論文 94-19-6-1

EDI 메시지 동보 전송에 적합한 디지탈 다중 서명 방법

正會員 尹 盛 鉉* 正會員 金 泰 潤*

A Digital Multisignature Scheme Suitable for Transmission to Multi-destination by EDI Message

Sung-Hyun Yun*, Tai-Yun Kim* *Regular Members*

要 約

EDI 메시지는 법적 구속력을 갖는 상용 전자 문서이므로 메시지의 무결성과 시래처간의 신분 확인을 위한 디지탈 서명 방법이 필요하다. 본 연구에서는 EDI 메시지의 동보 전송에 적합한 새로운 디지탈 다중 서명 방법을 제안하였다.

제안한 방법은 Fiat-Shamir 서명 방식에 기반을 두고 있으며, EDI 메시지에 적합한 메시지 인증을 수행할 수 있다. 침입자 및 서명자가 해쉬 함수공격으로 조작한 메시지를 검증할 수 있다. 침입자에 의한 가장 및 재전송과 같은 능동적 공격에 안전하다. 또한 Fiat-Shamir 서명 방법보다 작은 해쉬 값을 사용하여 디지탈 서명을 수행할 수 있다. EDI 메시지의 동보 전송시 지정된 수신자들은 빠르고 안전하게 디지탈 서명을 수행할 수 있다.

ABSTRACT

As the EDI message is the commercial electronic document having legal binding forces, it is necessary to use the method of digital signature for the message integrity and identification between trading partners. This research proposes a new digital multisignature scheme suitable for transmission to multi-destination of the EDI message.

The proposed scheme is based on Fiat-Shamir signature scheme and can perform the message authentication suitable for the EDI message. It can verify the message forged by a hash function attack from intruder or signer. It is safe against an active attack such as masquerade or retransmission by intruder. It also can perform the digital signature using smaller hash value than that of Fiat-Shamir signature scheme. When the EDI message is transmitted to multi-destination, all designated receivers can perform the digital signature faster and safer.

*高麗大學校 電算科學科
論文番號 : 93185
接受日字 : 1993年 10月 4日

I. 서 론

EDI(Electronic Data Interchange)란 기업간의 문서 교환을 표준화된 거래 서식에 맞게 컴퓨터를 이용하여 전자적으로 수행하는 방법이다. EDI 메시지는 상용 전자 문서이기 때문에 법적 구속력을 갖는다. 메시지의 무결성과 거래처간의 신분 확인을 보장할 수 있는 디지탈 서명 방법이 필요하다. 특히, EDI 메시지는 기업간의 이해 관계에 관련된 상용 문서이므로 일반 문서보다 강력한 메시지 인증 방법이 요구된다[11, 18, 19].

기업간의 EDI 메시지 교환에 있어서 부인 봉쇄 기능은 매우 중요하다. 서명 사실에 대한 부인, 메시지 내용에 대한 부인, 전송 사실에 대한 부인 등이 있다. 특히 서명자가 메시지 내용을 임의로 조작한 후 원래 메시지의 전송 사실을 부인할 수 있다. 해쉬 함수 공격으로 변조된 메시지는 이 경우 잘못된 메시지 임을 밝힐 수 없다. 디지탈 다중 서명 방법의 특성을 이용하여 서명자들이 검증 가능한 메시지를 만들 수 없도록 하는 안전한 서명 방법이 필요하다.

디지탈 다중 서명(digital multisignature)은 여러 사람의 서명이 요구되는 메시지에 대한 디지탈 서명 기법이다. 기업에서 하나의 문서를 대외적으로 제출하기 까지 여러 사람의 확인과 서명을 거쳐야 하는데, 이 경우 다중 서명 방법이 필요하다. 다중 서명의 안전도는 단순 서명과 똑 같은 안전도를 가져야 하며 부분적 다중 서명(partial multisignature)으로부터 서명 위조가 불가능해야 한다.

지금까지 개발된 대표적인 다중 서명 기법은 RSA 서명 방식을 이용한 방법과 Fiat-Shamir 서명 방식을 이용한 방법이 있다.

RSA 방법은 공개 키와 비밀 키를 이용하여 서명하는 방식으로 키 분배의 문제점을 해결한 안전도가 높은 서명 방식이다. RSA 서명 방식에 기반을 둔 Itakura-Nakamura 다중 서명 방식과 Okamoto 다중 서명 방식은 모듈라 연산 수가 많고 공개 디렉토리를 유지해야 하는 단점이 있다[6, 7].

Fiat-Shamir 방법은 스마트 카드를 이용하여 개인의 ID 만을 공개하여 서명하는 방식이다. 상대 방의 공개 키 관리를 위한 디렉토리가 필요없으며, RSA 방식보다 모듈라 연산 수가 적다. 보안 요구 조건에 따라 안전도를 조정할 수 있는 효율적인 서명 방법이다. Ohta-Okamoto가 제안한 디지탈 다중 서명 방식은 Fiat-Shamir 방식에 기반을 두었으며 수동적 공격에대한 다중 서명 방식의 안전도는 단순 서명 방식에서의 안전도와 같다는 것을 증명하였다[8].

디지탈 다중 서명 방법은 단순 서명 방법보다 서명자 및 통신 횟수의 증가로 인하여 많은 위험 요소를 갖게 된다. EDI 메시지는 기업간의 이해 관계에 관련된 점예한 기업 정보로 구성된다. 서명자의 내부 부정을 막을 수 있는 부인 봉쇄 기능과 함께 능동적 공격에 안전한 다중 서명 방법이 요구된다.

기업간에 EDI 메시지를 전송할 경우 같은 메시지를 여러 수신자에게 전송해야 하는 경우가 있다. 이를 동보 전송이라고 한다. EDI 메시지는 법적 구속력을 갖는 상용 전자 문서이기 때문에 동보 전송시 지정된 수신자들이 모두 같은 메시지를 받았다는 것을 검증할 수 있어야 한다.

본 연구에서는 Fiat-Shamir 서명 방식에 기반을 둔 EDI 메시지에 적합한 디지탈 다중 서명 방법을 제안하였다. 양단간의 메시지 전송 뿐만 아니라 한 기업에서 여러 수신처로 안전하게 메시지를 전송할 수 있다. 해쉬 함수 공격으로 변조된 메시지를 검증할 수 있고 재전송이나 가장과 같은 능동적 공격에 안전한 서명 방법이다.

II. Fiat-Shamir 서명 방식에 기반을 둔 디지탈 다중 서명 방법

Fiat-Shamir 서명 방식은 서명자의 고유한 식별 정보인 ID 만을 공개하여 디지탈 서명을 수행한다. 기존의 RSA 서명 방법보다 연산 속도가 빠르고 공개 키 관리의 어려움이 없는 효율적인 서명 방법이다. 2.1에서 Fiat Shamir 단순 서명 방식에 대해서 알아보고, 2.2에서 이 방식에 기반을 둔 Ohta-Okamoto 다중 서명 방식에 대해서 알아본다.

2.1 Fiat-Shamir 서명 방법[4]

87년 Fiat-Shamir는 ID를 이용한 개인 식별 방법 및 디지탈 서명 방법을 제안하였다. ID를 이용한 암호 시스템은 각 서명자의 고유한 ID에 대응하는 비밀 키를 생성해주는 신뢰할 수 있는 센터가 필요하다. 센터만이 자신의 비밀 정보를 이용하여 서명자의 비밀 키를 생성할 수 있기 때문에, 센터 이외의 사용자에 의한 ID 변경 및 비밀 키 생성은 불가능하다.

Fiat-Shamir 서명 방식은 스마트 카드 발급, 서명자들의 서명 생성 그리고 검증자의 검증 과정으로 이

루어진다. 각각에 대한 설명은 다음과 같다.

그림 1은 센터의 스마트 카드 발급 절차를 보여준다. 그림 2는 Fiat-Shamir 서명 방법을 이용하여 서명자와 검증자간에 디지탈 서명을 수행하는 과정이다.

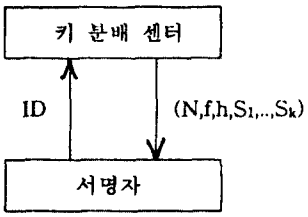

그림 1. 센터의 스마트 카드 생성
Fig. 1. Smart card generation by the center

단계 1. 신뢰할 수 있는 센터(trusted center)의 스마트 카드 발급

① 센터는 두 개의 큰 소수 p, q를 선택하고 모듈라 곱셈의 법 N을 구한다(N=p*q). 센터의 비밀 정보인 p, q는 안전하게 보관한다. 법 N은 공개 정보이므로 서명자들에게 공개 한다. N 값이 작을 경우 비밀 정보인 p, q를 지금까지 개발된 factoring 알고리즘과 고속 연산 chip을 이용하여 유추해낼 수 있다. 따라서 이러한 공격 법에 안전하기 위해서 공개 정보인 N 값은 최소 512 비트 이상의 큰 값이어야 한다.

② 사용자의 ID로부터 공개 정보 V를 구한다. f는 스마트 카드에 내장되는 해쉬 함수로 각 사용자의 ID 값을 입력으로 받아서 서명 검증에 필요한 V 값을 생성해낸다. I는 스마트 카드 발급을 원하는 사용자의 이름, 주소 등이 결합된 ID 정보다. j 값으로 작은 정수 값을 선택한다.

$$V_i = f(I, j)$$

③ ②에서 구한 V_i 값 중에서 법 N에 대한 평방 잉여(quadratic residue)인 값을 k개 선택한다. 사용자의 비밀 정보 S를 계산한다. k는 스마트 카드에 저장될 비밀 정보의 개수이다.

$$S_i^2 = \frac{1}{V_i} \bmod N \qquad (i = 1, \ldots, k)$$

④ (N, f, I, S_1, ..., S_k)가 기록된 스마트 카드를 발급한다.

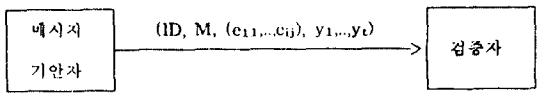

그림 2. Fiat-Shamir 디지탈 서명 방식
Fig. 2. The Fiat-Shamir digital signature scheme

단계 2. 메시지 기안자의 서명 생성

① 서명자는 0 보다 크고 N 보다 작은 t 개의 난수 $r_1, \ldots, r_t$를 구한다. 메시지 해쉬에 필요한 t 개의 x_i를 계산한다.

$$x_i = r_i^2 \bmod N \qquad (i = 1, \ldots, t)$$

② 메시지를 해쉬하여 그 결과 값의 처음 kt 비트를 구한다.

$$e_{ij} = f(M, x_1, \ldots, x_t) \qquad (1 \le i \le t,\ 1 \le j \le k)$$

③ ①에서 구한 난수, ②에서 계산한 해쉬 결과 값 그리고 서명자의 비밀 정보를 이용하여 다음과 같이 y_i를 구한다.

$$y_i = r_i \times \prod_{e_{ij}=1} S_j \bmod N \qquad (1 \le i \le t,\ 1 \le j \le k)$$

④ (I, M, (e_{11}, ..., e_{ij}), y_1, ..., y_t)를 검증자에게 전송한다.

단계 3. 검증자의 서명 검증

① 서명자의 개인 식별 정보인 I를 이용하여 검증에 필요한 k 개의 V_i 값을 구한다.

$$V_i = f(I, j) \qquad (1 \le i \le k)$$

② 검증 값 z_i를 구한다.

$$z_i = y_i^2 \times \prod_{e_{ij}=1} V_j \bmod N \qquad (1 \le i \le t,\ 1 \le j \le k)$$

③ 검증자는 ②에서 구한 z_i 값을 이용하여 수신한 메시지를 해쉬한다. 해쉬 결과 값이 수신한 e_{ij} 값과 같으면 올바른 서명으로 검증한다. 같지 않으면 잘못된 서명임을 알 수 있다.

$e_{ij} = f(M, z_1, \ldots, z_t)$: 메시지 기안자의 서명임을 확인한다.

$e_{ij} \neq f(M, z_1, \ldots, z_t)$: 메시지 기안자의 서명이 아님을 확인한다.

2.2 Ohta-Okamoto 디지탈 다중 서명 방법[8]

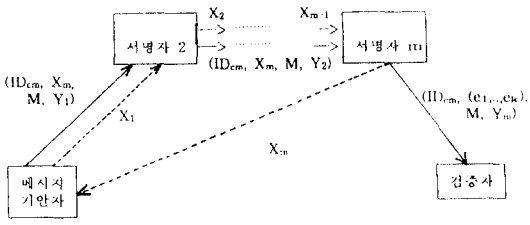

그림 3. Ohta-Okamoto 디지탈 다중 서명 방법
Fig. 3. The Ohta-Okamoto digital multisignature scheme

서명자의 수가 m 명일 때 Ohta-Okamoto 다중 서명 방법은 다음과 같다. 스마트 카드 발급은 2.1의 단계 1과 같다. 그림 3은 Ohta-Okamoto 방법을 이용하여 디지탈 다중 서명을 수행하는 과정이다.

(1) 공통 키 생성

다중 서명 검증에 필요한 공통 키 X_m을 서명자들이 발생한 난수 정보로 구성된다. 서명자들간의 m 번의 통신으로 다음과 같이 공통 키 X_m을 구한다. 그림 3의 점선으로 표시된 부분이다.

단계 1 메시지 기안자의 공통 키 생성

① 메시지 기안자는 난수 R_1을 선택한다. 난수는 0보다 크고 모듈라 곱셈의 법 N 보다 작아야 한다.

② 공통 키 생성을 위하여 다음과 같이 X_1을 구한 후 다음 서명자에게 전송한다.

$$X_1 = R_1^2 \bmod N$$

단계 2. 서명자 n의 공통 키 생성

① 서명자 n은 이전 서명자로부터 X_{n-1}을 수신한다. 자신이 발행한 난수 R_n을 가지고 다음과 같이 X_n을 구한다.

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

② X_n을 다음 서명자에게 전송한다. 서명자 n이 마지막 서명자이면 X_n을 메시지 기안자에게 전송한다.

(2) 다중 서명 발생 및 검증

(1)로부터 생성된 공통 키 X_m을 이용하여 다음과 같이 디지탈 서명을 수행한다. 그림 3에서 다중 서명의 진행 과정을 실선으로 표시된다.

단계 1. 메시지 기안자의 서명 발생

① 메시지 기안자는 메시지에 서명할 서명자들의 순서를 정한다. 서명자들의 ID를 서명 순서대로 결합한 ID_{cm}을 만든다.

$$ID_{cm} = ID_1 \| ID_2 \| \ldots \| ID_m$$

② 서명할 메시지 M, ID_{cm} 그리고 공통 키 X_m을 이용하여 서명 생성에 필요한 해쉬 결과 값 e_j를 계산한다.

$$e_j = f(ID_{cm}, X_m, M) \qquad (j = 1, \ldots, k)$$

③ ②에서 구한 해쉬 결과 값과 비밀 정보를 이용하여 다음과 같이 Y_1을 구한다.

$$Y_1 = R_1 \times \prod_{j=1}^{k} S_{1j} \bmod N \qquad (j = 1, \ldots, k)$$

④ 다음 서명자에게 (ID_{cm}, X_m, M, Y_1)을 전송한다.

단계 2. 서명자 n의 서명 발생

① 서명자 n은 서명자 n-1로부터 $(ID_{cm}, X_m, M, Y_{n-1})$을 수신한다. 다음과 같이 해쉬 결과 값 e_j를 계산한다.

$$e_j = f(ID_{cm}, X_m, M) \qquad (j = 1, \ldots, k)$$

② ①에서 구한 해쉬 결과 값과 비밀 정보를 이용하여 다음과 같이 Y_n을 구한다.

$$Y_n = R_n \times \prod_{j=1}^{k} S_{nj} \bmod N \qquad (j = 1, \ldots, k)$$

③ 다음 서명자에게 (ID_{cm}, X_m, M, Y_n)을 전송한다. 서명자 n이 마지막 서명자이면 $(ID_{cm}, (e_1, \ldots, e_k), M, Y_m)$을 검증자에게 전송한다.

단계 3. 검증자의 다중 서명 검증

① 검증자는 마지막 서명자로부터 $(ID_{cm}, (e_1, \ldots, e_k), M, Y_m)$을 수신한다. ID_{cm}과 해쉬 함수 f를 이용하여 검증에 필요한 V_n 값을 구한다.

$$V_{ij} = f(ID_i,\ j) \qquad (1 \le i \le t,\ 1 \le j \le k)$$

② 검증 값 Z_m을 구한다.

$$Z_m = Y_m^2 \times \prod_{i=1}^{m} \prod_{e_{ij}=1} V_{ij} \bmod N \qquad (j = 1,\ \ldots,\ k)$$

③ Z_m을 이용하여 메시지를 해쉬한다. 해쉬 결과 값이 수신한 $(e_1,\ \ldots,\ e_k)$와 같으면 다중 서명이 올바르게 수행되었음을 확인한다.

$(e_1,\ \ldots,\ e_k) = f(ID_{cm},\ Z_m,\ M)$: 다중 서명 메시지는 유효하다.

$(e_1,\ \ldots,\ e_k) \neq f(ID_{cm},\ Z_m,\ M)$: 다중 서명 메시지는 유효하지 않다.

Ⅲ. EDI 메시지에 적합한 디지탈 다중 서명 방법

그림 4는 EDI 메시지에 적합한 디지탈 다중 서명 방법이다. 다중 서명에 참여하는 서명자의 수가 m 명일 때 새로운 다중 서명 방법은 다음과 같다.

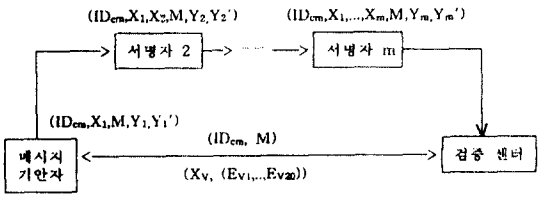

그림 4. EDI 메시지에 적합한 디지탈 다중 서명 방법
Fig. 4. The digital multisignature scheme suitable for the EDI message

3.1 신뢰할 수 있는 센터의 스마트 카드 배급

센터는 법적인 신분 확인을 거친 서명자에 대해서 센터의 공개 정보와 서명자의 비밀 정보로 구성된 스마트 카드를 다음과 같이 발급한다. 그림 5는 본 연구에서 구현한 시뮬레이터를 이용하여 서명자의 비밀 정보를 계산한 결과이다.

① 두 개의 큰 소수 p, q를 생성하고 비밀리에 보관한다.

② p와 q를 이용하여 다음과 같이 모듈라 곱셈의 법 N을 구한다.

그림 5. 센터에 의해 발급된 사용자 A의 스마트 카드
Fig. 5. The smart card of user A issued by the center

$$N = p \times q$$

③ 사용자 A의 식별 정보인 ID_A와 일방향 함수 f를 이용하여 비밀 정보 S_A를 다음과 같이 계산한다. k는 카드 내에 저장될 비밀 정보의 개수이다. 해쉬 함수 f는 충돌이 없는 일방향 함수이어야 한다. 본 연구에서의 구현에 사용된 해쉬 함수는 일반적으로 가장 많이 사용되고 있는 DES-CBC 알고리즘을 이용하였다. 따라서 사용자 A의 공개 정보인 V_A는 64 비트 크기의 해쉬 결과 값으로 구성된다. j는 작은 정수 값으로 이 값을 달리하여 공개 정보인 V_A 값을 여러 개 생성한다. 이 중 모듈라 곱셈의 법 N에 대한 평방 잉여인 것을 k 개 선택하여 비밀 정보를 구한다. 그림 5의 (2)는 k 개의 j 값, (3)은 523 비트 크기의 법 N 그리고 (4)는 k 개의 비밀 정보를 보여준다.

$$V_{Ai} = f(ID_A,\ j) \qquad (i = 1,\ \ldots,\ k)$$

$$S_{Ai}^2 = \frac{1}{V_{Ai}} \bmod N \qquad (i = 1,\ \ldots,\ k)$$

④ 스마트 카드에 센터의 공개 정보인 법 N, 일방향 함수 f와 사용자 A의 비밀 정보인 $(S_{A1},\ \ldots,\ S_{Ak})$를 기록하여 배급한다.

3.2 메시지 가안자와 검증 센터의 인증 값 생성 및 전송

메시지 기안자는 서명자의 수와 서명 순서를 정한다. 서명자의 수가 m 명일 때 서명 순서대로 각 서명자의 ID를 결합한 ID_{cm}을 만든다($ID_{cm} = ID_1 \| ID_2 \| \cdots \| ID_n \| \cdots \| ID_m$).

그림 4와 같이 메시지 기안자는 ID_{cm}과 서명 대상 메시지 M을 검증 센터로 전송하여 다중 서명 발생을 알린다. 검증 센터는 메시지 M에 대한 인증 값을 다음과 같이 생성하고 서명자들의 신분 인증을 위한 20 비트의 난수를 발생한다.

① 난수 발생기(random number generator)를 이용하여 인증 값 생성에 필요한 난수 R_I를 구한다. R_I는 0 보다 크고 모듈라 곱셈의 법 N 보다 작아야 한다.

② 해쉬 함수 h와 난수 R_I를 이용하여 다음과 같이 메시지 M을 해쉬한다. 인증 값 R_U는 해쉬 결과 값이 된다.

$$X' = R_I^2 \bmod N$$

$$R_U = h(ID_{cm}, X', M)$$

③ R_U 값을 이용하여 서명자들의 해쉬 연산에 필요한 공통 키 X_V를 다음과 같이 구한다. 검증 센터는 20 비트의 난수 (E_{V1}, ..., E_{V20})와 인증 값이 포함된 X_V를 메시지 기안자에게 전송한다.

$$X_U = R_U^2 \bmod N$$

3.3 다중 서명 발생 및 검증

검증 센터로부터 다중 서명에 필요한 인증 값이 전송되면 메시지 기안자와 서명자들은 다중 서명을 수행한다. 검증 센터는 다중 서명 값과 메시지를 검증하여 서명자들이 올바르게 순서대로 동일한 메시지를 서명했는지 확인한다.

단계 1. 메시지 기안자의 서명 발생

메시지 기안자는 인증 값이 포함된 X_V와 (E_{V1}, ..., E_{V20})를 검증 센터로부터 수신한다. 이 값들을 이용한 메시지 기안자의 서명 발생은 다음과 같다. Y_1은 메시지 기안자의 디지탈 서명 값이고, Y_1'은 작은 해쉬 결과 값 사용시 서명 위조를 막을 수 있는 신분 인증 값이다.

① 디지탈 서명 생성에 필요한 난수 R_1을 발생하고, 이 값과 인증 값이 포함된 X_V를 이용하여 다음과 같이 X_1을 구한다. R_1은 0 보다 크고 모듈라 곱셈의 법 N 보다 작아야 한다.

$$X_1 = X_U \times R_1^2 \bmod N$$

② 메시지 기안자의 해쉬 연산에 사용되는 공통 키 X_1'은 ①에서 구한 X_1 값과 같다. 서명 대상 메시지 M을 해쉬하여 결과 값의 처음 kt 비트를 다음과 같이 구한다. 제안한 방법은 k = 20, t = 1인 경우이다. 따라서 본 연구의 구현에 사용된 해쉬 함수 h는 64 비트 크기의 결과 값을 생성하는 DES-CBC 알고리즘을 사용하였다.

$$(E_{11}, \ldots, E_{1k}) = h(ID_{cm}, X_1', M)$$

③ ①에서 구한 난수 R_1, ②로부터 계산된 해쉬 결과 값 그리고 메시지 기안자의 비밀 정보 S_1을 이용하여 다음과 같이 디지탈 서명 Y_1을 생성한다.

$$Y_1 = R_1 \times \prod_{E_{1j}=1} S_{1j} \bmod N \qquad (j = 1, \ldots, k)$$

④ 작은 해쉬 결과 값 사용에 따른 서명 위조를 막기 위해서 다음과 같이 검증 센터로부터 전송된 20 비트의 난수 E_V를 이용하여 Y_1'을 구한다.

$$Y_1' = R_1 \times \prod_{E_{Vj}=1} S_{1j} \bmod N \qquad (j = 1, \ldots, k)$$

⑤ 다음 서명자에게 (ID_{cm}, X_1, (E_{V1}, ..., E_{V20}), M, Y_1, Y_1')을 전송한다.

②에서 구한 (E_{11}, ..., E_{1k})가 (E_{V1}, ..., E_{V20})와 같을 경우 서명자 및 침입자에 의한 서명 위조가 가능하다. 따라서 ①, ②의 과정을 반복하여 검증 센터가 전송한 (E_{V1}, ..., E_{V20})와 다른 해쉬 결과 값을 생성하는 난수 R_1을 구한다.

단계 2. 서명자 n의 서명 발생

서명자 n은 서명자 n-1로부터 (ID_{cm}, X_1, ..., X_{n-1}, (E_{V1}, ..., E_{V20}), M, Y_{n-1}, Y_{n-1}')을 수신한다. 수신받은 정보로부터 다음과 같이 다중 서명을 수행한다. 단계 2는 서명자 n이 최종 서명자가 될 때까지 반복한다.

① 서명자 n은 난수 R_n을 발생하여 공통 키 생성에 필요한 X_n을 구한다.

$$X_n = R_n^2 \bmod N$$

② 이전 서명자들의 공통 키 정보인 (X_1, ..., X_{n-1})

그리고 ①로부터 구한 X_n을 이용하여 다음과 같이 서명자 n의 공통 키 X_n'을 구한다.

$$X_n' = \prod_{i=1}^{m} X_i \bmod N$$

③ $(E_{n1}, \ldots, E_{nk}) = h(ID_{cm}, X_n', M)$

④ $Y_n = Y_{n-1} \times R_n \times \prod_{E_{nj}=1} S_{nj} \bmod N \quad (j = 1, \ldots, k)$

⑤ $Y_n' = Y_{n-1}' \times R_n \times \prod_{E_{vj}=1} S_{nj} \bmod N \quad (j = 1, \ldots, k)$

⑥ 다음 서명자에게 (ID_{cm}, X_1, ..., X_n, (E_{V1}, ..., E_{V20}), M, Y_n, Y_n')을 전송한다. 서명자 n이 마지막 서명자일 경우 이 값들을 검증 센터로 전송한다.

③에서 구한 (E_{n1}, ..., E_{nk})가 (E_{V1}, ..., E_{V20})와 같을 경우 ①, ②, ③의 과정을 반복하여 (E_{V1}, ..., E_{V20})와 다른 해쉬 결과 값을 생성하는 난수 R_n을 구한다.

단계 3. 검증 센터의 다중 서명 검증

검증 센터는 마지막 서명자로부터 (ID_{cm}, X_1, ..., X_m, (E_{V1}, ..., E_{V20}), M, Y_m, Y_m')을 수신한다. 3.2의 인증 값 생성에 사용된 X'와 ID_{cm}을 이용하여 다음과 같이 X_V'를 구해서 수신한 메시지가 메시지 기안자가 전송한 메시지와 동일한 메시지인지 확인한다.

① $R_U' = h(ID_{cm}, X', M)$

② $X_U' = R_U'^2 \bmod N$

③ $X_U' = X_U$: 메시지 기안자가 전송한 메시지와 같다.

$X_U' \neq X_U$: 메시지 기안자가 전송한 메시지와 다르다.

수신한 메시지의 무결성이 위의 과정으로부터 인증되면, 다음과 같이 서명 참여자들이 모두 동일한 메시지에 서명하였는 지의 여부를 검증한다.

① 서명 순서대로 결합된 서명자들의 ID 정보인 ID_{cm}으로부터 다음과 같이 각 서명자의 V 값을 계산한다. j는 각 서명자들의 V 값 생성에 필요한 k 개의 작은 정수 값이다.

$$ID_{cm} = ID_1 \| ID_2 \| \cdots \| ID_m$$

$$V_{ij} = f(ID_i, j) \qquad (1 \le i \le m,\ 1 \le j \le k)$$

② 서명자들의 난수 정보인 (X_1, ..., X_m)으로부터 각 서명자의 해쉬 연산에 사용된 (X_1', ..., X_m')을 다음과 같이 구한다.

$$X_j' = \prod_{i=1}^{j} X_i \bmod N \qquad (j = 1, \ldots, m)$$

③ ②에서 계산한 (X_1', ..., X_m')을 이용하여 다음과 같이 수신 메시지에 대한 해쉬 결과 값을 구한다. 침입자는 자신이 선택한 메시지에 대해서 특정 서명자로부터 검증 가능한 서명을 얻을 수 없다.

$$(E_{i1}, \ldots, E_{ik}) = h(ID_{cm}, X_i', M) \qquad (i = 1, \ldots, m)$$

④ 검증식 $Z_U = R_U^2 \times Y_m^2 \times \prod_{i=1}^{m} \prod_{E_{ij}=1} V_{ij} \bmod N$을 구한다. ($j = 1, \ldots, k$)

⑤ 검증식 $Z_U' = R_U^2 \times Y_m'^2 \times \prod_{i=1}^{m} \prod_{E_{vj}=1} V_{ij} \bmod N$을 구한다. ($j = 1, \ldots, k$)

⑥ $Z_U' = Z_U$: 올바르게 서명된 값이다.

$Z_U' \neq Z_U$: 위조된 서명이다.

⑦ $(E_1, \ldots, E_k) = h(ID_{cm}, Z_U, M)$

⑧ $(E_1, \ldots, E_k) = (E_{m1}, \ldots, E_{mk})$: 메시지 M에 대한 다중 서명이다.

$(E_1, \ldots, E_k) \neq (E_{m1}, \ldots, E_{mk})$: 메시지 M에 대한 다중 서명이 아니다.

Z_U와 Z_U'는 서명자들이 안전하게 서명을 수행할 수 있도록 해 주는 검증값이다. 침입자 혹은 서명자들의 모의에 의한 조작된 메시지를 서명할 경우 이를 검증할 수 있다. 서명자들이 검증된 메시지에 대해서 부인할 경우 이에 대한 부인 봉쇄를 할 수 있다.

③에서 (E_{v1}, ..., E_{v20})와 같은 해쉬 결과 값이 있을 경우 서명 위조가 발생했으므로 다중 서명 메시지를 기각한다.

⑥에서 Z_U와 Z_U'을 비교하여 이 값이 같을 경우 침입자 및 서명자에 의한 서명 위조가 발생하지 않았으므로 ⑦, ⑧과 같이 다중 서명 메시지를 검증한다. 같지 않을 경우 서명 위조가 발생했으므로 다중 서명 메시지를 기각한다.

(E_{m1}, ..., E_{mk})는 서명자들이 발생한 난수 값으로 이루어진 공통 키 X_m'을 이용하여 구한 최종 서명자의 해쉬 결과 값이다. (E_1, ..., E_k)는 검증 센터로부터 계산된 공통 키 Z_U를 이용하여 구한 해쉬 결과 값이다. 두 값이 같을 경우 메시지 M이 서명자에 의하여 순서대로 올바르게 서명되었음을 알 수 있다.

Ⅳ. 안전성 분석

제안한 방법은 메시지의 무결성이 강조되는 기업간의 전자 거래 서식인 EDI 메시지 다중 서명을 위한 기법이다. 해쉬 함수 공격으로 변조된 메시지를 검증할 수 있으며 능동적 공격에 안전하다. 해쉬 결과 비트 수를 kt라고 했을 때, 작은 해쉬 결과 값을 사용하여 서명 위조 및 메시지 조작에 대하여 2^{-kt}의 확률로만 공격이 성공할 수 있는 안전도를 제공한다.

4.1 메시지 인증의 안전성

메시지 무결성은 해쉬 함수를 이용하여 메시지를 압축한 메시지 인증 코드에 의하여 이루어진다. 제안한 방법은 그림 4와 같이 검증 센터에서 발생한 인증 값 R_U가 포함된 X_V와 서명자들이 발생한 난수 값을 이용하여 메시지를 해쉬한다. 침입자 및 서명자의 해쉬 함수 공격에 대한 안전성은 다음과 같다.

(1) 침입자(intruder)의 메시지 조작에 대한 안전성

침입자는 컴퓨터 통신망으로부터 메시지 해쉬에 사용된 정보를 획득한다. 해쉬 함수 공격법으로 원래의 메시지 M과 같은 인증 코드를 생성하는 변조된 메시지 M′을 찾아서 다음 서명자에게 전송한다. 검증 센터가 변조된 메시지 M′을 받았을 때 제안한 방법에서의 다중 서명 검증은 다음과 같다.

① $R_U' = h(ID_{cm}, X', M')$

② 검증식 $Z_U' = R_U'^2 \times Y_m'^2 \times \prod_{i=1}^{m} \prod_{e_{ij}=1} V_{ij} \bmod N$을 구한다.

③ ②에서 구한 Z_U'를 이용하여 수신한 메세지 M′에 대한 해쉬 결과 값인 $(e_1, \ldots, e_k)$를 구한다. R_U' 값과 R_U 값은 다르기 때문에 Z_U 값은 서명자들의 공통키 X_m'과 같지 않게 된다. 따라서 $(e_1, \ldots, e_k)$는 마지막 서명자가 생성한 해쉬 결과 값인 $(E_{m1}, \cdots, E_{mk})$와 다른 값을 갖게되므로 M′이 변조된 메시지임을 알 수 있다.

$$(E_{m1}, \ldots, E_{mk}) = h(ID_{cm}, Z_U', M') \quad (1)$$

$$h(ID_{cm}, X_n, M) = h(ID_{cm}, X_n, M') \quad (1 \le n \le m) \quad (2)$$

침입자는 (식 2)를 만족하는 메시지를 만들 수 있지만 인증 값 R_U를 모르기 때문에 (식 1)의 Z_U 값을 추정할 수 없다. 따라서 침입자는 (식 1)을 만족하는 변조된 메시지 M′을 만들 수 없다. 해쉬 결과 값의 비트 수를 kt라고 했을 때 침입자는 변조한 메시지가 2^{-kt}의 확률로 성공할 수 있다는 사실만 알고 전송하게 된다.

(2) 서명자의 내부 부정에 대한 안전성

서명자들이 해쉬 함수 공격을 통하여 메시지를 변조한 후, 원래의 메시지에 대한 서명 사실을 부인하고 변조된 메시지를 서명했다고 주장할 경우다. 침입자의 메시지 변조와 마찬가지로 서명자들은 (식 2)를 만족하는 메시지 M′을 생성할 수 있지만 인증 값 R_U를 모르기 때문에 (식 1)을 만족하는 메시지를 생성할 수 없다. 따라서 제안한 방법은 메시지 내용에 대한 서명자의 내부 부정을 무인 봉쇄할 수 있다.

4.2 재전송으로부터의 안전성

그림 6과 같은 방법으로 침입자는 이전에 전송된 메시지와 다중 서명 값을 가지고 있다가 후에 검증자에게 이 값들을 재전송할 경우다.

(1) 침입자의 재전송 공격은 그림 6과 같은 방법으로 이루어진다. 메시지 기안자가 검증 센터로 보낸 ID_{cm}과 메시지 M을 재전송한다. 마지막 서명자인 서명자 m이 전송한 다중 서명 값을 검증 센터로 재전송한다.

검증 센터는 매 번 다중 서명이 발생할 때마다 20비트의 난수를 메시지 기안자에게 전송하여 서명자들의 신분 인증을 수행한다. 침입자가 이전에 획득한 메시지와 서명을 가지고 재전송 공격을 성공하기 위해서는 수신한 20 비트 난수가 이전에 전송된 20 비트 난수와 같아야 한다. 결국 침입자가 재전송 공격을 성공할 확률은 2^{-20}이 된다. 제안한 방법은 타임스탬프(time-stamp)를 이용하지 않고 재전송 공격을 막을 수 있다. 침입자에 의한 재전송 공격시 검증 센터의 다중 서명 검증은 다음과 같다.

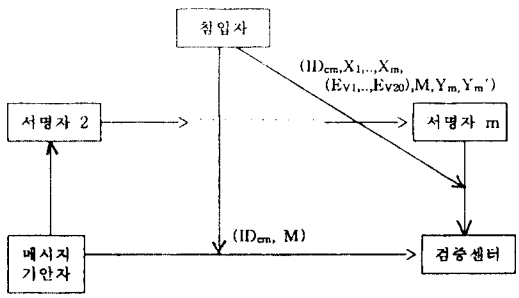

그림 6. 침입자에 의한 재전송 공격
Fig. 6. The retransmission attack by intruder

① 검증 센터가 침입자에게 전송한 난수는 (E_{V1}', E_{V20}')이고 침입자가 가지고 있던 검증 센터의 난수는 (E_{V1}, ..., E_{V20})일 경우다.

② 검증식 $Z_U' = R_U^2 \times Y'^2_m \times \prod_{i=1}^{m} \prod_{E_{Vj}=1} V_{ij} \bmod N$을 구한다.

③②로부터 구한 Z_U' 값은 (E_{V1}', ..., E_{V20}')을 이용하여 계산한 값이기 때문에 재전송 된 (X_1, ..., X_m)으로부터 계산된 X_m' 값과 같지 않게 된다. 침입자의 재전송 공격이 성공하기 위해서는 (E_{V1}', ..., E_{V20}')과 (E_{V1}, ..., E_{V20})가 같아야 한다. 따라서 침입자의 재전송 공격이 성공할 확률은 2^{-20}이 된다.

(2) 침입자가 이전에 전송된 메시지와 서명으로부터 해쉬 함수 공격법을 이용하여 변조된 메시지 M'을 만든다. 변조된 메시지와 서명을 재전송할 경우 (1)에서와 같이 신분 인증을 이룰 수 없다. 2^{-20}의 확률로 성공 하더라도 인증 값 R_U를 모르기 때문에 변조된 메시지가 올바른 메시지로 검증될지 알 수 없다. 따라서 제안한 방법은 침입자의 재전송을 통한 메시지 변조 공격에 대해서 안전하다.

4.3 작은 해쉬 결과 값 사용시 서명 위조로부터의 안전성

Fiat-Shamir 단순 서명 방식 및 Ohta-Okamoto 다중 서명 방법의 경우 침입자는 자신이 변조한 메시지에 대한 서명이 검증될 지 안 될 지를 미리 알 수 있다. 작은 해쉬 결과 값을 사용할 경우 2^{kt} 번의 시도로 서명 위조를 위한 가능한 R 값을 찾을 수 있다. kt는 해쉬 결과 값의 비트 수이다.

작은 해쉬 결과 값을 사용하는 제안한 방법에서 침입자가 서명자 n의 서명을 위조할 경우, 다음과 같은 검증을 통하여 서명 위조를 밝힐 수 있다.

①침입자는 통신 회선으로부터 서명자 n−1이 전송한 (X_{n-1}, Y_{n-1}, Y_{n-1}')을 획득한다.

②$2^{20}$ 번의 시도로 검증 가능한 (e_{n1}, ..., e_{nk})와 R_n을 구해서 다음과 같이 서명자 n의 서명을 위조한다.

$$X_n' = X_{n-1} \times R_n^2 \times \prod_{e_{nj}=1} V_{nj} \bmod N$$

$$Y_n = Y_{n-1} \times R_n \bmod N$$

$$Y_n' = Y_{n-1}' \times R_n \bmod N$$

③검증 센터는 검증식으로부터 다음과 같은 결과를 얻게 된다.

$$Z_n = Y_n^2 \times \prod_{i=1}^{n} \prod_{e_{ij}=1} V_{ij} \bmod N$$

$$= X_{n-1}' \times R_n^2 \times \prod_{e_{nj}=1} V_{nj} \bmod N$$

$$= X_n'$$

$$Z_n' = Y'^2_n \times \prod_{i=1}^{n} \prod_{E_{Vj}=1} V_{ij} \bmod N$$

$$= X_{n-1}' \times R_n^2 \times \prod_{E_{Vj}=1} V_{nj} \bmod N$$

$$\neq X_n'$$

④Z_n과 Z_n'이 같지 않으므로 서명 위조를 할 수 없음을 알 수 있다. 침입자가 검증 센터의 (E_{V1}, E_{V20})와 같은 해쉬 결과 값을 생성하는 R 값을 찾을 경우만 서명 위조가 가능하다. 제안한 방법은 검증 센터로부터 전송된 20 비트 난수와 같은 해쉬 결과 값을 생성할 수 없도록 되어 있다. 따라서 침입자는 서명자 n의 서명을 위조할 수 없다.

V. 성능 평가 및 비교

제안한 다중 서명 방법과 Ohta-Okamoto 방법에 대해서 통신 단계 수, 서명 정보 크기 및 서명 생성 시간을 기준으로 분석하였다. 본 연구에서 구현한 시뮬레이터를 이용하여 실험적 분석과 이론적 분석을 하였다.

5.1 통신 단계 수

서명자의 수가 m 명일 때, Ohta-Okamoto 방식의 통신 단계 수는 공통 키 생성에 필요한 m 번의 통신과 다중 서명 생성에 필요한 m−1 번의 통신이 요구된다. 제안한 방법은 메시지 기안자가 서명 대상 메시지와 서명자의 수를 검증 센터로 전송하는 단계와, 검증 센터가 이에 대한 인증 값을 생성하여 전송하는 2 번의 통신이 부가적으로 필요하다. 서명자들의 다중 서명 진행은 단계 별로 이전 서명자들이 구한 난수 값과 서명 값을 포함하여 새로운 값을 생성한다. 마지막 서명자에 의해서 전체 서명자의 난수 값으로 이루어지는 공통 값과 이에 대한 다중 서명 값이 구해진다. 따라서 제안한 방법의 통신 단계 수는 m+1

번이 된다. Ohta-Okamoto 방법의 2m−1 번 보다 효율적이다.

서명자의 수가 많아질수록 제안한 방법은 Ohta-Okamoto 방법에 비해서 통신 횟수가 반 정도로 줄어들게 된다. RSA 서명 방법에 기반을 둔 Itakura-Nakamura 방법과 Okamoto 방법은 통신 단계 수가 m−1 번으로 가장 좋다. RSA 서명 방법에 기반을 둔 방법과 제안한 방법의 통신 단계 수를 비교하면, 제안한 방법은 서명자의 수에 관계 없이 항상 2 번의 통신만을 더 수행하게 된다. RSA 방법은 통신 복잡도를 줄일 수 있는 장점이 있지만 연산 속도가 느리고, 상대방의 공개 키를 관리하고 유지해야 하는 어려움이 있다.

5.2 서명 정보 크기

서명 길이는 검증자가 보관해야 하는 다중 서명 정보의 크기로 계산한다. 서명자의 식별 정보, 서명 위조 여부를 판단하는 검증 값과 모듈라 곱셈에 사용되는 법의 전체 비트 수로 표현 된다. 서명자의 수가 m 명일 때 Ohta-Okamoto 방법의 서명 길이는 |ID|*m+k*t+|N| 비트가 된다. |ID|는 서명자 식별 정보의 비트 수이고, k*t는 검증 값 (E_{m1}, …, E_{mk})의 비트 수이고, |N|은 법 N의 비트 수이다. 제안한 방법은 검증 과정에 필요한 인증 값 R_U와 m 명의 서명자에 의해 생성된 해쉬 결과 값이 추가된다.

표 1의 실험 결과는 제안한 다중 서명 방법과 기존의 Ohta-Okamoto 방법에 대한 시뮬레이션 결과이다. 그림 7은 본 실험에 사용 된 EDI 메시지 화일이다. 서명 참여자의 수가 모두 10 명일 때, 각 단계 별로 서명자가 생성해 내는 디지탈 서명이 포함 된 전송 화일의 크기를 비교하였다.

```
UNA:+.?'
UNB+UNOA:1+999+111+931307:2003+001'
UNH+ME000001+ORDERS:90:1:UN:EAN004'
BGM+105+111'
DTM+:1993 04 09'
NAD+by+111+서울 성북구 안암동 5가 1++++02+136-701'
NAD+su+222+고려산업+서울시 안암동 5가 1+++02+333-111'
DTM+004:1993 04 09'
DTM+011:1993 04 15'
TOD+++:::bus'
UNS+D'
LIN+C+1++++21:10:kg+1000'
IMD+F+c01+::::banana:10000'
UNS+S'
CNT+01:10000'
UNT+14+ME000001'
UNZ+1+001'
```

그림 7. 실험용 화일 TEST1.ENV
Fig. 7. The 'TEST1.ENV' file to be tested

k=80일 때의 Ohta-Okamoto 방법의 서명 크기가 커지는 이유는 단계별로 서명자의 공개 정보인 V 값 계산에 필요한 80 개의 j 값을 전송해야하기 때문이다. k=10일 때의 Ohta-Okamoto 방법이 단계 별 서명 크기의 차가 가장 작다. Fiat-Shamir 서명 방식은 k와 t 값에 따라서 서명 크기가 결정된다. k=10, t=8인 경우의 Ohta-Okamoto 방법은 8 개의 서명 값 Y와 공통 키 X가 필요하다. 서명 참여자의 수가 매우 많을 경우(50~60 명) 효율적인 방법이다. 그러나 실제로 기업에서 EDI 메시지의 전자 결재 및 통보 전송 시 10 명 이내의 서명자가 참여하는 것을 고려할 경우 효율이 좋지 않은 방법이다. 따라서 표 1로부터 제안한 방법이 기존의 방법과 비교하여 서명 정보 크기 면에서 효율적임을 알 수 있다.

5.3 연산 속도

연산 속도는 서명 생성에 필요한 모듈라 연산 시간과 해쉬 연산 시간을 기준으로 측정한다. 표 2는 제안한 방법과 Ohta-Okamoto 방법의 모듈라 곱셈 시간에 대한 시뮬레이션 결과이다. 제안한 방법은 작은 해쉬 결과 값을 사용하기 때문에 해쉬 연산 시간이 기존의 방법과 달라지게 된다. 따라서 표 3과 같이 해쉬 연산을 포함한 전체 서명 생성 시간을 비교하였다.

해쉬 연산을 제외한 모듈라 곱셈 수는 Ohta-Okamoto 방법이 (k/2+3)*t 가 된다. 제안한 방법은 Y 값 계산을 두 번 수행하므로 (k+3)*t 가 된다. k와 t 값에 따라서 모듈라 곱셈 수가 결정되기 때문에 Ohta-Okamoto 방법이 더 많은 모듈라 곱셈을 수행한다.

표 2로부터 제안한 방법이 Ohta-Okamoto 방법보다 2배 내지 3배 정도 빠르다. k=80인 경우는 비용 및 저장 공간의 문제로 실제로 사용되지 않는다.

표 3은 해쉬 연산을 포함한 디지탈 서명 생성에 소요되는 전체 시간을 측정한 것이다. 서명 길이에 대한 실험에서와 마찬가지로 k와 t 값을 달리하여 실험하였다.

Ohta-Okamoto 방법은 메시지에 대한 해쉬 함수 공격을 고려하지 않을 경우 서명 시스템의 안전성을 위해서 최소 72 비트 이상의 해쉬 결과 값을 필요로 한다. 해쉬 함수 공격을 고려할 경우 더 큰 해쉬 결과 값이 요구된다. 현재의 과학 기술로 birthday 문제에 기반을 둔 해쉬 함수 공격법에 안전하기 위해서 128 비트 크기의 해쉬 결과 값이 요구된다[2, 3, 5, 14]. 64 비트 DES 알고리즘을 이용하여 128 비트 크기의 해쉬 결과 값을 생성하는 여러 방법이 있다[2, 5]. 이러

표 1. 서명자 수에 따른 단계별 서명 정보 크기 비교
Table 1. The comparison of signature information size with the number of signers

화일크기 (Bytes) / 서명자 수	디지탈 서명 정보 크기 비교			
	제안한 방법	Ohta-Okamoto 방법		
	k=20, t=1	k=80, t=1	k=20, t=4	k=10, t=8
1	897	964	1330	2028
2	1049	1202	1391	2059
3	1199	1432	1450	2088
4	1354	1682	1514	2120
5	1504	1912	1573	2149
6	1655	2146	1633	2179
7	1804	2372	1691	2207
8	1954	2602	1750	2236
9	2105	2836	1810	2266
10	2257	3065	1589	1651

표 2. 시뮬레이터를 이용한 모듈라 곱셈 수행 시간 비교
Table 2. Thecomparison of processing time of modular multiplication using simulator

모듈라 곱셈 수행 시간 비교 (단위 : 초)			
제안한 방법	Ohta-Okamoto 방법		
k=20, t=1	k=80, t=1	k=20, t=4	k=10, t=8
0.587912	0.945115	1.296703	1.527473

한 방법들의 공통점은 한 블럭에 대해서 최소 두 번 이상의 DES 연산이 요구되는 것이다. 따라서 한 번의 연산 수행으로 64 비트 해쉬 결과 값을 얻을 수 있는 DES-CBC 연산보다 느리게 된다.

제안한 방법은 20 비트 내지 최대 30 비트 만큼의 해쉬 결과 값으로 충분한 안전도를 제공할 수 있다. 따라서 한 번의 64 비트 DES-CBC 연산 수행으로 해쉬 결과 값을 구할 수 있다. 표 3으로부터 화일 크기가 커질 수록 제안한 방법이 Ohta-Okamoto 방법보다 연산 시간 면에서 효율적임을 알 수 있다.

5.4 디지탈 다중 서명 방식 비교

제안한 방법은 Fiat-Shamir 서명 방식에 기반을 둔 방법이고 작은 해쉬 결과 값을 사용한다. 서명 생성에 필요한 모듈라 곱셈 수가 기존의 방법보다 적다. Fiat-Shamir 서명 방식은 해쉬 결과 값의 비트 수에 따라 전송되는 정보량과 속도가 영향을 받는다. 서명 참여자의 수를 m 명이라고 했을 때 다중 서명 생성에 필요한 전체 통신 횟수는 m+1 번이다. 침입자 및 서명자의 해쉬 함수 공격에 안전하다. 표 4는 제안한 방법과 Ohta-Okamoto 방법의 안전성 및 효

표 3. 화일 크기에 따른 연산 속도 비교
Table 3. The comparison of processing speed with the file size

시간(sec) / 화일 크기(Bytes)	화일 크기에 따른 연산 속도 비교			
	제안한 방법	Ohta-Okamoto 방법		
	k=20, t=1	k=80, t=1	k=20, t=4	k=10, t=8
399	1.428572	2.252747	3.186813	4.065934
795	1.758242	3.241758	4.230760	5.109891
1587	2.637363	4.725275	5.604395	6.593406
3171	4.120880	8.021978	8.901099	9.780219
6339	7.417583	14.285710	15.219780	16.318680
12675	13.791200	27.252740	28.076920	29.065570
25347	26.648350	52.802190	53.736260	54.890110
50691	52.307690	104.175800	105.054900	105.879100
101378	103.626300	206.758200	207.912000	209.120800
202754	206.208700	411.978000	412.801800	413.791200

율성에 대한 비교 결과를 요약한 테이블이다.

표 4. 디지탈 다중 서명 방식 비교
Table 4. The comparison of digital multisignature schemes

	Ohta-Okamoto 방법	제 안 한 방 법
통신 단계 수	2m-1	m+1
서명 길이	|ID|×m+k×t+|N|	|ID|×m+k×t×m+|N|+|R_U|
연산 수	$(k/2+3)\times t^{1)}$	$(k+3)\times t^{2)}$
수동적 공격에 안전한가?	안전하다	안전하다
해쉬 함수 공식에 대한 안전도	$2^{kt/2}$ 연산 수행으로 충돌 메시지를 찾을 확률 1/2	충돌 메시지가 검증될 확률 $1/2^{kt}$
침입자의 메시지 변조	검증을 못 한다	검증 가능하다
서명자의 메시지 변조	부인봉쇄할 수 없다	부인봉쇄 가능하다

1) : k=80, t=1
2) : k=20, t=1

Ⅵ. 결 론

본 연구에서는 기업간의 EDI 메시지 통보 전송에 적합한 디지탈 다중 서명 방법을 제안하였다. 제안한 방법은 침입자 및 서명자의 해쉬 함수 공격에 안전하고 능동적 공격인 재전송과 가장을 막을 수 있다. 특히 기업간의 문서 교환시 중요시 되는 부인 봉쇄 기능을 제공할 수 있다. 서명 자들간의 결탁에 의한 특정 서명자의 메시지와 서명을 위조할 수 없다. 과학 기술 발전에 따른 고속 연산 기법의 개발에 따라서 해쉬 결과 값이 커질 필요 없이 항상 일정한 확률로 메시지 인증을 수행한다. Fiat-Shamir 서명 방식보다 작은 해쉬 결과 값을 사용하여 빠르고 안전하게 디지탈 서명을 수행할 수 있다.

제안한 방법은 통신 단계 수가 m+1 번으로 Ohta-Okamoto 방법의 2m-1번 보다 효율적이다. 작은 해쉬 결과 값을 사용하기 때문에 모듈라 곱셈의 수가 적고 해쉬 연산 시간을 줄일 수 있다. 가식 대 성능비를 고려할 경우 제안한 방법은 통신량 및 연산 시간에 대해서 Ohta-Okamoto 방법보다 좋다.

기업간의 EDI 메시지 통보 전송에 제안한 방법을 이용할 경우, 수신 거래처들은 동일한 메시지를 안전하게 전송받을 수 있다. EDI 메시지와 같이 기업간의 이해 관계에 관련된 상용 전자 문서를 여러 곳에 전송한 경우 적합한 방법이다. 빠르고 안전하게 다중 서명을 수행할 수 있고, 검증 센터의 다중 서명 검증에 의하여 서명자의 내부 부정을 부인 봉쇄할 수 있다.

참 고 문 헌

1. I.B.Damgard, "Design Principles for Hash Functions," Proceedings of Crypto'89, Lecture Lotes in Computer Science 435, pp.416-427, 1990.
2. R.R.Jueneman, S.M.Matyas and C.H.Meyer, "Message Authentication," IEEE Communications Magazine, Vol.23, No.9, pp.29-40, 1985.
3. R.R.Jueneman, "Electronic Document Authentication," IEEE Network Magazine, Vol.1, No.2, pp.17-23, 1987.
4. A.Fiat and A.Shamir, "How to prove yourself: Practical Solutions to Identification and Signature Problems," Proceedings of Crypto'86, Lecture Notes in Computer Science 263, pp. 186-194, 1987.
5. J.J.Quisquater and M.Girault, "2N-bit Hash Functions using N-bit Symmetric Block Cipher Algorithms," Proceedings of Eurocrypt'89, Lecture Notes in Computer Science 434, pp.102-109, 1990.
6. K.Itakura and K.Nakamura, "A Public-key Cryptosystem suitable for Digital Multisignature," NEC J.Res.Dev.71, 1983.
7. T.Okamoto, "A Digital Multisignature Scheme using Bijective Public-key Cryptosystems," ACM Trans. on Comp. Systems, Vol.6, No.8, pp.432-441, 1988.

8. K.Ohta and T.Okamoto, "A Digital Multisignature Scheme based on the Fiat-Shamir Scheme," Proceedings of Asiacrypt'91, pp.75-79, 1991.
9. A.Shamir, "Identity-based Cryptosystem and Signature Schemes," Proceedings of Crypto'84, Lecture Notes in Computer Science 196, pp. 47-53, 1985.
10. E.J.Humphreys, "Confidence in Your Trading Partners Certificates," Proceedings of the 7th International Conference and Exhibition on Information Security, pp.273-282, 1991.
11. G.Hutchison and C.L.Desmond, "Electronic Data Interchange," IEEE Network, Vol.1, No. 4, pp.16-20, 1987.
12. P.Landrock, "Protecting Your EDI Message," Proceedings of the 3rd International Congress of EDI Users, 1991.
13. G.Yuval, "How to Swindle Rabin," Cryptologia, Vol.3, No.3, pp.187-189, 1979.
14. D.Coppersmith, "Another Birthday Attack," Proceedings of Crypto'85, Lecture Notes in Computer Science 128, pp.14-17, 1986.
15. J.A.Cooper, Computer and Communications Security, McGraw-Hill Book Company, 1989.
16. S.Muftic, A.Patel, P.Sanders, R.Colon, J. Heijnsdijk and U.Pulkkinen, Security Architecture for Open Distributed Systems, John Wiley & Sons, 1993.
17. 강창구, 김대영, "새로운 순차 및 동시 다중서명 방식," 통신정보보호학회 논문지, 제 2 권, 제 1 호, pp.36-44, 1992.
18. 김태윤, 전자거래정보교환(EDI), 집문당, 1991.
19. 김태윤, 조광문, "EDI 보안 시스템과 디지탈 서명," 통신정보보호학회지, 제 3 권, 제 1 호, pp. 14-25, 1993.

尹 盛 鉉(Sung-Hyun Yun) 정회원
고려대학교 전산과학과(석사)
현재 : 고려대학교 대학원 전산과학과 박사과정 재학중
※주관심분야 : 컴퓨터 통신 보안, EDI시스템

金 泰 潤(Tai-Yun Kim) 정회원
고려대학교 산업공학과(학사)
미국 Wayne State University 전산과학과(석사)
미국 Auburn University 전산과학과(박사)
현재 : 고려대학교 전산과학과 교수
※주관심분야 : EDI시스템, ISDN, 이동통신, 위성통신, 컴퓨터 그래픽스