. r'는 제어기가 rekeying하기 위하여 r 다음에 생성한 난수이며 $(r \oplus r')$는 나머지 사용자가 변경된 트리를 계산하기 위하여 사용할 숨겨진 값이다. 예를 들어, 그림 3의 M_2가 그룹을 떠날 때 M_0이나 M_1이 키를 변경하는 과정을 보자. 메시지 P는 식 (3)과 같이 계산된다.

$$P=r_2 \cdot K_{11} \cdot K_4 \cdot K_3 + (r \oplus r') \quad (3)$$

키를 재계산할 때 각 멤버는 루트까지 그의 경로에 있는 키중의 한 개를 P에 modulo를 수행하여 $(r \oplus r')$를 구한다. M_0이나 M_1은 P를 구성하는 키 중에서 K_4를 가지고 있기 때문에 식 (4)와 같이 $(r \oplus r')$를 유도할 것이다.

$$P=(r_2 \cdot K_{11} \cdot K_1 \cdot K_3 + (r \oplus r')) \bmod K_4 = (r \oplus r') \quad (4)$$

멤버는 P를 사용하여 식 (5)에 의해 루트노드까지의 경로에 있는 모든 키를 계산한다.

$$K'_{2^i+j} = K_{2^i+j} \oplus P = F_{r1}(2^i + j) \oplus r' \quad (5)$$

r_1을 갖는 의사 임의함수와 r은 제어기외 어느 누구에게도 알려지지 않기 때문에 과거와 미래의 키들은 단지 (i,j)나 P를 가지고 계산할 수 없다. LKH와 Chang기법은 그룹키 관리를 위하여 그룹 제어기가 각각 2N개와 $2\log_2 N + 1$개의 키를 저장하여야 한다. 그룹 제어기가 저장하여야 하는 키의 수를 감소하기 위하여 LKH를 변형하여 단지 제어기만이 알고 있는 의사 임의(pseudo random) 함수에 의해 노드의 키를 만듦으로서 제어기가 저장할 키의 수를 줄일 수 있다. 제어기는 3개의 난수(r_1, r_2, r(혹은 r', r''…))를 저장하여야 한다.

3. 일괄 rekeying

t개의 멤버를 제거하기 위하여 키 변경 과정을 t번 반복할 수 있다. 그러나 더욱 효과적인 방법은 제거할 멤버들을 모아서 한꺼번에 제거하는 것이다. 어떤 시점에서 그룹의 멤버쉽은 UID의 부울함수 mem()에 의해 결정된다. 즉, $mem(x_{n-1}x_{n-2}\cdots x_0)=1$이라면 $x_{n-1}x_{n-2}\cdots x_0$는 그룹의 멤버이고, 그렇지 않으면 그룹에서 탈퇴시킨다.

Rekeying는 mem(UID)=0인 멤버를 제외한 모든 멤버들의 단말노드로부터 루트노드에 있는 모든 키를 변경하는 것이다. 키는 그룹 멤버만이 가지고 있는 P에 의해 암호화되어 변경된다. 확장성과 효율성을 고려한 rekeying은 그룹 멤버에 전송될 메시지의 수와 연산비용이 최소인 것이 바람직하다.

예를 들면, 그림 4에서 M_0(000), M_1(001)이 그룹으로부터 탈퇴한다고 가정하자. 그룹의 나머지 멤버 M_2~M_7에게 새로운 키를 분배하여야 한다. 제어기는 $P=r_2 \cdot K_5 \cdot K_3 + (r \oplus r')$를 모든 멤버에게 분배한다. K_5는 M_2와 M_3이 알고 있기 때문에 식 (4)에 의해 P로부터 $(r \oplus r')$를 구할 수 있으므로 식 (5)와 같이 단말노드부터 루트노드에 있는 모든 키를 변경할 수 있다. 또한 K_3은 M_4~M_7이 알고 있기 때문에 단말노드부터 루트로드에 있는 모든 키를 변경할 수 있다. 그러므로 M_0과 M_1을 제외한 나머지 모든 멤버는 그들이 소유하고 있는 키를 변경할 수 있다.

M_0과 M_1을 순차적으로 탈퇴시킬 경우 M_0을 위하여 $P=r_2 \cdot K_9 \cdot K_5 \cdot K_3 + (r \oplus r')$를 모든 멤버에게 멀티캐스트하여야 하며, M_1을 위하여 $P=r_2 \cdot K_8 \cdot K_5 \cdot K_3 + (r \oplus r')$를 멀티캐스트하여야 한다. 그러

므로 2개의 메시지를 분배하며, 각 메시지는 3개의 키의 곱을 가진다. 일괄 rekeying시 2개의 키의 곱을 가지는 1개의 메시지를 분배하면 된다. P를 구성하기 위하여 곱하는 키의 수(연산비용)는 1개의 멤버를 탈퇴시킬 경우보다 더 적은 수치이다.

또 다른 예를 들면, M_4~M_7을 그룹으로부터 탈퇴시킨다고 가정하자. 제어기는 $P=r_2 \cdot K_2 + (r \oplus r')$를 멀티캐스트하면 된다. 그러므로 1개의 키를 갖는 1개의 메시지를 분배하면 되므로 매우 효율적이다. 그러나 순차적으로 탈퇴시킬 경우 각 3개의 키의 곱을 가진 4개의 메시지가 필요하다.

직관적으로 메시지가 분배될 그룹의 나머지 멤버들의 UID가 공통인 비트가 1개 이상이면 P를 구성하는 키의 수가 줄어들게 됨을 알 수 있다. 위의 M_0과 M_1을 탈퇴시키는 예에서 M_2(010)과 M_3(011)의 UID의 x_2x_1이 공통적으로 '01'이며 이들의 멤버에 연결된 최하위 키 노드의 KID는 각각 1010(K_{10})과 1011(K_{11})이며 이들의 부모 노드의 KID는 101(K_5)이다. 마찬가지로 M_4(100)~M_7(111)의 UID의 x_2가 공통적으로 "1"이며 이들의 멤버에 연결된 최하위 키 노드의 KID는 각각 1100(K_{12}), 1101(K_{13}), 1110(K_{14}), 1111(K_{15})이며 이들의 부모 노드의 KID는 110(K_6), 111(K_7)이며, 또한 이들의 부모노드의 KID는 11(K_3)이다.

일괄제거 문제는 탈퇴시키는 멤버를 제외한 나머지 멤버들의 UID의 비트가 서로 공통인 멤버들의 그룹을 체계적으로 찾는 것과 같은 문제이다. 이러한 문제는 부울 함수의 최소화와 유사한 문제이다. 예를 들면, M_0과 M_1을 탈퇴시킬 때의 멤버쉽 함수는 식 (6)과 같이 표현할 수 있다.

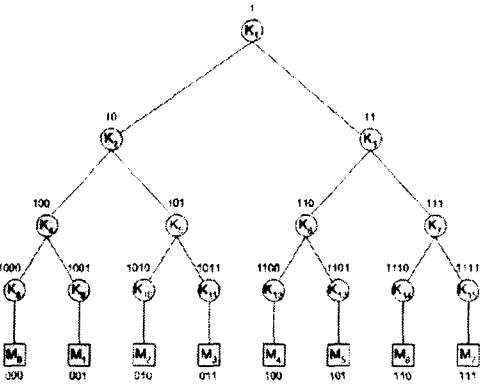

그림 4. M_0과 M_1이 탈퇴할 경우의 일괄 rekeying

$$\begin{aligned} mem(x_2, x_1, x_0) = {} & \bar{x}_2 x_1 \bar{x}_0 + \bar{x}_2 x_1 x_0 \\ & + x_2 \bar{x}_1 \bar{x}_0 + x_2 \bar{x}_1 x_0 \\ & + x_2 x_1 \bar{x}_0 + x_2 x_1 x_0 \end{aligned} \quad (6)$$

여기서 +는 논리적인 OR이며 변수들의 곱은 논리적인 AND를 나타낸다. 부울함수의 최소화방법이 최소의 키를 가지는 rekeying문제에 어떻게 적용되는지를 이해하기 위하여 예를 들어 설명한다. 그룹으로부터 M_0과 M_1을 제거한다면 멤버쉽 함수는 <표 1>과 같다. M_0과 M_1의 출력은 0, 나머지 멤버들의 출력은 1이다. 멤버쉽 함수를 최소항의 합(sum of minterms)으로 나타내면 식 (7)과 같이 표현할 수 있다. 식 (8)의 $\sum$는 항들이 OR된 것을 나타내고, 소문자 m과 숫자는 부울함수의 최소항을 나타낸다. 예를 들면, 최소항 $\bar{x}_2 x_1 \bar{x}_0$를 m(2), $x_2 x_1 \bar{x}_0$를 m(6)으로 표현한다.

$$mem(x_2, x_1, x_0) = \sum m(2, 3, 4, 5, 6, 7) \quad (7)$$

표 1. 멤버쉽 함수

입력 ($x_2x_1x_0$)	출력
000	0
001	0
010	1
011	1
100	1
101	1
110	1
111	1

그림 5는 표 1의 카르노맵을 나타낸다. 카르노맵의 사각형은 최소항이다. 최소화하는 과정은 인접해 있는 1의 최소항들의 블록의 크기가 가장 큰 것을 상위비트위주로 간소화한다. 이때 어떤 블록에 포함되는 최소항은 다른 블록에 포함되지 않도록 한다. 디지털 논리회로에서 부울함수의 최소화방법은 인접해 있는 1의 최소항들의 블록이 가능한 크게 2의 멱승개가 되도록 다른 블록의 최소항들과 공유할 수 있다는 것이 다르다. 그림 5에서 4개의 최소항으로 구성되는 블록이 2개 존재하는데 m(4), m(5), m(6), m(7)에 의해 x_2로 최소화되는 블록과 m(2), m(3), m(6), m(7)에 의해 x_1로 최소화되는 블록이 있다. 그 중에서 상위비트위주로 최소화된 x_2가 선정된다. 나머지 2개의 최소항들(m(2), m(3))의 블록은 $\bar{x}_2x_1$으로 최소화된다. <표 1>은 식 (8)과 같이 최소화된다.

$$mem(x_2, x_1, x_0) = x_2 + \bar{x}_2x_1 \quad (8)$$

부울식의 각 항으로부터 KID를 구하는 방법은 x_i이면 1, $\bar{x}_i$이면 0을 부여하고 최상위비트는 1을 붙인다. 제어기는 x_2로부터 KID $1x_2$=11(K_3)를 구하고 $\bar{x}_2x_1$로부터 $1\,\bar{x}_2x_1$=101(K_5)를 구한다. 그리하여 $P=r_2 \cdot K_5 \cdot K_3 + (r \oplus r')$를 생성하여 모든 멤버에게 멀티캐스트한다.

그림 5. 멤버쉽 함수의 카르노맵과 최소화

x_2 \ x_1x_0	00	01	11	10
0	0	0	1	1
1	1	1	1	1

M_0과 M_4가 동시에 탈퇴하는 다른 예를 들어 보자. 멤버쉽함수는 식 (9)와 같다.

$$mem(x_2, x_1, x_0) = \sum m(1, 2, 3, 5, 6, 7) \quad (9)$$

최소화하는 과정은 그림 6에서 4개의 최소항으로 구성되는 블록이 2개 존재하는데 m(2), m(3), m(6), m(7)에 의해 x_1로 최소화되는 블록과 m(1), m(3), m(5), m(7)에 의해 x_0으로 최소화되는 블록이 있다. 그 중에서 상위비트위주로 최소화된 x_1이 선정된다. 나머지 2개의 최소항들(m(1), m(5))의 블록은 $\bar{x}_1x_0$로 최소화하여 식(10)이 된다.

$$mem(x_2, x_1, x_0) = x_1 + \bar{x}_1x_0 \quad (10)$$

부울식의 첫째 항 x_1으로부터 KID $1Xx_1$=1X1을 구한다. 여기서 X는 don't care이며 0이거나 1의 값을 가진다. 그러므로 KID는 101(K_5)과 111(K_7)이 구해진다. 두 번째 항으로부터 KID $1X\,\bar{x}x_0$=1X01를 구하며 이것은 1001(K_9)과 1101(K_{13})이다. 그리하여 $P=r_2 \cdot K_5 \cdot K_7 \cdot K_9 \cdot K_{13} + (r \oplus r')$를 생성하여 모든 멤버에게 멀티캐스트한다.

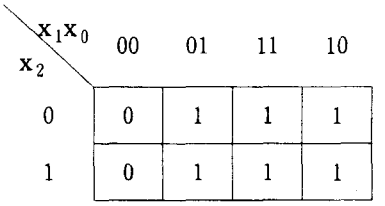

그림 6. 멤버쉽함수의 최소화

그룹크기가 N일 때 멤버의 집합 {2i, 2i+1 | i=0…(N/2-1)}에서 각 i에 대하여 반드시 2i나 2i+1중에서 반드시 한 개의 멤버가 탈퇴할 때 P를 구성하는 키의 수는 N/2개가 필요하며 최악의 경우이다.

4. 비교 분석

Pegueroles기법에 본 논문에서 제안한 일괄 rekeying 방법을 적용한 기법을 확장된 Pegueroles(E-Pegueroles)라 한다. E-Pegueroles기법과 기존의 그룹키 관리 기법인 일대일통신, LKH, Chang모델을 비교분석함으로서 E-Pegueroles기법의 성능이 우수함을 보인다. 4개의 모델 간에 다음의 5개의 항목에 대하여 비교한다.

(A) 제어기가 초기에 전달할 키의 수

(B) rekeying하기 위하여 전달하는 메시지의 수

(C) 제어기가 저장하는 키의 수

(D) 멤버가 저장하는 키의 수

(E) 일괄 rekey시 전달할 최악의 메시지의 수

제어기가 초기에 전달할 키의 수는 그룹키 관리를 위하여 키 트리를 형성하면서 각 노드의 키를 생성하여 모든 멤버에게 전달하여야 하며 이것은 제어기가 생성하여야 하는 각 노드의 키의 수를 의미한다. Rekeying하기 위하여 전달하는 메시지의 수는 rekeying시 키를 변경하기 위하여 제어기가 전달하는 메시지의 수를 나타낸다. 이 항목은 시스템의 대역폭과 관련되므로 그룹키 관리 체제의 확장성과 밀접한 관계가 있다. 제어기가 저장하는 키의 수와 멤버가 저장하는 키의 수는 키 트리를 유지하고 키 트리의 노드 값을 변경하기 위하여 각각 제어기와 멤버가 저장하여야 하는 키 혹은 데이터를 의미한다. 제어기와 멤버가 저장하는 키의 수도 그룹키 관리체제의 확장성을 측정하는 중요한 척도이다. 일괄 rekeying시 전달할 최악의 메시지의 수는 제어기가 rekeying시 전달하여야 하는 최악의 메시지수를 의미한다.

표 2는 그룹의 크기를 N이라 할 때 위에서 언급한 A~E 항목에 대하여 제어기와 모든 각 멤버간에 키를 일대일통신으로 관리하는 방법, LKH, Chang, E-Pegueroles기법간의 비교를 나타내고 있다.

표 2. 기존 모델과의 비교

항목	일대일	LKH	Chang	E-Pegueroles
A	N	O(2N)	$O(2\log_2 N)$	O(2N)
B	N	$O(\log_2 N)$	$O(\log_2 N)$	1
C	N	O(2N)	$O(2\log_2 N)$	3
D	1	$O(\log_2 N)$	$O(\log_2 N)$	$O(\log_2 N)$
E	N-2	O(N/2)	O(N/2)	1

그룹 키를 관리하기 위하여 제어기와 모든 각 멤버간에 일대일로 통신하는 방법에서는 rekeying시 전달하여야 메시지의 수와 제어기가 저장하는 키의 수는 N개이다. 일괄 rekeying시 전달할 최악의 메시지의 수는 2개의 멤버가 탈퇴할 때 제어기가 전달하여야 하는 메시지의 수는 N-2개이다.

LHK에서 제어기가 초기에 전달할 키의 수와 제어기가 저장하여야 할 키의 수는 키 트리의 키 노드의 수(2N-1)이며, rekeying시 전달하여야 할 메시지의 수와 멤버가 저장하는 키의 수는 $\log_2 N$개이다. 일괄 rekeying시 전달할 최악의 메시지의 수는 멤버 2i나 2i+1, i=0…N/2-1 중에서 반드시 한 개의 멤버가 탈퇴

할 때 최악의 경우이며, N/2개의 메시지가 필요하다.

Chang모델에서 제어기가 초기에 전달할 키의 수와 제어기가 저장하여야 할 키의 수는 그룹키를 포함하여 $2\log_2 N+1$개이며, rekeying시 전달하여야 할 메시지의 수와 멤버가 저장하는 키의 수는 $\log_2 N$개이다.

N/2개의 멤버가 탈퇴할 때, 나머지 N/2개의 멤버들에게 새로운 키를 분배하여야 하는 경우를 보자. 그룹에 남아 있는 나머지 멤버들의 임의의 2개의 멤버들간의 해밍거리가 2이상이 될 때 최악의 복잡도가 나오게 된다. 즉, rekeying을 하기 위하여 각 멤버 당 1개의 메시지를 보내야 하므로 일괄 rekeying시 전달할 최악의 메시지의 수는 N/2개가 필요하다.

E-Pegueroles의 기법은 제어기가 초기에 전달할 키의 수는 키 트리의 키 노드의 수와 같기 때문에 2N-1개이다. 이 수치는 LKH와 같으나 일대일기법이나 Chang모델보다 다소 높은 편이나 이것은 키 트리를 생성할 때 한번만 수행하므로 다른 항목에 비하여 중요한 항목은 아니다. 멤버가 저장하는 키의 수는 단말노드에서 루트노드까지의 경로에 있는 키 노드의 수와 같으므로 LKH와 Chang모델과 같이 $\log_2 N$개이다.

제어기가 저장하여야 할 키의 수는 3개이며 rekeying시 한 개의 메시지 P만 전달하면 되므로 이들 수치는 다른 모든 모델보다 월등히 우수함을 보여준다. rekeying시 제어기에 의해 새로운 키노드의 키를 생성하여 전달하는 것이 아니라 rekeying을 위한 정보만 전달한다. 그러면 각 멤버가 그 정보를 사용하여 그들이 가지고 있는 키를 변경하므로 연산이 분산되어 네트워크의 자원을 효율적으로 사용한다. 또한 일괄 rekeying시 전달할 최악의 메시지의 수는 1개이므로 다른 비교 모델보다 우수함을 알 수 있다.

5. 결론

KEK의 이진 키트리를 사용한 방법은 rekeying을 위한 메시지의 수가 트리의 깊이($\log_2 N$)에 비례하므로 확장성에 매우 효율적이다. 그룹의 크기가 크고 탈퇴가 빈번할 경우, 각 멤버를 제거할 때마다 새로운 키를 변경하고 분배하는 것은 많은 통신 대역폭과 연산을 요구한다. Rekeying을 수행할 때 고려하여야 할 가장 중요한 요소는 rekeying하기 위하여 제어기와 멤버가 저장하는 키의 수, rekeying할 때마다 제어기가 전달하는 메시지의 수, 초기단계에 제어기에 의해 전달되는 키의 수, 일괄 rekeying시 전달하는 메시지의 수 등이다.

E-Pegueroles기법은 제어기가 초기에 전달할 키의 수는 LKH와 같으나 일대일기법이나 Chang모델보다 다소 높은 편이나 이것은 키 트리를 생성할 때 한번만 수행하므로 다른 항목에 비하여 중요한 항목은 아니다. 멤버가 저장하는 키의 수는 LKH와 Chang모델과 같이 $\log_2 N$개이다.

제어기가 저장하여야 할 키의 수는 3개이며 rekeying시 한 개의 메시지 P만 전달하며 일괄 rekeying시 전달할 최악의 메시지의 수는 1개이므로 확장성이 뛰어나며 이 수치들은 다른 모든 모델보다 월등히 우수함을 보여준다. Rekeying시 제어기에 의해 새로운 키노드의 키를 생성하여 전달하는 것이 아니라 rekeying을 위한 정보만 전달한다. 그러면 각 멤버가 그 정보를 사용하여 그들이 가지고 있는 키를 변경하므로 연산이 분산되어 네트워크의 자원을 효율적으로 사용한다.

멤버의 탈퇴와 가입이 반복되면 이진 키트리의 형태가 불균형을 이루게 된다. 예를 들면, 오른쪽 서브트리에서는 탈퇴가, 왼쪽 트리에서는 가입이 반복된다면 키 트리가 극심한 불균형을 이루게 된다. Rekeying시 멤버가 저장하여야 하는 키의 수는 키 트리의 레벨수에 대응하므로 키 트리가 불균형은 그룹키 관리의 기법의 확장성이 나쁘게 한다. 그러므로 키트리의 재균형(rebalancing) 알고리즘에 대한 향후 연구가 필요하다.

참고 문헌

[1] R. Canetti, J. Garay, G.Itkis, D.Micciancio, M.Naor, B. Pinkas, "Multicast Security: A Taxonomy and Some Efficient Constructions," IEEE INFOCOM 99, Mar., 1999.

[2] H. Harney, E. Harder, "Logical Key Hierarchy Protocol(LKH)," I-draft Harney-sparta-lkhp-sec-00, Mar.,1999.

[3] C. K. Wong, M. Gouda, S. S. Lam, "Secure Group Communications using Key Graphs," Proceedings of ACM SIGCOMM, Vancouver, British Columbia, Sep., 1998.

[4] I. Chang, R. Engel, D. Kandlur, D. Pendarakis, D. Saha, "Key Management for Secure Internet Multicast using Boolean Function Minimization Techniques," Proceedings of Infocom, New York, Mar., 1999.

[5] S. Banerjee, B. Bhattacharjee, "Scalable secure Group Communication over IP Multicast," Proceedings of ICNP, Nov., 2001.

[6] M. Steiner, G. Tsudik, M. Waidner, "Key Agreement in Dynamic Peer Groups," IEEE Tr. on Parallel and Distributed Systems, Vol 11, No 8, pp.769-780, 2000.

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

[8] J. Pegueroles, W. Bin, M. Soriano, F. Rico-Novella, "Group Rekeying Algorithm Using Pseudo-Random Functions and Modular Reduction," LNCS 3032, p. 875-882, 2004.

[9] J. Pegueroles, F. Rico-Novella, L. Hernandez-Serrano, M. Soriano, "Improved LKH for Batch Rekeying in Multicast Groups, " IEEE ITRE 2003.

[10] J. Pegueroles, F. Rico-Novella, "Balanced Batch LKH New Proposal, Implementation and Performance Evaluation," ISCC 2003.

[11] J. Pegueroles, J. Hernandez-Serrano, F. Rico-Novella, M. Soriano, "Adapting GDOI for Balanced Batch-LKH," draft-irtf-gsec-gdoi-batch-lkh-00.txt, June, 2003.