通信情報保護學會誌
第2卷 第4號, 1992. 12

디지틀 다중서명방식 비교

Comparison of Digital Multisignature Schemes

강창구* · 김대영**

1. 서 론

정보화 사회의 출현과 함께 컴퓨터의 보급확산과 디지틀 통신망의 발전으로 종이없는 전자사무실이 등장하고 있으며, 이러한 전자사무실 환경에서는 컴퓨터를 이용한 디지틀 메세지 처리가 중요한 역할을 하고 있다. 종이없는 전자 사무실을 구현하기 위해서는 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.

손으로 쓴 서명은 최신의 사진복사 기술로 인해 위조하기 쉽고 증명하기 어렵다는 단점을 가지고 있기 때문에 디지틀 서명의 바람직한 특성은 위조하기 어렵고 증명하기 쉬워야 한다. 또한 디지틀 서명은 복사되기 쉽기 때문에 손으로 쓴 서명과는 달리 서명할때마다 달라야 하기 때문에 서명하는 메세지, 비밀번호, time stamp등에 의해서 서명이 이루어지는 것이 바람직하다.

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

1986년 Fiat와 Shamir는 ID를 이용한 서명방식을 제안하였다[4]. 이 서명방식은 고속처리와 ID에 근거하기 때문에 RSA에 근거한 서명방식 보다 효율적이다[5].

대부분의 사무실에서는 계층적 구조를 가지고 있으며, 사무실에서 작성한 문서는 그 문서에 대한 증명과 승인을 위해서 결재가 요구되며 이때 기안자의 서명 뿐만아니라 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[6]. 지금까지 많은 서명방식들이 개발되었으나 그들의 대부분은 단순서명(single signature) 방식이었다[2][7][8]. 이러한 단순서명 방식을 직접 반복함으로써 다중서명에 적용할 수 있으나 문서의 길이가 증가

* 한국전자통신연구소 선임연구원
** 충남대학교 정보통신공학과 교수

하기 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 방식(Itakura-Nakamura 방식)을 제안하였고[9], Okamoto는 RSA 방식과 같은 전단사(bijective) 공개키 암호시스템과 단방향 함수(one-way function)를 이용한 다중서명 방식(Okamoto 방식)을 제안하였다[10]. 이들 디지틀 다중서명 방식은 RSA 방식에 근거하고 있는 방식이다. 또한 Fiat-Shamir 방식에 근거한 방식으로는 Ohta와 Okamoto가 제안한 방식(Ohta-Okamoto 방식)과 본 저자들이 제안한 새로운 다중서명방식이 있다[11),12)].

본 논문에서는 이들 다중서명 방식에 대하여 서명발생 방법과 검증방법을 소개하고 이들 방식의 특징을 분석하였으며 효율성을 서명처리 속도, 서명 메세지 길이 및 통신복잡도 측면에서 분석 비교하였다.

2. 기호 정의

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
E_{ei}	: 키 e_i에 의한 공개키 암호함수
D_{di}	: 키 d_i에 의한 공개키 복호함수
$\|N\|$	: N의 비트길이
$[S]^L$	: S의 ($\|S\|$-L)개의 최상위 비트. 즉 $\|[S]^L\| = \|S\|-L$이다.
$[S]_L$	: S의 L개의 최하위 비트, 즉, $\|[S]_L\| = L$이다.
$^L\{S\}$	: 상위 (L-$\|S\|$)개의 '0' 비트 패딩을 갖는 S. 즉, $\|^L\{S\}\| = L$이다.
$\|\|$	: 연접(concatenation)
ID_i	: 서명자 i 의 ID(이름, 주민등록 번호, 운전면허 번호 등)
ID_{cm}	: 서명자들의 ID의 연접 즉, $ID_{cm} = ID_1\|\|ID_2\|\|\cdots\|\|ID_m$이다.
k	: 보안 변수(security parameter)

3. Itakura-Nakamura 다중서명 방식

본 방식은 RSA 방법을 직접 확장하여 디지틀 다중서명에 적용한 방식으로 상급자의 법 N은 하급자의 법 N 보다 커야하며 각 서명메세지의 길이와 메세지 블럭의 수는 변화되지 않는다.

3.1. 키 발생 및 배포

단계-1 : 두 개의 큰 소수 p, q를 선택하고 서명자 i 의 직위에 따른 작은 소수 r_i를 선택한다.

$$N_i = p\ q\ r_i = N_0 r_i \qquad (1)$$

상급자의 N은 하급자의 N 보다 항상 큰값이 되도록 소수 r_i를 선택하여야 한다.

단계-2 : $\gcd(e, (p-1)(q-1)(r_i-1)) = 1$을 만족하는 임의의 e를 계산한다. 이때 e는 $(p-1)(q-1)(r_i-1)$ 보다 작고 (r_i-1)의 최대값 보다 커야 한다.

단계-3 : $e\ d_i = 1 \bmod (p-1)(q-1)(r_i-1)$을 만

족하는 d_i를 계산한다.

단계-4 : e, N_0, r_i는 공개하고 d_i, p, q는 비밀리 보관한다.

3.2. 다중서명 발생

1) 서명자 1(기안자)의 서명발생

단계-1 : 서명할 메세지 M에 대하여 자신의 비밀키 d_i로 다음과 같이 서명을 수행한다.

$$S_1 = M^{d1} \bmod N_1 \quad (2)$$

단계-2 : 서명메세지 (S_1, M)을 다음 서명자에게 전송한다.

2) 서명자 n의 서명발생

단계-1 : 앞 서명자로부터 서명메세지 (S_{n-1}, M)을 수신하면 앞 서명자의 서명메세지를 점검한다.

$$(\cdots(S_{n-1}^e \bmod N_{n-1})^e \cdots \bmod N_2)^e \bmod N_1 = M \bmod N_1 \quad (3)$$

만약 앞 서명자의 서명을 확인하고 싶지 않으면 이 검증절차는 생략할 수 있다.

단계-2 : 서명자 n은 앞 서명자의 서명(S_{n-1})에 자신의 서명을 수행한다.

$$S_n = S_{n-1}^{dn} \bmod N_n \quad (4)$$

단계-3 : 서명메세지 (S_n, M)을 다음 서명자 n + 1에게 전송한다. 만약 서명자가 마지막 서명자(서명자 m)이면 서명메세지(S_m, M)을 검증자에게 보낸다.

3.3. 다중서명 검증

검증자는 다음식이 만족하는지를 검증한다.

$$(\cdots(S_m^e \bmod N_m)^e \cdots \bmod N_2)^e \bmod N_1 = M \bmod N_1 \quad (5)$$

만약 위 식이 만족되면 다중서명 메세지는 유효한 것으로 간주한다.

4. Okamoto 다중서명 방식

본 방식은 RSA와 같은 전단사 공개키 암호시스템과 단방향 해쉬 함수를 이용한 다중서명 방식이다.

4.1. 키 발생 및 배포

서명자 i는 공개키 e_i와 비밀키 d_i를 발생하고 공개키인 e_i와 단방향 해쉬함수 $h_i : X_i^* X_i^* \cdots X_i \rightarrow X_i$를 공개하고 비밀키 d_i를 비밀리 보관한다.

4.2. 다중서명 발생

1) 서명자 1(기안자)의 서명 발생

단계-1 : 메세지 M에 대하여 다음과 같이 서명 S_1과 M_1을 발생한다.

$$S_1 = D_{d1}(h_1(M)) \quad (6)$$
$$M_1 = M \quad (7)$$

단계-2 : 서명메세지 (S_1, M_1)과 자신의 식별자 ID_1을 다음 서명자에게 전송한다.

2) 서명자 n의 서명 발생

단계-1 : 앞 서명자로부터 서명메세지(S_{n-1}, M_{n-1})을 수신하면 앞 서명자의 서명메세지를 다음절에 기술된 다중서명 검증식(12)-(16)에 의거 점검한다. 만약 두번째 서명자이면 이 검증은 단순서명 검증과 같게 되고 식 (16)에 따른다. 만약 앞 서명자의 서명을 확인하고 싶지 않으면 이 검증절차는 생략할 수 있다.

단계-2 : 서명자 n은 앞 서명자의 서명메세지(S_{n-1}, M_{n-1})에 자신의 서명을 다음과 같이 수행한다.

만약 $|X_n| > |X_{n-1}|$이면

$$S_n = D_{dn}(^{|X_n|}\{S_{n-1}\}) \quad (8)$$
$$M_n = M_{n-1}, \quad (9)$$

그렇지 않으면

$$S_n = D_{dn}(^{|X_n|}\{[S_{n-1}]^{|X_n|^{-1}}\}) \quad (10)$$
$$M_n = M_{n-1} \| [S_{n-1}]^{|X_n|^{-1}}. \quad (11)$$

여기서 X_n은 서명자 n의 평문과 암호문의 유한 집합을 나타낸다.

단계-3 : 서명메세지(S_n, M_n)와 서명자의 식별자 ($ID_1, \cdots, ID_n$)을 다음 서명자 n+1에게 전송한다. 만약 서명자가 마지막 서명자(서명자m)이면 서명메세지(S_m, M_m)과 ($ID_1, \cdots, ID_m$)을 검증자에게 보낸다.

4.3. 다중서명 검증

검증자는 공개키 e_i(i = 1, 2, ···, m)를 이용하여 다중서명 메세지 (S_m, M_m)을 점검한다.
여기서 서명자의 순서는 서명 메세지에 첨부된 서명자의 식별자 ($ID_1, \cdots, ID_m$)에 의하여 표시된다.

단계-1 : 다음식에 의해서 M_i'와 S_i'(i = 1, 2, ···, m)을 구한다. 여기서 $M_m' = M_m$이고 $S_m' = S_m$이다.
만약 $|X_i| > |X_{i-1}|$이면

$$S_{i-1}' = [E_{ei}(S_i')]_{|X_{i-1}|} \quad (12)$$

$$M_{i-1}' = M_i', \quad (13)$$

그렇지 않으면

$$S_{i-1}' = [M_i']_{|X_{i-1}|-|X_i|+1} \| [E_{ei}(S_i')]_{|X_i|-1} \quad (14)$$

$$M_{i-1}' = [M_i']^{|X_{i-1}|-|X_i|+1}, \quad (15)$$

단계-2 : 위의 단계-1에서 얻은 S_1'와 M_1'가 다음식을 만족하면 다중서명 메세지 (S_m, M_m)는 유효한 것으로 간주한다.

$$E_{e1}(S_1') = h_1(M_1') \quad (16)$$

5. Ohta-Okamoto 다중서명 방식

본 방식은 Fiat-Shamir 방식에 근거하고 있으며 그림 1과 같이 수행된다.

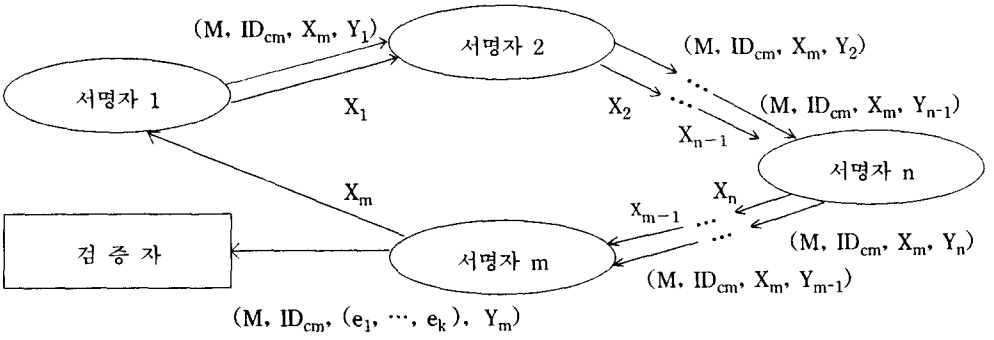

그림 1. Ohta-Okamoto 방식의 다중서명 절차

5.1. 키 발생 및 배포

본 방식에서 키 발생 및 배포절차는 서명자 i가 자신의 식별정보인 ID_i를 키 발급센터에 등록하면 키 발급센터는 다음 절차에 의해 키를 발생 배포한다.

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

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

단계-3 : 키 발급센터는 각 서명자 i에 대하여 S_{ij}를 다음과 같이 계산한다.

$$I_{ij} = f(ID_i, j), \quad j = 1, 2, \cdots, k \quad (17)$$

$$I_{ij}^{-1} = S_{ij}^2 \bmod N \quad (18)$$

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

5.2. 다중서명 발생

1) 공통키 생성 단계

가) 서명자 1(기안자)

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

$$X_1 = R_1^2 \bmod N \quad (19)$$

단계-2 : 기안자는 X_1을 다음 서명자에게 전송한다.

나) 서명자 n

단계-1 : 서명자 n은 앞 서명자로부터 X_{n-1}를 수신하면 랜덤수 $R_n \in Z_N$을 선택하여 다음을 계산한다.

$$X_n = R_n^2 X_{n-1} \bmod N \quad (20)$$

단계-2 : 서명자 n은 X_n을 다음 서명자 n+1에게 전송한다. 만약 서명자가 마지막 서명자 (서명자 m)이면 X_m을 기안자에게 전송한다.

2) 서명 생성 단계

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 다음과 같이 서명을 발생한다.

$$(e_1, \cdots, e_k)=h(M, ID_{cm}, X_m) \quad (21)$$

$$Y_1=R_1 \prod_{e_j=1} S_{1j} \bmod N, \ j=1, 2, \cdots, k \quad (22)$$

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

나) 서명자 n의 서명발생

단계-1: 서명자 n은 서명자(n-1)로부터 서명 메세지 $(M, ID_{cm}, X_m, Y_{n-1})$를 수신하면 다음을 계산한다.

$$(e_1, \cdots, e_k)=h(M, ID_{cm}, X_m) \quad (23)$$

$$Y_n=Y_{n-1}R_n \prod_{e_j=1} S_{nj} \bmod N, \ j=1, 2, \cdots, k \quad (24)$$

단계-2: 서명자 n은 (M, ID_{cm}, X_m, Y_n)을 다음 서명할 ID_{n+1}을 가진 서명자에게 전송한다.

단계-3 : 서명자가 마지막 서명자 (서명자 m)이면 $(M, ID_{cm}, (e_1, \cdots, e_k), Y_m)$을 검증자에게 전송한다.

5.3. 다중서명 검증

검증자가 마지막 서명자로부터 다중서명 메세지 $(M, ID_{cm}, (e_1, \cdots, e_k), Y_m)$를 수신하면 공개된 법 N과 단방향 함수 f, h를 이용하여 다음과 같이 다중서명을 검증한다.

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

$$I_{ij}=f(ID_i, j), \ i=1, 2, \cdots, m, \ j=1, 2, \cdots, k \quad (25)$$

단계-2 : 검증자는 Z_m을 다음과 같이 계산한다.

$$Z_m=Y_m^2 \prod_{i=1}^{m} \prod_{e_j=1} I_{ij} \bmod N, \ j=1, 2, \cdots, k \quad (26)$$

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

$$(e_1, \cdots, e_k) = h(M, ID_{cm}, Z_m) \quad (27)$$

만약 식(27)이 만족되면 그 다중서명 메세지는 유효한 것으로 판명한다.

6. 새로운 다중서명 방식

본 다중서명 방식은 Fiat-Shamir 방식에 근거하고

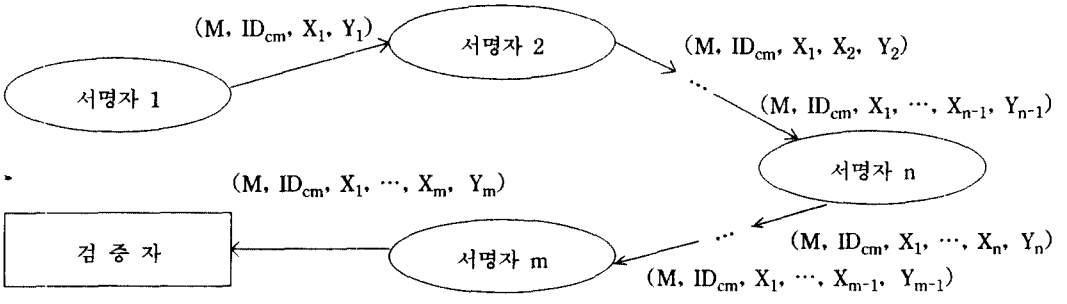

그림 2. 새로운 다중서명 방식의 다중서명 절차

있으며 그림 2와 같이 수행된다.

6.1. 키 발생 및 배포

본 방식에서 키 발생 및 배포절차는 서명자 i가 자신의 식별정보인 ID_i를 키 발급센터에 등록하면 키 발급센터는 다음 절차에 의해 키를 발생 배포한다.

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

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

단계-3 : 키 발급센터는 각 서명자 i에 대하여 S_{ij}를 다음과 같이 계산한다.

$$I_{ij} = f(ID_i, j), \quad j = 1, 2, \cdots, k \tag{28}$$

$$I_{ij}^{-1} = S_{ij}^2 \bmod N \tag{29}$$

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

6.2. 다중서명 발생

1) 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 랜덤수 $R_1 \in Z_N$을 선택한다. 여기서 Z_N은 {0, 1, ⋯, N−1}을 나타낸다. 그리고 다음을 계산한다.

$$X_1 = R_1^2 \bmod N \tag{30}$$

$$(e_{11}, \cdots, e_{1k}) = h(M, ID_{cm}, X_1) \tag{31}$$

$$Y_1 = R_1 \prod_{e_{1j}=1} S_{1j} \bmod N, \quad j=1, 2, \cdots, k \tag{32}$$

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

2) 서명자 n의 서명발생

단계-1: 서명자 n은 서명자(n−1)로부터 서명 메세지 (M, ID_{cm}, X_1, ⋯, X_{n-1}, Y_{n-1})를 받으면 다음절에 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.

$$X_n = R_n^2 X_{n-1} \bmod N \tag{33}$$

$$(e_{n1}, \cdots, e_{nk}) = h(M, ID_{cm}, X_n) \tag{34}$$

$$Y_n = Y_{n-1} R_n \prod_{e_{nj}=1} S_{nj} \bmod N, \quad j=1, 2, \cdots, k \tag{35}$$

단계-3: 서명자 n은 (M, ID_{cm}, X_1, ⋯, 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, ⋯, X_m, Y_m을 검증자에게 보낸다.

6.3. 다중서명 검증

1) 서명자 n의 검증

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

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

$$(e_{i1}, \cdots, e_{ik}) = h(M, ID_{cm}, X_i), \quad i = 1, \cdots, n-1. \tag{36}$$

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

$$I_{ij} = f(ID_i, j), \quad i=1, \cdots, n-1, \quad j=1, \cdots, k \tag{37}$$

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

$$Z_{n-1} = Y_{n-1}^2 \prod_{i=1}^{n-1} \prod_{e_{ij}=1} I_{ij} \bmod N, \quad j=1, 2, \cdots, k \tag{38}$$

단계-4: 서명자 n은 다음을 점검한다.

$$Z_{n-1} = X_{n-1} \tag{39}$$

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

2) 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.

$$(e_{i1}, \cdots, e_{ik})=h(M, ID_{cm}, X_i), \quad i=1, \cdots, m \quad (40)$$

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.

$$I_{ij}=f(ID_i, j), \quad i=1, 2, \cdots, m \quad j=1, 2, \cdots, k \quad (41)$$

단계-2: 검증자는 Z_m을 다음과 같이 계산한다.

$$Z_m=Y_m^2 \prod_{i=1}^{m} \prod_{e_{ij}=1} I_{ij} \bmod N, \quad j=1, 2, \cdots, k \quad (42)$$

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

$$(e_{m1}, \cdots, e_{mk}) = h(M, ID_{cm}, Z_m) \quad (43)$$

만약 식(43)이 만족되면 그 다중서명 메세지는 유효한 것으로 판명한다.

모든 서명자가 앞 서명자들의 서명을 검증한다면 서명자 n은 X_n 대신에 $(e_{n1}, \cdots, e_{nk})$를 전송함으로써 통신량을 줄일 수 있다. 그때 식(33)은 다음식으로 대치될 수 있고,

$$X_n=R_n^2 Z_{n-1} \bmod N \quad (44)$$

또한 검증자는 식(36)와 식(40)을 계산할 필요가 없으며 서명자 n의 검증에 있어서도 식 (39) 대신에 다음식을 이용하여 앞 서명자들의 서명 메세지를 검증할 수 있다.

$$(e_{(n-1)1}, \cdots, e_{(n-1)k}) = h(M, ID_{cm}, Z_{n-1}) \quad (45)$$

7. 방식별 특징 및 효율성 분석

7.1. 방식별 특징

Itakura-Nakamura방식은 RSA 방법을 직접 확장하여 디지틀 다중서명에 적용하였다. 이 방식은 RSA 방법을 직접 반복 적용시 서명메세지의 길이 증가 및 서명 발생 속도의 문제점을 해결하였으며 상급자의 법 N이 하급자의 법 N 보다 항상 크기 때문에 다중서명 메세지의 블럭수와 길이가 증가되지 않는다는 장점을 가지고 있으나 서명 순서가 제약받게 된다. 또한 서명자의 직위가 변경되면 서명자의 공개키는 변경하지 않아도 되나 비밀키는 변경하여야 한다.

Okamoto 방식은 Itakura-Nakamura 방식의 서명 순서가 제약받는다는 단점을 개선하였으며 다중서명 메세지의 길이는 단순서명 메세지의 길이와 거의 같다. 또한 단방향 해쉬함수를 사용함으로써 다중서명 발생 및 검증을 효율적으로 처리할 수 있으며 RSA 뿐만 아니라 어떠한 전단사 공개키 암호시스팀으로도 구성할 수 있다. 이 방식은 Itakura-Nakamura 방식과 마찬가지로 RSA 방식에 근거하고 있기 때문에 많은 계산량이 요구되며 서명 처리속도가 느리다는 단점을 가지고 있다.

Ohta-Okamoto 방식과 본 저자들이 제안한 새로운 방식은 Fiat-Shamir 방식에 근거하고 있기 때문에 서명속도가 RSA에 근거한 방식 보다 20배 정도 빠르고, ID에 근거한 서명방식으로 공개키 디렉토리가 불필요하며 키 관리를 단순화 할 수 있다. 또한 서명순서가 제약받지 않으며 서명메세지의 길이는 보안레벨에 의해서 결정된다. Ohta-Okamoto 방식은 통신 복잡도 문제를 가지고 있으며 본저자들의 새로운 방식은 이러한 문제점을 해결하였다.

7.2. 서명처리 속도

서명처리 속도는 서명자가 서명을 발생하는데 요구되는 처리량으로 평가하였다. 본 논문에서는 단방향 함수 f, h는 모듈라 곱셈에 비하여 훨씬 빠르

므로 모듈라 곱셈의 수만으로 계산하였다. RSA에 근거한 Itakura-Nakamura 방식과 Okamoto 방식은 1.5*|N|번의 모듈라 곱셈이 요구되고, Ohta-Okamoto 방식과 본저자들의 새로운 방식은 (k/2 + 3)*t 번의 모듈라 곱셈이 요구된다. 여기서 t는 비도를 높이기 위한 다중서명의 반복회수이며, |N|은 법 N의 비트 길이를 의미한다.

예를들면, k = 80이고, t = 1이고, |N| = 512일때 모듈라 곱셈수는 Ohta-Okamoto 방식과 본저자들의 새로운 방식은 43번이 요구되고, Itakura-Nakamura 방식과 Okamoto 방식은 768번의 모듈라 곱셈이 요구된다. 따라서 Fiat-Shamir 방식에 근거한 방식은 서명자가 서명을 발생하는 서명처리 속도면에서 RSA에 근거한 방식 보다 훨씬 효율적이라 할 수 있다.

7.3. 통신 복잡도

m명의 서명자가 다중서명 시스템에 가입되어 메세지를 순차적으로 서명한다고 할 때 Itakura-Nakamura 방식, Okamoto 방식 및 본저자들의 새로운 방식은 (m − 1)번의 통신으로 다중 서명을 수행할 수 있으나, Ohta-Okamoto 방식은 (2m − 1)번의 통신이 요구된다. 통신복잡도 측면에서는 Ohta-Okamoto 방식이 비효율적이다.

7.4. 서명 길이

본 논문에서는 서명 길이를 다중서명을 검증하기 위해서 검증자가 보관하여야 하는 서명 정보의 양으로 계산하였다. 서명자 수가 m명 일때 Itakura-Nakamura 방식과 Okamoto 방식은 [|ID|*m + |N|] 비트가 저장되어야 하고, Ohta-Okamoto 방식은 [|ID|*m + k*t + |N|] 비트, 본저자들의 새로운 방식은 [|ID|*m + k*t*m + |N|] 비트가 저장되어야 한다. Ohta-Okamoto 방식과 본저자들의 새로운 방식은 보안 레벨 변수(security level parameter) k*t 에 따라 서명길이가 달라진다.

예를들면, m = 5이고, |ID| = 104이고, |N| = 512 이고, k = 80일때 Itakura-Nakamura 방식과 Okamoto 방식은 1,032비트, Ohta-Okamoto 방식은 1,112비트, 본저자들의 새로운 방식은 1,432비트가 저장 되어야 한다.

8. 결 론

본 논문에서는 지금까지 개발된 주요 디지틀 다중서명 방식들을 소개하고 각 방식의 특징 및 효율성을 분석하였다. Itakura-Nakamura 방식은 RSA 방식을 다중서명에 직접 적용시 발생되는 메세지 길이의 증가 및 서명 발생 속도를 개선하였으나 서명순서가 제약받고, 서명자의 직위 변동시 비밀키를 변경하여야 한다는 단점을 가지고 있다. Okamoto 방식은 이러한 문제점을 개선하였으며 단방향 함수를 이용하여 다중서명 생성 및 검증을 효율적으로 처리할 수 있게 하였다. 그러나 이들 방식은 기본적으로 RSA 방식에 근거하고 있기 때문에 계산량이 많고 서명속도가 느리다는 단점을 가지고 있다.

Ohta-Okamoto 방식과 본저자들의 새로운 방식은 Fiat-Shamir 방식에 근거하고 있기 때문에 Fiat-Shamir 방식의 모든 장점을 가지고 있다. Ohta-Okamoto 방식은 서명메세지의 길이를 줄였으며 본저자들의 새로운 방식은 서명메세지 길이와 통신복잡도 문제를 개선하였다.

디지틀 다중서명은 종이없는 전자 사무실 환경에서 문서 결재 시스템을 구축하는데 필수적인 요소로서 앞으로 서명 메세지 길이, 서명처리 속도, 통신복잡도 등의 효율성을 보다 개선하고 능동공격에 대해서도 안전한 다중서명 방식에 관한 연구가 요구된다.

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. 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.

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

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

7. 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.

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

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

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

11. 강창구, 김대영, "순차적 다중서명 방식", 한국통신학회 하계종합학술발표회 논문집, pp. 31-35, 1992.

12. 강창구, 김대영, "새로운 순차 및 동시 다중서명 방식", 한국통신정보보호학회논문지, 제 2 권 1 호, pp. 36-44, 1992.

□ 著者紹介

강 창 구

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

김 대 영

1952년 5월생
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 Califormia Davis 분교 객원연구원
1983년～1987년 충남대학교 전자공학과 조교수
1987년～현재 충남대학교 정보통신공학과 교수