通信情報保護學會論文誌
第 2 卷 第 1 號, 1992. 6

새로운 순차 및 동시 다중서명 방식

강창구* · 김대영**

New Sequential and Simultaneous Multisignature Schemes

Chang-Goo Kang and Dae-Young Kim

요 약

컴퓨터로 구축된 전자 사무실에서 문서의 검증과 승인을 위해서 동일한 디지틀문서에 여러사람이 결재하여야 할때 디지틀 다중서명이 요구된다. 본 논문에서는 Fiat-Shamir 방식에 근거한 새로운 순차 다중서명 방식과 동시 다중서명 방식을 제안하고, 그 효율성과 안전성에 대하여 기술하였다. 제안된 다중서명 방식들은 Ohta-Okamoto 방식에서 야기되는 통신 복잡도 문제를 해결하였으며 순차 다중서명 방식에서는 중간 서명자로 하여금 앞서명자들의 서명을 검증할 수 있게 하였다.

제안된 방식은 서명처리 속도면에서 기존의 RSA에 근거한 방식보다 효율적이며 통신 복잡도 측면에서 Ohta-Okamoto 방식 보다 효율적이다. 또한 Fiat-Shamir 방식과 같이 안전성을 유지한다. 본 방식들은 효율적 측면에서 종이없는 전자사무실 구현에 있어서 전자결재 시스템에 적용할 수 있다.

Abstract

Digital multisignature may be needed where several persons sign the same digital message for the verification and approval of the document in the computer-based office.

In this paper we propose a sequential and a simultaneous digital multisignature scheme based on the Fiat-Shamir scheme and discuss the efficiency and security of our schemes. The proposed multisignature schemes overcome the communication complexity problem that arises in the Ohta and Okamoto scheme, and the sequential multisignature scheme enables any intermediate signer to verify the previous multisignature.

* 한국전자통신연구소(ETRI)
** 충남대학교 전자공학과(Dept. of Electronics Eng., Chungnam National Univ.)

Our schemes are more efficient than the RSA based schemes in the processing speed and the Ohta and Okamoto's scheme in the communication complexity, and are as secure as the Fiat-Shamir scheme. High efficiency makes themselves attractive alternatives for electronic approval systems in the paperless electronic offices.

1. 서 론

정보화 사회의 출현과 함께 컴퓨터의 보급확산과 디지틀 통신망의 발전으로 종이없는 전자사무실이 등장하고 있으며, 이러한 전자 사무실 환경에서는 컴퓨터를 이용한 디지틀 메세지 처리가 중요한 역할을 하고 있다. 종이없는 전자 사무실을 구현하기 위하여서는 손으로 쓴 서명대신에 디지틀 서명이 요구되며 또한 디지틀 메세지는 쉽게 복사되거나 변조되기 때문에 디지틀 메세지를 안전하게 보관하기 위한 조치가 강구되어야 한다. 이러한 요구사항을 만족시키기 위해서는 디지틀 메세지에 대한 데이타 무결성(data integrity)과 인증(authenticity)이 보장되어야 하며 이러한 데이타 무결성 및 인증은 디지틀 서명에 의해서 보장될 수 있다.

1976년 Diffie와 Hellman은 공개키 암호시스템의 개념을 처음 소개하였으며[1] 1977년 Diffie와 Hellman의 개념을 실현한 RSA 공개키 암호시스템이 개발되었다.[2] 이 RSA 방법은 디지틀 메세지에 디지틀 서명을 실현 가능하게 하였다. 오늘날까지 많은 공개키 암호시스템이 개발되었으나 그들의 대부분은 디지틀 서명에 사용되고 있다.[3]

대부분의 사무실에서는 계층적 구조를 가지고 있으며, 사무실에서 작성한 문서는 그 문서에 대한 증명과 승인을 위해서 결재가 요구되며 이때 기안자의 서명 뿐만 아니라 상급자의 서명이 요구된다. 이와같이 동일한 디지틀 메세지에 여러 사람이 서명하는 것을 디지틀 다중서명(digital multisignature)이라 한다. 이러한 다중서명에는 두 가지 종류가 있으며 하나는 같은 메세지를 서명자들이 순차적으로 서명하는 순차 다중서명 방식(sequential multisignature scheme)이고, 다른 하나는 서명자들이 같은 메세지를 동시에 서명하는 동시 다중서명 방식(simultaneous multisignature scheme)이다.[4] 지금까지 많은 서명방식들이 개발되었으나 그들의 대부분은 단순서명(single signature) 방식이었다.[2),5),6)] 이러한 단순서명 방식을 직접 반복함으로써 다중서명에 적용할 수 있으나 문서의 길이가 증가하기 때문에 비효율적이다. 이러한 문제를 해결하기 위해서 Itakura와 Nakamura는 두개의 큰 소수와 각 서명자의 직위에 따른 작은 소수의 곱을 이용하여 RSA 방법을 직접 확대 적용한 다중서명 방식을 제안하였다.[7] 이 방식은 서명자의 직위에 따라 정해진 서명자의 키에 의해서 서명 순서가 고정되어 있고, 또한 서명자의 직위가 변동될 때마다 자신의 비밀키를 변경하여야 한다는 단점을 가지고 있다.

Okamoto는 RSA 방식과 같은 전단사(bijective) 공개키 암호시스템과 단방향 함수(one-way function) 를 이용한 다중서명 방식을 제안하였다.[8] 이 방식은 다중서명 메세지의 길이가 거의 증가되지 않고 서명 순서가 제약 받지 않는다는 장점을 가지고 있다. 그러나 이들 디지틀 다중서명 방식은 RSA 방식에 근거하고 있기 때문에 서명을 생성하는데 많은 계산량이 요구되어 서명 처리속도가 느리다는 단점을 가지고 있다.

1986년 Fiat와 Shamir는 ID를 이용한 서명방식을 제안하였다.[9] 이 서명방식은 고속처리와 ID에 근거하기 때문에 RSA에 근거한 서명방식 보다 효율적이다.[10]

Brickell, 이필중 및 Yacobi는 원격회의를 위한 N-party 식별 및 서명방식을 제안하였으며 이 방식은 동시 다중서명 방식이라 할 수 있다.[11] 이 동시 다중서명 방식을 변형하여 Ohta와 Okamoto는 Fiat-Shamir 방식에 근거한 다중서명 방식을 제안하였다.[4] 이 방식은 m명의 서명자가 순차 다중서명을 수행하고자할 때 (2m−1)번의 통신을 수행해야 하고, 서명자는 첫번째 라운드에서 생성한 난수(ran

dom number)를 두번째 라운드 즉, 메세지를 직접 서명할때까지 보관해야 하며 또한 첫번째 라운드와 두번째 라운드의 서명자 순서가 다른 경우 중간 서명자는 앞 서명자의 서명을 확인할 수 없다.

본 논문에서는 Fiat-Shamir 방식에 근거하여 Ohta-Okamoto 방식에서의 통신 복잡성의 문제를 극복할 수 있는 새로운 디지틀 다중서명 방식을 제안하였다. 또한 제안된 방식에 대하여 기존의 방식들과 효율성을 비교하였다.

2. 순차 다중서명 방식

본 논문에서는 m명의 서명자가 다중서명 시스템에 참여하여 같은 메세지를 순차적으로 서명하고 검증자는 다중서명된 서명메세지를 검증한다고 가정한다.

본 논문에 사용되는 기호는 다음과 같이 정의한다.

M=서명할 메세지

f, h=공개된 단방향 함수

ID_i= 서명자 i의 ID(이름, 주민등록 번호, 운전면허 번호등)

ID_{cm}= 서명자들의 ID의 연접(concatenation)
즉, $ID_{cm}=ID_1 \| ID_2 \| \cdots \| ID_m$, 여기서 ‖은 concatenation을 말한다.

k=보안 변수(security parameter)

r_j는 $(I_{ij}^1/N)=1$을 만족하는 랜덤정수로서 1부터 증가한다.

2.1 키 발생 및 배포

본 방식에서 키 발생 및 배포절차는 그림 1과 같이 서명자 i는 자신의 식별정보인 ID_i를 키 발급 센터에 등록하면 키 발급센터는 다음 절차에 의해 키를 발생 배포한다.

단계-1: 키 발급센터(trusted center)는 두개의 큰 소수 p와 q를 선택하고 그들을 비밀히 유지한다.

단계-2: 키 발급센터는 p와 q의 곱인 $n=p^*q$를 공개한다.

단계-3: 키 발급센터는 각 서명자 i에 대하여 S_{ij} $(1 \le j \le k)$를 다음과 같이 계산한다.

$$I_{ij}=f(ID_i, r_j),\ j=1, 2, \cdots, k \quad (1)$$

$$I_{ij}^1=S_{ij}^2 \bmod N \quad (2)$$

단계-4: 키 발급센터는 서명자 i에 대하여 물리적 식별을 한 후 $(N, f, h, S_{i1}, \cdots, S_{ik})$가 기록된 스마트 카드를 발급 배포한다.

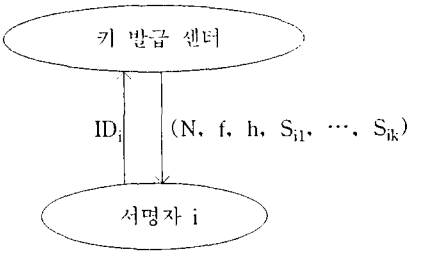

그림 1. 키 발급 절차

2.2 다중서명 발생

본 논문에서 제안한 순차 다중서명 방식은 그림 2와 같이 수행된다.

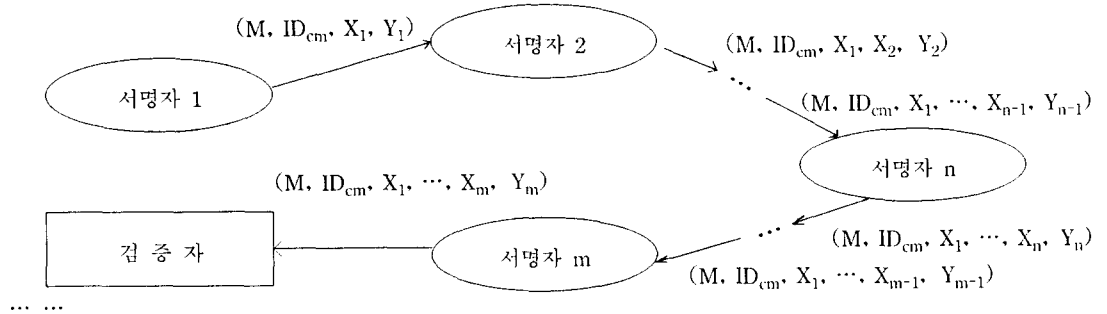

그림 2. 순차 다중서명 절차

가. 서명자 1 (기안자)의 서명발생

단계-1: 기안자는 메세지를 순차적으로 서명할 사람의 순서를 결정하고 $ID_{cm}=ID_1\|ID_2\|\cdots\|ID_m$을 구성한다. 여기서 ID_1은 기안자의 ID이고, ID_m이 최종 서명자의 ID이다.

단계-2: 기안자는 랜덤 수 $R_1 \in Z_N$을 선택한다. 여기서 Z_N은 {0, 1, ···, N-1}을 나타낸다.

그리고 다음을 계산한다.

$$X_1 = R_1^2 \mod N \tag{3}$$

$$(e_{11}, \cdots, e_{1k})=h(M, ID_{cm}, X_1) \tag{4}$$

$$Y_1=R_1 \prod_{e_{1j}=1} S_{1j} \mod N,\ j=1, 2, \cdots, k \tag{5}$$

단계-3: 기안자는 (M, ID_{cm}, X_1, Y_1)을 다음 서명할 ID_2를 가진 서명자에게 전송한다.

나. 서명자 n의 서명발생

단계-1: 서명자 n은 서명자(n-1)로부터 서명 메세지 $(M, ID_{cm}, X_1, \cdots, X_{n-1}, Y_{n-1})$를 받으면 다음 절에 기술된 서명자 n의 검증절차에 의거 앞 서명자들의 서명을 확인한다. 만약 앞 서명자들의 서명을 확인하고 싶지 않다면 이 검증절차는 생략할 수 있다.

단계-2: 서명자 n은 서명을 하기 위하여 랜덤 수 $R_n \in Z_N$을 선택하고 다음을 계산한다.

$$X_n = R_n^2\ X_{n-1} \mod N \tag{6}$$

$$(e_{n1}, \cdots, e_{nk})=h(M, ID_{cm}, X_n) \tag{7}$$

$$Y_n=Y_{n-1}R_n \prod_{e_{nj}=1} S_{nj} \mod N,\ j=1, 2, \cdots, k \tag{8}$$

단계-3: 서명자 n은 $(M, ID_{cm}, X_1, \cdots, X_n, Y_n)$을 다음 서명할 ID_{n+1}을 가진 서명자에게 전송한다. 만약 서명자가 마지막 서명자(서명자 m)이면 $M, ID_{cm}, X_1, \cdots, X_m, Y_m$을 검증자에게 보낸다.

2.3 다중서명 검증

가. 서명자 n의 검증

앞서명자로부터 서명 메세지$(M, ID_{cm}, X_1, \cdots, X_{n-1}, Y_{n-1})$를 받으면 서명자 n은 다음 절차에 의해 서명 메세지를 검증한다.

단계-1: 서명자 n은 $X_1, \cdots, X_{n-1}$로부터 $(e_{11}, \cdots, e_{1k}), \cdots, (e_{(n-1)1}, \cdots, e_{(n-1)k})$를 계산한다.

$$(e_{i1}, \cdots, e_{ik}) = h(M, ID_{cm}, X_i),\quad i = 1, \cdots, n-1. \tag{9}$$

단계-2: 서명자 n은 ID_{cm}으로부터 앞서명자들의 I_{ij}를 계산한다.

$$I_{ij} = f(ID_i, r_j),\ i=1, \cdots, n-1,\ \ j=1, \cdots, k \tag{10}$$

단계-3: 서명자 n은 Y_{n-1}, $(e_{11}, \cdots, e_{1k}), \cdots, (e_{(n-1)1}, \cdots, e_{(n-1)k})$와 I_{ij}를 이용하여 Z_{n-1}을 계산한다.

$$Z_{n-1}=Y_{n-1}^2 \prod_{i=1}^{n-1} \prod_{e_{ij}=1} I_{ij} \mod N,\ j=1, 2, \cdots, k \tag{11}$$

단계-4: 서명자 n은 다음을 점검한다.

$$Z_{n-1}=X_{n-1} \tag{12}$$

만약 $Z_{n-1}=X_{n-1}$이면 다중서명 메세지는 유효(valid)하다고 간주하며 메세지는 앞서명자들에 의해서 서명되었음을 확인할 수 있다.

나. 검증자의 다중서명 검증

검증자가 마지막 서명자로부터 다중서명 메세지 $(M, ID_{cm}, X_1, \cdots, X_m, Y_m)$를 수신하면 $(e_{11}, \cdots, e_{1k}), \cdots, (e_{m1}, \cdots, e_{mk})$을 계산한다.

$$(e_{i1}, \cdots, e_{ik})=h(M, ID_{cm}, X_i),\ \ i=1, \cdots, m \tag{13}$$

그리고 다중서명 검증을 위하여 $(M, ID_{cm}, (e_{11}, \cdots, e_{1k}), \cdots, (e_{m1}, \cdots, e_{mk}), Y_m)$을 저장보관한다. 다중서명 검증이 요구될 때 검증자의 검증절차는 다음과 같다.

단계-1: 검증자는 ID_{cm}으로부터 서명자들의 I_{ij}를 계산한다.

$$I_{ij}=f(ID_i, r_j),\ i=1, 2, \cdots, m \quad j=1, 2, \cdots, k \tag{14}$$

단계-2: 검증자는 Z_m을 다음과 같이 계산한다.

$$Z_m = Y_m^2 \prod_{i=1}^{m} \prod_{e_{ij}=1} I_{ij} \bmod N,\ j=1, 2, \cdots, k \quad (15)$$

단계-3: 검증자는 $h(M, ID_{cm}, Z_m)$을 계산하고 다음식이 만족되는지를 확인한다.

$$(e_{m1}, \cdots, e_{mk}) = h(M, ID_{cm}, Z_m) \quad (16)$$

만약 식(16)이 만족되면 그 다중서명 메세지는 유효한 것으로 판명한다.

모든 서명자가 앞에 기술한 서명절차를 따랐다면 다중서명 메세지는 다음과 같이 유효한 것으로 간주될 것이다. 정의에 의해서

$I_{ij}^{-1} = S_{ij}^2 \bmod N,\ i=1, \cdots, m,\ j=1, 2, \cdots, k$이고, Z_m은 다음과 같이 X_m이 된다.

$$\begin{aligned} Z_m &= Y_m^2 \prod_{i=1}^{m} \prod_{e_{ij}=1} I_{ij} \bmod N \\ &= (Y_{m-1}^2 R_m^2 \prod_{e_{mj}=1} S_{mj}^2) \prod_{i=1}^{m} \prod_{e_{ij}=1} I_{ij} \bmod N \\ &= R_m^2 \cdots R_2^2 R_1^2 \prod_{i=1}^{m} \prod_{e_{ij}=1} S_{ij}^2 I_{ij} \bmod N \\ &= R_m^2 \cdots R_2^2 R_1^2 \bmod N \\ &= X_m \end{aligned} \quad (17)$$

또한 모든 서명자가 앞서명자의 서명을 확인한다면 서명자 n은 X_n 대신에 $(e_{n1}, \cdots, e_{nk})$를 전송함으로써 통신량을 줄일 수 있다. 그때 식(6)은 다음식으로 대치될 수 있고,

$$X_n = R_n^2 Z_{n-1} \quad (18)$$

검증자는 식(9)와 식(13)을 계산할 필요가 없으며 서명자 n의 검증에 있어서도 식(12) 대신에 다음식을 이용하여 앞 서명자들의 서명 메세지를 검증할 수 있다.

$$(e_{(n-1)1}, \cdots, e_{(n-1)k}) = h(M, ID_{cm}, Z_{n-1}) \quad (19)$$

또한 다중서명의 비도를 높히기 위하여 위의 절차를 t회 반복하였을 경우, t회 반복시 서명자 n은 다음과 같이 서명을 수행한다.

$$X_{nt} = R_{nt}^2 X_{(n-1)t} \bmod N \quad (20)$$

$$(e_{(nt)1}, \cdots, e_{(nt)k}) = h(M, ID_{cm}, X_m) \quad (21)$$

$$Y_{nt} = Y_{(n-1)t} R_{nt} \prod_{e_{(nt)j}=1} S_{nj} \bmod N,$$

$$j = 1, 2, \cdots, k \quad (22)$$

최종적으로 $X_{mt} = R_{11}^2 \cdots R_{m1}^2 R_{12}^2 \cdots R_{mt}^2 \bmod N$이 되고, 검증절차는 위와 같은 방식으로 다중서명을 검증할 수 있다.

3. 동시 다중서명 방식

메세지가 bridge node 혹은 통신 버스에 의해서 동시에 여러 서명자들에게 전달될 수 있다면 동시 다중서명 방식이 적용될 수 있다. 동시 다중서명 방식에서 키 발생 및 배포절차는 순차 다중서명 방식과 같다. 본 제안된 동시 다중서명 방식은 그림 3과 같이 수행되며 본장에서는 동시 다중서명 방식에 대한 서명 발생 및 검증절차에 대하여 기술하고자 한다.

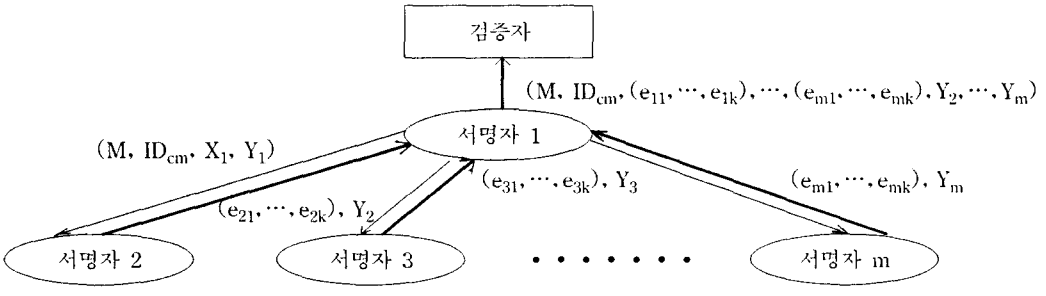

그림 3. 동시 다중서명 절차

3.1 다중서명 발생

가. 서명자 1(기안자)의 서명발생

단계-1: 기안자는 랜덤 수 $R_1 \in Z_N$을 선택하여 다음을 계산한다.

$$X_1 = R_1^2 \bmod N \tag{23}$$

$$(e_{11}, \cdots, e_{1k}) = h(M, ID_{cm}, X_1) \tag{24}$$

$$Y_1 = R_1 \prod_{e_{1j}=1} S_{1j} \bmod N, \quad j = 1, 2, \cdots, k \tag{25}$$

단계-2: 기안자는 메세지를 서명할 모든 사람에게 (M, ID_{cm}, X_1, Y_1)을 동시에 전송한다.

단계-3: 기안자는 각 서명자로부터 $((e_{21}, \cdots, e_{2k}), \cdots, (e_{m1}, \cdots, e_{mk}), Y_2, \cdots, Y_m)$을 수신하면 $(M, ID_{cm}, (e_{11}, \cdots, e_{1k}), \cdots, (e_{m1}, \cdots, e_{mk}), Y_2, \cdots, Y_m)$을 검증자에게 보낸다.

나. 서명자 n의 서명 발생

단계-1: 서명자 n은 기안자로부터 서명 메세지 (M, ID_{cm}, X_1, Y_1)을 수신하면, 먼저 기안자의 서명 메세지를 검증한다. 기안자의 서명 메세지를 검증하는 절차는 다음 절에 기술되어 있다. 서명자가 기안자의 메세지를 검증하고 싶지 않으면 이 검증 절차는 생략될 수 있다.

단계-2: 서명자 n은 랜덤 수 $R_n \in Z_N$을 선택하여 다음을 계산한다.

$$X_n = R_n^2 X_1 \bmod N \tag{26}$$

$$(e_{n1}, \cdots, e_{nk}) = h(M, ID_{cm}, X_n) \tag{27}$$

$$Y_n = Y_1 R_n \prod_{e_{nj}=1} S_{nj} \bmod N, \quad j = 1, 2, \cdots, k \tag{28}$$

단계-3: 서명자 n은 $(e_{n1}, \cdots, e_{nk})$와 Y_n을 기안자에게 전송한다.

3.2 다중서명 검증

가. 서명자 n의 서명검증

서명자 n은 기안자로부터 서명 메세지(M, ID_{cm}, X_1, Y_1)을 받으면 다음과 같이 메세지를 검증한다.

단계-1: 서명자는 X_1으로부터 $(e_{11}, \cdots, e_{1k})$을 계산한다.

$$(e_{11}, \cdots, e_{1k}) = h(M, ID_{cm}, X_1) \tag{29}$$

단계-2: 서명자는 ID_1을 이용하여 I_{1j}을 계산한다.

$$I_{1j} = f(ID_1, r_j), \quad j = 1, 2, \cdots, k \tag{30}$$

단계-3: 서명자는 Z_1을 다음과 같이 계산한다.

$$Z_1 = Y_1^2 \prod_{e_{1j}=1} I_{1j} \bmod N, \quad j = 1, 2, \cdots, k \tag{31}$$

단계-4: 서명자는 $Z_1 = X_1$이 만족되는지를 점검한다. 만약 $Z_1 = X_1$이면 그 메세지는 유효한 것으로 간주하고 기안자에 의해서 서명되었음을 확인할 수 있다.

나. 검증자의 다중서명 검증

검증자는 기안자로부터 다중서명 메세지 $(M, ID_{cm}, (e_{11}, \cdots, e_{1k}), \cdots, (e_{m1}, \cdots, e_{mk}), Y_2, \cdots, Y_m)$을 수신하면 다음과 같은 절차에 의해 다중서명 메세지를 검증한다.

단계-1: 검증자는 ID_{cm}으로부터 각 서명자에 대한 I_{ij}를 계산한다.

$$I_{ij} = f(ID_i, r_j), \quad i = 1, \cdots, m \quad j = 1, \cdots, k \tag{32}$$

단계-2: 검증자는 Y_i, $(e_{i1}, \cdots, e_{ik})$ 및 I_{ij}로부터 다음과 같이 Z_i를 계산한다.

$$Z_i = Y_i^2 \prod_{e_{1j}=1} I_{1j} \prod_{e_{ij}=1} I_{ij} \bmod N, \quad i = 2, \cdots, m, \quad j = 1, \cdots, k \tag{33}$$

단계-3: 검증자는 $h(M, ID_{cm}, Z_i)$을 계산하여 다음식이 성립하는지를 점검한다.

$$(e_{i1}, \cdots, e_{ik}) = h(M, ID_{cm}, Z_i), \quad i = 2, \cdots, m \tag{34}$$

식(34)가 만족되면 다중서명 메세지는 유효한 것으로 간주한다. 모든 서명자가 위의 서명절차를 따랐다면 다중서명 메세지는 다음과 같이 유효한 것

으로 간주될 것이다.

정의에 의해서

$I_{ij}^1 = S_{ij}^2 \bmod N$, $i = 2, \cdots, m$, $j = 1, 2, \cdots, k$ 이고, Z_i는 다음과 같이 X_i가 된다.

$$\begin{aligned} Z_i &= Y_i^2 \prod_{e_{1j}=1} I_{1j} \prod_{e_{ij}=1} I_{ij} \bmod N \\ &= (Y_1^2 R_i^2 \prod_{e_{ij}=1} S_{ij}^2) \prod_{e_{1j}=1} I_{1j} \prod_{e_{ij}=1} I_{ij} \bmod N \\ &= R_i^2 R_1^2 \prod_{e_{1j}=1} (S_{1j}^2 I_{1j}) \prod_{e_{ij}=1} (S_{ij}^2 I_{ij}) \bmod N \\ &= R_i^2 R_1^2 \bmod N \\ &= X_i \end{aligned} \quad (35)$$

4. 효율성과 안전성 검토

본 논문에서는 다중서명 방식의 효율성을 서명처리속도, 통신 복잡도 및 서명 메세지 길이에 대하여 비교 평가하였다. 본 논문에서 제안한 순차 다중서명 방식을 Fiat-Shamir 방식을 직접 반복적용시킨 방식, Ohta-Okamoto가 제안한 방식 및 RSA 방식에 근거한 방식과 비교하였다. 또한 제안한 다중서명 방식의 안전성을 검토하였다.

4.1 서명처리 속도

서명 처리속도는 서명자가 서명을 발생하는데 요구되는 처리량으로 비교평가 하였다. 본 논문에서는 단방향 함수 f, h는 모듈라 곱셈에 비하여 훨씬 빠르므로 모듈라 곱셈의 수만으로 계산하였다. Fiat-Shamir의 직접반복 적용방식은$(k/2+1)*t$번, 본 제안 방식과 Ohta-Okamoto 방식은 $(k/2+3)*t$번의 모듈라 곱셈이 요구된다. 한편 RSA에 근거한 방식은 $1.5*|N|$ 번의 모듈라 곱셈이 요구된다. 여기서 t는 비도를 높이기 위한 다중서명의 반복회수이며, $|N|$은 N의 비트 길이를 의미한다.

예를들면, 2^{80}의 비도를 유지하기 위해 k=80, t=1로 하고, $|N|$=512일때 모듈라 곱셈수는 제안된 방식과 Ohta-Okamoto 방식은 43번이 요구되고, Fiat-Shamir의 직접반복 적용방식은 41번이 요구되나, RSA에 근거한 방식은 768번의 모듈라 곱셈이 요구된다. 따라서 본 제안된 방식은 서명자가 서명을 발생하는 서명 처리속도면에서 RSA에 근거한 방식보다 훨씬 효율적이라 할 수 있다.

4.2 통신 복잡도

m명의 서명자가 다중서명 시스팀에 가입되어 메세지를 순차적으로 서명한다고 할 때 본 제안 방식과 Fiat-Shamir의 직접반복 적용방식 및 RSA에 근거한 방식은$(m-1)$번의 통신으로 다중서명을 수행할 수 있으나, Ohta-Okamoto 방식은 $(2m-1)$번의 통신이 요구된다.

만약 메세지가 bridge node 혹은 통신 버스(bus)에 의해서 동시에 전달될 수 있다면 본 논문에서 제안한 동시 다중서명 방식은 서명자의 수에 관계없이 두 단계로 이루어질 수 있으나, Ohta-Okamoto 방식은 세 단계가 요구되므로 본 논문에서 제안한 방식은 Ohta-Okamoto 방식 보다 통신복잡도 면에서 효율적이다.

4.3 서명길이

본 논문에서 서명 길이를 다중서명을 검증하기 위해서 검증자가 보관하여야 하는 정보의 양으로 계산하였다. 서명자 수가 m명일때 제안된 순차 다중서명 방식에서는 $\{|ID|*m+k*t*m+|N|\}$ 비트가 저장되어야 한다. 한편 Ohta-Okamoto 방식은 $\{|ID|*m+k*t+|N|\}$ 비트, 직접 반복 적용 방식은 $\{(|ID|+k*t+|N|*t)*m\}$ 비트, RSA에 근거한 방식은 $\{|ID|*m+|N|\}$ 비트가 저장되어야 한다. Fiat-Shamir 방식에 근거한 모든방식은 보안 레벨 변수(security level parameter) $k*t$에 따라 서명 길이가 달라진다.

예를들면, m=5이고, $|ID|$ = 104이고, $|N|$ = 512이고, t = 1이고 k=100일때 본 제안된 방식은 1,532 비트, Ohta-Okamoto 방식은 1,132비트, 직접 반복 적용 방식은 3,580비트, RSA에 근거한 방식은

1,032 비트가 저장되어야 한다. 본 제안된 순차 다중서명 방식의 서명길이는 직접 반복적용 방식 보다는 서명 길이가 거의 절반 정도 줄어지지만 Ohta-Okamoto 방식 보다는 약간 길다. 그러나 우리의 방식은 통신복잡도를 줄일 수 있다는 장점을 가지고 있다.

4.4 안전성

본 논문에서 제안한 방식은 Fiat-Shamir 방식에 근거하고 있으며 제안된 방식의 수동 공격(passive attack)에 대한 안전성은 다음 두가지 사항에 달려 있다.

첫째, 다중서명 메세지(ID_{cm}, M, X_i, Y_i)로부터 비밀 정보 S_{ij}를 유도하는 것은 어렵다.[4)]

둘째, 전체 다중서명 메세지가 검증식을 만족 하면서 부분 다중서명 메세지를 변형하기란 어렵다.

이러한 문제들은 본질적으로 Fiat-Shamir 방식에서의 문제와 같으므로 이들 문제를 푸는 것은 Fiat-Shamir 방식의 문제를 푸는 것과 마찬가지로 어렵다.

따라서 우리의 다중서명 방식은 수동공격에 대하여 Fiat-Shamir 방식과 같이 안전하다고 할 수 있다.

5. 결 론

본 논문에서는 Fiat-Shamir 방식에 근거한 새로운 다중서명 방식을 제안하였다. 제안된 디지틀 다중서명 방식은 Fiat-Shamir 방식에 근거하기 때문에 계산량이 RSA 방식에 근거한 다중서명 방식 보다 훨씬 줄어든다.

본 제안된 방식은 비록 서명 길이가 조금 증가하지만 최근에 발표된 Ohta-Okamoto 방식에 있어서 야기되는 통신 복잡도 문제를 해결할 수 있었다. 일반적으로 통신 복잡도는 무시할 수 없는 통신 접속 설정시간과 광 디스크 등과 같은 고 저장(high storage) 장치들 때문에 서명 메세지의 길이(redundancy) 보다 중요하다. 제안된 순차 다중서명 방식은 통신 복잡도가 (m-1)이고, 또한 중간 서명자가 앞 서명자들의 서명을 확인할 수 있을 뿐만 아니라 제안된 동시 서명방식은 두 단계로 다중서명을 수행할 수 있기 때문에 Ohta-Okamoto 방식 보다 효율적이다. 또한 제안된 방식은 Fiat-Shamir 방식과 같이 안전하다.

서명처리 속도와 통신복잡도를 고려한다면 본 논문에서 제안한 방식은 컴퓨터로 구축된 사무실 환경에서 전자 결재 시스템에 효과적으로 적용될 수 있다.

참 고 문 헌

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

2. R.L. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communication of the ACM, Vol. 21, No. 2, pp. 120-126, 1978.

3. D.W. Davies, "Applying the RSA Digital Signature to Electric Mail," IEEE Computer, pp. 55-62, Feb. 1983.

4. K. Ohta and T. Okamoto, "A Digital Multisignature Scheme Based on the Fiat-Shamir Scheme," Proceedings of Asiacrypt'91, pp. 75-79, 1991.

5. T. Okamoto and A. Shiraishi, "A Fast Signature Scheme Based on Quadratic Inequalities," Proceedings of the IEEE Symposium and Privacy, IEEE, pp. 123-132, 1985.

6. L.C. Guillou and J.J. Quisquater, "A Paradoxical Identity-Based Signature Scheme Resulting from Zero-Knowledge," Proceedings of Crypto'88, 1988.

7. K. Itakura and K. Nakamura, "A Public-key Cryptosystem Suitable for Digital Multisignature," NEC J. Res. Dev. 71, pp. 1-8, 1983.

8. T. Okamoto, "A digital Multisignature Scheme Using Bijective Public-Key Cryptosystems," ACM Trans. on Comp. Systems, Vol. 6, No. 8, pp. 432-441, 1988.

9. A. Fiat and A. Shamir, "How to prove

yourself:Practical Solutions to Identification and Signature Problems," Advances in Cryptology-Crypto'86, Lecture Notes in Computer Science 263, pp.186-199, 1987.

10. A. Shamir, "Identity-based Cryptosystems and Signature Schemes," Proceedings of Crypto '84, Lecture Notes in Computer Science 196, pp.47-53, 1985.

11. E.F. Brickell, P.J. Lee and Y. Yacobi, "Secure Audio Teleconference," Advances in Cryptology-Crypto'87, Lecture Notes in Computer Science 293, pp.418-426, 1988.

□ 著者紹介

강 창 구

1957년생
1979년 2월 한국항공대학 항공전자공학과 졸업(공학사)
1986년 2월 충남대학교 대학원 전자공학과(공학석사)
1990년 3월~현재 충남대학교 대학원 전자공학과 박사과정 재학중
1979년~1982년 한국공군 기술장교
현 재 한국전자통신연구소 선임 연구원

김 대 영

1952년생
1975년 2월 서울대학교 공과대학 전자공학과(B.S)
1977년 2월 KAIST 전기 및 전자공학과(M.S)
1983년 2월 KAIST 전기 및 전자공학과(Ph.D)
1978년~1981년 서독 RWTH Aachen, UNI Hannover 공대 연구원
1987년~1988년 미국 University of California Davis 분교 객원 연구원
1983년~1987년 충남대학교 전자공학과 조교수
1987년~현 재 충남대학교 전자공학과 부교수