확장성과 효율성을 갖는 프락시 암호 기반 안전한 그룹 통신

신영주° 허준범 윤현수
한국과학기술원
{yjshin°, jbhur, hyoon}@nslab.kaist.ac.kr

Scalable and efficient secure group communication using Proxy encryption

Young Joo Shin° Jun Beom Hur Hyunsoo Yoon
Dept. of EECS, Korea Advanced Institute of Science and Technology

요 약

프락시 암호 기법은 안전한 그룹 통신에서 중간 노드 신뢰 문제를 해결하기 위한 좋은 방법을 제공한다. 기존의 연구에서 프락시 암호 기법을 이용한 방법들이 제안되었으나 이들은 멀티캐스트 데이터전송 측면에서 매우 비효율적이거나 1-affect-n 문제로 인해 확장성이 저하되는 단점을 가지고 있다. 이에 따라 이 논문에서는 그룹 가입/탈퇴 분석 모델에 기반하여 동적으로 그룹을 분할/병합 함으로써 키 분배에서의 확장성과 데이터 전송에서 효율성을 같이 제공하는 새로운 프락시 암호 기반의 안전한 그룹 통신 기법을 제안한다. 제안한 방법은 인터넷이나 무선 네트워크와 같이 공개된 환경에서 가입/탈퇴가 빈번히 일어나는 대규모 가입자를 대상으로 하는 실시간 멀티미디어 방송 서비스에 적합하다

1. 서 론

최근 그룹 통신의 사용이 점차 증가함에 따라 그룹 통신의 안전성에 대한 필요성도 커지고 있다. 그룹 통신의 안정성은 인가된 멤버들만 그룹 키를 갖도록 함으로써 보장되는데 이를 위해 역방향 안정성Backward security과 순방향 안정성Forward security을 가져야 한다. 역방향 안정성은 그룹에 새로 가입한 멤버는 이전 통신내용을 알 수 없어야 하는 것을 의미하며 순방향 안정성은 그룹에서 탈퇴한 멤버가 이후의 통신 내용을 알 수 없어야 하는 것을 의미한다. 그러므로 멤버가 그룹에 가입하거나 탈퇴할 때마다 그룹 관리자Group controller는 새로운 그룹 키를 생성해서 멤버들에게 분배해야 한다.

그런데, 그룹 가입 및 탈퇴 시마다 그룹 키를 갱신하는 것은 그룹 통신의 확장성Scalability을 저하시키는 1-affect-n 문제를 유발한다 [1]. 즉, 그룹에 참여하는 멤버의 수가 증가할수록 키 분배 메시지의 크기 증가로 인한 통신 오버헤드와 높은 그룹 키 갱신 빈도에 의한 멤버의 계산 오버헤드를 야기하게 되는데, 1-affect-n 문제로 인한 확장성의 저하는 결국 그룹 통신 서비스의 질을 저하시키게 된다.

이러한 1-affect-n 문제를 해결하기 위해 분산된 그룹 키 관리Decentralized group key management 방법이 제안되었다 [1, 2]. 이는 하나의 그룹 관리자가 그룹 전체에 대해 키를 분배하는 대신 그룹을 여러 개의 서브그룹Subgroup으로 나누고 각 서브그룹별로 서브그룹 관리자를 두어 키를 분배하도록 하는 방법이다. 서브그룹 관리자는 일반적으로 멀티캐스트 트리에서 중간노드Intermediate node가 담당한다. 서브그룹은 각각 다른 그룹 키를 사용하므로 한 서브그룹 내에서 가입과 탈퇴가 발생하면 그 서브그룹의 그룹 키만 갱신하면 된다. 따라서 1-affect-n 문제를 완화하고 확장성이 저하되는 것을 막을 수 있다. 그러나, 분산된 그룹 키 관리 방법은 다시 다음과 같은 새로운 문제를 야기한다.

- **신뢰할 수 없는 중간 노드** - 중간 노드는 완전히 신뢰한다고 가정한다. 그러나 중간 노드는 인가된 그룹 사용자가 아니므로 이러한 가정은 적절하지 않으며 또한 인터넷이나 무선 네트워크와 같은 공개된 환경에서 중간 노드(라우터, AP 혹은 BS)는 쉽게 공격에 노출되므로 이들에게 그룹 키를 두는 것은 안전하지 못하다.

- **데이터 전송의 비 효율성Inefficiency** - 서브그룹별로 그룹 키가 다르므로 서브그룹 간에 데이터가 전달 되기 위해서는 중간 노드가 데이터를 자신의 그룹 키로 복호화한 후 다시 다른 서브그룹의 그룹 키로 암호화해서 전달해야 한다. 데이터 복호 / 재 암호 과정은 결국 데이터의 전송을 지연시킨다.

두 가지 문제점 가운데 중간 노드의 신뢰 문제를 해결하기 위해 프락시 암호 기법Proxy encryption을 이용한 방안이 이미 기존 연구에서 제시되었다 [3, 4]. 프락시Proxy의 역할을 하는 중간 노드는 그룹 키 없이 한 그룹 키로 암호화된 메시지를 다른

그룹 키로 암호화된 메시지로 변환하여 전달할 수 있다. 그러나, 제시된 방법들은 여전히 데이터 전송에서 비효율성과 키 분배에 따른 1-affect-n 문제를 해결하지 못하고 있다.

본 논문은 프락시 암호 기법을 이용하여 중간 노드의 신뢰 문제를 해결하면서 키 분배에서의 확장성과 데이터 전송에서 효율성을 같이 제공하는 새로운 안전한 그룹 통신 기법을 제안한다. 구체적으로, 본 논문은 멤버 가입/탈퇴 상태에 기반한 분석 모델을 만들고 이에 기반하여 동적으로 그룹을 여러 개의 서브그룹으로 분할하거나 다시 병합함으로써 효율성과 확장성을 잃지 않는 방법을 제시한다. 제시한 방법은 시뮬레이션 결과를 통해 기존의 방법들에 비해 높은 효율성과 확장성을 갖게 됨을 알 수 있다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 관련 연구를 다루고 3장은 본 연구에서 제안한 새로운 그룹 통신 기법에 대해 설명한다. 그리고 4장에서는 시뮬레이션의 결과를 통해 성능을 비교, 분석하고 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구 및 본 연구의 동기

2.1. 프락시 암호 기법을 이용한 안전한 그룹 통신

프락시 암호 기법은 [5] 에서 처음 제시되었다. 기본적인 개념은 프락시가 어떤 키로 암호화된 메시지를 키를 알지 않고서도 다른 키로 암호화된 메시지로 변환해주는 기법이다.

프락시 암호 기법은 안전한 그룹 통신 방법에 적용될 수 있다. 아래의 그림 1. (a) 는 일반적인 멀티캐스트 네트워크의 트리 구조를 보여준다. 트리의 중간 노드 P_i (i=1,2,...) 는 부모 노드로부터 수신한 멀티캐스트 데이터를 자식 및 단말 노드에게 전달한다. 이러한 중간 노드에게 프락시의 역할이 부여되는데, 그룹 멤버 M_i (i=1,2,...) 와 송신자 S 사이에는 여러 개의 중간 노드들이 위치하게 되고 이 중간 노드들은 그림 1. (b) 와 같이 하나의 프락시 열Proxy sequence 를 형성한다.

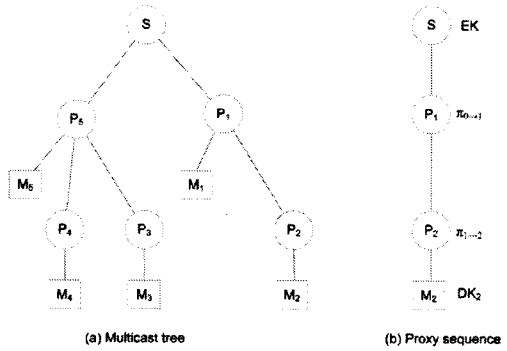

그림 1. 멀티캐스트 트리와 프락시 열

송신자 S 가 암호키 EK 로 메시지를 암호화해서 전달하면 중간의 프락시 노드들은 프락시 열을 따라 차례대로 자신의 프락시 키 π 로 메시지를 변환Transformation 하여 다음 노드에게 전달한다. 최종적으로 메시지를 수신한 멤버는 자신의 복호키 DK 로 메시지를 복호할 수 있게 된다.

프락시 암호 기법은 일반적으로 ElGamal, RSA 그리고 ID 기반 암호 알고리즘 등 여러 비대칭 키 암호 알고리즘을 이용하여 구현된다. 여기서는 ElGamal 암호 알고리즘[6] 을 이용한 구현을 기술하도록 한다. 기술하기에 앞서 두 개의 솟수Prime number q 와 p (p=2q+1), 그리고 Z_p^* 의 생성자Generator g 가 주어진 환경을 가정한다.

n 개의 프락시 노드로 이루어진 열 $P_1, P_2, ..., P_n$ 이 송신자 S와 그룹 멤버 M 사이에 있고 S 가 보낸 메시지는 프락시 노드들을 통해 M 에게 전달된다고 하자. 송신자 S 는 암호키 $EK = g^{x_0}$ $(x_0 \in Z_p)$ 을 갖고 멤버 M 은 복호키 $DK = x_n$ $(x_n \in Z_p)$ 를 갖는다. 그리고 각 프락시 노드 P_i 에는 프락시 키 $\pi_{i-1\to i} = x_{i-1} - x_i$ 가 주어진다. 송신자 S 가 메시지 m 을 암호화 한 메시지 $c_0 = (g^r, mg^{x_0 r})$ 를 전달하면 프락시 노드들은 수신한 메시지에 대해 변환을 수행한다. 프락시 노드 P_i 가 부모 노드 P_{i-1} (i=1 인 경우는 부모 노드가 송신자임) 로부터 메시지 $c_{i-1} = (g^r, mg^{x_{i-1} r})$ 를 수신하게 되면 다음과 같이 c_i 로 변환하고 자식 노드에게 전달한다.

$$c_i = (g^r, mg^{x_{i-1} r}/(g^{x_{i-1}-x_i})^r) = (g^r, mg^{x_i r})$$

최종적으로 멤버 M 은 메시지 $c_n = (g^r, mg^{x_n r})$ 을 수신하고 $m = mg^{x_n r}/g^{x_n r}$ 로 복호화하여 m 을 얻는다.

이미 기존 연구에서 프락시 암호 기법을 이용한 안전한 그룹 통신 방법들이 몇 가지 제시되었다. 이들은 중간 노드가 변환을 수행하는 메시지의 타입에 따라 데이터 변환Data transformation 방식과 키 변환Key transformation 방식으로 구분된다.

2.2. 데이터 변환 방식

Y-P.Chiu 및 그 외 연구자들[4]이 ElGamal 기반 프락시 암호 기법을 확장한 안전한 그룹 통신 기법을 제안했다. 제안한 방법은 소스 기반Source-based 멀티캐스트 트리에서 동작한다. 트리의 모든 중간 노드는 프락시의 역할을 수행하며 하향 변환Downstream conversion 키와 로컬 서브그룹 변환Local subgroup conversion 키라는 2개의 프락시 키를 갖는다. 그리고 그룹 멤버들은 연결된 프락시 노드에 따라 데이터 복호를 위한 그룹 키가 주어진다. 트리의 루트에 해당하는 송신자는 멀티캐스트 데이터를 자신의 키로 암호화해서 전달하며 데이터를 수신한 프락시 노드는 하향 변환 키와 로컬 서브그룹 변환 키로 데이터를 변환하여 각각 자식 프락시 노드와 자신의 서브그룹 멤버들에게 전달한다. 이 서브그룹 멤버들은 수신한 데이터를 복호키를 가지고 복호화할 수 있게 된다.

이 방법은 멤버들이 연결된 프락시 노드에 따라 (즉, 속한 서브그룹에 따라) 서로 다른 그룹 키를 갖게 하여 가입/탈퇴로 인한 그룹 키 갱신을 해당 서브그룹에만 한정시켰다. 따라서 1-affect-n 오버헤드가 크게 완화되어 그룹 통신에 확장성을 제공한다는 장점을 갖는다. 그러나, 이 방법은 데이터 전송에 대해서 매우 비효율적이다. 멀티캐스트 데이터는 전송되는 동안 프락시 노드마다 변환과정을 거치게 되는데, 이는 비대칭 암호 알고리즘의 높은 계산량으로 인해 프락시 노드에 극심한 계산 오버헤드를 주게 되며, 변환과정에

많은 시간이 소요되므로 결국 데이터 전송 지연을 유발하게 된다.

2.3. 키 변환 방식

프락시 암호 기법을 이용한 또 다른 방법은 C-Y.Huang 및 그 외 연구자들[3] 에 의해 제안되었다. 이들은 프락시 노드가 키 분배 센터로부터 프락시 키를 받는 대신 직접 자신의 프락시 키를 생성하도록 함으로써 키 분배 센터가 필요 없는 분산된 방법을 제안했다. 이 방법은 마찬가지로 ElGamal 기반의 프락시 암호기법을 이용하지만 기존의 방법과 다르게 데이터 전송에서의 효율성을 고려했다. 즉, 프락시 노드는 멀티캐스트 데이터를 변환하는 대신에 상대적으로 크기가 작은 그룹 키를 변환한다. 송신자가 데이터를 복호화하기 위한 TEK (Traffic Encryption Key) 를 자신의 키로 암호화하여 전송하면 프락시 노드는 이 TEK 를 변환하여 자식 프락시 노드와 로컬 멤버들에게 전달한다. TEK 가 멤버들에게 분배된 후에 송신자는 TEK 로 데이터를 암호화해서 전달하므로 프락시의 데이터 변환과정은 필요 없게 된다. 이 방식은 데이터 전송 측면에서 매우 효율적이다. 그러나, 그룹 내 모든 멤버들이 같은 TEK 를 공유하게 되므로 결국 1-affect-n 문제를 갖게 된다.

2.4. 본 연구의 동기

위에서 살펴본 바와 같이 각각의 변환 방식은 확장성과 데이터 전송 효율성에 있어서 서로 트레이드오프Trade-off의 관계에 있음을 알 수 있다. 즉, 데이터 변환 방식은 높은 확장성을 갖는 반면에 데이터 전송에서 낮은 효율성을 가지며 키 변환 방식은 반대로 높은 데이터 전송 효율성을 갖지만 키 분배에 있어서 낮은 확장성을 갖는다. 이러한 트레이드오프 관계는 멤버의 가입/탈퇴 상태에 따라 나타난다. 가입/탈퇴 빈도가 낮은 상황에서는 높은 효율성을 갖는 키 변환 방식이 유리한 반면 빈도가 높은 상황에서는 1-affect-n 오버헤드가 크기 때문에 높은 확장성을 갖는 데이터 변환 방식이 유리하다. 따라서, 멤버의 가입/탈퇴 상태에 따라 동적으로 데이터 변환 방식과 키 변환 방식을 적용한다면 확장성을 잃지 않으면서 데이터 전송의 효율성을 높이게 되므로 전체적으로 그룹 통신의 성능을 향상시킬 수 있게 된다.

3. 확장성과 효율성을 갖는 안전한 그룹 통신 기법

이 장에서는 본 논문에서 제안하는 방법인 확장성과 효율성을 제공하는 안전한 그룹 통신 기법에 대해 다룬다. 3.1. 절에서 간략한 개요를 설명하고 3.2. 절에서 가입/탈퇴 분석 모델에 기반해 동적으로 그룹을 분할, 병합하는 방법을 설명한다. 그리고 3.3. 절에서는 역방향 안정성과 순방향 안정성 보장을 위한 그룹 키 관리 방법에 대해 설명한다.

3.1. 개요

본 논문이 제안하는 방법에는 프락시 노드, 수신자 (그룹 멤버), 송신자 그리고 그룹 관리자가 참여한다. 송신자는 암호키를 가지고 메시지를 암호화하여 전송하며 수신자는 복호키를 가지고 메시지를 복호화한다. 멀티캐스트 트리의 중간 노드들은 프락시 노드의 역할을 수행한다. 그룹 관리자는 그룹 멤버에 대해 접근 제어를 담당한다.

본 논문의 그룹 통신 모델은 기존의 연구에서와 같이 소스 기반 멀티캐스트 트리 구조를 기반으로 한다. 그림 2는 멀티캐스트 트리 구조를 보여준다. 루트 노드인 S 는 송신자를 나타내며 중간 노드 P_1,...,P_6 은 프락시 노드를 나타낸다. 그리고 각 프락시 노드에 연결된 단말 노드 M_1,...,M_6 은 한 명 이상의 그룹 멤버들을 나타낸다.

프락시 노드는 키와 데이터를 변환할 수 있다. 기본적으로 모든 프락시 노드는 그룹 통신을 하는 동안 키를 변환하는 일을 수행하며 멤버의 가입/탈퇴 상태에 따라서 동적으로 데이터 변환을 수행할 수도 있다. 그림 2 의 예에서 P_2와 P_4 는 키와 데이터 변환을 수행하며 나머지 프락시 노드는 키 변환 만을 수행한다. 간결함을 위해 앞으로 키와 데이터를 변환하는 프락시 노드를 데이터 프락시Data proxy라 하고, 키만을 변환하는 프락시 노드를 키 프락시Key proxy 또는 프락시Proxy라 정의하도록 하겠다.

데이터 프락시는 멀티캐스트 트리에서 서브 트리의 루트 노드이기도 하다. 이 서브 트리에 속한 단말 노드들 즉 멤버들은 모두 같은 그룹 키를 공유한다. 이와 같이 같은 그룹 키를 공유하는 멤버들의 집합을 앞으로 서브그룹이라 정의한다. 모든 서브그룹마다 하나의 데이터 프락시 혹은 송신자가 존재한다.

앞서 언급 했듯이 데이터 변환은 멤버 가입/탈퇴 상태에 따라 동적으로 수행된다. 멤버 가입/탈퇴 빈도가 큰 경우에는 키 프락시 노드 가운데 하나가 선택되어 데이터 프락시로 전환되고 그 하위에 속한 멤버들에게 새로운 그룹 키를 분배한다. 그 결과 하나의 서브그룹이 두 개의 서브그룹으로 분할이 되어 1-affect-n 오버헤드는 줄어들게 된다. 반대로 가입/탈퇴 빈도가 작은 경우에는 데이터 프락시가 다시 키 프락시로 전환되고 하위 멤버들은 상위 서브그룹의 그룹키를 받게 된다. 그 결과로 두 개의 서브그룹이 다시 하나의 서브그룹으로 병합이 되어 데이터 변환에 따른 비용은 줄어들게 된다.

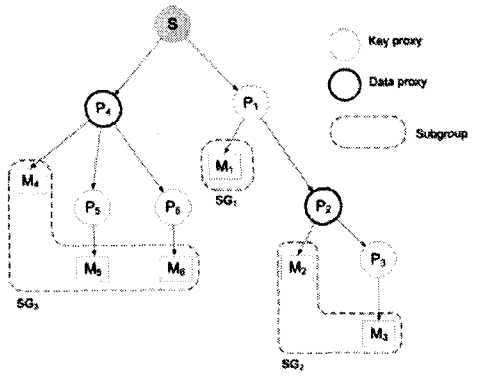

그림 2. 멀티캐스트 트리 예시

3.2. 동적인 그룹의 분할과 병합

그룹의 분할과 병합 즉, 프락시 노드의 역할 변화는 멤버의 가입/탈퇴 상태에 기반하여 동작한다. 이 절에서는 가입/탈퇴 상태를 파악하기 위한 분석 모델을 제시하고 이 모델에 기반한 그룹 분할 및 병합 동작에 대해 설명한다.

3.2.1. 멤버 가입 및 탈퇴 분석 모델

그룹 통신에서 멤버의 행동에 대해 [7]이 분석한 바에 의하면 단위 시간 동안 그룹에 가입하는 멤버의 수는 평균이 λ 인 포아송 분포를 따르고 가입 후 탈퇴까지 서비스를 이용하는 시간은 평균이 $1/\mu$ 인 지수 분포를 따른다. 각 분포는 시간에 따라 그 평균값이 변한다. 따라서, 멤버십의 변화는 마코프 프로세스 (Markov process) 모델로 설명이 가능하다. 집합 $S=\{0,1,2,...\}$ 를 현재 그룹에 가입하여 서비스를 이용중인 멤버들의 수를 나타내는 상태 집합이라 하자. $k \in S$ 일 때의 정상 상태 확률 π_k 은 다음과 같이 나타낼 수 있다 [8].

$$\pi_k = \frac{\left(\frac{\lambda}{\mu}\right)^k}{k!} e^{\left(-\frac{\lambda}{\mu}\right)}$$

상태 k 일때 멤버 가입/탈퇴에 의해 그룹 키를 갱신해야 하는 멤버의 수를 N_k 이라 하자. 즉, N_k 은 현재 서비스를 이용중인 멤버들의 수가 되므로 $E[N_k]= \lambda/\mu$ 이다. 단위 시간당 그룹 키 갱신 횟수에 대한 기대값 $E[U_{\lambda,\mu}]$ 은 다음과 같다.

$$E[U_{\lambda,\mu}] = \lambda \sum_{k=0}^{\infty} \pi_k E[N_k] = \lambda \sum_{k=0}^{\infty} \pi_k \frac{\lambda}{\mu}$$

$E[U_{\lambda,\mu}]$ 의 값을 계산하기 위해서는 π_k 을 근사화를 통해 간단히 나타낼 필요가 있다. π_k 는 δ-함수에 의해 다음과 같이 근사화된다 [7].

$$\pi_k \approx \delta\left(k - \frac{\lambda}{\mu}\right) \qquad ,\delta\left(k - \frac{\lambda}{\mu}\right) = \begin{cases} 1 & \text{if } k = \frac{\lambda}{\mu} \\ 0 & \text{otherwise} \end{cases}$$

그러므로, 다음과 같은 결과를 얻는다.

$$E[U_{\lambda,\mu}] \approx \lambda \sum_{k=0}^{\infty} \delta\left(k - \frac{\lambda}{\mu}\right)\frac{\lambda}{\mu} = \lambda\frac{\lambda}{\mu} = \frac{\lambda^2}{\mu}$$

송신자를 포함하여 모든 프락시 노드 P_i 는 주기적으로 자신을 루트 노드로 갖는 멤버들에 대해 λ 와 $1/\mu$ 값을 추정하여 멤버 가입/탈퇴 상태 $D_i=E[U_{\lambda,\mu}]$ 값을 계산한다.

3.2.2. 그룹의 분할 및 병합

앞서 언급했듯이 각 서브그룹마다 송신자 또는 데이터 프락시가 존재한다. 데이터 프락시 (혹은 송신자) 노드 P_i 는 자신의 서브그룹의 가입/탈퇴 상태를 주기적으로 관찰하여 상태에 따라 그룹을 분할 또는 병합할지를 결정하게 되는데, 이를 위해 다음과 같은 평가Evaluation 함수를 갖는다.

$$f(D_i) = \begin{cases} high & (D_i > ths) \\ low & (D_i < \frac{1}{2}D_{init}) \\ normal & (\text{otherwise}) \end{cases}$$

D_i : 멤버 가입/탈퇴 상태

ths : 그룹 분할 임계값 (임계값은 네트워크 상태나 애플리케이션의 타입에 따라 결정된다.)

D_{init} : 서브그룹이 최초 형성된 시점의 D_i 값 (송신자의 경우 $D_{init}=0$ 이다.)

송신자와 데이터 프락시 노드는 주기적으로 평가 함수 f 로부터 값을 얻고 그 결과에 따라 그룹 분할 또는 병합을 수행한다.

그룹 분할: f 의 값이 *high* 인 경우 그룹 분할을 시작한다. 즉, 데이터 프락시 (송신자) 노드 P_i 는 자신의 자손Descendant 프락시 노드들 가운데 하나를 새로운 데이터 프락시로 선택한다. 새 데이터 프락시 선택 과정은 분산된 방식으로 수행된다. 우선 노드 P_i 는 자식Child 프락시 노드들에게 D_i 값이 포함된 SPLIT 메시지를 전달한다. 메시지를 수신한 자식 노드 P_j 는 D_i 값과 자신의 D_j 값을 가지고 $\rho=D_j/D_i$ 를 계산한다.

만약 $\rho \le 1/2 + \varepsilon (\varepsilon > 0)$ 이면 노드 P_j 는 자신이 새 데이터 프락시가 되었음을 결정하고 이를 송신자에게 알린다. 송신자는 새로운 데이터 프락시 키와 그룹 키를 생성하여 각각 노드 P_j 와 노드 P_i 를 루트 노드로 하는 멤버들에게 분배한다.

ρ 가 그 외의 값을 갖는 경우에는 노드 P_j 는 SPLIT 메시지를 다시 자신의 자식 프락시 노드들에게 전달한다.

그룹 병합: f 의 값이 *low* 인 경우 그룹 병합을 시작한다. 데이터 프락시 (송신자) 노드 P_i 는 송신자에게 그룹 병합을 알리고 자신은 멀티캐스트 데이터의 변환 동작을 중지한다. (즉, 데이터 프락시에서 키 프락시로 역할이 전환된다.) 송신자는 해당 서브그룹의 멤버들에게 상위 서브그룹의 그룹키를 분배한다.

3.3. 그룹 키 관리

멤버가 그룹에 가입하거나 그룹을 탈퇴하면 역방향 안정성과 순방향 안정성을 위해 그룹 키가 갱신되어야 한다. 프락시 노드 P_i 에 연결된 그룹 멤버들의 집합을 M_i 라 하고 M_i 이 속한 서브그룹을 SG_m 이라 하자. 그룹 멤버는 2개의 키를 갖는다. 하나는 멀티캐스트 데이터를 복호화하기 위한 키인 TEK_m 이고 다른 하나는 TEK_m 을 복호화하기 위한 키인 k_i 이다. TEK_m 은 서브그룹 SG_m 의 멤버들 사이에 공유되며 k_i 는 M_i 사이에서 공유된다.

M_i 에서 멤버 가입/탈퇴가 일어났을 때 그룹키 갱신 과정은 두 단계로 진행이 된다. 우선 그룹 관리자는 k_i 를 새로운 k'_i 로 갱신하여 M_i 의 멤버들에게 분배한다. 다음에 송신자는 TEK_m 을 새로운 TEK'_m 으로 갱신하여 SG_m 의 멤버들에게

분배한다. 각 단계별로 자세한 과정을 보도록 하자. 모든 키는 항상 안전한 채널을 통해 분배된다고 가정한다.

단계 1) 그룹 관리자는 프락시 노드 P_i 의 새 프락시 키 π'_i 를 생성하여 P_i 에게 전달하고, M_i 의 새로운 키 k'_i 를 생성하여 M_i 의 멤버들에게 분배한다. 새 멤버가 가입하는 경우에는 가입하는 멤버에게도 키 k'_i 를 분배하며, 기존의 멤버가 탈퇴하는 경우에는 탈퇴하는 멤버를 제외한 나머지 M_i 의 멤버들에게만 k'_i 를 분배한다.

단계 2) M_i 의 키 k'_i 갱신이 완료되면 송신자는 서브그룹 SG_m 의 데이터 프락시 (또는 송신자) P_j 에 새 데이터 변환 프락시 키 $\Pi'_{m-1\to m}$ 와 k_j 로 암호화된 새 TEK 인 $E_{k_j}(TEK'_m)$ 를 전달한다. (여기서 TEK 를 암호화해서 전달하는 이유는 프락시 노드에게 TEK 를 노출하지 않기 위해서이다.) P_j 는 $E_{k_j}(TEK'_m)$ 를 멤버 집합 M_j 와 자식 프락시 노드에게 멀티캐스트로 전달한다. 멤버 M_j 는 k_j 로 $E_{k_j}(TEK'_m)$ 를 복호화해서 새 TEK 를 얻고, 자식 프락시 노드는 프락시 키로 변환하여 다시 멤버들과 자식 노드에게 분배한다.

위의 그림 2를 예로 들어보자. M_5 에서 멤버가 가입/탈퇴했을 경우 그룹 관리자는 P_5 에 $\pi'_{4\to5}$ 를, M_5 에 k'_5 를 각각 분배한다. 다음으로, 송신자는 $E_{k_4}(TEK'_3)$ 과 $\Pi'_{0\to3}$ 를 데이터 프락시 P_4 에게 전달한다. P_4 는 $E_{k_4}(TEK'_3)$ 를 멤버 M_4 와 자식 노드 P_5, P_6 에게 멀티캐스트 트리를 통해 분배한다. M_4 는 키 k_4 로 TEK'_3 을 얻으며 P_5 는 $\pi'_{4\to5}$ 로 $E_{k_4}(TEK'_3)$ 을 M_5 가 복호화 할 수 있는 $E_{k_5}(TEK'_3)$ 로 변환하여 M_5 에게 전달한다. P_6 와 M_6 의 경우도 마찬가지로 수행된다.

4. 시뮬레이션 결과

4.1. 시뮬레이션 모델

시뮬레이션은 멤버들이 인터넷을 통해 실시간 방송 서비스를 시청하는 상황을 설정했다. 멤버는 수시로 여러 채널(세션)에 가입/탈퇴하는데 이 논문에서는 단순함을 위해 하나의 채널만 시청한다고 가정한다. 기존 연구와의 비교를 위해 데이터 변환 방식과 키 변환 방식에 대해 같이 시뮬레이션을 수행했다. 시뮬레이션을 위한 네트워크 환경은 세 방식 모두 동일하게 프락시 노드의 개수는 총 10개, 각 프락시 노드 당 연결된 멤버의 수는 평균 20명, 즉 전체 평균 200명의 멤버가 그룹 통신에 참여하는 상황을 설정했다. 세션 시간은 총 22000 초 (약 6 시간) 이다.

멤버의 가입/탈퇴 모델은 이미 [7]에서 분석된 대로 도착 시간 간격Inter-arrival time 과 서비스 이용 시간Duration time 은 지수 분포를 따른다고 알려져 있다. 이 시뮬레이션에서는 도착 시간 간격은 평균 20초, 서비스 이용 시간은 평균 20분으로 설정했다. 멤버의 가입/탈퇴 상태는 시간에 따라 변하므로 각 확률 분포 또한 시간에 따라 변한다고 가정한다. 마지막으로, 본 논문에서 제안한 방법에서 사용하는 임계치 *ths* 는 8을 기본값으로 설정했다.

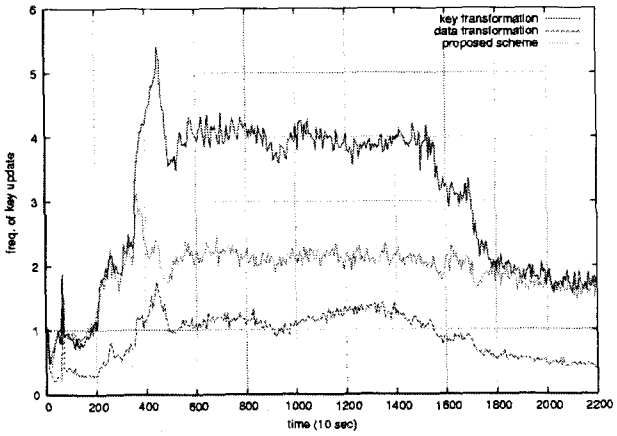

그림 3. 시간에 따른 그룹 키 갱신 빈도수 변화

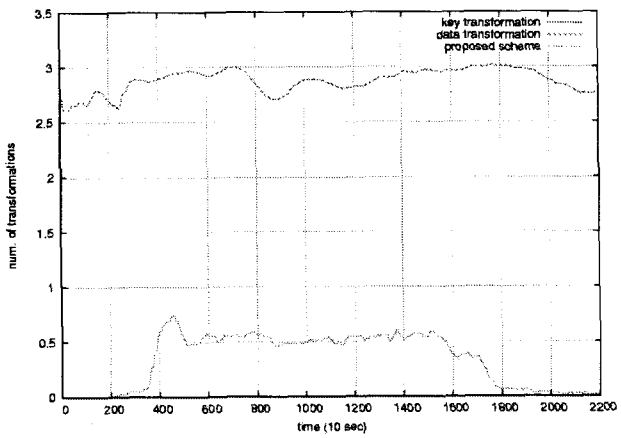

그림 4. 시간에 따른 데이터 변환 횟수 변화

4.2. 시뮬레이션 결과 및 분석

시뮬레이션을 통해 측정하고자 하는 값Metric 은 두 가지이다.

- **멤버 당 평균 키 갱신 빈도 수 (Freq. of key update):** 단위 시간 당 (1 분) 멤버의 평균 키 갱신 횟수이며 각 방식의 확장성Scalability을 나타낸다.
- **평균 데이터 변환 횟수 (Num. of data transformations):** 송신자로부터 멤버까지 데이터가 전달되는 과정에서 데이터 프락시에 의해 변환되는 평균 횟수이며 각 방식의 데이터 전송 효율성을 나타낸다.

그림 3과 4는 각각 시간에 따른 평균 키 갱신 빈도 수의 변화와 평균 데이터 변환 횟수의 변화를 보여준다. 키 변환 방식Key transformation approach 과 데이터 변환 방식Data transformation approach 은 각각 높은 데이터 전송 효율성과 확장성을 갖지만

반대로 각각 낮은 확장성과 데이터 전송 효율성을 보임을 알 수 있다. 한편, 본 논문에서 제안한 방법Proposed scheme은 멤버 상태에 따라 동적으로 키 변환 방식과 데이터 변환 방식의 장점을 취함을 볼 수 있다. 즉, 멤버 가입/탈퇴 빈도가 크지 않은 구간 (0s < t < 3000s, 18000s < t < 22000s) 에는 키 변환 방식과 거의 동일한 경향을 보이는 반면 빈도가 큰 구간 (3000s < t < 5000s)에는 데이터 변환 방식과 유사한 경향을 보인다. 그림 5는 단위 시간 당 (1 분) 멤버들의 평균 가입 횟수에 따른 평균 키 갱신 빈도 수의 변화를 보여준다. 평균 가입 횟수가 증가 할수록 멤버 가입/탈퇴 빈도수 또한 증가하게 되는데 본 논문에서 제안한 방법에서 평균 키 갱신 횟수는 데이터 변환 방식과 비슷한 양상으로 크게 증가하지 않음을 볼 수 있다. 한편, 그림 6은 제안한 방법의 임계치 *ths* 에 따른 키 갱신 빈도 수와 데이터 변환 횟수의 변화를 보여준다. 임계치가 클수록 그룹 분할이 잘 일어나지 않으므로 키 변환 방식과 비슷한 수치를 갖게 되며 임계치가 작을수록 그룹 분할이 잘 일어나므로 데이터 변환 방식과 비슷한 수치를 갖게 된다. 임계치가 약 4~8 의 값을 가질 때 확장성과 데이터 전송 효율성 양 측면에서 최적의 성능을 줄 수 있다.

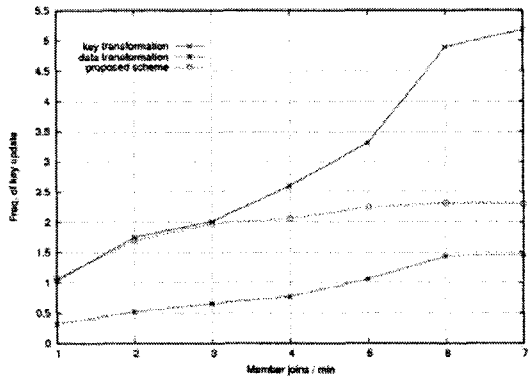

그림 5. 멤버 가입 횟수에 따른 그룹 키 갱신 빈도 변화

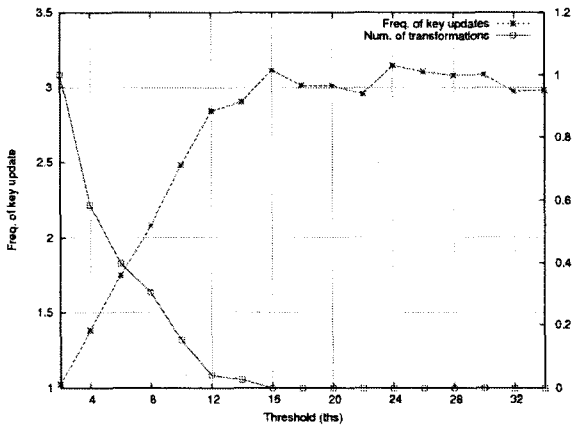

그림 6. 임계값에 따른 키 갱신 빈도 및 데이터 변환 횟수 변화

5. 결론

본 논문은 프락시 암호 기법을 이용하여 중간 노드의 신뢰 문제를 해결하면서 키 분배에서의 확장성과 데이터 전송에서 효율성을 같이 제공하는 새로운 안전한 그룹 통신 기법을 제안했다. 제안한 방법은 그룹 상태 분석 모델에 기반하여 가입/탈퇴 빈도가 큰 경우에는 그룹을 분할하여 데이터 변환 방식의 장점을 취하고, 빈도가 적은 경우에는 다시 병합하여 키 변환 방식의 장점을 취한다. 시뮬레이션 결과는 제안한 방법이 기존의 방법에 비해 확장성과 효율성을 잃지 않으면서 그룹 통신의 전체적인 성능을 향상시킨다는 사실을 뒷받침한다. 본 논문에서 제안한 방법은 인터넷이나 무선 네트워크와 같이 공개된 환경에서 가입/탈퇴가 빈번히 일어나는 대규모 가입자를 대상으로 하는 실시간 멀티미디어 방송 서비스에 적합하다

참고 문헌

[1] S. Mittra, "Iolus: a framework for scalable secure multicasting," *Proceedings of the ACM SIGCOMM'97 conference on Applications, technologies, architectures, and protocols for computer communication*, pp. 277-288, 1997.

[2] D. H. Sandro Rafaeli, "A Survey of Key Management for Secure Group Communication," *ACM Computing Surveys*, vol. Vol.35 No.3, pp. 309-329, 2003.

[3] Y.-P. C. Chun-Ying Huang, Kuan-Ta Chen and Chin-Laung Lei, "Secure multicast in dynamic environments," *Computer Networks*, vol. 51, pp. 2805-2817, 2007.

[4] Y. P. Chiu, C. L. Lei, and C. Y. Huang, "Secure Multicast Using Proxy Encryption," *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 3783, pp. 280, 2005.

[5] M. Blaze, G. Bleumer, and M. Strauss, "Divertible Protocols and Atomic Proxy Cryptography, Advances in Cryptology eurocrypt98, LNCS 1403, K. Nyberg ed," Springer-Verlag, 1998.

[6] T. Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *Information Theory, IEEE Transactions on*, vol. 31, pp. 469-472, 1985.

[7] K. C. Almeroth and M. H. Ammar, "Multicast group behavior in the Internet's multicast backbone (MBone)," *Communications Magazine, IEEE*, vol. 35, pp. 124-129, 1997.

[8] K. C. Chan and S. H. G. Chan, "Distributed servers approach for large-scale secure multicast," *Selected Areas in Communications, IEEE Journal on*, vol. 20, pp. 1500-1510, 2002.