情報保護學會論文誌
第19卷 第5號, 2009. 10

# 실생활 응용을 위한 짧은 그룹 서명 기법(BBS04)에 대한 연구*

강 전 일,[1†] 양 대 헌,[1‡] 이 석 준,[2] 이 경 희[3]
[1]인하대학교, [2]한국전자통신연구원, [3]수원대학교

# Study on the Use of Short Group Signature (BBS04) in Real Applications*

Jeonil Kang,[1†] DaeHun Nyang,[1‡] Sokjoon Lee,[2] KyungHee Lee[3]
[1]INHA University, [2]Electronics and Telecommunications Research Institute,
[3]University of Suwon

요 약

Boneh 등에 의해 소개되었던 짧은 그룹 서명 기법은 익명성을 지원하는 대표적인 서명 기법이다. 그러나 이를 실제 응용 환경에 적용할 때에는 짧은 그룹 서명 기법이 가진 여러 제한 사항에 대해서 반드시 고려해야한다. 어떤 서비스 제공자의 입장에 있어서는, 사용자의 완전한 익명성이 서비스가 불가능하도록 하는 등의 문제가 있을 수 있다. 따라서 일정 부분 사용자의 익명성을 제한하는 지역 연결성이 필요하다. 또한 그룹 매니저로부터 일방적으로 할당되는 그룹 서명키는 강한 무죄 입증성을 갖지 못하므로, 강한 무죄 입증성을 갖도록 짧은 그룹 서명 기법이 수정되어야할 필요가 있다. 이 논문에서는 이러한 관점에서 실용적인 짧은 그룹 서명 기법에 대한 깊이 있는 고찰하고 이를 바탕으로 지역 연결성과 무죄 입증성의 지원을 위한 몇몇 제안을 한다.

ABSTRACT

The short group signature introduced by Boneh et al. is one of famous anonymous signature schemes. However, for applying it to the real applications, several restrictions should be considered. The perfect anonymity of users, which is given by group signatures, prevents service providers to provide certain services or resources. For this reason, the local linkability which reduces the anonymity of users has to be provided to the service providers. In addition, the group signature keys, which are one-sidedly assigned from a group manager, cannot support the strong exculpability of users. Hence, the short group signature has to be modified for supporting the strong exculpability. In this paper, we perform a study on the use of the short group signature by proposing a few methods for supporting those two properties.

**Keywords:** short group signature, local linkability, strong exculpability

## I. 서 론

많은 인터넷 상에서의 서비스 제공자가 사용자에게 실명 인증을 요구하는 이유는, 어떤 사용자가 악의적인 글을 게시하는 등의 법적인 책임을 물어야하는 행위를 할 가능성이 있기 때문이다. 실제 익명 사용자에 의한 악의적인 댓글은 사회적 문제로까지 이어지고 있고, 사회 일각에서는 인터넷 실명제의 도입을 주장하기도 한다. 그러나 사용자의 프라이버시 또한 지켜져

접수일(2009년 1월 5일), 수정일(2009년 6월 12일), 게재확정일(2009년 8월 1일)
* 본 연구는 정보통신부 및 정보통신연구진흥원의 IT신성장동력핵심기술개발사업의 일환으로 수행하였음. (2008-F-036-01, 익명성기반의 U-지식 정보보호기술개발)
† 주저자, dreamx@isrl.kr
‡ 교신저자, nyang@inha.ac.kr

야 할 당연한 가치라는 것은 부인할 수 없는 사실이다. 이러한 사회의 상충된 요구를 동시에 해결하기 위한 한 가지 방법으로 기술적 접근을 생각해볼 수 있다.

그룹 서명 기법(group signature)[1], 링 서명 기법(ring signature)[2], 및 추적 가능 서명 기법(traceable signature)[3]은 익명성을 지원하는 서명 기법으로 잘 알려져 있다. 링 서명 기법을 제외한 다른 두 서명 기법은 그룹 마스터키를 이용하여 서명을 생성한 그룹 멤버를 알아내는 기능을 지원한다. 이러한 특징은 앞서 설명했던 모순처럼 보이는 상황을 해결할 수 있는 가능성을 제시한다. 추적 가능 서명 기법은 그룹 서명 기법을 변형해서 만들 수 있기 때문에, 그룹 서명 기법이 필요한 환경에 맞도록 변형하는 방법을 우선 고려해볼 필요가 있다. 한편, 사용자에 대한 완전한 익명성을 보장하는 익명 신용장(anonymous credential) 시스템[4,5]과 그룹 서명 기법은 동일한 서명 기법을 변형하거나 특정 요소를 더하여 각각 만들 수 있음이 알려져 있다[6]. 이는 그룹 서명 기법과 익명 신용장 시스템을 상호 변환할 수 있음을 의미하며, 이 둘을 같은 관점에서 바라볼 수 있음을 뜻한다.

그룹 서명 기법 중에서 짧은 그룹 서명 기법(short group signature, 이후의 해당 논문은 BBS04로 지칭)[7]은 겹선형 쌍함수(bilinear pairing)를 이용하여 200-Byte 정도의 짧은 서명 길이는 갖는 그룹 서명 기법으로, 이후의 많은 그룹 서명 기법에 직·간접적인 영향을 주었다. 그러나 논문 BBS04에 기술된 것만 가지고는 몇몇 환경을 제외하고는 다양한 환경에서 적합한 시스템을 만들기란 부족한 면이 있다. 그룹 매니저는 그룹 멤버의 서명키를 철폐할 수 있고, 그룹 멤버의 서명을 추적할 수 있지만, 그 외의 그룹 멤버나 그룹 멤버가 아닌 참여자는 단지 서명이 올바르다는 것만 확인이 가능하다. 서비스 제공자에게 있어서 잘못된 사용자에 대한 정보를 그룹 매니저에게 매번 물어보는 것은 쉽지 않으며, 설령 그러한 사용자의 실명을 알아냈다고 하더라도 사용자 서명키 철폐(revocation)가 일어나지 않으면 서비스 제공자는 여전히 서비스를 제공해야 한다. BBS04에서 사용자 서명키 철폐는 매우 많은 연산이 필요한 작업이며, 설령 연산이 적다고 하더라도 이는 지나치게 징벌적인 조치이다[8]. 이러한 조건을 실제 응용환경에 적용하였을 때, 사용자의 익명성을 위해서는 서비스 제공자는 단지 그룹 멤버의 역할만 수행하도록 해야 한다. 이는, 사용자의 연속성이 보장되어야 하는 일부 서비스가 불가능함을 의미한다. 또한, 논문 BBS04에서 기술된 것과 같이 무죄 입증성(exculpability)을 강화해야할 필요가 있으나 논문에서는 해당 내용에 대해서 깊이 있게 다루지 않고 있다.

이 논문에서는 짧은 그룹 서명 기법을 실제 환경에 적용하였을 때, 몇몇 미흡한 점을 살펴보고 이를 보충할 수 있는 지역 연결성(local linkability)과 무죄 입증성에 대해서 깊게 생각해본다. 이후의 논문 구성은 다음과 같다. 2장에서는 짧은 그룹 서명 기법의 전체적인 모습을 살펴보고, 3장에서는 앞서 지적했던 문제를 풀기 위한 시스템 구성 및 그에 따라 발생하는 문제들의 해결 방법을 찾아본다. 4장에서는 각각 방법론 간의 비교 분석을 수행한다. 5장은 이 논문의 결론과 더 해결해야하는 문제에 대해서 담고 있다.

## II. 짧은 그룹 서명 기법(BBS04)의 소개

### 2.1 수학적 가정

짧은 그룹 서명 기법은 q-SDH(q-Strong Diffie-Hellman) 가정과 DLDH(Decisional Linear Diffie-Hellman) 가정에 기반을 두고 그 안전성을 보장한다.

- q-SDH: 주어진 $(q+2)$개의 $(g_1, g_2, g_2^{\gamma}, g_2^{(\gamma^2)}, \cdots, g_2^{(\gamma^q)})$로부터 어떠한 $x \in \mathbb{Z}_p^*$에 대해서 $(g_1^{1/(x+\gamma)}, x)$를 구하는 것은 어렵다.
- DLDH: 주어진 $(u, v, h, u^a, v^b, h^c)$로부터 $c = a+b$를 확인하는 것은 어렵다.

### 2.2 겹선형 쌍함수(Bilinear Pairing)

위수가 소수 $p$인 곱셈 순환군(multiplicative cyclic group) $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$가 있을 때, 겹선형 쌍함수(또는 맵) $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$은 다음과 같은 성질을 만족한다. ($g_1$과 $g_2$는 각각 $\mathbb{G}_1$와 $\mathbb{G}_2$의 생성자이다.) 또한 $\mathbb{G}_2$에서 $\mathbb{G}_1$으로의 계산 가능한 isomorphism이 존재한다. 즉, $g_1 = \psi(g_2)$이다.

- Bilinear: 모든 $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$와 $a, b \in \mathbb{Z}_p$에 대해서 $e(u^a, v^b) = e(u, v)^{ab}$
- Non-degenerate: $e(g_1, g_2) \neq 1$
- Computable: 모든 $u \in \mathbb{G}_1$, $v \in \mathbb{G}_2$에 대해서 $e(u, v)$의 계산이 가능한 효율적인 알고리즘이 존재한다.

### 2.3 짧은 그룹 서명 기법

짧은 그룹 서명 기법에서 주로 사용되는 기호 및 변수는 〔표 1〕과 같다. 이 논문에서는 짧은 그룹 서명 기법과 같은 기호를 사용하며, 기술에 있어 추가적으로 필요한 기호와 변수는 해당 위치에서 다시 선언하여 사용한다. 대부분의 변수는 필요에 따라 랜덤하게 추출되거나 연산의 결과로써 할당된다.

짧은 그룹 서명 기법은 우선 q-SDH 문제와 겹선형 쌍함수를 기반으로 하여 영지식 증명(zero-knowledge proof) 프로토콜을 설계한다. 그런 뒤, Fiat-Shamir Heuristic을 적용하여 해당 영지식 증명 프로토콜을 그룹 서명 기법으로 변환한다.

그룹 마스터키를 $gmk=\{(\xi_1,\xi_2),\gamma\}$, 그룹 공개키를 $gpk=\{h,u,v,g_1,g_2,w\}$라고 할 때, 그룹 매니저는 그룹 멤버에게 $A_i^{x_i+\gamma}=g_1$에 해당하는 $(A_i,x_i)$를 할당한다. 그룹 멤버는 증명자(Prover)로써 자신에게 할당 받은 $(A,x)$의 소유를 증명하기 위하여 영지식 프로토콜을 수행한다. 우선 $\alpha,\beta\in_R\mathbb{Z}_p$을 선택한 뒤, $\delta_1\leftarrow\alpha x$, $\delta_2\leftarrow\beta x$라고 하고, $A$의 선형 암호화(linear encryption)에 해당하는 $T_1\leftarrow u^{\alpha}$, $T_2\leftarrow v^{\beta}$, $T_3\leftarrow A_i h^{\alpha+\beta}$를 계산한다. 그 뒤, $r_{\alpha},r_{\beta},r_x,r_{\delta_1},r_{\delta_2}\in_R\mathbb{Z}_p$를 선택한 후, $R_1\leftarrow u^{r_{\alpha}}$, $R_2\leftarrow v^{r_{\beta}}$, $R_4\leftarrow T_1^{r_x}u^{-r_{\delta_1}}$, $R_5\leftarrow T_2^{r_x}v^{-r_{\delta_2}}$, $R_3\leftarrow e(T_3,g_2)^{r_x}e(h,w)^{-r_{\alpha}-r_{\beta}}e(h,g_2)^{-r_{\delta_1}-r_{\delta_2}}$를 계산한다. $A_i$의 암호문에 해당하는 $T_1,T_2,T_3$와 commit에 해당하는 $R_1,R_2,R_3,R_4,R_5$를 검증자(verifier)에게 전송한다. (이 때, commit은 $T_1,T_2,T_3$과 관련되어 생성된다.) 검증자는 challenge $c\in_R\mathbb{Z}_p$를 증명자에게 보내면, 증명자는 response $s_{\alpha}\leftarrow r_{\alpha}+c\alpha, s_{\beta}\leftarrow r_{\beta}+c\beta$, $s_x\leftarrow r_x+cx_i$, $s_{\delta_1}\leftarrow r_{\delta_1}+c\delta_1$, $s_{\delta_2}\leftarrow r_{\delta_2}+c\delta_2$를 계산하여 검증자에게 보낸다. 검증자는 $s_{\alpha},s_{\beta},s_x,s_{\delta_1},s_{\delta_2}$를 이용하여 commit이 올바른지 검증하기 위하여 $R_1=u^{s_{\alpha}}T_1^{-c}$, $R_2=v^{s_{\beta}}T_2^{-c}$, $R_4=T_1^{s_x}u^{-s_{\delta_1}}$, $R_5=T_2^{s_x}v^{-s_{\delta_2}}$, $R_3=e(T_3,g_2)^{s_x}e(h,w)^{-s_{\alpha}-s_{\beta}}e(h,g_2)^{-s_{\delta_1}-s_{\delta_2}}(e(T_3,w)/e(g_1,g_2))^c$인지 확인한다.

이를 그룹 서명 기법으로 전환할 경우, 서명키 $(A_i,x_i)$를 가진 사용자는 위와 유사한 과정을 거쳐 $T_1,T_2,T_3$와 $R_1,R_2,R_3,R_4,R_5$를 계산한 뒤, 메시지 $m$과 함께 체크섬 $c\leftarrow H(m,T_1,T_2,T_3,R_1,R_2,R_3,R_4,R_5)$를 생성하고 $\sigma\leftarrow(T_1,T_2,T_3,c,s_{\alpha},s_{\beta},s_x,s_{\delta_1},s_{\delta_2})$를 서명으로 출력한다. 서명의 확인 작업은 $c,s_{\alpha},s_{\beta},s_x,s_{\delta_1},s_{\delta_2}$를 이용하여 $\widetilde{R}_1,\widetilde{R}_2,\widetilde{R}_3,\widetilde{R}_4,\widetilde{R}_5$를 계산하고 $c=H(m,T_1,T_2,T_3,\widetilde{R}_1,\widetilde{R}_2,\widetilde{R}_3,\widetilde{R}_4,\widetilde{R}_5)$인지 확인함으로써 이루어진다. (이는 영지식 증명 프로토콜에서 검증 작업과 유사하다.) 그룹 마스터키 $(\xi_1,\xi_2)$를 알고 있는 그룹 매니저는 그룹 공개키 $h,u,v$의 특별한 관계를 이용하여 그룹 멤버의 식별자 $A$를 계산해낼 수 있다. 즉, $A=T_3/(T_1^{\xi_1}T_2^{\xi_2})$이다.

〔표 1〕 짧은 그룹 서명 기법(BBS04)에 주로 사용되는 기호 및 변수

| 기호 및 변수 | 설명 |
|---|---|
| $e(,)$ | 겹선형 쌍함수 |
| $\mathbb{G}_1,\mathbb{G}_2$ | 차수가 소수 $p$인 두 순환 그룹 |
| $g_1,g_2$ | 순환 그룹 $\mathbb{G}_1,\mathbb{G}_2$의 생성자 |
| $\psi()$ | $\mathbb{G}_2\rightarrow\mathbb{G}_1$으로의 isomorphism |
| $\xi_1,\xi_2\in\mathbb{Z}_p^*$ | 선형 암호화를 위한 비밀키 |
| $u,v,h\in\mathbb{G}_1$ | $u^{\xi_1}=v^{\xi_2}=h$인 관계를 만족하는 선형 암호화를 위한 공개키 |
| $\gamma$ | 그룹 멤버 비밀키 발급을 위한 비밀 발급키 |
| $w=g_2^{\gamma}$ | 그룹 멤버 비밀키 발급을 위한 발급키에 해당하는 공개키 |
| $gmk=\{(\xi_1,\xi_2),\gamma\}$ | 그룹 마스터키 |
| $gpk=\{u,v,h,g_1,g_2,w\}$ | 그룹 공개키 |
| $gsk[i]=(A_i,x_i)$ | 사용자 $i$의 그룹 멤버 비밀키, $x_i\in\mathbb{Z}_p^*$, $A_i^{x_i+\gamma}=g_1$인 관계를 만족 |

### 2.4 짧은 그룹 서명 기법의 남겨진 문제들

짧은 그룹 서명 기법을 인증에 도입할 때, 앞서 언급했던 바와 같이 그룹 매니저가 단순히 서비스 제공자의 역할을 수행할 경우, 사용자의 익명성은 보장되지 않는다. 따라서 필연적으로 권한의 분할(separation of authority)을 통하여 사용자의 익명성을 보장하는 방법이 논의되어야 한다. 이때, 적용 부분에 따라서 짧은 그룹 서명 기법을 변형하여 만들 수도 있고, 외적인 방법론으로 만들어질 수도 있다.

논문 BBS04에서는 강한 무죄 입증성(strong exculpability)을 이루기 위해서는 그룹 멤버가 $A_i^{x_i+\gamma}h_1^{y_i}=g_1$의 관계에 있는 비밀키 $y_i$를 선택하도록 가입(join) 절차를 두어야 한다고 말하고 있다. 그러나 그룹 멤버의 새로운 비밀키 $y_i$를 도입함으로써 늘

어나게 되는 서명의 길이라든지 해결하기 힘들어지는 문제들에 대해서는 구체적으로 설명하고 있지 않다.

## III. 실제 응용 환경을 위한 짧은 그룹 서명 기법

### 3.1 지역 연결성

#### 3.1.1 권한의 분할

짧은 그룹 서명 기법에서 그룹매니저의 권한은 실제로 두 가지로 나눌 수 있다. 그룹 마스터키 $\{(\xi_1,\xi_2),\gamma\}$ 중에서 앞의 두 $(\xi_1,\xi_2)$는 그룹 멤버의 식별자 $A_i$를 개시(open) 하기 위해서 사용되며, 뒤의 $\gamma$는 그룹 멤버에게 서명키 $(A_i,x_i)$를 할당하기 위해서 사용되며, 이는 독립적으로 선택되고 계산될 수 있다. 따라서 전자를 알고 있는 그룹 내의 멤버를 개시자(opener), 후자를 알고 있는 그룹 내의 멤버를 발행자(issuer)라고 칭할 수 있다. 개시자의 비밀키 $(\xi_1,\xi_2)$는 그룹 공개키 $h(=u^{\xi_1}=v^{\xi_2}),u,v$의 형태로 존재하고, 발행자의 비밀키 $\gamma$는 그룹 공개키 $w=g_2^{\gamma}$ 형태로 존재하기 때문에 개시자와 발행자 상호간 어떠한 형태의 비밀 정보에 대한 공유가 필요하지 않다. 발행자는 그룹 멤버에게 서명키 $(A_i,x_i)$를 할당하기에 앞서, 해당 그룹 멤버를 실명으로 인증할 수도 있다. 그러나 이는 정책적인 문제로, Benjumea[9], 양대헌[10,11], 권태경[12] 등이 앞서 연구하였던 인증 권한 분할 기법을 또 다시 연결하여 사용할 경우, 실명 인증을 통하지 않아도 된다.

#### 3.1.2 전역 연결성과 지역 연결성

서비스 제공자는 그룹 공개키를 이용해서, 사용자가 서명한 값의 유효성을 확인할 수 있다. 이는 짧은 그룹 서명 기법에서 그룹 멤버가 수행할 수 있다. 그러나 경우에 따라서는 서비스 제공자가 서로 다른 메시지의 서명을 동일한 사용자가 했는지 확인해야할 필요가 있다. (예, 익명으로 진행되는 옥션 경매에서 당첨자의 확인) 단순하게는 서비스 제공자는 사용자 확인을 위하여 개시자에게 서로 다른 두 서명 $(\sigma,\sigma')$을 전달하고 같은 사용자가 서명한 것인지 질의할 수 있다. 만약 이러한 작업이 충분히 많이 하게 되면, 서비스 제공자는 사용자들에 대해서 연결성을 발견해낼 수 있다. 서비스 제공자는 사용자의 식별자를 알아내지 못하지만 사용자들에 대한 연결성을 갖게 할 수 있는 것이다. 이를 개시자가 갖는 전역 연결성(global linkability)에 비하여 지역 연결성이라고 한다. 이러한 지역 연결성은 서비스 제공자가 데이터 마이닝 등을 통한 사용자들의 서비스 이용 패턴 등을 분석하는데 사용될 수 있다.

그러나 이 경우, 단지 같은 사용자인지 확인하기 위한 서비스 제공자의 질의에서 개시자는 전달된 서명으로부터 사용자의 식별자를 알아낼 수 있다. 사용자는 자신과 한 번도 통신을 한 적이 없는 개시자를 신뢰해야만 하는 상황에 놓이게 된 것이다.

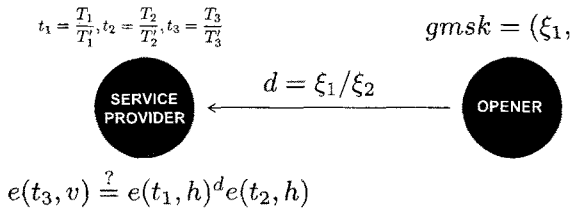
#### 3.1.3 지역 연결성 지원 방안

개시자가 사용자의 식별자를 알아내지 못하도록 하기 위해서 다음과 같은 두 가지 방법을 생각해볼 수 있다. 우선 첫 번째로는 서비스 제공자가 개시자에게 서명 $(\sigma,\sigma')$ 대신에 서명에 담겨져 있는 $(T_1,T_2,T_3)$와 $(T_1',T_2',T_3')$을 이용하여 안에 담겨진 식별자를 $t_1=T_1/T_1',t_2=T_2/T_2',t_3=T_3/T_3'$처럼 수학적으로 제거하고 이를 개시자에게 확인해달라고 하는 것이다. 만약 $(T_1,T_2,T_3)$와 $(T_1',T_2',T_3')$가 동일한 사용자의 식별자를 담고 있다면,

$$t_3=\frac{Ah^{\alpha+\beta}}{Ah^{\alpha'+\beta'}}=\frac{h^{\alpha+\beta}}{h^{\alpha'+\beta'}}=\frac{u^{\xi_1\alpha}v^{\xi_2\beta}}{u^{\xi_1\alpha'}v^{\xi_2\beta'}}=\left(\frac{u^{\alpha}}{u^{\alpha'}}\right)^{\xi_1}\left(\frac{v^{\beta}}{v^{\beta'}}\right)^{\xi_2}=t_1^{\xi_1}t_2^{\xi_2} \quad (1)$$

이어야 한다. 이 방법은 매번 필요할 때마다 서비스 제공자가 개시자에게 확인을 요청을 해야 하는 번거로움이 있다. 또한 개시자가 사용자의 식별자에 대한 정보를 알고 있는 발행자와 공모할 경우 사용자의 식별자와 사용자의 서명들이 서비스 제공자 밖으로 노출될 위험성이 존재한다. 이는 뒤에서 자세히 논의하도록 한다.

두 번째 방법은 [그림 1]처럼 개시자가 서비스 제공자에게 트랩도어(trapdoor)를 제공하여 서비스 제공자 스스로 사용자의 일치를 확인하는 것이다. 이러한 방법에서 $\mathbb{G}_1=\mathbb{G}_2$인 가정이 필요한데, 이러한 성질을 이용하여 연산의 효율을 높인 예는 Boneh와 Shacham의 논문에서 찾아볼 수 있다[13]. 개시자는 서비스 제공자에게 그룹 마스터키 $(\xi_1,\xi_2)$를 이용하여 트랩도어 $d=\xi_1/\xi_2$를 계산해서 전달한다. 서비스 제공자는 두 올바른 서명 $(\sigma,\sigma')$로부터 $t_1=T_1/T_1'$, $t_2=T_2/T_2'$,




〔그림 1〕 트랩도어를 이용한 사용자 일치의 확인

$t_3=T_3/T_3'$을 계산하고 $e(t_3,v)$이 $e(t_1,h)^d e(t_2,h)$와 일치하는 지 확인한다. 일치한다면 두 서명은 동일한 사용자에 의해서 작성된 것이다. 그러나 여전히 서비스 제공자는 서명이 누구에 의해서 작성되었는지는 알 수가 없다. 이 방법은 개시자와 발행자가 공모한다고 해도 서명에 대한 정보가 서비스 제공자로부터 노출되지 않는 이상 사용자의 익명성이 보장된다.

만약 두 서명이 일치한다면, 식 (1)에서 보다시피 $t_3=t_1^{\xi_1}t_2^{\xi_2}$인 관계가 성립한다. 따라서

$$\begin{aligned} e(t_3,v) &= e(t_1^{\xi_1}t_2^{\xi_2},v)=e((t_1^{\xi_1/\xi_2}t_2)^{\xi_2},v) \\ &= e(t_1^{\xi_1/\xi_2}t_2,v^{\xi_2})=e(t_1^{\xi_1/\xi_2},h)e(t_2,h) \\ &= e(t_1,h)^d e(t_2,h) \end{aligned} \quad (2)$$

이다. 이 방법은 두 번의 제곱 연산을 필요로 하는 첫 번째 방법에 비해서는 연산량이 크게 늘어난 것처럼 보인다. 그러나 서비스 제공자는 트랩도어를 이용하여 다음과 같이 사용자 서명으로부터 사용자의 식별자와 관련된 고유한 키를 계산해냄으로써, 두 개씩 뽑아 개시자에게 질의해야 하는 첫 번째 방법에 비해서 월등히 효율적으로 동작할 수 있다.

$$e(T_3,v)e(T_1^d T_2,h)^{-1}=e(A_i,v) \quad (3)$$

### 3.1.4 서비스 제공자의 권한 제한

첫 번째 방법에서는 개시자가 특정 서비스 제공자의 요청을 단순하게 거부함으로써 서비스 제공자가 서명들 사이에 연결을 더 이상 못하게 할 수 있다. 두 번째 방법은 이와 다르게 그룹 마스터키 $(\xi_1,\xi_2)$를 다른 것으로 바꾸고 트랩도어를 특정 서비스 제공자에게 알려주지 않는 방법을 사용해야 한다. 그룹 마스터키 $(\xi_1,\xi_2)$가 $(\xi_1',\xi_2')$로 바뀌게 되면 자연스럽게 그룹 공개키 $(h,u,v)$가 $(h',u',v')$바뀌게 된다. 만약 $h=h'$라면 $u'=h^{-\xi_1'}$, $v'=h^{-\xi_2'}$로 $(u,v)$만 $(u',v')$로 바뀌게 된다. 새로운 그룹 공개키는 그룹 멤버에게 알려지게 된다. 그러나 새로운 트랩도어를 받지 못한 서비스 제공자는 더 이상 서명 간 연결 확인을 할 수 없게 된다.

### 3.1.5 사용자 제외와 지역 연결성

짧은 그룹 서명 기법은 특정한 그룹 멤버를 그룹에서 제외(revocation) 시키기 위해서는 그룹 멤버 전체가 자신의 서명키를 RL(Revocation List)을 따라 새롭게 업데이트해야 한다. 이 경우, 이러한 과정의 부담은 차치하고서라도 지역 연결성의 지원에 있어서 문제가 발생할 수 있다. 사용자의 식별자가 업데이트 되어 서로 다른 경우 앞서 제시하였던 방법으로는 같은 사용자가 서명을 했다하더라도 그 사이에서 일치성을 찾을 수 없는 문제점이 있다. 이를 해결하기 위한 방법은 앞으로의 과제로 남겨둔다.

## 3.2 강한 무죄 입증성(Strong Exculpability)의 지원

### 3.2.1 가입 절차

그룹 매니저는 모든 그룹 멤버의 서명키 $(A_i,x_i)$ 만들어준다. 우선 특정 그룹 멤버를 위하여 $x_i$를 무작위로 선택하고 자신이 알고 있는 그룹 마스터키 $\gamma$를 이용하여 $A_i=g^{1/(x_i+\gamma)}$를 계산한다. 따라서 그룹 매니저는 특정한 그룹 멤버가 작성한 것처럼 서명을 생성할 수 있다. 이런 문제를 해결하기 위해서는 그룹 멤버가 그룹 매니저 모르게 선택하는 비밀키가 추가적으로 존재해야한다. 논문 BBS04에서는 $A_i^{x_i+\gamma}h_1^{y_i}=g_1$처럼 그룹 멤버가 추가적인 비밀키 $y_i$를 선택하는 가입 절차(join protocol)에 대해서 언급하고 있다. 이러한 가입 절차는 다음과 같이 동작한다.

1) 그룹 멤버 → 그룹 매니저: $Y,R$

그룹 멤버는 $y_i,r_y\in_R\mathbb{Z}_p$를 선택하고 $Y=h_1^{y_i}$,

$R=h_1^{r_R}$를 계산하여 이를 그룹 매니저에게 전송한다.

2) 그룹 매니저 → 그룹 멤버: $c$
그룹 매니저는 $c\in_R\mathbb{Z}_p$를 선택하고 이를 그룹 멤버에게 전송한다.

3) 그룹 멤버 → 그룹 매니저: $s_y$
그룹 멤버는 $s_y=r_y+cy_i$를 계산하고 이를 그룹 매니저에게 전송한다.

4) 그룹 매니저 → 그룹 멤버: $(A_i,x_i)$
그룹 매니저는 $R=h_1^{s_y}Y^{-c}$인지 확인하고, 올바르다면 $x_i\in_R\mathbb{Z}_p$를 선택하고 $A_i=(g_1/Y)^{1/(x_i+\gamma)}$를 계산하여 $(A_i,x_i)$를 그룹 멤버에게 전송한다.

위 과정은 그룹 매니저가 그룹 멤버가 제시한 $y_i$에 대해서 지식 증명(Proof of Knowledge)을 수행하는 과정을 포함하고 있다. 즉, $R$은 $y_i$의 증명을 위한 commit에 해당한다. 이러한 가입 절차가 끝나고 나면 그룹 멤버는 $(A_i,x_i,y_i)$를 서명키로 갖게 된다.

### 3.2.2 짧은 그룹 서명 기법의 지식 증명 구조

짧은 그룹 서명 기법에서 서명키 $(A_i,x_i)$가 직접적으로 연관된 부분은 $R_3$를 만드는 부분 이외에는 존재하지 않는다. $R_4$와 $R_5$는 도움 값(helper value) $\delta_1$과 $\delta_2$의 도입으로 인하여 필요해진 부분일 뿐이다. 영지식 증명 단계에서 $R_1,R_2,R_3,R_4,R_5$는 공개된 $T_1,T_2,T_3$에 따라서 생성된다. 예를 들자면, $T_1=u^{\alpha}$을 공개한 후 $\alpha$에 대한 지식 증명을 수행하기 위해서는 $R_1=u^{r_\alpha}$과 같은 commit를 생성해야 한다. 마찬가지로 $T_2$와 $R_2$는 $\beta$에 대한 지식 증명을 수행하기 위해 사용될 수 있다. $T_3=A_ih^{\alpha+\beta}$를 이용하여 $A_i$에 대한 증명을 시도하려고 할 경우 다음과 같은 고찰 과정을 거쳐야 한다. 이러한 방법은 기본적으로 $e(g_1,g_2)$에서부터 시작한다.

$$
\begin{aligned}
&e(g_1,g_2)\\
&=e(A_i^{x_i+\gamma},g_2)=e(A_i,g_2^{x_i+\gamma})=e(A_i,wg_2^{x_i})\\
&=e(A_i,wg_2^{x_i})e(h^{\alpha+\beta},wg_2^{x_i})e(h^{\alpha+\beta},wg_2^{x_i})^{-1}\\
&=e(A_ih^{\alpha+\beta},wg_2^{x_i})e(h^{\alpha+\beta},wg_2^{x_i})^{-1} \qquad (4)\\
&=e(T_3,wg_2^{x_i})e(h^{\alpha+\beta},wg_2^{x_i})^{-1}\\
&=e(T_3,w)e(T_3,g_2)^{x_i}e(h,w)^{-\alpha-\beta}\\
&\quad e(h,g_2)^{-x_i\alpha-x_i\beta}
\end{aligned}
$$

즉, $T_3$가 공개된 상황에서 $e(g_1,g_2)/e(T_3,w)=e(T_3,g_2)^{x_i}e(h,w)^{-\alpha-\beta}e(h,g_2)^{-x_i\alpha-x_i\beta}$인 관계가 성립한다. 이 수식에서 생성자에 해당하는 $e(g_1,g_2)/e(T_3,w)$, $e(T_3,g_2)$, $e(h,w)^{-1}$, $e(h,g_2)^{-1}$는 $T_3$가 공개된 상황에서는 모두 공개된 값이다. $x_i\alpha$를 $\delta_1$이라고 두고 $x_i\beta$를 $\delta_2$라고 하자. 이 관계를 만족하는 어떠한 $(x_i,\alpha,\beta,\delta_1,\delta_2)$에 대한 지식 증명을 위해서 $R_3\leftarrow e(T_3,g_2)^{r_x}e(h,w)^{-r_\alpha-r_\beta}e(h,g_2)^{-r_{\delta_1}-r_{\delta_2}}$과 같은 commit을 생성할 수 있다.

스크립트 $(T_1,R_1,c,s_\alpha)$는 $\alpha$를 위한 지식 증명에 사용되고, 스크립트 $(T_2,R_2,c,s_\beta)$는 $\beta$를 위한 지식 증명에 사용된다. 비슷하게 스크립트 $(T_3,R_3,c,(s_x,s_\alpha,s_\beta,s_{\delta_1},s_{\delta_2}))$는 $(x_i,\alpha,\beta,\delta_1,\delta_2)$에 대한 지식 증명에 사용된다. 그러나 이 스크립트는 단 하나의 $(x_i,\alpha,\beta,\delta_1,\delta_2)$에 대한 지식 증명이 아니라 식 (3)과 같은 관계를 만족하는 수많은 $(x_i,\alpha,\beta,\delta_1,\delta_2)$ 중에 하나를 알고 있다는 지식 증명이다. 따라서 스크립트 $(T_1,R_1,c,s_\alpha)$, $(T_2,R_2,c,s_\beta)$, $(T_3,R_3,c,(s_x,s_\alpha,s_\beta,s_{\delta_1},s_{\delta_2}))$를 동시에 만족하는 $(x_i,\alpha,\beta,\delta_1,\delta_2)$는 검증자(Verifier)의 입장에서 보면 증명자(Prover)는 유일한 $\alpha$와 $\beta$를 알고 있지만, 나머지 $(x_i,\delta_1,\delta_2)$에 대해서는 관계를 만족하는 수많은 쌍 중에 하나를 알고 있다는 것과 같다. 따라서 증명자는 $A_i^{x_i+\gamma}=g_1$인 관계에 놓인 $(A_i,x_i)$를 모른다고 하더라도 여기까지의 지식 증명을 통과할 수 있다. 여기서 $R_4$와 $R_5$는 각각 $(x_i,\delta_1)$, $(x_i,\delta_2)$ 쌍에 대한 지식 증명의 commit으로 사용되어, 검증자가 보기에 증명자는 유일한 $x_i$를 알고 있도록 믿게 한다. 유일한 $x_i$는 $A_i^{x_i+\gamma}=g_1$의 관계에 놓인 $A_i$가 $T_3$를 만드는데 사용되었음을 의미한다.

사실 $T_3$가 무작위로 선택될 경우 스크립트 $(T_3,R_3,c,(s_x,s_\alpha,s_\beta,s_{\delta_1},s_{\delta_2}))$를 만족하는 $(x_i,\alpha,\beta,\delta_1,\delta_2)$를 알아내는 것은 DLP(Discrete Logarithm Problem)를 해결하는 것만큼 매우 힘들다[13]. 공격자는 적절한 $(x_i,\alpha,\beta)$에 적합한 $T_3$를 선택해야하는데, 이 경우 $e(T_3,w)$와 $e(T_3,g_2)$를 원하는 값으로 두기 위해서 겹선형 쌍함수 $e$가 가역적(Reversible) 이어야 한다. 만약 겹선형 쌍함수 $e$가 가역적이라면, 공격자는 그룹 공개키로부터 유효한 그룹 비밀키를 생성해낼 수 있다.

### 3.2.3 새 비밀키를 위한 짧은 그룹 서명 기법의 변경

1) 간단한 짧은 그룹 서명 기법 변형 방법
기술의 편리함을 위해서 앞서 소개하였던 그룹 공

개키 $h_1$을 $g_3$로 다시 정의하자. BBS04에서의 제안을 따르면 $(A_i, x_i, y_i)$는 $A_i^{x_i+\gamma} g_3^{y_i} = g_1$의 관계에 있다. $y_i$에 대한 지식 증명을 위해서는 역시 $e(g_1, g_2)$와 $T_3$를 이용해야 한다.

$$\begin{aligned} & e(g_1, g_2) \\ &= e(A_i^{x_i+\gamma} g_3^{y_i}, g_2) = e(A_i^{x_i+\gamma}, g_2) e(g_3^{y_i}, g_2) \\ &= e(T_3, w) e(T_3, g_2)^{x_i} e(h, w)^{-\alpha-\beta} \\ & \quad e(h, g_2)^{-x\alpha-x\beta} e(g_3, g_2)^{y_i} \end{aligned} \tag{5}$$

원래의 기법과 마찬가지로 도움 값 $\delta_1$와 $\delta_2$를 두면, $R_3 \leftarrow e(T_3, g_2)^{r_x} e(h, w)^{-r_\alpha - r_\beta} e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}} e(g_3, g_2)^{r_y}$이 된다. 추가적으로 $s_y = r_y + cy_i$가 필요로 해진다. 문제는 이렇게 짧은 그룹 서명 기법의 틀을 유지하면서 단순하게 $y_i$만을 추가할 경우, 서명은 $\sigma \leftarrow (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_y, s_{\delta_1}, s_{\delta_2})$와 같이 되어 단지 $s_y$ 하나를 추가한 것과 같이 효율적이 되지만, 앞서 살펴본 짧은 그룹 서명 기법의 지식 증명 구조에서와 같이 $(x_i, y_i)$ 값이 검증자의 입장에서 유일해지지 않는다. 따라서 $(x_i, y_i)$와 관련된 새로운 지식 증명이 최소 하나 이상 필요하다.

2) 사용자 식별자 추가 기법

이 논문에서 제안하는 방법은 [그림 2]와 같이 $B_i^{x_i+\gamma} = g_4$와 같은 관계식을 추가하여 지식 증명을 수행하는 것이다. 즉, 그룹 매니저는 $x_i$와 함께 $B_i$를 생성하여 이를 서명키로 그룹 멤버에게 주는 것이다. 그룹 멤버는 자신이 선택한 $y_i$이외에 $B_i$에 대한 지식 증명을 수행해야 한다. 이에 따라 새로운 그룹 마스터키 $(\xi_3, \xi_4)$와 그룹 공개키 $(h_2, u_2, v_2, g_3, g_4) \in \mathbb{G}_1$가 필요로 해졌다. ($(\xi_3, \xi_4)$는 $(\xi_1, \xi_2)$과 동일한 것을 사용해도 된다.)

이 방법은 $R_6, R_7, R_8, R_9, R_{10}$과 관련하여 $x_i$가 유일해지므로, $R_1, R_2, R_3, R_4, R_5$와 관련된 $y_i$가 유일해질 수 있는 구조다. 위는 $B_i$의 추가에 따라 단순하게 연산을 확장한 것으로, 연산의 효율성을 위한다면 $T_1, T_2$(또는 $T_4, T_5$)를 삭제하고, 그에 따라 $R_1, R_2, R_4, R_5$(또는 $R_6, R_7, R_9, R_{10}$)을 삭제할 수 있다.(관련된 서명 확인 작업도 삭제된다.) $T_1, T_2$(또는 $T_4, T_5$)를 삭제할 경우, 서명의 길이는 짧아지게 되지만 그룹매니저는 그룹 멤버를 $B_i$(또는 $A_i$)로만 식별할 수 있다. 단 이 경우, $T_3 = T_6$일 때 $(x_i, y_i, \delta_1, \delta_2)$ 쌍의 수가 무한해질 수 있으므로, 반드시 이를 확인해야 한다.

한편, Delerablée와 Pointcheval은 $A_i$를 두 번

Setup

$gmk = (\xi_1, \xi_2, \xi_3, \xi_4, \gamma)$,
$gpk = \{h_1, u_1, v_1, h_2, u_2, v_2, g_1, g_2, g_3, g_4, w\}$
$gsk[i] = (A_i \leftarrow (g_1 g_3^{-y_i})^{1/(x_i+\gamma)}, B_i \leftarrow g_4^{1/(x_i+\gamma)}, x_i, y_i)$

Sign $\sigma \leftarrow sign(m, gpk, gsk[i])$

$\alpha, \beta \leftarrow_R \mathbb{Z}_p$, $\delta_1 \leftarrow \alpha x_i$, $\delta_2 \leftarrow \beta x_i$, $r_\alpha, r_\beta, r_x, r_y, r_{\delta_1}, r_{\delta_2} \leftarrow_R \mathbb{Z}_p$

$T_1 \leftarrow u_1^\alpha$, $T_2 \leftarrow v_1^\beta$, $T_3 \leftarrow A_i h_1^{\alpha+\beta}$, $T_4 \leftarrow u_2^\alpha$, $T_5 \leftarrow v_2^\beta$,
$T_6 \leftarrow B_i h_2^{\alpha+\beta}$,

$R_1 \leftarrow u_1^{r_\alpha}$, $R_2 \leftarrow v_1^{r_\beta}$, $R_4 \leftarrow T_1^{r_x} u_1^{-r_{\delta_1}}$, $R_5 \leftarrow T_2^{r_x} v_1^{-r_{\delta_2}}$, $R_6 \leftarrow u_2^{r_\alpha}$,
$R_7 \leftarrow v_2^{r_\beta}$, $R_9 \leftarrow T_4^{r_x} u_2^{-r_{\delta_1}}$, $R_{10} \leftarrow T_5^{r_x} v_2^{-r_{\delta_2}}$,

$R_3 \leftarrow e(T_3, g_2)^{r_x} e(h_1, w)^{-r_\alpha - r_\beta} e(h_1, g_2)^{-r_{\delta_1} - r_{\delta_2}}$,
$e(g_3, g_2)^{r_y}$

$R_8 \leftarrow e(T_6, g_2)^{r_x} e(h_2, w)^{-r_\alpha - r_\beta} e(h_2, g_2)^{-r_{\delta_1} - r_{\delta_2}}$

$c \leftarrow H(m, T_1, T_2, T_3, T_4, T_5, T_6, R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8, R_9, R_{10})$

$s_\alpha \leftarrow r_\alpha + c\alpha$, $s_\beta \leftarrow r_\beta + c\beta$, $s_x \leftarrow r_x + cx_i$, $s_y \leftarrow r_y + cy_i$,
$s_{\delta_1} \leftarrow r_{\delta_1} + c\delta_1$, $s_{\delta_2} \leftarrow r_{\delta_2} + c\delta_2$

return $\sigma \leftarrow (T_1, T_2, T_3, T_4, T_5, T_6, c, s_\alpha, s_\beta, s_x, s_y, s_{\delta_1}, s_{\delta_2})$

Verification $1/0 \leftarrow verify(m, \sigma, gpk)$

(if $T_3 = T_6$ then return 0)

$\widetilde{R}_1 \leftarrow u_1^{s_\alpha} T_1^{-c}$, $\widetilde{R}_2 \leftarrow v_1^{s_\beta} T_2^{-c}$, $\widetilde{R}_4 \leftarrow T_1^{s_x} u_1^{-s_{\delta_1}}$,

$\widetilde{R}_5 \leftarrow T_2^{s_x} v_1^{-s_{\delta_2}}$, $\widetilde{R}_6 \leftarrow u_2^{s_\alpha} T_4^{-c}$, $\widetilde{R}_7 \leftarrow v_2^{s_\beta} T_5^{-c}$,

$\widetilde{R}_9 \leftarrow T_4^{s_x} u_2^{-s_{\delta_1}}$, $\widetilde{R}_{10} \leftarrow T_5^{s_x} v_2^{-s_{\delta_2}}$,

$\widetilde{R}_3 \leftarrow e(T_3, g_2)^{s_x} e(h_1, w)^{-s_\alpha - s_\beta} e(h_1, g_2)^{-s_{\delta_1} - s_{\delta_2}}$,
$e(g_3, g_2)^{s_y} (e(T_3, w)/e(g_1, g_2))^c$

$\widetilde{R}_8 \leftarrow e(T_6, g_2)^{s_x} e(h_2, w)^{-s_\alpha - s_\beta} e(h_2, g_2)^{-s_{\delta_1} - s_{\delta_2}}$,
$(e(T_6, w)/e(g_4, g_2))^c$

if $c = H(m, T_1, T_2, T_3, T_4, T_5, T_6, \widetilde{R}_1, \widetilde{R}_2, \widetilde{R}_3, \widetilde{R}_4, \widetilde{R}_5, \widetilde{R}_6, \widetilde{R}_7, \widetilde{R}_8, \widetilde{R}_9, \widetilde{R}_{10})$
then return 1
else return 0

Open $A, B \leftarrow open(m, \sigma, gpk, gmk)$

if $1 = verify(m, \sigma, gpk)$
then return $A \leftarrow T_3/(T_1^{\xi_1} T_2^{\xi_2})$, $B \leftarrow T_6/(T_4^{\xi_3} T_5^{\xi_4})$

[그림 2] 강한 무죄 입증성을 위한 사용자 식별자 추가 기법

선형 암호화 하는 방법에 대해서 언급하였다[14]. 이 방법은 제안하는 기법과 유사해 보이지만, $(x_i, y_i)$의 유일함을 증명하는 방법이 다르다. 이 방법에서도 사용자의 식별자를 서로 다른 그룹 공개키로 암호화 시

켜야 하므로 새로운 그룹 마스터키 $(\xi_3,\xi_4)$와 그룹 공개키 $(h_2,u_2,v_2,g_3)\in\mathbb{G}_1$가 필요로 해진다. (역시 $(\xi_3,\xi_4)$는 $(\xi_1,\xi_2)$과 동일한 것을 사용해도 된다.) 이 방법에서는 $R_1,R_2,R_3,R_4,R_5$와 관련된 $(x_i,y_i)$에서도 $R_6,R_7,R_8,R_9,R_{10}$와 관련된 $(x_i,y_i)$에서도 각각은 유일함을 보장하지 않는다. 그러나 동시에 만족할 경우, 이원일차 방정식처럼 유일함을 보장하게 되는 구조를 가지고 있다. 여기서도 $T_4,T_5$와 그에 따른 $R_6,R_7,R_9,R_{10}$을 삭제할 수 있는데, 이때에는 $T_3=T_6$인 경우, $(x_i,y_i,\delta_1,\delta_2)$ 쌍의 수가 무한해질 수 있으므로, 반드시 이를 확인해야 한다.

### 3.2.4 서명 길이와 관련된 짧은 그룹 서명 기법의 고찰

강한 무죄 입증성을 위하여 그룹 멤버가 선택한 $y_i$에 따라서 위에서 살펴본 바와 같이 서명의 길이가 683비트(짧은 그룹 서명 기법 1533비트)까지 늘어날 수 있음을 보였다. 서명의 길이에 직접적으로 영향을 미치는 요소는 그룹 멤버의 식별자를 암호화 하는 방법이다. 짧은 그룹 서명 기법에서는 선형 암호화 기법을 사용함으로써, 두 가지 난수 $\alpha$와 $\beta$에 대한 증명을 수행해야만 했고, $T_3$와 $e(g_1,g_2)$의 구조적인 문제로 인하여 도움 값 $\delta_1,\delta_2$가 필요로 해졌다.

Delerablée와 Pointcheval은 선형 암호화 기법의 사용을 비판하며 이중(double) ElGamal 암호화를 사용하였다[14]. 동일한 $A_i$를 두 번 ElGamal 암호화함으로써 $y_i$를 추가하였음에도 짧은 그룹 서명 기법과 동일한 서명 길이를 보일 수 있다는 것이다. 그러나 이렇게 이중 ElGamal 암호화를 사용하기 위해서는 XDH(eXternal Diffie-Hellman) 가정을 추가로 필요로 하는 문제가 있다. XDH 가정은 겹선형 쌍함수 $e:\mathbb{G}_1\times\mathbb{G}_2\rightarrow\mathbb{G}_T$ 상에서 $\mathbb{G}_1$에서의 DDH (Decisional Diffie-Hellman) 문제가 어렵다는 것이다. 즉, $\mathbb{G}_1$에서 $\mathbb{G}_2$으로의 isomorphism이 존재하지 않는다.

만약, 이러한 가정이 없고, $\mathbb{G}_1=\mathbb{G}_2$인 상황에서 ElGamal 암호화를 사용하게 되면 그룹 멤버의 연결성에 치명적인 문제가 발생한다. 예를 들자면, 어떠한 서명 $\sigma$에서 $T_1=u^{\alpha},T_2=Ah^{\alpha}$(ElGamal 암호화)라고 하고 같은 그룹 멤버가 생성한 다른 서명 $\sigma'$에서 $T_1'=u^{\alpha'},T_2'=Ah^{\alpha'}$라고 하면

$$e(T_1/T_1',h)=e(u^{\alpha-\alpha'},h)=e(u,h^{\alpha-\alpha'})=e(u,T_2/T_2') \quad (6)$$

인 관계가 성립한다. 즉, 그룹 멤버 누구나 서명의 연결성을 확인해볼 수 있게 된다. XDH 가정은 대부분의 그룹 공개키를 그룹 $\mathbb{G}_1$에 존재하게 하여 위와 같은 연산 자체가 불가능하도록 한다. 따라서 XDH 가정이 없을 경우 BBS04에서처럼 어쩔 수 없는 선택으로 선형 암호화 기법을 사용해야만 할 것으로 보인다.

### 3.2.5 해결이 필요한 문제

강한 무죄 입증성을 갖기 위해서 사용자에게 어떠한 비밀을 선택하도록 하는 것은 사용자 서명키의 철폐에 문제를 만들어낸다. BBS04에서 사용자의 철폐는 전체가 그룹 공개키와 자신의 식별자를 바꾸지만, 해당 사용자는 자신의 식별자를 바꿀 수 없도록 하여 이루어진다. 한편, 사용자의 식별자 $A$는 그룹 매니저가 선택하는 사용자 서명키 $x$를 그룹 매니저의 비밀키 $\gamma$로 서명 해놓은 형태이다[16]. 여기에 강한 무죄 입증성을 위하여 사용자가 선택한 서명키 $y$가 더해지면 위와 같이 해당 사용자의 식별자 $A$를 동일한 방식으로 바꾸는 것이 불가능해진다. 사용자의 식별자가 단순히 $x$와 $\gamma$에 관련되어 생성된 값이 아니라, $y$값 또한 반영되어야 하지만, 이 값은 해당 사용자를 제외한 다른 사람들은 알 수 없기 때문에, $y$값을 RL에 반영시킬 방법이 없다. 결론적으로, $A^{x+\gamma}g_3^{y}=g_1$과 같은 관계식을 가지고, 사용자 철폐 절차(revocation)를 마련하기란 매우 힘들다. 이는 앞으로 반드시 해결되어야할 문제로 보인다.

## IV. 비교 분석 및 토의

### 4.1 지역 연결성 지원

서비스 제공자에게 주어지는 지역 연결성은 사용자에게 주어지는 최대한의 익명성 일부분을 희생하는 것과 같다. 이와는 반대로 지역 연결성은 사용자가 스스로 서명간의 연결성을 주장할 때, 자신의 식별자를 노출시키지 않고도 이를 확인해줄 수 있는 기능성을 제공하는 측면도 있다. 즉, 극단적인 상황에서 최소한의 익명성을 보장해주는 역할도 수행한다고 볼 수 있다.

서비스 제공자에게 사용자들에 대한 지역 연결성이 주어졌다고 하더라도 이를 이용하기 위해서는 현실적인 문제가 존재한다. 서비스 제공자가 임의로 두 서명을 선택하여 연결성을 확인하려고 하는 것은 서명의 수나 사용자의 수가 많지 않았을 때 가능하다. 따라서

서비스 제공자가 자신이 확보한 서명들의 연결성을 알고 싶다면 개시자에게 도움을 요청하는 것이 효율적이다. 사용자 식별자의 노출을 피하고 싶다면 선형 암호화가 갖는 순응성(malleability)을 이용하는 것도 한 방법이 될 수 있다.

이 논문에서는 짧은 그룹 서명 기법을 실제로 적용한 변형된 응용 환경에서 외부로 사용자의 식별자를 노출시키지 않으면서 서비스 제공자에게 지역 연결성을 지원하는 두 가지 방법에 대해서 제안하였다. 하나는 서비스 제공자가 두 서명 $\sigma,\sigma'$에 들어 있는 $T_1, T_2, T_3$와 $T_1', T_2', T_3'$을 변형하여 개시자에게 일치성을 물어보는 것이었으며, 다른 하나는 개시자로부터 받은 트랩도어를 이용하여 서비스 제공자가 스스로 일치성을 확인하는 것이었다. 첫 번째 방법의 경우, 연산량 자체는 두 번째 방법보다 효율적이다. 나누기 연산 3번과 제곱 3번으로 일치성을 확인하는 것이 가능했다. 두 번째 방법에서는 나누기 연산 3번과 제곱 1번 이외에 겹선형 쌍함수 연산을 3번 해야만 했다.

그러나 첫 번째 방법의 경우 일치성을 물어볼 때 두 서명의 작성자가 일치할 경우에는 별다른 문제가 발생하지 않지만 일치하지 않을 경우, 개시자에게 $A_i/A_j$에 대한 정보를 전송하게 된다. 개시자 스스로는 사용자의 식별자에 대한 정보가 없기 때문에 이러한 정보를 아는 것은 무의미해진다. 그러나 만약 개시자가 발행자의 도움을 받을 수 있는 경우, 개시자는 사용자의 식별자를 알아낼 수 있다. $A_i/A_j$로부터 추출해낼 수 있는 모든 $(A_i, A_j)$ 쌍은 $p(\simeq 2^{171})$개가 되어 추적이 불가능할 것 같아 보이지만, 실 사용자의 수는 이보다 훨씬 작을 것이므로 모든 가능한 $A_i/A_j$을 미리 계산해 놓은 발행자로부터 개시자는 높은 확률로 올바른 $(A_i, A_j)$ 쌍을 얻어낼 수 있다. 트랩도어를 사용하는 두 번째 방법은 이러한 문제는 발생하지 않는다.

### 4.2 강한 무죄 입증성 지원

강한 무죄 입증성을 위해서, 이 논문에서 제시한 방법은 $x_i$나 $y_i$에 대한 관계식을 명시적으로 하나 더 만들자는 것이다. $x_i$나 $y_i$는 기본적으로 서로 독립적으로 선택되기 때문에 이에 대한 관계식을 하나 더 만들고 이를 그룹 서명 기법에 도입해도 문제가 되지 않는다. 즉, $B_i^{x_i+\gamma}=g_3$와 같은 $x_i$에 관련된 관계식 말고도 $g_3^{y_i}=g_4$와 같이 $y_i$와 관련된 관계식을 사용할 수도 있다는 의미와 같다. 후자와 같은 경우 $g_4$를 공개할 수 없기 때문에, 직접적인 도입이 쉽지 않아 보이지만, 이미 Lysyanskaya 등에 의해서 연구되었던 내용을 응용할 수 있다[4].

결과적으로는 이 논문에서 제안하는 방법은 Delerablée와 Pointcheval이 그들의 논문 [14]에서 짧게 언급했던 방법과 유사한 결과(연산량, 서명길이)를 가져온다. 다른 점은 제안하는 방법은 서명에 두 개의 요소를 더 더하면 서명이 동시에 두 가지의 사용자 식별자를 갖도록 하는 것이 가능하다. 만약, 계층적인 구조를 가져야 하는 익명 시스템이 있고 사용자가 강한 무죄 입증성을 가져야 하는 경우라면 Delerablée와 Pointcheval이 생각하였던 방법보다 효율적으로 동작 가능하다.

[표 2]는 짧은 그룹 서명 기법과 이를 이용한 변형들, 그리고 Delerablée와 Pointcheval의 논문에서 제시한 XDH 가정을 기반으로 이중 ElGamal 기법을 사용한 방법에 대한 비교이다.

최적화를 위하여 증명자와 검증자는 사전 계산(pre-computation)을 해두어야 한다. 그룹 서명 기법에서 서명을 위해 필요한 연산량은 대부분 사전에 준비 해둘 수 있다. 그리고 준비된 값을 이용하여 체크섬 $c=H(m,\cdots)$를 만들고, 몇 번의 곱하기와 더하기

[표 2] 강한 무죄 입증성을 위한 BBS04의 변형 및 DP06의 비교(*: 실시간 연산량)

| | | BBS04[7] | BBS04 기반 변형 방법 | | | DP06[15] |
|---|---|---|---|---|---|---|
| | | | III.2.3.(1) (간단 변형) | 이중 선형 암호화[15] | III.2.3.(2) (제안) | |
| 수학적 가정 | | q-SDH, LDDH | | | | q-SDH,XDH |
| 서명 길이 (단위: 요소) | | 9 | 10 | 11 | 11 (13) | 9 |
| 연산 | 서명 | 12 exp | 13 exp | 19 exp | 19 exp | 11 exp |
| | 서명 확인* | 13 exp<br>1 pairing | 13 exp<br>1 pairing | 18 exp<br>2 pairing | 19 exp<br>2 pairing | 12 exp<br>1 pairing |
| 강한 무죄 입증성 | | 지원하지 않음 | 지원 | 지원 | 지원 | 지원 |
| 유일 서명키 증명 | | 증명 | 증명 못함 | 증명 | 증명 | 증명 못함 |
| 식별자의 수 | | 1 | 1 | 1 | 1 (2) | 1 |

연산 이후에 서명 값을 만들 수 있다. 서명 확인을 위해서도 몇몇 부분의 연산은 미리 해둘 수 있다. 예를 들자면, 이 논문에서 제안하는 기법에서 서명 확인을 위해 필요한 $\widetilde{R_3}$를 구하는 연산은 $\widetilde{R_3} \leftarrow e(T_3,g_2)^{s_z} \cdots (e(T_3,w)/e(g_1,g_2))^c$와 같은데 이를 $\widetilde{R_3} \leftarrow e(T_3,g_2^{s_z}w^{-c}) \cdots e(g_1,g_2)^{-c}$와 같이 바꾸어 쓰면 단지 1번이 겹선형 쌍 함수의 실행으로 원하는 값을 얻을 수 있다. ($e(g_1,g_2)$ 등은 미리 캐시 된다.)

대부분의 응용 환경에 있어서 위와 같은 연산량은 크게 문제가 되지 않는다. 사용자(사람)는 스스로 계산하는 것은 불가능하기 때문에 컴퓨터에서 실행되는 프로그램이나 USB 하드웨어 토큰과 같은 연산 장치에 대신 맡겨야 한다. 서명 길이 또한 원래의 1533비트에 비해서 크게 늘어난 서명 길이를 가지고 있지만, 최근의 통신 환경을 고려한다면 여전히 크게 문제되지 않아 보인다. 그러나 여전히 몇몇 응용 환경에서 위와 같은 연산량과 서명 길이는 문제가 될 수 있을 것이므로, 이는 해당 응용 환경에서 고민 해봐야 할 문제로 남겨둔다.

### 4.3 보안성 분석

(1) 지역 연결성을 갖는 짧은 그룹 서명 기법

정리 1. 만약 선형 암호화를 비밀키 $\xi_1,\xi_2$이외에 다른 값으로는 풀 수 없다면, $d=\xi_1/\xi_2$를 가지고 있다 하더라도 서로 다른 사용자가 생성한 짧은 서명으로부터 사용자의 식별자를 알아낼 수 없다.

증명(개요). 앞서 정의한 관계대로 선택된 $u,v,h,d$가 주어졌을 때, 비밀키 $\xi_1,\xi_2$를 알아낼 수 있는 다항식 시간 알고리즘 A가 있다면, DLP 문제를 해결할 수 있다. (DLP 문제 $(g,g^x)$가 있을 때, 시뮬레이터 B는 무작위로 선택한 $k_1,k_2 \in_R \mathbb{Z}_p$를 이용하여 $h \leftarrow (g^x)^{k_1k_2}$, $u \leftarrow g^{k_1}$, $v \leftarrow g^{k_2}$, $d \leftarrow k_2/k_1$를 생성하고 이를 알고리즘 A에게 넘겨준다. 알고리즘 A는 다항식 시간 안에 $\xi_1,\xi_2$를 계산할 수 있고, 시뮬레이터 B는 이로부터 DLP 문제의 $x=\xi_1/k_2=\xi_2/k_1$을 구할 수 있다.) 따라서 $d$를 안다고 하더라도 $\xi_1,\xi_2$는 계산할 수 없고, 사용자의 식별자를 알아낼 수 없다. □

당연하게도, BBS04에서 설명된 CPA-fully-anonymous 성질은 $d$를 아는 사용자(서비스 제공자)에게는 유효하지 않다. 왜냐하면 $d$를 알면 당연히 CPA 실험을 통과할 수 있기 때문이다.

(2) 강한 무죄 입증성을 갖는 짧은 그룹 서명 기법

정리 2.1. 강한 무죄 입증성을 갖는 짧은 그룹 서명 기법은 정당(correct)하다.

이는 BBS04에서의 증명과 동일하게 진행될 수 있어 이 논문에서는 생략하도록 한다.

정리 2.2. 선형 암호화 기법이 $\mathbb{G}_1$에서 $(t',\epsilon')$-의미론적으로 안전하다면, 강한 무죄 입증성을 갖는 그룹 서명 기법 또한 $(t,q_H,\epsilon)$-CPA-fully- anonymous이다. ($t=t'-q_HO(1)$, $\epsilon=\epsilon'$, $q_H$는 랜덤 오라클로의 질의 횟수)

증명(개요). 그룹 서명 기법을 $(t,q_H,\epsilon)$에 깰 수 있는 알고리즘 A를 가정하고, 선형 암호화 기법의 CPA 안전을 깰 수 있는 알고리즘 B를 가정하자. 알고리즘 B는 알고리즘 A로부터 생성될 수 있다. 알고리즘 B는 확인자 C로부터의 입력 $u,v,h$를 이용하여 자신이 무작위로 고른 $k \in \mathbb{Z}_p$를 이용하여 $u_1 \leftarrow u$, $v_1 \leftarrow v$, $h_1 \leftarrow h$, $u_2 \leftarrow u^k$, $v_2 \leftarrow v^k$, $h_2 \leftarrow h^k$를 생성한다. 나머지 공개키와 비밀키 $g_1,g_2,g_3,g_4,w,\gamma$를 앞서 설명한 바와 같이 생성하여 이 모두를 알고리즘 A에게 넘겨준다. 알고리즘 A는 $A_o,A_1,M$를 알고리즘 B에게 넘겨준다. 알고리즘 B는 확인자 C에게 $A_o,A_1$을 넘겨준다. 확인자 C는 자신이 알고 있는 비밀키 $\xi_1,\xi_2$를 이용하여 $A_o,A_1$중 하나를 랜덤하게 선택하고 이를 암호화하여 $(T_1,T_2,T_3)_b$를 알고리즘 B에게 전달한다. 알고리즘 B는 $T_4 \leftarrow T_1^k$, $T_5 \leftarrow T_2^k$, $T_6 \leftarrow T_3^k$로 생성하고 나머지 $c,s_\alpha,s_\beta,s_x,s_y,s_{\delta_1},s_{\delta_2} \in \mathbb{Z}_p$를 선택하고 BBS04에서와 같은 요령으로 $R_1,\cdots,R_{10}$을 계산한다. 알고리즘 B는 랜덤 오라클 $H$를 $c=H(M,T_1,\cdots,T_6,R_1,\cdots,R_{10})$의 관계가 되도록 하는데, 이것이 실패할 확률은 사실상 존재하지 않는다. 알고리즘 A의 질의에 대한 응답은 위와 같이 만든 서명 $\sigma \leftarrow (T_1,\cdots T_6,c,s_\alpha,s_\beta,s_x,s_y,s_{\delta_1},s_{\delta_2})$로 한다. 그러면 알고리즘 A는 다항식 시간 안에 $b'$을 대답할 것이고 알고리즘 B는 이를 도전자 C에게 전송할 수 있다. 따라서 만약 알고리즘 A가 $(t,q_H,\epsilon)$로 강한 무죄 입증성을 갖는 그룹 서명 기법이 CPA 안전하지 않음을 보일 수 있다면, 알고리즘 B는 확률 $\epsilon'=\epsilon$와 시간 $t'=t+q_HO(1)$으로 선형 암호화가 CPA 안전하지 않음을 보일 수 있다. □

정리 2.3. 만약 SDH가 $\mathbb{G}_1,\mathbb{G}_2$에서 $(q,t',\epsilon')$ 어렵다고 한다면, 강한 무죄 입증성을 갖는 그룹 서명은 $(t,q_H,q_S,n,\epsilon)$-fully-traceable하다. ($n=q-1$, $\epsilon=4n\sqrt{2\epsilon' q_H}$

$+n/p$, $t=\Theta(1)t'$, $q_H$는 적대적 알고리즘 A의 해시 오라클 질의 횟수, $q_S$는 서명 오라클 질의 횟수)

증명(개요). $n$개의 SDH instance를 $n-1$개의 $(A_i,x_i,y_i)$로 만드는 방법이 가능하다면, 증명 내용과 절차는 BBS04와 동일하게 이루어질 수 있다. BBS04에서는 논문 [16]에서 SDH instance를 $(A_i,x_i)$로 변화하는 방법을 사용한 뒤, forking lemma에 따라서 증명하고 있다. $n$개의 SDH instance $(g_1',g_2',(g_2')^{\gamma},(g_2')^{\gamma^2},\cdots,(g_2')^{\gamma^n})$가 있다고 하자. $m<n$에 대해서 $x_1,x_2,\cdots,x_m,y,k\in\mathbb{Z}_p^*$를 선택하자. $f(\gamma)=\Pi_{i=1}^{m-1}(\gamma+x_i)$라고 정의하면, $f(\gamma)=\Sigma_{i=0}^{m-1}\alpha_i\gamma^i$로 다시 쓸 수 있다. $f_i(\gamma)=f(\gamma)/(\gamma+x_i)$로 정의하면 마찬가지로 $f_i(\gamma)=\Sigma_{j=0}^{m-2}\beta_j\gamma^j$로 다시 쓸 수 있다. $g_5'\leftarrow(g_2')^k$로 하면 앞선 $n$개의 SDH instance로부터 $(g_1',g_5',(g_5')^{-y\gamma},(g_5')^{-y\gamma^2},\cdots,(g_5')^{-y\gamma^n})$를 구할 수 있다. 이로부터 다음과 같이 공개키 $g_1,g_2,g_3,w$와 $(A_i,x_i,y_i)$를 생성할 수 있다.

$$g_2\leftarrow\prod_{i=0}^{n-1}((g_2')^{\gamma^i})^{\alpha_i}=(g_2')^{f(\gamma)},g_1\leftarrow\psi(g_2)$$

$$g_5\leftarrow\prod_{i=0}^{n-1}((g_5')^{\gamma^i})^{\alpha_i}=(g_5')^{f(\gamma)},g_3\leftarrow\psi(g_5)$$

$$w\leftarrow\prod_{i=1}^{n}((g_2')^{\gamma^i})^{\alpha_{i-1}}=(g_2')^{\gamma f(\gamma)}=g_2^{\gamma}$$

$$A_i^*\leftarrow\prod_{j=0}^{n-2}((g_5')^{-y\gamma^j}(g_2')^{\gamma^j})^{\beta_j}=((g_5')^{-y}g_2')^{f_i(\gamma)}$$
$$=(g_4^{-y}g_2)^{1/(x_i+\gamma)}$$

$$A_i\leftarrow\psi(A_i^*)$$

나머지 증명 과정은 $y$와 크게 관련이 없이 랜덤 오라클 모델을 이용하여 증명하고 있다. 따라서 $n$개의 SDH instance를 $n-1$개의 $(A_i,x_i,y_i)$로 성공적으로 바꿀 수 있으며, BBS04의 증명을 그대로 따라 위와 같은 결과를 얻을 수 있다. □

위 과정에서 모든 $y_i$는 동일해야 변환을 성공할 수 있는데, 그렇다고 하더라도 $y$는 사용자에 의해서 선택되는 값이므로 모든 $(A_i,x_i,y)$는 타당한 서명키 값이라고 할 수 있다.

## V. 결론 및 앞으로의 과제

짧은 그룹 서명 기법은 익명성을 보장하는 효율적인 그룹 서명 기법으로 잘 알려져 있다. 이 논문에서는 짧은 그룹 서명 기법이 실제 응용환경에서 이용되기 위하여 필요한 권한 분할과 그와 더불어 필요해지는 지역 연결성을 지원하는 방법과 BBS04에서 구체적으로 다루어지지 않았던 강한 무죄 입증성을 위해 어떠한 절차를 거쳐야 하는지에 대해서 제안하였다. 또한 그와 더불어 다른 가능한 기법들도 함께 소개하고 이를 고찰함으로써 실제 응용 환경에 있어 추가적으로 더 필요로 하는 요건이나 속성이 발생하였을 때, 짧은 그룹 서명 기법을 효과적으로 변형하여 사용할 수 있도록 배려하였다. 또한 제안된 기법들은 타당한 보안성 증명을 통하여 그 안전성을 확인할 수 있었다.

그룹 멤버의 제외로 인하여 남겨진 그룹 멤버들의 그룹 공개키와 그룹 멤버의 서명키 변경은 지역 연결성 지원에 있어서 또 다른 문제를 가져왔고, 이는 앞으로의 과제로 남겨두었다. 물론, 짧은 그룹 서명 기법에서 그룹 멤버의 제외로 인하여 발생하는 과도하게 많은 연산으로부터 자유로울 수 있는 방법이 우선 마련되어야 할 것이다. 또한 짧은 서명 기법을 변형하여 강한 무죄 입증성을 갖게 할 경우, 사용자 서명키의 철폐 문제가 있는 것으로 보여 이를 해결할 수 있는 방법이 필요하다.

## 참 고 문 헌


[1] D. Chaum and E. van Heyst, "Group signatures," Proc. of EUROCRYPT '91, LNCS 547, pp. 257-265, 1991.

[2] R. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," Proc. of ASIACRYPT, LNCS 2248, pp. 552-565, 2001.

[3] A. Kiayias, Y. Tsiounis, and M. Yung, "Traceable Signatures," Proc. of EUROCRYPT 2004, LNCS 3027, pp. 571-589, 2004.

[4] A. Lysyanskaya, R.L. Rivest, A. Sahai, and S. Wolf, "Pseudonym systems," Proc. of Selected Areas of Cryptography (SAC 1999), LNCS 1758, pp. 184-199, 1999.

[5] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," Proc. of EUROCRYPT 2001, LNCS 2045, pp. 93-118,

2001.

[6] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," Proc. of Crypto 2004, LNCS 3152, pp. 56-72, 2004.

[7] D. Boneh, X. Boyen, and H. Shacham, "Short Group Signatures," Proc. of Crypto 2004, LNCS 3152, pp. 41-55, 2004.

[8] P.P. Tsang, M.H. Au, A. Kapadia, and S.W. Smith, "Blacklistable Anonymous Credentials: Blocking Misbehaving Users Without TTPs," Proc. of ACM Conference on Computer and Communications Security (ACM CCS 2007), pp. 72-81, Oct. 2007.

[9] V. Benjumea, J. Lopez, J.A. Montenegro, and J.M. Troya, "A First Approach to Provide Anonymity in Attribute Certificates," Proc. of Public Key Cryptography (PKC 2004), LNCS 2947, pp. 402-415, 2004.

[10] 양대헌, 이경희, "추적 가능한 가명 은밀 획득 프로토콜," 정보보호학회논문지, 16(5), pp. 113-118, 2006년 10월.

[11] 강전일, 양대헌, 이경희, "OT(Oblivious Transfer) 기반의 조건부 추적이 가능한 가명 프로토콜," 정보보호학회논문지, 19(1), pp. 33-42, 2009년 2월.

[12] T. Kwon, J.H. Cheon, Y. Kim, and J. Lee, "Privacy Protection in PKIs: A Separation-of-Authority Approach," Proc. of Workshop on Information Security Applications (WISA 2006), LNCS 4298, pp. 297-311, 2007.

[13] D. Boneh and H. Shacham, "Group Signatures with Verifier-Local Revocation," Proc. of ACM Conference on Computer and Communications Security (ACM CCS 2004), pp. 168-177, Oct. 2004.

[14] D. Chaum, E. van Heyst, and B. Pfitzmann, "Cryptographically strong undeniable signatures, unconditionally secure for the signer," Proc. of Crypto 92, LNCS 576, pp. 470-484, 1992.

[15] C. Delerablée and D. Pointcheval, "Dynamic Fully Anonymous Short Group Signatures," Proc. of VIETCRYPT 2006, LNCS 4341, pp. 193-210, 2006.

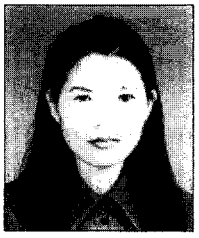
[16] D. Boneh and X. Boyen, "Short Signatures Without Random Oracles," Proc. of EUROCRYPT 2004, LNCS 3027, pp. 56-73, 2004.

## 〈著者紹介〉

강 전 일 (Jeonil Kang) 학생회원
2003년 2월: 인하대학교 컴퓨터 공학과 졸업
2006년 2월: 인하대학교 정보통신대학원 석사
2006년 3월 ~ 현재: 인하대학교 정보공학과 박사 과정
〈관심분야〉 RFID 보안, 생체 인식 보안, 무선 센서 네트워크 보안, 무선 인터넷 보안, 웹 인증 보안

양 대 헌 (Dae Hun Nyang) 정회원
1994년 2월: 한국과학기술원 과학기술 대학 전기 및 전자 공학과 졸업
1996년 2월: 연세대학교 컴퓨터 과학과 석사
2000년 8월: 연세대학교 컴퓨터 과학과 박사
2000년 9월 ~ 2003년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2003년 2월 ~ 현재: 인하대학교 정보통신대학원 조교수
〈관심분야〉 암호 이론, 암호 프로토콜, 인증 프로토콜, 무선 인터넷 보안

이 석 준 (Sokjoon Lee) 정회원
1998년 2월: 서울대학교 컴퓨터공학과 졸업
2000년 2월: 서울대학교 컴퓨터공학과 석사
2000년 2월 ~ 현재: 한국전자통신연구원 지식정보보호연구팀 선임연구원
〈관심분야〉 프라이버시 보호기술, 인증 프로토콜, 센서 네트워크 보안, RFID 보안, 무선침입탐지기술



이 경 희 (Kyung Hee Lee) 정회원
1989년: 서울대학교 식품영양학과 학사
1993년: 연세대학교 전산과학과 학사
1998년: 연세대학교 컴퓨터과학과 석사
2004년: 연세대학교 컴퓨터과학과 박사
1993년 1월 ~ 1996년 5월: LG소프트(주) 연구원
2000년 12월 ~ 2005년 2월: 한국전자통신연구원 선임연구원
2005년 3월 ~ 현재: 수원대학교 전임강사
〈관심분야〉 영상처리, 컴퓨터비젼, 인공지능, 패턴인식, 생체인식, 얼굴인식, 다중생체인식