트리를 이용한 효율적인 그룹키 동의 프로토콜*

박영희, 정병천, 윤현수

한국과학기술원 전자전산학과 전산학전공

An Efficient Group Key Agreement Protocol using a Tree

Younghee Park, Byungchun Chung, Hyunsoo Yoon

Dept. of Computer Science, KAIST

요 약

그룹에 속한 멤버들만이 통신하고자 하는 그룹웨어 응용프로그램이 보안상 중요한 문제로 인식되면서, 그룹키를 안전하게 생성하여 나누어 갖는 것이 필요하게 되었다. 본 논문은 어떤 그룹에 속한 모든 멤버들이 잘 알려진 두 멤버간의 Diffie-Hellman의 키 교환 프로토콜을 이용하여, 모든 메시지들을 안전하게 전달함으로써 그룹키를 나누어 갖는 새로운 그룹키 동의 프로토콜을 제안한다. 이 프로토콜은 완전 이진 트리를 이용하여 기존의 많은 양의 모듈리 멱승 연산을 일부 곱셈 연산으로 전환함으로써, 그룹키를 만드는데 있어 모듈리 멱승 계산량을 줄이는데 효과적이다.

I. 서론

현대사회에는 화상회의(video conferencing), 공동 작업장(collaborative workspaces), 쌍방향 대화(interactive chat), 네트워크 게임(network games) 등과 같은 여러 가지 그룹 단위의 응용 프로그램이 늘어나고 있으며, 그 사용빈도는 계속해서 증가 추세에 있다. 이런 응용분야는 그룹에 속하지 않는 사람들에게 정보를 유출하지 않고, 그룹의 구성원들만이 안전하면서도 비밀스러운 서비스를 받을 수 있도록 보장하여야 한다. 이를 위해서 그룹 구성원들은 그룹키를 안전하게 생성하고 공유해야 하며, 기본적으로 어떤 특정 그룹에서 사용되었던 혹은 사용 중인 그룹키는 수동적인 적(passive adversary)에 대한 공격에 안전해야 한다. 이를 그룹키 비밀성(group key secrecy) 이라고 한다.

키를 공유하는 방법에는 키 분배 프로토콜과 기여 프로토콜이 있다. 키 분배 프로토콜(key distribution protocol)은 다시 중앙방식(centralized)[3,7]과 비중앙방식(decentralized)[4]으로 나누어진다. 이들 방식은 단일 장애점(single point of failure)문제와 키를 나누어 주기 위한 안전한 채널에 대한 많은 비용이 야기된다.

기여 프로토콜(contributory protocol)은 키 동의 프로토콜(key agreement protocol)이라고도 하는데, 모든 멤버가 그룹키를 만드는데 참여하며, 안전한 채널이 필요 없으며, 모든 멤버가 참여하기 전까지는 그룹키를 예측할 수가 없다. 그러나, 키 동의 프로토콜은 대부분이 잘 알려진 Diffie-Hellman 키 교환 프로토콜을 기반으로 하고 있는데, 이것은 모듈리 멱승 계산에 대한 상당한 오버헤드를 가지고 있다. 특히, 모듈러 멱승 계산량을 줄이는 방법에 대한 연구가 한 분야로 자리해 있을 만큼 중요한 문제이다[1,2,5,6].

본 논문은 완전 이진 트리(complete binary tree)와 잘 알려진 두 멤버간의 Diffie-Hellman 키 교환 프로토콜을 이용하여 그룹으로 확장시킨 그룹키 동의 프로토콜이다. 두 멤버간의 Diffie-Hellman 키 교환 프로토콜을 이용하여 단계별로 그룹에 속해 있는 멤버들끼리 안전하게 메시지를 전달함으로써 그룹키를 생성하고 분배한다. 그리고 이전 관련 논문과는 달리, 본 논문에서는 일부 모듈리 멱승 연산을 곱하기 연산으로 전환함으로써 계산의 부담을 줄이고, 그룹을 구성하는 각각의 멤버들은 일정한 상수만큼의 모듈러 멱승 계산과 곱셈 계산을 하게 된다.

II. 본문

1. 관련 연구

본 논문과 관련된 키 동의 프로토콜이면서 모듈리 멱승 계산을 기반으로 하는 것에는 키트

*본 연구는 첨단정보기술 연구센터를 통하여 과학재단의 지원을 받았음.

리(key tree)를 이용한 방법[5,6]과 키트리를 이용하지 않는 방법[1,2]이 있을 수 있다. 중앙방식이면서 키트리를 이용한 방법[7]과 비중앙방식이면서 키트리를 이용한 방법[4]도 있다. 키 동의 프로토콜이면서 키트리를 이용한 방법은 모든 멤버가 독립적으로 모듈러 멱승 연산의 부담을 가지고 있지만, 그룹에 변화가 일어나 키를 바꾸어야 할 때, 키를 빠르게 갱신시킬 수 있다는 장점이 있다. 그러나 무엇보다 각각의 멤버가 키트리 높이만큼의 키들을 유지하고 있어야 한다는 단점이 있다. 본 논문과 관련된 키트리 이용하지 않은 방법은 GDH(generic Diffie-Hellman)시리즈가 있고 [1], Hypercube와 Octopus 프로토콜 등이 있다[2]. 이 방법들은 모두 모듈러 멱승 계산의 부담을 갖고 있다.

2. 제안 프로토콜

1) 표기

다음은 본 논문에서 사용되는 여러 가지 기호이다. 간략하게 설명하면, 그룹키는 $F(X_i)$ 와 $f(x_i)$ 의 두 함수(function)에 의해서 생성된 값이다. 다시 말해, 그룹키는 $F(X_i)$ 의 곱들이고, $F(X_i)$ 는 $f(x_i)$ 곱이 된다. 함수 $f(x_i)$ 는 두 멤버간의 Diffie-Hellman 키 교환으로 얻어지는 비밀 값이다.

어떤 두 멤버 m_i, m_{i+1}가 있을 때, 각 멤버가 Zq상에서 랜덤하게 생성한 비밀 값 r_i, r_{i+1}을 생성한다. 그런 다음, 각 멤버는 $g^{r_i} \bmod p$ 와 $g^{r_{i+1}} \bmod p$ 를 계산하여 서로 값을 교환하게 되고, 두 멤버 m_i, m_{i+1}은 자신만이 알고 있는 랜덤 값을 멱승함으로써 같은 함수 값 $f(x) = g^{r_i r_{i+1}} \bmod p$ 를 계산할 수 있다. 이것이 잘 알려진 두 멤버간의 Diffie-Hellman 키 교환 프로토콜이다. 이 과정을 마친 두 멤버 사이에는 비밀 키가 하나 생기게 된다. 마지막의 ID_i는 그룹을 구성하고 있는 한 멤버가 가지고 있는 정보이다. 그룹 구성원은 단지 트리가 구성되었을 때의 노드 번호(i)와 그룹키, 그리고 자기 자신의 부모와 자식이 누구인지만 알면 된다.

N	전체 그룹의 멤버들의 수
l	트리의 레벨 $l \in [1, \log_2 M]$
R,L	오른쪽 자식 노드와 왼쪽 자식 노드
m_i	i-th의 그룹 멤버 $i \in [1, M]$
G	소수 q의 위수를 갖는 순환 대수군(cyclic algebraic group)
q	대수군(algebraic group)의 위수(소수)
p	큰 소수
g	지수 기저: G의 생성자
r_i	Zq상에서 m_i의해 랜덤하게 생성된 비밀 값
$f(x_i)$	두 노드간의 Diffie-Hellman 키 교환에 의해 생성된 값
$F(X_i)$	i번째 내부노드가 둘 이하의 자식 노드와 키 교환한 값의 곱
KEY	그룹키
ID_i	i번째 멤버에 관련된 모든 정보를 가지고 있는 데이터

2) 그룹키 동의 프로토콜

본 논문에서 제안한 프로토콜을 수행하기 위해서 [그림 1]과 같이 논리적인 완전 이진 트리를 생성한다. 그리면 모든 노드는 자기의 ID_i에 해당하는 정보를 얻게 되고, 그런 다음 [프로토콜 1]을 수행하게 되는데, 모두 3단계로 이루어져 있다. 첫 번째 단계는 부모와 자식간의 Diffie-Hellman 키 교환을 하는 단계이고, 두 번째 단계는 트리의 각 내부 노드들이 자기 부모에게 Diffie-Hellman 키 교환을 한 결과값의 곱을 전달하여 그룹키를 생성하는 단계이고, 마지막 단계에서는 근 노드(root node)에서부터 단말노드(terminal node)까지 두 노드간에만 알고 있는 비밀키를 이용하여 안전하게 그룹키를 전송하는 단계이다.

[정리 1] 한 그룹에 있는 멤버가 N 명인 N 개의 노드를 가진 완전 이진 트리 [그림 1]과 같이 차례로 표현되어 있다면, m_i는 다음의 정보를 알 수 있다. (단, $1 \le i \le N$)

- $i \neq 1$이면 m_i는 $\lfloor \frac{i}{2} \rfloor$ 의 위치에 있게 되고, 만약 $i=1$이면 i는 근 노드이므로 부모가 없다.
- $2i \le N$ 이면 왼쪽 자식은 $2i$의 위치에 있게 되고, 만약 $2i > N$ 이면 i는 왼쪽 자식을 가질 수 없다.
- $2i+1 \le N$ 이면 오른쪽 자식은 $2i+1$의 위치에 있게 되고, 만약 $2i+1 > N$ 이면 i는 오른쪽 자식을 가질 수 없다.

[프로토콜 1] 그룹키 동의 프로토콜

· 단계 1. 두 멤버간의 Diffie-Hellman 키 교환

그룹을 구성하는 모든 m_i들은 다음의 함수들을 계산한다.

$$f(x_i)_L = g^{r_i r_{2i}} \bmod p$$

$$f(x_i)_R = g^{r_i r_{2i+1}} \bmod p$$

$$f(x_i) = g^{r_i r_{\frac{i}{2}}} \bmod p$$

· **단계 2. 그룹키 생성**

for $l = \lfloor \log_2 N \rfloor$ to 1*

for $i = 2^{l-1}$ to 2^{l-1}

$F(X_i) = f(x_i)_L f(x_i)_R F(X_{2i}) F(X_{2i+1})$

※ 널 노드(null node)에 해당하는 함수 값은 1로 한다.

※ * : 각각의 메시지는 $f(x_i)$를 이용한 암호문이다.

· **단계 3. 그룹키 전달**

$KEY(= F(X_1))$를 근노드에서 단말노드까지 비밀값 $f(x_i)$을 이용하여 전송한다.

[그림 1]과 같이 그룹이 구성되어 있을 때, 그룹키를 생성하는 과정을 간략하게 설명하겠다.

첫 번째 단계로, 각 노드는 함수 $f(x_i)$를 계산하여 부모 노드(parent node)와 자식 노드(child node)간의 비밀값을 얻게 된다. 예를 들면, m_4는 다음의 함수값들을 얻게 된다

$$f(x_4)_L = g^{r_4 r_8} \bmod p$$

$$f(x_4)_R = g^{r_4 r_9} \bmod p$$

$$f(x_4) = g^{r_4 r_2} \bmod p$$

두 번째 단계에서는 내부 노드가 첫 번째 단계에서 얻은 함수값들을 곱해서 자신의 부모 노드에게 전달하게 된다. 예를 들면 m_4 는 m_2 에게 함수값 $F(X_4) = f(x_4)_L f(x_4)_R F(X_8) F(X_9)$을 $f(x_4)$ 값을 이용하여 암호화하여 값을 전달하게 된다.

마지막 단계에서는 두 번째 단계에서 생성된 그룹키 $F(X_1)$*를 근 노드에서부터 단말 노드까지 암호화하여 전달하게 된다. 그리하여, [그림 1]과 같은 그룹은 공통으로 그룹키 KEY를 갖게 된다.

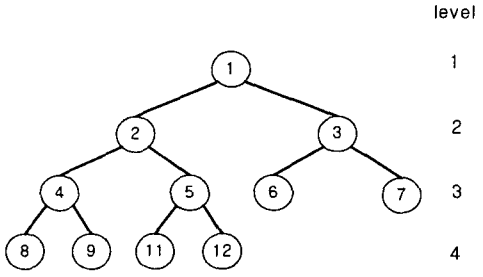

그림 1. 그룹의 논리적인 구성도

3) 안전성 및 복잡도 계산

제안한 본 프로토콜의 안전성은 Diffie-Hellman 문제의 안전성에 기반하고 있고, 이 Diffie-Hellman 문제의 어려움은 이산 대수의 어려움으로 귀착된다. [표 1]은 키트리를 이용한 방법을 제외하고, 본 논문과 같이 모든 멤버가 차례대로 그룹키를 만드는데 참여한 다른 관련 연구들과 비교한 것이다. 이것은 어떤 그룹이 안전한 통신을 하기 위해서, 처음으로 키를 나누어 갖는 경우에 대한 복잡도 계산이다. 아래 표에서 볼 수 있듯이, GDH에 비해 전체 모듈러 멱승 계산량이

	Proposed	2^d Hypercube	2^d Octopus	GDH.1	GDH.2
Round	$1+2\log_2 N$	d	2+d	2N-2	N
Total Message	2N-L	dN	$3(N-2^d)+2^d d$	2N-2	N
Total Bandwidth	5N-4-L	dN	$3(N-2^d)+2^d d$	N(N-1)	$\frac{N(N+3)}{2}-3$
Total Exponentiation	4N-M+3	$dN+2^d$	$2(N-2^d)+2^{d-1}d$	$\frac{N(N+3)}{2}-1$	$\frac{N(N+3)}{2}-1$
Exponentiation*	4	d+1	$d+(N-2^d)$	i+1	i+1
Total Multiplication	N-L				
Multiplication*	1				

$M = 2^l$, $L = 2^{l-1}$, * : per member

표 1. 복잡도 비교

* $F(X_1) = g^{r_1 r_2 + r_1 r_3 + r_2 r_4 + r_2 r_5 + r_3 r_6 + r_3 r_7 + r_4 r_8 + r_4 r_9 + r_5 r_{11} + r_5 r_{12}} \bmod p$

줄어 들었고, 한 멤버가 수행할 모듈러 멱승 연산과 곱셈 연산 양은 일정하다. 즉, 단말 노드일 경우는 모듈러 멱승 연산은 2번이고, 내부노드일 경우는 4번 그리고 근노드일 경우는 3번이 된다. 또한 곱하기 연산은 내부노드에서만 1번씩만 일어난다.

Ⅲ. 결론 및 향후 과제

본 논문에서 제안한 프로토콜은 모든 멤버가 참여하는 그룹키 동의 프로토콜로 기존의 관련 연구에 비해 모듈러 멱승 연산의 횟수를 줄였을 뿐만 아니라, $O(\log_2 N)$의 작은 라운드 수를 가지는 간단한 프로토콜이다.

그룹 통신에서 기본적인 키 분배 다음으로, 가장 중요한 것이 그룹 멤버의 관리이다. 즉, 한번 생긴 그룹은 고정된 멤버를 가지는 것이 아니라, 가입, 탈퇴 등과 같은 여러 가지 요인에 의해서 변하기 마련이다. 이런 상황에 잘 적용할 수 있는 좀 더 발전된 프로토콜이 필요하다.

또한, 안전한 그룹 통신을 위해 연구하는 분야에는 크게 3 가지로, 키 분배 알고리즘, 안전한 멀티 캐스트, 구현 영역으로 구분된다. 본 논문은 키 분배 알고리즘에 해당되며, 나아가 모든 분야에 걸쳐 연구가 계속되어야 할 것이다.

참고 문헌

[1] M.Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication", in ACM symposium on Computer and Communication Security, March 1996.

[2] C.Becker and U.Wille, "Communication Complexity of group key distribution", in ACM conference on Computer andCommunication Security, November 1998.

[3] S.Mittra, "Iolus: A Framework for Scalable Secure Multicasting", In Proceedings of ACM SIGCOMM'97, pages 277-288, September 1997.

[4] O.Rodeh, K.P. Birman and D.Dolv, "Optimized Goup Rekey for Group Communication Systems", In Proceedings of Network and Distributed System Security Symposium(NDSS'00), February 2000.

[5] Y.Kim, A.Perrig and G.Tsudik, "Simple and Fault-tolerant Key Agreement for Dynamic Collaborative Groups", In ACM CCS 2000, November 2000.

[6] Y.Kim, A.Perrig and G.Tsudik, "Communication-efficient group key greement", In Proceedings of IFIP SEC 2001, June 2001.

[7] Wong, C.K.,Gouda, M.and Lam, S.S.,"Secure group communication using key graphs", In IEEE/ACM Transactions on Networking, VOL. 8, NO.1, February 2000.