다중 코어 기반 트리를 이용한 2계층 그룹키 관리 구조 및 프로토콜

(2-Layered Group Key Management Structure and Protocols using Multi-Core Based Tree)

조 태 남 † 김 상 희 †† 은 상 아 †††
(Taenam Cho) (Sanghee Kim) (Sang-A Eun)
이 상 호 †††† 채 기 준 †††† 박 원 주 ††††† 나 재 훈 †††††
(Sang-Ho Lee) (Kijoon Chae) (Wonjoo Park) (Jaehoon Nah)

요 약 원격회의나 소프트웨어 배포 등 다양한 멀티캐스팅 응용 서비스의 보안을 위해서는 정당한 멤버들만이 비밀리에 그룹키를 공유하여야 한다. 특히 그룹의 규모가 크고 멤버의 변동이 빈번한 경우에는 확장성을 위하여 효율적으로 그룹키를 갱신할 수 있어야 하는데, 확장성 있는 키 갱신을 위한 한가지 방편으로서 서브 그룹 구조를 사용한다. 본 논문에서는 서브 그룹 관리자를 그룹 멤버와 분리시킨 DEP 구조와 다중 코어를 갖는 멀티캐스트 프로토콜인 CBT를 접목한 2계층 관리 방식을 제안하고, 제안한 구조에 적합한 서브 그룹키 관리 프로토콜을 선정하였으며, 서브 그룹 관리자를 그룹 통신으로부터 배제하기 위한 키 갱신 프로토콜을 설계하였다. 이것은 기존의 CBT에 기반한 키관리 프로토콜에 비하여 forward secrecy와 backward secrecy 및 확장성을 제공한다. 또한, 2계층 관리 구조를 갖도록 함으로써 DEP에 비하여 키 갱신 메시지의 암·복호화 수를 줄였으며, 키 관리의 효율성을 높이고 그룹 관리자가 유지해야 하는 그룹 멤버에 대한 정보량을 감소시켰다.

키워드 : 그룹키 관리, 보안, 멀티캐스트, CBT

Abstract Assuring the security of group communications such as tele-conference and software distribution requires a common group key be shared among the legal members in a secure manner. Especially for large groups with frequent membership change, efficient rekey mechanism is essential for scalability. One of the most popular ways to provide scalable rekey is to partition the group into several subgroups. In this paper, we propose a two-layered key management scheme which combines DEP and CBT, a protocol in which subgroup manager cannot access the multicast data and another that has a multi-core, respectively. We also select sub-group key management protocols suitable for our structure and design new rekey protocols to exclude the subgroup managers from the multicast data. Compared to previous protocols based on CBT, our scheme provides forward secrecy, backward secrecy and scalability. This would reduce the number of encryption and decryption for a rekey message and would improve the efficiency number of rekey messages and the amount of information related to group members that group managers must maintain compared to DEP.

Key words : group key management, security, multicast, CBT

· 본 연구는 한국전자통신연구원 정보보호기술연구본부 네트워크보안연구부 위탁연구과제에 의한 것임
† 정 회 원 : 이화여자대학교 컴퓨터학과
tncho@ewha.ac.kr
†† 학생회원 : 이화여자대학교 컴퓨터학과
kshee@ewha.ac.kr
††† 비 회 원 : 삼성전자 연구원
esa77@hanmail.net
†††† 종신회원 : 이화여자대학교 컴퓨터학과 교수
shlee@ewha.ac.kr
kjchae@ewha.ac.kr
††††† 비 회 원 : 한국전자통신연구원 연구원
wjpark@etri.re.kr
jhnah@etri.re.kr
논문접수 : 2002년 4월 9일
심사완료 : 2002년 7월 20일

1. 서 론

멀티캐스트와 같은 통신 기술의 발달로 원격회의나 실시간 정보 서비스, 유료 영상 서비스와 같은 그룹통신 응용 개발이 성장하고 있다. 사내 원격 회의나 분산 시뮬레이션과 같은 응용에서는 데이타의 기밀성이 보장되어야 하고, 유료 영상 서비스나 소프트웨어 배포와 같은 응용에서는 사용료를 지불한 사용자만이 접근할 수 있어야 한다. 이것은 멀티캐스트 그룹에 가입한 혹은 허용된 멤버들만이 비밀키를 공유하고 이를 이용하여 암호화 통신을 함으로써 해결될 수 있다[1]. 그러므로 그룹키의 안전하고 효율적인 공유 방법은 멀티캐스트 보안에 매우 중요하다.

가입과 탈퇴가 허용되는 동적인 그룹에서는 정당한 멤버로 가입 혹은 허용된 기간에만 데이타 접근이 허용되어야 하기 때문에 안전성을 위하여 다음과 같은 요구조건을 만족하여야 한다[2][3][4].

• GKS(Group Key Secrecy): 도청자가 그룹키를 알아내는 것은 계산상 불가능해야 한다.

• FS(Forward Secrecy): 일정 기간동안 생성된 그룹키들을 안다고 하더라도, 이로부터 그 이후에 생성되는 그룹키를 유도할 수 없어야 한다.

• BS(Backward Secrecy): 일정 기간동안 생성된 그룹키들을 안다고 하더라도, 이로부터 그 이전에 생성된 그룹키를 유도할 수 없어야 한다.

GKS를 제공하기 위해서는, 공개된 통신 채널을 통해 송수신되는 키의 분배 및 갱신 메시지를 보호하여 이로부터 키를 알아낼 수 없도록 해야 한다. FS는 합법적으로 통신에 참가하던 멤버가 탈퇴했을 때 탈퇴한 이후의 통신을 수신할 수 없도록 하기 위한 조건이며, BS는 합법적으로 가입하여 통신에 참가하는 멤버라도 가입 이전의 통신을 수신할 수 없도록 하기 위한 조건이다. FS와 BS를 만족시키기 위해서는 멤버쉽의 변동이 있을 때마다 그룹키를 갱신해 주어야 하며, 갱신되는 키는 이전이나 이후의 키로부터 유도될 수 없어야 한다.

그룹의 규모가 커지고 멤버쉽 변동이 빈번하게 발생할 경우에는 키 갱신 메시지로 인한 네트웍 부하 및 멤버들의 과부하로 그룹 통신의 성능이 저하된다. 그러므로 키 갱신 프로토콜의 효율성뿐 아니라 멤버의 가입과 탈퇴에 따른 키 갱신의 범위를 줄일 수 있도록 다음과 같은 조건을 만족해야 한다.

• 효율성 : 키 갱신 메시지는 갱신되는 키의 개수, 메시지 수, 메시지 크기 등에서 1:1 통신을 통한 키 갱신보다 효율적이어야 한다.

• "1 affects n" 확장성[5] : 한 멤버의 가입이나 탈퇴가 나머지 모든 멤버에게 영향을 미치지 않아야 한다.

그룹키 관리는 그룹의 크기와 멤버쉽 변화뿐 아니라 멀티캐스팅 되는 데이타의 비밀성 정도, 보안이 유지되어야 하는 기간, 멤버 장치의 계산 능력과 용량, 멤버들에 대하여 기반하고 있는 신뢰 모델 등에 따라 다르게 설계될 수 있다. 일반적으로 효율성과 안전성은 서로 상충관계(trade-off)에 있으므로, 응용에 따라 필요로 하는 보안 정도나 조건에 적절한 관리 기법이 필요하다.

본 논문에서는 동적 그룹의 확장성 있는 관리를 위하여 다중 코어를 가진 CBT(Core Based Tree)를[6] 이용한 서브 그룹형 그룹키 관리 구조 및 프로토콜을 제안한다. 이 구조는 DEP(Dual Encryption Protocol)[5]와 유사한 구조를 가지지만 각 서브 그룹의 특성을 분석하여 적절한 프로토콜을 적용함으로써 효율성을 높이고, 관리자 및 서브 그룹 관리자의 역할을 조정하여 멤버의 등록 절차를 간소화하였으며 관리자가 저장해야 하는 정보의 양을 줄였다. 또한 CBT를 접목함으로써 관리자와 멤버간의 공유키를 간단하게 갱신할 수 있도록 하였다.

2장에서는 관련 연구를 기술하고 3장에서는 본 논문에서 제안하는 키 관리 구조 및 프로토콜을 소개한다. 4장에서는 제안한 구조에 대한 안전성 및 효율성을 기존 연구와 비교 분석하고 5장에서 결론을 맺겠다. 본 논문을 통하여 공통적으로 사용되는 용어들을 다음과 같이 정의한다.

• 멤버(Member): 그룹의 일원으로서 멀티캐스팅 데이타에 접근이 허용된 개체이다. n은 그룹 멤버의 수이다.

• S_i(Subgroup): 그룹의 $i(1 \le i \le l$, l은 서브 그룹의 수)번째 서브 그룹이다. $n_i = |S_i|$

• *GM*(Group Manager): 그룹 관리자로서 그룹키를 생성하고 배포하며, 멀티캐스트 데이타를 송신하는 주체이다.

• SGM_i(SubGroup Manager): 서브 그룹 S_i $(1 \le i \le l)$의 관리자로서, 그룹 및 S_i의 멤버일 수도 있고 아닐 수도 있다.

• *TEK*(Traffic Encryption Key): 멀티캐스트 데이타 암호화에 사용되는 키, 즉 그룹키이다.

• *KEK*(Key Encryption Key): *TEK* 갱신을 위해 보조적으로 사용되는 키들의 총칭이다.

• LS_s(Local Subgroup key): *GM*과 *SGM*들간의 공유키이다.

• LS_i(Local Subgroup key): SGM_i와 S_i의 멤버들

과의 공유키이다.

- key^* : 갱신된 키 key이다.
- $\{msg\}_{key}$: msg를 key로 암호화한 암호문이다.
- $\{msg\}^{key}$: msg와 msg를 key로 서명한 서명문이다.
- $A \rightarrow B: msg$: A가 B에게 msg를 송신한다.
- U_x, R_x (pUblic key, pRivate key): x의 공개키와 개인키이다.

2. 관련 연구

mCBT-GKM은 멀티캐스트 라우팅 프로토콜인 CBT와 서브 그룹형 관리 구조인 DEP를 접목시킨 구조이다. CBT를 이용한 그룹키 관리 프로토콜들과 DEP의 장단점을 기술하고, 제안하는 구조에 적용할 키 관리 프로토콜들의 특징을 소개한다.

2.1 CBT를 이용한 프로토콜

- CBT(Core Based Tree)[6]

CBT는 수신자가 넓은 지역에 분포하면서 밀집되어 있지 않은 경우에 적합한 멀티캐스트 프로토콜이다. 이 프로토콜은 송신자들과 상관없이 코어(core)라고 불리는 특정 노드를 중심으로 그룹에 유일한 배달 트리(delivery tree)인 CBT가 형성되며, 수신자의 요구에 의해 트리가 확장되는 특성을 가진다. 새 수신자의 가입 신청 메시지는 코어를 향하여 전달되고 이때 메시지가 거친 경로는 확장된 CBT의 일부가 된다. 코어는 CBT의 루트(root)로서 데이타 전송의 중심이 된다. 코어는 하나의 주코어(primary core)와 여러 개의 부코어(secondary cores)들로 구성될 수 있는데, 일반적으로 부코어는 정적으로 선정된다. 멀티캐스트 효율성을 극대화시키기 위한 부코어 선정 방법은 여러 가지가 연구되어 있다[7].

- SMKD(Scalable Multicast Key Distribution)[8]

이 프로토콜에서는 CBT의 코어가 GM의 역할을 담당한다. 멤버가 작성한 토큰을 포함하는 가입 신청 메시지가 코어로 전달되면, 코어는 그룹키를 새 멤버의 공개키로 암호화하고 이를 가입 신청 메시지의 역경로를 통해 새 멤버에게 전달한다. 전달과정에서 경유하게 되는 모든 라우터들도 이 그룹키를 공유한다(그림 1 참조). 만약 가입 메시지가 코어에 도달하기 전에 이미 CBT에 속해 있는 라우터를 만나면, 이 라우터는 더 이상 메시지를 전달하지 않고 자신이 보유하고 있는 그룹키를 새 멤버에게 분배한다. 이 프로토콜에서는 CBT의 모든 라우터가 키 분배의 역할을 수행하며, 멤버의 가입과 탈퇴시에도 기존의 그룹키를 변경 없이 사용하기 때문에 FS와 BS를 보장하지 못한다.

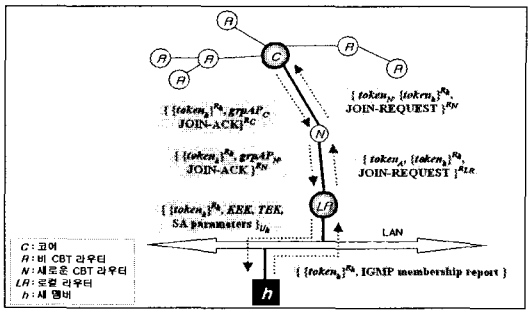

그림 1 SMKD에서 멤버 가입에 따른 키 분배

- KL[9]

이 프로토콜에서 사용하는 멤버 가입시의 그룹키 분배 방식은 SMKD와 유사하다. 즉 가입 요청 메시지가 GM으로 전달되면 현재의 KEK를 새로운 멤버에게 분배한다. 이 방식에서는 멤버의 변동이 발생할 경우에 그룹키 갱신 메시지를 사용하는 것이 아니라, 송신자가 매번 갱신한 키를 전송한다. 예를 들면, 송신자가 메시지 M_k을 보낼 때 $\{M_k\}_{TEK_k}$, $\{TEK_k\}_{KEK_k}$와 같이 사용된 키를 함께 멀티캐스팅하고, 이 메시지를 수신한 다른 멤버가 송신자로서 메시지 M_{k+1}를 보낼 때는 $\{M_{k+1}\}_{TEK_{k+1}}$, $\{TEK_{k+1}\}_{KEK_{k+1}}$ 를 보낸다. 이 때 TEK_{k+1}는 새로운 값이고, $KEK_{k+1} = TEK_k$이다. 메시지를 전달하는 경로상의 모든 라우터들은 메시지에 포함된 KEK_i를 저장한다.

이 구조에서는 키를 갱신하는 고정된 개체가 없다. 모든 멤버는 송신자가 될 수 있고, 각 송신자가 키를 생성하기 때문에 그룹 내의 모든 메시지가 순서화되어야 한다. 즉, 메시지가 $M_1, M_2, \ldots, M_f$의 순서로 전송되고 M_i의 전송시각을 S_i, M_i에 대한 모든 멤버의 수신완료 시각을 E_i라고 할 때, $E_i < S_{i+1}$ $(1 \le i < f)$이어야 한다. 새로 가입한 멤버 m이 TEK_{k+1}와 KEK_{k+1}을 분배받는다고 할 때, $KEK_{k+1} = TEK_k$이므로 m은 가입 이전의 메시지인 M_k를 알아낼 수 있지만, 다른 이전 메시지나 키는 알아낼 수 없으므로 가입 직전의 메시지와 키를 제외하면 BS를 만족한다고 할 수 있다. 그러나 TEK_{k+1}과 KEK_{k+1}를 알고 있는 멤버 m이 탈퇴할 경우에는 TEK_{k+1}로부터 $KEK_{k+2}, TEK_{k+2}, KEK_{k+3}, \ldots$의 순서로 탈퇴 이후의 모든 키를 알 수 있으므로 FS를 만족하지 못한다.

- HKTM(Height-balanced Key Tree based Management)[4]

이것은 하나의 코어를 가진 CBT를 이용한 키 관리

기법으로서 멤버의 가입과 탈퇴에 대한 효율성을 제공하기 위한 것이다. 관리자가 (2,4)-트리로 구성된 키-트리 *HKT*(Height-balanced Key Tree)를 이용함으로써, 멤버의 수를 n이라 할 때 멤버의 가입과 탈퇴에 따른 키 갱신을 $O(\log n)$에 보장한다. (2,4)-트리는 모든 단말노드가 동일한 깊이에 있고, 내부 노드의 차수 d가 $2 \le d \le 4$인 높이 균형 트리이다[10]. 멤버가 SMKD와 같은 방법으로 가입 메시지를 보내면, 이 메시지는 네트워크 구성 정보가 추가·수정되면서 관리자인 코어까지 전달된다. 관리자는 멤버의 가입 메시지의 경로를 따라 수집한 정보를 토대로 하여 *RT*(Reduced Tree)를 유지하고, 이에 따라 *HKT*의 해당 위치에 멤버를 삽입하여 갱신한다. 이렇게 함으로써 CBT상에서 임의의 하나의 링크 장애가 일어날 때 단절되는 멤버의 노드들이 *HKT*에서 연속된 위치에 있게 된다. 따라서 네트워크 장애로 일부 멤버가 탈퇴되거나 장애 복구로 재가입될 때에도 $O(\log n)$에 키 갱신을 할 수 있도록 보장해 준다.

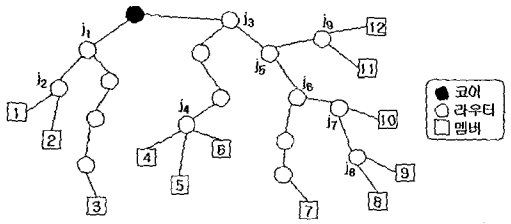

그림 2 *CBT*에서의 멤버 구성

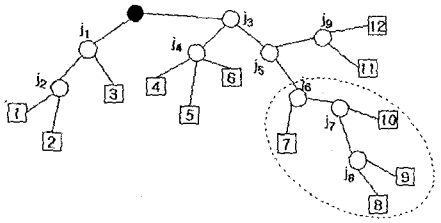

그림 3 *RT*

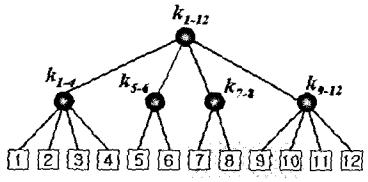

그림 4 *HKT*

그림 2는 그룹의 CBT 구성이고, 그림 3은 이에 대응되는 *RT*이다. 그림 4는 해당 키-트리로서, 단말 노드는 그룹의 멤버와 키 관리자간에 공유하는 1:1 키이고 루트 노드 k_{1-12}는 그룹키인 *TEK*이며 내부 노드들 $k_{1-4}, k_{5-6}, k_{7-8}, k_{9-12}$은 *TEK* 갱신에 사용되는 *KEK*들이다. 여기서 k_{i-j}는 $m_k (i \le k \le j)$사이에 공유되는 키이다. 멤버들은 해당 단말 노드로부터 루트까지의 경로상에 있는 모든 키를 소유해야 하므로 $O(\log n)$개의 키를 가진다. 그림 4는, 그림 3에서 노드 j_5와 j_6사이의 링크 장애가 있을 때 그룹으로부터 탈퇴되는 멤버들이 *HKT*에서 연속된 위치에 있음을 보여준다.

2.2 DEP(Dual Encryption Protocol)[5]

멤버들의 가입과 탈퇴가 허용되는 그룹에서 확장성을 제공하면서 FS와 BS를 만족시키기 위한 방법으로서 그룹을 여러 개의 서브 그룹으로 분할하고 각 서브 그룹이 독립적인 키를 공유하도록 한다. 이러한 서브 그룹형 구조를 가진 키 관리 메커니즘은 IGKMP(Intra-domain Group Key Management Protocol)[11], Kronos[12], Iolus[13], DEP(Dual Encryption Protocol)[5] 등이 있다. 특히 DEP는 다른 구조와 달리 서브 그룹 관리자를 그룹 통신 데이타로터 배제할 수 있는 특성을 갖는다. 이 절에서는 mCBT-GKM과 유사한 DEP를 소개한다.

DEP에서는 하나의 그룹 관리자 *GM*이 있고, 그룹을 서브 그룹 S_i들로 분할하여, 서브 그룹 관리자 SGM_i이 S_i의 멤버들과 로컬키 LS_i를 공유하고 이를 관리한다. *GM*과 *SGM*들은 키 LS_s를 공유하고, *SGM*들을 그룹통신으로부터 배제하기 위하여 *GM*과 멤버들 사이에 KEK_i를 공유한다. 이들의 키공유 관계는 그림 5와 같다. *GM*이 각 SGM_i에게 멀티캐스트 데이타 패킷마다 새로운 그룹키 TEK^*로 암호화된 $\{multicast-data\}_{TEK^*}$와 암호화된 TEK^*인 $\{\{TEK^*\}_{KEK}\}_{LS_i}$를 보내면, 각 SGM_i들은 $\{\{TEK^*\}_{KEK}\}_{LS_i}$를 복호화한 후, $\{\{TEK^*\}_{KEK}\}_{LS_i}$로 재암호화하여 $\{multicast-data\}_{TEK^*}$와 함께 S_i에게 보낸다. 그림 5에서 SGM_4와 같이 서브 그룹 관리자가 다른 서브 그룹의 일원일 수 있는데, S_4의 멤버들은 GM, SGM_2, SGM_4를 통해 TEK^* 수신을 하므로 세 번의 암·복호화 과정을 거쳐야만 한다. 즉, 계층구조의 깊이에 비례하여 암·복호화 비용 및 지연시간이 증가한다.

멤버가 아닌 서브 그룹 관리자가 가입하거나 탈퇴할 경우에는 LS_s만 변경하고, 멤버가 탈퇴할 경우에는 해당 서브 그룹의 키만 변경함으로써 멤버쉽 변화의 영향

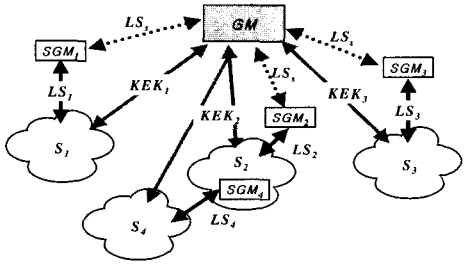

그림 5 DEP의 키공유 관계

을 지역화한다. 현재 비멤버이고 멤버 경력도 없는 SGM_i가 가입하는 경우에는 LS_i와 LS_s만 변경한다. 그러나 탈퇴한 멤버가 SGM_i로 가입하는 경우에는 LS_i와 S_i의 멤버들이 공유하던 KEK_j를 변경해 주어야 한다. LS_s, LS_i, KEK_j를 변경하려면 해당 키의 관리자가 새로운 키를 생성하여 수신자에게 일대일로 암호화하여 송신한다. 그러므로 각각 l, n_i, N_j(N_j는 KEK_j를 소유하는 멤버 수)개의 메시지가 필요하며 공개키를 이용하여 암·복호화하기 때문에 비용이 많이 든다. 가입하고자 하는 멤버는 임의의 SGM_i에 가입 요청을 하여, 자신이 받게될 KEK의 색인을 얻고, 이를 GM에게 보고한다. GM은 중복 가입 여부 등을 검사하고 인증서와 현재의 KEK_j를 멤버에게 전달한다. 가입 멤버는 GM으로부터 받은 가입 인증서를 SGM_i에 제출하여 갱신된 LS_i를 받음으로써 6 단계를 거쳐 가입 절차가 이루어진다. 또한 탈퇴한 멤버가 서브 그룹 관리자로 가입하는지를 검사하기 위해서, GM이 현재뿐 아니라 탈퇴한 과거의 멤버들에 대한 리스트도 보유해야 하는 단점이 있다.

2.3 VersaKey(Versatile group Key management) [14]

VersaKey 방식에서는 멤버들의 ID를 이용하여 키를 할당한다. 멤버들의 수가 n이라면 ID의 길이는 $\lceil \log_2 n \rceil$이므로 각 비트당 2개씩 $2 \cdot \lceil \log_2 n \rceil$개의 KEK가 존재한다. 각 멤버들은 가입시 ID와 함께 ID의 각 비트에 대응되는 $\lceil \log_2 n \rceil$개의 KEK 및 하나의 TEK를 할당 받는다. 예로, 그림 6에서 $ID=0010_{02}$인 멤버는 TEK, $KEK_{3.0}$, $KEK_{2.0}$, $KEK_{1.1}$, $KEK_{0.0}$을 소유한다. 키 관리자는 키 테이블을 소유하므로 $\lceil 2 \cdot \log_2 n \rceil + 1$개의 키만 저장하면 된다. 멤버가 가입할 경우에는 기존의 키에 일방향 해쉬함수(one-way hash function)를 사용하여 갱신한다. 즉, 새 멤버의 ID가 0010_{02} 이면, $KEK_{3.0}$, $KEK_{2.0}$, $KEK_{1.1}$, $KEK_{0.0}$를 받게 되므로, 이를 소유하고 있는 멤버들은 키들에 해쉬 함수를 적용하여 갱신함으로써 BS를 보장한다. 멤버가 탈퇴할 경우에는, 새로운 TEK^*를 그 멤버가 소유하지 않은 키들, 즉 $\overline{ID}$에 해당하는 KEK들로 각각 암호화하여 전송하고, ID에 해당하는 키들은 이 KEK들과 TEK^*로 이중 암호화하여 전송함으로써 FS를 보장한다. 이 방법에서는 그룹 관리자가 $\lceil 2 \cdot \log_2 n \rceil + 1$개의 키 저장 공간만을 필요하고 구조가 간단하기 때문에 효율적이다. 그러나 멤버의 ID가 미리 결정되어야 하기 때문에 멤버가 고정되어 있거나 미리 예상할 수 있을 때만 적합하며, ID가 서로 보수인 2명의 멤버가 결탁할 경우에는 모든 키가 노출되는 위험이 있다.

	TEK	
ID Bit #0	$KEK_{0.0}$	$KEK_{0.1}$
ID Bit #1	$KEK_{1.0}$	$KEK_{1.1}$
ID Bit #2	$KEK_{2.0}$	$KEK_{2.1}$
ID Bit #3	$KEK_{3.0}$	$KEK_{3.1}$
	value=0	value=1

그림 6 VersaKey의 키 테이블

3. 다중 CBT를 이용한 2계층 관리 구조 및 프로토콜(mCBT-GKM) 제안

멤버쉽의 변동이 빈번한 경우에는 이로 인한 그룹키 갱신이 잦게 되어 효율성이 저하된다. 그러므로 확장성을 제공하면서 멤버쉽 변동으로 인한 키 갱신의 범위를 줄이는 것이 효율적이다. 이러한 효율성 제공을 위한 방안으로서 DEP가 제안되었다. 이 장에서는 DEP의 단점을 보완하기 위하여 하나의 주코어와 그룹 정책 및 효율적 방법에 의해 선정된[7] 여러 개의 부코어를 가진 CBT에 접목한 구조와 키 갱신 프로토콜인 mCBT-GKM (multi-Core Based Tree-Group Key Management)를 제안한다.

3.1절에서는 mCBT-GKM의 관리 체계와 구성원간의 키 공유관계를 기술하고, 3.2절에서는 각 서브 그룹키의 관리에 사용될 메커니즘을 기술한다. 3.3절에서는 가입과 탈퇴에 따른 키 분배 및 갱신 프로토콜을 기술한다.

3.1 관리 구조

mCBT-GKM에서는 주코어를 GM으로 설정하고, 각 부코어들을 SGM_i ($1 \le i \le l$)로 설정한다. 멤버의 가입은 CBT 프로토콜을 따른다. 즉, 가입하고자 하는 멤버 m

은 IGMP[16]를 통하여 로컬 라우터에게 그룹으로의 가입 요청을 하고, 로컬 라우터는 가장 가까운 부코어인 SGM_i에게 전달함으로써[6], m은 SGM_i가 관리하는 서브 그룹 S_i에 소속된다. 그림 7은 이러한 방법으로 형성된 서브 그룹의 예이다. 멤버는 하나의 서브 그룹에만 소속될 수 있고 SGM은 정적으로 선정된 노드들로서 다른 서브 그룹에 소속되지 않으므로, 그룹의 멤버들은 l개의 서브 그룹으로 분할(partition)된다. 따라서 그림 8과 같이 GM을 루트로 하고 SGM들을 GM의 자식노드로 하며 각 서브 그룹의 멤버들은 해당 SGM의 자식노드가 되는, 깊이(depth)가 2인 트리로서 그룹의 계층을 표현할 수 있다. LS_s와 LS_i의 공유관계는 DEP와 동일하지만, GM과 S_i의 멤버들 사이에 두 개의 키 KEK_{ij} $(1 \le i \le l, j = 1,2)$를 공유하도록 한다. KEK_{i1}은 멤버로 등록된 S_i의 모든 멤버들이 공유하며, KEK_{i2}는 SGM_i가 아닌 S_i의 멤버만이 공유한다. KEK_{i1}은 TEK의 암호화에 사용되며, KEK_{i2}는 KEK_{i1}의 갱신에 사용된다. 그림 8은 그림 7과 같이 형성된 서브 그룹에서 SGM_3이 멤버일 경우의 키 공유 관계를 보여준다. GM은 멤버쉽 관리를 하지 않으며, TEK, KEK_{ij}와 LS_s의 관리 및 멀티캐스트 데이타의 전송을 담당한다. SGM_i들은 서브 그룹 S_i의 멤버쉽 관리와 서브 그룹키 LS_i의 관리를 담당한다. 이들은 멤버 가입을 하지 않는 한, 멀티캐스트 데이타에 대한 수신 권한을 가지고 있지 않다. SGM이 아닌 모든 멤버는 S_i에 가입되면서 SGM_i로부터 LS_i를 할당받고, GM으로부터 KEK_{i1}와 KEK_{i2}를 할당받는다. GM은 멤버쉽 관리를 하지 않으므로, 현재나 과거의 멤버 리스트를 유지할 필요가 없고 SGM들의 리스트만 유지하면서 그들이 멤버인지 아닌지만 인식한다. 대신 SGM들이 해당 서브 그룹에 속한 멤버들의 리스트를 유지한다.

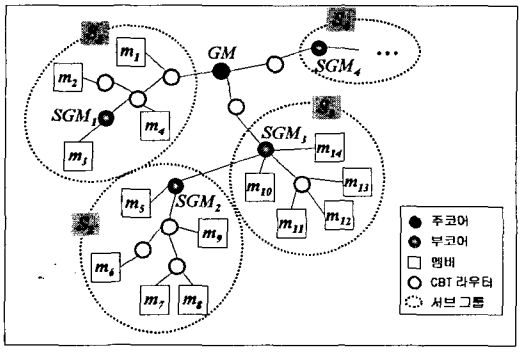

그림 7 CBT에서의 서브 그룹

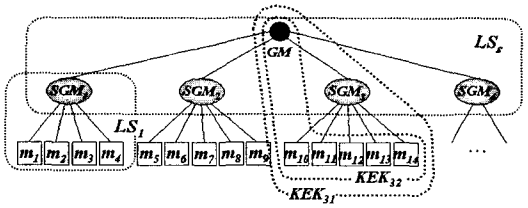

그림 8 키 공유 관계도

멀티캐스트 데이타는 패킷마다 새로운 TEK^*로 암호화되어 전송되며, TEK^*는 멀티캐스트 데이타와 함께 다음과 같이 두 단계를 거쳐 각 멤버들에게 전송된다. 따라서 모든 멤버는 2번의 암·복호화 과정만 거쳐 그룹키를 수신할 수 있다.

$GM \rightarrow SGM_i$ $(1 \le i \le l)$: $\{\{TEK^*\}_{KEK_{i1}}\}_{LS_s}$
$SGM_i \rightarrow m$ $(1 \le i \le l,$ 모든 $m \in S_i)$: $\{\{TEK^*\}_{KEK_{i1}}\}_{LS_i}$

3.2 키 관리 프로토콜

멤버쉽 변동이 있을 때마다 그룹의 모든 멤버가 키를 갱신한다거나 DEP에서와 같이 키 관리자가 멤버들에게 일대일로 새로운 키를 분배한다면, 멤버쉽 변동이 빈번한 대규모 그룹에서 매우 비효율적일 것이다. 제안하는 구조에서는 멤버의 가입과 탈퇴가 자유롭고 규모가 큰 그룹을 지원하기 위하여 SGM_i가 HKTM을 이용하여 LS_i를 관리한다. 즉, 멤버들이 가입 메시지를 통하여 네트웍 구성 정보를 SGM_i에 전달하고 SGM_i는 이를 이용하여 RT_i와 LS_i를 관리하기 위한 HKT_i를 유지한다. LS_i는 HKT_i의 루트이며 S_i의 각 멤버 x는 LS_i와 HKT_i에서 $O(\log n_i)$만큼의 보조키들을 소유한다. 각 서브 그룹 S_i는 SGM_i을 코어로 하는 CBT이기 때문에 서브 그룹 내에 효율적으로 멀티캐스팅할 수 있으므로 LS_i 갱신을 위한 메시지의 전송은 그룹 중심의 전송 방법을[1] 이용하여 멀티캐스팅할 수 있다.

SGM들은 그룹 소유자 혹은 멀티캐스트 데이타 제공자가 관리의 효율성이나 지역성, 혹은 경제성 등을 고려하여 정책적으로 미리 선정한 노드들이다. 따라서 추후 추가될 SGM까지 고려하여 그 수를 추정할 수 있으며 멤버에 비하여 수가 많지 않다. 또한 멀티캐스팅 데이타를 접근할 수 있는 그룹의 멤버가 아니기 때문에 가입과 탈퇴가 없으며 결탁의 가능성도 매우 낮다. 그러므로 이들 그룹을 정적인 그룹으로 간주하여 SGM의 가입과 탈퇴에 따른 키 갱신은 고려하지 않고, 안전성을 위하여 기존의 키로 새로운 키를 암호화하여 배포하는 주기적인 키 갱신만 수행하도록 할 수 있다.

그러나 네트워 장애 등으로 인한 변동을 고려한다면 각 *SGM*들에게 사전에 정의한 *ID*를 할당하고, 멤버의 수를 예상할 수 있고 결탁의 가능성이 낮은 그룹의 키 관리에 적합한 VersaKey 방식을 이용하는 것이 바람직할 것이다. 즉, LS_s를 *GM*와 *SGM*들로 구성된 관리자 그룹의 그룹키인 *TEK*로서 관리하며 *GM*은 LS_s와 $2 \cdot \lceil \log l \rceil$ 개의 *KEK*들을 관리하고, *SGM*들은 LS_s와 $\lceil \log l \rceil$ 개의 *KEK*들을 소유한다.

3.3 가입 및 탈퇴 프로토콜

멤버의 가입과 탈퇴가 일어날 때는 변경되는 멤버가 소속된(혹은 소속될) 서브 그룹 S_i의 LS_i만 갱신한다. 이것은 멤버의 변동으로 인한 키 갱신의 범위를 하나의 서브 그룹으로 국한시킴으로써 $O(\log n_i)$개의 메시지만으로 멀티캐스트 데이타의 보안을 유지할 수 있도록 한다. 한편, *SGM*들은 그룹 형성시에 관리자에 의해 지정되는 노드들이기 때문에 임의의 멤버가 *SGM*으로 가입할 수는 없지만, 기존의 *SGM*이 멤버로서 가입하거나 탈퇴할 수는 있다. 멤버였던 어떤 SGM_i가 탈퇴하고 서브 그룹 관리자의 역할만 하고자 할 경우에는 이 SGM_i로부터 멀티캐스트 데이타에 대한 접근을 막아야 하므로 SGM_i가 알고 있던 KEK_{i1}을 변경해 주어야 한다.

*SGM*이나 일반 멤버들의 가입과 탈퇴에 따른 LS_i와 KEK_{i1} 및 KEK_{i2}의 갱신 프로토콜에서 사용되는 표기와 프로토콜들은 다음과 같다.

$join\text{-}req$	그룹 가입 요청 토큰
$id_x, cert_x, addr_x$	x의 식별자, 공개키 인증서, 주소
$nonce_x$	x가 생성한 난수
LS_{ix}	LS_i를 루트로 하는 HKT_i에서 멤버 x가 소유해야 (혹은 갱신해야) 할 서브 그룹키 집합
LS^*_{ix}	갱신된 LS_{ix}
$\{LS^*_{ix}\}_{LS_{ix}}$	HTKM의 방식으로 LS_{ix}를 이용하여 LS^*_{ix}를 암호화한 메시지

■ *SGM*이 아닌 멤버 m이 S_i에 가입할 경우

새로운 멤버 m이 그룹에 가입할 때는 소속되는 S_i의 키인 LS_i를 갱신하여야 하고, m은 LS^*_i와 KEK_{i1}과 KEK_{i2}를 할당받아야 한다. 처리 절차는 다음과 같으며 프로토콜은 그림 9와 같다.

1. m은 가입 요청 메시지를 로컬 라우터에게 전송하고, 로컬 라우터는 CBT 프로토콜을 이용하여 가장 가까운 SGM_i에게 전달한다. 가입 요청 메시지는 m이 서명한 가입 요청 토큰 $join-req$, m의 식별자 id_m과 주소 $addr_m$, 재생 공격 방지를 위한 난수 $nonce_m$으로 이루어진다. 또한 서명 검증에 사용될 m의 공개키에 대한 인증서 $cert_m$도 함께 전송한다.

1	$m \rightarrow SGM_i: \{join\text{-}req, id_m, addr_m, nonce_m\}^{R_m}, cert_m$
2	$SGM_i \rightarrow m: \{\{LS^*_{im}\}_{U_m}, nonce_m, nonce_{SGM_i}\}^{R_{SGM}}, cert_{SGM_i}$
3	$SGM_i \rightarrow m'$ (모든 $m' \in S_i$): $\{LS^*_{im}\}_{LS_m}$
4	$SGM_i \rightarrow GM: \{\{join\text{-}req, id_m, addr_m, nonce_m\}^{R_m}, id_{SGM_i}, nonce_{SGM_i}\}^{R_{SGM}}, cert_m, cert_{SGM_i}$
5	$GM \rightarrow m: \{\{join\text{-}req, id_m, addr_m, nonce_m\}^{R_m}, nonce_{GM}, \{KEK_{i1}, KEK_{i2}\}_{U_m}\}^{R_{GM}}, cert_{GM}$

그림 9 *SGM*이 아닌 멤버의 가입 프로토콜

2. m의 가입 요청 메시지를 수신한 SGM_i는 멤버의 가입 여부를 결정하고, 이를 승인할 경우에는 LS_i를 갱신하고 m에게 LS^*_i와 보조키들을 m의 공개키로 암호화하여 전송한다. 만약 승인하지 않을 경우에는 이 단계에서 종료한다.

3. SGM_i는 기존의 S_i의 멤버들에게도 HKTM을 이용하여 LS_i 갱신 메시지를 보낸다. 기존 멤버 m'이 *HKT*에서 가지고 있던 보조키들 $LS_{im'}$로 HKT^*_i의 $LS^*_{im'}$를 암호화하여 전송한다.

4. SGM_i은 *GM*에게 새로운 멤버 m의 가입을 승인했음을 알린다. 이 메시지는 m이 서명한 가입 요청 메시지와 SGM_i의 id, 재전송 공격을 막기 위하여 SGM_i가 생성한 난수 $nonce_{SGM_i}$로 구성된다. 이 메시지는 SGM_i의 공개키로 서명하고, m과 SGM_i의 공개키에 대한 인증서를 함께 전송한다.

5. *GM*은 SGM_i로부터 받은 가입 승인 메시지의 서명을 검증하고, m이 *SGM*이 아님을 확인한다. 확인에 성공하면 m이 서명한 가입 요청 메시지, *GM*이 생성한 $nonce_{GM}$ 및 m의 공개키로 암호화한 현재의 KEK_{i1}과 KEK_{i2}를 m에게 전송한다.

■ $SGM_i(=m)$이 S_i의 멤버로 가입할 경우

SGM_i가 멤버로 가입할 경우에는 KEK_{i1}을 갱신하고, 갱신된 KEK^*_{i1}를 SGM_i 및 기존 S_i의 멤버들에게 배포하여야 한다. 처리 절차는 다음과 같으며 프로토콜은 그림 10과 같다.

1. SGM_i는 가입 요청 메시지를 작성하고 서명하여 *GM*에게 보낸다.

2. *GM*은 수신한 가입 요청 메시지의 id_m으로부터 이 메시지의 송신자가 SGM_i임을 확인하고, SGM_i가 보낸

1	$SGM_i \rightarrow GM : \{\{join\text{-}req, id_m, addr_m, nonce_m\}^{R_m}\}, cert_m$
2	$GM \rightarrow m : \{\{join\text{-}req, id_m, addr_m, nonce_m\}^{R_m}, nonce_{GM}, \{KEK_{i1}^*\}_{U_m}\}^{R_{GM}}, cert_{GM}$
3	$SGM_i \rightarrow S_i : \{KEK_{i1}^*\}_{LS_i}$

그림 10 SGM_i의 멤버 가입 프로토콜

가입 요청 메시지와 $nonce_{GM}$, GM 그리고 SGM_i의 공개키로 암호화한 새로운 KEK_{i1}^*을 의 공개키 인증서와 함께 SGM_i에게 전송한다. 이 때 가입하는 멤버가 SGM이기 때문에 KEK_{i2}는 전송하지 않는다.

3. SGM_i는 수신한 새로운 KEK_{i1}^*를 LS_i로 암호화하여 S_i의 멤버들에게 전송한다.

■ SGM이 아닌 S_i의 멤버 m이 탈퇴할 경우

탈퇴하는 m이 소속된 S_i의 LS_i만 변경한다. LS_i^*는 가입할 때와 마찬가지로 HKTM을 이용하여, m을 제외한 나머지 S_i의 멤버들에게 전송한다. 프로토콜은 그림 11과 같다.

1	$SGM_i \rightarrow m'$ (모든 $m' \in S_i$ & $m' \neq m$) : $\{LS_{im'}^*\}_{LS_{m'}}$

그림 11 SGM이 아닌 멤버의 탈퇴 프로토콜

■ $SGM_i(=m)$이 멤버를 탈퇴할 경우

SGM_i가 탈퇴할 경우에는 SGM_i가 알고 있던 KEK_{i1}을 갱신하여야 한다. 처리 절차는 다음과 같으며 프로토콜은 그림 12와 같다.

1. GM은 새로운 KEK_{i1}^*를 SGM_i를 제외한 S_i의 멤버만이 알고 있는 KEK_{i2}로 암호화하고 LS_s로 다시 암호화하여 SGM_i에게 전송한다.

2. SGM_i는 수신한 메시지를 LS_s로 복호화한 후, LS_i로 재 암호화하여 LS_i의 멤버들에게 전송한다.

1	$GM \rightarrow SGM_i : \{\{KEK_{i1}^*\}_{KEK_{i2}}\}_{LS_s}$
2	$SGM_i \rightarrow S_i : \{\{KEK_{i1}^*\}_{KEK_{i2}}\}_{LS_i}$

그림 12 SGM_i의 멤버 탈퇴 프로토콜

KEK_{i2}는 서브 그룹 관리자의 멤버 탈퇴시에만 사용되는 키로서 이를 변경시키는 이벤트가 없다. 서브 그룹 관리자의 멤버 가입과 탈퇴는 빈번하게 일어나는 이벤트가 아니기 때문에 빈번한 사용으로 인한 키의 노출 가능성은 높지 않으나 오랜 기간 동안 변경되지 않음으로써 발생할 수 있는 노출의 가능성을 배제하기 위하여 주기적으로 변경해 줄 수 있다. 처리 절차는 다음과 같으며 프로토콜은 그림 13과 같다.

1. GM은 새로운 KEK_{i2}^*를 SGM_i를 제외한 S_i의 멤버만이 알고 있는 KEK_{i2}로 암호화하고 LS_s로 다시 암호화하여 SGM_i에게 전송한다.

2. SGM_i는 수신한 메시지를 LS_s로 복호화한 후, LS_i로 재 암호화하여 LS_i의 멤버들에게 전송한다.

1	$GM \rightarrow SGM_i : \{\{KEK_{i2}^*\}_{KEK_{i2}}\}_{LS_s}$
2	$SGM_i \rightarrow S_i : \{\{KEK_{i2}^*\}_{KEK_{i2}}\}_{LS_i}$

그림 13 KEK_{i2}의 주기적 갱신 프로토콜

4. 안전성 및 효율성 분석

본 논문에서 제안한 mCBT-GKM은 DEP의 구조를 다중 코어를 가진 CBT에 접목시킨 구조로서, 기존의 CBT 기반의 프로토콜들과 DEP에 비하여 개선된 안전성 및 효율성을 가진다. 본 장에서는 mCBT-GKM의 안전성 및 효율성을 기존 프로토콜들과 비교한다.

4.1 안전성

GKS는 TEK 전송시 사용되는 공개키 암호 시스템이나 비밀키 시스템의 안전성에 의존하므로 이 시스템들이 안전하다는 조건하에 보장된다고 할 수 있다. 그러나 멤버의 가입과 탈퇴가 허용되는 동적인 그룹에서 필수적인 FS와 BS는 모든 프로토콜이 만족하지는 못한다. SMKD는 멤버쉽 변동에도 키 갱신이 이루어지지 않기 때문에 두 가지 모두 만족하지 못하며, KL은 TEK와 KEK의 연결 관계로 인하여 FS를 만족하지 못한다. DEP에서 TEK는 LS_s와 KEK_i로 암호화 되거나 LS_j와 KEK_i로 암호화되어 전송되므로 TEK를 복호화하려면 두 메시지 중 하나를 복호화할 수 있어야 한다. LS_s와 KEK_i를 모두 아는 참가자는 GM뿐이며, LS_j와 KEK_i를 모두 아는 참가자는 현재의 S_j의 멤버뿐이다. 즉, 탈퇴 이후의 TEK나 가입 이전의 TEK를 알 수 없으므로 FS와 BS를 보장한다.

mCBT-GKM의 각 키는 시간에 따라 소유자가 달라질 수 있다. 하나의 서브 그룹 S_i에 대하여 고려해 볼 때, 키를 소유하는 참가자는 다음과 같다.

- LS_s : VersaKey 방식에 의해 SGM의 변동이 있을 때마다 변경되어 FS와 BS를 만족하므로 현재의

LS_s는 GM과 현재의 SGM들만이 소유한다.

• LS_i : HKTM 방식에 의해 S_i의 멤버쉽 변동이 있을 때마다 변경되어 FS와 BS를 만족하므로 현재의 LS_i는 SGM_i과 S_i의 현재 멤버만이 소유한다.

• KEK_{l2} : 그룹이 형성된 시각을 t_0라 하고, KEK_{l2}가 주기적으로 갱신된 시각을 $t_j(j>0)$하자. 또한 $t_j \le t < t_{j+1}$ $(j \ge 0)$인 구간을 P_j라 하고, t_j에 갱신되어 P_j 동안 사용되는 KEK_{l2}를 KEK_{l2}^j라 하자. $t_j \le t < t_{j+1}$에서의 S_i를 S_i^t라 하고($S_i^{t_0} = \phi$), $t_j \le t' < t$동안 $S_i^{t'}$에서 탈퇴한 SGM_i가 아닌 멤버들의 집합을 $\widetilde{S}_i^t$라 하자(그림 14 참조). KEK_{l2}를 소유하는 참가자는 <정리 1>과 같다.

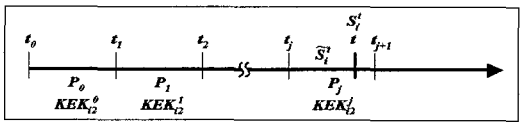

그림 14 시간에 따른 KEK_{l2}의 소유자 변화

<정리 1> $t_j \le t < t_{j+1}$ $(j \ge 0)$인 시각 t에서, KEK_{l2}^j는 GM과 SGM_i가 아닌 $S_i^t \bigcup \widetilde{S}_i^t$의 멤버만이 소유할 수 있다.

<증명> 수학적 귀납법에 의해 증명한다.

① $j=0$일 때:

t_0에서 GM이 KEK_{l2}^0을 생성한 후, S_i^0에 가입한 멤버가 SGM_i이 아닐 경우에만 KEK_{l2}^0를 분배하고 t_1까지 멤버의 가입과 탈퇴가 일어나더라도 변경되지 않으므로, KEK_{l2}^0는 $t_0 \le t < t_1$인 시각 t에서, $x = GM \vee x \in S_i^t \bigcup \widetilde{S}_i^{t'} \wedge x \neq SGM_i$인 x만이 소유할 수 있다.

② $j=k\,(k>0)$일 때:

$j=k-1\,(k>0)$일 때: <정리 1>이 성립한다고 가정하자.

$t_{k-1} \le t < t_k$인 t에서 LS_s와 LS_i 및 SGM 집합을 각각 LS_s^t, LS_i^t, SGM^t라 할 때, $t=t_k$에서의 키 갱신은 $GM \rightarrow SGM_i : \{\{KEK_{l2}^k\}_{KEK_{l2}^{k-1}}\}_{LS_i^t}$와 $SGM_i \rightarrow S_i : \{\{KEK_{l2}^k\}_{KEK_{i2}^{k-1}}\}_{LS_i^t}$를 통하여 이루어진다. $j=k-1\,(k>0)$일 때 <정리 1>이 성립한다고 가정하였으므로, 첫 번째 메시지는 $(x = GM \vee x \in S_i^t \bigcup \widetilde{S}_i^{t'} \wedge x \neq SGM_i) \wedge (x = GM \vee x \in SGM^t)$인 x만이 복호화할 수 있다. 따라서 GM만이 첫 번째 메시지로부터 KEK_{l2}^k를 알 수 있다. 두 번째 메시지도 가정에 의하여 $(x = GM \vee x \in S_i^t \bigcup \widetilde{S}_i^{t'} \wedge x \neq SGM_i) \wedge (x = SGM_i \vee x \in S_i^t)$인 x만이 복호화할 수 있으므로, GM이나 SGM_i가 아닌 S_i^t의 멤버만이 두 번째 메시지로부터 KEK_{l2}^k를 알 수 있다. 그러므로, t_k에서 갱신된 KEK_{l2}^k는 GM이나 SGM_i가 아닌 S_i^t의 멤버만이 소유하게 된다. 이 키는 t_{k+1}까지 변경되지 않으므로, $t_k \le u < t_{k+1}$에 대하여 $x \in \widetilde{S}_i^u$도 소유할 수 있다. 즉, KEK_{l2}^k는 $t_k \le t < t_{k+1}$인 시각 t에서 $x = GM \vee x \in S_i^t \bigcup \widetilde{S}_i^t \wedge x \neq SGM_i$인 x만이 소유할 수 있다.

③ ①,②에 의하여 <정리 1>이 성립한다. ■

• KEK_{i1} : KEK_{i1}의 소유하는 참가자는 <정리 2>와 같다.

<정리 2> KEK_{i1}은 GM과 S_i의 현재 멤버들만이 소유할 수 있다.

<증명> KEK_{i1}는 SGM_i가 멤버로 가입하거나 탈퇴할 경우에 갱신 및 분배된다.

① SGM_i가 $t \in P_j$에 멤버를 탈퇴할 경우:

이 때의 LS_s와 LS_i 및 SGM 집합을 각각 LS_s^t, LS_i^t, SGM^t라 하면, $GM \rightarrow SGM_i : \{\{KEK_{i1}^*\}_{KEK_{i2}}\}_{LS_i^t}$와 $SGM_i \rightarrow S_i : \{\{KEK_{i1}^*\}_{KEK_{i2}}\}_{LS_i^t}$를 통하여 갱신된다. 첫 번째 메시지는 <정리 1>에 의하여 $(x = GM \vee x \in S_i^t \bigcup \widetilde{S}_i^t \wedge x \neq SGM_i) \wedge (x = GM \vee x \in SGM^t)$인 x만이 복호화 할 수 있으므로 GM만이 KEK_{i1}^*를 알 수 있다. 두 번째 메시지는 $(x = GM \vee x \in S_i^t \bigcup \widetilde{S}_i^t \wedge x \neq SGM_i) \wedge (x = SGM_i \vee x \in S_i^t)$인 x만 복호화할 수 있으므로 $(x \in S_i^t \wedge x \neq SGM_i)$인 x만이 KEK_{i1}^*를 알 수 있다. 그러므로, 키 갱신 후에는 SGM_i가 제외된 현재 S_i의 멤버들만이 KEK_{i1}을 소유하게 된다.

② SGM_i가 $t \in P_j$에 멤버로 가입할 경우:

$GM \rightarrow SGM_i : \{KEK_{i1}^*\}_{U_{SGM}}$와 $SGM_i \rightarrow S_i^t : \{KEK_{i1}^*\}_{LS_i^t}$에 의해 갱신되므로, $(x = SGM_i \vee x \in S_i^t)$인 x만 키를 알 수 있다. 즉, SGM_i가 포함된 S_i^t의 멤버들만이 KEK_{i1}^*를 알 수 있다.

③ ①,②에 의하여 <정리 2>가 성립한다. ■

• TEK : TEK를 소유하는 참가자는 <정리 3>과 같다.

<정리 3> TEK는 모든 $i\,(1 \le i \le l)$에 대하여 GM과 S_i의 현재 멤버들만이 소유할 수 있다.

<증명> TEK는 매 데이타 패킷마다 새로운 값이 전달된다. $t \in P_j$일 때, $GM \rightarrow SGM_i$ $(1 \le i \le l) : \{\{TEK^*\}_{KEK_{i1}^*}\}_{LS_i^t}$와 $SGM_i \rightarrow S_i^t : \{\{TEK^*\}_{KEK_{i1}^*}\}_{LS_i^t}$를 통하여 갱신된다. <정리 2>에 의하여, 첫 번째 메시지는 $(x = GM \vee x \in S_i^t) \wedge (x = GM \vee x = SGM^t)$인 x만 복호화할 수 있고, 두 번째 메시지는 $(x = GM \vee x \in S_i^t) \wedge (x = SGM_i \vee x \in S_i^t)$인 x만 복호화할 수 있다. 따라서 <정리 3>이 성립한다. ■

그러므로, <정리 3>에 의하여 mCBT-GKM은 TEK에 대한 FS와 BS를 만족한다.

각 키의 갱신을 유발시키는 이벤트와 키 갱신 메시지 및 임의의 시각에서 이 키를 소유할 수 있는 참가자를 표 1에 요약하였으며, 표 2에 SMKD, KL, DEP 및

표 1 각 키의 갱신 및 접근이 가능한 참가자

키	키 갱신 유발 사건	키 갱신 메시지	키를 소유할 수 있는 참가자 x
LS_s	SGM의 가입 · 탈퇴	VersaKey	$x = GM \bigvee x \in \{SGM_i \mid 1 \le i \le l\}$
LS_i	S_i의 멤버 가입 · 탈퇴	HKTM	$x = SGM_i \bigvee x \in S_i$
KEK_{i2}	주기적	$GM \rightarrow SGM_i : \{\{KEK'_{i2}\}_{KEK_{i2}}\}_{LS_s}$ $SGM_i \rightarrow S_i : \{\{KEK'_{i2}\}_{KEK_{i2}}\}_{LS_i}$	$x = GM \bigvee x \in S_i \bigcup \tilde{S}_i \bigwedge x \neq SGM_i$
KEK_{i1}	SGM_i의 멤버 탈퇴	$GM \rightarrow SGM_i : \{\{KEK'_{i1}\}_{KEK_{i2}}\}_{LS_s}$ $SGM_i \rightarrow S_i : \{\{KEK'_{i1}\}_{KEK_{i2}}\}_{LS_i}$	$x = GM \bigvee x \in S_i$
	SGM_i의 멤버 가입	$GM \rightarrow SGM_i : \{KEK'_{i1}\}_{U_{SGM_i}}$ $SGM_i \rightarrow S_i : \{KEK'_{i1}\}_{LS_i}$	
TEK	매 데이타 패킷마다	$GM \rightarrow SGM_i \ (1 \le i \le l) : \{\{TEK'\}_{KEK_{i1}}\}_{LS_s}$ $SGM_i \rightarrow S_i : \{\{TEK'\}_{KEK_{i1}}\}_{LS_i}$	$x = GM \bigvee x \in S_i (1 \le i \le l)$

표 2 안전성 비교

비교 항목	SMKD	KL	DEP	mCBT-GKM
TEK에 대한 GKS	○	○	○	○
TEK에 대한 FS	×	×	○	○
TEK에 대한 BS	×	○	○	○

mCBT-GKM의 안전성을 비교하였다.

4.2 효율성

SMKD와 KL 및 HKTM 방식은 중앙집중형 구조로서 서브그룹형 구조는 구조의 단순성을 희생하고 효율성을 높인 것이므로, mCBT-GKM과 동일한 항목으로 효율성을 비교한다(표 3 참조). 이 절에서는 동일한 구조와 안전성을 갖는 DEP와의 효율성을 비교하고, 이를 표 3에 요약하였다. DEP는 멤버 탈퇴시 TEK를 제외한 모든 키의 갱신을 수신자와의 일대일 채널을 통해 전송하기 때문에 LS_s, LS_i, KEK_j 외에도 각 수신자의 공개키를 유지하여야 한다. 결과적으로 mCBT-GKM에서는 DEP에 비하여 멤버나 서브 그룹 관리자가 유지해야 하는 키의 수는 증가하지만, 그룹 관리자가 유지해야 하는 키의 수는 감소하였다. 또한 2계층 관리구조를 갖도록 함으로써 TEK의 갱신 메시지의 암 · 복호화 수와 전달 지연을 2로 줄였다. DEP에서는 KEK_i의 갱신을 위해 그룹 관리자가 이를 소유한 각 멤버들의 공개키로 암호화하여 전송하였으나, mCBT-GKM에서는 SGM들이 고정적이라는 특성을 이용하여 KEK_i대신 KEK_{i1}과 KEK_{i2}를 사용하여 매우 간단하게 갱신할 수 있는 프로토콜을 제안하였다. DEP에서의 LS_s와 LS_i 갱신도 수신자의 공개키로 암호화하여 일대일 통신으로 이루어진다. 제안한 구조에서는 SGM들이 고정적이고 각 S_i들이 하나의 CBT 구조를 이루도록 구성하였으므로, 이러한 특성에 적합한 VersaKey와 HKTM 방식을 적용함으로써 효율적으로 LS_s와 LS_i를 관리할 수 있도록 하였다.

DEP에서는 그룹 관리자가 멤버쉽 관리를 하기 때문에, SGM들 뿐 아니라 GM도 멤버들에 대한 리스트를 유지하여야 한다. 또한 멤버가 가입하여 필요한 키를 전송 받기 위해서는 SGM_i, GM, SGM_i과 순서대로 메시지를 교환함으로써 6단계의 처리 단계가 필요하였다. 그러나 mCBT-GKM에서는 서브 그룹 관리자가 멤버쉽 관리를 하도록 함으로써 GM이 현재의 멤버나 탈퇴한 멤버 리스트를 유지할 필요가 없으며, 멤버의 가입 절차를 4단계로 줄였다. 또한 갱신 비용이 큰 KEK의 갱신의 효율성을 위하여, DEP에서는 GM이 탈퇴한 멤버들의 리스트를 저장하여야 하지만 mCBT-GKM에서는 현재의 SGM들이 멤버인지 여부만 유지한다.

DEP에서는 멤버들이 임의의 서브 그룹에 소속되기 때문에 멀티캐스팅 경로와는 무관하게 유지될 뿐 아니라 일대일로 키 갱신 메시지를 전송한다. 반면, mCBT- GKM에서는 CBT 방식을 따라 서브 그룹에 소속되므로 LS_i의 갱신 메시지도 CBT 경로를 따라 멀티캐스트될 수 있다.

5. 결 론

동적 그룹에서 멤버쉽 변동으로 인한 그룹키 갱신의 범위를 줄이는 것은 대규모 그룹으로의 확장에 매우 중요한 요소이다. DEP는 확장성 제공을 위해 서브 그룹 구조를 가지면서 서브 그룹 관리자의 멀티캐스트 데이타 접근을 통제할 수 있는 구조이다. 그러나 임의의 노드가 서브 그룹 관리자가 될 수 있도록 함으로써 그룹 관리자와 멤버간에 공유하는 KEK의 갱신을 1:1로 수행해야

표 3 효율성 분석

비교 항목	DEP	mCBT-GKM
TEK 갱신 메시지의 암·복호화 단계 수	$O(l)$	2
LS_i 갱신 메시지 수	n_i	$O(\log n_i)$
LS_s 갱신 메시지 수	l	$2\lceil \log l \rceil$
$KEK_i(KEK_n)$ 갱신 메시지 수	n_i	2
S_i에 속한 멤버들이 유지하는 키의 수	2	$O(\log n_i)+2^{<1>}$
SGM_i가 유지하는 키의 수	$2+n_i^{<2>}$	$O(n_i)+\lceil \log l \rceil +1^{<3>}$
*GM*이 유지하는 키의 수	$O(l)+1+l+n^{<4>}$	$2l+2\lceil \log l \rceil +1^{<5>}$
*GM*이 유지하는 멤버 정보	멤버 리스트, *SGM* 리스트, 탈퇴한 멤버 리스트	*SGM* 리스트
멤버 가입 절차의 단계 수	6	4
멀티캐스트 배달 경로를 이용한 키 갱신 메시지의 멀티캐스팅	×	○

<1> HKT_{ik}, KEK_{ij}
<2> LS_s, LS_i, S_i에 속한 멤버들의 공개키
<3> HKT_i, LS_s키-테이블에서의 해당키
<4> KEK_i, LS_s, *SGM*들의 공개키, 멤버들의 공개키
<5> KEK_{ij}, LS_s키-테이블

하는 비효율성을 가진다. 한편 CBT는 수신자 중심의 멀티캐스팅 프로토콜로서 그룹키 관리가 용이하나 기존에 제안된 CBT 기반의 그룹키 관리 프로토콜들은 중앙 집중형으로써 "1 affects n" 확장성을 제공하지 못하고 FS 및 BS를 모두 만족하지는 못한다. 본 논문에서는 다중 코어를 가진 CBT에서 부코어들이 정적으로 선정된다는 점에 착안하여, DEP를 CBT에 적용함으로써 다음과 같은 특성을 갖는 구조 및 프로토콜을 제안하였다.

- 그룹 관리자가 CBT의 주코어가 되고, 선정된 부코어들이 서브 그룹 관리자의 역할을 수행하도록 함으로써 2계층의 서브 그룹형 구조를 가지도록 한다.
- 부코어는 멤버로 가입하지 않는 한, 그룹 데이타를 수신할 수 없다.
- 확장성 제공을 위하여 멤버의 가입과 탈퇴시, 서브 그룹키인 LS_i만 갱신한다.
- 서브 그룹 관리자가 멤버쉽 관리를 담당한다.
- 멤버들의 가입 및 탈퇴에 따른 그룹키 *TEK*의 GKS, FS 및 BS를 제공한다.

제안한 구조는 DEP에 비하여 다음과 같은 개선점을 가진다.

- 2계층 구조로 인하여, 키 갱신 메시지의 암·복호화 단계 및 전송지연을 2로 줄였다.
- 그룹 관리자와 멤버가 2개의 *KEK*를 공유하도록 함으로써, 서브 그룹 관리자의 멤버 탈퇴시 *KEK*를 2개의 메시지로 갱신할 수 있다.
- 서브 그룹 관리자가 멤버쉽 관리를 하도록 함으로써, 멤버 가입시 그룹 관리자로부터 가입 승인을 얻기 위한 절차를 없앴으며 그룹 관리자가 유지해야 하는 멤버 리스트를 제거하였다.
- 각 서브 그룹은 부코어를 중심으로 하는 하나의 CBT를 형성하므로, CBT에 적합한 서브 그룹키 관리 메커니즘인 HKTM을 적용할 수 있다.
- 서브 그룹 관리자가 정적인 그룹으로 제한되므로, 소유하는 키의 개수가 적은 VersaKey와 같은 메커니즘을 이용하거나 주기적 키 갱신 방법만 사용할 수 있다.
- 각 서브 그룹은 하나의 CBT를 이루므로, 서브 그룹 관리자로부터의 메시지는 CBT를 이용하여 효율적으로 서브 그룹에 멀티캐스트 될 수 있다.

참 고 문 헌

[1] C. K. Wong, M. Gouda, S. S. Lam, "Secure Group Communications using Key Graphs," Proc. of ACM SIGCOMM, 1988.
[2] T. Hardjono, B. Cain and B. Doraswamy, "A Framework for Group Key Management for Multicast Security," IETF Internet Draft, 2001.
[3] A. Perrig, D. Song and J. D. Tygar, "ELK, a New Protocol for Efficient Large-Group Key Distribution," 2001 IEEE Symposium on Security and Privacy, 2001.
[4] 조태남, 이상호, "(2,4)-트리를 이용한 그룹키 관리", 정보보호학회 논문지, 제 11권, 제 4호, 2001.

[5] L. R. Dondeti, S. Mukherjee, A. Samal, "Scalable Secure one-to-many Group Communication using Dual Encryption," Proc. of IEEE International Symposium on Computer Communication, 1999.
[6] A. Ballardie, "Core Based Trees(CBT) Multicast Routing Architecture," IETF RFC 2201, 1997.
[7] K. L. Calvert, E. W. Zegura, M. J. Donahoo, "Core Selection Methods for Multicast Routing," IEEE IC3N, 1995.
[8] A. Ballardie, "Scalable Multicast Key Distribution," IETF RFC 1949, 1996.
[9] 김봉한, 이재광, "그룹 통신을 위한 멀티캐스트 키 분배 프로토콜 설계 및 검증", 정보보호학회 논문지, 제 10권, 제 2호, 2000.
[10] K. Mehlorn, *Data Structures and Algorithms 1: Sorting and Searching*, Springer-Verlarg, 1984.
[11] T. Hardjono, B. Cain and I. Monga, "Intra-Domain Group Key Management Protocol," IRTF Internet Draft, 2001.
[12] S. Setia, S. Koussih, S. Jajodia and E. Harder, "Kronos: A Scalable Group Re-Keying Approach for Secure Multicast," RSP: 21th IEEE Computer Society Symposium on Research in Security and Privacy, 2000.
[13] S. Mittra, "Iolus: A Framework for Scalable Secure Multicast," Proceedings of ACM SIGCOMM '97, 1997.
[14] M. Waldvogel, G .Caronni, D. Sun, N. Weiler and B. Plattner, "The VersaKey Framework: Versatile Group Key Management," IEEE Journal on Selected Areas in Communications, 1999.
[15] I. Chang, R. Engel, D. Kandlur, "Key Management for Secure Internet Multicast using Boolean Function Minimization Techniques," INFOCOM, 1999.
[16] W. Fenner, Internet Group Management Protocol, Version 2, IETF RFC 2236, November 1997.

조 태 남
1986년 이화여자대학교 전자계산학과 학사. 1988년 이화여자대학교 대학원 전자계산학과 석사. 1988년 ~ 1996년 한국전자통신연구원 선임연구원. 1998년 ~ 현재 이화여자대학교 과학기술대학원 컴퓨터학과 박사과정. 관심분야는 정보보호, 알고리즘 설계, 네트워크 보안

김 상 희
2001년 이화여자대학교 컴퓨터학과 학사. 2001년 ~ 현재 이화여자대학교 과학기술대학원 컴퓨터학과 석사과정. 관심분야는 정보보호, 암호 프로토콜, 알고리즘 설계, 네트워크 보안

은 상 아
2000년 이화여자대학교 컴퓨터학과 학사. 2002년 이화여자대학교 과학기술대학원 컴퓨터학과 석사. 2002년 ~ 현재 삼성전자. 관심분야는 네트워크 보안, 무선이동통신, 네트워크 프로토콜

이 상 호
1979년 서울대학교 계산통계학과 학사. 1981년 한국과학기술원 전산학과 석사. 1987년 한국과학기술원 전산학과 박사. 1983년 ~ 현재 이화여자대학교 컴퓨터학과 교수. 관심분야는 알고리즘 설계, 정보보호, 바이오인포매틱스

채 기 준
정보과학회논문지 : 정보통신
제 29 권 제 3 호 참조

박 원 주
정보과학회논문지 : 정보통신
제 29 권 제 2 호 참조

나 재 훈
1985년 중앙대학교 컴퓨터공학과 학사. 1987년 중앙대학교 대학원 컴퓨터공학과 석사. 1987년 ~ 현재 한국전자통신연구원 책임연구원. 관심분야는 IPsec, Mobile IP, IPv6, 네트워크 보안