신원 정보 기반 그룹 서명

김 대 엽† · 주 학 수††

요 약

사용자 신원 정보를 공개키 값으로 이용하는 방법은 1985년 Shamir가 처음 개념을 소개한 이후, 공개키 기반 암호 시스템에서 사용자 공개키 인증의 어려움을 해결할 수 있는 효과적인 방안 중 하나로 간주되고 있다. 특히, Dan Boneh가 Weil Paring을 이용한 새로운 기법을 소개한 후, 그 연구가 더욱 활발히 진행되어지고 있으며, 신원 정보 기반의 공개키 암호 시스템을 그룹 서명이나 Conference Key Distribution등 기존의 다양한 응용 기법에 적용하려는 시도가 계속되고 있다. 본 논문에서는 최근 Shundong Xia 등이 [1]에서 제안한 신원 정보를 이용한 그룹 서명 알고리즘의 취약점을 분석하고, 그 원인과 보완책을 제시한다.

키워드 : 신원 정보 기반 암호시스템, 그룹 서명

A Study On ID-Based Group Signature

Kim DaeYoub† · Ju HakSoo††

ABSTRACT

In 2002, Shundong Xia proposed a new ID-based group signature scheme with strong separability. The nature of a strong separability is to divide group manager' s role into a membership manager and a revocation manager. Jianhong et al showed that the scheme was not coalition-resistant. In this letter, we first propose two new attacks, forgery and link attacks, for Xia-You' s scheme. We also propose a new ID-based group signature scheme that is resistant to coalition, forgery and link attacks

Key Words : Id-based Cryptosystem, Group Signature

1. 서 론

공개키 암호 시스템에서 사용자 신원 정보를 공개키로 사용하는 개념은 Shamir가 [2]에서 처음 제안한 후 꾸준히 연구되고 있다. 이와 같은 제안은 PKI 시스템과 같은 일반적인 공개키 시스템에서 사용자 공개키 인증 및 폐지 목록 확인의 어려움을 해결할 수 있는 효과적인 방안 중 하나로 간주되어져 왔다. 특히, Weil Pairing 기반의 알고리즘이 Dan Boneh에 의해서 소개된 후 사용자 신원 정보 기반의 공개키 시스템 연구가 더욱 활발히 진행되고 있다.

그룹 서명은 [4]에서 Chaum에 의하여 처음 소개된 개념이다. 그룹의 구성원이 소속된 그룹을 대표해서 메시지에 서명을 수행하는 기법으로, 그룹 관리자를 제외한 그룹의 다른 구성원이나 외부 사람이 서명 결과를 가지고 실제 서명자의 신원 정보를 획득하기 어렵다는 특징을 갖는다. 만약 서명에 대한 유효성 문제가 제기되면, 그룹 관리자는 서명 결과를 가지고 실제 서명자의 신원 정보를 획득/공개 할 수 있다.

† 종신회원 : 삼성종합기술원, 전문연구원
†† 정 회 원 : 한국정보보호진흥원, 연구원
논문접수 : 2005년 3월 11일, 심사완료 : 2005년 6월 9일

사용자 신원 정보 기반의 그룹 서명 기법은 1997년 S. Park이 [5]에서 제시한 방법으로, 그룹 구성원의 신원정보를 공개키로 사용해서 그룹 서명을 수행한다. 그러나 [5]에서 제시된 방법은 그룹 구성원의 변동에 유연하게 대처하기 어렵다는 단점을 갖고 있다. 즉, 그룹 구성원에 변동이 발생하면, 변동 이전에 수행된 서명 결과를 검증할 수 없다. 뿐만 아니라, 서명 길이는 그룹 구성원의 수에 비례해서 증가한다.이와 같은 단점을 보완하기 위한 방안이 1999년 Yuh-Min Tseng에 의해서 제안 되었다[6]. 제안된 방식은 그룹 구성원의 변동에 관계 없이 그룹 서명을 수행할 수 있도록 설계 되었으나 [9]에서 그 취약점이 분석되었다. 앞서 제안된 방식들은 그룹 관리자가 그룹 구성원의 개인키를 생성/배포할 뿐만 아니라 서명값에서 서명자 신원을 추출하는 기능까지 함께 수행하기 때문에 실제 시스템 구성에 있어서 그룹 관리자에게 역할과 책임이 집중된다는 단점을 갖고 있다. 실제로 이러한 구성은 시스템을 다양한 환경에 적용하는데 어려움이 될 수 있다. Xia 등이 [1]에서 Signature of Knowledge기법을 이용한 새로운 사용자 신원 정보 기반의 그룹 서명 기법을 제안했다. 특히 제안된 기법은 그룹 관리자의 기능을 그룹 구성원의 개인키를 생성/배포하는 Membership Manager와 서명에

대한 논란이 발생되었을 때 서명자의 신원을 공개하는 Revocation Manager의 기능을 분리해서 운영할 수 있도록 설계되었다.

본 논문에서는 Xia 등이 [1]에서 제안한 신원정보 기반 그룹 서명의 취약점을 분석하고, 그 원인과 이를 효율적으로 보완할 수 있는 방안을 제안한다.

본 논문은 다음과 같이 구성되었다. 1절에서는 [1]에서 제안한 방식을 간략하게 설명한다. 2절에서는 제안된 방식에 대하여 기존에 연구된 취약점 및 본 논문에서 새롭게 제시된 취약점을 소개한다. 3절에서는 취약점 발생 원인을 분석하고, 이를 효과적으로 대처할 수 있는 방안을 제안하고, 4절에서는 3절에서 제안된 대처 방안을 바탕으로 새로운 신원 정보 기반 그룹 서명을 제안한다.

2. 본 론

2.1 Xia-You의 사용자 신원 정보 기반 그룹 서명

이 절에서는 [1]에서 제안한 사용자 신원 정보 기반 그룹 서명을 소개한다. 제안된 기법의 구성요소는 신뢰기관, 그룹 관리자, 그룹 구성원, 그리고 서명 검증자로 구성된다. 특히, 그룹 관리자는 Membership Manager와 Revocation Manager로 역할을 분리시킬 수 있다.

2.1.1 신뢰기관 시스템 설정 및 사용자 등록

[12]와 [13]에서 제안된 것처럼 신뢰기관은 100자리 정도의 소수 p_1과 p_2를 선택한다. 이 때, 선택된 소수 p_1과 p_2는 다음과 같은 조건을 만족한다:

- p_1-1과 p_2-1는 13자리에서 15자리의 인수로만 소인수 분해된다.
- $\frac{p_1-1}{2}$와 $\frac{p_2-1}{2}$는 서로 소이다.
- $p_1 \equiv \pm 1 \bmod 8$을 만족하고 $p_2 \equiv \pm 3 \bmod 8$을 만족한다.

신뢰기관은 $m=p_1 \times p_2$을 계산한다. 이와 같은 조건을 만족하도록 선택된 p_1과 p_2를 사용할 경우, Jacobi symbol $(2/m)$은 항상 -1이 된다. 이 경우, 신뢰기관은 Pohlig and Hellman 기법을 사용해서 modulo p_1과 p_2에 대한 discrete logarithm을 쉽게 찾을 수 있다. 신뢰기관은 $g < \min(p_1, p_2)$를 만족하는 난수 g를 선택한다. 신뢰기관은 m과 g를 시스템 공개 정보로 제공하고, p_1과 p_2를 비밀정보로 안전하게 관리한다.

그룹 서명을 수행할 사용자 U_i의 신원정보를 ID_i라 하자. 사용자 등록을 위하여 신뢰기관은 사용자 U_i에게

$$ID_i \equiv g^{x_i} \bmod m \tag{1}$$

을 만족하는 비밀키 x_i를 생성해서 안전하게 전송한다. 단, 이 때 사용되는 ID_i의 값은 Jacobi symbol (ID_i/m)이 항상 1을 만족한다고 하자. ID_i와 x_i 계산 방법은 [1], [12], [13]에서 제시하고 있으므로, 이를 참조할 수 있다.

2.1.2 그룹 관리자 시스템 설정 및 구성원 등록

그룹 관리자는 안전한 소수 p_3과 p_4를 생성한 후, $n=p_3 \times p_4$를 계산한다. 이 때, $m < n$을 만족해야 된다. $\gcd(e, \phi(n))=1$을 만족하는 정수 e를 선택한 후, $d \times e \equiv 1 \bmod \phi(n)$을 만족하는 d를 계산한다.

그룹 관리자는 정수 $x \in Z_m = \{0, \cdots, m-1\}$와 $h \in Z_m^* = \{a \in Z_m \mid \gcd(m, a)=1\}$를 선택한 후, $y=h^x \bmod m$을 계산한다.

그룹 관리자의 공개키 정보는 (n, e, h, y, H)이고, 비밀키 정보는 (x, d, p_3, p_4)이다. 이 때, $H: \{0,1\}^* \rightarrow Z_m$는 안전한 해쉬 함수이다. 그룹에 가입하기 원하는 사용자 U_i를 위하여 그룹 관리자는

$$z_i = ID_i^d \bmod n \tag{2}$$

를 계산한 후, z_i를 U_i에게 안전하게 전달한다. 가입자의 공개키 정보는 ID_i이고, 비밀키는 (x_i, z_i)이다. 이 후부터 (x_i, z_i, ID_i)를 사용자 인증 정보로 부르기로 한다.

2.1.3 서명

문서 M에 그룹 서명을 수행하기 위하여 그룹 구성원 U_i는 난수 $\alpha, \beta, \theta, \omega \in Z_m$과 $\delta \in Z_n$을 선택한 후, 다음을 계산한다:

$$\begin{aligned}
&A=(y^{\alpha} \cdot z_i) \bmod n,\\
&B=y^{\omega} \cdot ID_i,\\
&C=h^{\omega} \bmod m,\\
&D=H(y \| g \| h \| A \| B \| \widehat{B} \| C \| \nu \| t_1 \| t_2 \| t_3 \| M),\\
&E=\delta - D \cdot \varepsilon,\ \varepsilon = \alpha e - w,\\
&F=\beta - D \cdot \omega,\\
&G=\theta - D \cdot x_i
\end{aligned} \tag{3}$$

단, $\widehat{B}=B \bmod m$, $\nu=(A^e/B) \bmod n$, $t_1=y^{\delta} \bmod n$, $t_2=(y^{\beta} \cdot g^{\theta}) \bmod m$, 그리고 $t_3=h^{\beta} \bmod m$을 만족한다. 서명자 U_i는 (A, B, C, D, E, F, G)를 검증자에게 전송한다.

2.1.4 서명 검증

검증자는 신뢰기관과 그룹 관리자의 공개키 정보, 그리고 서명자로부터 전송된 서명 값으로부터 다음을 계산한다:

$$\begin{aligned}
&\widehat{B}'=B \bmod m,\\
&\nu'=(A^e/B) \bmod n,\\
&t_1'=(\nu'^{D} \cdot y^{E}) \bmod n,\\
&t_2'=(\widehat{B}'^{D} \cdot y^{F} g^{G}) \bmod m,\\
&t_3'=(C^{D} \cdot h^{F}) \bmod m,\\
&D'=H(y \| g \| h \| A \| B \| \widehat{B}' \| C \| \nu' \| t'_1 \| t'_2 \| t'_3 \| M)
\end{aligned} \tag{4}$$

검증자는 계산된 D'와 서명 값의 D를 비교해서 $D'=D$ 이면 해당 서명을 유효한 것으로 판단한다.

2.1.5 서명자 신원 증명

서명 결과에 대하여 문제가 제기 되었을 때, 그룹 관리자는

$$ID_i=(B/C^x) \bmod m \tag{5}$$

을 계산해서 서명자 U_i의 신원 정보를 확보/공개한다.

2.2 취약점 분석

2.1 그룹 서명의 안전성

그룹 서명의 안전성은 다음과 같은 판단 기준으로 평가된다:

- Correctness : 그룹 구성원에 의해서 생성된 서명 결과는 검증 단계에서 항상 유효한 것으로 검증 된다.
- Unforgeability : 오직 그룹의 구성원만이 해당 그룹을 대표해서 그룹 서명을 수행할 수 있다.
- Anonymity : 그룹 관리자를 제외하고, 서명 결과를 가지고 실제 서명한 구성원의 신원 정보를 획득하는 것은 어렵다(Computationally hard).
- Unlinkability : 서로 다른 두 서명이 동일한 구성원에 의해서 생성되었는지 여부를 판단하는 것은 어렵다(Computationally hard).
- Traceability : 그룹 관리자는 서명 결과로부터 서명자 신원정보를 획득할 수 있다.

2.2.2 취약점 분석

Zang Jianhong은 [7]에서 Xia 등이 제안한 신원 정보 기반 그룹 서명에 대한 두 가지 공격 방법을 제안했다. 첫 번째 공격 방법은 유효 서명자 U_i와 U_j가 공모하여 새로운 인증 정보 (x_k, z_k, ID_k)를 위조하는 공모 공격 방법이고, 두 번째 공격 방법은 서명자 U_i가 자신의 인증정보를 이용하여 인증정보 (x_k, z_k, ID_k)를 위조하는 방법이다. 전자의 경우 두 공모자는 자신의 인증정보를 공유해서 다음과 같은 새로운 인증정보를 생성한다:

$$\begin{aligned} ID_k &= ID_i \cdot ID_j \ , \\ x_k &= x_i + x_j \ , \\ z_k &= z_i \cdot z_j \end{aligned} \tag{6}$$

후자의 경우 서명자 U_i은 난수 r을 선택한 후, 자신의 인증정보를 사용하여 다음과 같이 다른 인증정보를 생성한다:

$$\begin{aligned} ID_k &= (ID_i)^r \ , \\ x_k &= r x_i \ , \\ z_k &= z_i^r \end{aligned} \tag{7}$$

후자의 공격 방법은 r명의 공모 공격을 고려한다면 전자의 방법에서 쉽게 예측할 수 있다. 이처럼 위조된 인증정보 (x_k, z_k, ID_k)은 $ID_k \equiv g^{x_k} \bmod m$과 $z_k \equiv ID_k^d \bmod n$을 동시에 만족한다. 그러므로 검증자가 (x_k, z_k, ID_k)를 이용한 서명을 검증하면, 해당 그룹의 정당한 구성원이 수행한 유효한 서명으로 잘못 판단하게 된다.

이 절에서는 [7]에서 제안된 공격 방법 이외에 두 종류의 공격 방법을 제안한다. 첫 번째 공격 방법은 인증정보를 위조하는 방법이다. 공격을 위하여 우리는 [9]에서 사용한 공격 방법을 다시 사용한다. 공격자는 적당히 작은 난수 r를 선택한 후,

$$(x_k, z_k, ID_k) = (re, g^r, g^{r \cdot e}) \tag{8}$$

를 계산해서 인증정보로 사용한다. 위조된 인증정보 (x_k, z_k, ID_k)는 $ID_k \equiv g^{x_k} \bmod m$과 $z_k \equiv ID_k^d \bmod n$을 만족하므로 (x_k, z_k, ID_k)은 해당 그룹의 구성원의 유효한 인증정보처럼 사용될 수 있다. [7]에서 제안한 방법은 유효한 구성원의 인증정보를 이용하지만, 본 논문의 경우는 그룹 구성원의 도움 없이 유효한 인증정보를 생성할 수 있다는 차이가 있다.

또 다른 공격은 그룹 서명의 불연계성(Unlinkability)에 대한 것이다. 공격을 위해 동일한 그룹에 포함된, 임의의 그룹 구성원 U_i와 U_j에 대하여, 다음과 같은 가정을 설정한다:

(가정 1) $U_i \neq U_j$ 이면 $ID_i = k \cdot ID_j$를 만족하는 정수 k는 존재하지 않는다.

(가정 2) ID_i와 ID_j는 y가 아닌 서로 다른 인수를 갖는다. 즉, $K = \gcd(ID_i, ID_j)$라 하고, ID_i와 ID_j를 각각 $a_1 \cdot a_2 \cdots a_{n_i} \cdot K$와 $b_1 \cdot b_2 \cdots b_{n_j} \cdot K$ 라 하면, 적당한 r과 s가 존재해서 $a_r \neq y$이고 $b_s \neq y$를 만족한다.

공격자는 서로 다른 두 서명값 (A, B, C, D, E, F, G)와 $(A', B', C', D', E', F', G')$에서 B와 B'를 선택한다. $B = y^{\omega} \cdot ID_i$이고 $B' = y^{\omega'} \cdot ID_j$라 하면, 공격자는 다음을 계산한다:

$$\begin{aligned} k_1 &= \frac{B}{B'} = y^{\omega - \omega'} \frac{ID_i}{ID_j} = y^{\omega - \omega'} \frac{a_1 a_2 \cdots a_{n_i}}{b_1 b_2 \cdots b_{n_j}} \ , \\ k_2 &= \frac{B'}{B} = y^{\omega' - \omega} \frac{ID_i}{ID_i} = y^{\omega' - \omega} \frac{b_1 b_2 \cdots b_{n_j}}{a_1 a_2 \cdots a_{n_i}} \end{aligned} \tag{9}$$

(가정 1)과 (가정 2)에 의해서 $ID_i \neq ID_j$이면, k_1과 k_2는 정수가 아닌 유리수가 됨을 쉽게 알 수 있다. 만약 식 (9)를 계산한 결과, k_1이 정수라 가정하자. (가정 2)에 의하여 $\omega \geq \omega'$이고 $a_1 a_2 \cdots a_{n_i} = b_1 b_2 \cdots b_{n_j}$ 임을 알 수 있다. 그러므로 $ID_i = ID_j$가 된다. 또한 k_1이 정수이면, $\omega \leq \omega'$이므로 k_2는 정수가 아닌 유리수임을 알 수 있다. 그러므로 공격자는 k_1과 k_2를 계산해서 두 서명값의 연계성 여부를 판단할 수 있다.

3. 대응책

인증정보 (x_k, z_k, ID_k)를 위조하는 공격 방법과 관련된 취약점의 원인은 다음 두 가지로 요약 할 수 있다.

(a) 신뢰기관과 그룹 관리자가 그룹 구성원의 비밀키를 생성하기 위해 사용하는 함수를 각각 $Z(ID_i)=z_i$와 $X(ID_i)=x_i$라 하면, 두 함수는 선형성을 갖는다. 즉,

$$\begin{aligned} Z(ID_i \cdot ID_j) &= Z(ID_i) \cdot Z(ID_j) \ , \\ X(ID_i \cdot ID_j) &= X(ID_i) + X(ID_j) \end{aligned} \tag{10}$$

을 만족한다.

(b) 신뢰기관과 그룹 관리자가 생성한 키 x_i와 z_i 사이에는 $Z^{-1}(z_i)=ID_i=X^{-1}(x_i)$와 같은 관계가 성립된다. 그러므로, x_i 또는 z_i 중 하나가 주어졌을 때, 대응되는 나머지 하나의 값을 쉽게 계산할 수 있다.

(a)와 같은 취약점을 보완하기 위하여 사용자 ID_i에서 비밀키를 생성하는 함수가 선형성을 갖지 않도록 선택하는 방법을 고려할 수 있다. 또한, (b)와 같은 취약점에 대응하기 위하여 사용자 키 생성 기관이 키를 생성할 때, x_i와 z_i의 연관관계가 쉽게 노출되지 않도록 비밀키 생성 함수를 선택해야 된다.

서명값 연계성과 같은 취약점을 막기 위해서는 사용자가 그룹을 대표해서 서명 할 때 마다 서로 다른 난수를 생성한 후, 서명 기법에 사용하는 방법을 사용할 수 있다.

4. 신원 정보 기반 그룹 서명 제안

이 절에서는 앞에서 살펴본 취약점을 보완하기 위하여 제시했던 기법을 적용한 새로운 신원 정보 기반 그룹 서명을 제안한다. 제안하는 기법은 그룹 관리자, 그룹 구성원, 서명 검증자를 구성 요소로 갖는다.

제안하는 방식에서는 시스템 운영을 위하여 다음과 같은 두가지 가정을 설정한다:

(가정 3) 시스템에서 사용되는 해수 함수 H는 선형성을 갖지 않는다고 가정하자.

(가정 4) 그룹 관리자와 그룹 구성원 사이에 사용되는 통신 채널은 안전하다고 가정한다.

4.1 키 생성

4.1.1 그룹 관리자 키 생성

그룹 관리자는 1.1절에서 신뢰기관이 소수 p_1과 p_2를 선택하는 방법처럼 안전한 소수 p와 q를 선택하고 $m=p\times q$를 계산한다. 또한, $g<\min(p,q)$를 만족하는 난수 g를 선택한다.

정수 $h\in Z_m^*$와 $x\in\{a\in Z_m \mid \gcd(a,\phi(m))=1\}$를 선택한 후, $y=h^x \bmod m$을 계산한다.

선택한 x에 대하여 $x\cdot u\equiv 1 \bmod \phi(m)$을 만족하는 u를 계산한다.

$\gcd(e,\phi(m))=1$을 만족하는 정수 e를 선택한 후, $d\times e\equiv 1 \bmod \phi(m)$을 만족하는 d를 계산한다.

그룹 관리자는 m, g, y, h, u, e를 시스템 공개 정보로 제공하고, p, q, x, d를 비밀 정보로 안전하게 관리한다.

4.1.2 그룹 구성원 키 생성

그룹 구성원으로 등록을 원하는 사용자 U_i는 해당 그룹에 자신의 ID_i를 등록하고, $H(ID_i)$를 다음과 같이 새롭게 설정 한다:

$$H(ID_i)=\begin{cases} H(ID_i) & , \text{if } J=1, \\ 2H(ID_i) \bmod m & , \text{if } J=-1 \end{cases} \tag{11}$$

〈표 1〉 구성원 키 정보

	GM	U_i
공개	$m,\ g,\ h,\ y,\ u,\ e$	ID_i
비밀	$p,\ q,\ x,\ d$	$x_i,\ z_i$

단 $J=(H(ID_i)/m)$은 Jacobi Symbol을 의미하며, 식(11)에 의해 새롭게 생성된 해쉬값의 Jacobi Symbol $(H(ID_i)/m)$은 항상 1이 된다.

그룹 구성원은 ID_i와 식(11)에 의해서 생성된 $H(ID_i)$를 가지고 그룹 관리자로부터 다음을 만족하는 개인키 (x_i, z_i)를 할당 받는다:

$$\begin{aligned} H(ID_i) &\equiv g^{x_i} \bmod m \\ ID_i &\equiv z_i^e \bmod m \end{aligned} \tag{12}$$

x_i의 계산은 [1]에서 이미 언급하였던 것처럼 m의 소인수 분해 결과와 [12], [13]에서 제시한 방법을 이용해서 계산할 수 있다. 사용자 U_i가 해당 그룹을 대표해서 서명할 때 사용되는 인증정보는 (x_i, z_i, ID_i)가 된다.

4.2 서명 수행

문서 M에 그룹 서명을 수행하기 위하여 그룹 구성원 U_i는 난수 $\beta, \omega, \varepsilon, \tau, \pi \in Z_m$을 선택한 후, 다음을 계산한다:

$$\begin{aligned} &A=\varepsilon\cdot\tau\cdot z_i \bmod m \ , \\ &B=\varepsilon^{e\cdot u}\cdot h^{\omega} \bmod m \ , \\ &C=\varepsilon^{e}\cdot y^{\omega}\cdot ID_i \ , \\ &D=\tau\cdot g^{\omega}\cdot H(ID_i) \ , \\ &E=H(y\,\|\,g\,\|\,h\,\|\,A\,\|\,B\,\|\,C\,\|\,D\,\|\,t\,\|\,M) \ , \\ &F=\beta-E\cdot\omega \ , \\ &G=\pi-E\cdot e\cdot(\omega+x_i) \end{aligned} \tag{13}$$

단, E를 계산하기 위해 필요한 t값은 다음과 같이 계산한다.

$$t=(y^{\beta}\cdot g^{\pi}) \mod m \tag{14}$$

을 만족한다. 식 (13)을 계산한 후, 서명자 U_i는 $((\omega,\tau),A,B,C,D,E,F,G)$를 검증자에게 전송한다.

4.3 서명 검증 및 신원 정보 획득

검증자는 전송된 서명 (A,B,C,D,E,F,G)를 검증하기 위해 $\tau\cdot g^{\omega}$를 계산 한 후, $\tau\cdot g^{\omega}\neq D$ 이고, $\tau\cdot g^{\omega}\mid D$ 인지를 확인한다.

$$T=((\frac{D}{A})^{e}\cdot C)^{E}\cdot y^{F}\cdot g^{G} \bmod m \tag{15}$$

과

$$E'=H(y\|g\|h\|A\|B\|C\|D\|T\|M) \tag{16}$$

를 계산한다. 검증자는 $E=E'$이면 서명이 유효하다고 판단한다. 실제로 E와 E'가 동일하기 위해서는 계산에 사용된 t와 T가 동일해야 된다. T를 계산하기 위해 사용된 $((\frac{D}{A})^{e}\cdot C)^{E}$이

$$\begin{aligned}&((\frac{D}{A})^{e}\cdot C)^{E}\\&=(\frac{(\tau\cdot g^{\omega}\cdot H(ID_i))^{e}\cdot C}{(\varepsilon\cdot\tau)^{e}\cdot z_i^{e}})^{E} \bmod m\\&=(\frac{(g^{\omega}\cdot H(ID_i))^{e}\cdot(\varepsilon^{e}\cdot y^{\omega}\cdot ID_i)}{\varepsilon^{e}\cdot ID_i})^{E}\bmod m\\&=g^{e\cdot\omega\cdot E}\cdot H(ID_i)^{e\cdot E}\cdot y^{\omega\cdot E} \bmod m\end{aligned} \tag{17}$$

이므로,

$$\begin{aligned}T&=((\frac{D}{A})^{e}\cdot C)^{E}\cdot y^{F}\cdot g^{G}\bmod m\\&=g^{e\cdot\omega\cdot E}\cdot H(ID_i)^{e\cdot E}\cdot y^{\omega\cdot E}\cdot y^{F}\cdot g^{G} \bmod m\\&=g^{e\cdot\omega\cdot E}\cdot H(ID_i)^{e\cdot E}\cdot y^{\omega\cdot E}\cdot y^{\beta\ \omega\cdot E}\cdot g^{\pi-e\cdot(\omega+x)\cdot E} \bmod m\\&=g^{e\cdot\omega\cdot E}\cdot H(ID_i)^{e\cdot E}\cdot y^{\beta}\cdot g^{\pi}\cdot g^{-e\cdot\omega\cdot E}\cdot H(ID_i)^{-e\cdot E} \bmod m\\&=y^{\beta}\cdot g^{\pi}\\&=t\end{aligned} \tag{18}$$

가된다. 그러므로 서명이 유효하다면 E와 E'는 동일한 값을 갖는다.

서명 결과에 대한 문제가 제기 되면, 그룹 관리자는 비밀키 x를 사용해서 서명자 신원을 다음과 같이 획득할 수 있다:

$$\begin{aligned}\frac{C}{B^{x}}&=\frac{\varepsilon^{e}\cdot y^{\omega}\cdot ID_i}{(\varepsilon^{e\cdot u}\cdot h^{\omega})^{x}}\\&=\frac{\varepsilon^{e}\cdot y^{\omega}\cdot ID_i}{\varepsilon^{e}\cdot(h^{x})^{\omega}}\\&=\frac{y^{\omega}\cdot ID_i}{y^{\omega}}=ID_i\end{aligned} \tag{19}$$

5. 안전성 분석 및 연산량 비교

식 (12)에 의해서 생성된 인증정보 (z_i,ID_i)는 선형성을 갖는다. 즉, $ID_k=ID_i\cdot ID_j$로 놓고 $z_k=z_i\cdot z_j$로 계산하면 유효한 (z_k,ID_k)를 생성할 수 있다. 그러나 해쉬 함수 H가 선형성을 갖고 있지 않기 때문에

$$H(ID_k)=H(ID_i\cdot ID_j) \tag{20}$$

와

$$H(ID_i)\cdot H(ID_j)\equiv g^{x_i}\cdot g^{x_j}=g^{x_i+x_j} \tag{21}$$

은 서로 다르다. 즉, $x_k=x_i+x_j$를 사용할 수 없다. 그러므로 공격자는 주어진 $H(ID_k)$에 대하여 $H(ID_k)=g^{x_k}$를 만족하는 x_k를 새롭게 생성해야 된다. 이는 이산대수 문제의 어려움과 동치이므로, 실제 서명에 필요한 인증정보 (x_k,z_k,ID_k)를 생성할 수 없다.

또한 $(x_i,H(ID_i))$를 생성하는 방법도 선형성을 갖는다. 즉, 새로운 개인키 x_k를 x_i+x_j와 같다고 설정하면, 대응되는 해쉬값은 $H(ID_i)\cdot H(ID_j)$이 된다. 그러나 키 생성에 사용하는 해쉬 함수 H가 충분히 안전하다고 가정하면 주어진 해쉬값 $H(ID_i)\cdot H(ID_j)$에 대하여 $H(ID_k)=H(ID_i)\cdot H(ID_j)$를 만족하는 ID_k를 찾기 어렵다. 그러므로 그룹 구성원들의 공모를 통하여 유효한 인증 정보 (x_k,z_k,ID_k)를 위조하는 것은 어렵다. 제안된 키 생성 방식은 [7]에서 제안된 공격 방법이나 본 논문에서 제안된 공격 방법에 모두 안전하다.

식 (13)에 의해서 생성된 서명 결과 중에서 사용자 정보가 포함된 서명 정보 $\{A,B,C,D\}$에는 서명 시 마다 다른 난수값이 포함되기 때문에 서명 결과를 가지고 서명자 동일성 여부를 정확히 판단할 수 없다.

연산 효율성 분석을 위하여 Shundong Xia 등이 제안한 방법과 본 논문에서 제안하는 방법에서 서명 생성 및 검증 단계의 주요 연산 횟수를 비교했다. 각각의 알고리즘에서 서명 생성 및 검증에 필요한 연산 횟수는 <표 2>와 같다. 전체 연산 시간에 가장 큰 영향을 미치는 것은 지수승 연산으로 서명 생성에 필요한 지수 연산 횟수는 두 방법이 동일하나 서명 검증에 필요한 지수 연산은 본 논문에서 제안한 방식이 3회 더 적음을 알 수 있다. 특히 지수승 연산 횟수에서 괄호로 표시된 값은 지수의 길이가 $\log_2 m$ (또는 $\log_2 n$)인 지수승 연산의 횟수는 나타낸다. 이는 지수의 값으로 2 또는 e를 취하는 연산에 비해 많은 연산량을 필요로 한다. 그러므로 지수의 길이가 $\log_2 m$ (또는 $\log_2 n$)인 지수승 연산의 횟수가 적은 것이 더 효과적이라 할 수 있다.

Shundong Xia가 제안한 방법의 또 다른 특징은 그룹 관리 기능을 membership manager와 revocation mamager로 분리/적

용할 수 있다는 것이다. 본 논문에서 제안하는 방법 또한 이와 같은 특징을 유지하고 있다. <표 3>은 그룹 관리 기능을 분리하기 위해 각각의 관리자에게 할당되는 공개키 및 비밀키를 나타낸다.

〈표 2〉 연산 횟수 비교

		Xia's Scheme	New Scheme
서명 생성	곱셈	6	11
	나눗셈	1	0
	지수승	8 (7)	8 (6)
서명 검증	곱셈	3	4
	나눗셈	1	1
	지수승	8 (7)	5 (4)

〈표 3〉 그룹 구성원 기능 분리에 따른 키 정보

	Membership Manager	Revocation Manager
공개	e	u
비밀	d	x

3. 결 론

Xia 등이 제시한 서명기법은 그룹 구성원의 가입 및 탈퇴에 영향을 받지 않도록 설계되었고, 특히 그룹 관리자의 구성원 관리 및 서명 신원 증명 기능을 분리할 수 있다는 특징이 있다. 그러나 앞에 설명한 것처럼 제시한 기법은 서명 수행에 필요한 인증 정보의 위조가 가능하며, 서명 결과에 연계성(linkability)이 있다는 취약점이 있다. 인증 정보의 위조는 사용자 ID로부터 개인키를 생성하는 인증정보 생성에 사용된 함수들이 선형성 및 연관성을 갖기 때문에 발생한다. 특히 서명 결과의 연계성(Linkability)은 서명 생성 함수가 사용하는 입력 인자에 난수성이 없기 때문에 발생한다.

본 논문에서는 Xia 등이 제시한 기법의 취약성 및 그 원인 분석결과를 바탕으로 인증정보 및 서명값 위조가 어렵고, 서명결과 사이에 연계성 여부를 결정하기 힘들도록 개선한 신원 정보 기반 그룹 서명 알고리즘을 제안한다. 제안된 알고리즘은 서명 생성 및 검증에 필요한 지수 연산량이 Xia 등이 제안한 알고리즘의 연산량과 유사하며, Xia 등이 제안한 서명 알고리즘처럼 그룹 관리자 기능을 분리 해서 적용할 수 있다는 특징을 갖는다.

참 고 문 헌

[1] Shundong Xia, Jinyan You, "A group signature scheme with strong separability", *The Journal of Systems and Softwaware*, No. 60, pp.177-182, 2002.

[2] Adi Shamir, "Identity-based Cryptosystems and Signature Scheme", *LNCS 0196, Crypto'84*, pp.47-53, 1984.

[3] Dan Boneh, Matt Franklin "Identity-Based Encryptin form the Weil Pairing", *LNCS 2139, Crypto'01*, pp.213-229, 2001.

[4] D. Chaum, E. Van Heyst, "Group Signatures", *LNCS 1070, Eurocrypt'96*, pp.257-265, 1991

[5] S. Park, S. Kim, and D. Won, "ID-based group signature", *Electronics Letters, 33(19)*, pp.1616-1617, 1997.

[6] Yuh-Min Tseng, Jinn-Ke Jan, "A Novel ID-based group signature", *Information Sciences*, No.120, pp.131-141, 1999.

[7] Zhang Jianhong, Wang Ji-lin, Wang Yumin, "Two Attacks on Xia-You Group Signature", ePrint (http://eprint.iacr.org/2002/177/), 2002

[8] Claude Castelluccia, "How to convert any ID-based Signature Schemes into a Group Signature Scheme", ePrint

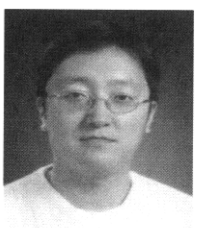
[9] Mare Joye, Seungjoo Kim, Narn-Yih Lee, "Cryptanalysis of two group signature schemes", *Proceedings of Information Security Workshop'99, LNCS 1729*, pp.271-275, 1999.

[10] Giuseppe Ateniese, Marc Joye, Gene Tsudik, "On the Difficulty of Coalition-Resistance in Group Signature Schemes", In G. Persiano, Ed., 2nd Workshop on Security in Communication Networks (SCN '99), Almafi, Italy, September 1999.

[11] Giuseppe Ateniese, Jan Camenisch, Marc Joye, Gene Tsudik, "A Practical and Provable Secure Coalition-Resistant Group Signature Scheme", *LNCS 1880, Crypto'00*, pp.255-270, 2000.

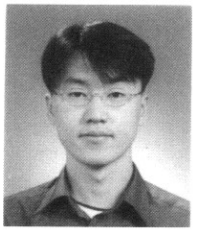
[12] Maurer, U.M., Yacobi, Y., "Non-interactive public-key cryptography", EUROCRYPT'91, LNCS 547, pp.498-507, 1992.

[13] Lim, C.H., Lee, P.J., "Modified Maurer-Yacobi's scheme and its application." AUSCRYPT'92, LNCS 718, pp.203-323, 1992.



김 대 엽

e-mail : daeyoub.kim@samsung.com
1994년 고려대학교 수학과(학사)
1996년 고려대학교 수학과(석사)
2000년 고려대학교 수학과(박사)
1997년~2001년 (주)텔리맨, 위성통신연구소, CAS팀, 책임연구원
2001년~2002년 (주)시큐아이닷컴, 정보보호연구소, 책임연구원
2003년~현재 삼성종합기술원, Security TG, 전문연구원
관심분야 : 암호 응용 프로토콜, CAS, DRM, Smart Card



주 학 수

e-mail : hsju@kisa.or.kr
1997년 고려대학교 수학과(학사)
1999년 고려대학교 수학과(석사)
2005년 고려대학교 수학과(박사)
2001년~현재 한국정보보호진흥원 암호응용팀
관심분야 : 암호 응용 프로토콜, DRM, Watermarking