情報保護學會論文誌
第13券 第3號, 2003. 6

Pairing을 이용한 트리 기반 그룹키 합의 프로토콜

이 상 원*, 천 정 희**, 김 용 대***

Tree-based Group Key Agreement Protocol using Pairing

Sang-won Lee*, Jung Hee Cheon**, Yongdae Kim***

요 약

안전하고 안정적인 그룹 통신은 최근 그룹 및 그룹 구성원간의 협조가 필요한 응용 분야가 발전하면서 점차 그 필요성이 대두되고 있다. 이중 가장 중요한 문제는 그룹 내의 키 관리 문제이다. 센터에 의존하는 키 관리 방식의 경우 대용량의 멀티캐스트 그룹에 어울리는 반면 구성원간의 협조에 의하여 이루어지는 그룹(즉, 센터가 없는 그룹)의 경우 분산 키 관리 방식(즉 그룹 키 합의 방식)이 필요하다. 기존의 그룹 키 합의 방식의 경우 계산량을 효율적으로 하는 데만 그 연구가 치중되어 있는 실정이다. 단 한가지 예외는 STR 프로토콜로 이 방식의 경우 Diffie-Hellman protocol을 키 트리에 응용하되 키 트리가 한쪽으로 치우친 구조를 가지고 있어 통신량을 최적화하고 있다. 하지만 계산량에 있어서 그룹 멤버의 변경 시 현재 그룹 멤버의 수에 비례한 계산량을 필요로 한다. 본 논문에서는 STR 키 관리 방식에 pairing을 응용하여 계산량에 있어서 효율성을 제공하는 동시에 통신량을 유지할 수 있는 새로운 방식을 제시한다.

ABSTRACT

Secure and reliable group communication is an increasingly active research area prompted by the growing popularity of many types of group-oriented and collaborative applications. The central challenge is secure and efficient group key management. While centralized methods are often appropriate for key distribution in large multicast-style groups, many collaborative group settings require distributed key agreement techniques. Most of prior group key agreement protocols have been focused on reducing the computational costs. One exception is STR protocol that optimizes communicational cost. On the other hand, it requires O(n) number of modular exponentiations. In this paper, we propose a new group key agreement protocol that modifies STR protocol by utilizing pairing based cryptography. The resulting protocol reduces computational cost of STR protocol while preserving the communication cost.

Keyword : *Group key agreement, STR, Bilinear map, Pairings, BDH*

Ⅰ. 서 론

최근 컴퓨팅 환경에서는 급속한 통신 기술의 발달과 함께 신뢰할 수 있는 통신이 중요시되고 있으며 전자우편이나 파일공유와 같은 중앙 집중적인 서비스는 근래에는 다중의 서버와 네트워크를 통해 제공되는 분산시스템으로 바뀌고 있다. 이러한 새로운 분산되고 협동적인 응용프로그램들은 안전한 통신을 필요로 한다. 현재 그룹에 기반한 어플리케이션들은 화상회의, 다중 사용자 게임, 인터랙티브 채팅 등이

* 한국정보통신대학원대학교 공학부(swlee@icu.ac.kr)
** 서울대학교 수리과학부(jhcheon@math.snu.ac.kr)
*** Department of Computer Science at University of Minnesota at Twin Cities(kyd@cs.umn.edu)

있다. 이러한 어플리케이션들이 필요한 보안 요구사항은 전형적인 기밀성, 자료의 무결성, 인증 그리고 접근제어와 같은 것들이다. 이러한 서비스는 그룹이 하나의 키를 소유할 때 이루어 질 수 있다.

구성원의 멤버십이 자주 바뀌고, 모든 구성원이 동일한 권리와 의무를 가지며 또한 작은 규모이며 어떠한 구성원이든 수신자나 송신자가 될 수 있는 그룹을 DPG (Dynamic Peer Group)라 하며 우리는 이러한 DPGs 환경에서의 보안 요구사항 중에서 그룹 키 관리에 주목한다. 모든 구성원들은 네트워크 오류나 지연 등의 이유로 인한 그룹의 분할과 같은 동적인 그룹 이벤트에 대처할 수 있어야 한다.

지금까지 DPG 상의 그룹 키 관리 방식은 대부분 계산 오버헤드를 줄이는 방향으로 연구되어 왔으며, 하드웨어의 발달과 함께 많은 발전이 있었다. 반면 이에 비해 통신 지연은 그에 따라갈 만큼 발전하지 못했다. 네트워크 기기와 설비는 상당히 빨라지고 값싸졌으며 게다가 어디서나 통신에 접속할 수 있는 환경이 되었다. 이는 결국 엄청난 네트워크 대역폭을 요구하게 되었다. 특히 최근 P2P 응용 서비스의 발전은 네트워크 대역폭의 50% 이상을 차지하고 있다. 이로 인해 네트워크 정체는 크게 줄고 있지 않은 상황이다. 예를 들어 한국과 미국의 왕복 통신의 경우 최소한 200ms 정도 소요되고 있다. 이러한 점을 주목할 경우 통신 오버헤드를 줄이는 것이 오히려 암호 프로토콜을 효율성을 높이는 데 계산량을 줄이는 것이 비교하여 더 큰 효율성을 보일 수 있음을 알 수 있다.

본 논문에서 제안하는 암호 프로토콜은 Y. Kim등에 의해 제안된 STR 그룹 키 합의 프로토콜을 기반으로 한다[8]. STR은 간단하며, 안전하며 통신에 최적화 되어 있는 프로토콜이다. 하지만 STR의 경우 계산량이 현재 사용자의 숫자에 비례한다. (즉, 4 x 현재 그룹 멤버 수)

한편 pairing은 아이디 기반의 암호화와 서명 개념 또한 아이디 기반의 양자간 키 합의 프로토콜을 포함한 여러 암호학적 근간을 생성하는 데 사용되어 왔다. 본 논문에서는 pairing에 이용하여 STR의 계산량을 줄이는 방식을 제안한다. 현재 pairing 계산의 경우 모듈라 지수 승에 비하여 더 많은 계산량이 필요하나 현재 pairing 계산의 경우 계산량에 대한 연구가 초기 단기임을 주시할 경우 미래에 본 논문에서 제시하는 그룹 키 합의 프로토콜은 기존의 STR 프로토콜에 비하여 더욱 효율적인 그룹 키 합의 프로토콜이 되리라 예상한다.

이 논문의 나머지는 다음과 같이 구성된다. 2장에서는 그룹키에 관련한 연구들을 설명하고, 3장에서 자세한 프로토콜에 대한 내용을 소개한다. 4장에서는 성능에 대해 언급하고 5장에서 결론을 통해 이 논문을 맺는다.

II. 관련연구

2.1 그룹키 관리

DPG 환경에서 그룹키를 관리하는 방법은 크게 그룹키 분배와 그룹키 합의로 나눌 수 있다. 중앙 그룹키 분배(Centralized Group Key Distribution) 방식은 하나의 서버가 각 구성원의 키를 생성하여 이를 배포하는 방식이다. 키 서버는 실제 키 분배를 위한 안전한 양자간 통신을 위해 각 그룹 구성원과 장기 공유키(long-term shared key)를 유지한다. 이러한 방법의 한 가지 유형은 고정된 TTP(Trusted Third Party)를 키 서버로서 사용하는 것이다. 이러한 접근은 항상 서버가 가용해야 한다는 것과 네트워크 분할 이벤트에서 계속적인 운용을 지원하기 위해 모든 가능한 그룹의 부분집합에 TTP가 존재해야 한다는 두 가지 문제점을 갖는다.

이러한 중앙 그룹키 분배 방식의 변형으로, 키를 생성하고 분배하는 키 서버를 동적으로 선택할 수 있는 방식인 비 중앙 그룹키 분배(Decentralized Group Key Distribution)방식이 있다[4,10]. 이 방식은 키 서버를 선택함으로써 어떠한 그룹분할 상황에서도 계속적으로 운영될 수 있기 때문에 좀더 견고하여, 결국 다대다 그룹에 더 적용 가능하다. 하지만 중앙 그룹키 분배방식과 마찬가지로, 키 서버는 반드시 그룹키 배분을 위한 모든 그룹 구성원과의 장기간의 안전한 쌍방향 채널을 가져야 한다는 단점을 가진다. 결국 매번 새로운 키 서버가 선택되어 그룹키 분배 역할을 하게 되어 쌍방향의 안전한 채널을 만드는데 심각한 비용이 초래된다. 또 다른 단점으로는 안전한 (예로, 암호학적으로 강한) 키를 생성하는 한 개체에 대한 의존성이다.

위의 접근에 대조적으로, 그룹키 합의(Group Key Agreement)방식은 각 그룹 구성원이 공통의 그룹키 (이는 모든 구성원들의 기여의 함수로서 계산된다.)에 동등한 몫을 기여하는 것을 요구한다. 따라서 신용의 집중과 한 지점에서의 오류의 문제를 피할 수

있다. 또한 몇몇의 기여 그룹키 관리 방법은 그룹 구성원들 간의 안전한 쌍방향 채널의 수립을 요구하지 않는다[9].

2.2 그룹 멤버십 이벤트

복잡한 그룹키 합의 솔루션은 기초를 이루는 그룹 통신 시스템에서의 모든 멤버십 변화에 따른 그룹 안전에 대한 조정을 다룰 수 있어야 한다.

구성원에 대한 멤버십은 단일 또는 다중 구성원에 대한 조작으로 구분된다. 단일 구성원 변화는 구성원의 추가와 삭제를 나타내며 구성원이 그룹에 가입 또는 탈퇴를 원할 때 발생한다. 다중 구성원 변화 또한 추가와 삭제를 포함하며 둘 또는 더 많은 그룹이 하나의 그룹을 형성하는 경우를 그룹병합으로 간주하고 하나의 그룹이 작은 그룹으로 나뉘는 것을 그룹분할이라 간주한다. 그룹의 병합과 분할은 네트워크의 오류와 이러한 오류의 수정 등의 이유로 빈번히 발생하게 된다. 결국 그룹병합과 분할을 다루는 것은 그룹키 합의에서 매우 중요한 요소 중의 하나이다.

추가적으로 멤버십의 운영에 있어서 같은 키로 생성되는 암호문의 양을 제한하고 구성원의 공모 등을 막기 위해 그룹키의 주기적인 갱신이 필요하다. 따라서 구성원의 가입과 탈퇴, 그룹의 병합과 분할 그리고 그룹키 갱신의 멤버십 이벤트가 발생하게 된다.

DPG 환경에서 동적 멤버십을 관리하기 위하여 스폰서 개념을 사용한다. 어떠한 구성원이든 스폰서가 될 수 있으며 각 이벤트에 따라 스폰서가 정해진다. 스폰서는 이벤트 처리를 위한 중추적 역할을 하며 나머지는 일반 구성원과 동일하다.

2.3 보안 요구 사항

동적인 특성을 갖는 그룹에 대한 키 관리방법이 제공해야 하는 특성은 다음과 같다.

- Group key secrecy : 그룹키는 그룹 구성원들만이 공유하여야 한다.
- Forward secrecy : 과거 그룹키들의 일부를 알더라도 이후의 그룹키를 알아낼 수 없다.
- Backward secrecy : 과거 그룹키들의 일부를 알더라도 이전의 그룹키를 알아낼 수 없다.
- Key Independent : 적절한 그룹키들의 일부를 알고 있더라도 어떠한 그룹키도 알아낼 수 없다.

그룹키 안전성(Group Key Secrecy)는 그룹키 특성상 기본적으로 제공하는 성질이다. 전방위 안전성(Forward Secrecy)은 구성원의 탈퇴(또는 그룹분할)의 경우, 탈퇴한 구성원이 이전의 공유된 그룹키로부터 탈퇴(또는 그룹분할) 이후의 그룹키를 생성할 수 없도록 하는 조건이며, 후방위 안전성(Backward Secrecy)은 새로운 구성원이 가입(또는 그룹병합)하여 그룹키를 공유하게 되더라도 이전의 그룹키를 알아낼 수 없도록 하기 위한 조건이다. 그룹키 독립성(Key Independent)은 전방위 안전성과 후방위 안전성 모두를 포함한다.

2.4 타원곡선을 이용한 키 합의

먼저 겹선형성에 대하여 설명하고, 키 합의에 필요한 두 가지 문제에 대하여 알아본다.

소수 위수 q의 덧셈 군을 G_1이라 하고 동일한 위수 q의 곱셈 군을 G_2라 한다. $G_1 \times G_1$에서 G_2로의 효과적으로 계산 가능한 겹선형의 사상 $\hat{e}$의 확장을 가정한다. 전형적으로 G_1는 유한체 상의 타원곡선 위의 점 군의 부분 군이 될 것이다. G_2는 관련된 유한체의 곱셈 군의 부분 군이 될 것이고 $\hat{e}$는 타원곡선 상의 Weil이나 Tate pairing으로부터 파생될 것이다. 우리는 또한 G_1에 속하며 $\hat{e}(P,P) \neq 1_{G_2}$를 만족하는 P가 알려져 있다고 가정한다. 겹선형인 $\hat{e}$에 의해, 모든 $Q, P \in G_1$과 $a, b \in Z_q^*$에 대해

$$\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$$

이다.

겹선형 디피-헬만 (BDH: Bilinear Diffie-Hellman Problem) 문제는 P, aP, bP, cP이 주어졌을 때 $\hat{e}(P,P)^{abc}$을 계산하는 문제이다. 삼자간의 키합의는 이러한 겹선형 디피-헬만 문제의 어려움에 기반한다[2].

타원곡선 상에서 양자간 키 교환은 타원곡선 위의 디피헬만(ECDH: Elliptic Curve Diffie-Hellman) 문제에 기반하는데, 미 문제는 P, aP, bP가 주어졌을 때 abP를 계산하는 문제이다[7].

Ⅲ. 제안 프로토콜

3.1 기본개념

이 논문에서 사용하는 표기법은 [표 1]과 같다. [그림 1]은 키-트리(Skinny Key Tree)의 예를 보인다. 키-트리는 다음과 같은 노드로 구성되는 트리이다.

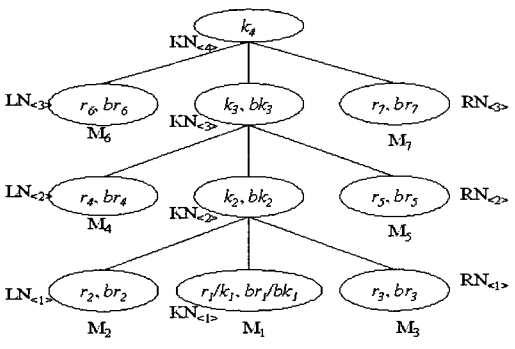
〔표 1〕 표기법

n, N	그룹 구성원들의 수
M_i	i번째 그룹 구성원; $i \in \{1,...,N\}$
r_i	M_i의 비밀키
br_i	M_i의 은닉 비밀키, 예로 r_iP
k_j	$M_1...M_{2j+1}$간에 공유된 비밀키
bk_j	은닉된 k_j, 예로 k_iP
P	공개정보, 타원곡선 상의 한점
H_1	해쉬함수, $H_1 : G_2 \rightarrow Z^*_q$
H_2	해쉬함수, $H_2 : G_1 \rightarrow Z^*_q$
$KN_{<l>}$	레벨 l에서의 키-노드
$LN_{<l>}$	레벨 l에서의 왼쪽 구성원-노드
$RN_{<l>}$	레벨 l에서의 오른쪽 구성원-노드
$T_{<i>}$	구성원 M_i의 트리
$BT_{<i>}$	구성원 M_i의 은닉키를 모두 포함한 트리

- 구성원-노드 : 말단노드로서 그룹 구성원을 표현한다.
- 키-노드 : 내부 노드로서 하나의 키에 대응된다. 이 키는 이 노드를 루트로 하는 부트리의 모든 구성원이 공유한다.
- 루트 : 그룹키를 나타낸다.

키-노드 $KN_{\langle l \rangle}$는 세 개의 자식노드를 가진다: 하위 키-노드 $KN_{\langle l-1 \rangle}$과 두개의 구성원-말단노드 $RN_{\langle l-1 \rangle}$, $LN_{\langle l-1 \rangle}$. 다만 가장 아래 중앙에 위치해 M_1과 연결된 $KN_{\langle 1 \rangle}$은 예외이다.

구성원-노드 $RN_{\langle i \rangle}(LN_{\langle i \rangle})$은 $M_{2i}(M_{2i+1})$에 의해 선택되고 비밀리에 유지되는 비밀키 r_i를 가진다. 이 비밀키의 공개정보는 $br_i(=r_iP)$이며 은닉키라 부른다. 모든 키-노드는 비밀키 k_i와 은닉키 $bk_i(=k_iP)$를 가진다. 이 비밀키 $k_i(i\rangle 1)$는 하위 노드들의 pairing 계산 결과의 해쉬값이다.



〔그림 1〕 트리구조

[그림 1]에서 루트노드에 연결되는 그룹키는 다음과 같은 형태를 갖는다:

$$k_4 = H_1(\hat{e}(P,P)^{r_6 r_7 H_1(e(P,P)^{r_4 r_5 H_1(e(P,P)^{r_1 r_2 r_3})})})$$

기본적인 그룹키 프로토콜은 다음과 같다. 우리는 모든 구성원들이 키-트리의 구조와 트리 내에서의 위치를 알고 있다고 가정한다. 각각의 구성원은 자신의 비밀키를 알고 또한 모든 다른 구성원의 은닉키와 은닉 비밀키도 안다. 최초 구성원 M_1, M_2 그리고 M_3는 $KN_{\langle 2 \rangle}$에 연결된 그룹키를 계산할 수 있다. M_1은 다음을 계산할 수 있다:

$$k_1 = r_1$$
$$k_2 = H_1(e(r_2P, r_3P)^{r_1}) = H_1(e(P,P)^{r_1 r_2 r_3})$$
$$k_3 = H_1(e(br_4, br_5)^{k_2}) = H_1(e(P,P)^{k_2 r_4 r_5})$$
$$\cdots$$
$$k_l = H_1(e(P,P)^{k_{n-1} r_{2n-2} r_{2n-1}})$$

다음으로 M_1은 모든 은닉키 bk_i $(1 \le i \le \frac{N-1}{2})$와 br_j $(1 \le j \le N)$를 담은 $BT_{\langle 1 \rangle}$을 브로드캐스트한다. 이 메시지를 받음으로써 모든 구성원들은 그룹키 k_l을 계산할 수 있다.

구성원 수에 의해 최상위 레벨에서 자식노드의 수가 둘 또는 셋이 된다. 각 경우에 대해 불균형트리와 균형트리라 한다. 불균형트리의 경우는 페어링 계산을 사용할 수 없으며 이 경우는 타원곡선 디피-헬만을 이용한 양자간의 키 합의방법을 사용한다. 예로, 그룹에 $\{M_1, M_2, \cdots, M_6\}$의 구성원이 있는 경우, r_6P와 k_3P는 공개정보이므로 $\hat{e}(k_3P, r_6P)$는 누구나 계산할 수 있는 정보(심지어 공격자까지도)이다. 따라서 우리는 양자간의 합의키인 r_6k_3P를 그룹키로 사용한다($k_4 = H_2(r_6k_3P)$).

또 다른 방법으로 스폰서가 하나의 더미노드를 더 생성하여 위의 문제를 해결할 수 있다. 스폰서 M_s는 자신의 은닉 비밀키 r_s외에 $r_{s'}$을 더 생성하고 그에 대한 은닉키 $r_{s'}P$를 $BT_{\langle s \rangle}$와 함께 브로드캐스트함으로써 모든 구성원은 pairing 계산만으로 그룹키를 계산할 수 있다. 모든 구성원은 키-트리를 항상 균형트리로 유지할 수 있다. 본 논문에서는 타원곡선 디피-헬만을 이용한 양자간의 키 합의방법을 사용하여 균형트리와 불균형트리의 경우에 대한 그룹 이벤트 운

영에 대하여 논의한다.

3.2 가입

구성원이 n명인 그룹 $\{M_1, M_2, \cdots, M_n\}$이 있다고 가정한다. 새로운 구성원 M_{n+1}이 그룹 통신 시스템에 가입하게 되면 새로운 구성원과 기존의 구성원 모두 이러한 공지 메시지를 받게 된다. 새로운 구성원 M_{n+1}은 자신의 은닉 비밀키 br_{n+1}을 포함한 가입 요청 메시지를 브로드캐스트한다.

〔표 2〕 가입 프로토콜

단계 1: 새로운 구성원은 가입에 대한 요청을 브로드캐스트한다.

$$M_{n+1} \xrightarrow{br_{n+1}} C=\{M_1, \cdots, M_n\}$$

단계 2: 기존의 트리 형태가

균형트리의 경우

- 모든 구성원은 새로운 노드를 기존의 루트노드에 추가함으로써 키-트리를 갱신한다.
- 스폰서 M_n은 $BT_{\langle n\rangle}$을 브로드캐스트한다.

$$M_n \xrightarrow{BT_{\langle n\rangle}} C \cup\{M_{n+1}\}=\{M_1, \cdots, M_{n+1}\}$$

불균형트리의 경우

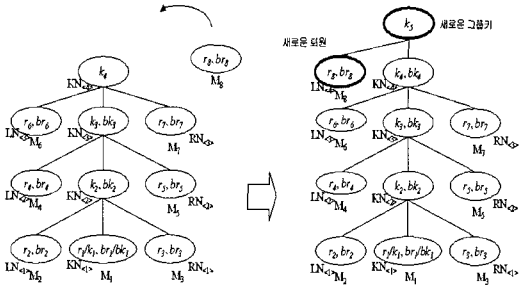
- 모든 구성원은 새로운 노드와 루트노드를 추가함으로써 키-트리를 갱신한다.
- 스폰서 M_n은 새로운 비밀키 r_n을 생성하고 br_n, k_n, bk_n을 계산한다.
- 스폰서 M_n은 갱신된 트리 $BT_{\langle n\rangle}$을 브로드캐스트한다

$$M_n \xrightarrow{BT_{\langle n\rangle}} C \cup\{M_{n+1}\}=\{M_1, \cdots, M_{n+1}\}.$$

단계 3: 모든 구성원은 $BT_{\langle n\rangle}$을 이용하여 그룹키를 계산한다.

이때, 기존 트리의 형태에 따라 두 가지 경우가 발생한다. 우선 불균형트리의 경우, 각각의 구성원 M_i는 빈자리에 새로운 구성원을 넣음으로써 쉽게 그룹키를 계산할 수 있다. 스폰서는 bk_n의 계산 없이 은닉키-트리 $BT_{\langle n\rangle}$을 M_{n+1}에게 브로드캐스트한다.

하지만 균형트리의 경우에 각각의 구성원 M_i는 새로운 루트노드 $KN_{\langle n/2+1\rangle}$를 생성한 후 루트노드에 새로운 구성원을 추가한다. M_n는 자신의 비밀키를 새롭게 하여 br_n, k_n, bk_n을 계산한 후 모든 은닉키와 은닉 비밀키를 담은 은닉키-트리 $BT_{\langle n\rangle}$을 M_{n+1}을 포함한 그룹전체에 보낸다. 가입 프로토콜을 [표 2]에 자세히 기술되어있다.



〔그림 2〕 가입 시 트리갱신

모든 기존의 구성원들은 단지 새로운 구성원의 은닉 세션 비밀키(br_{n+1})만이 필요하며 새로운 구성원은 기존 그룹의 은닉 그룹키(bk_n)와 스폰서의 은닉 비밀키(br_n)가 필요하다. 이러한 이유로 모든 구성원은 그룹키($k_{n+1}=H_2(r_{n+1}k_nP)$)를 계산할 수 있다.

가입 과정에서, 스폰서는 항상 최상위 오른쪽 노드이다(없으면 왼쪽노드). 예로 기존 가입과정에서 가장 최근에 가입한 구성원이 스폰서가 된다.

[그림 2]는 새로운 구성원이 그룹에 가입하는 예를 보인다. 스폰서 M_7은 자신의 세션 비밀키 r_7을 갱신한 후, br_7, $k_4(=\hat{e}(br_6, bk_3)^{r_7})$, bk_4를 계산한다. 스폰서는 모든 은닉키와 은닉 비밀키를 담은 $BT_{\langle 7\rangle}$을 브로드캐스트한다. 모든 각각의 구성원은 이 메시지를 받음으로써 새로운 그룹키($k_5=H_2(r_8k_4P)$)를 계산한다.

가입 프로토콜은 두 번의 통신 라운드와 새로운 그룹키를 계산하기 위한 5(3)번의 암호학적 과정(스폰서에 의해 4(2)번과 다른 모든 구성원에 의해 1번)을 거친다. 가입 프로토콜은 전방위 및 후방위 안전성을 제공한다.

3.3 탈퇴

n명의 구성원을 가진 그룹에서 구성원 M_d가 그룹을 떠난다고 가정한다. 탈퇴의 경우 스폰서는 가장 최상위 오른쪽 노드이다(없으면 왼쪽 노드). 예로, M_n이 스폰서가 된다. 그룹 통신 시스템으로부터 탈퇴 이벤트를 받으면, 각각의 남아있는 구성원들은 M_d와 관계되는 구성원-노드와 부모노드를 삭제함으로써 자신의 키-트리를 갱신한다. 스폰서 M_n은 탈퇴하고 빈 자리 M_d로 자신의 위치를 바꾸고 새로운 비밀키를 선택한 후, 루트노드 아래 모든 키들(그리고 은닉 키들)을 계산한다. 그런 후 $BT_{\langle n\rangle}$를 브로드캐스트한다. 이 정보는 모든 구성원들이 새로운 그

룹키를 다시 계산할 수 있도록 한다. 탈퇴 프로토콜을 [표 3]에서 자세히 기술한다.

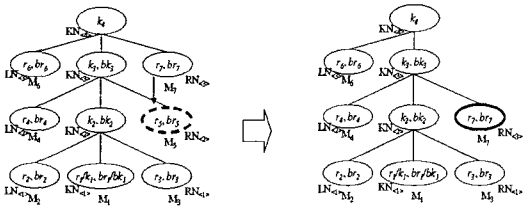
〔표 3〕 탈퇴 프로토콜

단계 1: 모든 구성원은
- 탈퇴 구성원의 노드를 삭제하고 스폰서 M_n의 위치를 이동시킴으로써 키 트리를 갱신한다.
- 스폰서 노드로부터 루트노드까지 모든 키와 은닉키를 삭제한다.

스폰서 M_n은 추가적으로
- 자신의 위치를 M_d로 이동한다.
- 새로운 비밀키 r_n'을 생성하고 br_n', k_n', bk_n'을 계산한다.
- 루트노드 아래 모든 키와 은닉키를 갱신한다.
- 갱신된 트리 $BT_{\langle n\rangle}$을 브로드캐스트한다.

$$M_n \xrightarrow{BT_{\langle d\rangle}} C-\{M_d\}$$

단계 2: 모든 구성원은 $BT_{\langle n\rangle}$을 이용하여 그룹키를 계산한다.



〔그림 3〕 탈퇴 시 트리갱신

[그림 3]는 구성원이 그룹을 탈퇴하는 예를 보인다. 구성원 M_5가 그룹을 떠나면, 그 자리로 M_7이 옮겨간다. 그런 다음 M_7는 새로운 비밀키 r_7'를 선택하고, br_7', k_3', bk_3'을 계산하고 트리 $BT_{\langle 7'\rangle}$를 갱신한 후 모든 구성원에게 브로드캐스트한다. 브로드캐스트 메시지를 받음으로써, 모든 구성원(M_7를 포함해서)은 그룹키 k_4를 계산한다. 탈퇴한 M_5는 모든 은닉키를 알고 있다 할지라도 자신의 세션 비밀키가 더 이상 그룹키의 일부가 아니기 때문에 그룹키를 계산할 수 없다.

탈퇴 프로토콜은 한번의 통신을 하고 한번의 브로드캐스트를 한다. 암호학적 비용은 분리되는 구성원의 위치, 그리고 새로운 키를 계산할 필요가 있는 남아있는 구성원의 위치에 의해 변동된다.

탈퇴시의 계산량(Point multiplication과 pairing computation)은 다음과 같다:

$$\lceil \frac{n+1}{2} \rceil - \lceil \frac{d-1}{2} \rceil \quad when\ d>1$$

$$\lceil \frac{n+1}{2} \rceil \quad when\ d=1$$

이전의 구성원들은 스폰서의 세션 비밀키가 바뀌기 때문에 새로운 키를 계산할 수 없다. 따라서 탈퇴 프로토콜은 전방위 안전성을 제공한다. 이 프로토콜은 또한 새로운 키에 대한 정보가 이전의 키를 알아내는데 사용될 수 없기 때문에 키 독립성(Key Independence)을 제공한다; 이것은 스폰서가 자신의 세션 비밀키를 새로이 갱신하기 때문이다.

3.4 병합

병합의 경우 가입의 경우와 동일하게 통신 시스템이 동시에 모든 그룹 구성원(또는 모든 그룹)에게 병합 이벤트에 대해 알린다. 큰 트리 위에 보다 작은 트리를 붙이며 만일 어떤 두개의 트리가 동일한 높이를 가지고 있다면, 각 그룹 스폰서의 인식자 비교와 같은 따로 순서 정하는 방법을 사용하여야 한다.

〔표 4〕 병합 프로토콜

단계 1: 각 그룹의 스폰서 $M_s^i(i\in[1,k])$는
- 트리 $BT_{\langle s\rangle}^i$를 브로드캐스트한다.

$$M_{si} \xrightarrow{BT_{\langle s\rangle}^i} \cup C_i$$

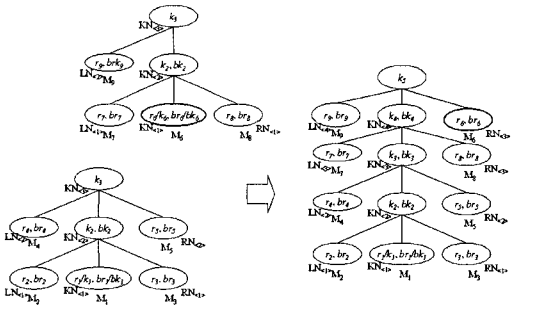
단계 2: 모든 구성원은
- 모든 트리를 병합함으로써 키 트리를 갱신한다.
- 스폰서 노드로부터 모든 키와 은닉키를 제거한다.

스폰서 M_s는 추가적으로
- 새로운 비밀키 r_s를 생성하고 br_s를 계산한다.
- 루트노드까지의 모든 키와 은닉키를 계산한다.
- 갱신된 트리 $BT_{\langle s\rangle}$를 브로드캐스트한다.

$$M_s \xrightarrow{BT_{\langle s\rangle}} \cup C_i$$

단계 3: 모든 구성원은 $BT_{\langle s\rangle}$를 이용해서 그룹키를 계산할 수 있다.

우선 하단 트리가 불균형인 경우 상위 트리의 스폰서는 M_1과 M_2를 위쪽의 빈 노드로 옮긴다. 빈 노드가 없는 경우 새로운 루트노드를 생성한 후 그 빈 노드로 옮긴다. 결국 M_3를 하단의 빈 노드에 채움으로써 트리는 하나가 된다. 만일 하단의 트리가 균형 트리인 경우 M_1만을 위쪽의 빈 노드로 옮긴다. 그 후 하단의 트리의 최상단 키 노드와 M_2, M_3을 이용하여 두 트리를 연결한다. 이 기술을 재귀적으로 사용함으로써, 우리는 여러 개의 트리를 병합할 수 있다. k-ary 병합 프로토콜은 [표 4]에서 보인다.



〔그림 4〕 병합 시 트리갱신

[그림 4]는 두개의 트리가 병합하는 예를 보인다. 병합 공지 후, 스폰서 M_5와 M_9은 모두 은닉 세션 비밀키를 포함한 그들의 키-트리를 브로드캐스트한다. 이 브로드캐스트 메시지를 받음으로써, 양쪽 그룹의 모든 구성원은 키-트리를 만들 수 있게 된다. 모든 구성원은 상위 트리의 M_6을 위의 빈자리로 옮기고 그 자리에 하위 트리의 키-노드 $KN_{\langle 3\rangle}$을 위치시킨다. 옮겨진 노드들로 인해 상위 트리는 번호가 다시 매겨져야 한다.

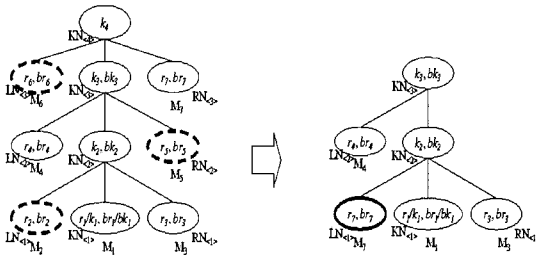
두개의 트리가 연결된 두 번째 라운드에서 스폰서 M_5는 루트노드를 제외한 모든 중간 비밀키와 은닉키값들을 계산한다. 마지막으로 모든 은닉키와 은닉 세션 비밀키를 포함한 $BT_{\langle 5\rangle}$를 브로드캐스트한다. 브로드캐스트를 받음으로써, 모든 구성원은 그룹키를 계산할 수 있다.

3.5 분할

네트워크 오류는 그룹의 분할을 초래하며 남아있는 구성원들에게 있어서 이것은 실제적으로 여러 구성원들의 동시적인 탈퇴로서 보인다. 최소한의 수정만으로 탈퇴 프로토콜을 한번의 라운드로 여러 명의 탈퇴를 다룰 수 있다. 스폰서는 남아있는 구성원 중의 가장 상단의 노드이다.

모든 탈퇴 구성원들을 삭제한 후, 스폰서 M_s는 자신을 비롯하여 최상단의 구성원들을 차례로 빈자리에 채워 넣는다. 그 후 스폰서 자신의 비밀키를 새로이 갱신하고 키와 은닉키를 계산한다. 그런 후 은닉키만을 포함한 갱신된 키-트리 $BT_{\langle s\rangle}$를 브로드캐스트한다. 각 구성원(M_s를 포함하여)은 그룹키를 계산할 수 있다.

분할 프로토콜의 계산량과 통신 복잡도는 탈퇴에서의 경우와 동일하다. 보안특성에 대해서도 동일한 사항에 대해 유효하다.



〔그림 5〕 분할 시 트리갱신

Ⅳ. 성능 및 안전성

4.1. 성능

Pairing 기반의 암호시스템에서 연산속도에 가장 영향을 많이 미치는 것은 pairing 연산이다. Pairing 연산속도에 관련 자료로는 Ben Lynn의 전자서명에 관한 실험으로 얻은 결과가 있다[3].

〔표 5〕 각종 연산 비교

	Key Size (bits)	ms	Flatform
Tate	160	26.2	PIII 1GHz, C/C++
RSA	1007	7.9	
DSA	160	4.09	

[표 5]에서 보는 바와 같이 동일한 안전성을 고려하였을 때 RSA 1024 bits와 타원곡선 $|p|$=512 bits, $|q|$=160 bits를 고려하면 RSA의 서명, 즉 복호화의 경우 약 8ms가 소요된다. 또한 F_{3^m} BLS 서명의 확인을 위해서는 두 번의 pairing 연산이 필요하므로 결국 한 번의 pairing 연산은 약 26ms가 소요된다는 결과를 얻을 수 있다[3]. Ben Lynn 실험 자료에 근거하여 비교하면 일반적인 경우에 지수연산과 pairing 연산은 3배가량 차이가 있음을 알 수 있다.

〔표 6〕 통신량의 비교

통신량		가입	탈퇴	병합	분할
STR	라운드	2	1	2	1
	메시지	2	1	k+1	1
제안	라운드	2	1	2	1
	메시지	2	1	k+1	1

[표 6]에서 제안 프로토콜은 STR과 같은 통신량을 보인다. 이는 제안 프로토콜 또한 통신에 대해 최적화되어 있음을 보이는 것이다. 가입의 경우 스폰서는 새로운 구성원에 대한 정보를 알아야 하기 때문에 2개의 메시지는 최적이다. 표에서 k는 병합되는 그룹의 수이다. 병합의 경우, 그룹끼리 정보를 교환하기 위해서 최소한 k개의 메시지는 필요하며 마지막 메시지 하나는 새로운 은닉키를 배포하기 위해 필요하다.

〔표 7〕 계산량의 비교

	STR	제안 프로토콜	
	계산량	계산량	
	지수제곱	Pairing 연산	포인트 곱
가입	4	1	$\frac{3}{2}$
탈퇴	$\frac{3}{2}n+2$	$\frac{n}{4}$	$\frac{3n}{10}$
병합	$3m+1$	$\frac{m+1}{2}$	$\frac{m+3}{2}$
분할	$\frac{3}{2}n+2$	$\frac{n}{4}$	$\frac{3n}{10}$

제안 프로토콜과 STR의 계산량에 대한 비교가 [표 7]에 정리되어 있다. 표에서 n은 현재 구성원의 수이고, m은 새로운 구성원의 수이다. 계산량에서 가입의 경우, 기존의 트리가 균형인지 불균형인지에 따라 계산량의 차이가 있다. 균형트리의 경우 새로운 구성원이 가입하게 되면 새로운 루트노드를 생성해야 하므로 한번의 포인트 곱셈과 한번의 pairing연산이 더 필요하다. 따라서 평균값을 계산한다. 탈퇴 이벤트에서의 계산량은 탈퇴 노드의 위치에 의존적이기 때문에, 결국 우리는 $n/2$번째 노드와 같은 경우의 평균값을 계산하였다.

스폰서의 입장에서 STR과 비교하면 가입의 경우 계산량이 절반이상 낮아지며, 가입의 경우를 제외한 탈퇴, 분할 그리고 병합의 경우 계산량이 약 1/6로 낮아짐을 알 수 있다. 이러한 결과는 기존의 STR과 비교하여 상대적으로 키-트리의 높이가 낮아진데서 기인한다.

하지만 이러한 계산량의 감소에도 불구하고 제안 프로토콜의 계산량은 병합 시 $O(m)$, 탈퇴와 분할 시는 $O(n)$로 높은 편이며 pairing 연산은 위에서 살펴본 바와 같이 일반적으로 지수제곱에 비해 느린 것으로 알려져 있다. 하지만 pairing 연산에 대한 고속화 연구가 진행되고 있으며 이러한 높은 계산량도 고-지연 광대역 네트워크에서는 무시되어질 수 있다.

4.2 안전성

Joux에 의해 최초 제안된 타원곡선을 이용한 삼자간의 키 합의 방법은 타원곡선의 겹선형성을 이용하였으며 이 프로토콜의 안전성은 BDH 문제의 어려움에 기반한다[5,6]. 따라서 삼자간의 키 합의 프로토콜을 그룹으로 확장한 제안 프로토콜의 안전성 또한 BDH 문제에 기반한다. 불균형트리의 경우, BDH 문제와 함께 타원곡선의 ECDH 문제의 어려움에 기반한다[7]. 이는 트리의 h-1 높이까지는 삼자간의 경우와 같은 BDH 문제와 연관되며 트리의 최상위에서는 양자간의 경우와 같은 ECDH와 연관된다. 수동적인 공격자는 모든 구성원의 은닉키와 이를 포함한 은닉 트리 정보를 얻을 수 있지만 이를 이용하여 pairing 계산(또는 타원곡선 디피-헬만키 계산)을 할 수 없으며, 결국 그룹키를 알아낼 수 없다.

제안 프로토콜에서 그룹 구성원만이 그룹키를 공유하며, 가입과 그룹의 병합 이벤트 발생 시 스폰서가 자신의 비밀키를 갱신하기 때문에 수동적인 공격자는 이전의 그룹키들로부터 새로운 그룹키에 대한 어떠한 정보도 얻을 수 없다. 따라서 후방위 안전성을 제공한다. 또한 탈퇴와 그룹의 분할 시 스폰서가 자신의 비밀키를 갱신함으로써 수동적인 공격자가 이전의 그룹키로부터 새로운 키를 알아내지 못하게 하는 전방위 보안성을 제공한다. 제안 프로토콜은 전·후방 보안성 모두를 제공하고, 결국 키 독립성을 제공한다.

V. 결 론

새롭게 제안된 프로토콜은 pairing을 사용한 삼자간 키 합의 프로토콜을 트리구조를 이용하여 그룹으로 확장시킨 것이다. 또한 STR과 마찬가지로 통신에 대해 최적화된 안전한 기여 그룹키 합의 프로토콜이다. 이 프로토콜은 가입, 탈퇴, 병합 그리고 분할과 같은 모든 동적인 그룹 조작을 지원하며 또한 STR과 마찬가지로 보안특성을 만족한다. 제안 프로토콜은 STR과 비교하여 최적화 되어있는 통신량과 적은 계산량을 보인다. pairing 계산의 부담이 있으나 하드웨어의 발달과 함께 지수연산과 같은 계산에 대한 비용은 점점 싸지고 있다. 결국 통신 지연에 대한 비

용이 프로토콜에 대한 실행시간을 결정하는데 있어 계산비용을 지배하게 된다. 결국 제안 프로토콜은 STR과 마찬가지로 고-지연 광대역 네트워크에서 효율적인 그룹키 합의 프로토콜이다. 또한 pairing을 이용한 삼자간의 키 합의를 그룹키 합의로 확장하여 적용함으로써 그룹키 합의 프로토콜에 있어서 새로운 시도가 될 것이다.

참 고 문 헌

[1] S. Al-Riyami and K. Paterson, "Authenticated three party key agreement protocols from pairings," Cryptology ePrint Archive, Report 2002/035, available at http://eprint.iacr.org/2002/035/.

[2] D. Boneh and M. Franklin. "Identity-based encryption from the Weil pairing," Advances in Cryptology-Crypto 2001, LNCS 2139, pp.213~229, Springer-Verlag, 2001. http://www.crypto.stanford.edu/~dabo/abstracts/ibe.html.

[3] P.S.L.M. Barreto, H.Y. Kim, B.Lynn, and M.Scott, "Efficient Algorithms for pairing-based cryptosystems," To appear in Cryptology-Crypto'2002, http://eprint.iacr.org/2002/008/.

[4] Wallner, Debby M., Eric J. Harder, and Ryan C. Agee, "Key management for multicast: Issues and architectures," RFC 2627, June 1999.

[5] A. Joux, "A one round protocol for tripartite Diffie-Hellman," In W. Bosma, editor, Proceedings of Algorothmic Number Theory Symposium-ANTS IV, volume 1838 of LNCS, pp.385~394. Springer-verlag, 2000.

[6] A. Joux, "The Weil and Tate Pairings as building blocks for public key cryptosystems," in Algorithm Number Theory, 5th International Symposium ANTS-V, LNCS 2369, Springer-Verlag, 2002, pp. 20~32.

[7] N. Koblitz, "Elliptic curve cryptosystems," Mathematics of Computation, vol. 48, pp.203-209, 1987.

[8] Y. Kim. A. Perrig and G. Tsudik, "Communication-Efficient Group Key Agreement," IFIP SEC 2001, Jun. 2001.

[9] Y. Kim, A. Perrig, G. Tsudik, "Tree-based Group Diffie-Hellman Protocol," In Submission.

[10] A. Perrig, D. Song, and J. D. Tyger, "ELK, a New Protocol for Efficient Large Group Key Distribution," In 2001 IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 2001.

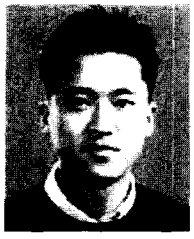
[11] N.P. Smart, "An identity based authenticated key agreement protocol based on the weil pairing," Election. Lett., Vol.38, No.13, pp.630-632, 2002.

[12] F. Zhang, S. Liu, "ID-Based One Round Authenticated Tripartite Key Agreement Protocol with Pairings," Cryptology ePrint Archive, Report 2002/122.

〈著者紹介〉

이 상 원 (Sang-won Lee) 학생회원
2000년 2월 : 충남대학교 기계설계공학과 졸업
2001년 6월~현재 : 한국정보통신대학교 공학부 정보보호트랙 석사과정
<관심분야> 정보보호 및 응용, 네트워크 보안

천 정 희 (Jung Hee Cheon) 정회원
1997년 2월 : 한국과학기술원 수학과 박사
1997년 3월~2000년 1월 : 한국전자통신연구원 선임연구원
2000년 1월~2000년 12월 : Brown 대학 박사후 연구원
2000년 12월~2003년 2월 : 한국정보통신대학교 공학부 조교수
2003년 3월~현재 : 서울대학교 수리과학부 조교수
<관심분야> 응용정수론, 암호론, 응용암호론



김 용 대 (Yongdae Kim)
1991년 2월 : 연세대학교 수학과 학사
1993년 2월 : 연세대학교 수학과 석사
1993년 2월~1998년 6월 : 한국전자통신연구원 연구원
1998년 9월~2002년 5월 : University of Southern California, 박사
2001년 1월~2002년 7월 : UC Irvine 연구원
2002년 8월~현재 : University of Minnesota - Twin Cities, 조교수
<관심분야> 암호론, 네트워크 보안