저전력 모바일 장치에 적합한 효율적인 동적 그룹 키 동의

조 석 향[†] · 남 정 현[†] · 김 승 주[††] · 원 동 호[††] · 이 혜 주[†††] · 최 진 수[†††]

요 약

그룹 키 동의 프로토콜은 공개된 통신망을 통해 안전하게 통신하려는 그룹 구성원들에게 세션키를 제공하기 위해 설계된다. 모바일 컴퓨팅 기술은 프로토콜 참가자의 계산적인 능력 측면에서 비대칭적이다. 즉 프로토콜 참가자는 충분한 계산 능력을 가진 고정된 서버(애플리케이션 서버)와 제한된 계산 자원을 가진 모바일 장치들(클라이언트)로 구성되어 있다. 스마트카드나 개인 휴대 정보 단말기(PDA)와 같은 저전력 모바일 장치를 포함하는 그룹에서는 각 구성원의 계산량을 최소화 하는 것이 바람직하다. 또한 그룹 구성원이 현재의 그룹을 탈퇴하거나 다수의 새로운 구성원이 현재의 그룹에 참여할 필요가 있을 때 적은 계산 비용으로 그룹 키의 갱신이 요구된다. 본 논문에서는 저전력 모바일 장치인 클라이언트 쪽의 계산 효율성이 높은 동적 그룹 키 동의 프로토콜을 제안한다. 제안한 프로토콜과 기존 프로토콜을 통신과 계산 비용 측면에서 비교하고, 제안한 프로토콜의 안전성은 랜덤 오라클 모델에서 수동적인 공격자에 대하여 안전함을 증명한다.

An Efficient Dynamic Group Key Agreement for Low-Power Mobile Devices

Seokhyang Cho[†] · Junghyun Nam[†] · Seungjoo Kim[††] · Dongho Won[††] · Hyejoo Lee[†††] · Jinsoo Choi[†††]

ABSTRACT

Group key agreement protocols are designed to provide a group of parties securely communicating over a public network with a session key. The mobile computing architecture is asymmetric in the sense of computational capabilities of participants. That is, the protocol participants consist of the stationary server(application servers) with sufficient computational power and a cluster of mobile devices(clients) with limited computational resources. It is desirable to minimize the amount of computation performed by each group member in a group involving low-power mobile devices such as smart cards or personal digital assistants(PDAs). Furthermore we are required to update the group key with low computational costs when the members need to be excluded from the group or multiple new members need to be brought into an existing group. In this paper, we propose a dynamic group key protocol which offers computational efficiency to the clients with low-power mobile devices. We compare the total communicative and computational costs of our protocol with others and prove its security against a passive adversary in the random oracle model.

키워드 : 그룹 키 동의(Group Key Agreement), 모바일 장치(Mobile Device), 멀티캐스트(Multicast), 계산적 Diffie-Hellman 가정(Computational Diffie-Hellman Assumption)

1. 서 론

모바일 장치 상에서 인터넷을 통한 전자 거래가 대중화됨에 따라 무선 기술이 더 널리 퍼지게 되었다. 그러나 모바일 장치의 계산적인 능력의 제약으로 인하여 암호 프로토콜을 설계할 때 현재의 보안 기술을 무선 인터넷에 그대로 적용하기는 어렵다. 한편 저장 공간에 대한 제약사항은 메모리 카드를 추가로 이용할 수 있으므로 점점 문제가 줄어들고 있다.

모바일 컴퓨팅 기술은 프로토콜 참가자의 계산적인 능력 측면에서 비대칭적이다. 즉, 프로토콜 참가자는 충분한 계산 능력을 가진 고정된 서버(애플리케이션 서버 또는 서비스 제공자)와 클라이언트라 부르는 제한된 계산 자원을 가진 모바일 장치의 집합으로 구성되어 있다. 이러한 비대칭적 모바일 환경은 인터넷 주식 시세, 오디오와 음악 전송, 시청한 프로그램 수에 따라 요금을 지불하는 pay-per-view 방식의 TV 서비스, 소프트웨어 갱신 등과 같은 많은 애플리케이션에 공통이다[1, 2, 3, 4].

이러한 애플리케이션을 사용하기 위해서는 그룹 구성원들이 공개된 통신망을 통하여 안전한 방법으로 공통의 그룹 키를 나눠가져야 한다. 그룹 키 프로토콜은 다자간 프로토콜로, 프로토콜에 참여하는 개체에게만 알려진 공통 비밀키를 생성하고 어떤 개체도 이 키를 미리 결정할 수 없도록

※ 본 연구는 2004년도 한국학술진흥재단(KRF-2004-003-D00392)의 지원 및 한국전자통신연구원 위탁연구 과제(0201-2004-0027)의 지원으로 수행되었음.

†준 회 원 : 성균관대학교 정보통신공학부 박사과정
††종신회원 : 성균관대학교 정보통신공학부 교수
†††정 회 원 : 한국전자통신연구원 방송미디어연구그룹 선임연구원

논문접수 : 2004년 9월 2일, 심사완료 : 2004년 12월 21일

공개 정보를 교환한다. 일반적으로 세션키로 불리는 이 비밀 키는 개체들 사이의 인증(authentication), 기밀성(confidentiality), 데이터 무결성(data integrity) 등과 같은 보안 서비스에 사용될 수 있다.

키 설정 프로토콜은 세션키 생성 관점에서 키 전송(key transport)과 키 동의(key agreement) 프로토콜로 나눌 수 있다. 키 전송 프로토콜은 참가자 한 명이 세션키를 생성하여 안전하게 다른 참가자들에게 전송하는 방식이다. 반면, 키 동의 프로토콜은 한 명 이상의 참가자가 공통의 세션키를 생성하는 데 자신의 정보를 제공하는 방식이다. 또한 프로토콜에 참여하는 모든 참가자가 세션키 설정에 자신의 정보를 제공하는 경우를 contributory 키 동의 프로토콜이라 한다. 그리고 모든 참가자가 동일한 구조를 갖고 메시지를 전송하고 계산을 수행할 때 역할 대칭적(role symmetric)이라 한다. 본 논문은 클라이언트의 계산량이 적은 비대칭적(asymmetric) contributory 키 동의 프로토콜로, 참가자가 선택하는 수의 랜덤성이 보장된다면 서로 공모하더라도 세션키 값을 제어할 수 없는 특징을 갖는다.

공개키 암호 기법에서, 단기간 사용하게 되는 공개키(short-term public key) 값은 일반적으로 세션키를 설정하는 데 한 번만 사용되고, 종종 일회용 키(ephemeral key)라고 부른다. 한편, 장기간 사용하게 되는 공개키(long-term public key)는 주로 프로토콜 참여자를 인증하는데 사용되는 고정 키이다.

키 동의는 암호기법의 기본 요소 중의 하나이다. 키 동의에 관한 최초의 프로토콜은 Diffie-Hellman[5]이 제안하였으나, 통신하는 당사자를 인증하는 내용이 없어서 man-in-the middle 공격을 당한다. 이후에 많은 프로토콜이 전자 서명을 키 동의 프로토콜에 결합시켜 이를 해결하였다. 공격은 프로토콜이 의도한 목표가 충족되지 않거나 안전성 속성이 만족되지 않을 때 발생한다. 수동적 공격(passive attack)은 공격자가 프로토콜 실행을 단순히 관찰함으로써 공격 목표를 달성하는 것이고, 능동적 공격(active attack)은 공격자가 프로토콜의 여러 개의 인스턴스 생성(instantiation)을 가로챔으로써, 메시지를 삭제, 삽입, 변경 또는 리다이렉트(redirect)할 수 있는 능력이다. 본 논문에서는 Diffie-Hellman 가정에 기반 하여 효율적인 동적 그룹 키 동의 프로토콜을 제안한다. 제안한 프로토콜은 Emmanuel Bresson 등[6]의 아이디어와 Junghyun Nam 등[7, 8]의 프로토콜을 결합한 것으로, 클라이언트 쪽의 계산 효율성이 높고, 그룹 구성원의 탈퇴나 가입 시 계산 비용이 적어 동적 그룹에 적합하다. [6]에서는 전방향 안전성을 제공하지 않지만, 제안한 프로토콜에서는 완전한 전방향 안전성(perfect forward secrecy)을 제공하고, [7, 8]에 비해서는 곱셈 연산 대신 해쉬 함수와 XOR 연산을 사용하여 계산적인 효율성을 높였다. 또한 제안한 프로토콜이 수동적 공격자에 대하여 안전함을 증명한다.

본 논문의 구성은 다음과 같다. 2장에서는 그룹 키에 대한 관련 연구들을 살펴보고, 3장에서는 제안한 프로토콜의 초기 세션키 설정으로부터 그룹의 구성원이 탈퇴하거나, 다시 참여 또는 새로이 가입할 경우를 자세히 설명한다. 4장에서는 제안한 키 동의 방식과 기존 방식의 효율성을 비교하고, 제안한 프로토콜의 안전성을 증명한다. 마지막으로 5장에서는 결론을 맺고, 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 Ingemarsson 등(ING)의 프로토콜

Ingemarsson 등[9]은 널리 알려진 Diffie-Hellman의 양자간 키 분배 프로토콜을 다자간으로 확장하는 아이디어를 처음으로 생각해냈다. 이들은 그룹의 구성원을 논리적인 원(ring)으로 배열하여, 모든 구성원이 이전 구성원에게서 수신한 중간에 생성된 값에 자신이 선택한 임의의 수를 지수승한 값과 곱한 값을 다음 구성원에게 전송하는 방식으로 $n-1$라운드 만에 n명이 모두 동일한 키를 계산해 낸다.

예를 들면 그룹의 구성원이 U_1, U_2, U_3, U_4인 경우 U_2가 U_3에게 각 라운드마다 전송하는 메시지는 다음과 같다. 여기서 p는 큰 소수이고, g는 모듈러 p 상에서 원시원소이고, $r_i\,(i \in [1,4])$는 $[1, p-1]$ 사이의 U_i가 선택한 임의의 수이다.

1라운드 $g^{r_2} \bmod p$

2라운드 $g^{r_1+r_2},\ g^{r_1 r_2} \bmod p$

3라운드 $g^{r_4+r_1+r_2},\ g^{r_4 r_1 + r_4 r_2 + r_1 r_2} \bmod p$

마지막으로 U_3는 마지막 메시지의 첫 부분에 자신이 선택한 임의의 수 r_3를 지수승한 값과 두 번째 부분을 곱하여 다음과 같은 공통 세션키를 얻는다.

$$(g^{r_4+r_1+r_2})^{r_3} g^{r_4 r_1 + r_4 r_2 + r_1 r_2} \bmod p$$
$$= g^{r_4 r_3 + r_1 r_3 + r_2 r_3 + r_4 r_1 + r_4 r_2 + r_1 r_2} \bmod p$$

ING 프로토콜에서는 $n-1$라운드, 라운드 당 n개의 메시지로 총 $n(n-1)$개의 메시지, 라운드 당 1번의 지수승과 마지막 공통 세션키 계산에 1번의 지수승이 필요하므로 사용자당 n번의 지수승과 $\frac{1}{2}(n+1)(n-2)$번, 즉 $O(n^2)$의 곱셈 연산이 필요하다.

2.2 Burmester/Desmedt(BD)의 프로토콜

Burmester와 Desmedt은 [10]에서 스타(star) 기반의 시스템, 트리(tree) 기반의 시스템, 순환(cyclic) 시스템, 브로드캐스트(broadcast) 시스템을 제안하였는데, 브로드캐스트 시스템에서 다음과 같이 n명의 구성원이 2라운드 만에 공통 세션키 $g^{r_1 r_2 + r_2 r_3 + \cdots + r_n r_1} \bmod p$를 얻는 매우 효율적인 프로토콜을 구성하였다. 이를 흔히 BD 프로토콜이라 부르고 있다. 다음과 같은 단계를 거쳐 공통의 세션키를 얻을 수 있고, 여기서 인덱스(index)는 사이클(cycle)로 생각하여 U_{n+1}은 U_1이 되고 U_0는 U_n이 된다.

단계 1. 각 U_i는 $r_i \in_R Z_q$를 선택하여, $z_i = g^{r_i} \bmod p$를 계산한 다음, z_i를 브로드캐스트 한다.

단계 2. 각 U_i는 $X_i = (z_{i+1}/z_{i-1})^{r_i} \bmod p$를 계산하여 X_i를 브로드캐스트 한다.

단계 3. 각 U_i는 공통 세션키 $SK_i = (z_{i-1})^{nr_i} X_i^{n-1} X_{i+1}^{n-2} \cdots X_{i-2} \pmod{p}$를 계산할 수 있다.

BD 프로토콜에서는 라운드 당 n번의 브로드캐스트를 필요로 하므로, $2n$번의 브로드캐스트, 각 사용자는 단계마다 한 번의 지수승으로 총 3번의 지수승과 마지막 단계에서 $O(n\log n)$번의 곱셈 연산을 필요로 한다. 물론 2단계에서 역연산도 필요하다. 또한 각 사용자는 2번의 서명을 생성하고 $2(n-1)$번의 서명을 검증한다. 따라서 BD 프로토콜은 각 사용자의 서명 검증과 모듈러 곱셈 비용 때문에 사용자의 수가 증가함에 따라 계산 효율성이 매우 낮아진다.

2.3 Steiner 등의 프로토콜

Steiner 등은 [11]에서 Diffie-Hellman의 양자간 키 분배를 그룹 통신에 확장한 3가지 그룹 키 분배 프로토콜을 제안하였는데, 이 중 두 번째 프로토콜 GDH.2는 다음과 같이 공통키를 설정한다. q는 길이가 k비트인 소수, $2q+1=q'$ (q'은 소수)이고, g는 위수가 q인 Z_q^*의 부분군 $\mathbb{G}$의 생성원일 때, 그룹의 구성원 $U_i (i \in [1, n-1])$는 U_{i-1}에게 받은 데이터 각각에 자신이 선택한 임의의 수 $r_i (\in \mathbb{G})$ 지수승을 한 값과 $g^{r_1 r_2 \cdots r_{i-1}}$을 U_{i+1}에게 전송한다. U_n은 U_{n-1}에게 받은 데이터에 r_n 지수승을 하여, 공통키 $g^{r_1 r_2 r_3 \cdots r_{n-1} r_n}$을 얻고, 마지막 라운드에서 U_n은 중간에 생긴 데이터를 다른 구성원에게 브로드캐스트 하여 다른 구성원도 공통의 세션키를 계산할 수 있게 한다. <표 1>에 GDH.2의 세션키 설정 과정에서 전송하는 메시지를 나타내었다.

〈표 1〉 GDH.2의 세션키 설정 과정

$U_i \rightarrow U_{i+1}$	전송하는 메시지
$U_1 \rightarrow U_2$	$\{g, g^{r_1}\}$
$U_2 \rightarrow U_3$	$\{g^{r_2}, g^{r_1}, g^{r_1 r_2}\}$
$U_3 \rightarrow U_4$	$\{g^{r_2 r_3}, g^{r_1 r_3}, g^{r_1 r_2}, g^{r_1 r_2 r_3}\}$
$U_4 \rightarrow U_5$	$\{g^{r_2 r_3 r_4}, g^{r_1 r_3 r_4}, g^{r_1 r_2 r_4}, g^{r_1 r_2 r_3}, g^{r_1 r_2 r_3 r_4}\}$
...	...
$U_{n-1} \rightarrow U_n$	$\{g^{r_2 r_3 \cdots r_{n-1}}, \cdots, g^{r_1 r_2 \cdots r_{n-1}}\}$
$U_n \rightarrow \{U_1, U_2, \cdots, U_{n-1}\}$	$\{g^{r_2 r_3 \cdots r_{n-1} r_n}, \cdots, g^{r_1 r_2 \cdots r_{n-2} r_n}\}$

위의 프로토콜에서는 $n-1$번의 유니캐스트(unicast)와 1번의 브로드캐스트(broadcast)로 총 n라운드 만에, n개의 메시지로 동일한 키를 얻고, U_i당 $i+1 (i<n)$번의 지수승과, U_n은 n번의 지수승 연산이 필요하다.

또한 Steiner 등은 [12]에서 동적 동등 그룹(Dynamic Peer Groups : DPGs)에 GDH.3(3번째 Group Key Distribution)를 적용하여, $n+1$라운드 만에 공통 세션키를 얻는 방법을 제안하였고, 키 동의에 여러 가지 연산을 정의하였다. 즉, 이 프로토콜은 동등(peer) 그룹에서 동적으로 세션키를 생성하기 위하여, 새 구성원 한 명이 기존 그룹에 추가로 들어오는 연산(member addition), 기존 구성원 한 명이 탈퇴하는 연산(member deletion), 그룹을 형성하지 않은 여러 명의 구성원이 동시에 프로토콜에 참여하는 연산(mass join), 탈퇴하는 연산(mass leave), 두 그룹이 더 큰 그룹으로 병합하는 연산(group fusion), 이전의 병합된 그룹을 따로 분할하는 연산(group fission), 하나로 된 그룹을 좀더 작은 그룹으로 분할하는 연산(group division) 등을 고려하였다.

스마트카드나 PDA와 같은 낮은 전력을 필요로 하는 장치 또는 대규모 그룹 환경에서는 그룹 구성원의 계산량을 최소화 하는 것이 바람직하다. [12]의 초기 설정 프로토콜(Initial Key Agreement : IKA) 중 두 번째 프로토콜 IKA.2는 이러한 문제를 다루기 위하여 구성되었다.

IKA.2에서는 GDH.3 프로토콜을 사용하여 다음과 같이 4단계로 구성되어 있다.

단계 1. $n-2$라운드까지는 구성원 U_i가 U_{i-1}에게 받은 중간 값 $g^{r_1 r_2 \cdots r_{i-1}}$에 자신이 선택한 값의 지수승을 한 $g^{r_1 r_2 \cdots r_i}$를 U_{i+1}에게 전송한다.

단계 2. $n-1$라운드에서는 U_{n-1}이 나머지 $U_i (i \in [1, n-2])$에게 $g^{r_1 r_2 \cdots r_{n-1}}$을 계산하여 브로드캐스트 한다.

단계 3. n라운드에서는 $U_i (i \in [1, n-1])$가 각각 U_n에게 $g^{r_1 r_2 \cdots r_{n-1}/r_i}$를 전송한다.

단계 4. $n+1$라운드에서는 U_n이 각 $U_i (i \in [1, n-1])$에게 $g^{r_1 r_2 \cdots r_n / r_i}$를 브로드캐스트 한다.

위의 프로토콜에서는 $n+1$라운드 만에 $2n-1$개의 메시지, 즉 2번의 브로드캐스트와 $2n-3$번의 유니캐스트 통신이 필요하고, $U_i (i \in [1, n-1])$는 3번, U_n은 n번의 지수승이 필요하여 총 $4n-3$번의 지수승 연산이 필요하다[12].

2.4 Bresson 등의 프로토콜

최근, Bresson 등의 초기 연구[13]에서는 그룹 구성원이 고정되어 있다고 가정한 반면, 후반의 연구[6, 14, 15]는 동적인 경우에 초점을 맞춰, 기본 설정 알고리즘뿐만 아니라 탈퇴 알고리즘(remove algorithm)과 참여 알고리즘(join algorithm)으로 구성되어 있다. [6]에서 Bresson 등은 전력 소모가 낮은 제약사항을 갖는 장치로 구성된 비균형 네트워크와 무선 게이트웨이에 적합한 매우 효율적이고 안전성이 증명된 그룹 키 동의 프로토콜을 제안하였다. 즉, 유한 순환군(cyclic group)에서 온라인으로 계산을 수행하는 데는 자원이 부족하지만, 대칭적인 암호화 연산은 가능한 모바일 장치에 적용시킨 그룹 키 동의 방법을 제시하였다. 이 프로토콜에서 서버의 묵시적 인증은 복호화 능력을 증명함으로써 이루어지고, 모바일의 묵시적 인증은 서명을 통하여 달성하였다. 저전력 모바일 장치에서 계산해야 하는 서명과 암호화 연산은 미리 계산할 수 있어서 온라인상에서는 아주

적은 계산만 수행하므로 계산적으로 매우 효율적이다. 구체적인 세션키 설정 과정은 다음과 같다.

단계 1. 각 클라이언트 $U_i (i \in [1, n-1])$는 $r_i \in_R Z_q$를 선택하여, $z_i = g^{r_i}$, $x_i = z^{r_i} (z = g^r$은 서버의 공개키 값)을 계산한 후, z_i의 서명값 σ_i와 z_i를 서버 U_n에게 전송한다.

단계 2. 서버 U_n은 각 U_i의 서명을 검증한 다음, 옳다면 $x_i = z_i^r$을 계산한다. 또한 카운터 $c = 0 \in \{0, 1\}^l$를 초기화한 다음, 공유 비밀 데이터 $K = H_0(c \| x_1 \| \cdots \| x_{n-1})$를 정의하여 $K_i = K \oplus H_1(c \| x_i)$를 계산한 다음, 카운터 값 c와 K_i를 U_i에게 전송한다.

단계 3. 각 클라이언트 U_i(그리고 서버)는 공통 비밀 데이터 $K = K_i \oplus H_1(c \| x_i)$를 계산한 다음, 세션키 $SK = H(K \| U \| U_n)$을 계산할 수 있다.

여기서, H_0, H_1은 해쉬 함수이고, U는 클라이언트의 집합, U_n은 서버를 나타낸다.

Bresson 등의 프로토콜에서는 라운드 당 클라이언트 수인 $n-1$번의 유니캐스트 통신을 필요로 하므로, 총 $2(n-1)$번의 유니캐스트, 각 클라이언트 당 2번의 지수승과 기지국인 서버는 $(n-1)$번의 지수승 연산이 필요하다. 클라이언트에 대한 거의 모든 계산이 오프라인 상으로 미리 계산할 수 있어서 매우 효율적인 프로토콜이다. 그러나 카운터를 사용하므로 프로토콜 참가자들 사이에 동기를 맞추어야 하는 제약사항이 따른다. 또한 서버의 장기간 사용하는 키 r이 노출되면 모든 x_i가 z_i와 r로부터 쉽게 계산되어 모든 세션키가 복구될 수 있으므로 전방향 안전성(forward secrecy)을 만족하지 않고, 서버가 공유 비밀 데이터 K 값을 프로토콜의 실행대로 계산하지 않고 임의로 선택할 수 있어서 키 제어가 쉽다.

3. 제안하는 방식

먼저 초기 그룹 키 동의 프로토콜 P를 설명하고, 몇몇 클라이언트가 탈퇴하는 프로토콜 P_{leave}와 새로운 클라이언트 또는 기존의 클라이언트가 합류하는 프로토콜 P_{join}을 알아본다.

본 논문에서 사용하는 세션키 설정을 위한 시스템 파라미터의 정의는 다음과 같다.

- $\{U_1, U_2, \cdots, U_n\}$: 그룹 구성원의 집합
- U_n: 서버, $U_1, U_2, \cdots, U_{n-1}$: 클라이언트
- $p = 2 \cdot q + 1$ (q는 소수) : 큰 소수
- g: 모듈러 p 상에서 위수 q를 갖는 순환군 $\mathbb{G}$의 생성원
- K_i: 그룹 구성원 U_i의 비밀 서명키
- PK_i: 그룹 구성원 U_i의 서명 검증키
- $Sign$: 서명 알고리즘, $Verify$: 검증 알고리즘
- $H()$: 일방향 해쉬 함수
- SK: 그룹 구성원 사이에 공유된 세션키

3.1 초기 세션키 설정 프로토콜(P)

공개 파라미터 p, g가 모든 프로토콜 참가자에게 알려져 있다고 가정한다. 그룹 구성원의 집합 U는 클라이언트의 집합 $\{U_i | i = 1, 2, \cdots, n-1\}$와 서버의 집합 $\{U_n\}$의 합집합이고, 제안한 프로토콜의 자세한 실행 과정은 다음과 같다.

단계 1. 1라운드에서, 각각의 클라이언트 $U_i \neq U_n$는 임의의 $r_i \in_R Z_q^*$를 뽑아, $z_i = g^{r_i}$을 계산한 다음 각각의 비밀 키 K_i로 서명하여 $s_i = Sign_{K_i}(z_i)$를 얻는다. 그런 다음, 응용(application) 서버 U_n에게 메시지 $m_i = (z_i, s_i)$를 전송한다. 또한 서버는 임의의 $r, r_n \in_R Z_q^*$을 뽑아 $z = g^r$, $x_n = z^{r_n} (= g^{r r_n})$을 계산한다.

단계 2. 2라운드에서, 서버 U_n은 z_i의 서명 s_i를 검증한 다음, 서명 값이 모두 옳다면 $x_i = z_i^r$을 계산한다. 그런 다음, 일회용(nonce)의 l비트, $\delta \in \{0, 1\}^l$ (여기서, l은 안전성 파라미터)를 뽑아

$$X = \oplus_{i=1}^{n} H(\delta \| x_i)$$

를 계산한다. X_i의 집합을 $Y = \{X_i | 1 \le i \le n-1\}$로 놓고, 비밀 키 K_n으로 $\delta \| z \| Y$를 서명하여 $s_n = Sign_{K_n}(\delta \| z \| Y)$를 얻는다. 이 때, $X_i = X \oplus H(\delta \| x_i)$이다. 이제 서버 U_n은 메시지 $m_n = (\delta, z, Y, s_n)$을 전체 그룹 구성원에게 브로드캐스트 한다.

단계 3. 공통키를 계산하는 단계로, 서버 U_n으로부터 브로드캐스트 메시지를 받은 후, 각각의 클라이언트 $U_i \neq U_n$는 서명 s_n을 검증한 다음, U_i는 다음과 같이

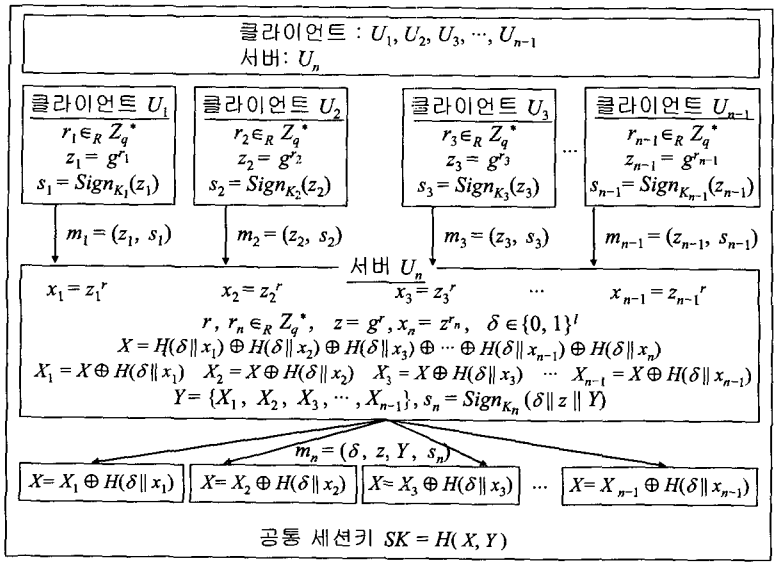
$$X = X_i \oplus H(\delta \| x_i),\ x_i = z^{r_i}$$

를 복구할 수 있다. 이제 서버 U_n을 포함한 모든 클라이언트는 공통의 세션키 $SK = H(X, Y)$를 계산할 수 있다. 여기서 H는 일방향 해쉬 함수이다.

제안한 프로토콜의 실행 과정을 간단히 (그림 1)에 나타내었다. (그림 1)에서 보듯이 이 프로토콜은 $n-1$번의 유니캐스트와 한 번의 브로드캐스트 통신을 필요로 하고, 2라운드 만에 세션키를 나눠 갖게 된다.

3.2 탈퇴 프로토콜(Leave Protocol : P_{leave})

모바일 환경에서는 무선 단말기의 잦은 이동으로 그룹 구성원이 동적으로 변하고, 그에 따라 공통의 새로운 세션키



(그림 1) 초기 설정 프로토콜 P

설정이 필요하다. 이에 우리는 초기 설정 프로토콜에서의 클라이언트 U_2와 U_{n-1}이 그룹에서 탈퇴한다고 가정하자. 탈퇴 프로토콜 P_{leave}의 실행 과정은 다음과 같다.

단계 1. 클라이언트 그룹의 구성원을 갱신하고, 갱신된 구성원의 집합을 U_C라 하면,

$U_C = \{U_i \mid 1 \le i \le n-1\} \setminus \{U_2, U_{n-1}\}$이다.

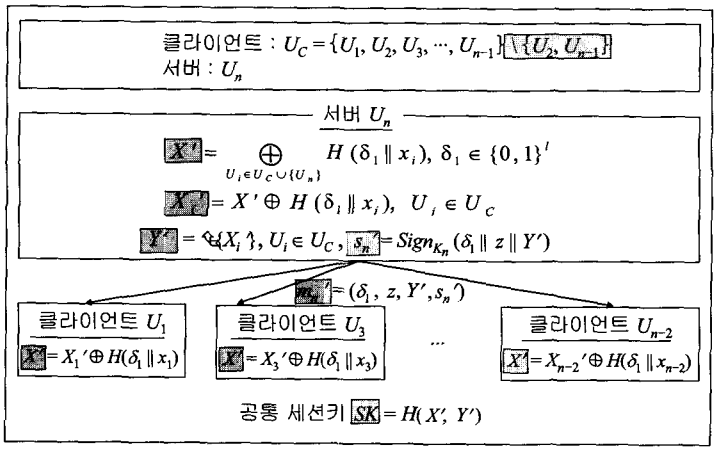
단계 2. 서버 U_n은 각 클라이언트의 x_i 값을 이미 알고 있으므로, 서버 U_n은 새로운 일회용(nonce)의 l비트, $\delta_1 \in \{0,1\}^l$(여기서, l은 안전성 파라미터)을 뽑아 $X' = \oplus_{i=3}^{n-2} H(\delta_1 \| x_i) \oplus H(\delta_1 \| x_1) \oplus H(\delta_1 \| x_n)$을 계산한 후, $Y' = \{X_i' \mid i = 1, 3 \le i \le n-2\}$로 놓고, 서버의 비밀 키 K_n으로 $\delta_1 \| z \| Y'$을 서명하여 $s_n' = Sign_{K_n}(\delta_1 \| z \| Y')$을 얻는다. 이 때, $x_i = z_i^r$이고 $X_i' = X' \oplus H(\delta_1 \| x_i)$이다. 이제 서버 U_n은 메시지 $m_n' = (\delta_1, z, Y', s_n')$을 전체 그룹 구성원에게 브로드캐스트 한다.

단계 3. 서버 U_n으로부터 브로드캐스트 메시지를 받은 후, 각각의 클라이언트 $U_i \ne U_n$는 다음과 같이

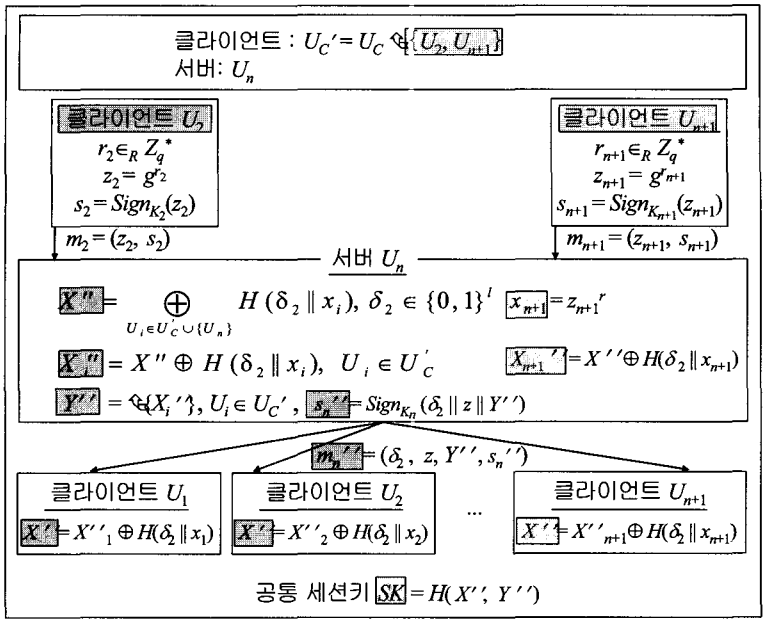
$X' = X_i' \oplus H(\delta_1 \| x_i), \; x_i = z^{r_i}$

을 복구할 수 있다. 이제 서버 U_n을 포함한 모든 클라이언트는 새로운 공통의 세션키 $SK = H(X', Y')$을 계산할 수 있다.

프로토콜 P_{leave}의 실행 과정을 간단히 (그림 2)에 나타내었다. 새로 계산되는 값은 어두운 부분으로 표시하였다. (그림 2)에서 보듯이 클라이언트 쪽에서는 ⊕(XOR) 연산만으로 새로운 공통의 세션키를 계산해 낼 수 있어서, 계산적인 비용 측면에서 매우 효율적이다.



(그림 2) 탈퇴 프로토콜 P_{leave}



(그림 3) 참여 프로토콜 P_{join}

3.3 참여 프로토콜(Join Protocol : P_{join})

그룹을 탈퇴하였던 클라이언트 U_2와 새로운 구성원 U_{n+1}이 그룹에 들어와 세션키 설정에 참여한다고 가정하자. 참여 프로토콜 P_{join}의 실행 과정은 다음과 같다.

단계 1. 클라이언트 그룹의 구성원을 갱신하고, 갱신된 구성원의 집합을 U_C'이라 하면, $U_C' = U_C \cup \{U_2, U_{n+1}\}$이다. 새로운 클라이언트 U_{n+1}은 임의의 $r_{n+1} \in_R Z_q^*$을 뽑아, $z_{n+1} = g^{r_{n+1}}$을 계산한 다음 자신의 비밀 키 K_{n+1}로 서명하여 $s_{n+1} = Sign_{K_{n+1}}(z_{n+1})$을 얻는다. 그런 다음, 응용(application) 서버 U_n에게 메시지 $m_{n+1} = (z_{n+1}, s_{n+1})$을 전송한다. 이전의 그룹 구성원이었던 U_2는 초기 설정 1라운드에 사용했던 동일한 메시지 $m_2 = (z_2, s_2)$를 서버에게 전송하거나, 새로운 값 $r_2' \in_R Z_q^*$을 뽑아, $z_2' = g^{r_2'}$을 계산한 다음 자신의 비밀 키 K_2로 서명하여 $s_2' = Sign_{K_2}(z_2')$을 얻는다. 그런 다음, 응용 서버 U_n에게 메시지 $m_2' = (z_2', s_2')$을 전송한다.

단계 2. 서버는 z_2(또는 z_2')와 z_{n+1}의 서명을 검사한 다음, 서명 값이 옳다면 x_2(또는 $x_2' = (z_2')^r$)와 $x_{n+1} = z_{n+1}^r$을 계산한다. 그런 다음, 서버 U_n은 새로운 $\delta_2 \in \{0,1\}^l$(여기서, l은 안전성 파라미터)를 뽑아 $X'' = \oplus_{i=1}^{n-2} H(\delta_2 \| x_i) \oplus H(\delta_2 \| x_n) \oplus H(\delta_2 \| x_{n+1})$을 계산한 후, $Y'' = \{X_i'' | 1 \le i \le n-2, i = n+1\}$로 놓고, 서버의 비밀 키 K_n으로 $\delta_2 \| z \| Y''$을 서명하여 $s_n'' = Sign_{K_n}(\delta_2 \| z \| Y'')$을 얻는다. 이 때, $x_i = z_i^r$이고 $X_i'' = X'' \oplus H(\delta_2 \| x_i)$이다. 이제 서버 U_n은 메시지 $m_n'' = (\delta_2, z, Y'', s_n'')$을 전체 그룹 구성원에게 브로드캐스트 한다.

단계 3. 서버 U_n으로부터 브로드캐스트 메시지를 받은 후, 각각의 클라이언트 $U_i \neq U_n$는 다음과 같이

$$X'' = X_i'' \oplus H(\delta_2 \| x_i),\ x_i = z^{r_i}$$

를 복구할 수 있다. 이제 서버 U_n을 포함한 모든 클라이언트는 새로운 공통의 세션키 $SK = H(X'', Y'')$을 계산할 수 있다.

프로토콜 P_{join}의 실행 과정을 간단히 (그림 3)에 나타내었다. 새로 계산되는 값은 P_{leave}와 마찬가지로, 어두운 부분으로 표시하였다. (그림 3)에서 보듯이 새로운 구성원 U_{n+1}이 초기 설정 프로토콜을 따라 각 클라이언트가 1라운드에 한 동일한 작업을 제외하면 프로토콜 P_{join}과 P_{leave}는 유사하다.

4. 제안한 키 동의 방식의 효율성과 안전성 분석

4.1 효율성 분석

제안한 키 동의 방식의 효율성을 통신과 계산 비용 측면에서 고려해 본다.

4.1.1 통신 복잡도(communication complexity)

제안한 키 동의 방식은 (그림 1)에서 보는 바와 같이 초기에 세션키를 설정하기 위하여 첫 라운드에 $n-1$번의 유니캐스트 통신을 필요로 하고, 둘째 라운드에서는 단 한 번

의 브로드캐스트 통신만을 필요로 한다. 따라서 2라운드의 통신만으로 제안한 프로토콜이 수행되고, 필요한 총 메시지 수는 n으로 최적이다[16]. 그리고 탈퇴 프로토콜(P_{leave})에서는 한 번의 브로드캐스트 통신만을 필요로 하고, 참여 프로토콜(P_{join})에서는 새로이 참여하는 구성원의 수만큼의 유니캐스트와 한 번의 브로드캐스트 통신을 필요로 한다.

4.1.2 계산 복잡도(computational complexity)

제안한 키 동의 방식에서 각각의 클라이언트 U_i는 한 번의 서명 생성과 군 $\mathbb{G}$에서 2번($z_i = g^{r_i}$, $x_i = z^{r_i}$)의 모듈러 지수승을 계산해야 한다. 그리고 서버 U_n은 각 클라이언트를 인증하기 위하여 $n-1$번의 서명을 검증하고, $n+1$번(각 클라이언트의 x_i, $n-1$번과 자신의 z, x_n 계산에 각 한 번씩 2번)의 모듈러 지수승을 계산해야 한다. 그리고 탈퇴 프로토콜(P_{leave})에서 기존의 클라이언트와 서버는 새로운 세션키를 계산하기 위하여 이미 이전에 계산된 모듈러 지수승을 사용하므로 해쉬 연산과 XOR 연산만이 필요하다. 또 참여 프로토콜(P_{join})에서는 새로이 프로토콜에 참여하는 클라이언트만 2번의 모듈러 지수승이 필요하고, 서버는 참여 클라이언트의 수만큼의 모듈러 지수승이 필요하다.

4.1.3 제안한 키 동의 방식과 기존 방식의 효율성 비교

제안한 키 동의 방식은 ING[9]와 BD[10] 프로토콜보다 통신 비용과 계산 비용 측면에서 훨씬 효율적이고, Bresson 등[6]의 프로토콜과는 성능이 비슷하다. 그러나 Bresson 등의 프로토콜에서는 서버의 장기간 사용하는 키 r이 노출되면 모든 x_i가 z_i와 r로부터 쉽게 계산되어 모든 세션키가 복구될 수 있으므로 전방향 안전성(forward secrecy)을 달성할 수 없다[6].

제안한 키 동의 방식과 기존 방식의 복잡도를 비교하여 <표 2>에 나타내었다. Steiner 등의 IKA.2[12]에서는 라운드 수가 구성원의 수에 따라 선형적으로 증가하므로 통신 비용 측면에서 매우 비효율적이다. 또한 그룹 구성원의 변화 시, 현재의 그룹 제어자 $U_c(c \in [1, n])$가 마지막 라운드에서 전송한 메시지의 내용을 저장하고 있다고 가정하여 모든 구성원이 그룹 제어자(서버 역할)가 될 수 있으나, 이 IKA.2에서는 그룹에 새로운 구성원(또는 그룹을 탈퇴했다가 다시 참여하는 구성원)이 참여할 때 계산 복잡도가 커서 비효율적이다.

그리고 Bresson 등의 프로토콜에서는 구성원의 탈퇴나 재참여, 또는 새로운 구성원의 추가에서 이전의 카운터와 현재 카운터의 값을 비교해야 하는데, 현재 실행 중인 프로토콜의 카운터를 구분하기 어려운 약점이 있다. 또한 [6]에서 제안한 방식은 서버가 공유 비밀 데이터 K 값을 프로토콜의 실행대로 계산하지 않고 임의로 선택할 수 있어서 키 제어가 쉬우므로 약한 키 동의 프로토콜이다. 그리고 제안한 프로토콜에서도 Bresson 등의 프로토콜에서와 마찬가지로 카운터를 사용하는 것도 가능하나, 동기를 맞춰야 하는 제약 사항이 따르므로 그룹 탈퇴나 참여의 경우에 서버에서 새로운 임의의 수를 선택하여 바꿔 줌으로써 이를 극복하였다. 또한 제안한 키 동의 방식은 서버를 인증하기 위해 2라운드에서도 전자 서명을 사용하므로 Bresson 등의 프로토콜보다 클라이언트마다 한 번의 서명 검증이 더 필요하고, 1번의 모듈러 지수승을 사전에 계산할 수 없으며, 전송하는 메시지의 크기는 Bresson 등의 프로토콜과 비교하여 z값을 추가하는 정도로 안전성 요구 조건(전방향 안전성)을 만족한다.

<표 2> 제안한 키 동의 방식과 기존 방식의 복잡도 비교

회수		Steiner 등의 IKA.2[12]	Bresson 등[6]의 프로토콜	제안한 프로토콜
초기	라운드	$n+1$	2	2
	유니캐스트	$2n-3$	$2(n-1)$	$n-1$
	브로드캐스트	2	0	1
	평균 Exp	3(U_n은 $O(n)$)	2(U_n은 $O(n)$)	2(U_n은 $O(n)$)
탈퇴	라운드	1	1	1
	유니캐스트	0	m	0
	브로드캐스트	1	0	1
	평균 Exp	1(U_c는 $(n-m)$)	0	0
참여	라운드	$m+1$	2	2
	유니캐스트	m	$2m+n-1$	m
	브로드캐스트	1	0	1
	평균 Exp	$U_i(i \in [1, n])$는 1, U_N은 2, U_{n+m}은 $(n+m)$	U_N만 2 (U_n은 $O(m)$)	U_N만 2(U_n은 $O(m)$)

(단, n은 초기 구성원의 수, m은 탈퇴하는 구성원(또는 다시 참여하는 구성원)의 수, U_c는 서버 역할의 구성원, U_N은 기존 프로토콜에 새로 참여하는 구성원, Exp는 군 $\mathbb{G}$에서의 모듈러 지수승을 나타낸다.)

4.2 안전성 분석

제안한 키 동의 방식의 안전성 분석은 여러 가지 문헌([13-15, 17-19])에서 널리 쓰이고 있는 표준적인 안전성 모델을 사용하여 수동적인 공격자에 대하여 안전함을 증명한다.

또한 서버와 클라이언트의 상호 인증은 양쪽 모두 전자 서명을 사용하여 검증하므로, 제안한 프로토콜은 능동적 공격자에 대하여 안전하다.

4.2.1 전방향 안전성(Forward Secrecy)

전방향 안전성(forward secrecy)이란 인증에 사용된 장기간 사용하는 비밀 키의 노출이 이전에 설정된 세션키의 안전성을 손상시키지 않음을 의미한다. Bresson 등의 프로토

콜에서는 $z(=g^r)$이 서버의 공개키로 그 값을 공개하는데, 서버의 인증에 필요한 이 r 값을 장기간 사용하는 비밀 키로 볼 수 있는 반면, 제안한 프로토콜에서는 서명키 $K_i(\in [1,n])$를 장기간 사용하는 비밀 키로 볼 수 있다. 또한 저자들이 밝힌 것처럼 r이 노출되면 모든 x_i가 z_i와 r로부터 쉽게 계산되어[6], r이 노출되는 세션 뿐만 아니라 그 이전의 모든 세션키가 복구될 수 있으므로 전방향 안전성을 달성할 수 없다. 그러나 제안한 프로토콜에서는 비밀 키인 K_i가 노출되어도 세션키가 노출되지 않는 것은 자명하므로 전방향 안전성을 달성한다.

동적인 그룹키 동의 프로토콜에서는 그룹 구성원 간의 안전한 통신을 하기 위하여 한 세션 내에 초기 세션키 설정 프로토콜(P)이 실행되고 이후에는 그룹 구성원이 동적으로 변경됨에 따라 탈퇴 프로토콜(P_{leave})이나 참여 프로토콜(P_{join})이 반복적으로 사용될 수 있다. 예를 들어 다음과 같은 순서로 n개의 세션이 실행되는 시나리오를 고려해 보자.

세션(session) 1 :
$P_1 - P_{leave-11} - P_{leave-12} - P_{join-11} - P_{leave-13} - \cdots$
세션(session) 2 :
$P_2 - P_{join-21} - P_{leave-21} - P_{leave-22} - P_{join-22} - \cdots$
$\vdots$
세션(session) n : $P_n - P_{leave-n1} - P_{join-n1} - \cdots$

위의 세션 n의 초기 세션키 생성 프로토콜 P_n에서 r 값이 노출된다고 가정하면, Bresson 등의 프로토콜에서는 n개의 세션 전체 프로토콜에 대하여 r 값을 사용하므로 모든 세션에서 모든 프로토콜의 세션키가 복구될 수 있는 반면, 제안한 프로토콜에서는 매 세션마다 r 값을 다시 선택하므로 가장 최근에 실행된 초기 세션키 프로토콜 P_n 이후부터의 세션키만이 복구되고, 그 이전에 실행된 모든 세션(세션 1에서 세션 $n-1$까지)의 세션키는 노출되지 않는다.

4.2.2 모델(model)

- **프로토콜 참가자(participant)** : 그룹 키 동의 프로토콜 P에 참가하는 구성원의 집합들로, 각각의 구성원은 오라클(oracle)이라고 부르는 인스턴스(예를 들면 Π_i^s는 구성원 U_i의 s번째 인스턴스를 나타낸다)를 가질 수 있다.
- **파트너(partner)** : 오라클 Π_i^s의 파트너(PID_i^s)는 한 번의 프로토콜 실행에서 Π_i^s와 동일한 세션키를 계산해야만 하는 모든 인스턴스의 집합이고, SID_i^s는 오라클 Π_i^s의 세션 ID를 나타낸다.
- **수동적 공격자(passive adversary)** : 수동적 공격자는 네트워크 상에서 모든 통신 흐름을 제어할 수 있고, 다음과 같은 4가지 질의(query), 즉 실행, 유출, 손상, 테스트 질의를 통해 참가자들과 상호 작용한다.
- **실행 질의 Execute(U)** : 이 질의에 대한 응답으로 프로토콜 참가자의 인스턴스 중에서 정직한 프로토콜 실행으로 얻어지는 전달 메시지(transcript)를 돌려받는다.
- **유출 질의 Reveal(Π_i^s)** : 이 질의는 세션키를 획득하기 위한 공격자의 능력을 모델화한 것으로 오라클 Π_i^s가 계산한 세션키 값 SK를 돌려받는다.
- **손상 질의 Corrupt(U_i)** : 전방향 안전성(forward secrecy)을 다루기 위하여 고려한 질의로, 참가자 U_i의 장기간 사용하는 비밀 키를 돌려받는다.
- **테스트 질의 Test(Π_i^s)** : 이 질의는 세션키의 의미론적(semantic) 안전성을 모델화 한 것으로, 공격자가 실제 세션키와 임의의 위조키를 구분하기 원할 때 단 한 번만, 그리고 fresh한 오라클 Π_i^s에 대해서만 요청할 수 있다. 동전 b를 던져, $b=1$이면 실제 세션키 SK를 돌려받고, $b=0$이면 세션키 길이만큼의 임의의 스트링을 돌려받는다.

4.2.3 전방향 안전성과 Freshness

freshness 개념은 공격자에게 명백히 세션키가 알려져 있지 않음을 의미한다[20]. 즉 오라클 Π_i^s가 fresh하려면, 공격자가 PID_i^s에 속하는 어느 파트너에게도 유출 질의를 하지 않았고, 또한 Π_i^s가 NULL 값이 아닌 세션키 값을 계산하였으며, 공격자가 U에 속하는 어느 참가자에게도 손상 질의를 하지 않았어야 한다[13]. 본 논문에서는 전방향 안전성에 대해서도 고려하지만, 만약 전방향 안전성이 요구되지 않는 경우라면 freshness의 정의에서 참가자 U_i의 장기간 사용하는 비밀 키를 돌려받는 손상 질의에 대한 조건은 필요하지 않다.

4.2.4 계산적 Diffie-Hellman(CDH) 문제

제안한 프로토콜이 기반하고 있는 계산적(computational) Diffie-Hellman(CDH) 문제는 적당한 $a, b \in Z_q^*$에 대하여 (g, g^a, g^b)를 인스턴스(instance)로 받아 $g^{ab} \bmod q$ 값을 계산하여 해(solution)로 출력하는 문제이다. 이제 CDH 문제를 해결하기 위하여, 임의의 확률적 다항식 시간 안에, g^{ab} 값을 출력함에 있어 알고리즘 A의 이익(advantage)은 다음과 같이 정의할 수 있다.

$$Adv_{\mathbb{G}}^{\mathrm{CDH}}(A) = |P[g^{ab} \leftarrow A(\mathbb{G}, g, g^a, g^b) \mid g \in \mathbb{G}; a, b \in_R Z_q^*]|$$

이 때, CDH 가정은 모든 확률적 다항식 시간 안에, 알고리즘 A에 대하여, $Adv_{\mathbb{G}}^{\mathrm{CDH}}(A)$의 값이 무시할 수 있을 만큼(negligible) 작음을 나타낸다. 즉, (g, g^a, g^b) 값이 주어진다 하더라도 g^{ab} 값을 계산할 수 없음을 의미한다. 그리고 $Adv_{\mathbb{G}}^{\mathrm{CDH}}(t)$는 기껏해야 t시간 내에 모든 공격자에 의해 실행

되는 알고리즘 A의 이익 $Adv_{\mathbb{G}}^{\mathrm{CDH}}(A)$의 최대값을 나타낸다.

4.2.5 그룹 키 동의 프로토콜의 안전성 정의

공격자의 알고리즘 A가 그룹 키 동의 프로토콜 P의 실행 중에 fresh한 오라클에게 테스트 질의를 요청하면 이 질의에 대한 응답으로 임의의 l비트의 스트링을 돌려받고, 이후에 숨겨진 비트 b에 대한 추측으로 비트 b'을 출력한다. CG(Correct Guess)를 $b=b'$인 사건이라 하면 프로토콜 P를 공격함에 있어 공격자 알고리즘 A의 이익(advantage)은 다음과 같이 정의한다.

$$Adv_{A,P}(k)=2\cdot \Pr[CG]-1$$

$Adv_{A,P}(k)$의 값이 무시할 만하면 프로토콜 P는 공격자 알고리즘 A에 대하여 안전하다고 말한다.

4.2.6 수동적 공격자에 대한 안전성 증명

[정리 1] A를 시간 t 내에 랜덤 오라클 H의 최대 q_h개의 질의와 q_{ex}개의 실행 질의를 수행하는, 동적 그룹 키 설정 방식을 공격하는 수동적 공격자라고 하자. 그러면 다음이 성립한다.

$$Adv_{A,P}(k)=2q_hq_{ex}\cdot Adv_{\mathbb{G}}^{\mathrm{CDH}}(t'),$$

여기서 $t'=t+O(nq_{ex}t_{\mathrm{Exp}})$이고, t_{Exp}는 $\mathbb{G}$에서 지수승을 계산하는 데 필요로 하는 시간이다.

[증명] A가 $1/2+\epsilon$의 확률로 숨겨진 비트 b를 올바르게 추측할 수 있다고 가정하자. 그러면 ϵ/q_hq_{ex}의 확률로 $\mathbb{G}$에서 CDH를 푸는 알고리즘 B를 A로부터 구축한다.

먼저, 다음과 같은 2가지 분포(distribution)를 정의한다.

$$Real=\left\{(T,SK)\;\middle|\;\begin{array}{l} r_1,r_2,\cdots,r_n,r\in_R Z_q^*;\ \delta\in\{0,1\}^l;\\ z_1=g^{r_1},\ z_2=g^{r_2},\ \cdots,\ z_n=g^{r_n},\ z=g^r;\\ x_1=g^{r\,r_1},\ x_2=g^{r\,r_2},\ \cdots,\ x_n=g^{r\,r_n};\\ h_1=H(\delta\|x_1),\ h_2=H(\delta\|x_2),\ \cdots,\ h_n=H(\delta\|x_n);\\ X=\oplus_{i=1}^n h_i;\\ y_1=X\oplus h_1,\ y_2=X\oplus h_2,\ \cdots,\ y_{n-1}=X\oplus h_{n-1}\end{array}\right\},$$

$$Rand=\left\{(T,SK)\;\middle|\;\begin{array}{l} r_1,r_2,\cdots,r_n,r\in_R Z_q^*;\\ \delta,\ w_1,w_2,\cdots,w_n\in\{0,1\}^l;\\ z_1=g^{r_1},\ z_2=g^{r_2},\ \cdots,\ z_n=g^{r_n},\ z=g^r;\\ h_1=w_1,\ h_2=w_2,\ \cdots,\ h_n=w_n;\\ X=\oplus_{i=1}^n h_i;\\ y_1=X\oplus h_1,\ y_2=X\oplus h_2,\ \cdots,\ y_{n-1}=X\oplus h_{n-1}\end{array}\right\},$$

여기서

$$T=(z,z_1,z_2,\cdots,z_{n-1},\delta,y_1,y_2,\cdots,y_{n-1})$$

이고 $SK=H(y_1,y_2,\cdots,y_n,X)$이다.

[도움정리 1] 두 가지 분포 $Real$과 $Rand$ 중의 하나에서 나온 (T,SK)가 주어질 때, A'을 시간 t 내에 0 또는 1의 값을 출력하는 알고리즘이라 하자. 그러면 다음이 성립한다.

$$\begin{aligned}&|P[A'(T,SK)=1\,|\,(T,SK)\leftarrow Real]\\&\quad-P[A'(T,SK)=1\,|\,(T,SK)\leftarrow Rand]|\\&\qquad\le\frac{1}{q_h}Adv_{\mathbb{G}}^{\mathrm{CDH}}(t+2nt_{\mathrm{Exp}})\end{aligned}$$

[증명] 알고리즘 A'이 두 가지 분포를 무시할 수 없는 확률로 구분한다고 가정하자. 그러면 H가 랜덤 오라클이고 $Real$과 $Rand$ 분포에서는 $h_i(i\in[1,n])$를 계산하는 방식만 차이가 있으므로 두 분포를 구분한다는 것은 적어도 하나의 x_i 값을 구할 수 있음을 의미한다. 이제 우리는 입력 $(g,A=g^r,B=g^\alpha)\in\mathbb{G}^3$이 주어졌을 때, $r\alpha=\beta \bmod q$인 $C(=g^\beta)$ 값을 출력하는 알고리즘을 다음과 같이 구성한다.

먼저 임의의 $\gamma_i\in_R Z_q^*$를 선택하여, 지수를 $r_i=\alpha+\gamma_i \bmod q$로 정의하면 $z_i=Bg^{\gamma_i}$로 계산할 수 있다. 그리고 l비트의 랜덤한 $h_i\in\{0,1\}^l$로 $X=\oplus_{i=1}^n h_i$를 계산하여, $y_i=X\oplus h_i$를 구성할 수 있다. 즉 다음과 같은 분포를 생각해 보자.

$$Simul=\left\{(T,SK)\;\middle|\;\begin{array}{l}\gamma_1,\gamma_2,\cdots,\gamma_n,x_i'\in_R Z_q^*;\\ \delta,h_1,h_2,\cdots,h_n\in\{0,1\}^l;\\ r_1=\alpha+\gamma_1,\ r_2=\alpha+\gamma_2,\ \cdots,\ r_n=\alpha+\gamma_n;\\ z_1=Bg^{\gamma_1},\ z_2=Bg^{\gamma_2},\ \cdots,z_n=Bg^{\gamma_n};\\ X=\oplus_{i=1}^n h_i;\\ y_1=X\oplus h_1,\ y_2=X\oplus h_2,\ \cdots,\ y_{n-1}=X\oplus h_{n-1};\end{array}\right\},$$

여기서 T와 SK는 위에서 정의한 것과 같다. 이 구성으로부터 모든 $i\in[1,n]$에 대하여 $z_i=g^{r_i}(=Bg^{\gamma_i})$이므로 $Rand\equiv Simul$이 성립한다.

우리는 분포 $Simul$에서 선택된 (T,SK)를 A'의 입력 값으로 제공하면서 동시에 랜덤 오라클 H를 시뮬레이트 한다. 최종적으로 A'이 실행을 종료할 때 랜덤 오라클 시뮬레이션 테이블에서 입력이 $\delta\|x_i'$ 형태인 것 중의 하나를 임의로 선택한다. $x_i'=x_i$인 경우 $x_i=CA^{\gamma_i}$이므로 $C=x_i'(A^{\gamma_i})^{-1}$을 계산할 수 있어 CDH 문제를 해결할 수 있게 된다. 따라서 알고리즘 A'은 두 가지 분포를 구분할 수 없다. □

[도움정리 2] 계산적인 능력이 무한한 임의의 공격자 A에 대하여, 다음이 성립한다.

$$\begin{aligned}&P[A(T,SK_0)=b\,|\,(T,SK_1)\leftarrow Rand;\\&\qquad SK_0\leftarrow\{0,1\}^l;b\leftarrow 0,1]=1/2\end{aligned}$$

[증명] $Rand$ 실험에서, 전달 메시지 T로부터 $y_i\ (i\in[1,n-1])$를 다음과 같이 나타낼 수 있다.

$$\begin{aligned}y_1&=h_2\oplus h_3\oplus\cdots\oplus h_n=h_1\oplus h_n\oplus y_n\\ y_2&=h_1\oplus h_3\oplus\cdots\oplus h_n=h_2\oplus h_n\oplus y_n\\&\vdots\\ y_{n-1}&=h_1\oplus h_2\oplus\cdots\oplus h_{n-2}\oplus h_n=h_{n-1}\oplus h_n\oplus y_n\end{aligned}$$

즉 위의 식을 만족하는 해 $(h_1, h_2, \cdots, h_n)$의 형태는 다음과 같이 고쳐 쓸 수 있다.

$$\begin{aligned} h_1 &= y_1 \oplus y_n \oplus h_n \\ h_2 &= y_2 \oplus y_n \oplus h_n \\ &\vdots \\ h_{n-1} &= y_{n-1} \oplus y_n \oplus h_n \\ h_n & \end{aligned}$$

따라서 해의 개수는 주어진 독립변수 h_n 값이 취할 수 있는 집합의 크기인 2^l만큼의 해가 존재하므로 개별적인 메시지 정보로부터 공격자는 X에 대한 어떤 정보도 얻지 못한다. 즉 다음 식

$$\begin{aligned} &\Pr[A(T, X_b) = b \mid (T, X_1) \leftarrow Rand; X_0 \leftarrow \{0,1\}^l; b \leftarrow \{0,1\}] \\ &= \frac{1}{2} \end{aligned}$$

이 성립하며, H가 랜덤 오라클이므로 도움정리 2가 성립한다. □

이제 위의 [도움정리 1, 2]를 가지고, 분포 $Simul$을 구성한 알고리즘 B를 자세히 설명한다. 공격자 A가 δ번째의 실행 질의에 의해 활성화된 오라클에 테스트 질의를 한다고 가정하자. 먼저 알고리즘은 δ의 추측 값으로 임의의 $d \in \{1, 2, \cdots, q_{ex}\}$를 선택한다. 그런 다음 A를 호출하고 A의 질의를 시뮬레이트 한다. 알고리즘은 d번째 질의를 제외하고, A의 모든 질의에 대하여 프로토콜에 정확히 명시된 대로 응답한다. 공격자 A가 d번째 질의를 한 경우, 알고리즘은 $Simul$에 따라 (T, SK)를 생성하여, A의 d번째 실행 질의에 응답한다.

알고리즘은 $d \neq \delta$이면 $\mathbb{G}$에서 선택된 임의의 원소를 출력한다. 그렇지 않으면, SK를 가지고 A의 테스트 질의에 응답한다. 나중에, A가 추측 값 b'을 출력하고 끝낸다. 그런데 $\Pr[b = b'] = 1/2$이고 $\Pr[d = \delta] = 1/q_{ex}$이므로 도움정리 1, 2에 의해,

$$\begin{aligned} &\Pr[A(T, SK_b) = b \mid (T, SK_1) \leftarrow Real; SK_0 \leftarrow \{0,1\}^l; b \leftarrow \{0,1\}] \\ &\quad = \frac{1}{2} + \epsilon, \\ &Adv_S^{CDH}(B) = \frac{\epsilon}{q_h q_{ex}} \end{aligned}$$

따라서 [정리 1]이 성립한다. □

5. 결론 및 향후 연구 방향

본 논문에서는 계산적 Diffie-Hellman 가정에 기반하여, 클라이언트 쪽의 계산 효율성이 높은 2라운드의 키 동의 프로토콜을 제안하였다. 제안한 프로토콜은 Emmanuel Bresson 등[6]의 아이디어와 Junghyun Nam 등[7, 8]의 프로토콜을 결합하였고, 이동성이 잦은 모바일 장치와 서버 환경에 적당하다. Junghyun Nam 등의 프로토콜에서의 곱셈 연산 대신 해쉬 함수와 XOR 연산으로 대체하여 계산적인 비용 측면에서 훨씬 효율적이고, Emmanuel Bresson 등의 프로토콜에서의 카운터 대신 일회용의 랜덤한 비트를 추가하여 키 freshness를 달성하였다. 또한 그룹 탈퇴나 참여의 경우에도 빠른 계산이 가능하고, 전방향 안전성도 제공한다. 그리고 제안한 프로토콜은 랜덤 오라클 모델에서 CDH 가정에 기반하여 수동적인 공격자에 대하여 안전함을 증명하였다.

앞으로의 연구는 제안한 프로토콜을 능동적인 공격자에 대하여 안전함을 증명하는 것이 필요하다.

참 고 문 헌

[1] B. Bhargava, M. Annamalai, and E. Pitoura, "Digital Library Services in Mobile Computing," ACM SIGMOD Record, Vol.24, No.4, pp.34-39, December, 1995.

[2] Y. Huang and H. Garcia-Molina, "Publish/Subscribe in a Mobile Environment," Proc. of the 2nd ACM International Workshop on Data Engineering for Wireless and Mobile Access(MobiDE 2001), pp.27-34, 2001.

[3] T. Phan, L. Huang, and C. Dulan, "Challenge: Integrating Mobile Wireless Devices into the Computational Grid," Proc. of the 8th ACM Conference on Mobile Computing and Networking(MOBICOM 2002), pp.271-278, September, 2002.

[4] S.-H. Lim and J.-H. Kim, "Real-time Broadcast Algorithm for Mobile Computing," The Journal of Systems and Software, Vol.69, No.2, pp.173-181, 2004.

[5] W. Diffie and M. E. Hellman, "New Directions in Cryptography," IEEE Transactions on Information Theory, Vol.IT-22, No.6, pp.644-654, 1976.

[6] E. Bresson, O. Chevassut, A. Essiari, and D. Pointcheval, "Mutual Authentication and Group Key Agreement for Low-Power Mobile Devices(Full version)," Proc. of the 5th IEEE International Conference on Mobile and Wireless Communications Networks(MWCN 2003), pp.59-62, World Scientific Publishing, 2003.

[7] Junghyun Nam, Sungduk Kim, Seungjoo Kim, and Dongho Won, "Dynamic Group Key Exchange over High Delay Networks," Proc. of the International Scientific-Practical Conference on Communication (ISPC COMM 2004), pp.22-29, 2004.

[8] Junghyun Nam, Seokhyang Cho, Seungjoo Kim, and Dongho Won, "Simple and Efficient Group Key Agreement based on Factoring," Proc. of the 2004 Interna-

tional Conference on Computational Science and Its Applications(ICCSA 2004), LNCS 3043, pp.645-654, May, 2004.

[9] I. Ingemarsson, D. T. Tang, and C. K. Wong, "A Conference Key Distribution System," IEEE Transactions on Information Theory, Vol.IT-28, No.5, pp.714-720, September, 1982.

[10] M. Burmester and Y. Desmedt, "A Secure and Efficient Conference Key Distribution System," Advances in Cryptology, Eurocrypt 1994, LNCS 950, pp.275-286, 1994.

[11] M. Steiner, G. Tsudik, and M. Waidner, "Diffie-Hellman Key Distribution Extended to Group Communication," Proc. of the 3rd ACM Conference on Computer and Communication Security(CCS 1996), pp.31-37, March, 1996.

[12] M. Steiner, G. Tsudik, and M. Waidner, "Key Agreement in Dynamic Peer Groups," IEEE Transactions on Parallel and Distributed Systems, Vol.11, No.8, pp.769-780, August, 2000.

[13] E. Bresson, O. Chevassut, D. Pointcheval, and J.-J. Quisquater, "Provably Authenticated Group Diffie-Hellman Key Exchange," Proc. of the 8th ACM Conference on Computer and Communication Security(CCS 2001), pp. 255-264, 2001.

[14] E. Bresson, O. Chevassut, and D. Pointcheval, "Provably Authenticated Diffie-Hellman Key Exchange-The Dynamic Case," ASIACRYPT 2001, LNCS 2248, pp.290-309, 2001.

[15] E. Bresson, O. Chevassut, and D. Pointcheval, "Dynamic Group Diffie-Hellman Key Exchange under Standard Assumptions," EUROCRYPT 2002, LNCS 2332, pp.321-336, 2002.

[16] K. Becker and U. Wille, "Communication Complexity of Group Key Distribution," Proc. of the 5th ACM Conference on Computer and Communication Security (CCS 1998), pp.1-6, 1998.

[17] E. Bresson, O. Chevassut, and D. Pointcheval, "Group Diffie-Hellman Key Exchange Secure against Dictionary Attacks," ASIACRYPT 2002, LNCS 2501, pp.497-514, 2002.

[18] J. Katz and M. Yung, "Scalable Protocols for Authenticated Group Key Exchange," CRYPTO 2003, LNCS 2729, pp.110-125, 2003.

[19] C. Boyd and J. M. G. Nieto, "Round-Optimal Contributory Conference Key Agreement," PKC 2003, LNCS 2567, pp.161-174, 2003.

[20] W. Diffie, P. Oorschot, and M. Wiener, "Authentication and Authenticated Key Exchanges", Designs, Codes, and Cryptography, Vol.2, No.2, pp.107-125, 1992.

조 석 향

e-mail : shcho@dosan.skku.ac.kr

1986년 이화여자대학교 수학과(이학사)

1986년~1998년 (주)중앙교육진흥연구소

2001년 서울산업대학교 산업대학원 전자계산학과(공학석사)

2001년~현재 성균관대학교 정보통신공학부 박사과정

관심분야 : 암호 프로토콜, 암호 이론, 정보 보안

남 정 현

e-mail : jhnam@dosan.skku.ac.kr

1997년 성균관대학교 정보공학과(공학사)

2002년 Computer Science, University of Louisiana, Lafayette(M.S.)

2003년~현재 성균관대학교 정보통신공학부 박사과정

관심분야 : 암호 프로토콜, 암호 이론, 네트워크 보안

김 승 주

e-mail : skim@ece.skku.ac.kr

1994년 성균관대학교 정보공학과(공학사)

1996년 성균관대학교 대학원 정보공학과(공학석사)

1999년 성균관대학교 대학원 정보공학과(공학박사)

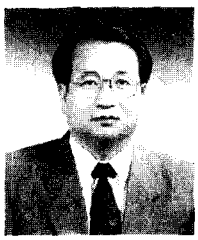
1998년~2004년 한국정보보호진흥원(KISA) 팀장

2004년~현재 성균관대학교 정보통신공학부 교수

2001년~현재 한국정보보호학회 논문지 편집위원

2002년~현재 한국정보통신기술협회(TTA) IT 국제표준화 전문가

관심분야 : 암호이론, 정보보호표준, 정보보호제품 및 스마트카드 보안성 평가, PET



원 동 호

e-mail : dhwon@dosan.skku.ac.kr.

성균관대학교 전자공학과(학사, 석사, 박사)

1978년~1980년 한국전자통신연구소 전임연구원

1985년~1986년 일본 동경공대 객원연구원

1995년~1997년 성균관대학교 교학처장

1996년~1997년 국무총리실 정보화추진위원회 자문위원

1999년~2001년 성균관대학교 전기전자 및 컴퓨터공학부 학부장

1999년~2001년 성균관대학교 정보통신대학원 원장

2002년 한국정보보호학회 회장

1982년~현재 성균관대학교 정보통신공학부 교수

2000년~현재 정통부 지정 정보보호인증기술연구소 소장

관심분야 : 암호 이론, 정보 보안

이 혜 주

e-mail : hyejoo@etri.re.kr

1994년 부경대학교 전자계산학과(이학사)

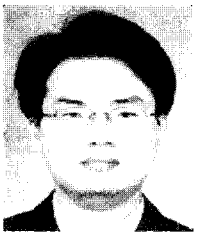
1997년 부경대학교 대학원 전자계산학과(이학석사)

2000년 부경대학교 대학원 전자계산학과(이학박사)

2000년~2001년 한국정보통신대학원대학교 박사 후 연구과정생

2001년~2005년 한국전자통신연구원 방송미디어연구그룹 선임연구원

관심분야 : 디지털 콘텐츠 보호 및 관리, 워터마킹, 멀티미디어 처리 기술



최 진 수

e-mail : jschoi@etri.re.kr

1990년 경북대학교 공과대학 전자공학과(공학사)

1992년 경북대학교 대학원 전자공학과(공학석사)

1996년 경북대학교 대학원 전자공학과(공학박사)

1996년~현재 한국전자통신연구원 선임연구원/데이터방송연구팀장

관심분야 : 영상통신, 멀티미디어 데이터방송