동적 그룹을 위한 개선된 그룹 인증키 합의 프로토콜

허 성 철*, 김 광 조*
*한국정보통신대학교, 국제정보보호기술연구소

An Improved Group Authenticated Key Agreement Protocol for Dynamic Groups

Sungchul Heo* and Kwangjo Kim*
*International Research center for Information Security (IRIS),
Information and Communications University (ICU)

요 약

그룹키 합의 프로토콜은 개방된 네트워크 상에서 그룹의 모든 개체들이 참가하여 그 그룹만을 위한 키를 만드는 과정이다. 이 키를 이용하여 개체들은 그룹 간에 안전한 채널을 설정할 수 있다. 최근에 WISA 2004에서 Ren *et al.*은 효율적인 그룹키 합의 프로토콜을 동적 그룹에 대해서 제안하였고[1], 이 프로토콜은 안전한 두 개체간의 키 합의 프로토콜을 기반으로 만들어진다. WISA 2005에서 Nam *et al.*은 Ren *et al.*이 제안한 기법에서 키 확립 프로토콜의 결점을 발견하였다[2]. 그 결점은 공모한 2명의 공격자에 의한 적극적인 공격에 키 확립 프로토콜이 취약하다는 것이다. 이 논문에서는 Ren *et al.*이 제안한 기법과 비슷한 효율성을 지니면서 결점을 극복하는 개선된 그룹키 합의 프로토콜을 제안한다.

I. 서론

최근 몇 년 동안에, 많은 어플리케이션들이 P2P 그룹 통신을 사용해 왔다. 예를 들면, 원격 회의, 명령과 제어 시스템과 애드혹 네트워크 등에서 통신이 이에 해당한다. 이러한 환경에서 유비쿼터스와 믿을 수 있는 보안 서비스를 제공하는 것은 아주 중요하고, 중요한 이슈가 되고 있다. 안전한 그룹 통신을 위해서 기본적인 요구사항은 멤버들 간에 공통적인 그룹키가 사용가능해야 한다. 따라서, 키 관리가 중요한 관심사항이 된다. Pairing에 기반을 둔 프로토콜을 제외하고, 많은 그룹키 합의 프로토콜들은 두 개체간의 디피-헬만 키 교환 프로토콜을 이용한다 [1-3, 5-7]. 그룹키 합의 프로토콜은 키를 공유하는 개체들의 구성방식이 고정 또는 변동에 따라 정적 그룹과 동적 그룹으로 크게 나뉜다.

최근에, WISA 2004에서 Ren *et al.*은 동적 그룹에 대한 효율적인 그룹키 합의 프로토콜을 제안하였다 [1]. (이하 "RLKY04"라고 한다.) RLKY04는 적극적인 공격자에 안전한 두 개체간의 키 확립 프로토콜을 이용하여 구성되었고, 두개의 서브 프로토콜로 이루어져 있다: 키 확립(Key Establishment) 프로토콜 (RLKY04-KE)와 키 갱신(Key Update) 프로토콜 (RLKY04-KU). RLKY04-KE 프로토콜은 그룹키라 불리는 공통적인 비밀키를 그룹 간에 확립하는 프로토콜이고, RLKY04-KU 프로토콜은 그룹 안에서 효율적으로 동적 멤버쉽의 변화를 관리하는 프로토콜이다. WISA 2005에서 발표된 Nam *et al.*의 논문에서, 공모한 두 공격자에 의해서 적극적인 공격에 취약한 RLKY04-KE 프로토콜의 결점이 발견되었다. (이하 "NKW05"라고 한다.)

이 논문에서는 NKW05의 공격을 막는 개선된 RLKY04-KE 프로토콜 제안한다. RLKY04-KE 프로토콜과 비교했을 때, 라운드 수는 같고, 조금 더 많은

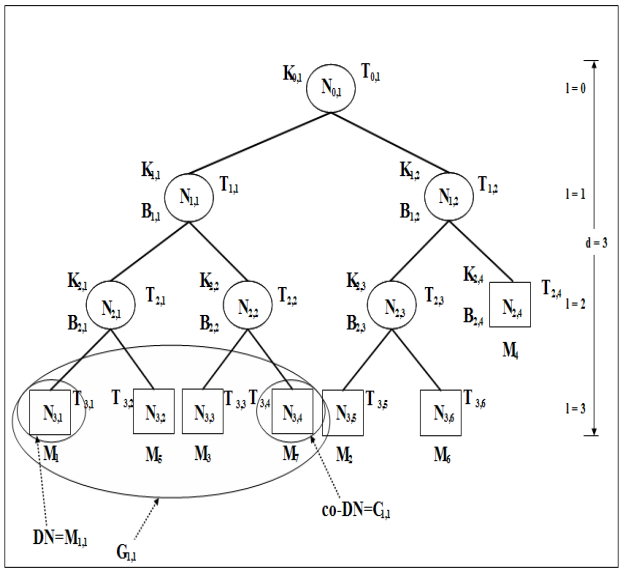
계산량을 보인다. 이것은 RLKY04-KE 프로토콜이 가지고 있는 효율성을 위반하지 않는다.

본 논문의 구성은 다음과 같다. 먼저 II장에서 키 트리 구조를 위한 표기법과 키 트리 구조에 대해서 소개한다. III장에서 RLKY04-KE 프로토콜과 NKW05의 공격기법을 설명한다. IV장에서 제안 프로토콜인 IGAKA-KE 를 제안하고, 안전성을 분석한 후, V장에서는 다른 프로토콜과 효율성을 비교한다. VI장에서 결론을 제시한다.

II. 기호의 정의

수학적인 기호를 위해서 $\mathbb{G}(=<\alpha>)$를 Z_p^*의 서브그룹인 위수 q를 갖는 순환 그룹이라고 한다. 여기서 p는 tq+1이고, t는 작은 자연수이다(예를 들어, t=2). 여기서 NKW05에서 사용한 기호를 따르고, 이 논문에서 제안한 기법을 위한 기호를 추가한다.

M_i: 개체를 나타내는 최하단 노드(leaf node)
$N_{l,r}$: 레벨 l에서 왼쪽에서 r번째 노드
$\widehat{N_{l,r}}$: $N_{l,r}$의 형제 노드(sibling node)
$T_{l,r}$: $N_{l,r}$이 루트인 서브트리
$\widehat{T_{l,r}}$: $T_{l,r}$의 형제 서브트리, 즉, $\widehat{N_{l,r}}$ 노드가 루트인 서브트리
$G_{l,r}$: $T_{l,r}$의 멤버들로 구성된 서브그룹
$K_{l,r}$: $G_{l,r}$의 비밀키(secret key)
$B_{l,r}$: $K_{l,r}$과 연관된 은닉키(blinded key)
Designated Negotiator(DN) : 서브트리에서 가장 왼쪽 최하단 노드
$M_{l,r}, \widehat{M_{l,r}}$: 개별적으로, $T_{l,r}, \widehat{T_{l,r}}$에서 DN
$k_{i,j}$: M_i와 파트너 또는 코파트너 M_j의 pairwise 키
co-DN : 서브트리에서 가장 깊고, 오른쪽에 위치한 최하단 노드
$C_{l,r}, \widehat{C_{l,r}}$: 개별적으로, $T_{l,r}, \widehat{T_{l,r}}$에서 co-DN
$\{\ \}_k$: 대칭키 k를 사용하여 메시지 암호
$[\ ]_K$: 공개키 K를 사용하여 메시지 암호
$h(\ )$: 일방향 해시 함수
r_i, s_i : M_i의 랜덤 수($r_i, s_i \in Z_q^*$)
S_i : M_i의 비밀 정보($S_i \in \alpha^{s_i} \bmod q$)
$S_{l,r}$: $G_{l,r}$의 비밀 정보($=h(S_{l+1,2r-1} \| S_{l+1,2r})$)
(x_i, α^{x_i}) : 순차적으로 M_i의 개인키와 공개키



(그림 1) 그룹 $G=\{M_1,...,M_7\}$에 대한 키 트리 구조

모든 노드는 두 노드의 부모이거나 최하단 노드이다. 키 트리의 높이가 d일 때, 루트는 레벨 0에 위치하고, 모든 최하단 노드들은 레벨 d 나 d-1에 있다. $G=\{M_1,...,M_n\}$를 그룹키를 만들기 위한 그룹이라고 하면, 그룹 멤버들은 트리의 최하단 노드에 정렬이 되고, 내부 노드들은 그룹 멤버가 아닌 논리적인 노드이다.

각 노드 $N_{l,r}$은(여기서, $l \neq d$ 이다.) 키 쌍($K_{l,r}$, $B_{l,r}$)에 연관된다. 비밀키 $K_{l,r}$는 서브그룹 $G_{l,r}$에 속한 멤버들만 가지고 있고, 루트 키 $K_{0,1}$는 그룹 G의 모든 멤버들에 의해 공유된다.

DN의 정의에 의해서 그룹의 특정한 멤버는 여러 서브 트리(높이 d까지)의 DN이 될 수 있다. 예를 들어, (그림 1)에서, M_1 는 서브트리 $T_{3,1}$, $T_{2,1}$ 와 $T_{1,1}$의 DN이고, 반면에, M_4는 단지 하나의 노드로 구성되는 $T_{2,4}$의 DN이다. 비슷하게, co-DN의 정의에 의해서, M_7은 서브트리 $T_{3,4}$, $T_{2,2}$와 $T_{1,1}$의 co-DN이고, 반면에, M_6은 $T_{3,6}$, $T_{2,3}$ 와 $T_{1,2}$의 co-DN이다. 여기서 DN인 $M_{l,r}$과 $\widehat{M_{l,r}}$은 함께 파트너 관계에 있다고 정의한다. 따라서, 서로 파트너이다. 또한, 비슷하게, 2개 co-DN인 $C_{l,r}$과 $\widehat{C_{l,r}}$은 함께 코-파트너

(co-partner) 관계에 있다고 정의한다. 서로 코-파트너이다. DN $M_{l,r}$과 co-DN $C_{l,r}$은 서브그룹 $G_{l,r}$에서 대표로 지명되고, 개별적으로 $\widehat{M_{l,r}}$ 과 $\widehat{C_{l,r}}$은 그의 파트너들(또는 코-파트너들)과 pairwise 키 $k_{i,j}$를 생성하는 임무를 가진다. S_i 는 M_i의 비밀 정보를 나타내고, $S_{l,r}$는 서브그룹 $G_{l,r}$의 비밀 정보를 나타내며, 서브그룹 $G_{l+1,2r-1}, G_{l+1,2r}$의 비밀 정보들로 만들어진다.

III. 기존 연구

1. RLKY04-KE 프로토콜[1]

여기서 프로토콜을 묘사할 때, 그룹 멤버들이 완전한 전방향 안전성과 키 안전성을 제공하는 두 개체 간의 인증된 키 합의 프로토콜을 사용한다고 가정한다. 이런 프로토콜 중에 Atenise가 제안한 A-DH[3]가 한 예이다. 또한, 모든 그룹 멤버들은 키 트리 구조와 그 트리에서 자신의 위치를 알고 있다고 가정한다. 임의적으로 선택된 한 멤버가 트리와 연관된 정보를 구성하고, 나머지 멤버들에게 트리 정보를 전송하므로 위의 가정을 성립할 수 있다. (그림 1)의 예에서 외관상으로 그룹 멤버들의 조직적인 정렬에도 불구하고, 키 트리에서 멤버들의 순서는 중요한 것이 아니다. 멤버들은 RLKY04에서 묘사된 것처럼 임의적으로 자리배치가 이루어진다. 여기서 중요한 것은 한 노드의 2개 서브트리의 높이 차이가 기껏해야 하나 차이가 나야한다는 것이다. 이것은 키 트리가 잘 균형잡히기 위함이다.

지금부터 RLKY04-KE 프로토콜을 상세히 묘사하겠다. 이 프로토콜은 넓게 2가지 측면으로 나뉜다. 하나는 pairwise 키 확립 단계와 다른 하나는 비밀키와 은닉키 확립 단계이다.

1.1 Phase I : Pairwise 키 확립 단계

이 단계에서는, 파트너인 $M_{l,r}$과 $\widehat{M_{l,r}}$이 두 개체 간의 키 합의 프로토콜을 수행하여 pairwise 키를 확립한다. 여기서 n개의 멤버로 구성된 그룹에서는 키 트리에서 n-1개 pairwise 키가 생성된다. 예를 들어, (그림 1)의 키 트리에서, 파트너인 DN 쌍이 6개 존재한다:(M_1,M_5), (M_3,M_7), (M_2,M_6), (M_1,M_3), (M_2, M_4) 과 (M_1,M_2). n-1개의 두 개체 간의 키 합의 프로토콜이 동시에 실행되기 때문에, 이 단계에서 실행되는 통신 라운드는 두 개체 간의 키 합의 프로토콜의 통신 라운드 수와 같다.

만약, A-DH[3]으로 예를 들자면, 이 과정은 다음과 같다.

Protocol RLKY04-KE Phase I :

$\{M_1,...,M_n\}$: 그룹 멤버들

P_i : M_i와 파트너들의 집합

Round 1:

$M_i, i\in[1,n]\rightarrow\{M_j|M_j\in P_i, j>i\}:\alpha^{r_i},\ r_i\in Z_q^*$

$M_i, i\in[1,n]\rightarrow\{M_j|M_j$는 M_i의 형제노드$, j>i\}:\alpha^{r_i}$

Round 2:

$M_i, i\in[1,n]\rightarrow\{M_j|M_j\in P_i, j<i\}:\alpha^{r_j f(\alpha^{r_i r_j})}$,

$r_j\in Z_q^*,\ f(x)=\begin{cases}x & ,\text{if } x\le q\\ p-x & ,otherwise\end{cases}$

$M_i, i\in[1,n]\rightarrow\{M_j|M_j$는 M_i의 형제노드$, j<i\}:\alpha^{r_j f(\alpha^{r_i r_j})}$

$\therefore\ k_{i,j}=\alpha^{r_i r_j}$.

이 pairwise 키들은 안전하게 두번째 단계에서 DN끼리 은닉키를 교환할 때, 이용된다. 여기서 n=2인 경우는 고려대상에서 제외한다. 왜냐하면 개체가 2개인 그룹의 그룹키는 첫번째 단계에서 두 개체 간의 생성된 pairwise 키 자체가 되기 때문이다.

1.2 Phase II : 비밀키와 은닉키 확립 단계

그룹 멤버들이 자신의 파트너들과 pairwise 키를 생성할 때, 노드들은 비밀키와 은닉키들을 bottom-up 방식으로 계산한다. 즉, 레벨 d-1에서 시작하여 레벨 0까지의 루트 노드까지 진행하면서 키를 생성한다. 은닉키는 항상 비밀키를 해시 함수에 적용시켜 계산된다. 즉, $B_{l,r}=h(K_{l,r})$이다. 많은 예외가 존재할지라도, 비밀키를 계산하기 위해서는 두 자식 노드들의 은닉키를 알아야 한다. 좀 더 구체적으로 설명하자면, 모든 $K_{l,r}$는 순환적으로 다음과 같이 계산된다:$K_{l,r}=h(B_{l+1,2r-1}\|B_{l+1,2r})$. (그림 1)에서 M_1과 M_2는 그룹키를 확립하기 위해서 여러 서브트리의 DN으로서 은닉키를 서브트리의 멤버에게 전달하는 역할도 맡으므로 다른 멤버들보다 많은 계산량이 필요하다. 즉, RLKY04-KE 프로토콜은 그룹키를 확립하기 위해 특정한 두 개체에게 다른 그룹 멤버들보다 좀더 많은 계산량을 요구한다.

이런 방법으로, 루트의 비밀키(그룹키)를 결정하기 위하여 그룹 멤버들에게 필요한 통신 라운드 수는 d

이다. i번째 라운드 끝에서는, 레벨 d−i에서 노드 $N_{l,r}$의 비밀키와 은닉키는 서브그룹 $G_{l,r}$에게 이용가능해야 한다. 각 라운드의 구체적인 설명은 다음과 같다.

Protocol RLKY04−KE Phase II :
$\{M_1,...,M_n\}$: 그룹 멤버들
Round 1 : l=d−1;
$K_{l,r}$:$N_{l,r}$의 두 자식 노드들이 공유하는 pairwise 키
각 $M_{l,r}\rightarrow\widehat{M_{l,r}}$: $\{B_{l,r}\|M_{l,r}\}_{k_{i,j}}$, $B_{l,r}=h(K_{l,r})$

Round i : l = d−i;
$M_{l+1,2r-1}$는 $M_{l+1,2r}$로부터 받은 암호문을 풀어 $B_{l+1,2r}$을 획득,
$M_{l+1,2r-1}\rightarrow G_{l+1,2r-1}$:$\{B_{l+1,2r}\|M_{l+1,2r-1}\}_{K_{l+1,2r-1}}$
모든 $M_i(\in G_{l+1,2r-1})$는 $B_{l+1,2r}$을 획득하고,
$K_{l,r}=h(B_{l+1,2r-1}\|B_{l+1,2r})$와 $B_{l,r}=h(K_{l,r})$을 계산
각 $M_{l,r}\rightarrow\widehat{M_{l,r}}$: $\{B_{l,r}\|M_{l,r}\}_{k_{i,j}}$.
동시에 $M_{l+1,2r}$도 위의 연산 수행.

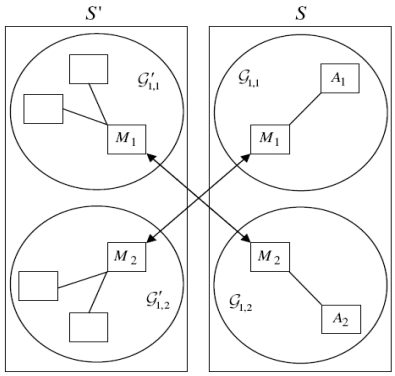
Round d:
$M_{1,1}$ $M_{1,2}$으로부터 받은 암호문을 풀어 $B_{1,2}$을 획득
$M_{1,1}\rightarrow G_{1,1}$: $\{B_{1,2}\|M_{1,1}\}_{K_{1,1}}$
모든 $M_i(\in G_{1,1})$는 $B_{1,2}$을 획득하고, 그룹키 계산:
$K_{0,1}=h(B_{1,1}\|B_{1,2})$
동시에 $M_{1,2}$ 위의 연산 수행.

2. NKW05의 공격 기법[2]

많은 애플리케이션에서 한 개체가 다른 개체들과 동시 발생하는 여러 세션들을 생성할 수 있다고 가정할 수 있다. 따라서, 메시지를 읽고, 수정하고, 삽입하고, 삭제하고, 지연시키고, 반복시킬 수 있는 적극적인 공격자 앞에서 한 프로토콜이 동시에 여러 번 실행할 수 있다. 부정직한 내부자의 도움 없이는 키를 알 수 없는 암시적 키 인증을 이루는 프로토콜을 인증된 키 확립 프로토콜이라고 한다. 그러나 Ren의 기법은 암시적 키 인증을 만족하지 못한다. 그 이유는 다음과 같다.

NKW05의 공격기법에서 공격자들은 다른 사용자들과 정상적으로 그룹키를 생성할 수 있는 합법적인 사용자들이다. RLKY04의 기법에서는 단지 은닉키와 함께 송신자의 ID가 암호화되어 전송된다. 이것은 공모한 A_1와 A_2에 의한 적극적인 공격에 취약성을 드러낸다. G'의 멤버들이 수행하는 프로토콜 세션 S'를 고려하자. 여기서 A_1와 A_2는 G'에 속하지 않는다. M_1와 M_2는 세션 S'에서 개별적으로 $M_{1,1}$과 $M_{1,2}$이 된다고 가정한다. 또한, M_1과 M_2는 동시에 형성되는 새로운 세션 S에서 개별적으로 $M_{1,1}$과 $M_{1,2}$라고 가정한다. 세션 S에서 그룹은 G이다. 즉, 공격을 하기 위해서 2개의 세션이 동시에 수행된다; 세션 S: $G=\{M_1,M_2,A_1,A_2\}$와 세션 S': $G'=\{M_1,M_2,...,M_n\}$. 간단히 설명하기 위해서, G'의 크기는 16개 이하로 가정한다. 왜냐하면, G'의 크기가 크면, d−1 라운드까지 G는 세션키를 만드는 과정을 지연시키는 시스템적 절차를 거쳐야 하기 때문이다.

공격의 마지막 단계에서는 G'의 모든 멤버들은 성공적으로 그룹키를 생성하여 세션을 성공적으로 끝냈다고 생각한다. 실제적으로 적극적인 공격자 A_1과 A_2도 그룹키를 사용할 수 있다. 이것은 NKW05의 공격 기법이 심각한 결과를 초래한다는 것을 의미한다. (그림 2)는 이러한 공격을 보여준다.



(그림 2) RLKY04−KE 프로토콜 공격

NKW05의 공격 기법을 구체적으로 설명하자면, 먼저 M_i^S과 $M_i^{S'}$을 각각 S과 S'에서 M_i의 인스턴스라고 한다. 공격자들의 전략은 2개의 세션이 어떤 메세지를 교환하는 것을 제외하고 나머지 부분에서는 프로토콜대로 수행하는 것으로 다음과 같다:

첫번째 pairwise 키 확립 단계에서, 공격자들은 M_1^S이 $M_2^{S'}$ 대신에 $M_2^{S'}$과 pairwise 키를 생성하도록 조작한다. 같은 방법으로 $M_1^{S'}$이 $M_2^{S'}$ 대신에 M_2^S와 pairwise 키를 생성하도록 조작한다. 이러한 메시지 교환 조작은 탐지되지 않을 수 있다.

두번째 비밀키와 은닉키 확립 단계에서는 마지막 라운드(d−1 라운드)에서 $M_1(M_{1,1})$과 $M_2(M_{1,2})$는 서로에게 pairwise 키를 이용하여 은닉키를 암호화하고, 전송한다. 그러나, 공격자들은 두 세션에서 보내

진 이 암호문들을 각각 다른 세션으로 전송한다. 즉, M_1^S, M_2^S, $M_1^{S'}$와 $M_2^{S'}$에 의해 보내진 암호문은 개별적으로 $M_2^{S'}$, $M_1^{S'}$, M_2^S와 M_1^S으로 전송된다.

송신자의 ID가 올바르게 전송되므로, M_1 과 M_2는 이러한 조작을 알아내기 힘들다. 따라서, M_1과 M_2는 프로토콜 실행을 성공적으로 진행시킨다. 구체적으로 설명하면, $K'_{l,r}$과 $B'_{l,r}$를 세션 S'에서 키 트리의 노드 $N_{l,r}$의 비밀키와 은닉키로 정의한다. 이 때 세션 S의 마지막 라운드에서, M_1^S은 $M_2^{S'}$로부터 받은 암호문을 풀어서 $B'_{1,2}$을 얻고, $G_{1,1}$의 멤버들에게 $\{B'_{1,2} \| M_1\}_{K_{1,1}}$을 보낸다. M_2^S는 $M_1^{S'}$으로부터 받은 암호문을 풀어서 $B'_{1,1}$을 얻고, $G_{1,2}$의 다른 멤버들에게 $\{B'_{1,1} \| M_2\}_{K_{1,2}}$을 전송한다. 비슷한 방법으로 세션 S'의 마지막 라운드에서, $M_1^{S'}$은 M_2^S로부터 받은 암호문을 풀어서 $B_{1,2}$을 얻고, $G'_{1,1}$의 나머지 멤버들에게 $\{B_{1,2} \| M_1\}_{K'_{1,1}}$을 전송한다. $M_2^{S'}$는 M_1^S으로부터 받은 암호문을 풀어서 $B_{1,1}$을 얻고, $G'_{1,2}$의 나머지 멤버들에게 $\{B_{1,1} \| M_2\}_{K'_{1,2}}$을 전송한다.

따라서, 두 세션의 마지막에서, $G_{1,1}$과 $G'_{1,2}$의 멤버들은 다음과 같은 세션키 $K_{fake1} = h(B_{1,1} \| B'_{1,2})$를 생성하고, $G'_{1,1}$과 $G_{1,2}$는 다음과 같은 세션키 $K_{fake2} = h(B'_{1,1} \| B_{1,2})$를 생성하게 된다.

NKW05의 공격기법을 통해서, RLKY04의 인증 메카니즘[1]이 완전히 무너지게 된다. 이 공격기법은 중간자 공격이라고 할 수 있다. G'의 모든 멤버들은 안전한 그룹키를 생성했다고 믿을 것이다. 실제적으로는 공격자들과 K_{fake1}와 K_{fake2}을 함께 공유하고 있는 것이다. 그 결과, 공격자들은 G'의 멤버들 사이에 기밀한 통신을 공격자들은 접근하고, 지연시킬 수도 있고, 위장하여 자신들에게 이로운 메시지를 전송할 수도 있다. 결과적으로, 동시에 Ren이 제안한 2개 프로토콜이 실행되고, 2개의 키 트리의 $M_{1,1}$과 $M_{1,2}$에 대한 멤버가 같다면, RLKY04-KE 프로토콜이 암시적 키 인증을 제공한다고 기대할 수 없다.

여기서 NKW05의 공격기법에는 한계점이 존재한다. 공격자들은 어떤 그룹의 그룹키를 획득하는 것이 아니라 2개의 위조 세션키를 생성하기 때문에, A_1은 단지 $G'_{1,2}$을 공격할 수 있고, A_2는 단지 $G'_{1,1}$을 공격할 수 있다. 또한, $G'_{1,2}$의 멤버들끼리만 통신이 가능하고, 또한, $G'_{1,1}$의 멤버들끼리만 통신이 가능하다. $M_i \in G'_{1,1}$와 $M_j \in G'_{1,2}$다른 키를 가지고 있어서 서로 통신이 불가능하기 때문에, NKW05의 공격 기법은 빠른 시간 내에 탐지될 수 있다.

IV. 제안 프로토콜

여기서 제안하는 기법은 RLKY04-KE 프로토콜을 수정하여 NKW05의 공격을 막는 개선된 그룹키 합의 프로토콜에서 키 확립 프로토콜이다. (이하 "IGAKA-KE"라고 한다.)

1. Phase I : Pairwise Key 확립 단계

IGAKA-KE 프로토콜은 RLKY04-KE 프로토콜의 모든 요구사항과 가정을 따른다. 부가적으로, 이 단계에서는 코-파트너인 $C_{l,r}$과 $\widehat{C_{l,r}}$은 두 개체 간의 키 합의 프로토콜을 통해서 pairwise 키를 생성한다. (그림 1)에서 6개의 co-DN 쌍이 존재한다: (M_1,M_5), (M_3,M_7), (M_2,M_6), (M_5,M_7), (M_4,M_6)과 (M_7,M_6). 어떤 쌍은 파트너의 쌍과 겹친다. 즉, (M_1,M_5), (M_3,M_7), (M_4,M_6)과 (M_2,M_6). 여기서 M_4는 DN과 동시에 co-DN이 된다. 즉, 형제가 없는 최하단 노드는 DN과 co-DN이 동시에 될 수 있다. 따라서, 여기서 n개의 멤버로 구성된 그룹에 대한 키 트리에서 파트너 쌍과 코-파트너 쌍이 많아야 2(n-1)-n/2를 넘지 않는다 것을 알 수 있다. 만약 그룹키 생성을 위한 키 트리가 포화 이진트리라면, 2(n-1)-n/2의 쌍이 존재할 수 있다. 따라서, 많아야 2(n-1)-n/2번 프로토콜이 동시에 실행된다. IGAKA-KE 프로토콜은 RLKY04-KE 프로토콜과 다음과 같이 비슷하다:

Protocol IGAKA-KE Phase I :

$\{M_1,...,M_n\}$: 그룹 멤버들

P_i : M_i와 파트너들의 집합

CP_j : M_j의 코-파트너들의 집합

Round 1 :

$M_i, i \in [1,n] \rightarrow \{M_j | M_j$는 M_i의 형제 노드, $j > i\} : [\alpha^r \| S_i]_{\alpha^{s_j}}$

$M_i, i \in [1,n] \rightarrow \{M_j | M_j \in P_i, j > i\} : [\alpha^{r_i} \| S_i]_{\alpha^{x_j}}$,

$r_i, s_i \in Z_q^*$, $S_i = \alpha^{s_i}$

$M_k, k \in [1,n] \rightarrow \{M_l | M_l \in CP_k, l > k\} : [\alpha^{r_k} \| S_k]_{\alpha^{x_l}}$,

$r_k, s_k \in Z_q^*,\ S_k = \alpha^{s_k}$

Round 2 :

$M_j, j \in [1,n] \rightarrow \{M_i | M_j \in P_i, i<j\} : [\alpha^{r_j f(\alpha^{x_{ji}})} \| S_j]_{\alpha^{x_i}}$

$, r_j, s_j \in Z_q^*,\ S_j = \alpha^{s_j},\ f(x) = \begin{cases} x & , \text{if } x \le q \\ p-x & , otherwise \end{cases}$

$M_j, j \in [1,n] \rightarrow \{M_i | M_j$ 는 M_i의 형제 노드, $i<j\}$

$: [\alpha^{r_j f(\alpha^{x_{ji}})} \| S_j]_{\alpha^{x_i}}$

$M_l, l \in [1,n] \rightarrow \{M_k | M_l \in CP_k, k<l\} : [\alpha^{r_l f(\alpha^{x_{lk}})} \| S_l]_{\alpha^{x_k}}$

$, r_l, s_l \in Z_q^*,\ S_l = \alpha^{s_l}$

$\therefore k_{i,j} = \alpha^{r_i r_j}.$

개체의 비밀 정보를 나타내는 임의의 nonce는 형제 노드와 파트너 또는 코-파트너의 공개키에 의해서 암호화되고, 상대 개체에게 보내진다. M_i을 사용하는 대신에, 여기서는 S_i을 사용한다. 이것은 서브그룹 비밀 정보를 생성할 때 사용되고, 인증을 강화하기 위해서 각 개체는 서브그룹의 모든 멤버들의 비밀 정보들을 가져야 한다는 것을 의미한다.

2. Phase II : **비밀키와 은닉키 확립 단계**

IGAKA-KE 프로토콜에서 비밀키와 은닉키를 확립하는 방법은 RLKY04-KE 프로토콜과 흡사하다. 그룹의 모든 멤버들이 그룹키를 결정하는데 필요한 라운드 수는 트리의 높이(d)이다. 이 논문에서는 인증을 강화하고, NKW05의 공격을 막기 위해서 서브그룹 비밀 정보, 즉, $G_{l,r}$의 $S_{l,r}$(= $h(S_{l+1,2r-1} \| S_{l+1,2r})$)을 새롭게 정의한다. RLKY04-KE 프로토콜에서는 특정한 두 개체에게 그룹 멤버들보다 많은 계산량을 요구하는데 IGAKA-KE 프로토콜에서는 특정한 두 개체뿐만 아니라 서브그룹 비밀 정보를 계산하기 위한 특정한 두 개체에게도 다른 그룹 멤버들보다 많은 계산량을 요구한다. 각 라운드에 대한 구체적인 묘사는 다음과 같다:

Protocol IGAKA-KE Phase II :

$\{M_1, ..., M_n\}$: 그룹 멤버들

Round 1 : l=d-1;

$K_{l,r}$는 $N_{l,r}$의 두 자식 노드들이 공유하는 pairwise 키.

각 $M_{l,r} \rightarrow \widehat{M_{l,r}} : \{B_{l,r} \| S_{l,r}\}_{k_{ij}}$,

$S_{l,r} = h(S_{l+1,2r-1} \| S_{l+1,2r})$

각 $C_{l,r} \rightarrow \widehat{C_{l,r}}$: $\{G_{l,r}$의 멤버들의 비밀 정보 집합$\}_{k_{kl}}$

Round i : l = d-i;

$M_{l+1,2r-1}$는 $M_{l+1,2r}$로부터 받은 암호문을 풀어 $B_{l+1,2r}$을 획득.

$M_{l+1,2r-1} \rightarrow G_{l+1,2r-1}$: $\{B_{l+1,2r} \| S_{l+1,2r-1}\}_{K_{l+1,2r-1}}$

$C_{l+1,2r-1}$는 $C_{l+1,2r}$로부터 받은 암호문을 풀어 $G_{l+1,2r}$의 멤버들의 비밀 정보들을 획득.

$C_{l+1,2r-1} \rightarrow G_{l+1,2r-1}$:

$\{G_{l+1,2r}$의 멤버들의 비밀 정보 집합$\}_{K_{l+1,2r-1}}$

h($G_{l+1,2r}$의 멤버들의 비밀 정보들)이 $S_{l+1,2r}$과 같으면, $B_{l+1,2r}$ 이 유효, 그렇지 않으면, 다시 시작.

모든 $M_i (\in G_{l+1,2r-1})$는 $B_{l+1,2r}$을 획득하고,

$K_{l,r} = h(B_{l+1,2r-1} \| B_{l+1,2r})$ 와 $B_{l,r} = h(K_{l,r})$을 계산.

$M_{l,r} \rightarrow \widehat{M_{l,r}}$: $\{B_{l,r} \| S_{l,r}\}_{k_{ij}}$,

$C_{l,r} \rightarrow \widehat{C_{l,r}}$: $\{G_{l,r}$의 멤버들의 비밀 정보 집합$\}_{x_{kl}}$

동시에 $M_{l+1,2r}$과 $C_{l+1,2r}$는 위의 연산 실행

Round d:

$M_{1,1}$은 $M_{1,2}$로부터 받은 암호문을 풀어 $B_{1,2}$를 획득.

$M_{1,1} \rightarrow G_{1,1}$: $\{B_{1,2} \| S_{1,1}\}_{K_{1,1}}$

$C_{1,1} \rightarrow G_{1,1}$: $\{G_{1,1}$의 멤버들의 비밀 정보 집합$\}_{K_{1,1}}$

만약 h($G_{1,2}$의 멤버들의 비밀 정보들)이 $S_{1,2}$와 같으면, $B_{1,2}$이 유효, 그렇지 않으면, 다시 시작.

모든 $M_i (\in G_{1,1})$는 $B_{1,2}$을 획득하고, 그룹키 생성:

$K_{0,1} = h(B_{1,1} \| B_{1,2})$

동시에 $M_{1,2}$과 $C_{1,2}$ 위 연산을 실행.

서브그룹 비밀 정보는 $G_{l,r}$에서 받은 비밀 정보들의 해시 값과 비교하기 때문에, 서브그룹 비밀 정보이 유효한지 아닌지 확인할 수 있다. 따라서 NKW05의 공격기법과 같은 공격은 쉽게 막을 수 있다. 이 공격이 탐지된다면, 프로토콜은 중단하고, 키 트리를 새롭게 만들어 다시 시작한다. NKW05의 공격 기법으로 이 논문에서 제안한 프로토콜을 공격하기 위해서는 그룹 크기만큼의 공격자들이 있어야 한다. 이것은 현실적으로 불가능하다는 것을 의미한다.

세션키와 마찬가지로 서브그룹 비밀 정보를 그룹의 pairwise 키와 비밀키를 통해서 전송을 하기 때문에 RLKY04에서 언급한 것처럼 안전하다.

V. 복잡도 분석 및 비교

이 절에서는 IGAKA-KE 프로토콜의 복잡도를 명확한 비교를 위해서 두 개체간의 키 합의 프로토콜을 A-DH[3]로 이용해서 분석할 것이다. [표 1]

은 알려진 많은 프로토콜과 IGAKA-KE 프로토콜을 비교하여 보여준다. RLKY04-KE 프로토콜과 Yang *et al.*에 의해서 제안된 프로토콜이 가장 좋은 성능을 보인다. 단지 5n-4 지수승과 d+2 라운드 수가 RLKY04-KE 프로토콜에 필요하다. 그러나 IGAKA-KE 프로토콜은 pairwise 키 생성단계에서 부가적으로 RLKY04-KE 프로토콜보다 $n-1-\lfloor n/2 \rfloor$ 지수승을 더 계산하기 때문에 IGAKA-KE 프로토콜은 $5n-4+(n-1-\lfloor n/2 \rfloor)$의 지수승을 요구한다. 전체 메시지 측면에서 보면, IGAKA-KE 프로토콜은 RLKY04-KE 프로토콜보다 많은 메시지 교환이 발생한다. 파트너의 수와 비슷한 코-파트너들끼리의 메시지 교환이 이루어지기 때문에 RLKY04-KE 프로토콜보다 대략적으로 2배 많다. 그러나, 지수승이 많아서 계산량이 많고, 메시지 수가 많다고 하더라도 서브그룹 인증이 가능하고, 동일한 라운드 수로 그룹키를 생성하므로 IGAKA-KE 프로토콜은 RLKY04-KE 프로토콜만큼 효율적이라 할 수 있다. Yang *et al.*이 제안한 프로토콜[4]은 ID 기반을 둔 두 개체간의 키 합의 프로토콜을 이용하기 때문에 각 라운드 실행하면서 4개의 지수승이 필요하므로 가장 적은 지수승이 필요하다는 사실을 알 수 있다. 이 프로토콜은 주로 동적 그룹이 아닌 정적 그룹에 적합하다. 한편, Bresson이 제안한 프로토콜[5]은 소극적인 공격과 적극적인 공격에 안전하지만, 계산량이 $(n^2+4n)/2-1$로 다른 프로토콜보다 너무 많다. 마찬가지로 Burmester가 제안한 프로토콜[6]도 라운드 수는 2이지만 계산량이 많다.

부가적으로, 각 개체는 IGAKA-KE 프로토콜 실행이 끝나면, 그룹의 모든 비밀 정보들을 소유하게 되는데, 이것은 그룹의 크기가 n이라고 했을 때, RLKY04-KE 프로토콜보다 메모리 사용량이 $O(n)$만큼 많아진다.

[표 1] 키 생성 프로토콜 비교

방식	라운드	총 메시지	총 지수승
IGAKA-KE	d+2	2*2(n-2)	$5n-4+(n-1-\lfloor n/2 \rfloor)$
RLKY04-KE	d+2	2(n-2)	5n-4
A-DH	n	n	$(n^2+4n)/2-1$
Yang *et al.*	d+1	2(n-2)	4n-4
Bresson *et al.*	n	n	$(n^2+4n)/2-1$
Burmester *et.*	2	2n	n(n+1)

VI. 결론

이 논문에서는 RLKY04-KE 프로토콜을 수정하여 NKW05의 공격에 안전한 IGAKA-KE 프로토콜을 제안하였다. 그룹키가 확립되는 동안에, 서브 그룹 비밀 정보를 이용하여 서브그룹에 대한 인증도 동시에 수행되므로, 똑같은 라운드 수로 그룹키를 생성하고, 적극적인 공격자에 의한 공격의 한 종류인 NKW05의 공격을 방어할 수 있었다. 이 논문에서는 RLKY04의 그룹키 합의 프로토콜에서 키 확립 프로토콜만 언급하였다. 나머지 프로토콜인 Join 과 Leave 프로토콜도 약간의 수정을 통해서 RLKY04의 기법과 동일한 효율성을 가지는 프로토콜을 개선할 수 있고, 그룹키 합의 프로토콜에서 중요한 프로토콜인 Merge와 Partition 프로토콜을 추가하는 것이 향후 과제중 하나이다.

[참고문헌]

[1] Kui Ren, Hyunrok Lee, Kwangjo Kim and Taewhan Yoo, " Efficient Group Authenticated Key Agreement Protocol for Dynamic Groups", Proc. of WISA 2004, LNCS, Vol. 3325, pp.233-247, 2005.

[2] Junghyun Nam, Seungjoo Kim and Dongho Won, " Security Weakness in Ren et al.'s Group Key Agreement Scheme Built on Secure Two-Party Protcols", Proc. of WISA 2005, LNCS, Vol. 3786, pp.1-9, 2006

[3] G. Ateniese, M. Steiner and G. Tsudik, "New Multi party Authentication Services and Key Agreement Protocols", IEEE JSAC on Secure Communication, Vol. 18, No. 4, pp.628-639, 2000.

[4] W. Yang, and S. Shieh, "Secure Key Agreement for Group Communications", ACM/PH International Journal of Network Management, Vol. 11, No. 6, pp.365-374, 2001.

[5] E. Bresson, O. Chevassut and D. Pointcheval, "Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions", Proc. of EUROCRYPT 2002, LNCS, Vol. 2332, pp.

321-336, 2002.

[6] M. Burmester and Yvo Desmedt, "Towards practical 'proven secure' authenticated key distribution", Proc. of ACM-CCS'93, pp. 228-231, Fairfax, Virginia, ACM Press, 1993.

[7] Y. Kim, A. Perrig, and G. Tsudik, "Tree-based group key agreement," ACM Trans. on Information and System Security, Vol. 7, No. 1, pp. 60~96, 2004.