Journal of The Korea Society of Computer and Information
Vol. 23 No. 2, pp. 53-61, February 2018
www.ksci.re.kr
https://doi.org/10.9708/jksci.2018.23.02.053

An Efficient Group Key Agreement Using Hierarchical Key Tree in Mobile Environment

Seokhyang Cho*

Abstract

In this paper, the author proposes an efficient group key agreement scheme in a mobile environment where group members frequently join and leave. This protocol consists of basic protocols and general ones and is expected to be suitable for communications between a mobile device with limited computing capability and a key distributing center (or base station) with sufficient computing capability. Compared with other schemes, the performance of the proposed protocol is a bit more efficient in the aspects of the overall cost for both communication and computation where the computational efficiency of the scheme is achieved by using exclusive or operations and a one-way hash function. Also, in the aspect of security, it guarantees both forward and backward secrecy based on the computational Diffie-Hellman (CDH) assumption so that secure group communication can be made possible. Furthermore, the author proves its security against a passive adversary in the random oracle model.

▸Keyword: Group key agreement, Hierarchical key tree, Multicast, Computational Diffie-Hellman assumption, Random oracle model

I. Introduction

인터넷 환경에서 모바일 기기를 사용한 전자 거래가 보편화됨에 따라 무선 기술이 일상생활에 널리 퍼지게 되었다. 모바일 컴퓨팅 기술은 모바일 기기의 콘텐츠를 기반으로 로컬 및 원격지 네트워크를 동적으로 구성하여 실시간으로 정보를 공유할 수 있도록 지원하고 있다. 정보를 공유하기 위해 통신하는 당사자들은 충분한 계산 능력을 가진 고정된 서버(애플리케이션 서버 또는 서비스 제공자)와 클라이언트라 부르는 제한된 계산 자원을 가진 모바일 기기로 구성되어 있다. 이러한 계산 능력 측면에서 비대칭적인 모바일 환경은 네트워크를 통한 공동 작업, 다중 사용자 게임, 인터넷 주식 시세, 오디오와 음악 전송, 시청한 프로그램에 따라 요금을 지불하는 유료 TV 프로그램 서비스, 소프트웨어 갱신 등과 같은 많은 애플리케이션에 공통이다[1, 2, 3, 4].

그룹 지향적인 어플리케이션의 발전과 함께 안전한 그룹 통신을 위한 보안 서비스의 필요성도 급속히 증가하고 있다. 특히 모바일 환경에서는 도청, 정보의 위·변조 및 파괴, 데이터의 불법적 사용 등 많은 보안상의 취약점을 가지고 있다. 그래서 정보 보호가 제대로 이루어지지 않을 경우 인가되지 않은 사용자에게 기밀 정보가 노출될 수 있고, 특히 금융·경제적인 정보가 노출될 경우 금전적인 손실의 위험도 존재하게 된다. 따라서 안전한 그룹 통신은 그룹의 모든 구성원이 그룹키라고 불리는 하나의 비밀키를 공유함으로써 효과적으로 이루어질 수 있다.

예를 들어 그룹 내의 한 구성원이 그룹의 나머지 모든 구성원에게 비밀 메시지를 전송한다고 가정하자. 이 때 그룹의 모든 구성원이 하나의 그룹키를 공유하게 되면, 송신자는 그룹키를 사용하여 메시지를 한 번만 암호화해서 전송해도 그룹 내의 모든 수신자들은 송신자가 암호화 시 사용한 그룹키와 같은 키를 가지고 있으므로 이를 사용하여 수신한 메시지를 안전하게 복호화할 수 있게 된다. 따라서 그룹 구성원 간에 그룹키를 안전하고 효율적으로 공유할 수 있는 방법이 필요하며, 이러한 목적으로 설계되는 프로토콜을 그룹 세션키 설정 프로토콜이라고 한다.

• First Author: Seokhyang Cho, Corresponding Author: Seokhyang Cho
*Seokhyang Cho (cshlch@ptu.ac.kr), Dept. of Information and Communication, Pyeongtaek University
• Received: 2018. 01. 15, Revised: 2018. 01. 29, Accepted: 2018. 02. 14.
• This work was supported by Pyeongtaek Univ. Research Grant.

이러한 공통의 세션키는 그룹 구성원들이 공개된 통신망을 통하여 안전한 방법으로 나눠가져야 한다. 세션키를 설정하는 프로토콜은 통신에 참여하는 구성원에게만 알려진 공통의 비밀키를 설정하고 어떤 구성원도 이 키를 미리 결정할 수 없도록 공개 정보를 교환한다. 또한 세션키 설정에서는 통신에 참여했던 구성원이 탈퇴한 경우 탈퇴 이후에 이전에 공유했던 세션키의 일부로부터 탈퇴 이후의 새로운 세션키를 알 수 없도록 순방향 안전성(forward secrecy)을 제공해야 할 뿐만 아니라, 새로운 구성원이 통신에 참여하는 경우 새로이 세션키를 공유하게 되더라도 가입 이전의 세션키를 알지 못하도록 역방향 안전성(backward secrecy)도 만족해야 한다.

키를 설정하는 프로토콜은 세션키를 생성하는 관점에서 키를 전송하는(key transport) 프로토콜과 키 생성에 합의하는(key agreement) 프로토콜로 나눌 수 있다. 키 전송 프로토콜은 구성원 한 명이 세션키를 생성하여 안전하게 다른 구성원들에게 전송하는 방식인 반면, 키 합의 프로토콜은 한 명 이상의 구성원이 공통의 세션키를 생성하는 데 자신의 정보를 제공하는 방식이다. 특히 프로토콜에 참여하는 모든 구성원이 세션키 설정에 자신의 정보를 제공하는 경우를 기여(contributory) 키 합의 프로토콜이라 한다. 그리고 모든 구성원이 동일한 구조를 갖고 메시지를 전송하고 계산을 수행할 때 역할 대칭적(role symmetric)이라 한다. 본 논문은 모바일 기기의 계산량이 적은 비대칭적(asymmetric) 기여 키 합의 프로토콜로, 그룹 구성원이 선택하는 수의 랜덤성이 보장된다면 서로 공모하더라도 세션키 값을 제어할 수 없는 특징을 갖는다.

키 합의는 암호기법의 기본 요소 중의 하나로, 키 합의에 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 합의 프로토콜에 결합시켜 이를 해결하였다.

본 논문에서는 Diffie-Hellman 가정에 기반 하여 효율적인 그룹키 합의 프로토콜을 제안한다. 제안한 프로토콜은 클라이언트 쪽의 계산 효율성이 높고, 그룹 구성원의 탈퇴나 가입 시 계산 비용이 적어 동적 그룹에 적합하다. 제안한 프로토콜에서는 완전한 순방향 안전성(perfect forward secrecy)을 제공하고, 해시함수와 배타적 논리합(xor) 연산을 사용하여 계산적인 효율성을 높였다.

본 논문의 2장에서는 기존의 관련 연구들을 살펴보고, 3장에서는 제안하는 키트리에 적용한 그룹키 합의 프로토콜에 대하여 기본 프로토콜과 기본 프로토콜을 키트리의 서브그룹에 적용하여 일반화한 프로토콜을 설명한다. 4장에서는 기존의 프로토콜과 제안한 프로토콜의 성능을 계산 복잡도와 통신 복잡도 측면에서 비교 분석한다. 5장에서는 보안 고려사항을 살펴본 후, 랜덤 오라클 모델에서 수동적인 공격자에 대하여 안전함을 증명하고, 마지막으로 6장에서 결론을 맺는다.

II. Related Works

1. Yongdae Kim et al's protocol

김용대 등[6]은 계산 비용보다는 통신 비용 측면에서 효율적인 그룹키 합의 프로토콜을 제안하였다. 이 프로토콜은 지연이 높은 네트워크에 적당하면서 네트워크에 장애가 발생하더라도 장애가 발생한 노드를 제외하고 통신이 가능하다. 그 이유는 장애가 발생한 노드를 삭제하고 키 트리를 갱신하면 되기 때문인데, 그룹 구성원의 가입과 탈퇴 연산이 가능한 키 트리를 이용한 그룹키 합의 프로토콜의 특징이기도 하다.

키 트리에 기반한 이 프로토콜은 통신에 참여하는 각 구성원의 비밀키를 모듈러 연산 상의 지수승으로 은닉하여 그룹키를 계산하였다. 그룹키 값은 모든 구성원의 비밀키를 기저 값 α의 지수승(거듭제곱)으로 계산하므로 일방향 해시함수와 xor 연산을 사용하는 다른 프로토콜에 비교한다면, 계산 비용 측면에서의 효율성은 상대적으로 낮다.

그림 1과 같이 7명의 그룹 구성원 $M_i\ (i=1,2,\cdots,7)$로 구성된 키 트리의 루트 노드에 해당하는 그룹키 $K_{<0,0>}$은 다음 식 1과 같이 계산된다. 여기서 노드 $<l,v>$는 트리의 레벨 l에서의 v번째 노드, r_i는 M_i의 랜덤 비밀키, α는 지수승을 하게 될 기저(base) 값을 나타낸다.

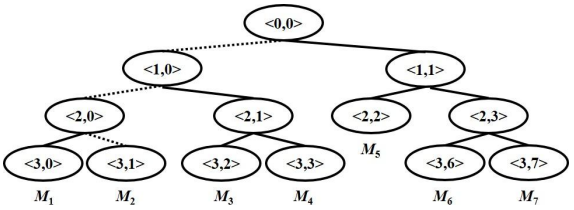

Fig. 1. Example of Yongdae Kim et al's Key Tree

$$K_{<0,0>}=\alpha^{(\alpha^{r_0r_1}\alpha^{r_2r_3})(\alpha^{r_4}\alpha^{r_5r_6})} \quad \text{(식 1)}$$

예를 들어, 그림 1에서 구성원 M_2의 경우 자신이 속한 서브트리의 중간(내부) 노드, 형제 노드의 은닉키(blinded key) $BK_{<3,0>}, BK_{<2,1>}$과 자신이 속하지 않은 서브트리의 은닉키 $BK_{<1,1>}$과 자신의 비밀키 $K_{<3,1>}$을 사용하여 최하위 노드부터 루트 노드까지 자신의 경로 상에 존재하는 키 값 $K_{<2,0>}, K_{<1,0>}, K_{<0,0>}$의 계산이 가능하다. 이 때 은닉키의 값은 $\alpha^{r_i} \bmod p$(p는 매우 큰 소수)로 계산한다.

2. Sangwon Lee et al's protocol

이상원 등[7]은 TGDH(Tree-based Group Diffie-Hellman) 프로토콜을 pairing 기반의 겹선형 사상(bilinear map)으로 변경하여 삼진 키 트리(ternary key tree)에 적용함으로써 통신 복잡도는 TGDH의 성능을 유지하면서 계산상 효율성을 개선한 그룹키 합의 프로토콜을 제안하였다.

그림 2와 같이 7명의 그룹 구성원 $M_i\ (i=1,2,\cdots,7)$로 구

성된 키 트리의 루트 노드에 해당하는 그룹키 $K_{<0,0>}$은 다음 식 2와 같이 계산된다. 여기서 P는 타원곡선 상의 한 점으로 공개된 값이고, H_1은 겹선형 사상의 공역 G_2에서 Z_q^*로의 해시함수, $\hat{e}(P,P)$는 G_2의 생성원소(생성자, generator)이며, r_i는 구성원 M_i의 Z_q^*에 속하는 원소로 랜덤하게 선택된 비밀키 값을 나타낸다.

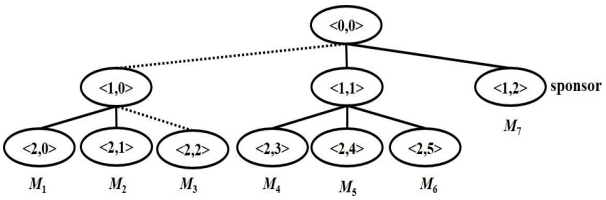

Fig. 2. Example of Sangwon Lee et al's Key Tree

$$K_{<0,0>} = H_1(\hat{e}(H_1(\hat{e}(P,P)^{r_4 r_5 r_6})P, r_7 P)^{H_1(\hat{e}(r_1 P, r_2 P)^{r_3})}) \quad \cdots(\text{식 } 2)$$

예를 들어, 그림 2에서 구성원 M_3은 형제 노드의 은닉키인 $BK_{<2,0>}$, $BK_{<2,1>}$과 M_3이 속하지 않은 부분트리의 은닉키 $BK_{<1,1>}$, 자신의 비밀키 $K_{<2,2>}$를 사용하여 자신의 경로 상에 존재하는 키 값 $K_{<1,0>}, K_{<0,0>}$의 계산이 가능하다. 이 때 은닉키의 값은 $BK_{<l,v>} = K_{<l,v>}P$ 로 계산한다.

3. EHBT: Sandro Rafaeli et al's protocol

Sandro Rafaeli 등[8]은 계층적인 이진 키 트리(EHBT, Efficient Hierarchical Binary Tree)를 사용하여 키 갱신 메시지의 크기를 줄이고, 다른 HBT 프로토콜과 비교하여 저장 공간과 처리 요구사항을 증가시키지 않는 그룹키 관리 프로토콜을 제안하였다.

EHBT 프로토콜에서는 키분배센터(KDC, Key Distribution Centre)가 키 트리를 유지 관리하는 서버 역할을 하고, 그룹 구성원은 클라이언트로서 리프 노드와 연관된 KEK(Key Encryption Key)와 트리의 최하위 노드에서 루트 노드까지 경로 상에 있는 조상들의 KEKs를 갖고 있다.

다른 키로부터 키들을 생성하는 식 3은 다음과 같이 일방향 해시함수 h와 xor 연산을 사용한다. 여기서 h는 연산한 결과를 일방향 함수에 통과시키므로 두 값을 숨기는 역할을 하고, xor 연산은 두 값을 섞어 새로운 값을 생성하는 역할을 한다.

$$F(x, y) = h(x \oplus y) \quad \cdots\cdots(\text{식 } 3)$$

그리고 구성원 M_i가 키 k_i를 새로운 키 k_i'로 변경하거나, 구성원 M_j가 키 k_i를 갱신해야 할 때 현재 키 k_i와 식별자(identifier) 역할을 하는 인덱스(i, index)를 키 생성함수의 입력 정보로 제공하거나 인덱스와 함께 i의 왼쪽 또는 오른쪽 자식 노드의 키를 입력 정보로 제공한다. 즉 식 4와 같이 나타낼 수 있다.

$$k_i' = F(k_i, i) \text{ or } k_i' = F(k_i^{left|right}, i) \quad \cdots\cdots(\text{식 } 4)$$

예를 들어, 한 명의 그룹 구성원 M_1이 이진트리의 리프 노드에 할당되어 있을 때, M_2가 이 그룹 통신에 참여하고자 한다면, KDC는 그림 3의 오른쪽 그림과 같이 트리를 갱신한다. 새로운 구성원 M_2는 리프 노드 n_2에 위치하게 되고 랜덤하게 선택된 키 k_2를 KDC로부터 할당받는다. 또한 노드 n_{12}가 n_1과 n_2의 부모 노드로 트리에 삽입된다. 이 때 다음과 같이 최하위 노드부터 루트 노드까지의 경로에 있는 n_1의 조상 노드 집합에 속하는 키들은 새로운 값으로 변경된다.

$$\begin{cases} k_1' = F(k_1, 1), K_{12} = F(k_1', 2), \\ K_{14}' = F(K_{14}, 14), K_{18}' = F(K_{18}, 18) \end{cases} \quad \cdots\cdots(\text{식 } 5)$$

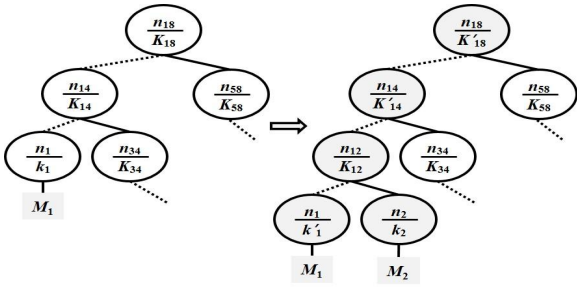

Fig. 3. Example of Sandro Rafaeli et al's Key Tree

앞에서 소개한 프로토콜 이외에도 타원곡선 상의 점을 키 트리에 적용한 그룹키 합의 프로토콜에 대한 연구들[9-11]과 계산적 Diffie-Hellman 가정에 기반하여 비밀값이 생성원소의 지수승이 되는 여러 연구들[12-14]이 진행되어왔다.

본 논문에서는 계산적 Diffie-Hellman 가정에 기반하고 해시함수와 xor 연산을 사용하여 계산 비용 측면에서 효율성을 향상시킴과 동시에, 다른 프로토콜과 비교해 통신 비용 측면에서도 효율적인 그룹키 합의 프로토콜을 제안한다.

III. The Proposed Protocols

1. Basic protocol using ternary key tree

먼저 트리의 깊이가 1일 때, 루트 노드에 해당하는 KDC와 그룹 구성원으로 구성된 기본 프로토콜 BP를 설명한다.

본 논문에서 사용하는 시스템 파라미터는 다음 표1과 같다.

Table 1. Notations

Parameters	Details
$<l, v>$	v-th node at level l in the tree
d	Height of the tree
n	Number of members in the group
m	Number of leaving members in the group
p	Large prime number
g	Generator in the group Z_p^*
G	Subgroup of prime order q in Z_p^*
M_i	i-th group member
xor	Exclusive or operation
PK_i	Signature verification key of M_i
$Sign$	Signing algorithm
H	Hash function
SK	Shared session key among members

L_0, L_1, L_2를 각각 레벨 0, 1, 2에서의 그룹 구성원의 집합이라 하자. 즉 $L_0 = \{S_1\}$, $L_1 = \{M_1, M_2, M_3\}$, $L_2 = \varnothing$, 여기서 S_1은 서버인 키분배센터이고, $M_i\,(i = 1, 2, 3)$는 리프 노드에 해당하는 그룹 구성원이다. 그룹 통신에 참여하는 각 구성원들을 키트리로 나타내면 그림 4와 같다.

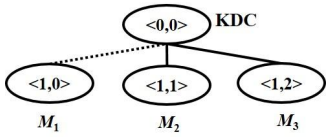

Fig. 4. Example of Basic Protocol's Key Tree

공개 파라미터인 큰 소수 p와 지수승의 기저가 되는 생성원소 g가 통신에 참여하는 모든 그룹 구성원에게 알려져 있다고 가정한다. 제안한 프로토콜의 실행 과정은 다음과 같다.

설정 단계1: 각 구성원 M_i는 랜덤수 r_i를 Z_q^*에서 선택하여, 기저값 g의 지수승을 하여 은닉한 값 $z_i (= g^{r_i} \bmod p)$를 계산한 다음 자신의 비밀키 K_i로 서명하여, $\sigma_i = Sign_{K_i}(z_i)$, 메시지 $m_i = (z_i \| \sigma_i)$를 서버(KDC)에게 보낸다. 이 때 서버 KDC도 자신의 비밀키 랜덤수 r_0, s를 Z_q^*에서 선택하여, $z (= g^s \bmod p)$와 $x_0 = z^{r_0} (= g^{sr_0})$를 계산한다.

설정 단계2: 각 구성원 M_i의 메시지 m_i를 받은 서버는 각 구성원의 공개키 PK_i로 서명 σ_i를 검증한 다음, 각 구성원의 세션키 계산에 필요한 $x_i (= z_i^s \bmod p)$값을 계산한다. 이후 식 6과 같이 통신에 참여하는 그룹 구성원을 식별할 수 있는 ID_i를 포함한 공유 공통 값 X를 계산한다.

$$X = \bigoplus_{i=0}^{k} H(ID_i \| x_i) \,(k\text{는 리프 노드의 수}) \cdots (\text{식 } 6)$$

또한 구성원 M_i에게 필요한 $X_i (= X \oplus H(ID_i \| x_i))$의 집합을 $Y\,(= \{X_1, X_2, X_3\})$라 할 때, 서버의 비밀키 K_s로 서명하여 $\sigma_s (= Sign_{K_s}(I \| z \| Y))$를 얻고, 메시지 $m_s (= (I \| z \| Y \| \sigma_s))$를 그룹 구성원에게 멀티캐스트 한다. 여기서 I는 ID_i의 집합이다.

키 합의 단계: 공통키를 계산하는 단계로, 각각의 그룹 구성원은 서명 σ_s를 검증한 다음, 식 7과 같이 공유 공통 값 X와 x_i를 복구할 수 있다.

$$X = X_i \oplus H(ID_i \| x_i),\ x_i = z^{r_i} \cdots\cdots\cdots\cdots\cdots (\text{식 } 7)$$

이제 서버를 포함한 그룹의 구성원 모두는 일방향 해시함수를 사용하여 공통의 세션키 $SK (= H(X \| Y))$를 계산할 수 있다.

제안한 기본 프로토콜의 실행 과정에서 알 수 있듯이, 서버를 포함한 그룹 구성원이 n명일 때, $n-1$번의 유니캐스트와 한 번의 멀티캐스트로 공통의 세션키를 나눠 갖게 된다.

2. Generalized protocol

그룹 통신에서 기본 프로토콜을 각각의 서브트리에 적용하여 일반화한 프로토콜 GP를 설명한다.

L_0, L_1, L_2를 각각 레벨 0, 1, 2에서의 그룹 구성원의 집합이라 할 때, $L_0 = \{S_0\}$:KDC, $L_1 = \{S_1, S_2, S_3\}$, $L_2 = \{M_1, M_2, \cdots, M_8\}$, 여기서 $S_j\,(j = 1, 2, 3)$는 기지국으로 각 서브트리에서 서버 역할을 하고, $M_i\,(i = 1, 2, \cdots, 8)$는 리프 노드에 해당하는 그룹 구성원이다. 그룹 통신에 참여하는 각 구성원들을 키트리로 나타내면 그림 5와 같다.

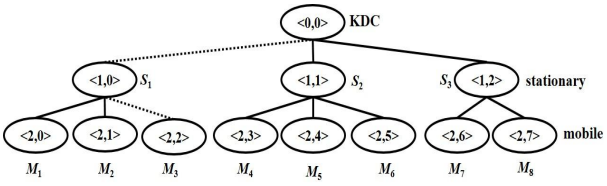

Fig. 5. Example of Generalized Protocol's Key Tree

제안한 프로토콜의 실행 과정은 다음과 같다.

설정 단계1: L_2에 속하는 각 구성원 M_i는 랜덤수 r_i를 Z_q^*에서 선택하여, 기저값 g의 지수승을 하여 은닉한 값 $z_i (= g^{r_i} \bmod p)$를 계산한 다음 자신의 비밀키 K_i로 서명하여, $\sigma_i = Sign_{K_i}(z_i)$, 메시지 $m_i = (z_i \| \sigma_i)$를 중간(내부)노드인 기지국과 KDC에 보낸다. 이것은 기본 프로토콜의 설정 단계1과 동일하다. 또한 L_1에 속하는 기지국 S_j도 M_i와 마찬가지로 랜덤수 r_{s_j}를 Z_q^*에서 선택하여, 기저값 g의 지수승을 하여 은닉한 값 $z_{s_j} (= g^{r_{s_j}} \bmod p)$를 계산한 다음 자신의 비밀키 K_{s_j}로 서명하여, $\sigma_{s_j} = Sign_{K_{s_j}}(z_{s_j})$, 메시지 $m_{s_j} = (z_{s_j} \| \sigma_{s_j})$를 루트 노드인 KDC에 보낸다. 이 때 기지국은 s_j, 서버 KDC는 자신의

비밀키 랜덤수 r_0, s를 Z_q^*에서 선택하여, 각각 $z_{s_j}(=g^{s_j} \bmod p)$와 $x_{s_j}=z_{s_j}^{r_{s_j}}(=g^{s_j r_{s_j}}), z(=g^s \bmod p)$ 와 $x_0=z^{r_0}(=g^{sr_0})$를 계산한다.

설정 단계2: 레벨 2로부터 자식노드에 해당되는 구성원 M_i의 메시지를 받은 각 기지국 S_j는 각 구성원의 공개키 PK_i로 서명 σ_i를 검증한 다음, 키트리의 서브그룹에 해당하는 서브그룹키 계산에 필요한 $x_{s_j}(=z_i^{s_j} \bmod p)$값을 계산한다. 이후 식 8과 같이 서브그룹 각각은 구성원을 식별할 수 있는 I_j를 포함한 공유 공통 값 X_{s_j}를 각각 계산한다.

$$X_{s_j}=\bigoplus_{j=0}^{k} H(ID_j \| x_{s_j}) (k\text{는 리프 노드의 수}) \cdots (식\ 8)$$

여기서 $x_{s_j}=(g^{rr_i})^{s_j}$의 형태로, r은 KDC의 비밀정보, s_j는 각 기지국의 비밀정보를 나타낸다.

또한 그룹 구성원에게 필요한 $X_{s_j} \oplus H(ID_j \| x_{s_j})$ 의 집합을 Y'라 할 때, 기지국의 비밀키 K_{s_j}로 서명하여 $\sigma_{s_j}(=Sign_{K_{s_j}}(I_j \| z_{s_j} \| Y'))$를 얻고, 메시지 $m_{s_j}(=(I_j \| z_{s_j} \| Y' \| \sigma_{s_j}))$를 그룹 구성원에게 멀티캐스트 한다. 여기서 I_j는 ID_j의 집합이다. 서버 KDC는 기지국을 구성원으로 하는 기본 프로토콜의 설정 단계 2의 실행 과정과 동일하다.

키 합의 단계: 공통키를 계산하는 단계로, 각각의 그룹 구성원은 서명 σ_{s_j}를 검증한 다음, 식 9와 같이 공유 공통 값 X_{s_j}와 x_{s_j}를 복구할 수 있다.

$$X_{s_j}=X_i \oplus H(I_i \| x_{s_j}),\ x_{s_j}=z_{s_j}^{r_i} \cdots\cdots\cdots\cdots\cdots\cdots (식\ 9)$$

이제 서버를 포함한 그룹의 구성원 모두는 일방향 해시함수를 사용하여 공통의 세션키 $SK(=H(X_{s_j} \| Y'))$를 계산할 수 있다.

제안한 기본 프로토콜의 실행 과정에서 알 수 있듯이, 서버를 포함한 그룹 구성원이 n명일 때, $n-1$번의 유니캐스트와 $\lceil (n-1)/3 \rceil + 1$번의 멀티캐스트로 공통의 세션키를 나눠 갖게 된다.

IV. Evaluation of Computation and Communication

1. Evaluation of the proposed protocol

본 연구에서 제안한 키 합의 방식은 세션키를 설정하기 위하여 설정 단계1에서 구성원 각자의 비밀키를 은닉하여 KDC(또는 기지국)에 전송하므로 KDC를 제외한 구성원의 수인 $n-1$ (여기서 n은 KDC를 포함한 그룹 구성원의 수)만큼의 유니캐스트 통신이 필요하고, 설정 단계2에서 KDC는 한 번의 멀티캐스트 통신을 필요로 한다. 따라서 2라운드의 통신만으로 제안한 프로토콜이 실행되고, 필요한 메시지 수도 n으로 최적이다 [15].

그리고 구성원이 그룹을 탈퇴할 경우, KDC(또는 기지국)는 한 번의 멀티캐스트 통신을 필요로 하고, 새로운 구성원이 그룹 통신에 참여할 경우, 참여하는 구성원의 수만큼의 유니캐스트와 한 번의 멀티캐스트를 필요로 한다.

제안한 키 합의 방식에서 KDC를 제외한 구성원 M_i는 각각 한 번의 서명 생성과 2번의 모듈러 상의 지수승(g^{r_i}, z^{r_i}) 연산을 한다. 그리고 KDC(기지국)는 구성원 M_i의 서명을 검증하기 위하여 $n-1$번의 서명을 검증하고, $n+1$번의 모듈러 상의 지수승 연산을 한다. 왜냐하면 자신을 제외한 구성원의 수 $n-1$번의 $x_i(=(g^{r_i})^s)$를 계산하고, KDC 자신의 은닉값 g^{r_0}과 $x_0(=(g^{r_0})^s)$를 계산해야 하기 때문이다.

또 구성원이 기존의 그룹을 탈퇴할 경우, 원래의 구성원은 이전에 미리 계산해 놓은 모듈러 지수승을 그대로 사용할 수 있으므로 해시함수 계산과 xor 연산만을 필요로 한다. 그리고 새로운 구성원이 그룹 통신에 참여할 경우, 새로이 참여하는 구성원만 2번의 모듈러 상의 지수승 연산이 필요하고, KDC는 새로운 구성원의 수만큼의 모듈러 상의 지수승 연산이 필요하다.

2. Evaluation of other protocols

n은 원래 그룹 구성원의 수이고, d는 갱신된 트리의 깊이, m은 그룹을 탈퇴하는 구성원의 수를 나타낸다고 하자.

Yongdae Kim et al.[6]에서 탈퇴하는 구성원의 수가 1이나 2인 경우, 지수승 연산은 $3n-7$번, 탈퇴하는 구성원의 수가 3 이상인 경우, 지수승 연산은 $3n-3m+2$번으로 평균 연산 비용은 $3(n/2)+2$번 필요로 한다.

또한 [6]에서 통신 비용은 기존 그룹에 새로운 구성원이 가입 시 가입하는 구성원이 각각의 지수승 연산을 하여 브로드캐스트 하는 단계, 모든 그룹 구성원이 키트리를 갱신하고 스폰서는 자신의 새로운 랜덤한 공유 값을 생성하여 지수승하여 은닉하고, 은닉키 값을 계산하여 갱신된 키트리를 브로드캐스트 하는 단계로 2라운드 만에 모든 구성원이 그룹키를 계산할 수 있다. 반면에 구성원 탈퇴 시에는 스폰서만이 자신의 새로운 공유 값을 생성하고 모든 비밀키와 은닉키를 계산하여 갱신된 트리를 브로드캐스트 하면 되므로 1라운드 만에 모든 구성원이 그룹키를 계산할 수 있다.

Sangwon Lee et al.[7]에서 pairing 계산은 그룹 구성원이 새로 그룹에 가입하거나 탈퇴할 때 모두, 리프 노드에 그룹 구성원을 할당할 수 있는 삼진 트리의 깊이 $\lceil \log_3 n \rceil$에서 1을 뺀 횟수만큼 필요하다. 또한 타원곡선 위의 점 P를 여러 번 더하는 연산, 즉 $2P(=P+P)$, $3P(=2P+P)$, … 등 점 P에 대한 곱셈 연산은 $\lceil \log_3 n \rceil + 1$ 만큼의 횟수만큼 필요하다. 그런데, 이 pairing 연산 속도는 Barreto et al.[16] 등의 연구에 의하면 모듈러 상의 지수승 연산보다 약 3배 정도 느려서 계산

Table 2. Communication and Computation Costs

	Evaluation	Yongdae Kim et al.[6]		Sangwon Lee et al.[7]		EHBT[8]		Proposed Protocol	
		Join	Leave	Join	Leave	Join	Leave	Join	Leave
Communication	Rounds	2	1	2	1	2	1	2	1
	Messages	3	1	3	1	$\log_2 n$	$\log_2 n$	2	1
Computation	Hash Functions	0	0	$\lceil \log_3 n \rceil + 1$	$\lceil \log_3 n \rceil + 1$	$3n+d$	$2n+d$	$2(n-1)$	$2m$
	xor Operations	0	0	0	0	$3n+d$	$2n+d$	$2(n-1)$	$2m$
	Exponentiations (or [7]Point Multiplications)	dn	$\frac{3n}{2}+2$	$\lceil \log_3 n \rceil + 1$	$\lceil \log_3 n \rceil + 1$	0	0	$n+1$	m
	Pairings	0	0	$\lceil \log_3 n \rceil - 1$	$\lceil \log_3 n \rceil - 1$	0	0	0	0

비용 측면에서 다른 프로토콜보다는 효율성이 떨어진다. 그리고 [7]에서의 통신 비용은 [6]에서와 같은 방법으로 키트리를 갱신하고, 스폰서의 비밀값을 갱신하므로 프로토콜 [6]의 통신 비용과 동일하다.

EHBT[8]의 계산 비용 측면에서 해시함수 연산은 KDC가 $3n-1$번, 새로이 통신에 가입한 구성원의 형제 노드가 $d+1$번으로 총 $3n+d$번 필요하다. 또한 그룹 구성원의 반이 그룹을 탈퇴할 때, 해시함수 연산은 KDC가 $2n-1$번, 트리가 갱신되고 탈퇴한 구성원의 형제 노드가 $d+1$번으로 총 $2n+d$번 필요하다. 그리고 xor 연산은 해시함수의 입력 값을 xor 연산을 사용하여 계산하므로 해시함수의 연산 횟수만큼 필요하다.

또한 EHBT 프로토콜의 안전성은 일방향 해시함수 h의 암호학적인 성질에 의존하는 반면, 제안한 프로토콜은 해시함수의 일방향 성질뿐만 아니라, CDH 가정에 기반하여 수동적인 공격자에 대하여도 안전하다.

EHBT의 통신 비용은 구성원 가입 시 가입하는 구성원과 그 형제 노드가 각각 n번의 유니캐스트, KDC가 $n-1$번의 멀티캐스트를 필요로 한다. 또한 구성원이 탈퇴하는 경우 탈퇴하는 구성원의 형제 노드가 n번, KDC가 $n-1$번으로 $2n-1$번의 멀티캐스트를 필요로 한다.

표2에서는 제안한 프로토콜과 관련 연구에서 살펴본 3가지 프로토콜을 통신 비용과 계산 비용 측면에서 비교하여 정리하였다. 제안한 프로토콜은 계산 방식이 각각 다른 프로토콜들에 비해 전송 메시지 횟수에서 1회 적은 효율성과 계산 비용 측면에서도 모듈러 지수승을 줄이고 해시함수와 xor 연산으로 효율성을 증가시켰음을 알 수 있다.

Ⅴ. Security Analysis

1. Security considerations

제안한 프로토콜이 기반하고 있는 계산적 Diffie-Hellman (CDH) 문제는 임의의 $a, b \in Z_q^*$와 생성원소 g에 대하여 $g^a, g^b \pmod q$이 주어질 때 $g^{ab} \pmod q$을 구하는 문제이다[17]. 즉 제안한 프로토콜에서 $x_i (= g^{r_i s})$와 $z (= g^s)$ 값을 알 때, 그룹 구성원은 자신이 갖고 있는 랜덤한 비밀값 r_i로 세션키 값을 복구할 수 있는 $z^{r_i} (= x_i)$값을 계산할 수 있지만, 그룹 구성원 이외의 사람이 다항식 시간 안에 x_i값을 알아낼 확률이 무시할 수 있을 만큼 아주 작으므로, 안전하다는 의미이다.

또한 그룹 통신에 새로운 구성원이 참가하거나 기존 구성원이 탈퇴할 때, KDC(또는 기지국)는 매 세션마다 자신의 비밀키 r값을 갱신함으로써 이전의 그룹키로부터 새로운 세션키를 알아내지 못하므로 순방향과 역방향 안전성을 제공한다.

2. Security analysis

제안한 키 합의 방식의 안전성 분석은 표준적인 안전성 모델을 사용하여 수동적인 공격자에 대하여 안전함을 증명한다[18].

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

2.1 Security model

• 프로토콜 참가자(participant) : 그룹키 합의 프로토콜 P에 참가하는 구성원의 집합들로, 각각의 구성원은 오라클(oracle)이라고 부르는 인스턴스(예를 들면 Π_i^s는 구성원 M_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 프로토콜 참가자들과 상호 작용한다.

• 실행 질의 Execute(M) : 이 질의에 대한 응답으로 프로토콜 참가자의 인스턴스 중에서 정직한 프로토콜의 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(M_i) : 순방향 안전성(forward secrecy)을 다루기 위하여 고려한 질의로, 그룹 구성원 M_i의

장기간 사용하는 비밀키를 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.

2.2 Computational Diffie-Hellman(CDH) problem

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 해결함에 있어, 임의의 확률적 다항식 시간 안에, g^{ab} 값을 출력함에 있어 알고리즘 A의 이익(advantage)은 다음과 같이 정의할 수 있다.

$$Adv_G^{CDH}(A)=\left|\Pr[g^{ab}\leftarrow A(G,g,g^a,g^b)|g\in G;a,b\in_R Z_q^*]\right|$$

이 때, CDH 가정은 모든 확률적 다항식 시간 안에, 알고리즘 A에 대하여, $Adv_G^{CDH}(A)$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 $Adv_G^{CDH}(t)$는 기껏해야 t 시간 내에 모든 공격자에 의해 실행되는 알고리즘 A의 이익 $Adv_G^{CDH}(A)$의 최대값을 나타낸다.

2.3 Definition of security

공격자의 알고리즘 A가 그룹키 합의 프로토콜 P의 실행 중에 공격자에게 세션키가 명백하게 알려져 있지 않은 fresh한 오라클에게 테스트 질의를 요청하면 이 질의에 대한 응답으로 임의의 비트의 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 이익은 다음과 같이 정의한다.

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

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

2.4 Security proof against passive adversaries

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

$$Adv_{A,P}(k)=2q_hq_{ex}Adv_G^{CDH}(t'),\ t'=t+O(nq_{ex}t_{\mathrm{Exp}})$$

여기서 t_{Exp}는 G에서 지수승을 계산하는 데 필요로 하는 시간이다.

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

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

$$Real=\left\{T(S,K)\left|\begin{array}{l} r_1,r_2,\cdots,r_n,r\in_R Z_q^*;ID_i\in\{0,1\}^l;\\ z_1=g^{r_1},z_2=g^{r_2},\cdots,z_n=g^{r_n},z=g^r;\\ x_1=g^{rr_1},x_2=g^{rr_2},\cdots,x_n=g^{rr_n};\\ h_1=H(ID_1\|x_1),h_2=H(ID_2\|x_2),\ \cdots,\\ \qquad\qquad h_n=H(ID_n\|x_n);\\ X=\bigoplus_{i=1}^{n}h_i;\\ y_1=X\oplus h_1,y_2=X\oplus h_2,\cdots,\\ \qquad\qquad y_{n-1}=X\oplus h_{n-1};\end{array}\right.\right\}$$

$$Rand=\left\{T(S,K)\left|\begin{array}{l} r_1,r_2,\cdots,r_n,r\in_R Z_q^*;\\ ID_i,w_1,w_2,\ \cdots,w_n\in\{0,1\}^l;\\ z_1=g^{r_1},z_2=g^{r_2},\cdots,z_n=g^{r_n},z=g^r;\\ x_1=g^{rr_1},x_2=g^{rr_2},\cdots,x_n=g^{rr_n};\\ h_1=w_1,h_2=w_2,\ \cdots,h_n=w_n;\\ X=\bigoplus_{i=1}^{n}h_i;\\ y_1=X\oplus h_1,y_2=X\oplus h_2,\cdots,\\ \qquad\qquad y_{n-1}=X\oplus h_{n-1};\end{array}\right.\right\}$$

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

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

[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 다음과 같은 입력 값이 주어졌을 때, $r\alpha=\beta \bmod q$인 $C(=g^\beta)$ 값을 출력하는 알고리즘을 다음과 같이 구성한다.

$$(g,A=g^r,B=g^\alpha)\in G^3$$

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

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

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 $(ID_i \| x_i')$ 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.

$$\Pr[A(T, SK_b) = b | (T, SK_1) \leftarrow Rand; \\ SK_0 \leftarrow \{0,1\}^l; b \leftarrow \{0,1\}] = 1/2$$

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

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

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

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

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

$$\Pr[A(T, X_b) = b | (T, X_1) \leftarrow Rand; \\ X_0 \leftarrow \{0,1\}^l; b \leftarrow \{0,1\}] = 1/2$$

이 성립하며, H가 랜덤 오라클이므로 도움정리 2가 성립한다. □

이제 위의 도움정리 1, 2를 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 G에서 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,

$$\Pr[A(T, SX_b) = b | (T, SK_1) \leftarrow Real; \\ SK_0 \leftarrow \{0,1\}^l; b \leftarrow \{0,1\}] = 1/2 + \epsilon,$$
$$Adv_G^{CDH}(B) = \epsilon / (q_h q_{ex})$$

따라서 정리 1이 성립한다. □

Ⅵ. Conclusion

본 논문에서 제안한 프로토콜은 계산적 Diffie-Hellman 가정에 기반하여 그룹 구성원들만이 자신이 선택한 랜덤한 비밀 값과 KDC(또는 기지국)가 전송해준 정보를 사용하여 그룹 공통의 세션키를 복구해 낼 수 있는 키 합의 방식으로 이동성이 잦은 모바일 환경에 적합하다.

또한 겹선형 Diffie-Hellman (BDH) 문제에 기반한 TDGH (Tree-based Group Diffie-Hellman)를 확장한 기존의 연구 [7]와 비교해 볼 때, pairing 연산보다는 3배 빠른 지수승 연산을 사용하였고, 기존 연구 [8]과 비교해 볼 때 해시함수와 xor 연산을 사용하여 계산적인 비용을 감소시켰다. 그리고 EHBT 프로토콜의 안전성은 일방향 해시함수에 의존하는 반면, 제안한 프로토콜은 CDH 가정에 기반하여 수동적인 공격자에 대하여도 안전함을 랜덤 오라클 모델에서 증명하였다.

향후 연구에서는 효율적인 키트리 관리에 대한 연구와 계산적인 효율성에 대한 실험이 진행되어야 할 것이다.

REFERENCES

[1] B. Bhargava, M. Annamalai, and E. Pitoura, "Digital Library Services in Mobile Computing", ACM SIGMOD Record, Vol. 24, No. 4, pp. 34-39, Dec. 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, Sep. 2002.
[4] Sung-Hwa Lim and Jai-Hoon Kim, "Real-time Broadcast Algorithm for Mobile Computing", The Journal of Systems and Software, Vol. 69, No. 2, pp. 173-181, Jan. 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, Nov. 1976.
[6] Yongdae Kim, Adrian Perrig, and Gene Tsudik, "Group Key Agreement Efficient in Communication", IEEE Transactions on Computers, Vol. 53, No. 7, pp. 905-921, Jul. 2004.
[7] Sangwon Lee, Yongdae Kim, Kwangjo Kim, and Dae-Hyun Ryu, "An Efficient Tree-Based Group Key Agreement Using Bilinear Map", ACNS 2003, LNCS 2846, pp. 357-371, 2003.
[8] Sandro Rafaeli, Laurent Marthy, and David Hutchison, "EHBT: An Efficient Protocol for Group Key Management", NGC(Networked Group Communication) 2001, LNCS 2233, pp. 159-171, Oct. 2001.
[9] Lijun Liao and Mark Manulis, "Tree-based group key agreement framework for mobile ad-hoc networks", Elsevier, Future Generation Computer Systems, Vol. 23, No. 6, pp. 787-803, July 2007.
[10] Sang-won Lee, Jung Hee Cheon, and Yongdae Kim, "Tree-based Group Key Agreement Protocol using Pairing", Journal of The Korea Institute of Information Security and Cryptology, Vol. 13, No. 3, pp. 101-110, Jun. 2003.
[11] Abhimanyu Kumar and Sachin Tripathi, "Ternary Tree Based Group Key Agreement Protocol Over Elliptic Curve for Dynamic Group", International Journal of Computer Applications(0975-888), Vol. 86, No. 7, pp. 17-25, Jan. 2014.
[12] Yvo Desmedt, Tanja Lange, and Mike Burmester, "Scalable Authenticated Tree Based Group Key Exchange for Ad-Hoc Groups", FC 2007 and USEC 2007, LNCS 4886, pp. 104-118, 2007.
[13] Junghyun Nam, Juryon Paik, Youngsook Lee, Jin Kwak, Ung Mo Kim, and Dongho Won, "Infringing Key Authentication of an ID-Based Group Key Exchange Protocol Using Binary Key Trees", KES 2007/WIRN 2007, Part Ⅰ, LNAI 4692, pp. 672-679, 2007.
[14] Minghui Zheng, Guohua Cui, Muxiang Yang, and Jun Li, "Scalable Group Key Management Protocol Based on Key Material Transmitting Tree", ISPEC 2007, LNCS 4464, pp. 301-313, 2007.
[15] 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.
[16] P.S.L.M. Barreto, H.Y. Kim, B. Linn, and M. Scott, "Efficient Algorithms for Pairing-Based Cryptosystems", Advances in Cryptology-Crypto 2002, LNCS 2442, pp. 354-368, Aug. 2002.
[17] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone, "*Handbook of Applied Cryptography*", CRC Press, p. 113, 1997.
[18] M. Bellare and P. Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols", Proc. of the 1st ACM Conference on Computer and Communication Security(CCS 1993), pp. 62-73, Nov. 1993.

Author

Seokhyang Cho received the B.S. degree in Mathematics from Ewha Womans University, Korea, in 1986 and the Ph.D. degree in Electrical and Computer Engineering from Sungkyunkwan University, Korea, in 2006. Dr. Cho joined the faculty of the Dept. of Information and Communication at Pyeongtack University, Pyeongtack-si, Korea, in 2016. She is currently an Assistant Professor in the Dept. of Information and Communication, Pyeongtaek University. She is interested in cryptographic protocols, information security, and sensor network.