안전한 멀티캐스트 키분배 프로토콜

조현호* · 박영호* · 이경현**

*부경대학교 전자계산학과

**부경대학교 전자컴퓨터정보통신공학부

A Secure Multicast Key Distribution Protocol

Hyun-ho Cho* · Young-ho Park* · Kyung-hyune Rhee**

*Dept. of Computer Science, Pukyung National University

**Division of Electronic, Computer & Telecommunication Engineering,

Pukyung National University

E-mail : dkfreud@mail1.pknu.ac.kr

요 약

인터넷의 활성화에 힘입어 멀티캐스트 응용을 위한 다양한 연구가 진행되어져오고 있다. 본 논문에서는 OFT(One-way Function Trees)를 사용한 안전한 멀티캐스트 키분배 프로토콜을 제안한다. 제안하는 프로토콜은 DKMP(Distributed Key Management Protocol)방식과 CKMP(Centralized Key Management Protocol)방식 각각의 특성 및 장점을 살린 혼합 방식으로, DKMP의 특성인 그룹 키 생성시 모든 멤버의 참여를 보장하는 측면과 CKMP의 특성인 키 관리 및 프로토콜 설계의 용이성을 고려하여 설계하였다. 또한 제안된 프로토콜은 그룹 비밀키 생성시 해쉬 함수와 비트단위 XOR 연산만을 이용하기 때문에 멤버들의 연산 오버헤드를 줄일 수 있어, 멤버의 가입 및 탈퇴가 빈번히 발생하는 동적인 환경에 효율적으로 활용될 수 있다.

ABSTRACT

In this paper we propose a secure multicast key distribution protocol using OFT(One-way Function Trees). The proposed protocol is a hybrid scheme of DKMP(Distributed Key Management Protocol) that guarantees all group member's participation for generating a group key, and CKMP(Centralized Key Management Protocol) that makes it easy to manage group key and design a protocol. Since the proposed protocol also computes group key using only hash function and bitwise-XOR, computational overhead can be reduced. Hence it is suitably and efficiently adaptive to dynamic multicast environment that membership change event frequently occurs.

Ⅰ. 서 론

그룹 멤버들간의 안전한 통신을 지원하기 위해 동적인 환경의 그룹에 대한 암호 키들을 효과적으로 관리하는 것은 매우 어려운 문제이다. 즉, 그룹 멤버의 가입, 탈퇴가 발생할 때마다 그룹 키는 갱신되어야 할 필요가 있으며, 그룹에 속하는 멤버들은 반드시 그 그룹 키를 효과적으로 계산할 수 있어야만 한다.

이 분야에 대한 연구는 활발히 연구되어 있으며, 그룹 키의 생성시 모든 그룹 멤버의 참여를 보장하는 DKMP(Distributed Key Management Protocol)와 그룹 키의 생성 및 관리등 모든 연산을 TTP(Trusted Third Party)가 수행하는 CKMP(Centralized Key Management Protocol)의 두 범주로 진행되고 있다. DKMP 방식에서는 주로 DH(Diffie-Hellman) 키 교환 방식을 응용하고 있다.[1,2,3,4] 특히 Kim[4]등은 그룹 키 생성시 이진트리를 사용하여 동적인 환경에서 효율적인 키 관리를 수행할 수 있지만 DH 방식을 사용하는 DKMP는 공개키 연산으로 인해 실시간적인 처리가 부족하게 되는 것이 큰 단점이다. 이에 반해

CKMP에서 TTP의 계산 속도의 향상 및 키 저장 공간의 효율성을 위해 최근 OFT(One-way Function Trees)를 응용한 방법이 제안되고 있다.[5, 6, 7]

제안하는 프로토콜은 위 두 방식의 하이브리드 형태를 취함으로써 동적 환경에서의 효율적인 그룹키를 관리할 뿐만 아니라 일방향 해쉬함수와 비트단위 XOR 연산만을 사용함으로써 속도를 향상시킨다.

2장에서 OFT에 대해 소개하고, 3장에서는 제안 프로토콜을 기술하고 분석한다. 마지막으로 4장에서 결론 및 향후 과제에 대하여 논한다.

II. One-way Function Trees

키 트리를 계산하기 위해 일방향 함수를 사용하는 방식을 OFT방식이라고 한다. 키 트리의 각 노드들의 key는 단말노드에서부터 루트노드까지 bottom-up으로 구성이 이루어지며, 키 트리를 구성하기 위해 다음의 두 함수가 사용되어진다.

g : 일방향 함수(ex. MD-5, SHA-1)
f : 두 입력 값을 단지 섞어주는 함수(예. $x \oplus y$)

위의 두 함수를 이용하여 (그림 1)과 같이 이진 키 트리를 구성할 수 있다.

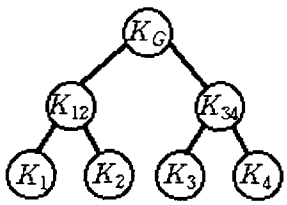

그림 1. OFT 설계

$$K_{12} = f(g(K_1), g(K_2))$$
$$K_{34} = f(g(K_3), g(K_4))$$
$$K_G = f(g(K_{12}), g(K_{34}))$$

함수 g는 중요한 특성을 가지는데, 일방향 함수로서의 특징, 즉 $g(x)$가 주어졌을 때, x를 구하는 것은 계산적으로 불가능해야만 한다.

III. 제안 프로토콜

[4]와 [5]의 프로토콜은 초기 그룹 키 설정에 관련된 기술이 미흡하다. 따라서 본 논문에서 제안하는 프로토콜은 [3]에서 제시하고 있는 IKA(Initial Key Agreeement)와 AKA(Auxiliary Key Agreement)의 두 프로토콜로 나누어져 있으며, 그룹 키 생성시 모든 멤버의 참여를 보장하고, 멤버의 가입, 탈퇴가 빈번히 발생하는 동적 환경에서도 효율적으로 동작할 수 있다.

제안 프로토콜을 기술하기 전에 먼저 본 논문에서 사용되는 표기들을 [표 1]에 정의하였다.

표 1. 용어 정의 및 표기법

blinded key	노드의 키를 함수 g에 적용하여 생성된 값
key path	멤버 자신의 단말노드로부터 루트노드까지의 경로
sibling	key path상의 노드들의 형제노드
M_i	i 번째 멤버
SK_i	대칭키 암호화에 사용되는 i 번째 멤버의 비밀 키
PK_i	공개키 암호화에 사용되는 i 번째 멤버의 공개 키
BK	blinded key
BK_i^*	i 번째 멤버의 blinded key 집합
K'	갱신된 키 K
E	암호화 알고리즘(공개키 또는 비밀키 알고리즘)

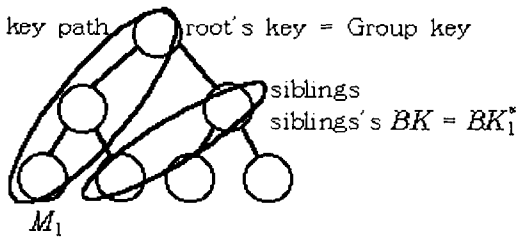

그림 2. 이진 키 트리

제안 프로토콜에서 그룹의 모든 멤버는 (그림 2)와 같은 이진 키 트리를 가지고 있으며, 그룹 키를 계산하기 위해 단지 자신의 키와 key path 상의 sibling들의 BK 만을 소유하고 있기 때문에 Group key를 계산할 수 있다. 그 이외에 각 멤버의 이진 키 트리상에서의 위치 정보를 알고 있다고 가정한다. 또한 어떤 특정 환경에서 멤버들 중 임의로 선정되는 sponsor가 존재하는 데 그 역할은 다음과 같다.

• sponsor : 그룹 멤버의 가입, 탈퇴시 새로운 그룹 키로 갱신하기 위해 현존하는 멤버들중 동적으로 유일하게 결정되어 생성된 BK 를 다른 멤버들에게 안전하게 전송할 것을 요청하는 책임을 진다. 또한 각 멤버는 어느 멤버가 sponsor인지를 독립적으로 알 수 있다.

또한 TTP로서 BCC(Broadcast Control Center)

를 두고 있는데, 그룹 키의 생성에 어떠한 참여를 하지 않는다는 점에서 기존의 TTP와 차이가 있다.

• BCC : 안전한 통신을 지원하기 위한 암호화 기능과 요청된 메시지에 대한 전달 기능을 담당하며, 그룹의 멤버들과 같이 이진 키 트리를 소유하고, 모든 멤버의 키와 BK^*를 알 수 있다.

III.1 IKA 프로토콜

IKA 프로토콜은 초기 그룹 키를 설정하기 위한 프로토콜이며, 본 프로토콜의 전체 도식도는 (그림 3)과 같이 수행되며 각 멤버의 수행동작은 (그림 4)에 주어졌다.

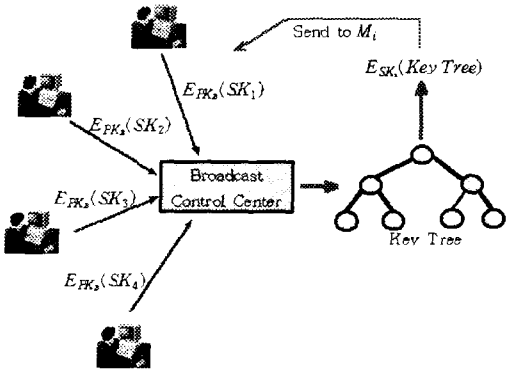

그림 3. IKA 프로토콜

먼저 각 멤버 M_i는 BCC의 공개키로 자신의 비밀키 SK_i를 암호화하여 전송하면 BCC는 OFT 방식을 사용하여 이진 키 트리 및 그룹 키를 생성할 수 있다.

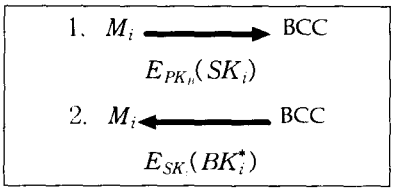

그림 4. IKA 프로토콜의 절차

그 후, BCC는 각 멤버 M_i에 대해 적절한 BK_i^*를 비밀키로 암호화하여 전송하게 되고, 따라서 각 멤버 역시 OFT방식을 사용하여 이진 키 트리를 생성하여 그룹 키를 계산할 수 있다.

III.2 AKA 프로토콜

AKA 프로토콜은 멤버의 가입 또는 탈퇴가 발생했을 경우 그룹 키를 갱신하는 프로토콜이다.

III.2.1 Join(가입) 프로토콜

먼저 한 멤버의 가입이 발생했을 경우를 고려한다.

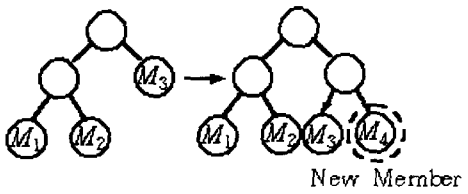

그림 5. Join 프로토콜

(그림 5)는 새로운 멤버가 가입할 경우 이진 키 트리의 변화를 보여주고 있다. Join 프로토콜의 일반적인 절차는 (그림 6)과 같다.

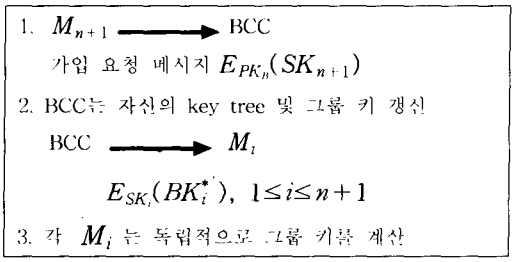

그림 6. Join 프로토콜의 절차

그룹에 가입하기를 원하는 M_{n+1}이 BCC에게 자신의 비밀키 SK_{n+1}를 BCC의 공개키로 암호화하여 전송하고, BCC는 전송된 정보로 자신의 키 트리 및 그룹 키를 갱신한다. 그 후 가입을 원하는 멤버를 포함한 모든 멤버들에 대해 적절한 BK_i^* 를 전송하여 모든 멤버들이 독립적으로 자신들의 키 트리 및 그룹 키를 갱신하게 된다. 여기서 새로 가입하는 멤버에 대한 키 트리 상의 삽입 위치는 모든 멤버가 독립적으로 유일하게 결정되며 다음의 조건에 의한다.

- 이진 키 트리의 깊이가 제일 얕은 단말 노드
- 만일 키 트리가 균형 이진 트리이면, 삽입 위치는 루트 노드

위의 조건에 의해 키 트리를 관리함으로써 그룹 키의 갱신시 연산의 최소화를 기대할 수 있다.
또한, 여러 명의 멤버가 가입할 경우에는 위 방식을 반복적, 또는 동시적으로 처리함으로서 그룹 키를 효율적으로 관리할 수 있다.

III.2.2 Leave(탈퇴) 프로토콜

한 멤버의 탈퇴시 이진 키 트리는 (그림 7)과 같이 갱신되어진다.

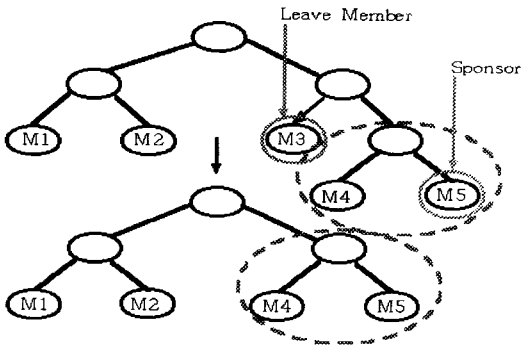

그림 7. Leave 프로토콜

Join 프로토콜과는 다르게 멤버의 탈퇴시에는 멤버중 다음의 조건으로 인해 sponsor가 결정되어지고 또한 다른 모든 멤버는 그 사실을 독립적으로 알 수 있다.

• sponsor의 결정은 탈퇴 멤버의 형제 노드를 부모노드로 하는 서브 트리에서 깊이가 제일 깊은 최 우측 멤버로 결정한다.

Leave 프로토콜의 일반적인 절차는 (그림 8)과 같이 수행된다. 먼저 탈퇴를 원하는 멤버 M_k가 BCC로 탈퇴 통보 메시지 m을 전송하면 BCC는 단지 다른 모든 멤버들에게 메시지 m을 전송하여 알리고, 메시지 m을 전송받은 M_i는 키 트리를 갱신하게 된다.

1. $M_k \longrightarrow$ BCC
   탈퇴 통보 메시지 m
   BCC $\longrightarrow M_i$: m
   M_i는 키 트리를 갱신
2. $M_{sponsor}$는 자신의 비밀키 $SK_{sponsor}$ 및 키 트리를 갱신하고 그룹 키를 계산
   $M_{sponsor} \longrightarrow$ BCC
   $E_{PK_n}(SK_{sponsor}')$
3. BCC는 키 트리 및 그룹 키를 갱신
   BCC $\longrightarrow M_i$
   $E_{SK_i}(BK_i^{*'}),\ i \neq k$
4. 각 M_i는 키 트리 및 그룹 키를 갱신

그림 8. Leave 프로토콜의 절차

그 후 sponsor 멤버가 결정이 되어 자신의 비밀키를 갱신하고 자신의 키 트리 및 그룹 키를 계산한다. 동시에 자신의 비밀키를 BCC의 공개키로 암호화하여 BCC에게 전송하고, 수신받은 BCC 역시 키 트리 및 그룹키를 갱신하여, 각각의 멤버 M_i에 적절한 BK_i^*를 암호화하여 전송한다. 그러면, 각 멤버는 수신된 정보를 이용하여 키 트리 및 그룹 키를 갱신할 수 있게된다.

여러 명이 동시에 탈퇴할 경우에도 위 방식을 적용시켜 효율적으로 처리할 수 있다.

III.3 제안 프로토콜의 분석

III.3.1 Security

본 논문에서 제안하는 프로토콜은 다음과 같은 Security 측면을 제공한다.

• Forward Secrecy: 악의적인 공격자가 이전의 그룹 키들을 알고 있다하더라도 갱신이후의 그룹 키를 계산해낼 수 없다.
• Backward Secrecy: 악의적인 공격자가 그룹 키들을 알고 있다하더라도 그에 선행하는 그룹 키를 계산해 낼 수 없다.
• Key Independence: 악의적인 공격자가 그룹 키들의 부분집합을 알고 있다하더라도 또 다른 그룹 키를 계산해낼 수 없다.

위의 특성은 사용되어지는 일방향 함수의 Security에 의존한다.

III.3.2 비교 분석

제안된 프로토콜과 기존의 DKMP방식[4], CKMP방식[6]을 분석한 결과를 [표 2]에 나타내었다.

표 2. 프로토콜 비교 분석표

구 분	DKMP[4]	CKMP[6]	제안 프로토콜
IKA/AKA	X/O	X/O	O
TTP	X	O	O
동적환경지원	O	O	O
키 생성방식	DH	OFT	OFT
키 생성 주체	멤 버	TTP	멤 버
전송메시지크기	$O(2^{l+1}-1)$	$O(l)$	$O(l)$
전송메시지 수 Join/Leave (멤버/TTP)	(N-1 0)/ (N-1 0)	(1/N-1)/ (1/N 1)	(1/N-1)/ (1/N 1)
연산 횟수	l	l	l

먼저 [4]와 [6]에서는 초기 그룹 키 설정 프로토콜에 대한 기술부분이 다소 미흡하나, 본 프로토콜에서는 TTP의 장점을 이용하여 간단히 설계하였다. 또한 가입, 탈퇴등이 빈번히 발생하는 동

적환경에 대해서도 기존의 프로토콜등과 동등하게 대처할 수 있으며, 전송메세지의 크기 비교에서 l은 트리의 깊이를 나타내는 것이며, 키 트리가 완전 이진 트리일 경우의 최대 크기를 가정하였다. [6]과는 거의 동일한 방식으로 키를 생성하기 때문에 유사한 결과를 나타내었지만 [4]의 방식과 비교하였을 경우 훨씬 효율적임을 알 수 있다.

또한 [4]의 경우와 비교하여 깊이 만큼의 연산은 동일하지만 키 생성에 있어서의 연산 방식의 간소화로 인해 훨씬 효율적이다. CPU 330MHz, 개발언어는 JDK 1.3으로 멤버의 수가 1024 ($l=10$) 경우의 시뮬레이션 결과에서 [4]의 경우는 1032ms인 반면 제안 프로토콜은 10ms인 것으로 나타났다.

IV. 결론 및 향후 연구과제

본 논문에서는 공개키 방식(DH)을 이용하는 DKMP 프로토콜과 OFT방식을 채택한 CKMP 프로토콜의 하이브리드 방식의 프로토콜을 설계하여 양자의 이점을 최대한 살릴 수 있는 프로토콜을 설계하였다. DKMP의 특성으로 그룹 키의 생성시 모든 멤버의 참여를 보장하면서도 CKMP의 특성인 프로토콜 설계의 용이성을 살렸으며, 더 나아가 키 생성 속도 및 전송 메시지의 크기를 고려하여 설계하였다. 추후 연구에서는 구현에 초점을 맞추어 보다 세부적인 프로토콜을 설계할 예정이다.

참고문헌

[1] I. Ingemasson, D. Tang, C. Wong. "A conference key distribution system", IEEE Transactions on Information Theory, 1982. 9.

[2] D. Steer, L. Strawczynski, W. Diffie, and M. Wiener, "A secure audio teleconference systems", In Crypto'88 pp.520-528. LNCS 403, 1988

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

[4] Y. Kim, A. Perring, G Tsudik, "Simple and Fault-Tolerant Key Agreement for Dynamic Collaborative Groups", ACM CCS'2000, 2000. 11.

[5] A. Fiat, M. Naor, "Broadcast Encryption", Crypto'93.

[6] D. McGrew, A. Sherman, "Key establishment in large dynamic groups using one-way function trees", Manuscript, 1998. 5.

[7] L. Dondeti, S. Mukherjee, A. Samal, "DISEC: A Distributed Framework for Scalable Secure Many-to-many Communication", IEEE ISCC 2000. 5.