고차잉여류 문제에 기반을 둔 다중서명 방식

이 보 영† · 박 택 진†† · 원 동 호†††

요 약

Itakura와 Nakamura는 RSA 서명방식을 이용하여 최초의 다중서명 방식을 제안하였다. 그러나 이 방식은 많은 사용자들이 하나의 서류에 서명하고자 할 경우, 각 사용자들이 사용하는 RSA 법(modulus) n 크기의 차이로 blocking이 발생하는 문제점이 존재한다. 1991년 Ohta와 Okamoto가 제안한 Fiat-Shamir 서명방식을 이용한 다중서명 방식은 서명자가 첫 번째 라운드에서 생성한 난수(random number)를 두 번째 라운드 즉, 메시지를 직접 서명할 때까지 보관해야 하는 문제가 있다. 또한 L. Harn은 ElGamal의 변형된 서명방식을 이용한 다중서명 방식을 제안하였다. 국내에서는 김성덕외 2인이 Park-Won 방식의 변형된 서명방식을 이용하여 추가적인 라운드를 요구하지 않고, 서명문의 길이가 일정한 효율적인 순차 다중서명 방식을 제안하였다.

본 논문에서는 기 제안된 Park-Won 방식의 변형된 서명방식을 이용한 다중서명 방식이 안전성에 문제가 있음을 보이고, 고차 잉여류를 이용하여 안전성을 개선시킬 수 있는 방법을 제안하고자 한다.

An Efficient ID-Based Multisignature Scheme Based on the High Residuosity Problem

Bo-Young Lee† · Taek-Jin Park†† · Dong-Ho Won†††

ABSTRACT

Itakura and Nakamura proposed the first mutisignature scheme based on the RSA signature scheme. But if many users sign on one paper, then their scheme has a reblocking problem. In 1991, Ohta and Okamoto proposed a multisignature scheme by using Fiat-Shamir signature scheme. But in this scheme, the group of signers must generate common random number in the first round, and in the second round, they sign the message with common random number. Also L.Harn proposed a multsignature scheme which is based on the ElGamal's. In Korea, S.D.Kim et al. at ICEIC'95 conference, proposed an efficient sequential multisignature scheme by using the modified Park-Won scheme. This scheme is not require an additional round to generate common random number, and has fixed signature length. In this paper, we analyze problem of Kim's multisignature scheme, and propose a new multisignature scheme based on r^{th} residuosity problem.

1. 서 론

1976년 Diffie와 Hellman이 공개키 암호방식의 개념을 처음으로 소개한 후, 1978년 Rivest, Shamir, Adleman은 인수 분해 문제의 어려움에 근거한 RSA 디지털 서명방식을 제안하였으며, 그 후로 많은 디지털 서명방식들이 개발되어 왔다.[1] 그러나 한 문서에 여러 사람이 서명하는 경우, 보통의 디지털 서명방식을 직접 반복해서 적용하기에는 서명의 길이가 증가하기 때문에 비효율적이다. 이러한 문제점을 해결하기

† 준 회 원 : 성균관대학교 대학원 전기전자 및 컴퓨터공학부
†† 정 회 원 : 영동대학 전자과 교수
††† 종신회원 : 성균관대학교 공과대학 전기전자 및 컴퓨터공학부 교수
논문접수 : 1998년 12월 8일, 심사완료 : 1999년 1월 26일

위해 나온 개념이 다중서명(multisignatures)이다.

1983년 Itakura와 Nakamura는 RSA 서명방식을 확대 적용한 다중서명 방식을 제안하였다.[2] 이 방식은, blocking 문제를 해결하기 위하여, 서명자의 직위에 따라 다음 서명자의 RSA modulus가 이전 서명자의 RSA modulus보다 크도록 서명 순서가 고정되어 있고, 또한 서명자의 직위가 변동될 때마다 자신의 비밀키를 변경하여야 한다는 단점을 가지고 있다. 이외에도 Okamoto, Harn, Kiesler 등이 RSA 서명방식을 이용한 다중서명 방식을 제안하였다.[3,4,5]

Ohta와 Okamoto는 Fiat-Shamir 방식에 근거한 다중서명 방식을 제안하였다. 이 방식은 서명 처리속도가 빠르고 ID에 근거하기 때문에 RSA에 근거한 서명방식 보다 효율적이다. 그러나 서명 생성에 앞서서 난수를 생성하기 위한 준비 단계가 필요하다는 단점이 있다.[6]

이밖에 L. Harn은 ElGamal의 변형된 서명방식을 이용한 다중서명 방식을 제안하였다. 그러나 이 방식 역시 추가적인 통신 횟수가 요구된다는 단점이 있다.[7]

국내에서의 다중서명 방식에 대한 연구는 강창구 등에 의해 연구된 다중서명 방식과 김성덕 등에 의하여 연구된 다중서명 방식이 있다. 또한 강창구, 김대영 등은 Fiat-Shamir 방식에 근거하여 Ohta-Okamoto 방식에서의 통신 복잡성의 문제를 극복할 수 있는 다중서명 방식을 제안하였다.[8] 이 방식은 Ohta-Okamoto 방식보다 통신 복잡도 측면에서 효율적이지만 서명문의 길이가 서명자의 수에 따라 증가한다는 단점이 있다.

박성준, 원동호(Park-Won)등은 자체인증 개인식별정보 방식을 자체인증 공개키 방식으로 변환하여 자체인증 공개키 방식에 기반을 둔 서명방식을 제안하였다.[16] 이 방식은 certification-based 방식이 아닌 id-based방식이면서도 사용자가 자신의 비밀키를 선택할 수 있는 id-based 방식이다. 또한 김성덕 등은 Park-Won(PW) 방식의 변형된 서명방식을 이용하여 서명 순서가 제약받지 않고, 추가적인 통신이 필요치 않으며, 서명문의 길이가 일정한 효율적인 순차 다중서명 방식을 제안하였다.[9,10,11]

본 고에서는 기 제안된 다중서명 방식[9,10,11]이 서명방식으로 안전하지 못함을 보이고, 고차 잉여류를 이용한 다중서명 방식의 안전성을 개선시킬 수 있는 방법을 제안하고자 한다.

2. 다중서명 방식

2.1 다중서명의 개요

지금까지 개발되어 온 대부분의 전자서명은 문서에 한 사람이 서명하는 단순서명(Single Signature) 방식이었다. 그러나 이런 단순서명 방식을 실제 생활에 적용하기에는 여러 가지 문제점이 있다. 이런 문제중에 하나가 결재와 서명, 계약의 경우와 같이 여러 사람이 한 문서에 서명하는 경우이다. 단순서명을 반복해서 적용하면 서명의 길이가 늘어나고 서명을 검증하려면 서명자의 수만큼 검증과정을 거쳐야 하기 때문에 서명자가 많은 경우 시간이 오래 걸린다는 단점이 있다. 이러한 단순서명 방식의 문제점을 해결하기 위해 나온 개념이 다중서명(Multisignature) 방식이다.

다중서명 방식은 같은 메시지를 서명자들이 순차적으로 서명하는 순차 다중서명방식(Sequential multi-signature scheme)과 서명자들이 메시지에 동시에 서명하는 효과를 갖게하는 동시 다중서명방식(Simultaneous multisignature scheme)으로 나눌 수 있다.

최초의 다중서명 방식은 소인수 분해 문제를 이용한 RSA 단순서명 방식에 기반을 둔 순차다중서명 방식이다. 그러나 RSA 방식이 연산수가 많기 때문에 이것을 해결하고자 잉여류 문제에 바탕을 둔 Fiat-Shamir 단순서명 방식을 기반으로 하는 다중서명 방식이 개발되었다.

Fiat-Shamir방식은 연산수가 RSA비해 적고 ID-based 방식이어서 키 디렉토리가 필요없는 장점이 있지만, 통신횟수가 많다는 단점이 있다. 후에 이산대수 문제를 이용한 ElGamal 단순서명 방식을 기반으로 하는 다중서명 방식들이 개발되었다. 이산대수문제를 기반으로 하는 경우, Fiat-Shamir 단순서명 방식을 이용하는 다중서명 방식보다 기본 연산수는 많지만 서명자의 수에 영향을 적게 받는다는 장점이 있다.

특히 김성덕 등은 PW 방식의 변형된 서명방식을 이용하여 서명 순서가 제약받지 않고, 추가적인 통신이 필요치 않으며, 서명문의 길이가 일정한 효율적인 순차 다중서명 방식을 제안하였다. 먼저, PW 방식에 대해 설명하고 PW방식의 변형된 서명방식인 효율적인 순차서명 방식에 대해 기술한다.

2.2 PW(Park-Won)방식 : 자체인증 개인식별정보에 기반을 둔 방식

1994년 박성준 등은 자체인증 공개키 방식을 개인

식별정보에 기반을 둔 방식에 적용하여 만든 새로운 개념인 자체인증 개인식별정보(self-certified identity information)방식을 제안하였다.[16]

자체인증 개인식별정보 방식은 자체인증 공개키 방식에서 인증자의 역할을 하는 공개키가 바로 개인식별정보인 경우를 말한다. 여기에서 제안된 자체인증 개인식별정보 방식은 효율성을 개선하기 위해, 사용자의 비밀키 중 인증센터에 의해 생성된 i와 x를 공개키로 사용함으로써 자체인증 공개키 방식으로 변환시킬 수 있다. 제안된 방식의 안전성은 고차 잉여류 문제(γ^{th}-잉여류 문제)와 이산대수 문제의 어려움에 기반을 두고 있다. 각 암호방식의 비교는 <표 2.1>과 같다.

〈표 2.1〉 암호방식의 비교
〈Table 2.1〉 A comparision of cryptosystem

	시스템 구성	공개키(생성)	비밀키(생성)	증자(생성)
공개키 방식	(S, PK)	PK : 가입자	가입자	-
인증자 방식	(I,S,PK,C)	I,PK : 가입자 C : 신뢰센타	가입자	신뢰센터(I,PK)에 대한 전자서명
개인식별 방식	(I,S) PK=I, C=S	I	신뢰센터	신뢰센터
자체인증 공개키 방식	(I,S,PK) PK=C	(I,PK)	가입자	신뢰센터
자체인증 개인식별 방식	(I,S) I=PK=C	I	가입자	신뢰센터

* I : 개인식별정보 S : 비밀키 PK : 공개키 C : 인증자

[시스템의 초기화]

신뢰 센터는 acceptable triple (n, γ^d, y)를 선택한다. 단, n = p · q = $(2\gamma^d fp'+1)(2fq'+1)$, 여기서 f, p', q'는 서로 다른 소수이고, gcd(γ, q') = 1, gcd(γ, p') = 1이다.

y는 modulus n 상에서 $(\gamma^d)^{th}$-비잉여류이고, b는 modulus p와 modulus q상에서의 위수(order)가 f인 Z_n의 원소로 법 n상에서의 위수(order)가 f이다. 신뢰센터의 공개키는 (n, γ^d, y, b, f)이고 비밀키는 (p', q')이다.

이러한 성질을 이용하여 센터는 개인식별정보 ID를 갖는 사용자의 비밀키를 다음과 같이 생성한다.

단계 1) 사용자 A는 자신만의 비밀키 정보 $0 < s < f$를 생성하고 자신의 개인식별정보 I와 b^s (mod n)을 센터에게 전송한다.

단계 2) 센터는 A를 확인한 뒤, ID = $b^s y^{-i} x^{-\gamma^d}$ (mod n)을 만족하는 i와 x를 계산하여 사용자 A에게 안전하게 전송한다.

단계 3) 사용자 A는 (s, i, x)를 자신의 비밀키로서 관리한다.

여기서 각 사용자의 비밀키는 (s, i, x) 이나, 신뢰센터는 사용자가 선택한 비밀키 s를 알 수 없다.

[서명의 생성]

자체인증 개인식별정보 개념을 사용한 서명방식으로 Schnorr방식과 유사하다.

사용자 A가 평문 m을 사용자 B에게 서명하고자 할 경우의 서명 프로토콜은 다음과 같다.

단계 1) A는 [0, f-1]상에 있는 랜덤수 r을 선택하고, v, e를 계산한다.

$$v = b^r \pmod n$$

$$e = h(v, m)$$

여기서 h는 안전한 해쉬 함수이다.

단계 2) A는 다음의 z를 계산한다.

$$z = r + se \pmod f$$

단계 3) 평문 m의 서명문은 (z, e)이다.

단계 4) A는 (z, i, x, e)을 B에게 전송한다.

[서명의 검증]

단계 1) B는 다음의 v를 계산한다.

$$(Iyix\gamma)^e\, b^z \pmod n = v$$

단계 2) B는 e = h (v, m)를 검증한다.

[PW 서명방식의 특성]

위에서 언급한 서명 방식의 특성은 다음과 같다.

1) 완전성(completeness) 특성 : B는 확률 1로 올바른 서명문을 검증한다.
2) 건전성(soundness) 특성 : A의 비밀키 s를 알지 못하는 사용자는 올바른 서명문을 생성할 수 없다.
3) 최소 지식(minimum knowledge) 특성 : 위의 프로토콜은 비밀키 s에 대하여 최소 지식 프로 토콜이다.

2.3 효율적인 순차 다중서명 방식

1995년 김성덕 등은 자체인증특성을 갖는 공개키를 이용한 단순서명 방식인 PW방식과 Schnorr와 ElGamal 방식을 이용한 효율적인 순차 다중서명 방식을 제안하였다.[9,10,11] 이 서명방식은 기 제안된 박성준 등의 고차 잉여류를 이용한 공개키 암호 시스템의 개념에 근거한다. 즉, 신뢰 센터가 사용자의 비밀키를 생성하는 과정에서 고차 잉여류를 이용한 공개키 암호 시스템의 복호화 과정이 요구된다. 제안된 효율적인 다중서명 방식은 다음과 같은 특성을 갖는다.

1) 자체인증 개인식별정보(self-certified identity-information)에 기반을 둔 서명방식이다.
2) 서명자가 많아도 서명의 검증시간이 크게 늘어나지 않는다.
3) 서명 순서가 제약받지 않는다.
4) 통신 복잡도 측면에서 효율적이다.
5) 서명문의 길이가 일정하다.

기 제안된 다중서명 방식은 다음과 같다.

[시스템의 초기화]

시스템의 초기화 과정은 PW 방식과 같다. 센터는 개인식별정보 ID를 갖는 사용자의 비밀키를 다음과 같이 생성한다.

단계 1) 사용자 A는 자신만의 비밀키 정보 $0 < s_A < f$를 생성하고 $b^{s_A} \pmod n$을 센터에게 전송 한다.

단계 2) 센터는 A를 확인한 뒤, $ID_A = b^{s_A} y^{-i_A} x_A^{-f^d} \pmod n$을 만족하는 i_A와 x_A를 계산하여 사용자 A에게 안전하게 전송한다.

단계 3) 사용자 A는 (s_A, i_A, x_A)를 자신의 비밀키로서 관리한다.

여기서 각 사용자의 비밀키는 (s_A, i_A, x_A)이나, 신뢰 센터는 사용자가 선택한 비밀키 s_A를 알 수 없다.

[다중서명의 생성]

생성된 키 정보를 이용하여 사용자가 평문 m에 대한 다중서명문을 생성하는 과정은 다음과 같다.

[서명자$_1$ (기안자)의 서명 생성]

단계 1) 기안자는 메시지를 순차적으로 서명할 사람의 순서를 결정하고 $ID_{cn} = ID_1 \| ID_2 \| \cdots \| ID_n$을 구성한다. 여기서 ID_1은 기안자의 ID이고, ID_n은 최종 서명자의 ID이다.

단계 2) 기안자는 랜덤수 $0 < r_i < f$을 선택하고, k_1, e_1을 계산한다.

$$k_1 = b^{r_1} \pmod n$$
$$e_1 = h(m, k_1)$$

여기서 h는 안전한 해쉬 함수이다.

단계 3) 기안자는 다음의 z_1을 계산한다.

$$z_1 = ms_1 + r_1 e_1 \pmod f$$

단계 4) 기안자는 평문 m에 대한 서명(z_1, k_1)을 다음 서명자에게 전송한다.

[서명자$_i$ (단, $2 \le i \le n$)의 서명 생성]

단계 1) 서명자$_i$는 랜덤수 $0 < r_i < f$을 선택하고, k_i, e_i을 계산한다.

$$k_i = b^{r_i} \pmod n$$
$$e_i = h(m, k_i)$$

단계 2) 서명자$_i$는 다음의 z_i를 계산한다.

$$z_i = z_{i-1} + ms_i + r_i e_i \pmod f$$

단계 3) 또한, 서명자$_i$는 다음의 K_i를 계산한다.

$$e_{i-1} = h(m, k_{i-1})$$
$$K_i = K_{i-1} \cdot k_{i-1}^{e_{i-1}} \pmod n$$

단, $K_1 = 1 \pmod n$ 이다.

단계 4) 서명자$_i$는 평문 m에 대한 서명 (z_i, k_i, K_i)을 서명자$_{i+1}$에게 전송한다. 만약 서명자가 마지막 서명자(서명자$_n$)이면 다중서명을 검증센터로 보낸다.

[다중서명의 검증]

각 서명자와 검증센터는 다음의 절차에 따라 서명을 검증하게 된다.

[서명자$_i$ (단, $2 \le i \le n$)의 서명 검증]

단계 1) 서명자$_i$는 해쉬값 $e_{i-1} = h(m, k_{i-1})$을 계산한다.

단계 2) 서명자$_i$는 다음의 수식이 만족되는지 확인한다.

$$b^{z_{i-1}} \stackrel{?}{=} \left(\prod_{user=1}^{i-1} ID_{user} \cdot y^{i_{user}} \cdot x_{user}^{f^d}\right)^m \cdot (K_{i-1} \cdot k_{i-1}^{e_{i-1}}) \pmod n$$

[검증센터의 서명 검증]

단계 1) 검증센터는 해쉬값 $e_n = h(m, k_n)$을 계산한다.

단계 2) 검증센터는 다음의 수식이 만족되는지 확인한다.

$$b^{z_n} \doteq \left(\prod_{user=1}^{n} ID_{user} \cdot y^{j_{user}} \cdot x_{user}^{\gamma^d}\right)^m \cdot (K_n \cdot k_n^{e_n}) \pmod n$$

단계 3) 검증센터는 평문 m에 대한 다중서명 (ID_{cn}, z_n, k_n, K_n)을 저장보관한다.

제안된 방식은 n명의 서명자가 참여하는 경우에 최소한 n번의 통신횟수가 필요하고, 서명문의 길이가 일정하므로 다른 다중서명 방식에 비해 효율적이다.

3. 기 제안된 서명방식의 문제점 분석

앞절에서 소개된 효율적인 다중서명 방식에서는 정당한 서명자 이외의 제3자가 서명자들의 다중서명을 생성할 수 있다는 문제점이 있다. 즉, 서명자$_j$ (제3자)는 ID_1, …, ID_{j-1}의 비밀키를 모르더라도 다음과 같은 절차로 앞서명자 ID_1, …, ID_{j-1}의 다중서명문을 위조할 수 있다.

[서명자$_j$의 서명 생성]

단계 1) 서명자$_j$는 랜덤수 $0 < r_j < f$을 선택하고, $\underline{k_j}$, e_j을 계산한다.

$$\underline{k_j} = b^{r_j} \pmod n$$
$$e_j = h(m, \underline{k_j})$$

단계 2) 서명자$_j$는 다음의 z_j를 계산한다.

$$z_j = m s_j + r_j e_j \pmod f$$

단계 3) 또한, 서명자$_j$는 K_j를 다음과 같이 계산한다.

$$e_j = h(m, \underline{k_j})$$
$$\underline{K_j} = \left(\prod_{user=1}^{j-1} ID_{user} \cdot y^{j_{user}} \cdot x_{user}^{\gamma^d}\right)^m \pmod n$$

단계 4) 서명자$_j$는 평문 m에 대한 서명 (z_j, $\underline{k_j}$, $\underline{K_j}$)을 다음 서명자에게 전송한다.

위 단계에서 본 것과 같이 정당한 서명자 이외의 제3자가 서명자들의 다중서명을 생성할 수 있는 문제점이 발생된다. 제3자에 의한 서명이 가능함을 다음의 증명과정을 통하여 알 수 있다.

proof)

$$\left(\prod_{user=1}^{j} ID_{user} \cdot y^{j_{user}} \cdot x_{user}^{\gamma^d}\right)^m \cdot (K_j \cdot k_j^{e_j}) \pmod n$$

$$= \left(\prod_{user=1}^{j} ID_{user} \cdot y^{j_{user}} \cdot x_{user}^{\gamma^d}\right)^m \cdot \left(\left(\prod_{user=1}^{j-1} ID_{user} \cdot y^{j_{user}} \cdot x_{user}^{\gamma^d}\right)^m \cdot k_j^{e_j}\right) \pmod n$$

$$= \left(\underline{ID_j} \cdot y^{\underline{j_j}} \cdot \underline{x_j}^{\gamma^d}\right)^m \cdot k_j^{e_j} \pmod n$$

$$= b^{m s_j + r_j e_j} \pmod n$$

$$= b^{z_j} \pmod n$$

단, $e_j = h(m, \underline{k_j})$

4. 고차잉여류 문제에 기반을 둔 다중서명 방식

앞장에서 소개된 효율적인 다중서명 방식에서는 정당한 서명자 이외의 제3자가 서명자들의 다중서명을 생성할 수 있다는 문제점이 있었다. 본 장에서는 2장에서 소개된 다중서명 방식을 개선하여 고차잉여류 문제에 기반을 둔 새로운 다중서명 방식을 제안하고자 한다.[12,13,14]

4.1 시스템의 초기화

시스템의 초기화 절차는 2.2절에서 언급한 PW 방식과 같다. 신뢰 센터는 사용자 A를 확인한 후, $ID_A = b^{s_A} y^{-i_A} x_A^{-\gamma^d} \pmod n$을 만족하는 i_A와 x_A를 계산하여 사용자 A에게 안전하게 전송한다. 여기서 각 사용자의 비밀키는 (s_A, i_A, x_A)이나, 신뢰 센터는 사용자가 선택한 비밀키 s_A를 알 수 없다.

4.2 다중서명의 생성

평문 m에 대한 다중서명문을 생성하는 과정은 다음과 같다.

[절차 1 : 난수 생성 단계]

단계 1) 서명자$_i$는 랜덤수 $0 < r_{i,1} < f$, $0 < r_{i,2} < \gamma^d$, $0 < r_{i,3} < n$을 선택하고, v_i를 계산한다.

$$v_i = (b^{r_{i,1}} y^{r_{i,2}} r_{i,3}^{\gamma^d}) \cdot v_{i-1} \pmod n$$

단, $v_0 = 1 \pmod n$이다.

단계 2) 서명자$_i$는 v_i를 서명자$_{i+1}$에게 전송한다. 만약 서명자가 마지막 서명자(서명자$_n$)이면 v_n를 기안자(서명자$_1$)에게 보낸다.

[절차 2 : 서명 생성 단계]

단계 1) 서명자$_i$는 메시지 m의 해쉬값 e = h(v_n, ID_{cn}, m)을 계산한다.

단계 2) 서명자$_i$는 다음의 $z_{i,1}$, $z_{i,2}$, $z_{i,3}$를 계산한다.

$$z_{i,1} = r_{i,1} + s_i e + z_{i-1,1} \pmod f$$

$$z_{i,2} = r_{i,2} + i_i e + z_{i\ 1,2},$$

$$z_{i,3} = x_i \cdot r_{i,3}^{e} \cdot z_{i-1,3} \pmod n$$

단, $z_{0,1} = 0 \pmod f$, $z_{0,2} = 0$, $z_{0,3} = 1 \pmod n$ 이다.

단계 3) 서명자$_i$는 평문 m에 대한 서명 (ID_{cn}, v_n, $z_{i,1}$, $z_{i,2}$, $z_{i,3}$)을 서명자$_{i+1}$에게 전송한다. 만약 서명자가 마지막 서명자(서명자$_n$)이면 다중서명 (ID_{cn}, e, $z_{i,1}$, $z_{i,2}$, $z_{i,3}$)을 검증센터로 보낸다.

4.3 다중서명의 검증

각 서명자와 검증센터는 다음의 절차에 따라 평문 m에 대한 다중서명을 검증하게 된다.

[서명자$_i$ (단, $2 \le i \le n$)의 서명 검증]

단계 1) 서명자$_i$는 메시지 m의 해쉬값 e = h(v_n, ID_{cn}, m)을 계산한다.

단계 2) 서명자$_i$는 다음의 수식이 만족되는지 확인한다.

$$v_{i-1} \doteq \left(\prod_{user=1}^{i-1} ID_{user}\right)^{e} \cdot b^{z_{i-1,1}} \cdot y^{z_{i-1,2}} \cdot z_{i-1,3}^{r} \pmod n$$

[검증센터의 서명 검증]

단계 1) 검증센터는 다음의 v_n을 계산한다.

$$v_n = \left(\prod_{user=1}^{n} ID_{user}\right)^{e} \cdot b^{z_{n,1}} \cdot y^{z_{n,2}} \cdot z_{n,3}^{r} \pmod n$$

단계 2) 검증센터는 다음의 수식이 만족되는지 확인한다.

$$e \doteq h(v_n, ID_{cn}, m)$$

단계 3) 검증센터는 평문 m에 대한 다중서명 (ID_{cn}, e, $z_{n,1}$, $z_{n,2}$, $z_{n,3}$)을 저장보관한다.

4.4 성능 분석

제안된 방식은 난수 생성 단계와 생성된 난수를 바탕으로 서명을 생성하는 단계로 구성되어 있다. 이 방식은 다중서명의 길이가 증가되지 않고 고속처리와 ID에 근거하기 때문에 RSA에 근거한 서명방식보다 효율적이다. 또한 개인식별정보에 기반을 둔 방식이면서도 센터가 각 사용자의 비밀키를 알 수 없다. 그러나, 이 방식은 Ohta-Okamoto 방식과 같이 n명의 서명자가 다중서명을 수행하고자 할 때 (2n-1)번의 통신을 수행해야하고, 서명자는 첫 번째 라운드에서 생성한 난수를 두 번째 라운드 즉, 메시지를 직접 서명할 때까지 보관해야 하며 또한 첫 번째 라운드와 두 번째 라운드의 서명자 순서가 다른 경우 중간 서명자는 앞 서명자의 서명을 확인할 수 없다.

제안된 방식은 통신 복잡성의 문제를 극복하기 위하여 쉽게 강창구-김대영 방식처럼 변형될 수 있다. 그러나 이 방식은, 강창구-김대영 방식과 같이, 통신 복잡도 측면에서는 효율적이지만 서명문의 길이가 서명자의 수에 따라 증가한다는 단점이 있다.

기존에 제안된 순차 다중서명 방식들과 제안된 다중서명 방식을 비교하면 <표 4.1>과 같다.

〈표 4.1〉 다중서명 방식의 비교
〈Table 4.1〉 A comparision of multisignatures

	서명방식	통신횟수	서명길이	문제점 분석
Itakura Nakamura	RSA	n번	고정	서명순서에 제한이 없다. 연산수가 많다.
Harn Kiesler	RSA	n번	고정	관리해야할 키가 두배이다. 서명순서에 제한이 있다. 연산수가 많다.
Ohta Okamoto	Fiat Shamir	2n번	고정	통신횟수가 많다.
강창구 김대영	Fiat Shamir	n번	증가	서명문 길이가 증가한다.
김성덕 원동호	PW Schnorr Elgamal	n번	고정	정당한 서명자 이외에 제3자가 서명을 생성할 수 있다.
제안한 방식	PW	(2n-1)번	고정	통신횟수가 많다

5. 결 론

본 논문에서는 김성덕외 2인이 제안한 효율적인 순차 다중서명 방식의 문제점을 분석하였으며, 또한 고차 잉여류를 이용한 안전성을 개선시킬 수 있는 방법을 제안하였다. 제안된 방식은 다중서명의 길이가 일정하고, 서명 처리속도가 빠르며, ID에 근거하기 때문에 RSA에 근거한 서명방식 보다 효율적이다. 이 방식은 Ohta-Okamoto 방식과 같이 n명의 서명자가 다중서명을 수행하고자 할 때 (2n-1)번의 통신을 수행해야 한다. 제안된 방식은 통신 복잡성의 문제를

극복하기 위하여 쉽게 강창구-김대영 방식처럼 변형될 수 있다.

또한 제안된 방식은 인터넷이나 LAN 등을 통한 전자 결재 시스템과 같은 응용분야에 효과적으로 이용될 수 있다고 사료된다.

참 고 문 헌

[1] W. Diffie and M. Hellman, "New Direction in Cryptography," IEEE Trans. Inform. Theory, Vol. IT-22, pp.644-654, 1976.

[2] Itakura, K. Nakamura, K. "A public-key cryptosystem suitable for digital multisignatures," NEC J. Res Dev. 71, pp.1-8, Oct. 1983.

[3] T. Okamoto, "A digital multisignature scheme using bijective public-key cryptosystems," ACM Trans. on Comp. Systems, Vol.6, No.8, pp.432-441, 1988.

[4] L. Harn, and T. Kiesler, "New scheme for digital multisignatures," Electronic Letters, Vol.25, No.15, pp.1002-1003, July, 1989.

[5] T. Kiesler and L. Harn, "RSA blocking and multisignature schemes with no bit extension," Electronic Letters, Vol.26, No.18, pp.1490-1491, Aug. 1990.

[6] K. Ohta and T. Okamoto, "A digital multisignature scheme based on the Fiat-Shamir scheme," Proc. of Asiscrypt'91, pp.75-79, 1991.

[7] L. Harn, "New digital signature scheme based on discrete logarithm," Electronic Letters, Vol.30, No.5, pp.396-398, March, 1994.

[8] 강창구, 김대영, "동시성을 갖는 새로운 디지탈 다중서명 방식", 한국통신학회 논문지, Vol.18, No.9, pp.1295-1303, 1993.

[9] 김성덕, 김태훈, 원동호, "효율적인 순차 다중서명 방식", 한국전자공학회 하계종합학술발표회 논문집, pp.215-218, 1995. 6.

[10] S. D. Kim, H. K. Yang and D. H. Won, "An efficient sequential multisignature scheme," ICEIC'95, pp.II-72~II-75, China, 1995. 8.

[11] 김성덕, "다중서명 방식의 모델링 및 효율적인 순차 다중서명 방식에 관한 연구", 성균관대학교, 정보공학과, 석사학위논문, 1995.

[12] Y. Zheng, T. Matsumoto, and H. Imai, "Residuosity Problem and its Application to Cryptography," Trans. IEICE, Vol.E71, No.8, pp. 759-767, 1988.

[13] S. J. Park, B. Y. Lee, and D. H. Won, "A Generalized Public Key Residue Cryptosystem and Its Applications," IEEE GLOBECOM'95, Singapore, pp.1179-1182, 1995. 11.

[14] 박성준, "분산통신망을 위한 확률론적 암호 알고리즘 및 정보보호 프로토콜에 관한 연구", 성균관대학교, 정보공학과, 박사학위논문, 1995.

[15] B. Y .Lee, S. J. Kim and D. H. Won, "ID-based Multisignature Scheme based on the High Residuosity Problem," JWISC'97, pp.227-230. 1997.

[16] S. J. Park and D. H. Won, "A paradoxical identity-based scheme based on r^{th}- residuosity problem and discrete logarithm problem," KIISC Vol.4, No.2, pp.113-118, 1994.

이 보 영

e-mail : bylee@(dosan,ece).skku.ac.kr

1989년 성균관대학교 정보공학과 졸업(공학사)

1995년 성균관대학교 대학원 정보공학과 졸업(공학석사)

1996년~현재 성균관대학교 대학원 전기전자 및 컴퓨터공학부 박사과정

박 택 진

e-mail : T.J.park@yeongdong.ac.kr

1985년 서울산업대학교 전자공학(공학사)

1990년 한양대학교 전자공학(공학석사)

1987년 한국통신 재직

1991년 한국통신기술 재직

1993년~현재 영동전문대학 전자과 교수

원 동 호

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

1976년 성균관대학교 전자공학과 졸업(공학사)

1978년 성균관대학교 대학원 전자공학과 졸업(공학석사)

1988년 성균관대학교 대학원 전자공학과 졸업(공학박사)

1978년~1980년 한국전자통신연구원 연구원

1985년~1986년 일본 동경공대 객원연구원

1996년~현재 성균관대학교 공과대학 전기전자 및 컴퓨터공학부 정교수

관심분야 : 암호이론, 정보이론