일정한 수의 라운드를 가지는 그룹 키 합의 프로토콜의 조사 분석

박혜원*, 김광조*

*한국정보통신대학교

An Analysis of Constant-Round Group Key Agreement Protocols

Hyewon Park*, Kwangjo Kim*

*Information and Communications University

요 약

최근 인터넷 회의 또는 채팅 시스템에서는 안전하고 신뢰성 있는 통신을 위한 그룹 키 합의 (GKA) 프로토콜이 중요시 되고 있다. 그룹의 구성원들은 GKA 프로토콜을 이용하여 하나의 비밀 키를 공유하고, 이 키를 이용하여 상호간에 전송되는 메시지를 암호화 또는 복호화 한다. 하지만 많은 기존의 프로토콜들은 통신시간이 그룹 구성원의 수에 비례하여 그 수가 많아질수록 그룹 키를 합의하는데 많은 통신 시간을 소요해야 한다는 문제점이 있었다. 이 문제를 해결하기 위해, 구성원의 수에 상관없이 일정 라운드를 가지는 효율적인 GKA 프로토콜이 제안되었다. 이 논문에서는 현재까지 제안된 일정 라운드 그룹 키 합의 프로토콜들을 조사 및 분석하고, 앞으로의 연구 방향을 모색한다.

ABSTRACT

In recent internet conference systems, group key agreement (GKA) protocol is one of the important protocols for secure and reliable communication. Group Participants share one secret key using a GKA protocol to communicate with encrypted messages. However, previous protocols have much time to share a group key because communication time depends on the number of participants. To deal with this problem, GKA with constant communicating rounds have been proposed. With constant-round GKA protocol, participants can share a group key in constant time which does not depend on the number of participants. In this report, we review and analyze recently proposed GKA protocols which has constant communicating rounds.

Ⅰ. 서 론

인터넷을 통해 이루어지는 다자간 원격 회의 시스템이나 채팅 등의 그룹 통신은 허가된 사용자가 아닌 악성

사용자들도 쉽게 통신을 도청하거나 방해할 수 있기 때문에 외부로부터의 많은 위협에 노출되어 있다. 이러한 시스템들은 신뢰성 있는 통신환경을 제공하기 위해 지속적으로 연구하고 있으며, 사용자들 또한 그들의 프라이버시를 위해 안전한 통신을 필요로 한다. 그룹 키 합의 (Group Key Agreement, GKA)는 사용자들의 안전하고 신뢰성 있는 통신을 위한 프로토콜로. 사용자들이 통신을 시작하기 전에 비밀 키를 미리 공유하는 과정을 말한다. M. Manulis는 GKA에 대해 두 명 또는 그 이상의 사용자들이 각자의 정보를 교환하고 결합하여 그룹 키를 확립하며, 어떤 사용자도 그 결과를 미리 예측할 수 없게 하는 프로토콜 또는 메커니즘이라 정의하였다 [12]. 그룹 구성원들은 공유된 비밀 키를 사용해서 통신 메시지를 암호화 또는 복호화 할 수 있다. GKA와 함께 상호간의 그룹 키에 대한 인증을 제공하는 프로토콜을 인증된 그룹 키 합의 (Authenticated Group Key Agreement, AGKA) 라고 하며, 이는 현재의 그룹 통신 시스템에 있어 필수적인 요소이다.

GKA 프로토콜에 있어 가장 중요한 두 가지 특성은 효율성과 안전성이다. 그룹 구성원들이 그룹 키를 합의하는 과정에서의 시간과 계산 비용은 적어야 하고, 악성 사용자는 그들의 통신을 방해하거나 도청을 통해 얻은 정보만으로 그룹 키를 계산할 수 없어야 한다. 효율적인 GKA를 위한 방법 중 하나는 구성원의 수에 상관없이 일정 횟수만 통신을 하게 하는 것이다. 트리 구조를 기반으로 하던 많은 기존의 프로토콜들에서는 정보를 주고받기 위한 통신 횟수가 그룹 구성원의 수에 비례하였기 때문에 구성원의 수가 많은 경우에는 비효율적이라는 문제점이 있었다. 그에 반해 일정 라운드 (Constant-round) GKA 프로토콜은 항상 일정한 횟수의 통신을 하기 때문에 구성원의 수가 많은 경우에 비용을 감소시킬 수 있다.

이 논문에서는 최근까지 제안된 일정 라운드 GKA 프로토콜들에 대해 알아보고 성능과 안전성에 대해 분석하도록 한다. Ⅱ에서는 GKA의 안전성에 대한 요구 사항과 프로토콜에 적용될 수 있는 두 종류의 암호 시스템에 대해 간략히 정리하고, Ⅲ에서는 여러 일정 라운드 GKA 프로토콜에 대해서 조사한 뒤 성능과 취약점에 대해서 분석한다. Ⅳ에서는 Ⅲ에서 조사한 프로토콜들의 안전성과 성능을 통합적으로 비교 분석하고, Ⅴ에서는 결론과 앞으로의 연구 방향을 제시한다.

Ⅱ. 기초 지식

2.1. GKA 프로토콜의 안전성에 대한 요구사항

GKA 프로토콜의 안전성은 프로토콜을 방해하는 공격자(Adversary)의 종류에 따라 정의될 수 있다. 수동 공격자(Passive adversary)는 통신 메시지를 변경 시키거나 프로토콜의 진행을 방해하지 않고 도청을 통해 정보를 얻는 공격자이고, 능동 공격자(Active adversary)는 통신 메시지를 변경, 삽입, 또는 삭제 하는 등 프로토콜의 진행을 방해할 수 있는 공격자이다. 악성 구성원 (Malicious participant)은 능동 공격자의 한 종류로, 인증된 그룹 구성원이지만 통신을 방해하거나 다른 구성원을 위장 (Impersonation), 또는 프로토콜 진행을 따르지 않는 그룹 내부 공격자를 말한다. 이전의 많은 논문들에서 안전한 GKA 프로토콜을 위한 요구사항들을 각각 다른 방식으로 정리한 바 있다. 여기에서는 공격자의 종류에 따른 GKA 프로토콜의 요구사항들과 그에 대한 설명을 나열하였다.

1) 수동 공격자

- 키 비밀성 (Key Privacy) : 그룹 구성원만이 세션 그룹 키를 계산할 수 있으며, 공격자는 도청된 정보만으로 키를 계산할 수 없다.
- 분별 불가성 (Indistinguishability) : 공격자는 세션 그룹 키와 무작위로 나열된 문자열을 분별할 수 없다.
- 최근성 (Freshness) : 세션 그룹 키는 최근에 생성된 것이어야 하고, 재사용 될 수 없다.

2) 능동 공격자

- 전방향/역방향 안전성 (Forward/Backward Secrecy, F/B Secrecy) : 세션 그룹 키들이 부분적으로 노출되더라도 공격자는 차후의/이전의 세션 그룹 키의 안전성에 영향을 미칠 수 없다.
- 완전 전방향 안전성 (Perfect Forward Secrecy, PFS) : 사용자의 비밀 키가 노출되더라도 공격자는 이전 세션 그룹 키의 안전성에 영향을 미칠 수 없다.
- 키 무결성 (Key Integrity) : 확립된 그룹 키는 인증된 구성원으로부터의 정보로만 생성되며, 공격자에 의해서 변경될 수 없다.
- 키 확인 (Key Confirmation) : 모든 그룹 구성원은 다른 구성원이 실제로 특정 비밀 키를 소유하고 있다는 것을 증명하여야 한다.
- 사용자 인증 (Entity Authentication) : 모든 그룹 구성원의 신원은 인증되어야 한다.
- 기여도 (Contributory) : 각 그룹 구성원들은 키 합의 과정에 대한 다른 구성원의 기여도에 대해 확증할 수 있어야 한다.

2.2. 암호 시스템의 종류

3장에서 조사하는 GKA 프로토콜들은 서로 다른 암호 시스템을 기반으로 하고 있다. 이 장에서는 신원 기반 암호 시스템 (ID-based Cryptosystem)과 비밀번호 기반 암호 시스템 (Password-based Cryptosystem)에 대해서 간략히 설명하고자 한다.

1) 신원 기반 암호 시스템 (ID-based Cryptosystem)

신원 기반 암호 시스템에서는 공개키를 제 3자로부터 발급 받는 통상적인 공개키 암호 시스템과 달리 이미 알려진 사용자 자신의 신원 (ID나 메일 주소 등)을 공개키로 사용한다. 이는 따로 공개키에 대한 인증을 할 필요가 없기 때문에 훨씬 효율적이다. 현재 제안된 많은 프로토콜들이 Boneh와 Franklin의 논문에서 제안된 신원 기반 암호 시스템의 기본 구성을 따른다.[3] 여기에서 키 생성 센터 (Key Generation Center, KGC)는 시스템 변수를 먼저 생성한다.

$$param =< G_1, G_2, q, e, P_{pub}, H > \quad (1)$$

(1)에서 G_1은 덧셈 그룹, G_2은 곱셈 그룹, e는 겹선형 Pairing 이고 H는 해쉬함수 이다. KGC의 비밀 키가 s라고 할 경우 사용자의 공개/비밀 키 쌍은 다음과 같다.

$$< Q_i, S_i > = < H(ID), sH(ID) > \quad (2)$$

2) 비밀번호 기반 암호 시스템 (Password-based Cryptosystem)

비밀번호 기반 암호 시스템에서는 그룹 구성원들이 낮은 엔트로피를 가지는 비밀번호 pw를 가지고, 이를 이용해서 그룹 키 합의 과정을 거쳐 높은 엔트로피를 가지는 세션 그룹 키를 계산하게 된다. 여기에서 pw는 곧 그룹 구성원임을 인증하는 증명서가 된다.

III. 일정 라운드 GKA 프로토콜의 조사 분석

이 장에서는 최근에 제안된 일정 라운드 GKA 프로토콜들의 성능과 취약점에 대해서 분석하도록 한다.

3.1. Burmester와 Desmedt

1994년에 Burmester와 Desmedt은 디피-헬만(Diffie-Hellman)의 키 교환 프로토콜에 기반 하여 그룹 구성원들이 하나의 세션 그룹 키를 공유할 수 있게 하는 GKA 프로토콜을 제안하였다 [1,2] (이하 "BD94"라고 한다.) Star, Tree, Broadcast, Cyclic 등 4 종류의 네트워크 형태에 따른 각각의 프로토콜이 명시되어 있지만 Star 네트워크를 이용한 프로토콜의 경우에는 그룹 키를 한 서버에서 생성하여 나누어주는 그룹 키 분배 형식이 적용되고 Tree 네트워크를 이용한 프로토콜의 경우 통신 라운드의 수가 일정하지 않기 때문에, Broadcast와 Cyclic 네트워크를 사용한 프로토콜만이 GKA 프로토콜로 간주된다. [표 1]은 U_1에서 U_n까지 n명의 구성원이 그룹 키를 합의하는 broadcast 네트워크 기반의 GKA 프로토콜로, 각 구성원은 자신의 메시지를 방송(broadcast) 함으로써, 한 번의 메시지 전송으로 다른 모든 구성원들에게 메시지를 수신하도록 할 수 있다. 프로토콜의 진행이 끝난 후 각 구성원은 하나의 비밀 키 $K = K_i = \alpha^{r_1 r_2 + r_2 r_3 + \ldots + r_n r_1}$ 를 가지게 된다. Cyclic 네트워크를 기반으로 한 프로토콜 역시 [표 1]의 프로토콜과 같은 진행방식을 가진다. BD94 프로토콜과 함께, 각 구성원 U_i가 z_i를 다음 구성원 U_{i+1} 에게 순차적으로 인증하게 하고, 인증이 실패할 경우 프로토콜을 중지시키는 인증 알고리즘 또한 논문에 함께 제안되어 있다.

Burmester와 Desmedt은 공개된 정보를 도청하는 수동 공격자에 대해 Computational Diffie-Hellman (CDH) 문제를 기반으로 하여 안전성이 증명하였다. 하지만, 능동 공격자에 대한 안전성에 대해서는 언급하고 있지 않으며, 인증 알고리즘이 제시되어 있지만 이는 지수 z_i의 정확성에 대해서만 인증할 뿐 구성원의 신원 자체를 식별하지는 않기 때문에 악성 사용자의 신원 위장 공격 (Impersonation Attack)을 막을 수 없다. 그럼에도 불구하고 이 프로토콜은 방송 채널 (Broadcast Channel)을 이용하여 매우 효율적이므로, 이후의 많은 GKA 프로토콜에서 응용된다.

[표 1] BD94 프로토콜

Round 1. U_i selects random $r_i \in Z_q$ and broadcasts $z_i = \alpha_i^r \bmod p$.
Round 2. U_i broadcasts $X_i = (z_{i+1}/z_{i-1})^{r_i} (\bmod p)$
Key Computation. U_i computes the key $K_i = z_{i-1}^{nr_i} X_i^{n-1} X_{i+1}^{n-1} \cdots X_{i-2} \pmod p$

3.2. Katz와 Yung

2003년에 Katz와 Yung은 BD94 프로토콜을 발전시킨 GKA 프로토콜을 제시하였다 [4] (이하 "KY03" 이라고 한다.) KY03에서는 BD94의 안전성 증명이 짝수의 구성원에 대해서만 적용되기 때문에, 홀수의 구성원에 대해서 새로운 안전성 증명이 필요하다는 것과, BD94에서는 구성원들이 방송 채널을 사용한다고 명시되어 있지만 실제로는 단대단 통신을 한다는 것을 강조하였다. 프로토콜의 진행은 BD94 프로토콜의 진행과 같기 때문에 본 논문에서는 명시하지 않았다. 이에 더해서, 전자 서명을 사용하여 GKA 프로토콜에 인증 알고리즘을 제공하도록 하는 컴파일러와 이에 대한 안전성의 증명 또한 KY03에 제시되어 있다. 하지만 이 프로토콜은 상호 인증 (Mutual Authentication)을 제공하지 않기 때문에 악성 구성원으로부터의 신원 위장 공격에 취약하다. 또한, BD94 프로토콜과 마찬가지로 키 확인(Key Comfirmation) 과정이 없어 이로 인한 공격 또한 발생할 수 있다.

3.3. Choi, Hwang과 Lee

Choi, Hwang과 Lee 역시 BD94 프로토콜에 기반 하여 두 번의 통신 라운드를 가지는 신원 기반 그룹 키 인증 및 합의 프로토콜을 제시하였다 [6] (이하 "CHL04"라고 한다.) 이 프로토콜은 2.2장에서 명시한 신원 기반 암호시스템의 기본 셋업을 사용하므로 구성원의 공개 키와 비밀 키는 수식 (2)와 같다. 프로토콜의 진행은 [표 2]에 나타나 있다.

키 합의 과정을 통해 각 구성원은 하나의 비밀 키 $K = K_i = e(P,P)^{a_1 a_2 a_3 + \cdots + a_{n-1} a_n a_1 + a_n a_1 a_2}$ 를 가지게 된다. 여기에서는 기존의 디피-헬만 문제를 겁선형 필드로 확장 시킨 Decisional Hash Bilinear Diffie-Hellman (DHBDH) 문제를 기반으로 하여 프로토콜의 안전성을 증명하였다. 하지만 이 프로토콜의 두 번째 라운드에서 수행되는 $e(\sum_{k \in -1,1,2} T_k, P) = e(\sum_{k \in -1,1,2} (P_k + h_k Q_k), P_{pub})$에서 각 구성원 U_i는 세 명의 이웃 구성원 U_{i-1}, U_{i+1}, U_{i+2}의 $< P, T >$ 쌍에 대해서만 부분적으로 인증하기 때문에 그 외에 다른 구성원의 신원을 증명할 수 없다는 문제점이 있다.

Zhang과 Chen은 두 명의 악성 사용자가 인증된 그룹 구성원인 U_i의 이전의 인증 사본(Authentication Transcript)을 재전송 하여 U_i에 대한 신원 위장 공격을 할 수 있다는 것을 증명하였다. [5]

[표 2] CHL04 프로토콜

Round 1. U_i selects $a_i \in Z_q^*$, then broadcasts
$< P_i = a_i P,\ T_i = a_i P_{pub} + h_i S_i >$
where $h_i = H(P_i)$.
Round 2. After receiving $< P_i, T_i >$, U_i verifies
$e(\sum_{k \in -1,1,2} T_k, P) = e(\sum_{k \in -1,1,2} (P_k + h_k Q_k), P_{pub})$
If fails, then the protocol halts.
Otherwise, U_i broadcasts
$D_i = e(a_i(P_{i+2} - P_{i-1}), P_{i+1})$.
Key Computation. U_i computes the session key
$K_i = e(a_i P_{i-1}, P_{i+1})^n D_i^{n-1} D_{i+1}^{n-1} \ldots D_{i-2}$.

[표 3] CHL08 프로토콜

Setup. $IID = ID_1 \| \ldots \| ID_n$
Round 1. U_i selects $a_i \in Z_q^*$, then broadcasts
$< P_i = a_i P,\ T_i = a_i P_{pub} + h_i S_i >$
where $h_i = H(P_i \| IID)$.
Round 2. After receiving $< P_i, T_i >$, U_i verifies
$e(\sum_{k \in -1,1,2} T_k, P) = e(\sum_{k \in -1,1,2} (P_k + h_k Q_k), P_{pub})$.
If fails, then the protocol halts.
Otherwise, U_i computes
$D_i = e(a_i(P_{i+2} - P_{i-1}), P_{i+1})$,
makes signature pair $< W_i, V_i >$ on a message
$D_i \| SID \| IID$ where $SID = P_1 \| \ldots \| P_n$,
and broadcasts $ID_i \| < D_i, (W_i, V_i) >$.
Key Computation. If all $< W_i, V_i >$ is verified,
U_i computes the session key
$K_i = e(a_i P_{i-1}, P_{i+1})^n D_i^{n-1} D_{i+1}^{n-1} \ldots D_{i-2}$.

그들은 이 문제를 해결하기 위해 각 메시지에 시간 변수(Time Parameter)를 포함시켜 재전송 공격(Replay Attack)을 막도록 하였다. 하지만, 2007년에 Shim은 세 명의 악성 사용자 U_{i-2}, U_{i-1}, U_{i+1} 가 이전의 인증 사본 없이도 서로 협상하여 U_i를 위장할 수 있다는 것을 증명하면서, 이 공격을 막기 위해서는 모든 메시지에 서명을 첨부하여 각 구성원이 다른 모든 구성원들을 인증하도록 해야 한다고 제안하였다. [14]

2008년에 Choi, Hwang과 Lee는 Shim의 제안이 여전히 내부자 공격 (Insider Attack)에 취약하다는 것을 증명하면서, 그러한 공격을 막는 발전된 CHL04 프로토콜을 제시하였다 [15] (이하 “CHL08”이라고 한다.) 이 프로토콜에서는 세션 식별자가 메시지와 함께 전송되고, 다른 구성원들에 의해 인증된다. 또한, CHL04에서와 같이 서명 인증의 시간을 감소시키기 위해 모든 구성원의 서명을 한 번의 계산으로 인증하는 Batch Verification 알고리즘이 사용되었다.

[표 4] BC04 프로토콜

Round 1. U_i selects $a_i,\ r_i, b_{i,1}, \ldots, b_{i,n} \in Z_q^*$, sends $f_i(z) = r_i + b_{i,1}z + \ldots + b_{i,n-1}z^{n-1} \pmod p$ and signature to $U_j (1 \le i \ne j \le n)$.
Round 2. U_i checks the signature and sends $f_i = f_i(i) + \sum_{j \ne i} f_j(i) \bmod p$ to U_j.
Round 3. U_i interpolates $f(z)$ and retrieve r, then broadcasts $sk_{(i)} = a_{(i)} \cdot g^r \bmod p$ with a signature.
Key Computation. U_i computes the session key $sk = H(sk_{(i)})$.

[표 3]에 명시한 CHL08 프로토콜에서, *PID*는 각 세션마다 구성원이 바뀌기 때문에 변하고 *SID*는 random 값을 가지고 있기 때문에 세션마다 다른 값을 가지게 되므로, 서로 다른 세션에서 같은 값을 가질 확률은 희박하다. 따라서 메시지는 재전송되거나 위장될 수 없고, 내부자 공격 역시 막을 수 있다. 이 프로토콜은 키 계산 전에 Batch Verification이 한 번 더 추가되어 있기 때문에 각 구성원 당 CHL04 프로토콜보다 두 번 늘어난 6번의 Pairing 계산을 요구하며, 키 확인 과정이 없기 때문에 이와 관련된 취약점이 발생할 수 있다.

3.4. Bresson과 Catalano

Bresson과 Catalano는 세 번의 라운드를 가지는 GKA 프로토콜을 제안하였다 [7] (이하 "BC04"라고 한다.) 이 프로토콜에서는 각 구성원으로부터 세션 그룹 키를 조합하기 위해 보간법(Interpolation)을 사용하였다.

[표 4]의 프로토콜에서 각 구성원은 $2n$번의 단대단 메시지 전송과 키 확인을 위한 한번의 Broadcast 전송을 한다. Bresson과 Catalano는 이 프로토콜에 대해 Decisional Diffie-Hellman(DDH) 문제를 기반으로 하여 안전성을 증명하였다. 하지만 전송되는 모든 메시지는 구성원의 신원 정보 등 출처에 대한 정보를 포함하지 않기 때문에 악성 구성원에 의한 신원 위장 공격을 막을 수 없으며, 저자들 역시 이를 문제로 제기하고 이러한 능동 공격자의 위장 공격에 대한 방어가 필요하다고 명시하였다.

3.5. Kim, Kim, Ha와 Yoo

Kim, Kim, Ha와 Yoo는 한 번의 방송으로 정보를 교환하고 키를 합의 하는 신원 기반 GKA 프로토콜을 제안하였다 [8] (이하 "KKHY04"라고 한다.) 이 프로토콜은 구성원에 대한 인증 알고리즘을 포함하고 있으며, 각 구성원 U_i의 공개 키와 비밀 키 쌍은 수식 (2)와 같다.

[표 5] KKHY04 프로토콜

Round 1. U_i selects $a, a_i \in Z_q^*$, then broadcasts $< a_i P_{pub},\ P_i = aP,\ \ T_i = aa_i P_{pub} + H(P_i, a_i P_{pub}) S_i >$ *Key Computation.* U_i verifies $e(T_j, P) = e(H(P_j, a_j P_{pub}) Q_j, P_{pub}) \cdot e(a_j P_{pub}, P_j)$ for $1 \le i \ne j \le n$. If fails, then the protocol halts. Otherwise, U_i computes the session key $K_i = H_2(e(Q_1, a_1 P_{pub}) \dots e(a_i S_i, P) \dots e(Q_n, a_n P_{pub}))$

[표 5]의 KKHY04 프로토콜은 구성원 당 한 번의 메시지 전송만을 필요로 하므로 통신 시간에 있어서는 매우 효율적이지만, 전체 구성원의 수가 n명이라 하였을 때 각 구성원은 $4n-3$번의 Pairing 계산을 해야 하기 때문에 계산 면에서는 비효율적이다. 그리고 안전성의 측면에서 보았을 때는 몇 가지 취약점을 찾을 수 있다. 먼저 수동 공격자는 구성원의 개인 정보 없이 전송되는 메시지 $< a_i P_{pub},\ (P_i,\ T_i) >$ 만으로 세션 그룹 키를 계산할 수 있다. [표 5]의 키 합의 과정에서 세션 키를 생성하기 위한 식은 다음과 같다.

$$
\begin{aligned}
K_i &= H_2(\, e(Q_1,\, a_1 P_{pub}) \dots e(a_i S_i,\, P) \dots e(Q_n,\, a_n P_{pub})\,) \\
&= H_2(\, e(Q_1,\, a_1 P_{pub}) \dots e(a_i Q_i,\, P_{pub}) \dots e(Q_n,\, a_n P_{pub})\,) \\
&= H_2(\, e(Q_1,\, a_1 P_{pub}) \dots e(Q_i,\, a_i P_{pub}) \dots e(Q_n,\, a_n P_{pub})\,)
\end{aligned}
$$

위의 식과 같이 세션 그룹 키는 공개된 정보인 구성원의 공개 키와 $a_i P_{pub}$만으로 계산될 수 있으므로, 수동 공격자 또한 그룹 키를 쉽게 얻을 수 있다. 또한 방송되는 메시지가 타임스탬프나 세션 식별자를 포함하지 않기 때문에, 악성 사용자가 이전의 스크립트 $< a_i P_{pub},\ (P_i,\ T_i) >$를 가지고 있는 경우 이를 재전송 하여 U_i에 대한 신원 위장 공격을 할 수 있다. 그러므로 KKHY04 프로토콜은 재전송 공격 또는 수동 공격자로부터의 키 계산 공격에 취약함을 증명할 수 있다.

3.6. Dutta와 Barua

Dutta와 Barua는 KY03 프로토콜을 동적인 그룹으로 확장시킨 GKA 프로토콜을 제안하였다 [9] (이하 "DB05"라고 한다.) 이 프로토콜은 $U_n = U_0$이며 $U_{n+1} = U_1$인 Ring 네트워크를 사용하며, 구성원의 가입/탈퇴 (Join/Leave) 시 효율적으로 그룹 키를 재합의 하는 프로토콜도 함께 제시되어 있다.

[표 6]은 인증을 제공하지 않는 기본 GKA 프로토콜을 나타낸다. 이와 함께 디지털 서명을 사용한 인증 알고리즘이 포함된 프로토콜도 제안 되어 있지만 능동 공격자에 대한 안전성의 증명이 되어 있지 않기 때문에 여전히 KY03 프로토콜에서와 같이 악성 구성원으로부터의 신원 위장 공격에 취약하며, 키 확인(Key Comfirmation) 과정이 없어 이로 인한 취약점 또한 발생할 수 있다. 가입/탈퇴 (Join/Leave) 프로토콜 역시 안전성 증명에 있어 같은 취약점을 가진다.

[표 6] DB05 프로토콜

Round 1. Each user U_i selects random $r_i \in Z_q$ and sends $z_i = \alpha_i^r \bmod p$ to U_{i-1} and U_{i+1}.
Round 2. U_i computes $K_i^L = g^{x_{i-1} x_i}$ and $K_i^R = g^{x_{i+1} x_i}$, then broadcasts $X_i = K_i^L / K_i^R$.
Key Computation. U_i computes the key $K_i = \Pi_i K_i^R$.

[표 7] SCL05 프로토콜

Round 1. U_i selects random $a_i \in Z_q^*$, then sends $T_{i,j} = a_i Q_j$. to each U_j $(1 \le i \ne j \le n)$.
Key Computation. U_i computes the session key $K_i = e(T_{1,i} + \ldots + T_{i-1,i} + a_i Q_i + T_{i+1,i} + \ldots + T_{n,i}\ ,\ S_i)$.

3.7. Shi, Chen과 Li

Shi, Chen, Li는 한 번의 라운드를 가지는 신원 기반 그룹 키 인증 및 합의 프로토콜을 제안하였다 [10] (이하 "SCL05"라고 한다.) 이 프로토콜에서는 2.2장에서 설명한 신원 기반 암호시스템의 기본 구성과는 달리 각 구성원 U_i의 공개/비밀 키 쌍으로 $< Q_i = (H(ID_i)s_1 + s_2)P\ ,\ S_i = H(H(ID_i)s_1 + s_2)^{-1}P >$ 을 사용한다. 프로토콜의 진행은 [표 7]에 명시되어 있다.

이 프로토콜은 각 구성원들이 $n-1$번의 곱셈과 한 번의 Pairing만을 계산하는 등 매우 효율적이다. 하지만 각 구성원이 다른 구성원에 대해 메시지를 개별적으로 전송해야 하므로 방송 채널을 사용할 수 없다는 단점이 있으며, 안전성에 대한 증명이 제공되어 있지 않다. 2006년에 Zhou *et al.*은 그룹에 속해 있으며 유효한 공개/비밀 키 쌍을 가지고 있는 악성 구성원이 현재 시점에서 그룹 키를 합의하고 있지 않더라도 세션 그룹 키를 언제든 계산할 수 있다는 사실을 증명하였다. [13]

3.8. Abdalla, Bresson, Chevassut와 Pointcheval

Abdalla, Bresson, Chevassut와 Pointcheval은 BD04 프로토콜을 비밀번호 기반 암호 시스템을 이용하여 확장한 GKA 프로토콜을 제안하였다 [11] (이하 "ABCP06"이라 한다.) 2.2장에서 설명한 바와 같이 비밀번호 기반 프로토콜에서 각 구성원은 낮은 엔트로피의 비밀번호 pw를 가지고, 이를 이용해서 높은 엔트로피를 가지는 세션 그룹 키를 계산하게 된다.

[표 8] ABCP06 프로토콜

Round 1. U_i selects N_i and broadcasts (U_i, N_i)
Round 2. Session Identifier $S = U_1 \| N_1 \| \ldots \| U_n \| N_n$
U_i computes $k_i = H(S, i, pw)$
then broadcasts $z_i^* = E_{k_i}(z_i) = E_{k_i}(g^{x_i})$
where x_i is a random value.
Round 3. $z_{i-1} = D_{k_{i-1}}(z_{i-1}^*)$ and $z_{i+1} = D_{k_{i+1}}(z_{i+1}^*)$
U_i computes $Z_i = z_{i-1}^{x_i}$, $Z_{i+1} = z_{i+1}^{x_i}$
and broadcasts $X_i = Z_{i+1} / Z_i$.
Round 4. U_i computes
$K_i = Z_i^n \, X_i^{n-1} X_{i+1}^{n-2} \ldots X_{i+n-2}$
and broadcasts
$Auth_i = Auth(S, (z_j^*, X_j)_j, K_i, i)$.
Key Computation. U_i computes the session key $sk_i = G(S, (z_j^*, X_j, Auth_j)_j, K_i)$.

[표 8]에 명시된 프로토콜에서 각 구성원들은 3번의 라운드를 통해 그룹 키를 생성하고 1번의 라운드를 통해 생성된 키를 서로 확인한다. Abdalla *et al.*은 Random Oracle Model (ROM)과 Ideal Cipher Model (ICM)을 적용하여 수동 공격자와 능동 공격자 모두에 대한 안전성을 증명하였다. 하지만 비밀번호 기반 암호 시스템에서는 비밀번호 pw가 곧 구성원에 대한 인증이 되기 때문에 이것의 안전성에만 의존할 뿐 구성원에 대한 개별적인 인증 알고리즘은 제공하지 않는다. 따라서 pw를 가지고 있는 악성 구성원은 키 합의 과정에서 대칭 키로 사용되는 $k_i = H(S, i, pw)$ 역시 계산 할 수 있으므로 다른 구성원의 신원을 거짓 인증하여 위장 공격을 할 수 있다는 문제점이 있다.

3.9. Zhou, Susilo와 Mu

Zhou, Susilo와 Mu는 각각 한 번의 라운드와 두 번의 라운드를 가지는 두 종류의 GKA 프로토콜을 제안하였다 [13] (이하 "ZSM06-1". "ZSM06-2"라고 한다.)

ZSM06-1 프로토콜의 진행은 [표 9]와 같다. 이 프로토콜은 구성원들이 각자의 임시 키 k_i를 다른 구성원들의 공개키를 사용하여 암호화 한 뒤 보내면 다른 구성원들이 자신의 비밀 키를 이용해 복호화 하여 조합하는 형식으로, 한 번의 방송을 통해 그룹 키를 생성할 수 있어 통신에서는 매우 효율적이지만, 반면에 각 구성원이 암호화를 위한 $2(n-1)$번의 Pairing을 해야 하며 한 번에 전송되는 메시지의 크기가 크기 때문에 계산은 비교적 많은 편이다. 또한 세션 그룹 키를 계산하는 과정에서 해쉬 (Hash) 함수를 사용하고 있지 않으므로 마지막으로 메시지를 전송하는 구성원이 자신이 원하는 대로 키를 구성할 수 있게 하는 Key Control 문제가 발생한다. 이 프로토콜에서도 역시 키 확인 과정은 제공하지 않는다.

[표 9] ZSM06-1 프로토콜

Round 1. U_i selects $\delta_i \leftarrow G_2$, $r_i, k_i \leftarrow \{0,1\}^n$,
and computes
$P_i^j = r_i \oplus H_2(e(s_i, Q_j) \cdot \delta_i) \quad (1 \le i \ne j \le n)$
then broadcasts
$D_i = <\delta_i, P_i^1, \ldots, P_i^{i-1}, P_i^{i+1}, \ldots, P_i^n, H_3(r_i) \cdot k_i, L)$
Key Computation. U_i computes
$k_j' = H_3(H_2(e(Q_j, S_i) \cdot \delta_j) \oplus P_j^i) \oplus V_j$
then computes session key
$K = K_i = k_1' \oplus \ldots \oplus k_n'$.

[표 10] ZSM06-2 프로토콜

Round 1. Initiator U_1 selects random
$\delta \leftarrow G_2$, $r \leftarrow \{0,1\}^n$, $k_1 \leftarrow Z_P^*$,
and computes
$P_i = r \oplus H_4(e(S_1, Q_i) \cdot \delta) \ (2 \le i \le n)$
then broadcasts
$D_1 = <\delta, P_2, \ldots, P_n,$
$X_1 = H_5(r) \cdot k_1 P, Y_1 = k_1 P_{pub}, L>$
Round 2. $U_i (2 \le i \le n)$ finds P_i from D_1
using the label L
computes $r' = H_4(e(S_i, Q_1) \cdot \delta) \oplus P_i = r$
selects random $k_i \leftarrow Z_P^*$
then broadcasts
$D_i = <X_i, Y_i> = <H_5(r) \cdot k_i P, k_i P_{pub}>$.
Key Computation. $U_i (1 \le i \le n)$ computes
$z_i = H_5(r)^{-1} \cdot X_i$
If $e(P, \sum_{j=1}^{n} Y_j) = e(P_{pub}, \sum_{j=1}^{n} z_j)$ is verified,
then computes session key
$K = K_i = H_6(z_1) \oplus \ldots \oplus H_6(z_n)$.

[표 10]은 두 번의 라운드를 가지는 ZSM06-2 프로토콜이다. ZSM06-2 프로토콜은 그룹을 구성하는 Initiator(U_1)가 임시 그룹 키 r을 각 구성원의 공개키로 암호화 하여 방송하면 다른 구성원들이 복호화한 뒤 각자의 임시 키 k_i를 암호화 하여 전송 및 조합하는 방식이다. 이 프로토콜은 ZSM06-1보다 한 번 더 많은 라운드를 가지지만 첫 번째 라운드에서는 U_1만이 메시지를 전송하고 두 번째 라운드에서 다른 구성원들이 메시지를 전송하므로 각 구성원은 한 번의 메시지 전송만을 필요로 한다. 그리고 여기에서는 메시지의 인증 시간을 줄이기 위해 Batch Verification을 사용하였다. 하지만 ABCP06 프로토콜과 같이 그룹 키의 안전성은 임시 그룹 키 r의 안전성에 의존하고 메시지의 인증 또한 r의 소유 여부에 대해서만 수행되기 때문에, 이 값을 알고 있는 악성 구성원의 경우 다른 구성원을 위장 할 수 있다. 악성 구성원 $U_k (i \ne k)$의 U_i에 대한 위장 공격은 다음과 같다.

[표 11] YWJ08 프로토콜

Round 1. U_i selects random $a_i \in Z_q^*$,
then broadcasts $< P_i = a_i P,\ V_i = a_i P_{pub} + h_i S_i >$
where $h_i = H(U, e(P_i, P_{pub}))$.
Round 2. After receiving $< P_i, V_i >$ pairs,
U_i verifies
$e(\sum_{j \neq i} V_j, P) = e(\sum_{j \neq i} (P_j + h_j Q_j),\ P_{pub})$.
If fails, then the protocol halts.
Otherwise, U_i computes
$T = H_0(ID_1 \| P_1 \| ... \| ID_n \| P_n)$
and broadcasts
$< X_i, Y_i > = < a_i(P_{i+1} - P_{i-1} + T),\ a_i T >$.
Round 3. After receiving $< X_i, Y_i >$ pairs,
U_i verifies $e(\sum_{j \neq i} Y_j, P) = e(\sum_{j \neq i} P_j,\ T)$.
If fails, then the protocol halts.
Otherwise, U_i computes
$Z_i = e(n a_i P_{i-1} + \sum_{j=1}^{n-1} (n-1-j)(X_{i+j} - Y_{i+j}),\ P_{pub})$
and broadcasts
$C_i = H(i \| U \| P_1 \| ... \| P_n \| X_1 \| ... \| X_n \| Y_1 \| ... \| Y_n \| Z_i)$.
Key Computation. After checking the validity of every $C_j (1 \leq i \neq j \leq n)$,
U_i computes the session key
$K_i = H(U \| P_1 \| ... \| P_n \| X_1 \| ... \| X_n \| Y_1 \| ... \| Y_n \| Z_i \| C_1 \| ... \| C_n)$.

Round 2. $U_k (i \neq k)$ finds appropriate P_k from D_1 using the label L

computes $r' = H_4(e(S_i, Q_1) \cdot \delta) \oplus P_i = r$

selects random $k_i, k_k \leftarrow Z_P^*$

then broadcasts $D_i = < X_i, Y_i > = < H_5(r) \cdot k_i P,\ k_i P_{pub} >$

and $D_k = < X_k, Y_k > = < H_5(r) \cdot k_k P,\ k_k P_{pub} >$.

첫 번째 라운드에서 U_i에게 향하는 메시지를 가로챈 악성 구성원 $U_k (i \neq k)$는 자신의 비밀 키로 임시 키 r을 얻은 뒤, U_i의 메시지 D_i를 함께 생성하여 방송한다. 여기서 D_i의 계산은 U_i에 대한 개인 정보를 포함하고 있지 않기 때문에 r만 사용하여 계산 할 수 있다. 메시지를 받은 다른 구성원들은 이 값을 가지고 인증에 성공하기 때문에 U_i의 실제 존재여부를 알 수 없고, 따라서 그룹 키가 생성된다. 따라서 ZSM06-2에서는 악성 내부 공격자에 의한 신원 위장 공격이 가능하다.

3.10. Yao, Wang과 Jiang

2008년에 Yao, Wang, Jiang은 3번의 라운드를 가지며 인증 알고리즘을 포함하는 신원 기반 GKA 프로토콜을 제안하였다 [16] (이하 "YWJ08"이라고 한다.) 각각의 라운드는 사용자의 신원 인증, 그룹 키 합의, 그리고 키 확인으로 나누어진다. 이 프로토콜 역시 BD94 프로토콜을 기반으로 하였다.

이 프로토콜의 서명과 인증 방법은 CHL04 프로토콜에서와 같이 Pairing을 이용한 신원 기반 batch verification을 사용하고 있다. 또한, 세 번째 라운드에서 키 확인 과정을 수행하고, 각 메시지에 대한 인증과 세션 식별자를 포함시키는 등 안전성에 대한 요구사항을 단계별로 충족시키고 있다. 하지만 키 확인 과정에서 전송되는 메시지 C_i는 구성원의 비밀 키를 이용하여 암호화 되지 않았기 때문에 이에 따른 취약점이 발생할 우려가 있다.

Ⅳ. 각 프로토콜의 안전성과 성능 비교

앞 장에서는 최근에 제안된 일정 라운드 GKA 프로토콜들에 대해 조사하고 간략히 분석해 보았다. 이 장에서는 2.1장에서 설명한 GKA 프로토콜의 안전성에 대한 요구사항에 따라 제안된 프로토콜들을 분석해보고, 또한 몇 가지 기준에 따른 프로토콜의 성능을 자세히 분석해 보도록 한다.

4.1. 안전성

2.1장에서 설명한 것과 같이 GKA 프로토콜의 안전성은 수동 공격자, 능동 공격자, 악성 구성원 등 공격자의 종류에 따라 정의할 수 있다. [표 12]는 3장에서 조사한 프로토콜에 대한 안전성 요구사항 만족 여부를 나타낸 것이다. "o"는 프로토콜이 요구사항을 만족시키거나 제공하는 것이고, "x"는 그의 반대이며, "△"는 논문에서 해당 요구사항을 만족 하거나 제공한다고 명시되어 있지만 실제로는 공격이 가능하여 취약점을 가지는 경우 말한다. 예를 들어 프로토콜이 Entity Authentication에 대해 "△"로 표기되어 있다면 이는 신원 인증 알고리즘을 제공하지만 위장 공격이 여전히 가능한 경우이다. "△" 에 대한 분석은 다른 논문에서 증명 되었거나 본 논문에서 제기한 분석을 통해 표기되어 있다. ABCP06 프로토콜이나 CHL08 또는 YWJ08 등 최근에 나온 프로토콜의 경우 요구사항을 비교적 많이 만족시키고 있다. Key Confirmation의 경우 대개 한 라운드를 더 거쳐야 하기 때문에 효율성을 중요시하는 대부분의 일정 라운드 GKA 프로토콜에서는 키 확인 과정을 제공하지 않고 있다. 그리고 이전의 프로토콜들은 Entity Authentication을 만족시키지 못하는 것에 비해 최근에 제안되는 프로토콜들은 이를 거의 만족시키고 있다. 이것은 최근의 논문들에서 Entity Authentication을 주요 요구사항으로 간주하고 GKA 프로토콜에서 기본적으로 인증 알고리즘을 제공하는 AGKA 프로토콜로 제안하고 있기 때문이다.

4.2. 성능

[표 13]은 프로토콜들의 성능 비교를 나타낸다. 프로토콜들 중 인증 알고리즘에 대해 명확히 기술하지 않은 경우에는 인증을 제공하지 않는 GKA 프로토콜의 성능을 표기하였고 (U)로 표시하였다.

[표 12] 안전성 비교

프로토콜	BD94	KY03	CHL04	BC04	KKHY04	DB05	SCL05	ABCP06	ZSM06-1	ZSM06-2	CHL08	YWJ08
Key Privacy	o	o	o	o	x	o	x	o	o	o	o	o
Indistinguishability	o	o	o	o	x	o	x	o	o	o	o	o
Freshness	o	o	o	o	o	o	o	o	o	o	o	o
F/B Secrecy	o	o	o	o	o	o	o	o	o	o	o	o
PFS	o	o	o	o	x	o	x	o	x	x	o	o
Key Integrity	x	x	x	x	x	o	o	o	o	o	o	o
Key Confirmation	x	x	x	o	x	x	x	o	x	x	x	△
Correctness	o	o	o	o	o	o	o	o	o	o	o	o
Entity Authentication	x	x	△	o	△	x	o	x	o	△	o	o
Contributory	x	x	o	o	o	x	o	o	x	o	o	o

[표 13] 성능 비교

Freshness	Round	Ucasts	Bcasts	Msize	Exponent	G_1-Mul	G_2-Mul	Pairing
BD94 (U)	2	0	$2n$	$2n$	$n(n+1)$	$n(n+1)$	0	0
KY03 (U)	2	0	$2n$	$2n$	$n(n+1)$	$n(n+1)$	0	0
CHL04	2	0	$2n$	$3n$	$n(n-1)$	$8n$	$n(n-1)$	$4n$
BC04	3	$2n(n-1)$	$2n$	$n(3n+1)$	$2n^2$	$n(4n-2)$	0	0
KKHY04	1	0	n	$3n$	0	$n(n+4)$	0	$n(4n-3)$
DB05	2	$2n$	n	$2n$	$3n$	$n(2n-1)$	0	0
SCL05	1	$(n-1)^2$	0	$n(n-1)$	0	n^2	0	n
ABCP06	4	0	$4n$	$3n$	$n(n+2)$	n^2	0	0
ZSM06-1	1	0	n	$n(n+2)$	0	0	$2n(n-1)$	$2n(n-1)$
ZSM06-2	2	0	n	$3n+1$	0	$n(n+3)$	$2(n-1)$	$3n$
CHL08	2	0	$2n$	$6n$	$n(n-1)$	$11n$	$n(n-1)$	$6n$
YWJ08	3	0	$3n$	$5n$	0	$2n(n+3)$	0	$n(n+5)$

성능 비교의 기준은 다음과 같으며, n은 전체 그룹 구성원의 수를 나타낸다.

Round : 전체 라운드 수

Pairing : 전체 구성원의 Pairing 계산 횟수

Ucasts : 전체 구성원의 단대단 메시지 전송 횟수

Bcasts : 전체 구성원의 방송 메시지 전송 횟수

Msize : 전송되는 전체 메시지의 수 (한 번의 방송이 두개 이상의 메시지를 포함하는 경우 방송 메시지 전송 횟수와 메시지의 수가 같지 않은 경우도 있다.)

Exponent : 전체 구성원의 지수 계산 횟수

G_1-Mul : 전체 구성원의 G_1 그룹 내에서 곱셈 횟수

G_2-Mul : 전체 구성원의 G_2 그룹 내에서 곱셈 횟수

CHL08이나 YWJ08 등 대부분의 안전성 요구사항을 만족하는 프로토콜들의 경우 많은 시간을 필요로 하는 Pairing이나 곱셈 계산을 수행하는 것을 알 수 있다. 또한 인증 알고리즘을 제공하지 않는 BD94와 KY03을 제외

하고는 성능 면에서 ZSM06-2 프로토콜이 가장 효율적이지만 PFS나 Key Confirmation을 만족시키지 못하고 신원 위장 공격이 가능하기 때문에 안전성 면에서는 불안정하다. 이와 같이 성능과 안전성을 비교한 표를 통해서 현재 제안된 GKA 프로토콜들이 안전성과 성능에 있어 Tradeoff를 가지는 것을 알 수 있다.

Ⅴ. 결 론

본 논문에서는 최근에 제안된 GKA 프로토콜들을 조사하고, 앞서 제시된 기준에 따라 성능과 안전성에 대해 비교 및 분석해 보았다. 현재까지 제안된 프로토콜들은 이전에 발견된 취약점을 개선하고 안전성을 증명하거나 성능을 개선하는 등 발전해 오고 있지만 성능이 뛰어난 프로토콜들이 몇 가지의 안전성 요구사항을 부합시키지 못하거나 안전한 프로토콜이 비교적 성능이 뒤떨어지는 등 각자 부족한 점을 가지고 있기 때문에, 특정 응용 프로그램에 적용하기 위해서는 프로그램 환경의 특성을 분석한 뒤 적합한 GKA 프로토콜을 선택해야 할 것이다. 또한, 현재까지 제안된 프로토콜들이 효율성과 안전성에 있어 완전하지 않기 때문에 GKA 프로토콜의 설계에 있어서도 아직까지 개선의 여지가 있다. 그렇기 때문에 여러 환경에서 보다 안전하고 효율적으로 그룹 구성원들이 키를 합의하도록 하는 GKA 프로토콜의 설계에 대한 계속적인 연구가 필요하다.

참 고 문 헌

[1] W. Diffie and M. Hellman, "New Direction in Cryptography," *IEEE Transactions on Information Theory* 22(6), pp. 644-654, 1976.

[2] M. Burmester and Y. Desmedt. "A Secure and Efficient Conference Key Distribution System," In Advances in Cryptology, EUROCRYPT'94, LNCS 950, pp. 275-286. *Springer*, May 1994.

[3] D. Boneh and M. Franklin, "Identity-based Encryption from the Weil Pairing," Proc. of Crypto'01, LNCS 2139, pp. 213-229, *Springer-Verlag*, 2001.

[4] J.Katz and M.Yung. "Scalable Protocols for Authenticated Group Key Exchange," In proc. of Crypto'03, LNCS 2729, pp.110-125, *Springer*, 2003.

[5] F. G. Zhang and X.F. Chen, "Attack on Two ID-based Authenticated Group Key Agreement Schemes," *Cryptology ePrint Archive*: Report 2003/259.

[6] K. Y. Choi, J. Y. Hwang and D. H. Lee, "Efficient ID-based Group Key Agreement with Bilinear Maps," PKC'04, LNCS 2947, pp.130-144, *Springer-Verlag*, 2004.

[7] E. Bresson and D. Catalano. "Constant Round Authenticated Group Key Agreement via Distributed Computation," PKC'04, LNCS 2947, pp.115-129, *Springer-Verlag*, 2004.

[8] J. S. Kim, H. C. Kim, K. J. Ha, K. Y. Yoo, "One Round Identity-Based Authenticated Conference Agreement Protocol," ECUMN 2004, LNCS 3262, pp. 407-416, *Springer-Verlag*, 2004.

[9] R. Dutta and R. Barua. "Constant Round Dynamic Group Key Agreement," In Information Security: 8th

International Conference -ISC'05, LNCS 3650, pp. 74-88. *Springer-Verlag*, 2005.

[10] Y. Shi, G. Chen, J. Li, "ID-Based One Round Authenticated Group Key Agreement Protocol with Bilinear Pairings," International Conference on Information Technology: Coding and Computing (ITCC'05), -Volume I, pp.757-761, 2005.

[11] M. Abdalla, E. Bresson, O. Chevassut, and D. Pointcheval, "Password-Based Group Key Exchange in a Constant Number of Rounds," PKC'06, LNCS 3958, pp. 427-442. *Springer-Verlag*, 2006.

[12] M. Manulis, "Survey on Security Requirements and Models for Group Key Exchange," Technical Report 2006/02, Horst-Gortz Institute, November 2006.

[13] L.Zhou, W. Susilo, Y. Mu, "Efficient ID-based Authenticated Group Key Agreement from Bilinear Pairings," Mobile Ad-hoc and Sensor Networks -MSN 2006, LNCS 4325, pp. 521-532, *Springer-Verlag*, 2006.

[14] K. A. Shim, "Further Analysis of ID-Based Authenticated Group Key Agreement Protocol from Bilinear Maps," *IEICE Trans. Fundamentals*, vol.E90-A, no.1, pp.231-233, 2007.

[15] K. Y. Choi, J. Y. Hwang and D. H. Lee, "ID-Based Authenticated Group Key Agreement Secure against Insider Attacks," *IEICE Trans. Fundamentals*, vol.E91-A, no.7, pp.1828-1830, 2008.

[16] G. Yao, H. Wang, Q. Jiang, "An Authenticated 3-Round Identity-Based Group Key Agreement Protocol," In proc. of the third International Conference on Availability, Reliability, and Security - ARES'08, pp. 538-543, *ACM*, 2008.