通信情報保護學會論文誌
第 10 券 第 2 號, 2000. 6

이동 통신 환경에서의 전자 상거래에 적용할 수 있는 Proxy-Signcryption 방식*

오 수 현**, 김 현 주**, 원 동 호**

Proxy-Signcryption scheme for Electronic Commerce in Mobile Environment

Soo-Hyun Oh**, Hyun-Jue Kim**, Dong-Ho Won**

요 약

C. Gamage 등은 M. Mambo의 대리 서명 방식과 Y. Zheng의 signcryption 방식을 이용하여 사용자가 상대적으로 계산 능력이 뛰어난 서버에 의존하여 암호화 및 서명을 생성할 수 있는 proxy-signcryption 방식을 제안하였다. 그러나 그들이 제안한 방식을 실제 응용에 적용할 경우 사용자가 proxy agent를 대신하여 정당한 proxy-signcryption을 생성할 수 있을 뿐만 아니라 자신이 전송한 메시지에 대해 부인하는 경우 이를 판단할 수 없으므로 proxy agent를 보호할 수 없다는 문제점이 있다. 따라서, 본 논문에서는 대리인 보호형 대리 서명 방식과 N. Asokan의 S^3(Server Supported Signatures)를 이용하여 proxy agent를 보호할 수 있고 송신자 부인 봉쇄를 제공하여 실제 응용에 적용할 수 있는 proxy-signcryption 방식을 제안하고자 한다. 또한, 본 논문에서 제안하는 proxy-signcryption 방식은 한국형 디지털 서명 표준안인 KCDSA(Korean Certificate-based Digital Signature Algorithm)를 이용한다.

ABSTRACT

C. Gamage *et al.* proposed a "proxy-signcryption scheme" that combines proxy signatures proposed by M. Mambo and signcryption scheme proposed by Y. Zheng. However, their scheme has some disadvantages that proxy agent cannot be protected because the original signer can create a valid proxy-signcryption instead of proxy agent and, if the original signer denies the fact that he sends the message, it is impossible to decide whether he is denying or not. In this paper, we present proxy-signcryption scheme for electronic commerce which combines proxy-protected proxy-signcryption and S^3(Server Surpported Sifnature) proposed by N. Asokan *et al.*. Therefore, the proposed scheme protects a proxy agent and provides non-repudiation of origin. Furthermore, the proposed proxy-signcryption scheme is based on KCDSA(Korean Certificate-based Digital Signature Algorithm).

key words : *Signcryption, Proxy-signature, Proxy-signcryption, Non-repudiation*

Ⅰ. 서 론

최근 네트워크의 발전과 함께 컴퓨터를 이용한 전자 상거래가 활성화되고 있다. 또한 이동 통신의 발전으로 인해 사용자들은 네트워크에 직접 연결된 컴퓨터를 사용하지 않고도 이동 중에 소형 노트북이나 휴대폰, PDA(Personal Digital Assistance) 등과 같은 휴대용 단말기를 이용하여 인터넷에 접속하고

* 본 연구는 한국 과학 재단의 특정기초연구(97-01-00-13-01-5) 지원 사업에 의해 수행하였습니다.
** 성균관대학교 전기전자 및 컴퓨터공학과({shoh, hjkim, dhwon}@dosan.skku.ac.kr)

상품의 주문이나 예약 등과 같은 서비스를 사용할 수 있게 되었다. 더욱이 2000년 이후부터 상용화될 차세대 이동 통신 서비스(IMT-2000)는 각 사용자가 하나의 휴대용 단말기를 이용하여 세계 어느 곳에서나 음성뿐만 아니라 고속의 데이터 서비스를 받을 수 있도록 하는 것을 목표로 하고 있다.

그러나 이와 같이 사용자가 이동 중에 휴대용 단말기를 이용하여 인터넷 전자 상거래를 이용하는 경우, 기밀성 뿐만 아니라 사용자 인증 및 부인 봉쇄 등과 같은 여러 가지 문제가 발생할 수 있게 된다. 이처럼 네트워크 상에서 발생하는 안전성과 관련된 여러 문제들을 해결하는데 가장 효과적인 방법 중에 하나가 암호 기술이며 그중 공개키 암호 방식은 인터넷과 같은 불특정 다수와의 통신에 사용할 수 있는 가장 적합한 방법이라 할 수 있다. 즉, 전송하고자 하는 메시지에 사용자의 디지털 서명을 생성한 후 수신자의 공개키로 암호화하여 전송하면 기밀성, 사용자 인증, 부인 봉쇄와 같은 문제를 해결할 수 있게된다. 그러나 디지털 서명이나 공개키 암호 방식은 모두 모듈라 멱승과 같은 많은 계산량을 요구하므로 상대적으로 적은 계산 능력을 가진 휴대용 단말기에 사용하기에는 어려운 점이 있다.

이러한 문제점을 해결하기 위해 1997년 Y. Zheng은 Crypto'97 국제학술회의에서 디지털 서명과 암호 시스템의 기능을 동시에 만족하면서 요구되는 계산량이나 전송 정보의 확장면에서는 기존의 방식에 비해 훨씬 더 효율적인 공개키 암호 방식의 새로운 패러다임인 signcryption 방식을 제안하였다.[1,2] 그 후, C. Gamage등은 Y. Zheng의 signcryption 방식과 M. Mambo가 제안한 대리 서명 방식[3,4]을 이용하여 signcryption을 생성하는데 요구되는 계산을 상대적으로 계산 능력이 뛰어난 서버에 의존함으로써 휴대용 단말기가 수행해야 할 계산량을 더욱 감소시킨 proxy-signcryption 방식을 제안하였다.[5]

그러나 C. Gamage등이 제안한 방식을 실제로 인터넷 전자 상거래 등에 적용하는 경우, 사용자가 proxy agent를 이용하여 상품을 주문한 후에 이 사실을 부인하는 경우, 실제로 주문하지 않은 것인지 부인하는 것인지를 판단할 수 없으므로 proxy agent를 보호할 수 없다는 문제점이 있다.

따라서, 본 논문에서는 적은 계산량으로 송신자 부인 봉쇄를 제공할 수 있는 N. Asokan 등이 제안한 Server supported signature[6,7]를 이용하여 단말 사용자에게 요구되는 계산량은 해쉬 함수나 관용 암호방식 등으로 제한하고 송신자 부인 봉쇄를 제공할 수 있는 대리인 보호형 proxy-signcryption 방식을 제안하고자 한다. 제안하는 방식은 공개키 암호 방식에 기반하고 있지만 실제 사용자는 해쉬 함수나 관용 암호 방식의 계산만을 수행하므로 계산 능력이 적은 단말기를 이용하는 이동 통신 환경에 적용할 수 있다.

II. 연구 배경

2.1 C. Gamage 등이 제안한 proxy-signcryption 방식

네트워크를 통해 두 사용자가 메시지를 주고 받을 때, 전송되는 메시지의 기밀성을 유지하고 당사자간에는 메시지의 출처를 확인할 수 있도록 하는 가장 효과적인 방법은 공개키 암호 방식을 사용하는 것이다. 즉, 전송할 메시지를 송신자의 비밀키로 서명한 후 수신자의 공개키로 암호화하여 전송하는 것이다. 이러한 방법을 signature-then-encryption이라 하는데, 이 방법은 메시지의 기밀성 유지나 송신자의 인증에는 적합하나 서명 생성 및 암호화 과정에 공개키 암호 방식을 사용하므로 많은 계산량이 요구된다는 단점이 있다.

이러한 문제점을 해결하기 위해 1997년 Y. Zheng은 디지털 서명과 암호 시스템의 기능을 동시에 만족하면서 요구되는 계산량이나 전송 정보의 확장면에서는 훨씬 더 효율적인 방식인 signcryption을 제안하였다.

그 후, C. Gamage 등은 [5]에서 M. Mambo가 제안한 부분 위임 대리 서명 방식과 Zheng의 signcryption 방식의 장점을 이용하여 proxy-signcryption 방식을 제안하였다.

Proxy-signcryption이란 사용자가 지정한 대리인이 자신을 대신하여 정당한 signcryption 메시지를 생성할 수 있도록 하는 방식으로 signcryption을 생성하는데 요구되는 계산을 상대적으로 계산 능력이 뛰어난 proxy agent에 의존하는 것이다. C. Gamage 등이 제안한 proxy-signcryption 방식은 다음과 같다.

[시스템 설정]

- p : 512비트 이상의 큰 소수
- q : q|p-1인 큰 소수

- g : 위수가 q인 Z_p상의 원소
- x_A : $x_A \in_R Z_q$, Alice의 비밀키
- y_A : $y_A \equiv g^{x_A} \bmod p$, Alice의 공개키
- x_p : $x_p \in_R Z_q$, proxy agent의 비밀키
- y_p : $y_p \equiv g^{x_p} \bmod p$, proxy agent의 공개키
- KH( ) : keyed 해쉬 함수
- E( )/D( ) : 관용 암호/복호 알고리즘

1) (대리 서명용 키 생성)

Alice는 $x \in_R Z_q$를 선택하고 $K \equiv g^x \bmod p$를 계산하여 대리 서명용 키 $x_{AP} \equiv x_A + x \cdot K \bmod q$를 생성하여 (x_{AP}, K)를 proxy agent에게 전송한다.

2) (대리 서명용 키의 검증)

Proxy agent는 $g^{x_{AP}} \stackrel{?}{=} y_A \cdot K^K \bmod p$를 이용하여 자신이 받은 대리 서명용 키가 정당한지 확인한다.

3) (Proxy agent에 의한 signcryption 생성)

Proxy agent는 비밀 랜덤수 $x' \in_R \{1, \ldots, q-1\}$를 선택하여 $k = y_B^{x'} \bmod p$를 계산한다. $k = k_1 || k_2$로 나누고 다음과 같이 메시지 m에 대한 signcryption을 생성한다.

$$r' = KH_{k_2}(m)$$
$$s' = x' / (r' + x_{AP}) \bmod q$$
$$c = E_{k_1}(m)$$

메시지 m에 대한 signcrypted 메시지 (c, r', s', K)를 Bob에게 전송한다.

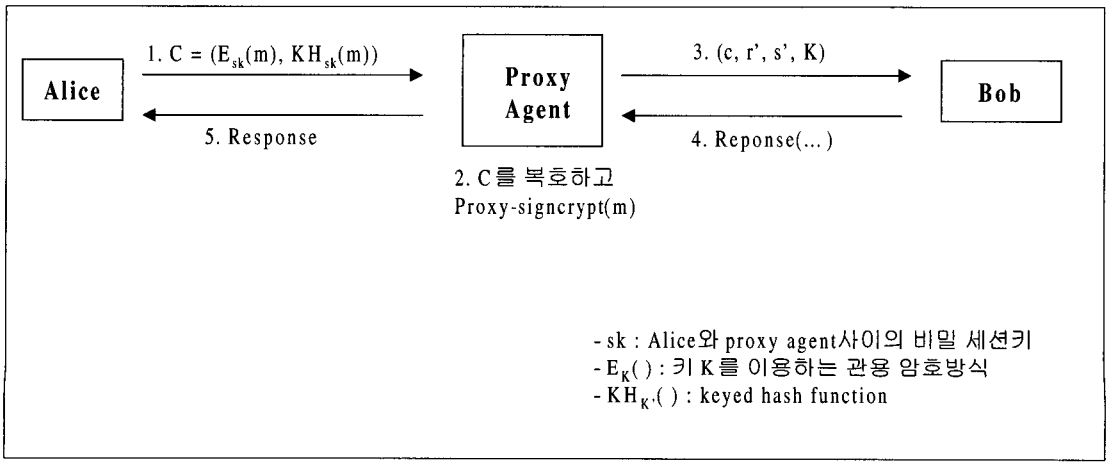
4) (proxy-signcryption 의 검증)

Bob은 $y_{AP} \equiv y_A \cdot K^K \bmod p$ 를 계산한 후, 자신의 비밀키를 이용하여 $k = (y_{AP} \cdot g^{r'})^{s' \cdot x_B} \bmod p$ 를 구한다. $k = k_1 || k_2$로 나누고 다음과 같이 메시지를 복호한다.

$$m = D_{k_1}(c)$$

단, $KH_{k_2}(m) = r'$인 경우에만 정당한 signcryption으로 받아들인다.

C. Gamage 등이 제안한 proxy-signcryption 방식을 실제 전자 상거래에 적용하는 경우를 생각해보자(그림 1 참조). 휴대용 단말기와 같이 상대적으로 계산 능력이 적은 기기를 사용하는 Alice가 인터넷을 통해 Bob으로부터 어떤 상품을 구입하려는 경우, Alice는 자신이 원하는 메시지 m에 대한 암호문 $C = (E_{sk}(m), KH_{sk}(m))$를 생성하여 사전에 지정한 proxy agent에게 전송한다.(단, E는 관용 암호 시스템, KH는 keyed 해쉬 함수이고 sk는 Alice와 proxy agent 사이에 사전에 공유한 비밀키이다.) 암호문 C를 받은 proxy agent는 이를 복호하고 앞에서 설명한 방식에 의해 proxy-signcryption을 생성하여 Bob에게 전송한다. Bob은 수신한 메시지가 Alice의 위임에 의해 proxy agent가 보낸



〔그림 1〕 C. Gamage 등이 제안한 시스템

것임을 확인하고 이에 대한 대답을 proxy agent에게 전송한다. Proxy agent는 이를 확인하고 마지막으로 Alice에게 성공적으로 상품을 주문하였음을 알려준다.

먼저, C. Gamage 등이 제안한 방식은 M. Mambo의 대리인 비보호형 대리 서명방식을 이용하므로 Alice가 proxy agent로 가장하여 정당한 proxy-signcryption을 생성할 수 있으므로 proxy agent를 보호할 수 없다는 문제점이 있다. 다시 말해, Alice가 proxy-signcryption을 생성하여 전송한 후에 proxy agent가 임의로 생성한 것이라고 주장하는 경우 제 3자는 이를 판단할 수 없게 되는 것이다.

또한, 이 방식에서는 Alice가 처음에 proxy agent에게 보내는 메시지를 두 사람 사이의 비밀키를 이용하여 암호화하여 전송하므로 나중에 Alice가 이러한 메시지를 보낸 사실을 부인하는 경우 제 3자는 Alice가 실제로 보내지 않은 것인지 아니면 부인하는 것인지를 판단할 수 없게된다. 즉, Alice가 proxy agent를 이용하여 상품을 주문한 후에, 자신의 요구없이 proxy agent가 임의로 생성한 메시지라고 주장하는 경우에 실제로 주문하지 않은 것인지 부인하는 것인지를 판단할 수 없게 되는 것이다. 그리고 Alice의 요구없이 proxy agent가 임의로 proxy-signcryption을 생성한 경우에도 이를 판단할 수 없게 된다.

그러므로 이러한 proxy-signcryption 방식을 인터넷 전자 상거래에 실제로 적용하기 위해서는 반드시 사용자의 요구에 의해서만 proxy-signcryption이 생성되어야 하고 사용자가 물건을 주문한 후에 그 사실을 부인하지 못하게 하는 부인 봉쇄가 필수적으로 제공되어야 한다. 따라서 본 논문에서는 M. Mambo의 대리인 보호형 대리 서명 방식과 N. Asokan 등이 제안한 S^3를 이용하여 앞에서 언급한 문제들을 해결할 수 있는 새로운 방식을 제안하고자 한다.

2.2 N. Asokan 등이 제안한 S^3 (Server Supported Signatures) 방식

실제 생활에서 이루어지는 거래를 네트워크 상에서 실현하는데 있어 송신자가 자신이 보낸 메시지에 대해 부인할 수 없고 수신자는 이를 수신한 사실에 대해 부인할 수 없도록 하는 부인 봉쇄는 필수적으로 제공되어야 하는 서비스 중에 하나이다. 이러한 부인 봉쇄는 대칭키 암호 방식이나 공개키 암호 방식을 이용하여 구현할 수 있다.[8]

대칭키 암호 방식을 이용하는 부인 봉쇄 기술은 신뢰 기관과 메시지 인증 코드(MAC)를 이용하며, 요구되는 계산량은 비교적 적은 편이지만 신뢰 기관을 절대적으로 신뢰해야 한다는 단점이 있다. 반면에, 공개키 암호 방식을 이용하는 부인 봉쇄 기술은 디지털 서명 방식을 이용하므로 신뢰 기관은 필요하지 않으나 요구되는 계산량이 비교적 크다는 단점이 있다. N. Asokan 등은 이러한 두 가지 방식의 장점을 결합하여 각 사용자들에게 요구되는 계산량은 공개키 암호 방식을 이용하는 경우보다 작고 센터의 부정을 검출할 있는 S^3 방식을 제안하였다. 본 절에서는 N. Asokan 등이 제안한 방식 중 송신자 부인 봉쇄(NRO, non-repudiation of origin)를 제공하는 방식에 대해 간단히 언급하고자 한다.

[시스템 설정]

- $h(\ )$: 일방향 해쉬 함수
- K_O : 사용자 O의 비밀키
- K_O^i : 사용자 O의 (n-i)번째 공개키,
  $K_O^0 = K_O$, $K_O^i = h^i(K_O) = h(K_O^{i-1})$
- PK_O : 사용자 O의 루트 공개키,
  $PK_O = K_O^n$
- $Sig_S(\ )$: 서명 서버 S의 디지털 서명

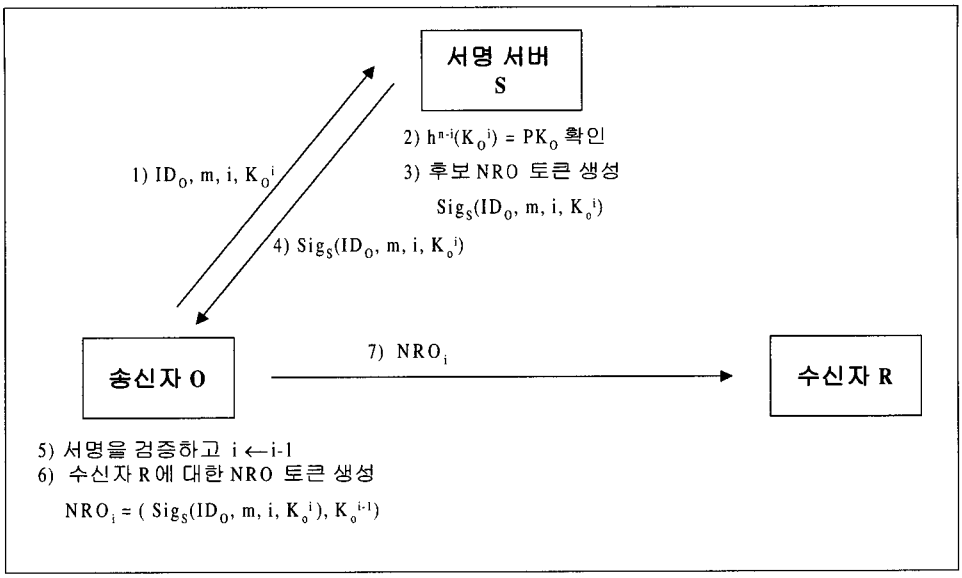
[송신자 부인 봉쇄(NRO) 토큰 생성 과정](그림 2 참조)

1) 송신자 O는 (ID_O, m, i)와 자신의 현재의 공개키 K_O^i를 서명 서버 S에게 전송한다. 이때 서명 서버에게 메시지 m을 드러내지 않고자 할 경우에는 $h(m)$을 전송한다.
2) 서명 서버 S는 사용자 O의 루트 공개키를 이용하여 다음과 같이 수신한 공개키를 검증한다.

$$h^{n-i}(K_O^i) \stackrel{?}{=} PK_O$$

3) 서명 서버 S는 K_O^i를 포함한 메시지에 대한 NRO 토큰을 생성한 적이 있는지를 검사하고 없는 경우, 다음과 같이 메시지 m에 대한 디지털 서명을 생성한다. 이러한 메시지를 후보 NRO 토큰(candidate NRO token)이라 한다.

$$Sig_S(ID_O, m, i, K_O^i)$$



〔그림 2〕 N. Asokan 등이 제안한 송신자 부인 봉쇄 프로토콜

4) 서명 서버 S는 K_O^i가 사용되었음을 기록하고 송신자 O에게 후보 NRO 토큰을 전송한다.
5) 송신자 O는 서명 서버의 서명을 검증하고 K_O^i를 사용하였으므로 저장된 i를 i-1로 바꾼다.
6) 송신자 O는 후보 NRO 토큰과 i-1번째 토큰 공개키를 이용하여 다음과 같이 실제 NRO 토큰을 생성하여 수신자 R에게 전송한다.

$$NRO_i = (Sig_S(ID_O, m, i, K_O^i), K_O^{i-1})$$

실제 NRO_i 토큰을 생성하기 위해서는 서버의 서명과 함께 다음에 사용할 공개키 K_O^{i-1}이 필요하고 공개키를 생성하는데 사용한 함수 h()의 일 방향성에 의해 이러한 NRO 토큰은 송신자만이 생성할 수 있게된다.

또한, N. Asokan 등이 제안한 방식에서 송신자에게 요구되는 계산량은 해쉬 함수와 서명의 검증 뿐이다. 이때, 송신자에게 요구되는 계산량을 최소화하기 위해 서명의 검증은 계산상 효율적이어야 하므로, 공개키를 3으로 하는 RSA 서명 방식 등이 적합하다.

Ⅲ. 제안하는 방식

3.1 대리인 보호형 proxy-signcryption 방식

C. Gamage 등이 제안한 proxy-signcryption 방식은 대리인 비 보호형 대리 서명 방식을 이용하므로 원 서명자인 Alice가 대리 서명자인 proxy agent와 동일한 proxy-signcryption을 생성할 수 있다. 따라서 Alice가 임의의 메시지에 대해 proxy-signcryption 생성한 후 proxy agent가 생성한 것이라고 주장하는 경우에 제3자는 이를 판단할 수 없다는 문제점이 있다. 따라서 본 논문에서는 M. Mambo가 제안한 대리 서명 방식 중 대리인 보호형 대리 서명방식을 이용하여 proxy agent를 보호할 수 있는 대리인 보호형 proxy-signcryption 방식을 제안한다. 그리고 제안하는 signcryption 방식은 한국형 디지털 서명 표준안인 KCDSA[9]를 이용한다.

시스템 설정은 2장과 동일하다.

1) (대리 서명용 키 생성) Alice는 $x \in_R Z_q$를 선택하고 $K \equiv g^x \bmod p$를 계산하여 대리 서명용 키 $x_{AP} \equiv x_A + x \cdot K \bmod p-1$를 생성하여 비밀리에 (x_{AP}, K)를 proxy agent에게 전송한다.
2) (대리 서명용 키의 검증 및 변환) Proxy agent는 $g^{x_{AP}} \equiv y_A \cdot K^K \bmod p$를 이용하여 자신이 받은 대리 서명용 키가 정당한지 확인한 후, 다음과 같이 alternative proxy x_{AP}'를 생성한다.

$$x_{AP}' \equiv x_{AP} + x_P y_P \bmod q$$

3) (Proxy agent에 의한 proxy-signcryption 생성) Proxy agent는 비밀 랜덤수 $x' \in R \{1, \ldots, q-1\}$를 선택하여 $k \equiv y_B^{x'} \bmod p$를 계산한다.

$k=k_1||k_2$로 나누고 다음과 같이 메시지 m에 대한 signcryption을 생성한다.

$$r' = KH_{k_2}(m)$$
$$s' \equiv (x_{AP}')^{-1} \cdot (x' - r') \mod q$$
$$c = E_{k_1}(m)$$

메시지 m에 대한 signcrypted 메시지 (c, r', s', K)를 Bob에게 전송한다.

4)(proxy-signcryption의 검증) Bob은 $y_{AP}' \equiv y_A \cdot y_P^{y_P} \cdot K^K \mod p$를 계산하고 비밀키를 이용하여 $k \equiv ((y_{AP}')^{s'} \cdot g^{r'})^{x_B} \mod p$를 구한다. $k=k_1||k_2$로 나누고 다음과 같이 메시지를 복호한다.

$$m = D_{k_1}(c)$$

단, $KH_{k2}(m)=r'$인 경우에만 정당한 signcryption으로 받아들인다.

제안하는 방식에서 alternative proxy x_{AP}'를 생성하는데 proxy agent의 비밀키 x_P가 사용되므로 이 값을 모르는 Alice는 정당한 proxy-signcryption을 생성할 수 없게 된다. 그리고 proxy-signcryption의 수신자인 Bob이 y_{AP}'를 계산하는 과정에 Alice의 공개키 y_A와 proxy agent의 공개키 y_P를 동시에 사용하므로 Alice의 위임에 의해 proxy agent가 생성한 proxy-signcryption임을 확인할 수 있게 된다.

3.2 제안하는 방식 : N. Asokan의 S^3와 대리인 보호형 proxy-signcryption 이용

본 절에서는 앞에서 설명한 대리인 보호형 proxy-signcryption 방식을 실제로 인터넷 전자 상거래에 적용할 수 있는 방식을 제안하고자 한다. 제안하는 방식은 C. Gamage 등이 제안한 방식이 갖는 문제점을 해결하기 위하여 대리인 보호형 대리 서명 방식을 이용하고 Alice가 전송한 메시지에 대해 부인 봉쇄를 제공하기 위해 N. Asokan등이 제안한 S^3를 이용한다. 제안하는 방식은 다음과 같이 4개의 구성요소로 이루어진다.

- Alice : 계산 능력이 적은 단말기를 이용하여 인터넷을 통해 전자 상거래를 하려는 사용자
- 서명 서버 S : 송신자 부인 봉쇄를 제공하기 위해 후보 NRO 토큰을 생성하는 서버
- Proxy agent : 사용자의 위임에 의해 proxy-signcryption을 생성하는 대리인
- Bob : 인터넷 쇼핑몰

Alice가 인터넷을 통해 proxy agent를 이용하여 Bob으로부터 물건을 구입하는 과정은 서명 서버로부터 부인 봉쇄 토큰을 받는 단계와 이를 이용하여 실제로 물건을 주문하는 단계로 나눌 수 있다. 또한, 제안하는 방식에서는 두 개의 공개키(부인 봉쇄 토큰을 발급받기 위한 공개키와 proxy-signcryption을 검증하기 위한 공개키)가 사용된다.

먼저, Alice는 $K_A \in Z_q$를 랜덤하게 선택하고 다음과 같이 n개의 해쉬 체인을 생성한다.

$$K_A^0 = K_A,\ K_A^1 = h(K_A), \cdots, K_A^i = h^i(K_A) = h(K_A^{i-1}), \cdots,\ K_A^n = h^n(K_A)$$

Alice는 부인 봉쇄 토큰을 발급받을 때 사용할 공개키 $PK_A = K_A^n$를 생성하여 인증기관으로부터 인증서를 발급 받는다. 그리고 비밀키 $x_A \in Z_q$를 선택하고 proxy-signcryption을 검증하는데 사용할 공개키 $y_A \equiv g^{x_A} \mod p$를 계산하여 인증기관으로부터 인증서를 발급받는다.

그리고 나서, Alice는 자신의 proxy agent를 선택하고 Ⅲ.1절에서 설명한 바와 같이 대리인 보호형 proxy-signcryption을 생성할 수 있는 대리 서명용 키를 proxy agent에게 발급한다.

[시스템 설정]
- p : 512비트 이상의 큰 소수
- q : q|p-1인 큰 소수
- g : 위수가 q인 Z_p상의 원소
- x_A : $x_A \in_R Z_q$, Alice의 비밀키
- y_A : $y_A \equiv g^{x_A} \mod p$, proxy-signcryption 검증에 사용하는 Alice의 공개키
- PK_A : 부인 봉쇄 토큰의 생성/검증에 사용하는 Alice의 루트 공개키, $PK_A = K_A^n = h^n(K_A)$
- K_A^i : Alice의 (n-i)번째 공개키, $K_A^0 = K_A,\ K_A^i = h^i(K_A) = h(K_A^{i-1})$

- x_p : $x_p \in_R Z_q$, proxy agent의 비밀키
- y_p : $y_p \equiv g^{x_p} \bmod p$, proxy agent의 공개키
- h( ) : 충돌 회피성 일방향 해쉬 함수
- KH( ) : keyed 해쉬 함수
- E( )/D( ) : 관용 암호/복호 알고리즘
- sk : Alice와 proxy agent 사이에 공유한 비밀키
- Sig_S( ) : 서명 서버 S의 디지털 서명

[제안하는 프로토콜]

1) Alice는 먼저 Bob에게 보내고자 하는 메시지 m_i를 생성하고 (ID_A, $h(m_i)$, i, K_A^i)를 서명 서버 S에게 전송한다.
2) 서명 서버 S는 다음 식을 이용하여 Alice의 현재의 공개키 K_A^i를 검증한다.

$$h^{n-i}(K_A^i) \stackrel{?}{=} PK_A$$

3) 서명 서버 S는 디지털 서명을 생성하여 다음과 같이 후보 NRO 토큰을 생성한다.

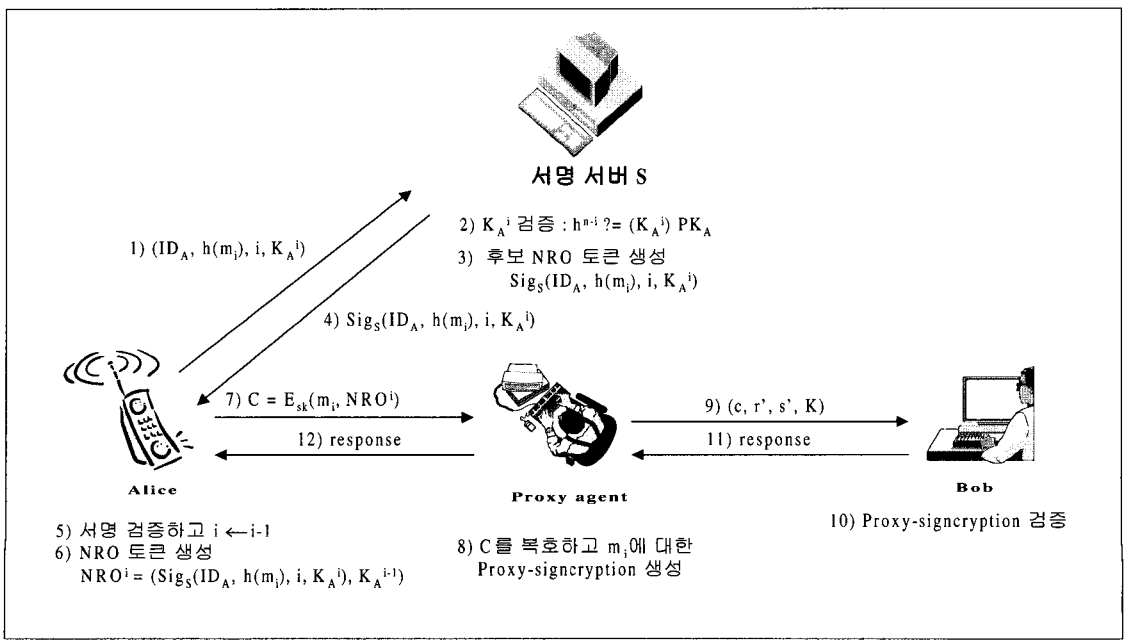
$$Sig_S(ID_A, h(m_i), i, K_A^i)$$

4) 서명 서버 S는 생성한 후보 NRO 토큰을 Alice에게 전송한다.
5) Alice는 서명을 검증하고 i를 i-1로 변경한 후, 다음과 같이 실제 NRO 토큰을 생성한다.

$$NRO^i = (Sig_S(ID_A, h(m_i), i, K_A^i), K_A^{i-1})$$

6) Alice는 Bob에게 전송할 메시지와 NRO 토큰을 proxy agent와 사전에 공유한 비밀키를 이용하여 암호화하여 $C = E_{sk}(m_i, NRO^i)$를 proxy agent에게 전송한다.
7) Proxy agent는 암호문 C를 복호하고 NRO 토큰을 저장한다. 그리고 사전에 위임받은 대리 서명용 키를 이용하여 m_i에 대한 대리인 보호형 proxy-signcryption을 생성한다.
8) Proxy agent는 생성한 proxy-signcryption (c, r′, s′, K)를 Bob에게 전송한다.
9) Bob은 proxy-signcryption을 검증하여 Alice의 요구에 의해 proxy agent가 보낸 메시지임을 확인한다.
10) Bob은 9)에 대한 응답을 proxy agent에게 전송한다.
11) Proxy agent는 메시지가 성공적으로 전달되었음을 Alice에게 알려준다.

제안하는 방식에서 서명 서버가 후보 NRO 토큰을



[그림 3] 전자 상거래에 적용할 수 있는 제안하는 proxy-signcryption 방식

생성할 때에는 서명의 검증 과정이 서명의 생성 과정보다 훨씬 더 적은 계산량을 요구하는 서명 방식(예를 들어, 공개키 e=3을 사용하는 RSA 서명 방식)을 이용하여 Alice가 수행해야 하는 계산량을 감소시킬 수 있다. 그러므로 Alice는 관용 암호 방식이나 해쉬 함수와 같이 상대적으로 적은 양의 계산만을 수행하게 되고 디지털 서명의 생성과 같이 많은 계산량을 요구하는 부분은 서명 서버와 proxy agent에 의해 수행된다.

그리고 제안하는 방식은 대리인 보호형 proxy-signcryption을 이용하여 Alice도 정당한 proxy-signcryption 메시지를 생성할 수 없으므로 Alice가 proxy-signcryption을 생성한 후에 proxy agent가 임의로 생성한 것이라고 주장하는 것이 불가능하게 된다. 또한, Alice의 요구없이 proxy agent가 임의로 proxy-signcryption을 생성하는 경우 그 메시지에 대한 NRO 토큰을 생성할 수 없으므로 반드시 Alice의 요구에 의해서만 정당한 proxy-signcryption을 생성할 수 있게된다. 또한, Alice가 proxy agent를 통해 상품을 주문한 후에 이 사실을 부인하는 경우 proxy agent는 메시지와 함께 수신한 NRO 토큰을 이용하여 Alice가 부인하고 있다는 사실을 증명할 수 있게되어 송신자 부인 봉쇄를 제공할 수 있다.

Ⅳ. 제안하는 방식의 특징

4.1 계산량 비교

제안하는 방식을 실제 전자 상거래 등에 적용하는 경우 가장 큰 장점은 서명 후 암호화하는 방식에 비해 단말 사용자에게 요구되는 계산량을 감소시킬 수 있다는 것이다. Alice가 proxy agent를 이용하지 않고 Bob으로부터 물건을 구입하려는 경우, 메시지의 기밀성, 인증 및 부인 봉쇄를 보장하기 위해서는 전송하는 메시지에 디지털 서명을 생성한 후 암호화하여 전송해야 한다. 이때, 가장 효율적인 방법으로는 메시지 m에 대해 Alice가 디지털 서명을 생성한 후, 메시지와 서명문을 관용 암호 방식을 이용하여 암호화하고 이때 사용한 키를 Bob의 공개키로 암호화하여 전송하는 것이다.

이러한 서명 후 암호화 방식과 제안하는 proxy-signcryption을 이용하는 방식에서 Alice에게 요구되는 계산량을 비교하면 〔표 1〕과 같다.

서명 후 암호화 방식 중 서명과 암호화에 모두 RSA 방식을 이용하는 경우, Alice가 Bob에게 전송하는 메시지는 $C = (E_K(m)\|Sig_A(m)\|K^{e_B})$이므로 서명 생성과 비밀키 암호화(K^{e_B}) 에 각각 한번의 모듈라 멱승이 필요하고 메시지 암호화에 1번의 관용 암호 방식의 계산이 필요하다. 그리고 이산대수에 기반한 방식 중 가장 효율적인 Schnorr 서명 방식[10]과 ElGamal 암호 방식[11]을 사용하는 경우에는 Alice가 Bob에게 전송하는 메시지는 $C=(E_K(m)\| Sig_A(m)\|g^x)$(단, $K=y_B^x$)가 된다. 이러한 경우에는, K와 g^x의 계산 그리고 Schnorr 방식의 서명 생성에 각각 1번씩 모듈라 멱승이 필요하고 메시지를 암호화하기 위해 1번의 관용 암호 방식을 계산해야 한다.

반면에, 제안하는 방식에서는 수신자 부인 봉쇄 기능을 제공하기 위해 NRO 토큰을 생성하는데 1번의 서명 검증이 필요하고 proxy-signcryption을 생성하기 위해 1번의 관용 암호 방식을 계산해야 한다. 또한, NRO 토큰을 생성하는데 센터의 공개키를 3으로 하는 RSA 서명 방식을 이용하면 서명의 검증 또한 모듈라 3승과 같은 적은 계산만을 요구하도록 할 수 있다. 따라서 제안하는 방식은 단말 사용자에게 후보 NRO 토큰의 검증과 한번의 관용 암호 방식만을 요구하면서 전송하는 메시지에 대한 기밀성, 인증 및 부인봉쇄를 제공할 수 있게된다.

4.2 송신자 부인 봉쇄 기능

C. Gamage 등이 제안한 방식은 Alice가 proxy agent에게 메시지를 전송한 사실을 부인하는 경우

〔표 1〕 서명 후 암호화 방식과 제안하는 방식의 계산량 비교

	서명 후 암호화 방식		제안하는 방식
	RSA 방식 이용	Schnorr 서명 방식과 ElGamal 암호 방식 이용	
Alice에게 요구되는 계산량	Exp : 2번 Enc : 1번	Exp : 3번 Enc : 1번	NRO 토큰 생성 : 서명 검증 1번 (모듈라 3승) proxy-signcryption 생성 : Enc 1번

단, Exp는 모듈라 멱승이고, Enc는 관용 암호 방식을 의미한다.

진위 여부를 판단할 수 없고 proxy agent가 Alice의 요구없이 임의의 메시지에 대한 proxy-signcryption을 생성하는 것이 가능하므로 실제 전자 상거래에 적용하기는 어렵다. 그러나 제안하는 방식은 Alice가 proxy agent에게 메시지를 전송하기 전에 서명 서버로부터 해당 메시지에 대한 NRO 토큰을 발급 받으므로 이러한 문제를 해결할 수 있다.

제안하는 시스템에서 Alice가 NRO 토큰 $NRO^k=(Sig_S(ID_A, h(m_k'), k, K_A^k), K_A^{k-1})$에 대해 메시지 m_k'가 자신이 생성한 것이 아니라고 하는 경우, 재판관은 Alice가 실제로 생성하지 않은 것인지 아니면 부인하는 것인지를 판단할 수 있게 된다.

먼저, 토큰 공개키 K_A^{k-1}이 아직 공개되지 않은 경우에는 함수 h()의 일 방향성에 의해 Alice외의 다른 사람은 아무도 K_A^{k-1}을 계산할 수 없으므로 이는 Alice가 부인하는 것이 된다. 그러나 K_A^{k-1}이 이미 공개된 경우, Alice는 같은 공개키 K_A^k에 대응하는 다른 NRO 토큰 $NRO^k=(Sig_S(ID_A, h(m_k), k, K_A^k), K_A^{k-1})$을 재판관에게 제출하여 m_k'가 자신이 생성한 메시지가 아님을 증명할 수 있다. 제안하는 방식에서는 반드시 Alice가 자신이 생성한 메시지 m_k와 공개키 K_A^k가 포함된 서버의 서명을 받은 후에만 K_A^{k-1}을 공개하게 된다.

따라서 실제로 m_k'가 Alice가 생성한 메시지가 아니라면 Alice는 동일한 공개키 K_A^k에 대응하는 다른 NRO 토큰 $NRO^k=(Sig_S(ID_A, h(m_k), k, K_A^k), K_A^{k-1})$을 가지고 있게된다. 그러므로 재판관에게 이러한 NRO 토큰을 제출함으로써 NRO 토큰이 위조되었음을 증명할 수 있게된다. 같은 공개키에 대응하는 서로 다른 두 개의 NRO 토큰이 존재한다는 것을 재판관에게 증명하지 못하는 경우에는 Alice가 부인하는 것이 된다.

따라서 제안하는 방식에서는 Alice가 proxy agent에게 메시지를 전송한 사실을 부인하는 경우 이를 확인할 수 있고 Alice의 요구없이 proxy agent가 임의로 proxy-signcryption을 생성하는 경우에는 해당 메시지에 대한 NRO 토큰을 생성할 수 없으므로 Alice의 요구가 있는 경우에만 proxy-signcryption을 생성할 수 있게된다.

4.3 서명 서버의 부정 검출 가능

제안하는 방식에서는 송신자 부인봉쇄를 제공하기 위해 사용자에게 후보 NRO 토큰을 생성해주는 서명 서버를 사용한다. 따라서 공개키 암호 방식에 기반한 부인봉쇄 방식에 비해 사용자에게 요구되는 계산량을 감소시킬 수 있다. 또한 본 논문에서 사용하는 서명 서버는 부정이 있는 경우 이를 검출할 수 있는 verifiable third party[6]이므로 서버를 절대적으로 신뢰해야 하는 관용 암호 방식에 기반한 부인봉쇄 방식에 비해 실용적이라는 장점이 있다.

제안하는 방식에서 Alice의 메시지 m_i에 대한 NRO 토큰을 생성하기 위해서는 서버의 서명 $Sig_S(ID_A, h(m_i), i, K_A^i)$와 Alice의 공개키 K_A^{i-1}이 필요하다. 앞에서 설명한 바와 같이, Alice는 서명 서버로부터 K_A^i이 포함된 서명을 받은 후에만 K_A^{i-1}을 공개하므로 만약 서버가 다른 메시지 m_i'에 대해 서명을 생성하여 $NRO_i'=(Sig_S(ID_A, h(m_i'), i, K_A^i), K_A^{i-1})$와 같이 위조된 NRO 토큰을 생성하는 경우에 Alice는 동일한 공개키 K_A^i를 포함한 다른 NRO 토큰을 이용하여 서버의 부정을 증명할 수 있다.

4.4 토큰을 저장하는데 요구되는 메모리를 감소시키는 방법

앞에서 설명한 바와 같이, 제안하는 시스템에서 위조된 NRO 토큰에 의해 분쟁이 생겼을 때 Alice는 같은 토큰 공개키를 포함하는 또 다른 NRO 토큰을 제시함으로써 자신이 생성한 메시지가 아님을 증명할 수 있다. 따라서 Alice는 문제가 생길 경우를 위해 서명 서버로부터 발급받은 모든 NRO 토큰을 저장하고 있어야 하는데 이것은 휴대폰이나 PDA와 같은 비교적 메모리 크기가 작은 휴대용 단말기에는 적절하지 않을 수 있다.

이러한 메모리 문제를 해결하기 위해 S. Haver 등이 [12]에서 제안한 방식을 이용하여 서명 서버가 이전에 발급한 모든 토큰들에 대한 정보가 포함된 해쉬 값을 NRO 토큰에 포함시키면 Alice가 저장해야 할 정보의 크기를 감소시킬 수 있다. 이러한 NRO 토큰은 다음과 같다.

$$NRO_i=(Sig_S(ID_A, m_i, i, K_A^i, H^i), K_A^{i-1})$$

여기서 해쉬 값은 $H^i=f(H^{i+1}, NRO^{i+1})$와 같이 계산한다(단, f는 충돌 회피성 일방향 해쉬 함수이고 $H^n=PK_A$에 대한 인증서가 된다). 이때, Alice는 자신의 단말기에 서명 서버 S로부터 받은 모든 NRO 토큰을 저장하는 것이 아니라 마지막으로 발급받은 NRO 토큰과 해쉬 값 H^i만을 저장하면 된다.

제안하는 방식에서 NRO 토큰은 서명 서버의 서명으로 만들어진 후보 NRO 토큰과 다음에 사용할 공개키 K_A^{i-1}로 이루어져 있으므로 하나의 NRO 토큰의 크기는 512비트+160비트가 된다(단, 서명 서버가 후보 NRO 토큰을 생성하는데 모듈라 n의 크기가 512 비트인 RSA 서명 방식을 이용하고 사용자가 공개키를 만드는데 출력이 160 비트인 해쉬 함수를 사용한다고 가정한 경우).

그러나 앞에서 언급한 S. Haver 등이 제안한 chaining 방식을 적용하면, 사용자는 과거에 발급받은 모든 토큰들을 저장하는 것이 아니라 가장 마지막으로 발급받은 토큰 하나와 해쉬값 하나만을 저장하면 되므로 n개의 메시지에 대해 사용자가 저장해야 하는 정보의 크기를 512비트+160비트+160비트로 줄일 수 있다. 대신, 서버 S는 나중에 분쟁이 생기는 경우 이를 해결하기 위해 발급한 모든 토큰을 저장해두어야 한다.

V. 결 론

네트워크와 휴대용 컴퓨터가 급속도로 발전함에 따라 사용자들이 이동 중에 휴대용 단말기를 이용하여 인터넷 전자 상거래를 이용하는 경우가 많아지고 있다. 이러한 네트워크를 통한 전자 상거래에서 기밀성, 인증 및 부인 봉쇄와 같은 문제를 해결하기 위해 공개키 암호 방식이 가장 많이 사용되고 있지만 암호화나 서명 생성에 많은 계산량이 요구되므로 계산 능력이 적은 단말기를 사용하는 응용에는 적합하지 않다는 단점이 있다.

이러한 문제를 해결하기 위해 C. Gamage 등은 상대적으로 계산 능력이 뛰어난 서버에 의존하여 암호화 및 서명을 생성할 수 있는 proxy-signcryption 방식을 제안하였다. 그러나 이들이 제안하는 방식은 사용자가 proxy agent를 대신하여 정당한 proxy-signcryption을 생성할 수 있을 뿐 아니라 자신이 전송한 메시지에 대해 부인하는 경우 이를 판단할 수 없으므로 proxy agent를 보호할 수 없다는 문제점이 있다.

따라서, 본 논문에서는 대리인 보호형 대리 서명 방식과 N. Asokan의 Server Supported Signatures를 이용하여 원 서명자인 Alice도 proxy agent를 대신하여 정당한 proxy-signcryption을 생성할 수 없고 Alice는 자신이 전송한 메시지에 대해 후에 그 사실을 부인할 수 없으며, proxy agent는 사용자의 요구가 있는 경우에만 proxy-signcryption 을 생성할 수 있는 새로운 proxy-signcryption 방식을 제안하였다.

제안하는 방식에서는 기밀성, 인증 및 송신자 부인 봉쇄를 제공하기 위해 단말 사용자는 해쉬 함수나 관용 암호 방식과 같이 적은 양의 계산만을 수행하고 디지털 서명의 생성과 같이 많은 양의 계산을 필요로 하는 과정은 상대적으로 계산 능력이 뛰어난 서버와 proxy agent에 의존한다.

따라서, 적은 계산 능력을 가진 이동 통신용 단말기를 이용하는 인터넷 전자 상거래 등에 적용할 수 있을 것으로 기대한다.

참 고 문 헌

[1] Y. Zheng, "Digital Signcryption or How to Achieve Cost(Signature & Encryption) << Cost(Signature) + Cost(Encryption)", *Advances in Cryptology - CRYPTO'97*, Springer-verlag, LNCS 1294, pp. 165-179, 1997.

[2] Y. Zheng, "Signcryption and Its Applications in Efficient Public Key Solutions", *Proceedings of 1997 Information Security Workshop (ISW'97)*, LNCS 1397, pp.291-312, Springer-verlag, 1998.

[3] M. Mambo, K. Usuda and E. Okamoto, "Proxy signatures : Delegation of the power to sign message", *IEICE Transaction on Fundamentals*, E79-A(9):1338-1354, 1996

[4] M. Mambo, K. Usuda and E. Okamoto, "Proxy signatures for delegating signing operation", *Proc. Third ACM Conference on Computer and Communications Security*, pp.48-57, 1996

[5] C. Gamage, J. Leiwo and Y. Zheng, "An Efficient scheme for Secure Message Transmission using Proxy-Signcryption", *Proceeding of the Twenty Second Australasian Computer Science Conference*, Auckland, New Zealand. January 18-21, 1999

[6] N. Asokan, G. Tsudik and M. Waidner, "Server-Supported Signatures", *Proc. of the Fourth European Symposium on Research in Computer Security (ESORICS)*, LNCS 1146, pp. 131-143, Springer-Verlag, September 1996.

[7] N. Asokan, G. Tsudik and M. Waidner, "Server-

Supported Signatures", *Journal of Computer Security,* November 1997.

[8] J. Zhou and D. Gollmann, "Observation on non-repudiation", *Advances in Cryptology - ASIACRYPT '96*, Springerverlag, LNCS 1163, pp. 133-144, 1996.

[9] C. H. Lim and P. J. Lee, "A Study on the Proposed Korean Digital Signature Algorithm", *Advances in Cryptology - Asiacrypt'98*, LNCS 1514, Springer-Verlag, pp.175-186, 1998

[10] C.P. Schnorr, "Efficient identification and signatures for smart cards", *In Advances in Cryptology - CRYPTO '89*, LNCS 435, Springer-Verlag, pp. 387-398, 1989

[11] T. ElGamal, "A Public key cryptosystem and a signature scheme based on discrete logarithms", *IEEE Transactions on Information Theory IT-31*, pp. 469-472, 1985

[12] S. Haber and W.S. Stornetta, "How to time-stamp a digital document", *Advances in Cryptology-CRYPTO'90*, pp. 437-455, Springer-Verlag, 1991

[13] R. Rivest, A. Shamir, and L. Adleman, "A method for Obtaining Digital signature and Public key Cryptosystems", *Communication of the ACM*, pp. 120-128, 1978.

[14] Y. Zheng, "Shortened digital signature, signcryption and compact and unforgeable Key agreement schemes", IEEE P1363 Standard for Public Key Cryptography : Additional Techniques.

〈著者紹介〉

오 수 현(Soo-Hyun Oh)
1974년 10월 16일생
1998년 2월 성균관대학교 정보공학과 졸업(공학사)
2000년 2월 성균관대학교 전기전자 및 컴퓨터 공학과 대학원 졸업(공학석사)
2000년 3월 ~ 현재 성균관대학교 전기전자 및 컴퓨터 공학과 박사 과정
※ URL : http://dosan.skku.ac.kr/~shoh

김 현 주 (Hyun-Jue Kim)
1973년 3월 1일생
1995년 세명대학교 수학과 (이학사)
1997년 서강대학교 수학과 대학원 졸업 (이학석사)
1999년 ~ 현재 성균관대학교 전기전자 및 컴퓨터 공학과 박사과정
※ URL : http://dosan.skku.ac.kr/~hjkim

원 동 호 (Dong-Ho Won)
성균관대학교 전자공학과 졸업(학사, 석사, 박사)
1978년 4월 ~ 1980년 3월 한국전자통신연구소 연구원
1985년 9월 ~ 1986년 8월 일본 동경공대 객원 연구원
1996년 4월 ~ 1998년 4월 정보화 추진위원회 자문위원
1982년 3월 ~ 현재 성균관대학교 전기전자 및 컴퓨터공학부 교수
1999년 ~ 현재 한국통신정보보호학회 부회장
1999년 ~ 현재 성균관대학교 전기전자 및 컴퓨터공학부 학부장
1999년 ~ 현재 성균관대학교 정보통신대학원 원장
※ URL : http://dosan.skku.ac.kr/~dhwon
※ 주관심 분야 : 암호 이론, 정보 이론