신뢰할 수 없는 동적 네트워크 환경을 위한 비중앙화 그룹키 관리 기법

(Decentralized Group Key Management for Untrusted Dynamic Networks)

허 준 범 † 윤 현 수 ††
(Junbeom Hur) (Hyunsoo Yoon)

요 약 비중앙화(decentralized) 그룹키 관리 알고리즘은 그룹 멤버의 변화에 대한 영향을 지역적인 곳으로 한정시킴으로서 안전한 멀티캐스트 네트워크 구조의 확장성(scalability)과 안정성 (reliability)을 향상시킬 수 있다. 그러나 기존의 비중앙화 그룹키 관리 기법들에서는 멀티캐스트 과정에서 중간의 중계 노드들이 멀티캐스트 데이터의 평문 정보를 복호화함으로써 그룹 통신의 비밀성이 보장되지 않거나, 하위 그룹간의 안전한 통신을 위해서 외부의 중앙화된 그룹키 분배 센터가 필요했다. 본 연구에서 제안하는 그룹키 관리 방식에서는 서비스 제공자가 대리 암호(proxy encryption)를 이용하여 그룹키를 분산화(distributed) 방식으로 합법적인 그룹 멤버에게만 분배한다. 따라서 멀티캐스트 데이터를 전송하는 과정에서 데이터의 비밀성을 보장하는 안전한 통신을 가능케 함과 동시에 중앙화된 그룹키 분배 센터에 대한 필요성을 제거시킬 수 있다. 제안하는 그룹키 관리 기법은 중앙화된 네트워크 관리자가 없이 네트워크의 구조가 빈번히 변화하는 신뢰할 수 없는 동적인 네트워크 환경에서도 안전한 그룹통신을 효과적으로 보장할 수 있다.

키워드 : 비중앙화 그룹키 관리, 대리 암호, 안전한 그룹통신

Abstract Decentralized group key management mechanisms offer beneficial solutions to enhance the scalability and reliability of a secure multicast framework by confining the impact of a membership change in a local area. However, many of the previous decentralized solutions reveal the plaintext to the intermediate relaying proxies, or require the key distribution center to coordinate secure group communications between subgroups. In this study, we propose a decentralized group key management scheme that features a mechanism allowing a service provider to deliver the group key to valid members in a distributed manner using the proxy cryptography. In the proposed scheme, the key distribution center is eliminated while data confidentiality of the transmitted message is provided during the message delivery process. The proposed scheme can support a secure group communication in dynamic network environments where there is no trusted central controller for the whole network and the network topology changes frequently.

Key words : Group key management, proxy encryption, secure multicast

· This research is supported by the Ubiquitous Computing and Network (UCN) Project, Knowledge and Economy Frontier R&D Program of the Ministry of Knowledge Economy(MKE) in Korea as a result of UCN's subproject 09C1-T1-20S, and the Korea Science and Engineering Foundation(KOSEF) grant funded by the Korea government(MEST) (No. R01-2007-000-20865-0).

† 학생회원 : 한국과학기술원 전산학과
jbhur@nslab.kaist.ac.kr
†† 종신회원 : 한국과학기술원 전산학과 교수
hyoon@nslab.kaist.ac.kr

논문접수 : 2007년 12월 21일
심사완료 : 2009년 5월 13일

정보과학회논문지: 정보통신 제36권 제4호(2009.8)

1. 서 론

안전한 멀티캐스트 서비스를 위해서는 오직 권한을 위임받음으로서 그룹키를 알고 있는 합법적인 사용자만이 암호화되어 전송되는 그룹 통신을 복호화 할 수 있어야 한다. 따라서 안전한 멀티캐스트 문제는 암호화 키 생성 및 그룹키 분배 문제로 환원될 수 있다[1]. 특히, 애드혹(ad hoc) 네트워크 또는 매쉬(mesh) 네트워크와 같은 대규모의 동적 네트워크 환경에서는 전체 네트워크를 관리하는 신뢰할 수 있는 중앙화된 관리자가 존재하지 않고, 서로 다른 도메인에서 관리되는 동적인 그룹 멤버들로 인해 그룹키 분배가 어려워진다[2].

[1]의 연구에 따르면, 그룹키 관리 알고리즘은 중앙화(centralized), 비중앙화(decentralized), 그리고 분산화(distributed) 알고리즘으로 분류될 수 있다. 중앙화 그룹키 관리 알고리즘에서는 하나의 그룹 관리자가 전체 그룹 멤버를 관리하면서 키 계층 트리(key hierarchy tree)를 이용해 합법적인 멤버에게만 그룹키를 분배한다[3-6]. 그러나 이러한 중앙화 그룹키 관리 방식에서는 그룹 관리자에 대한 역할이 집중되기 때문에 관리자가 기능상의 문제가 발생할 경우 전체 그룹키를 관리하는데에 문제가 발생할 수 있다(a single point of failure problem). 또 다른 중앙화 그룹키 관리 방식의 문제로 한 명이라도 그룹 멤버에 변화가 생길 경우 안전한 그룹 통신을 위해서 전체의 그룹이 기존에 사용하던 그룹키를 변경해야 하는 "1-affects-n" 확장성 문제가 발생할 수 있다[7]. 이와는 대조적으로 비중앙화 알고리즘 방식은 하나의 그룹을 다수의 독립적인 하위 그룹으로 나누어 그룹키 갱신을 멤버의 변화가 일어난 해당 하위 그룹 안에서만 독립적으로 이루어 질 수 있도록 한다. 따라서, 비중앙화 그룹키 관리 방식에서는 중앙화 그룹키 관리 방식에서의 안정성 및 확장성 문제가 해결될 수 있다[7]. 그러나 이 방식에서는 독립적인 하위 그룹 사이에 그룹 통신을 해야 할 경우 각 하위 그룹의 관리자가 다른 하위 그룹으로 전송하는, 혹은 다른 하위 그룹으로부터 전송받는 데이터에 대해서 복호화를 한 후 또 다시 재암호화를 하기 때문에 암호화된 통신의 평문이 각 그룹 관리자에게 노출되게 된다. 이러한 문제를 '제삼자신뢰문제(trusting third party problem)'라고 정의한다[1].

애드혹 네트워크 및 매쉬 네트워크와 같은 기반 네트워크 구조가 없는(infrastructure-less) 환경에서는 그림 1과 같이 한 멀티캐스트의 서비스 영역이 서로 다른 도메인의 프록시에 의해 동적으로 중계됨으로써 보다 넓은 영역으로 확장되거나 조정될 수 있다. 다양한 도메인의 프록시(proxy)에 의해 네트워크가 구성되는 이러한 동적 네트워크 환경에서 각 도메인의 인증체계가 서로

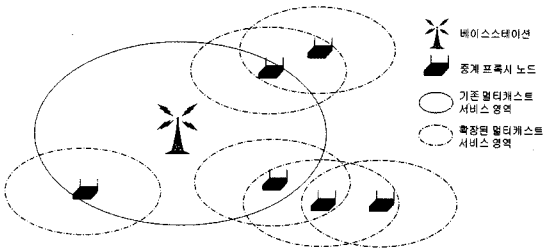

그림 1 동적인 프록시 노드의 중계에 의한 멀티캐스트 서비스 영역의 확장

독립적으로 구성되는 경우, 한 도메인에 속하는 프록시는 다른 도메인에서는 신뢰될 수 없게 된다. 게다가 신뢰할 수 있는 중앙화된 키 관리 센터(key distribution center, KDC)가 존재하지 않는 이러한 동적 네트워크 환경에서 안전한 그룹 통신을 위해서는 분산화된 방식으로 그룹키를 관리할 수 있는 효과적인 그룹키 관리 알고리즘이 필요하다. 네트워크의 위상이 빈번하게 변화하는 이러한 동적인 네트워크 환경에서는 확장성있는 그룹키 관리가 필수적이라고 할 수 있다[8]. 본 논문에서는 대리 암호(proxy cryptography)를 이용한 새로운 그룹키 관리 방법을 제안한다. 제안하는 알고리즘의 특징은 다음과 같다: (1) 중앙화된 KDC의 부재에 대비한 분산화된 그룹키 분배, (2) 비중앙화 그룹키 관리를 통한 네트워크의 확장성 향상, (3) 신뢰할 수 없는 프록시 노드에 대한 데이터 비밀성(confidentiality) 보장, (4) 동적인 멤버 및 네트워크 위상 변화에 대한 효과적인 그룹키 갱신 및 네트워크 재구성, (5) 역방향 안전성(backward secrecy) 및 순방향 안전성(forward secrecy) 보장. 역방향 안전성은 새로 그룹에 가입한 멤버가 가입하기 전의 그룹 통신을 저장하고 있더라도 과거 통신의 내용을 복호화할 수 없도록 하며, 순방향 안전성은 그룹을 탈퇴한 맴버가 탈퇴한 후의 그룹 통신에 접근할 수 없도록 함을 의미한다[1]. 본 연구의 분석 결과에 따르면 제안하는 그룹키 관리 기법은 동적인 네트워크 환경에서 확장성과 안정성 측면에서 장점을 갖는다.

본 논문은 다음과 같이 구성된다. 제 2 절에서는 안전한 그룹 통신을 위한 기존의 관련 연구들에 대해서 설명한다. 제 3 절에서는 대리 암호를 사용하는 비중앙화 그룹키 관리 알고리즘을 제안한다. 제 4 절에서는 기존 그룹키 관리 방식들과의 비교를 통해 제안하는 기법의 성능을 분석하고 이후에 안전성을 분석한다. 마지막으로 제 5 절에서 결론을 맺는다.

2. 관련 연구

중앙화 키관리 방식에서는 하나의 KDC가 전체 그룹

멤버에 대한 그룹키를 관리하고 키 상태정보를 유지한다. Logical key hierarchy(LKH)[3], one-way function tree(OFT)[4,5], ELK[6] 기법과 같은 중앙화 키관리 알고리즘은 KDC가 논리적 키계층트리를 이용해 그룹키를 관리하는데, 키계층트리의 각 말단 노드에 각 그룹 멤버를 할당하고, 트리의 중간 노드들은 키암호화키(key encryption key, KEK)를 할당한다. 그리고 루트 노드에는 현재 세션에 대한 그룹키가 할당된다. 각 그룹 멤버는 자신의 말단 노드로부터 최상위 노드에 이르는 경로키(path key)들을 저장하고 있다. 만약 한 멤버가 그룹에 가입 또는 탈퇴할 경우, KDC는 그 멤버의 해당 경로키들을 갱신해서 새로운 그룹키를 합법적인 멤버들에게 안전하게 전달함으로써 그룹 통신의 역방향 안전성 및 순방향 안전성을 보장한다. 해당 KEK를 알고 있는 멤버들은 키갱신 메시지를 복호화한 후 경로키를 갱신함으로써 새로운 그룹키를 복호화할 수 있다. 그러나 전체 네트워크가 하나의 KDC에 의해서 통제되기 때문에, 이러한 중앙화 키 관리 방식의 안전성은 KDC의 안전성에 의존하게 된다. 또한 하나의 멤버 변화에도 전체 네트워크의 그룹키를 갱신해야 하므로 '1-affects-n' 확장성 문제를 피할 수 없다[7].

반면 분산화 키관리 방식에서는 중앙화된 그룹 관리자가 존재하지 않고 각 멤버들이 독립적으로 그룹키를 생성하고 관리하게 된다. 그룹키는 각 멤버가 나누어 가지고 있는 자신의 비밀 정보를 이용해 전체 그룹키를 만들어내는 기여 방식(contributory fashion)에 의해 생성되거나, 특정한 멤버가 생성한 후 전체 멤버에게 전달하는 방식으로 생성된다. 그룹 Diffie-Hellman 키교환과 같은 대부분의 기여 방식의 키생성 알고리즘에서는 키생성에 필요한 시간(time complexity) 및 계산(computational complexity) 요구량이 그룹 멤버의 수에 비례해서 급격하게 증가하게 된다[9]. 또한, 기여 방식 프로토콜에서는 프로토콜의 견고성(robustness)을 위해서 각 그룹 멤버가 다른 그룹 멤버에 대한 리스트를 유지하고 있어야 한다. 이러한 구조적인 확장성 문제로 인해 분산화 방식의 키관리 알고리즘은 대규모의 네트워크 그룹에는 적합하지 않게 된다[1].

중앙화 키 관리 방식 또는 분산화 키 관리 방식의 확장성 및 안정성 문제를 해결하기 위해 비중앙화 알고리즘들이 제안되었다. Iolus[7]와 같은 비중앙화 그룹키 관리 알고리즘에서는 멀티캐스트 그룹의 멤버를 다수의 하위그룹으로 나누고, 각 하위그룹은 각각의 독립적인 하위그룹 관리자에 의해 통제가 된다. 각 하위그룹의 관리자는 그룹 간의 통신을 조정하고 자신의 하위그룹 멤버를 독립적으로 관리하게 된다. 따라서 멀티캐스트 그룹의 멤버가 변하는 경우, 해당 멤버의 하위그룹에서만 그룹키를 갱신하면 되기 때문에 '1-affects-n' 문제가 완화됨으로써 전체적인 네트워크의 확장성이 높아지게 된다. 그러나 그룹 SG_A에서 그룹 SG_B로 데이터를 전송하는 경우, SG_A의 그룹 관리자는 SG_A의 그룹키로 암호화된 메시지를 복호화 한 후, SG_B의 그룹키로 재암호화 한 후 SG_B로 데이터를 전송하게 된다. 따라서, 비중앙화 알고리즘에서는 그룹 통신의 평문이 메시지 전환 과정에서 그룹 데이터를 중계하는 그룹 관리자들에 노출되게 되므로 그룹 통신의 비밀성이 보장될 수 없는 문제가 발생한다. 따라서 이러한 비중앙화 그룹통신 기법에서는 안전한 그룹통신이 완전하게 각 그룹 관리자의 신뢰도에 의존하게 된다(trusting third party problem).

이러한 비중앙화 알고리즘의 문제를 해결하기 위해, Chiu 등은 [10]에서 제안된 ElGamal 대리 암호를 이용한 키분배 기법을 송신자 기반의 멀티캐스트 트리 네트워크로 확장시켰다[11]. 대리 암호의 기본 개념은 프록시로 하여금 주어진 프록시키를 이용해 ElGamal이나 아이디기반 암호화[12]와 같은 공개키 암호화 알고리즘을 이용해 한 대상의 키로 암호화된 암호문을 다른 대상의 키에 대한 암호문으로 변환할 수 있도록 하는 것이며, 그 과정에서 비밀키나 평문에 대한 어떠한 정보도 누출되지 않도록 하는 것이다. [10]에서 제안된 단방향(unidirectional) ElGamal 대리 암호방식에서는 먼저 사용자 A의 비밀키 x를 $x=x_1+x_2$를 만족하는 두 x_1과 x_2로 나눈다. 공개키는 (g,p,q,y), 비밀키는 x가 되며, 여기서 q는 소수(prime number), p는 $2q+1$의 형태를 만족하는 소수, g는 $\mathbb{Z}_p$의 생성원(generator), x는 $\mathbb{Z}_p$로부터 임의로 선택된 값, 그리고 $y=g^x \bmod p$이다. A가 B에게 중간의 프록시 P를 통해서 메시지를 전달하고자 할 때, P는 키서버로부터 x_1을, B는 x_2를 전달받는다. 그러면 P는 A에 대한 메시지를 B에 대한 메시지로 변환시킬 수 있게 된다. 단방향 대리 암호 방식의 정확성은 다음과 같이 확인할 수 있다. 프록시가 $(g^r \bmod p,\ mg^{xr} \bmod p)$를 전송받게 되면 $mg^{xr}/(g^r)^{x_1}=mg^{xg^r}$를 계산한 후 $(g^r \bmod p,\ mg^{xg^r} \bmod p)$를 B에게 전송한다. 그러면, B는 $mg^{xg^r}/(g^r)^{x_2}=m$를 통해 메시지 m을 복호화하게 된다. 이러한 대리 암호는 송신자와 수신자 사이에 다수의 신뢰할 수 없는 프록시들이 존재하는 네트워크 환경에서 안전한 멀티캐스트 구조를 형성하는데에 사용될 수 있다. Chiu의 방식에서는 KDC가 멀티캐스트 트리 위에 구성된 각 송신자와 수신자에 비밀키를 할당하고, 트리의 위상에 따라 각 프록시키를 계산해 각 프록시 노드들에 할당하게 된다. 프록시 배열이 그림 2와 같을 때, KDC는 A와 B, 그리고 데이터 경로

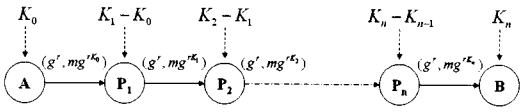

그림 2 ElGamal 대리 암호를 이용한 데이터 전송

상의 프록시들의 키를 생성한 후 각 키를 해당 대상에 전송하게 된다. 멤버가 그룹에 가입 또는 탈퇴를 할 경우, KDC는 그 멤버의 해당 프록시키를 갱신한다. 이 방식은 안전한 통신과 네트워크의 확장성을 보장할 수 있지만, 전체 멀티캐스트 트리의 위상을 이해하고 각 프록시들의 키를 관리하기 위해 여전히 중앙화된 KDC가 필요하다.

Huang 등은 프록시 암호화를 이용해서 그룹키 갱신 메시지를 각 사용자에게 분산적으로 전송하는 비중앙화 그룹키 관리 기법을 제안했다[13]. 이 프로토콜은 송신자 기반의 멀티캐스트 트리 네트워크 상에서 구성된다. 이 기법에서는 패스키 구성 알고리즘을 통해 추가적으로 송신자로부터 수신자에 이르는 패스 상의 모든 프록시 노드들에 대한 패스키를 수신자에게 전송함으로써 KDC를 제거할 수 있게 한다. 사용자가 그룹에 새롭게 가입하는 경우, 키구성 프로토콜은 그 멤버에게 새로운 패스키를 전송함으로써 사용자가 그룹키 갱신 메시지를 복호화할 수 있게 한다. 송신자 A와 수신자 B 사이의 멀티캐스트 데이터 패스가 그림 3과 같이 구성되어 있을 때, 패스키 구성 프로토콜은 $k_0+k_1+k_3+k_7+k_{11}$ 값을 그 새로운 멤버에게 안전하게 전송하게 된다. 여기서 k_0은 A의 비밀키이고 k_i는 프록시 p_i의 비밀키를 가리킨다. 암호화된 메시지 m이 A로부터 B에 이르게 되면, 암호문의 형태는 $\{c_1,c_2\}=\{g^r, mg^{r(k_0+k_1+k_3+k_7+k_{11})}\}$가 된다. 그러면 B는 자신의 패스키 $k_0+k_1+k_3+k_7+k_{11}$를 이용해서 메시지를 복호화한다. 이러한 방식은 프록시키 분배과정에서 KDC의 도움이 필요없다는 특징이 있지만 패스키 구성 과정에 추가적인 통신 비용을 요구한다는 단점이 있다. 또한 그룹 멤버의 변화에 따른 그룹키 갱신이 아닌 주기적인 그룹키갱신 알고리즘을 사용하기 때문에 그룹 통신에 대한 순방향 및 역방향 안전성을 보장할 수 없게 되며 그 안전성은 그룹키 갱신 주기에 영향을 받게 된다[14].

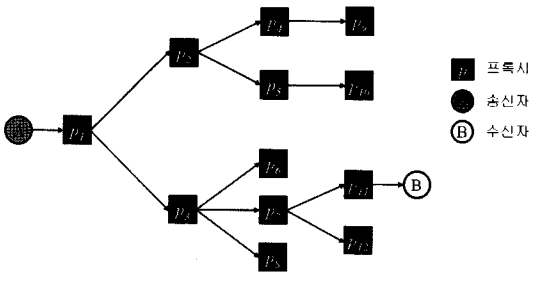

그림 3 멀티캐스트 트리

3. 제안 기법

이 절에서는 ElGamal 대리 암호를 이용한 새로운 비중앙화 그룹키 관리 기법을 제안한다. ElGamal 대리 암호는 그룹 멤버의 변화가 있을 때, 그룹키 갱신 메시지를 유효한 멤버들에게 안전하게 전달하는데에 사용된다. 제안하는 그룹키 관리 프로토콜에서는 그룹키 갱신 메시지를 암호화하고 전송하는데에 두개의 키암호화키(KEK)를 사용하게 된다. 그중 하나는 프록시와 프록시가 관리하는 하위 그룹 멤버들 사이에 공유되어 있는 프록시키이고, 다른 하나는 프록시를 제외하고 멀티캐스트 서비스의 송신자와 합법적인 그 멀티캐스트 그룹의 멤버들 사이에 공유되어 있는 이전 세션의 그룹키이다. 따라서, 두 키암호화키를 모두 아는 사용자만이 키갱신 메시지를 복호화하고 새로운 그룹키를 알아낼 수 있게 된다. 키갱신 메시지는 분산화 방식으로 멀티캐스트 경로를 따라 대리 암호를 이용해 재암호화되며 전달되기 때문에, 각 프록시 키를 관리하기 위한 중앙화된 KDC가 필요없게 된다. 또한 제안하는 그룹키 관리 기법의 구조에서는 각 프록시들이 자신의 하위 그룹에 대한 프록시키와 멤버십을 관리하기 때문에 Huang의 기법에서와 같은 추가적인 키구성 과정이 불필요하다[13]. 각 프록시들은 멀티캐스트 네트워크 환경을 동적으로 재구성할 수 있으며, 멀티캐스트 위상의 변화 및 그룹 멤버의 변화로 인한 프록시키 갱신을 해당 하위 그룹 지역으로 한정시킴으로써 네트워크의 확장성을 향상시킬 수 있다.

3.1 구조

제안하는 프로토콜은 그룹키 관리와 분배를 위해서 송신자 기반의 멀티캐스트 트리 네트워크를 가정한다. 트리의 루트 노드는 멀티캐스트의 서비스 제공자 혹은 송신자가 되고, 중간의 노드들은 프록시가 된다. 각 프록시 노드는 독립적으로 자신의 멤버십을 관리하고 지역적인 하위 그룹을 형성할 수 있다. 송신자는 멀티캐스트 트리 상의 자식 프록시들에게 ElGamal 대리 암호를 이용해 그룹키 갱신 메시지를 전송하게 된다. 중간의 프록시는 부모 프록시로부터 전달받은 암호화된 키갱신 메시지를 자신의 하위 그룹 멤버만이 복호화 할 수 있는 암호문으로 변형시킨 후, 자신의 그룹 멤버 및 자식 프록시에게 변형된 암호문을 전송한다. 키갱신 메시지는 트리의 말단 프록시에 전달될 때까지 이와같이 분산화된 방식으로 대리 암호 알고리즘으로 재암호화되며 전달되게 된다.

데이터를 중계해 주는 각 프록시들은 고정된 AP

(access point)와 같이 이동성이 없는 노드일 수도 있지만, 차량이나 보행자의 단말과 같은 이동성을 갖춘 노드가 될 수도 있다[2]. 따라서 본 논문에서는 사용자 뿐만 아니라 프록시 노드들도 네트워크에 빈번하게 가입 또는 탈퇴 할 수 있는 네트워크 상황을 고려한다.

3.1.1 안전성 요구사항

동적 네트워크 환경에서의 그룹 기반 애플리케이션에서는 사용자 뿐 아니라 프록시 또한 네트워크에 수시로 가입 또는 탈퇴할 수 있다[2]. 또한 프록시 노드는 다른 도메인에 의해 전적으로 신뢰될 수 없기 때문에 안전한 그룹통신을 위해서는 다음과 같은 안전성 요구사항이 만족되어야만 한다.

1. 데이터 비밀성(data confidentiality): 멀티캐스트 그룹에 가입되어있지 않은 프록시는 자신이 중계하는 해당 그룹의 데이터에 대한 어떠한 정보도 얻을 수 없어야 한다.
2. 그룹키 안전성(group key secrecy): 멀티캐스트 그룹에 가입되어있지 않은 외부 사용자는 해당 그룹의 그룹키에 대한 어떠한 정보도 얻을 수 없어야 한다.
3. 역방향 안전성(backward secrecy): 새로운 멤버가 i 세션에 멀티캐스트 그룹에 가입하는 경우 자신이 가입한 이전 $i-1$ 세션에 전송된 그룹데이터 또는 이전 세션의 그룹키에 대한 어떠한 정보도 얻을 수 없어야 한다.
4. 순방향 안전성(forward secrecy): 멤버가 i 세션에 멀티캐스트 그룹에서 탈퇴하는 경우 자신이 탈퇴한 이후 $i+1$ 세션의 그룹데이터 또는 그룹키에 대한 어떠한 정보도 얻을 수 없어야 한다.

3.1.2 가정 및 기호

고려하는 동적 네트워크 환경에서 멀티캐스트 트리 상의 프록시들은 기존의 기법들과 같이 부분적으로(partially) 신뢰된다고 가정한다[10,11,13]. 즉, 프록시들은 전송받은 암호문을 정해진 알고리즘에 따라 대리 암호를 이용해 올바르게 재암호화시켜 재전송한다고 가정한다.

제안하는 키 관리 기법에서 사용되는 기호는 다음과 같이 정의된다.

1. GK^i: i 세션의 그룹키. GK^i는 그룹 멤버간 안전한 그룹 통신을 위해 사용된다. GK^i는 송신자와 그룹 멤버간에 공유되는 비밀키이며, $y=g^{GK^i} \pmod p$는 공개키이다. 프록시는 그룹 멤버가 아닌 신뢰할 수 없는 상황을 가정하기 때문에 각 프록시는 GK^i를 모르게 된다.
2. PK_j: 프록시 p_j의 프록시키. 프록시키 PK_j는 p_j와 p_j에 연결된 그룹 멤버들, 그리고 멀티캐스트 트리에서 p_j의 자식 프록시들 간에 공유된 비밀키이다. PK_{init}은 송신자와 그 부모 프록시 사이에 공유된 비밀키를 나타낸다. 프록시키는 멤버가 가입하거나 탈퇴할 경우 키갱신을 해당 하위 그룹에 한정시킬 수 있도록 한다.
3. r_j: 프록시 p_j의 난수. r_{init}은 송신자의 난수를 나타낸다. r_j는 Zp로부터 임의로 선택된 ElGamal 암호화 알고리즘의 비밀 매개 변수이다.
4. PRF(M): M을 입력으로 받아 의사난수를 생성하는 단방향(one-way) 의사난수 생성 함수이다.

3.2 시스템 초기화

안전한 그룹키 분배를 위한 멀티캐스트 시스템은 다음과 같은 과정을 통해 초기화된다. 여기서는 시스템의 초기 세션을 0으로 가정한다.

1. 멀티캐스트 그룹의 송신자 s는 그룹의 모든 멤버들에게 초기 세션의 그룹키 GK^0을 안전하게 분배한다.
2. 모든 하위그룹의 각 프록시 p_j는 독립적으로 자신의 해당 하위그룹의 멤버들에게 프록시키 PK_j를 안전하게 분배한다.

초기화 이후 그룹 멤버의 가입 또는 탈퇴의 변화에 따라 세션이 증가되며 안전한 그룹 통신을 위해 그룹키는 각 세션별로 갱신되게 된다. 갱신된 그룹키는 다음 절에서 소개되는 그룹키 분배 알고리즘을 통해 합법적인 그룹의 멤버들에게 안전하게 전송되게 된다.

3.3 프록시 암호화를 이용한 그룹키 분배

제안하는 그룹키 분배 프로토콜은 크게 Encrypt, Decrypt 알고리즘으로 구성된다. 여기서는 송신자 s와 수신자 u 사이의 경로에 n 개의 프록시 $p_1, p_2, \cdots, p_n$이 존재한다고 가정한다.

1. Encrypt: 이 알고리즘은 송신자가 메시지를 암호화하거나 프록시가 전달받은 암호문을 재암호화하는데에 사용된다. 따라서 Encrypt 알고리즘은 다음과 같이 두 개의 알고리즘으로 구성된다.
   - Encrypt$_S$: 이 알고리즘은 멀티캐스트 그룹 송신자 s가 메시지 m을 암호화하는 데에 사용된다. EncryptS 알고리즘은 먼저 난수 r_{init}을 생성하고 $c_1=g^{r_{init}}$을 계산한다. 그리고 s 자신의 프록시키 PK_{init}과 현재 세션의 그룹키 GK^i를 이용해서 $c_2=mg^{(r_{init}+PK_{init})GK^i}$를 계산한다. 그 후 $\{c_1, c_2\}$를 자신의 프록시에게 전송한다.
   - Encrypt$_P$: 이 알고리즘은 프록시 노드 $p_j(1 \le j \le n)$가 수신한 암호문 $\{c_1, c_2\}$를 재암호화하는데에 사용된다. p_j는 먼저 난수 r_j를 생성하고 $c_1' = c_1 g^{r_j}$를 계산한다. 그리고 자신의 프록시키 PK_j와 이전 부모 프록시키인

PK_{j-1}를 이용해서 $c_2' = c_2 g^{(r_j - PK_{j-1} + PK_j)GK^i}$를 계산한다(여기서는 $PK_0 = PK_{init}$으로 가정한다). 그 후 $\{c_1', c_2'\}$는 수신자에게 전송될 때까지 PK_{j+1}에게 전송된다.

2. Decrypt: 이 알고리즘은 수신자 u가 전송받은 암호문 $\{c_1, c_2\}$를 복호화하는데에 사용된다. u가 전송받는 암호문은 $\{c_1, c_2\} = \{g^{r_{sk}+r_1+\cdots+r_n}, g^{(r_{sk}+r_1+\cdots+r_n+PK_n)GK^i}\}$의 형태가 되는데, u는 자신의 프록시키 PK_n과 그룹키 GK^i를 이용해서 $c_2/(c_1 g^{PK_n})GK^i = m \pmod p$ 연산을 통해 m을 복호화하게 된다.

제안하는 그룹키 분배 프로토콜은 하이브리드 암호화(hybrid encryption)[15] 방식으로 키갱신 메시지를 생성한다. 즉, 멀티캐스트 그룹의 송신자는 그룹키 전달 프로토콜을 통해 암호화된 그룹키를 자신의 그룹 멤버에게 전송한다($m = GK^{i+1}$). 그 후, 멀티캐스트 그룹 데이터를 전송할 때는 데이터를 대칭키 암호화 알고리즘을 통해서 그 그룹키로 암호화한 후 합법적인 그룹 사용자에게만 그룹 데이터를 전송하게 된다. 따라서 그룹키 갱신 메시지만이 대리 암호화 알고리즘을 이용해 분산화된 방식으로 각 멤버에 전송되게 되는데, 이러한 하이브리드 암호화는 그룹 데이터 전달에 대한 통신 및 계산비용 측면에서 네트워크의 성능을 향상시키게 된다[15].

제안한 키갱신 과정에서 새롭게 갱신된 그룹키 GK^{i+1}은 이러한 ElGamal 대리 암호화 과정을 통해 멀티캐스트 패스를 따라 재암호화 되어 각 멤버에 전송되게 된다. 따라서 프록시키와 이전 세션의 그룹키 GK^i를 모두 알고 있는 유효한 멤버들만이 키갱신 메시지를 복호화함으로써 GK^{i+1}을 얻어낼 수 있다. 그림 4는 멀티캐스트 송신자 s가 전송한 그룹키 갱신 메시지가 멀티캐스트 트리를 따라 전달되며 대리 암호를 이용해 재암호화 되는 과정을 보여준다.

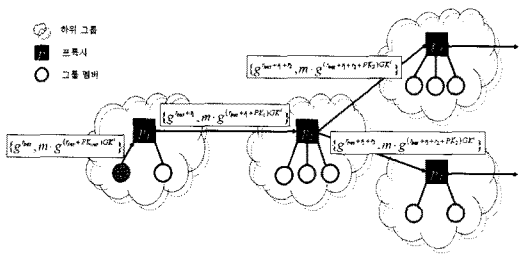

그림 4 대리 암호를 이용한 메시지 전송 과정

3.4 그룹키 갱신

연속된 멤버십 변화 간의 시간을 세션으로 정의할 때, 제안하는 그룹키 관리 기법에서는 사용자가 그룹에 가입 또는 그룹으로 부터 탈퇴할 경우 세션이 변화하며 각 세션별로 새로운 그룹키를 생성하게 된다. 이전 세션의 그룹키를 새로운 키로 갱신함으로써 새로 가입하는 멤버에 대해 역방향 안전성을 보장하고 탈퇴한 멤버에 대해 순방향 안전성을 보장하게 된다.

3.4.1 멤버 가입

한 사용자가 그룹에 멤버로 가입하기 위해서는 먼저 가장 가까운 프록시에 가입 요청 메시지를 전송하게 된다. 가입 요청 메시지를 받은 프록시는 그 멤버의 부모 프록시가 되어 요청 메시지를 멀티캐스트 트리의 루트 노드인 송신자에게 전송한다. 송신자가 그 멤버에 대한 인증을 하게 되면, 세션은 변하고 그룹키는 새로운 그룹키로 갱신된다. 멤버가 $i-1$ 세션에 프록시 p_j를 통해 그룹에 가입을 하게 되었다고 가정할 경우, 그룹키 갱신 알고리즘은 다음과 같이 진행된다.

1. 송신자는 $GK^i = PRF(GK^{i-1})$의 연산을 통해 새로운 그룹키 GK^i를 생성한다.
2. 부모 프록시 p_j는 기존의 프록시키 PK_j를 이용해 새로운 프록시키 $PK_j' = PRF(PK_j)$를 생성한다.
3. 송신자는 새로운 그룹키 GK^i를 가입 멤버에게 안전하게 단일전송(unicast)한다.
4. 부모 프록시 p_j는 자신의 프록시키 PK_j'를 가입 멤버에게 안전하게 단일전송한다.
5. PK_j와 GK^{i-1}을 알고있는 합법적인 그룹 멤버들은 의사난수 생성 함수를 이용해 PK_j'과 GK^i로 갱신한다.

그림 5의 예에서 멤버 u_1이 프록시 p_4를 통해 그룹에 가입을 할 경우, 송신자 s와 부모 프록시 p_4로부터 각각 그룹키 GK^i와 프록시키 PK_4를 전달받게 된다. 이전 세션의 프록시키나 그룹키를 알고 있던 합법적인 멤버들은 PRF 함수를 이용해서 새로운 키를 계산해낼 수 있다. 따라서 멤버 가입 시 키를 갱신하는데 필요한 통신 비용은 두 번의 단일전송이 전부이고, 그 외의 추가적인 통신 요구량은 없게 된다. 이러한 키갱신은 새로 가입한 멤버가 가입하기 이전의 그룹 통신을 복호화 할

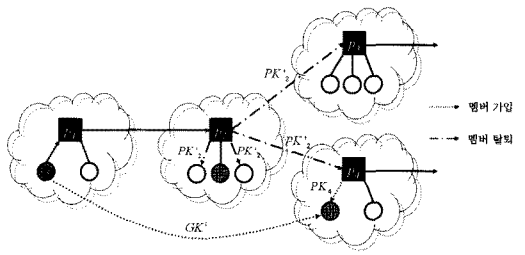

그림 5 멤버 변화에 따른 지역적 키갱신 과정

수 없게 만들기 때문에 역방향 안전성을 보장할 수 있게 한다.

3.4.2 멤버 탈퇴

멤버 탈퇴에 따른 키갱신 알고리즘은 해당 하위그룹의 프록시키 갱신과 새로운 그룹키 전달과정으로 구성된다. 멤버가 그룹을 탈퇴하는 경우, 세션이 바뀌게 되며 그 멤버의 부모 프록시 p_j는 프록시키 갱신을 시작하는데, 이러한 프록시키 갱신은 해당 멤버의 하위그룹 안에서만 한정되어 실행된다. 한 멤버가 i 세션에서 그룹을 탈퇴할 경우, 그룹키를 갱신하는 알고리즘은 다음과 같이 진행된다.

1. 프록시 p_j는 이전 프록시키 PK_j와 다른 새로운 프록시키 PK_j'을 선택한다.
2. 프록시 p_j는 PK_j'을 모든 하위 그룹 멤버와 자신에게 연결된 자식 프록시들에게 안전하게 전달한다.
3. 송신자는 이전의 세션의 그룹키 GK^i와 다른 새로운 그룹키 GK^{i+1}을 생성한 후, 멀티캐스트 트리를 따라 그룹키 전송 프로토콜을 통해 전체 그룹 멤버들에게 전송한다.

부모 프록시 p_j로부터 PK_j'을 받은 합법적인 하위 그룹 멤버들과 자식 프록시들은 그들의 부모 프록시키 PK_j를 PK_j'으로 갱신한다. 그 이후, 송신자는 그룹키 갱신 메시지 $\{c_1,c_2\}$를 멀티캐스트 경로를 따라 전송하게되고, 그 과정에서 그룹키 갱신 메시지를 받은 프록시 p_j는 PK_j'을 이용해 메시지를 대리 암호화 알고리즘으로 재암호화를 한 후 하위 멤버들과 자식 프록시들에게 전송하게 된다. 프록시 p_j의 하위 그룹 멤버들은 $\{c_1,c_2\} = \{g^{r_{sk}+r_1+\cdots+r_j}, GK^{i+1}g^{(r_{sk}+r_1+\cdots+r_j+PK_j')GK^i}\}$를 전달받게 된다. 그러면 그들은 GK^i와 PK_j'을 이용해 메시지를 복호화 할 수 있게 된다. 탈퇴한 멤버는 이전 세션의 그룹키 GK^i를 알지라도 PK_j'을 알 수 없기 때문에 키 갱신 메시지를 복호화 할 수 없게 되므로, 키갱신 과정에서 순방향 안전성이 보장된다. 그림 5는 프록시 p_2에 연결된 하위 멤버 u_2의 그룹 탈퇴로 인한 지역적인 키 갱신 과정을 보여주고 있다.

하위 그룹 멤버의 수가 n일 경우, 단일전송으로 하위 그룹 멤버들에게 프록시키를 전달하기 위해서는 $O(n)$의 통신 비용이 요구된다. 제안한 기법에서는 네트워크의 확장성을 향상시키기 위해서 지역적인 프록시키 전달 알고리즘으로 기존의 중앙화 그룹키 관리 방식인 LKH를 사용한다. LKH는 각 멤버당 $O(\log n)$의 메모리 비용으로 $O(n)$의 키갱신 통신 비용을 $O(\log n)$으로 줄여준다. [4]와 [5] 등의 기존 연구들은 $2\log n$의 키 갱신 통신 비용을 방향 해쉬 함수를 이용해서 $\log n$으로 줄일 수 있지만 제안한 기법에서는 기존 기법들과의 공정한 비교를 위해 LKH 기법을 하위 그룹 프록시키 갱신 프로토콜로 채택했다.

3.5 위상 제어

동적 네트워크 환경에서 접근 제어를 위한 그룹키 관리는 동적인 그룹 멤버 뿐 아니라 동적인 프록시에 의해서도 영향을 받는다. 동적인 네트워크 환경에서는 프록시들이 자유롭게 멀티캐스트 트리에 가입 또는 탈퇴할 수 있다. 특히 애드혹 네트워크 또는 매쉬 네트워크와 같은 환경에서는 동적인 중계 노드가 네트워크의 위상을 변화시킴으로써 서비스 범위를 확장 또는 감소시키는 경우가 빈번하다. 이처럼 프록시의 가입 또는 탈퇴로 인한 네트워크 위상의 변화는 멀티캐스트 트리의 위상 및 이웃하는 프록시들 간의 프록시키 공유 관계에도 영향을 미치게 된다. 따라서 효율적인 그룹키 관리를 위해서는 네트워크 위상의 유동성이 충분히 고려되어져야 한다.

3.5.1 프록시 가입

새로운 프록시가 네트워크에 가입하기 위해서는, 그 프록시는 먼저 자신의 부모 프록시를 선택한 후 그 프록시와 기존의 자식 프록시들 중 한 프록시 사이에 자신을 위치시킨다. 새로 가입한 프록시는 그 부모 프록시로부터 프록시키를 전달받고, 자신의 프록시 키를 자신의 새로운 자식 프록시에게 전달하게 된다. 그림 6은 프록시 p_j가 p_1과 p_2 사이의 경로를 통해 네트워크에 가입한 예를 나타낸다. p_1은 자신의 프록시키인 PK_1을 p_j에게 안전하게 전달하고, p_j는 자신의 프록시키 PK_j를 자신의 자식 프록시 p_2에게 안전하게 전달한다. p_2는 자신이 가지고 있던 이전 부모 프록시키 PK_1을 p_j로부터 받은 새로운 프록시키 PK_j로 갱신하게 된다. 만약 프록시가 멀티캐스트 트리의 말단 프록시로 가입을 하는 경우, 그 프록시는 부모 프록시로부터 프록시키를 전달받기만 한다.

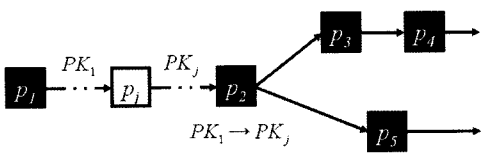

그림 6 프록시 p_j의 가입으로 인한 프록시키 갱신

새로 가입하는 프록시가 멤버를 가지고 있지 않은 경우, 또는 그룹 통신에 무관한 멤버만을 가지고 있는 경우, 그 프록시는 단지 자신의 부모 프록시가 전달하는

모든 메시지를 자식 프록시들에게 그대로 전달만 하게 된다. 따라서 이러한 프록시가 네트워크에 가입하게 될 경우에는 프록시키에 대한 업데이트 과정이 필요없다. 만약 이후에 그 프록시를 통해 그룹 통신에 참여하고자 하는 멤버가 네트워크에 가입을 하는 경우, 위에서 언급한 프록시 가입에 따른 프록시키 갱신 과정을 거치게 되고 그룹키 갱신과정에 참여하는 네트워크의 프록시로 동작하게 된다.

3.5.2 프록시 탈퇴

프록시가 네트워크를 탈퇴하는 경우, 탈퇴하는 프록시의 부모 프록시 및 자식 프록시 사이의 멀티캐스트 패스가 끊어지게 된다. 따라서 프록시의 탈퇴가 이루어지면 그 부모 프록시는 탈퇴한 프록시의 자식 프록시 중 하나의 프록시를 선택해 탈퇴한 프록시의 역할을 담당시킴으로서 멀티캐스트 트리의 위상을 재구조화시킨다. 기존의 하위그룹 멤버들은 새롭게 대체된 프록시에 속하게 되고 새로운 프록시키를 전달받는다. 그 부모 프록시는 새로이 대체된 자식 프록시에게 자신의 프록시키를 안전하게 전달하고, 그 자식 프록시는 다시 자신의 새로운 자식 프록시에게 자신의 프록시키를 안전하게 전달한다. 그림 7은 프록시 p_2가 네트워크를 탈퇴한 경우 멀티캐스트 트리의 위상 변화를 나타내고 있다. p_2가 네트워크를 탈퇴하면, 그 부모 프록시 p_1은 p_3을 선택해 기존의 p_2를 대체하고 자신의 프록시키 PK_1을 p_3에게 안전하게 전달한다. 그러면 p_3은 자신의 프록시키 PK_3을 자신의 새로운 자식 프록시 p_5에게 안전하게 전달한다. p_4는 이미 PK_3을 알고 있기 때문에 PK_3이 p_4에게 전달될 필요는 없다. 말단 프록시가 네트워크를 탈퇴하는 경우 멀티캐스트 트리 구조에서 프록시키에 대한 갱신은 필요가 없다.

그룹 통신에 참여하는 그룹 멤버가 없고 단지 그룹 데이터를 그 자식 프록시들에게 중계만을 해주는 프록시가 네트워크를 탈퇴하는 경우, 그 프록시의 부모 프록시는 자신이 직접 그 탈퇴 프록시를 대체하고 멀티캐스트 트리의 위상을 재구조화한다. 기존의 하위그룹 멤버

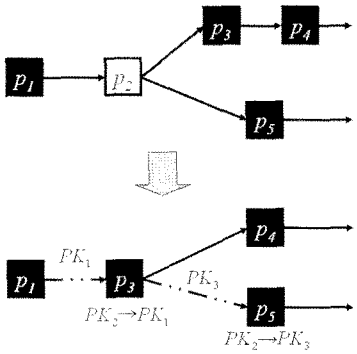

그림 7 프록시 p_2의 탈퇴로 인한 프록시키 갱신

는 그 프록시에 속하게 되고 새로운 프록시키를 전달받게 된다. 이 경우 그 부모 프록시키는 이미 새로운 자식 프록시들에게 전달이 되어져 있어서 데이터 통신을 하고 있던 상태이기 때문에 프록시키에 대한 갱신은 필요하지 않다.

4. 프로토콜 분석

이번 절에서는 제안한 기법의 성능 및 안전성을 분석한다. 4.1 절에서는 제안하는 그룹키 관리 기법의 성능을 분석하고 기존의 기법들과 비교한다. 4.2 절에서는 제안하는 기법의 안전성 분석을 한다.

4.1 성능 분석

표 1에서는 각 프로토콜의 데이터 비밀성 보장 여부 및 KDC에 대한 의존 여부를 확인 할 수 있다. 또한 표 1은 각 멤버와 프록시에게 요구되는 키 저장 비용 및 키 갱신 과정에서 필요한 통신 비용 등에 대한 분석 결과를 보여주고 있으며, 각 프로토콜이 안전한 데이터 전송을 위해 중간 프록시들에 대해 어느 정도의 신뢰도를 필요로 하는지를 보여주고 있다.

표 1에서 기호 N과 M은 각각 네트워크의 그룹 멤버 및 하위 그룹 멤버의 평균 수를 나타낸다. P는 프록시의 평균 자식 프록시의 수를 나타낸다. 멀티캐스트 그룹의 송신자와 수신자 사이의 평균 프록시 노드의 수

표 1 그룹키 관리 프로토콜 성능 비교

	LKH[3]	Iolus[7]	Chiu의 기법[11]	Huang의 기법[13]	제안 기법
데이터 비밀성	-	no	yes	yes	yes
프록시 신뢰도	-	total	partial	partial	partial
KDC	yes	no	yes	no	no
키갱신 비용 (가입,탈퇴)	$2\log N-1$, $2\log N$	$M+P$, $M+P$	$2\log M-1$, $2\log M$	$(2+L)+1$, $2\log M+1$	2, $2\log(M+P)+1$
키저장 비용 (멤버, 프록시)	$\log N$, -	1, $M+P$	$\log M$, $2M$	$\log M+1$, $P+2M+1$	$\log M+1$, $2M+1$
역방향/순방향 안전성	yes	yes	yes	no	yes

는 L로 나타낸다. 네트워크의 모든 멤버가 그룹 통신에 참여하는 그룹 서비스 가입자라고 가정할 때, 각 기법의 성능 비교 결과는 표 1과 같이 종합되어질 수 있다.

제안한 그룹키 관리 기법에서는 그룹키가 각 멤버에게 분산화된 방식으로 전달된다. 따라서 KDC가 불필요하며, 이것은 기존의 중앙화 그룹키 관리 알고리즘 방식에서 그룹 관리자에 오류가 생길 경우 전체 그룹에 영향을 미칠 수 있는 단일결함문제(a single point of failure problem)를 해결할 수 있다. 게다가 제안한 기법에서는 프록시 암호화 함수의 원자성(atomicity property)으로 인해서 멀티캐스트 경로 상의 중간 프록시들에 대한 그룹 통신의 비밀성이 보장된다[10]. 따라서 동적인 중간 프록시들에 대해 전체 신뢰가 되어야하는 제삼자 신뢰문제(trusting third party problem)를 해결할 수 있다.

제안한 기법은 각 멤버의 변화에 따른 키갱신 매커니즘(immediate rekeying)을 사용하기 때문에 그룹 통신에 대한 역방향 및 순방향 안전성을 보장할 수 있다. 반면에 Huang의 기법은 주기적인 키갱신 매커니즘(periodic rekeying)을 이용해 그룹키를 갱신하기 때문에 그룹에서 탈퇴한 이후 해당 프록시로부터 새로운 키암호화키를 전송받지 못한 멤버들도 그룹키가 갱신될 때까지 탈퇴한 이후의 그룹 통신에 대한 복호화를 할 수 있다는 문제가 있다. 게다가 새로이 그룹에 가입한 멤버들도 그룹키가 갱신되기 전까지 가입하기 이전에 전송된 그룹 통신을 복호화할 수 있다는 문제가 있다. 이러한 문제는 그룹 통신에 대한 역방향 및 순방향 안전성을 해치는 요인이 된다.

4.1.1 통신 비용

멤버가 그룹에 가입할 때, 제안한 기법에서는 새로 가입한 멤버에게 그룹키와 프록시키 전달을 위한 두 번의 단일전송(unicast)만을 필요로 한다. 따라서 Huang의 기법[13]과 같이 프록시 암호화에 대한 복호화키를 계산하고 새로 가입한 멤버에게 그 키를 전달하는데 필요한 추가적인 키구성 과정이 불필요하게 된다. 따라서 제안한 기법이 명백히 다른 기법에 비해 멤버 가입에 따른 통신 비용이 적다는 것을 알 수 있다.

멤버가 그룹을 탈퇴하는 경우, 탈퇴한 멤버의 부모 프록시키는 새 프록시키로 갱신된 후, 해당 하위 그룹의 멤버들과 자식 프록시들에게 LKH 프로토콜을 사용해서 전달되게 된다. 따라서, 멤버 탈퇴 시 키갱신 과정에서 필요한 통신 비용은 $2\log(M+P)$이 되므로 $O(\log N)$의 통신 비용을 요구하는 '1-affects-n' 확장성 문제를 완화시키게 된다. 일반적으로 $N \gg M \gg P$인 네트워크 환경의 경우, $O(\log M+P)$의 키갱신 비용은 $O(\log N)$의 비용보다 훨씬 적으며 $O(\log M)$의 비용과 거의 같다고 볼 수 있다.

4.1.2 저장 비용

저장 비용은 안전한 그룹키 전달을 위해 각 멤버가 저장해야 하는 키암호화키(KEK)의 개수로 평가한다. 제안한 그룹키 관리 기법에서 각 멤버는 하위 그룹 안에서 LKH 프로토콜을 이용한 프록시키 전달을 위해 $\log M$ 개의 KEK와, 부모 프록시로부터 전달 받는 한 개의 프록시키를 저장하게 된다. 각 프록시는 자신의 하위 그룹의 LKH 키트리를 위한 $2M-1$개의 KEK와 두 개의 프록시키(자신의 프록시키와 부모 프록시키)를 저장하게 된다. 표 1에서 분석 되었듯이, 제안한 기법에서는 Chiu의 기법[11]보다 각 멤버와 프록시가 각각 하나의 프록시키를 더 저장하도록 요구되어 지는데, 이것은 제안한 기법의 키 분배 구조가 추가적인 키구성 과정이나 KDC 없이 분산화된 방식으로 동작하도록 하는데에 사용된다.

4.2 안전성 분석

여기서는 멤버의 가입 또는 탈퇴에 따른 역방향 안전성 및 순방향 안전성을 분석한다. 또한, 제안한 ElGamal 프록시 암호 시스템의 확률적 다항시간(probabilistic polynomial-time) 공격에 대한 그룹키 안전성을 분석한다.

4.2.1 Diffie-Hellman 및 이산 로그 문제

안전성 분석에 앞서 Diffie-Hellman의 두가지 변형 문제와 이산로그 문제(discrete logarithm problem)에 대해 설명한다.

- 계산 Diffie-Hellman 문제(Computational Diffie-Hellman Problem): 곱셈연산 가능 그룹 $(G, \cdot)$, 차수 n을 갖는 원소 $\alpha \in G$, 그리고 두 원소 $\beta, \gamma \in \langle\alpha\rangle$의 값이 주어졌을 때; $\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod n$를 만족하는 $\delta \in \langle\alpha\rangle$를 찾는 문제. ($\alpha^b$와 α^c가 주어졌을 때, α^{bc}를 계산하는 문제와 동일하다.)
- 결정 Diffie-Hellman 문제(Decision Diffie-Hellman Problem): 곱셈연산 가능 그룹 $(G, \cdot)$, 차수 n을 갖는 원소 $\alpha \in G$, 그리고 세 원소 $\beta, \gamma, \delta \in \langle\alpha\rangle$의 값이 주어졌을 때; $\log_\alpha \delta \equiv \log_\alpha \beta \times \log_\alpha \gamma \pmod n$에 대한 명제의 참거짓을 결정하는 문제. ($\alpha^b$, α^c, 그리고 α^d가 주어졌을 때, $d \equiv bc \pmod n$의 참거짓을 결정하는 문제와 동일하다.)
- 이산 로그 문제(Discrete Logarithm Problem): 곱셈연산 가능 그룹 $(G, \cdot)$, 차수 n을 갖는 원소 $\alpha \in G$, 그리고 한 원소 $\beta \in \langle\alpha\rangle$의 값이 주어졌을 때; $\alpha^a = \beta$를 만족시키는 유일한 정수 $a (0 \le a \le n-1)$를 찾는 문제. 정수 a는 $\log_\alpha \beta$로 나타낼 수 있으며, 이것을 β의 이산로그라고 부른다.

4.2.2 역방향 안전성

먼저 새로이 그룹에 가입한 사용자에 대한 제안한 기법의 역방향 안전성에 대한 정리를 증명한다.

정리 1. (역방향 안전성).

i 번째 세션에 그룹에 가입한 사용자는 $i-1$ 번째 세션의 그룹키 GK^{i-1}에 대한 정보를 알 수 없다.

증명. 사용자가 $i-1$ 세션에 그룹에 가입을 하는 경우 세션은 i 세션으로 변하고, 이전의 그룹키는 $GK^i = PRF(GK^{i-1})$로 갱신된다. 새로운 멤버가 i 세션 이전의 암호화된 그룹 통신 내용을 저장하고 있고 현재의 그룹키 GK^i를 전달받았을 지라도 이전의 그룹키 없이는 가입하기 이전의 그룹 통신을 복호화할 수 없다. 현재의 정보 GK^i를 가지고 이전 세션의 그룹키를 알아내기 위해서는 공격자가 $GK^i = PRF(x)$를 만족하는 해 x를 알아낼 수 있어야 한다. 그러나 의사난수함수는 주어진 y에 대해서 $y = PRF(x)$를 만족하는 x를 찾는 것이 계산적으로 불가능한(computationally infeasible) 단방향 특성을 갖는 함수이기 때문에 공격자는 현재 세션의 그룹키 GK^i를 가지고 이전 세션의 그룹키 GK^{i-1}을 알아낼 수 없다.

공격자가 이전의 그룹키 GK^{i-1}을 알아낼 수 있는 다른 공격 방식은 그 그룹키를 포함하고 있는 이전의 키갱신 메시지 $\{c_1, c_2\} = \{g^{r_{new}+r_1+\cdots+r_n}, GK^{i-1} g^{(r_{new}+r_1+\cdots+r_n+PK_n)GK^{i-2}}\}$로 부터 GK^{i-1}을 복호화하는 것이다. 그룹키 갱신 메시지를 통해 GK^{i-1}을 알아내기 위해서는 공격자에게 PK_n이 알려져 있다고 가정할 경우, 비밀키 GK^{i-2}를 모르는 상태에서 ElGamal 암호 시스템을 깰 수 있어야 한다. ElGamal 암호 알고리즘의 의미안전성(semantic security)은 실질적으로 결정 Diffie-Hellman 문제와 동일하다고 알려져 있는데[16], 그 문제는 확률적 다항시간 공격자가 결정 Diffie-Hellman 문제를 해결하는 것이 매우 어렵다는 것을 뜻한다. [17]에서 다음의 튜링환원(Turing reduction)이 존재한다는 것이 증명되었다:

$$\{\text{결정 Diffie-Hellman 문제}\} \propto_T$$
$$\{\text{계산 Diffie-Hellman 문제}\} \propto_T \{\text{이산 로그 문제}\}.$$

따라서 결정 Diffie-Hellman 문제를 해결하는 것이 어렵다는 가정은 최소한 계산 Diffie-Hellman 문제를 풀기 어렵다는 가정 만큼 강한 의미를 가지며, 이것은 다시 최소한 이산 로그 문제를 풀기 어렵다는 가정 만큼 강한 의미를 갖는다 \cite{stinson06}. 그러므로 Zp에서 이산 로그 문제를 풀 수 없을 만큼 p가 크다면 제안한 그룹키 관리 방식은 역방향 안전성을 보장한다. ■

4.2.3 순방향 안전성

다음은 그룹에서 탈퇴한 멤버에 대한 제안한 기법의 순방향 안전성에 대한 정리를 증명한다.

정리 2. (순방향 안전성).

i 번째 세션에 그룹에 탈퇴한 사용자는 $i+1$ 번째 세션의 그룹키 GK^{i+1}에 대한 정보를 알 수 없다.

증명. 그룹 멤버가 부모 프록시 p_j로부터 그룹을 탈퇴할 경우, p_j는 새로운 프록시키 PK_j'을 생성하고 유효한 하위 그룹 멤버에게 안전하게 전달한다. 새로 갱신된 그룹키 GK^{i+1}은 프록시 암호를 이용해 암호화되어 각 멤버에게 전달되며, 이때 p_j의 그룹 멤버들에게 전달되는 키갱신 메시지는 $\{c_1, c_2\} = \{g^{r_{new}+r_1+\cdots+r_j}, GK^{i-1} g^{(r_{new}+r_1+\cdots+r_j+PK_j')GK^i}\}$가 된다. 탈퇴한 멤버가 GK^i와 이전 프록시키 PK_j를 알고 있더라도 새로운 프록시키 PK_j' 없이는 그룹키 갱신 메시지를 복호화하지 못하기 때문에 새로운 그룹키를 얻을 수 없다.

공격 모델을 간략화시키면, 공격자에게 GK^i와 $\{c_1, c_2\} = \{g^r, mg^{r'GK^i}\}$ (단, $r' = r + PK_j'$)가 주어졌을 때, 공격자가 m을 알아내기 위해서는 r' 값을 알아내야 한다. 여기서 고정된 상수 r에 대해서, r'은 비밀키 PK_j'에 의해서 결정된다. r'의 엔트로피(entropy)는 정보이론의 엔트로피 함수 H에 의해

$$H(r'|PK_j') = 0 \quad (1)$$

로 나타내어 질 수 있다[18]. 따라서 공격자에게 GK^i와 c_2가 주어졌을 때, m을 알아내기 위해 PK_j'를 유추하기 위해서는 $m = g^t \pmod p$을 모르는 조건에서

$$t + xGK^i = \log_g c_2 \pmod q$$

를 만족하는 해 x를 찾아내야 한다. 그런데 이 식은 t에 대해서 유일한 해를 갖게 된다. 다시 말하면, 프록시키 PK_j'의 모든 가능한 $x \in Zp$에 대해서 공격가 알 수 있는 정보의 양이 동일하다. 따라서 식 (1)에 의해 r'의 모든 값에 대해 공격자가 알 수 있는 정보의 양도 동일하다. 그러므로 제안한 프로토콜은 GK^i를 알고 있을지라도 PK_j'를 알 수 없는 공격자에게 무조건 안전(unconditionally secure)하다. 이것은 탈퇴한 그룹 멤버가 새로운 그룹키 갱신 메시지를 복호화 할 수 없는 것을 의미하며, 따라서 그룹 멤버가 그룹을 탈퇴한 후에는 그룹 통신에 접근할 수 없음을 의미한다. 그러므로 제안한 그룹키 관리 기법에서는 그룹 멤버의 탈퇴에 대한 순방향 안전성이 보장된다. ■

4.2.4 그룹키 안전성

다음은 제안한 프로토콜이 외부의 확률적 다항시간 공격자에 대해서 안전함을 증명한다.

정리 3. (그룹키 안전성).

그룹의 멤버가 아닌 외부의 사용자가 그룹키에 대한 정보를 알아내는 것은 계산적으로 실행불가능(computationally infeasible)하다.

증명. ElGamal 암호시스템은 송신자, 프록시, 그리고 수신자에 대해서 CPA (chosen plaintext attack) 공격에 대해 안전함이 증명되었지만, CCA(chosen ciphertext attack) 공격에 대해서는 안전하지 못하다[16]. 그러나 이것은 우리가 제안한 키 관리 구조에서는 문제가 되지 않는데, 그 이유는 프록시들이 자신이 전달받은 암호문에 대해 재암호화만을 하기 때문이다. 따라서 공격자들은 랜덤오라클(random oracle) 공격모델 하에서 프록시를 오라클로 사용함으로써 ElGamal 암호 시스템을 공격할 수 없다. 공격자가 한 프록시에 대한 물리적 공격(compromise)에 성공함으로써 그 프록시와 부모 프록시의 프록시키들을 알아낸다 할지라도, 그러한 공격이 키 관리 기법 구조의 안전성에는 영향을 미치지 않는다. ■

공격자가 프록시키를 알아내더라도 송신자가 전달하는 그룹 통신 메시지를 복호화하기 위해서는 그룹키가 필요하기 때문에 공격자는 여전히 송신자 또는 수신자 중 한 멤버에 대한 물리적인 공격을 통해 그룹키를 알아내야 한다. 만일 송신자 또는 수신자가 공격에 취약하다면 공격자는 암호 시스템을 공격할 필요 없이 공격을 통해 손상(compromise)시킨 송신자 또는 수신자를 통해 그룹키에 직접적으로 접근할 수 있다. 따라서 송신자와 수신자는 물리적인 탈취 공격에 대해서 안전하게 보호될 수 있어야 한다. 만일 GK^i를 알고 있는 멤버가 그룹을 탈퇴 한 후, 임의의 프록시 p_l과 공모를 할 수 있다면, 그 멤버는 공모하는 프록시 p_l로부터 프록시키 PK_l을 불법적으로 받을 수 있다. 그러면 그 멤버는 두 비밀키 GK^i와 PK_l을 이용하여 그룹키 갱신 메시지를 복호화함으로써 GK^{i+1}을 얻어낼 수 있다. 이러한 공모 공격(collusion attack)은 멤버 탈퇴에 따른 키갱신 프로토콜에서 갱신된 프록시키가 불법적인 사용자에게 전송되는 상황과 동일하다. 따라서, 제안한 프로토콜에서 이러한 형태의 다른 대상 간 공모 공격은 허용되어서는 안된다.

5. 결 론

본 논문에서는 새로운 비중앙화 그룹키 관리 기법을 제안하였다. 제안한 방식은 프록시 암호 알고리즘을 이용해 데이터를 전달하는 중간의 프록시 노드들에 대해서 통신의 비밀성을 보장함으로써 신뢰할 수 없는 동적 네트워크 환경에서도 안전한 그룹 통신을 할 수 있도록 한다. 그룹키는 분산화된 방식으로 각 그룹 멤버들에게 역방향 및 순방향 안전성을 보장하면서 암호화되어 전달되기 때문에 중앙화된 키 분배 센터가 불필요하다. 또한 그룹키 갱신에 필요한 프록시키 갱신이 멤버의 변화가 일어난 하위 그룹에 한정되기 때문에 확장성이 향상되었다. 제안한 방식은 동적인 그룹 멤버의 변화 뿐 아니라 동적인 네트워크 위상의 변화에도 효과적으로 대처할 수 있다. 따라서 제안한 그룹키 관리 기법은 애드혹 네트워크 및 매쉬 네트워크와 같이 중앙화된 네트워크 관리자가 없고, 네트워크 환경을 신뢰할 수 없는 동적인 네트워크 환경에서 안전하고 효율적인 그룹 통신을 지원할 수 있다.

참 고 문 헌

[1] S. Rafaeli, D. Hutchison, "A Survey of Key Management for Secure Group Communication," *ACM Computing Surveys*, vol.35, no.3, pp.309-329, 2003.

[2] I. F. Akyildiz, X. Wang, W. Wang, "Wireless Mesh Networks: A Survey, Computer Networks," vol.47, pp.445-487, 2005.

[3] C. K. Wong, M. G. Gouda, and S. S. Lam, "Secure Group Communications Using Key Graphs," *ACM SIGCOMM*, pp.68-79, 1998.

[4] D. A. McGrew and A. T. Sherman, "Key Establishment in Large Dynamic Groups Using One-way Function Trees," Tech. Rep. 0755, TIS Labs at Network Associates, Inc., Glenwood, Md.

[5] R. Canetti, J. Garay, G. Itkis, D. Miccianancio, M. Naor, and B. Pinkas, "Multicast security: A taxonomy and some efficient constructions," *Proceedings of IEEE INFOCOM 1999*, pp.708-716.

[6] A. Perrig, D. Song, and J. D. Tygar, "ELK, a New Protocol for Efficient Large-Group Key Distribution," *Proceedings of IEEE Symposium on Security and Privacy*, pp.247-262, 2001.

[7] S. Mittra, "Iolus: A Framework for Scalable Secure Multicasting," *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pp.277-288, 1997.

[8] R. Molva, A. Pannetrat, "Scalable Multicast Security in Dynamic Groups," *Proceedings of the 6th ACM Conference on Computer and Communications Security (ACM CCS)*, pp.101-112, 1999.

[9] M. Steiner, G. Tsudik, M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication," *Proceedings of the 3rd ACM Conference on Computer and Communications Security (ACM CCS)*, pp.31-37, 1996.

[10] A. Ivan, Y. Dodis, "Proxy Cryptography Revisited," *Proceedings of the Tenth Network and Distributed System Security Symposium*, 2003.

[11] Y. Chiu, C. Lei, C. Huang, "Secure Multicast Using Proxy Encryption," *Proceedings of Interna-*

tional Conference on Information and Communications Security, ICICS 2005, Lecture Notes in Computer Science 3783, pp.280-290, 2005.
[12] D. Boneh, M. Franklin, "Identity-Based Encryption from the Weil Pairing," *Proceedings of Crypto 2001*, Lecture Notes in Computer Science 2139, pp. 213-229, 2001.
[13] C. Huang, Y. Chiu, K. Chen, C. Lei, "Secure Multicast in Dynamic Environments," *Computer Networks*, vol.51, pp.2805-2817, 2007.
[14] L. Dondeti, S. Mukherjee, A. Samal, "Scalable Secure One-to-many Group Communication Using Dual Encryption," *Computer Communication*, vol. 23, pp.1681-1701, 1999.
[15] R. Cramer, V. Shoup, "Secure Hybrid Encryption from Weakened Key Encapsulation," *Proceedings of Crypto*, pp.553-571, 2007.
[16] Y. Tsiounis, M. Yung, "On the Security of El-Gamal Based Encryption," *Proceedings of the 1st International Workshop on Practice and Theory in Public Key Cryptography*, PKC'98, Lecture Notes in Computer Science 1431, pp.117-134, 1998.
[17] D. R. Stinson, Cryptography Theory and Practice (third edition), Chapman & Hall/CRC, 2006.
[18] T. M. Cover, J. A. Thomas, Elements of Information Theory (second edition), Wiley, 2006.

허 준 범

2001년 고려대학교 컴퓨터교육과 학사
2005년 한국과학기술원 전산학과 석사
2005년~현재 한국과학기술원 전산학과 박사과정. 관심분야는 네트워크 보안, 정보보안, 암호학

윤 현 수

1979년 서울대학교 전자공학과 학사. 1981년 한국과학기술원 전산학과 석사. 1988년 미국 오하이오 주립대학 전산학과 박사. 1989년~현재 한국과학기술원 교수 관심분야는 병렬 컴퓨터 구조, 무선 이동통신, 애드혹 및 센서 네트워크, 정보보안