동적인 멀티캐스트 그룹에서 효율적이고 안전한 그룹키 전송방식

장문영*, 김중태*, 조영주*, 정일용*

*조선대학교, 전자계산학과

Efficient and Secure Group Key Distribution Scheme For Dynamic Multicast Group

Mun-yeong Jang*, Jung-tae Kim*, Yeong-ju Cho*, Il-yong Chung*

*Department of Computer Science Chousun Univ.

요 약

최근에 인터넷을 이용한 화상회의, 실시간 정보서비스, 협동작업, VOD 서비스등의 요구가 증가하면서 이런 서비스들을 효율적으로 처리할 수 있는 멀티캐스트 통신이 등장하였다. 멀티캐스트 통신에서는 데이터의 보호를 위하여 그룹 키를 사용하는데 수시로 회원의 가입과 탈퇴가 이루어지는 동적인 멀티캐스트 그룹에서 안전하고 효율적인 그룹 키의 전송이 중요한 문제로 등장하였다. 본 논문에서는 지금까지의 그룹 키 전송에 대한 연구들을 살펴보고 효율적이고 안전한 그룹 키 전송 방법을 제안함으로서 통신에 필요한 키들을 보관해야 하는 저장공간의 용량을 최소화하고 그룹키 전송시 일대일 통신방식인 유니캐스트방식을 이용하여 효율적이고 안전한 전송이 될 수 있도록 하였다.

I.서론

1.연구배경및목적

인터넷이 대중화 되고 고속화되면서 다양한 서비스들에 대한 요구가 증가하였고 그중에서도 주문형 비디오 서비스, 원격 진료, 화상회의, 인터넷 방송등 그룹 통신 서비스에 대한 요구들이 갈수록 증가하고 있다. 위의 서비스들을 효율적으로 처리하기 위하여 멀티 캐스트 방식이 등장하였다.

멀티캐스트 서비스는 현재의 공개된 인터넷 환경을 이용하므로 데이터보호와 관련하여 많은 취약성에 노출되어 있으므로 멀티캐스트 데이터를 보호하기 위한 보안관련 요구도 증가하고 있다. 멀티캐스트 데이터를 보호하기 위한 일반적인 해결책으로는 합법적인 구성원만이 소유하고 있는 그룹키로 데이터를 암호화하여 전송 하는 방법이 일반적이다. 그룹내의 통신 메시지를 보호하기 위해서 멤버의 가입이나 탈퇴의 경우 항상 멤버가 소유하고 있던 그룹키를 변경해 주어야 하는데 규모가 크고 많은 수의 멤버가 수시로 가입과 탈퇴가 이루어지는 그룹의 경우 키의 갱신이나 재분배에 많은 자원을 낭비하게 되어 결과적으로 네트워크의 성능을 떨어트리는 주요한 원인이 된다. 이 문제를 해결하기 위하여 많은 방법들이 연구되어 왔다[1-14].

본 논문에서는 그룹 키 분배에 관해 현재까지 진행되어 오고 있는 여러 가지 그룹키 분배 기법들에 대하여 알아보고 기억장소의 크기를 줄이고 네트워크의 부하를 줄이는 키 분배 방법을 제안한다.

II.키 분배 관련 연구

1)요구사항

멀티캐스트 서비스는 인터넷 환경을 이용하기 때문에 보안과 관련한 여러 가지 위협요소에 노출되어 있다. 안전한 통신을 위해서는 인증성, 기밀성, 무결성, 접근 제어, 부인봉쇄등이 보장되어야 한다. 또한 구성원들이 언제라도 가입하고 탈퇴할 수 있는 동적인 멀티캐스트 그룹에서는 Group Key Secrecy, Forward Secrecy, Backward Secrecy 등의 보안 요소들이 보장되어야 [15].

2)논문의 표현

본문에서의 이해와 내용의 간결한 표현을 위해 아래와 같이 정의한다.

가. 그룹내의 i 번째 멤버 : Mi

나. 그룹내의 멤버의 수 : N

다. 그룹의 공통기로 사용하는 그룹키 : K_G

라.. 멤버 Mi 의 개인키 : Ki

마. 멤버 Mi부터 Mj 까지 사용하는 공통키 : Kij

바. 메시지 M을 암호화 키 Ki 로 암호화 : $\{M\}_{K_i}$

3)그룹키 분배 방법

가. 전통적인 방법

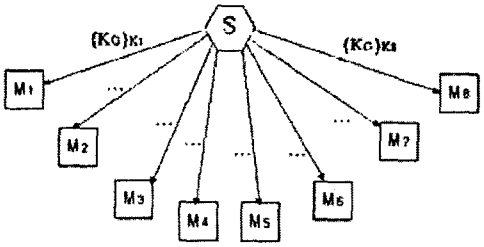

그림 1: 전통적인 그룹키 전송 방법

멀티캐스트 데이터를 보호하기 위한 전통적인 방법으로 그림 1에서 처럼 중앙 서버가 각 구성원에게 안전한 채널을 이용하여 구성원들의 개인키로 그룹키를 암호화하여 전송한 뒤 보내고자 하는 멀티캐스트 메시지를 그룹키로 암호화하여 보내고 멤버들은 보유하고 있는 그룹키로 메시지를 해독하는 방법이다. 멤버들이 가입하고 탈퇴하는 경우 멤버의 수가 N 이라면 변경된 키를 N 번 전송하여야 하기 때문에 키 변경시 전송해야 하는 메시지의 수는 O(N)이다. 이 방법은 멤버의 수가 늘어날수록 전송해야 하는 메시지의 수도 비례해서 늘어나기 때문에 멤버의 수가 커질 경우 규모확장성 문제에 봉착하게 된다.

나.Logical Key Hierarchy(LKH)

LKH 에서는 KDC(Key Distribution center)가 그룹키를 업데이트하고 배포하기 위하여 키 트리를 유지한다 [7]. 그림 2에 간단한 키 트리를 나타내었다. 트리의 각 노드는 암호화된 대칭키이다 KDC는 트리의 말단 노드에 각각의 그룹멤버들을 결합시킨다. 각 멤비는 자신으로부터 루트에 이르는 모든 노드의 키를 알고 있다. 이런 키들의 집합을 Key Path라 한다. M_4의 Key Path는 $\{K_4,K_{34},K_{14}\}$이다. 모든 멤버들은 루트 노드를 다 알고 있기 때문에 이 키는 그룹키로 사용되어진다. 그리고 이것을 K_G로 표현한다. 그림 2에서 그룹키는 K_{14}이다.

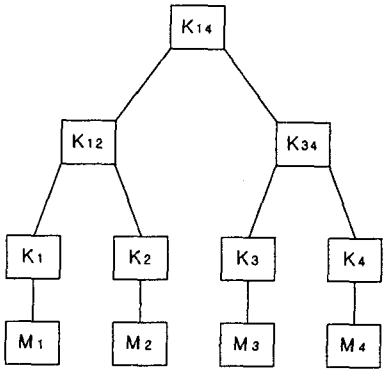

그림 2 : LKH의 구조

멤버 가입의 경우에는 새로운 멤버를 키 트리 상의 말단 노드에 적당하게 배치한 다음 가입 노드로부터 루트까지의 경로 상에 있는 모든 키들에 대해 단방향 함수 f를 적용시켜 키를 변경한다[16]. 그리고 이 변경된 키를 새로운 멤버에게 안전한 채널을 이용하여 전송한다. 이 방법을 사용하면 멀티 캐스트 전송 없이 멤버의 가입을 처리할 수 있다.

멤버의 탈퇴의 경우는 그룹에 남아있는 합법적인 가입자만이 새로 변경된 키를 받아야 하고 탈퇴한 멤버는 받아볼 수 없어야 한다. 또한 떠난 멤버들이 알고 있는 모든 키는 변경되어야 한다.

N 명의 멤버가 존재하고 키트리가 균형 잡혀 있다고 가정하면 $O(\log_2 N)$ 만큼의 키만 업데이트하면 되기 때문에 업데이트된 그룹키를 일대일 방식으로 N번만큼 전송해야 하는 전통적인 방식에 비해서 효율적이다.

다.Canetti의 방법

이 방법[8]에서도 각 노드마다 하나씩의 키가 대응된다. 한 멤버가 탈퇴하는 경우에 키의 변경은 의사 난수 발생기(pseudo-random generator) G(x)를 이용해 이루어진다. G(x)는 입력 x의 두 배의 길이를 가지는 출력을 발생시키는 함수이다. G(x)의 왼쪽 반을 L(x), 오른쪽

반을 R(x)라고 하자.

그림 3에서 멤버 M_1이 탈퇴하였다고 가정하자. 서버는 난수 r을 발생한 후에 K_{12}는 L(r), K_{14}는 L(R(r)), K_{18}은 L(R(R(r)))로 변경한다. 그리고 r값은 K_2로 암호화하여 전송한다. 이렇게 하면 각 노드는 G(x)를 이용해 자신에게 필요한 노드의 키 값만을 계산할 수 있다. Canetti의 방법에서는 그룹 멤버가 탈퇴할 때 $O(\log_2 N)$개의 키만 멀티캐스트하면 된다.

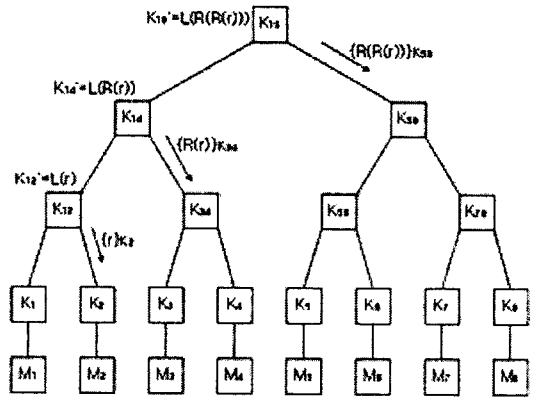

그림 3 :Canetti의 방법에서의 키 변경과정

라.ELK

이 방법[14]에서도 각 노드마다 하나씩의 키가 대응된다. 한 멤버가 탈퇴하는 경우에 키의 변경은 그림 4에서 보여 주는 것처럼 두 자식노드에게서 새로운 키를 생성하는데 필요한 C_L과 C_R를 받아서 결합한 C_{LR}를 이용하여 새로운 키를 생성한다.

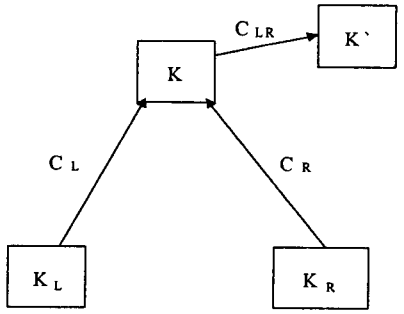

그림 4 : ELK의 키의 갱신

이 때 C_L 과 C_R을 생성할 때 pseudo-random functions(PRFs)를 이용한다. PRFs 는 n 비트의 문자열 s를 seed로 받아 n 비트의 문장 x를 출력하는 함수이다. C_L 과 C_R은 다음과 같이 구할 수 있다.

$$C_L = PRF^{\langle n \rightarrow n_1 \rangle}_{K^{\alpha}_{L}}(K) \quad (n_1 \text{ bits long})$$

$$C_R = PRF^{\langle n \rightarrow n_2 \rangle}_{K^{\alpha}_{R}}(K) \quad (n_2 \text{ bits long})$$

C_L 과 C_R 을 위와 같이 구한 다음 둘을 인접시켜 아래와 같이 C_{LR} 을 구한다.

$$C_{LR} = C_L | C_R$$

그 다음 C_{LR}을 이용하여 변경되는 키를 구한다.

$$K' = PRF^{\langle n \rightarrow n \rangle}_{C_{LR}}(K)$$

다음에 이 변경된 키를 업데이트하기 위하여 서버는 아래의 키를 브로드캐스트 하면 된다.

$$\left\{ PRF^{\langle n \rightarrow n_2 \rangle}_{K^{\alpha}_{R}}(K) \right\}_{K^{\beta}_{L}}, \quad \left\{ PRF^{\langle n \rightarrow n_1 \rangle}_{K^{\alpha}_{L}}(K) \right\}_{K^{\beta}_{R}}$$

키 업데이트 메시지의 길이는 (n_1+n_2) bits 의 길이를 가지면 K_L 을 알고 있는 멤버는 K^{β}_{L} 를 유도할 수 있고 $PRF^{\langle n \rightarrow n_2 \rangle}_{K^{\alpha}_{R}}(K)$ 를 키 업데이트 메시지로부터 해독할 수 있고 K' 을 계산할 수 있다. K_R을 알고 있는 멤버들에게도 같은 방법이 적용된다.

또한 힌트 메시지를 사용하여 키의 업데이트 중에 분실된 키를 서버에게 부담을 주지 않고 변경된 키를 복구할 수 있다.

멤버의 가입 시에는 멀티캐스트 메시지가 필요하지 않고 멤버의 탈퇴시 $(\log_2 N-1)$개의 메시지만 브로드캐스팅한다.

Ⅳ. 그룹키 분배 기법

1) 그룹키 분배 프로토콜

본 논문에서 제안하는 키 관리 기법은 기존의 키 트리에 기반한 방법이 아닌 멤버 트리에 기반한 방법이다.

가.트리의 구성

효율적인 그룹키 배포를 위해서 이전의 연구들과 마찬가지로 트리 구조를 이용한다. 본 논문에서 제안하는 방법은 이 전의 연구들과는 달리 그림5과 같이 트리의 각 노드에 키가 아닌 각 멤버들을 대응 시킨다.

서버는 그룹의 멤버 들을 차례 차례로 트리의 맨 위쪽에서부터 아래쪽으로 같은 레벨에서는 왼쪽에서부터 오른쪽으로 대응시켜 멤버트리를 구성한다.

각 멤버는 가입시 서버로부터 대응되는 노드의 번호를 부여 받고 트리에서 자신의 위치를 안다.

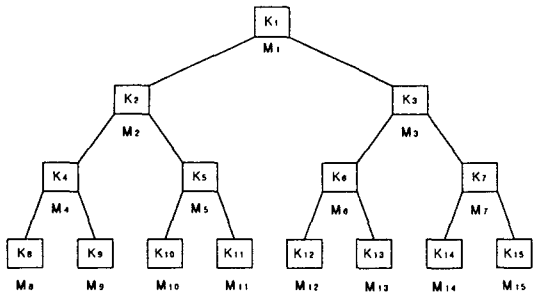

그림 5: 멤버트리의 구조

M_1을 제외한 모든 노드는 자신의 부모 노드에게 자신의 개인 키를 공개키 교환 방식을 통해 전송한다.

트리가 구성된 후 그룹의 총 멤버의 수가 N 이라고 할 때 서버는 O(N) 개의 키를 보유하게 되고 각 멤버 Mi 는 자신의 비밀 키와 그룹키 그리고 자식노드의 키인 K_{2i} 와 K_{2i+1}등 4개의 키를 소유한다.

①.그룹키의 전송

그림 7에서의 그룹키의 분배 과정은 아래와 같다.(이 때, 같은 레벨에서 자식노드에게로의 그룹키 전송은 동시에 수행된다고 가정하며 그룹키 전송시 유니캐스트 방식을 사용한다.)

a. S -> M_1 : $\{K_G\}_{K_1}$

b. M_1 -> M_2,M_3 : $\{K_G\}_{K_2, K_3}$

c. M_2 -> M_4,M_5 : $\{K_G\}_{K_4, K_5}$

M_3 -> M_6,M_7 : $\{K_G\}_{K_6, K_7}$

d. M_4 -> M_8,M_9 : $\{K_G\}_{K_8, K_9}$

M_5 -> M_{10},M_{11} : $\{K_G\}_{K_{10}, K_{11}}$

M_6 -> M_{12},M_{13}: $\{K_G\}_{K_{12}, K_{13}}$

M_7 -> M_{14},M_{15}: $\{K_G\}_{K_{14}, K_{15}}$

나. 멤버의 가입

그룹에 원하는 멤버는 서버에 자신의 공개키와 함께 가입 의사를 보내면 서버는 새로운 멤버를 그룹에 참여시키기 전에 그룹에게 새로운 멤버의 가입을 알린다 이 때 새로운 멤버가 어떤노드에 대응되는지도 같이 알린다. 새로운 멤버가 가입한다는 메시지를 받은 각 멤버들은 단방향 함수를 이용하여 그룹키와 개인키를 변경한다. 이 작업이 끝난 후 서버는 트리의 말단 적절한 노드에 새로운 멤버를 대응 시킨다. 이 때 가입된 새로운 노드는 서버에 저장되어 있는 자신의 부모노드의 공개키를 이용하여 자신의 개인키를 부모노드에게 보낸다. 가입된 노드의 개인키를 받은 부모 노드는 자식노드이 개인키를 이용하여 변경된 그룹키를 새로운 멤버에게 전송한다. 이 때에는 다른 이전의 연구들과 마찬가지로 브로드 캐스팅되는 메시지의 수는 0 이다.

다. 멤버의 탈퇴

그룹에서 탈퇴를 원하는 멤버는 서버에게 탈퇴의사를 보낸다. 탈퇴 메시지를 받은 서버는 해당 멤버의 탈퇴를 평문으로 알린다. 메시지를 수신 받은 멤버들은 가입에서처럼 단방향 함수를 이용 이번에는 자신의 개인키만 변경한다. 개인 키의 변경이 끝난후 트리에서 탈퇴 멤버를 확인하고 자기에게 해당되는 경우 노드의 상승과 상승후 자식노드의 개인키를 전달 받는다. 이 때 제거된 부모 멤버의 노드는 왼 쪽 자식노드가 상승한다. 상승으로 인한 노드의 공백은 역시 왼쪽 자식 노드가 상승하여 대응된다. 멤버 Mi 가 탈퇴했다고 가정할 경우 M_{2i},M_{4i},... 등은 계속하여 상승한다. 이 때 남아 있는 오른쪽 자식노드는 새로운 부모노드의 공개키를 이용 변경된 자신의 키를 부모 멤버에게 전달한다. 이 때 노드의 상승과 키의 전달은 동시에 이루어 진다고 가정한다. 트리의 재 구성이 끝난 후 서버는 새로운 그룹키를 생성하여 위의 그룹키를 전달하는 방법을 이용하여 그룹키를 전송한다.

라. 제안된 방법의 성능 분석

① 보유키의 수

키분배방법	서버보유키의 수	멤버보유키의 수
키 트리 기반	$\sum_{n=1}^{Log_2 N} 2^{(n-1)}$	Log_2N
멤버트리기반	O(N)	4

표 1 : Storage 비교

② 멀티캐스트 전송 수

	SKDC	LKH	ELK	제안방식
멤버의 가입	N	2	2	2
멤버의 탈퇴	O(N)	Log_2N	$2Log_2N$	$2Log_2N$

표 2 : 멀티 캐스트 전송 수의 비교

V. 결론

미래의 인터넷 환경은 정보 통신 분야의 발전에 힘입어 다양한 멀티캐스트 관련 서비스 요구가 증가할 것이다. 그러나 멀티캐스트 서비스는 기본적으로 안정성, 효율성 및 확장성 부분에서 많은 문제점을 가지고 있다.

본 논문에서는 현재까지 진행되어 오고 있는 여러 가지 그룹 키 분배 프로토콜에 대해 살펴보고, 안전한 인터넷 멀티캐스트를 위한 확장성 있는 그룹 키 분배 프로토콜을 제안하고 기존의 방법들과 비교 분석하였다. 제안한 기법을 통해 사용자는 안전하게 통신할 수 있다. 멤버의 가입과 탈퇴시 키 전송에 필요한 전송수는 이전의 연구들과 같거나 비슷하나 각 멤버들이 보유하여야 하는 키의 수를 Log2N에서 4개로 줄이고 특히 항상 멤버의 가입과 탈퇴 등 전체적인 그룹의 리스트를 관리하여야 하는 서버의 키의 보유의 수를 $\sum_{n=1}^{Log_2 N} 2^{(n-1)}$ 에서 O(N)으로 줄이고 키 전송의 부담도 각각의 노드가 담당함으로서 서버의 부하를 줄였다. 또한 각 노드에서 자식노드로의 키 전송시 안전한 채널을 통한 일대일 통신방식을 사용함으로서 이전의 연구보다 데이터의 보안면에서 더 효율적임을 알 수 있다. 현재까지는 멀티 캐스트 그룹 서비스가 실험 차원에서 대부분 무료로 화상과 음성을 제공하고 있지만 앞으로 인터넷 인구의 팽창과 함께 멀티캐스트 서비스를 원하는 사용자들이 계속 증가할 것으로 예상되어 화상 회의나 원격 교육 등 많은 분야에서 서비스 요구가 증가할 것이다. 따라서 더욱 더 효율적인 키 분배 기법이나 상용화에 걸림돌이 되고 있는 표준화 문제, 과금 문제 등에 대해 향후 많은 연구가 필요할 것이다.

참고문헌

[1] McCAnne, S. and Jacobson, V.. "Receiver-driven layered multicast." ACM SIGCOMM 96, pp. 117-130, 1996

[2] Wong, C. K., Gouda, M. and Lam, S. S.,"Secure gorup communicastion using key graphs." ACM SIGCOMM 98, pp. 68-79, 1998R

[3] Dondeit, L. R. and Mukherjec, S., "A dual encryption protocol for scalable secure multicasting." IEEE International Symposium on Computers and Communications. pp. 2-8, 1999

[4] Wallner, D., Harder, E. and Agee, R., "Key management for multicast: issues and architectures." IETF RFC 2627

[5] Mittra. S., "Iolus: a framework for scalable secure multicasting," ACM SIGCOM 97, pp. 277-288, 1997

[6] Canetti, R., Garay, J., Itkis, G., Micciancio, D., Naor, M. and Pinkas, B., "Multicast security: a taxonomy and some efficient constructions." IEEE INFOCOM 99, pp. 708-716, 1999

[7] Harney, H. and Harder., E., "Logical key hierarchy protocol," IETF Internet draft.

[8] MeGrew, D. A. and Sherman, A. T., "Key establishemnt in large dynamic groups using one-way function trees." submitted to IEEE Transactions on Software Engineering.

[9] Moyer, M. J., Rao, J. and Rohatgi, p., "A survey of security issues in multicast communications," IEEE Networks Magazine, Vol. 13, No. 6, pp. 12-23, 1999

[10] Debby M. Wallner, Eric J. Harder, and R. Agee. "Key management for multicast:Issues and architectures." Internet Draft draft-wallner-key-arch-01.txt, IETF, Network Working Group, September 1998

[11] H. Harney and c. Muckenhirm. "Group key management protocol architecture." RFC 2094, IETF, 1997

[12] Tony Ballardie. "Scalable Multicast Key Distribution." RFC 1949, May 1996

[13] Moyer, M. J., Rao, J. and Rohatgi, p., "A survey of security issues in multicast communications." IEEE Networks Magazine, Vol. 13, No. 6, pp. 12-23, 1999

[14] Adrian Pwrrig, Dawn Song and J. D. Tygar. "ELK, a New Protocol for Efficient Large-Group Key Distribution." 2001 IEEE Symposium on Research in Security and Privacy, 2001

[15] M. Steiner, G. Tsudik, and M. Waidner. Cliques. "A new approach to group key agreement." IEEE Transactions on Parallel and Distributed Systems, 2000

[16] D. Balenson, D. McGrew, and A. Sherman. "Key management for large dynamic groups: One-way function trees and amortized initialization." Internet Draft, IETF, 3. 1999.