다단계 대리서명을 위한 권한위임 프로토콜 설계

김성열*

Design of a Protocol to Delegate Signing Right for Multi-level Proxy Signature

Seong-yeol Kim*

요 약

원서명자의 서명권한을 대리서명자가 수행하도록 지정하는 대리서명기법은 Mambo[1]이래로 많은 연구가 이루어졌으며, 분산네트워크, 그리드 컴퓨팅, 전자 상거래 등 많은 분야에 응용되고 있다. Araki[6]에서는 기존의 대리서명 기법을 확장하여 다단계 대리서명을 제안하였다. 그러나 이 연구결과는 보안상의 취약점이 존재하는 것으로 드러났다. 이 논문에서는 다단계 대리서명을 위한 서명권한 위임 프로토콜을 설계하였다. 이 프로토콜은 보안채널을 요구하지 않으며 권한위임 및 위임수락 부인이 불가능하고 지정된 대리서명자 이외에 제3자에 의한 서명위조가 불가능하며 권한위임기간 만료이전이라도 위임을 철회할 수 있다는 장점을 갖는다

ABSTRACT

Proxy signature schemes which allows original signer to delegate proxy signer to sign message on its behalf have a considerable amount of interest from researchers since Mambo[1] and have found many practical applications such as distributed network, Grid computing and electronic commerce. Araki[6] extended them to multi-level proxy signature. But it could not satisfy some security requirement. In this paper we propose a protocol to delegate signing right to another entity for multi-level proxy signature. Our protocol do not require secure channel and guarantee that nobody is able to repudiate delegation or acceptance of signing right, it is impossible for anyone to generate signature except designee and original signer can withdraw the delegation before expiration if it is necessary.

키워드

Ⅰ. 서 론

인터넷을 기반으로 하는 다양한 서비스, 전자상거래 의 활용, 다양한 조직과 기업의 업무 처리가 일반화됨에 따라 정보보안을 위한 암호 기술의 사용이 보편화되었다. 또한 사용자 인증과 메시지 인증의 양면성을 가진 전자서명 기술의 활용도가 점차 증대되어 가고 있다. 하지만 조직과 기업에서 책임과 권한을 가진 자가 출장, 병가, 휴가 등 여러 여건 때문에 서명을 하지 못하는 경우가 발생할 수 있다. 이러한 경우를 해결하고자 M.Mambo, K.Usuda와 E.okamoto는 대리서명(Proxy Signatu

*울산과학대학 컴퓨터정보학부

접수일자 2005. 2. 1

re)을 제안하였다[1][2]. 대리서명은 대리서명자가 원 서명자를 대신하여 원 서명자의 서명과 동일한 효력을 갖는 대리서명을 생성하고 이를 검증할 수 있는 암호프로토콜이다.

원활한 대리서명을 위하여 인증서 사용을 고려할 수 있다. 실제 조직에서 운용되는 인증서는 권한을 위임하기 위하여 인증서와 비밀키를 대리서명자에게 위임하여야 한다[1][3]. 그러나 이 방법은 인증서의 모든 권한을 위임하는 의미를 내포하기 때문에 보안상의 문제점을 가지고 있다. 보안상의 문제는 대리서명자의 인증서와 비밀키의 오남용을 막기가 힘들다는 것, 대리서명 후 부인방지를 막을 수 없다는 것, 대리서명자가 제삼자에게 원 위임자의 동의 없이 인증서와 비밀키를 알려 줌으로써 대리서명 능력을 갖게 할 수 있다는 점, 비밀키의 노출이 반복적으로 일어남으로써 안전성에 심각한 문제를 일으킬 수 있다는 점 등이다[2][4]. 따라서 B2B 전자거래 및 다양한 활용 범위에 반영되어질 수 있는 보다 안전하고 효율적인 대리서명이 요구된다[5]. 대리서명 상황에서 권한을 위임받은 대리서명자가 출장을 가야하는 상황이라면 그도 서명권한을 위임하여야 할 것이다. 이렇게 서명 권한을 위임받은 자의 서명 권한을 다시 위임하는 것을 다단계 대리서명(Multi-level Proxy Signature)방식이라 한다. 대리 서명 방식의 확장된 형태인 다단계 대리 서명 방식은 Araki[6]에 의하여 제안되었다. 이 서명 방식은 원 서명자의 위임 부인을 방지하여 대리 서명자를 보호할 수 있어야 하며, 원 서명자에게는 자신이 위임한 서명이 실제로 이루어진 상황을 알 수 있게 하여 원 서명자의 서명에 대해 보호할 수 있어야 한다. 대리서명 방식은 검증성, 위조불가능성, 신원확인성, 오용방지부인, 불가능성 권한의 제약, 양도 불가, 적합성 확인 등의 요구 조건을 만족하여야 한다[7].

이 논문에서는 이와 같은 보안요구사항을 만족하면서 동시에 보안채널을 요구하지 않으며 권한위임 및 위임수락 부인이 불가능하고 지정된 대리서명자 이외에 제3자에 의한 서명위조가 불가능하며 권한위임기간 만료이전이라도 위임을 철회할 수 있다는 특징을 갖는 다단계 대리 서명을 위한 서명권한위임 프로토콜을 제안한다.

II. 관련 연구

2.1 대리서명 기법

(1) Mambo 기법

[1][2]는 대리 서명 기법을 원 서명자의 서명 권한을 위임하는 형태에 따라 완전 위임, 부분 위임, 보증 위임 방식으로 분류하였다. 완전 위임 방식은 원 서명자가 대리 서명자에게 자신의 비밀키를 주는 방식으로 대리 서명자의 서명과 원 서명자의 서명이 구분이 되지 않는 방식이다. 부분 위임은 완전 위임 보다 안전한 방식이다. 이는 원 서명자가 대리 서명용 비밀키를 자신의 비밀키를 이용하여 생성하는 방식이다. 이 때 비밀키는 대리 서명용 비밀키로부터 계산이 불가능하여야 한다. 보증 위임 방식은 원 서명자가 대리 서명자에게 보증서를 발행함으로써 대리 서명을 구현하는 방식이다.

이 기법의 단점은 위임 권한에 대한 제약이 없으므로 대리인에 의한 오남용이 가능하다는 점과 원 서명자의 동의 없이 제3자에게 전달하여 대리서명이 가능하고 제3자가 명백한 위임자인지에 대한 결정을 할 수 없다는 것이다[7].

(2) S.Kim 기법

[8]에서는 위의 부분 위임과 보증 위임의 장점만을 취하여 보증 부분 위임 대리 서명 방식을 제안하였다. 보증 부분 위임은 원 서명자가 대리 서명용 비밀키를 자신의 비밀키와 유효기간과 대리서명자와의 관계 등이 언급된 보증서를 이용하여 생성하는 경우를 말한다. 이 때 원 서명자의 비밀키는 대리 서명용 비밀키와 보증서로부터 계산 불가능하여야 한다. 이 기법은 위임 개인키가 대리인에 의해서만 표현되기 때문에 대리인이 보호되는 장점이 있다. 그러나 대리서명 내에 원 서명자와 위임자의 역할이 동일하다는 단점이 있다[7].

2.2 Araki의 다단계 대리서명 방식

Araki는 Mambo의 대리 서명 방식을 확장하여 다단계 대리 서명 방식을 제안하였다[6]. Mambo의 대리서명 방식을 이용한 다단계 대리서명 방식은 서명용 키를 생성할 때 자신의 비밀키와 공개키를 사용하여 다시 서명용 키를 생성한다.

(1) 대리서명용 키 생성

원 서명자 U_0는 아래와 같이 대리서명용 키 σ_0를 생성하여 대리서명자 U_1에게 전송한다.

① U_0은 난수 $k_0 \in Z_{p-1}$를 선택한 후

$K_0 \equiv g^{k_0} \pmod{p}$를 계산한다.

② U_0은 대리서명용 키 σ_0를 계산한다.

$\sigma_0 \equiv x_0 + k_0 K_0 \pmod{p-1}$

③ U_0는 대리서명용 키 σ_0를 안전한 채널을통해 U_1에게 전송하고, K_0는 신뢰센터에 보낸다.

i번째 대리서명자 U_i $(i>0)$가 다른 대리 서명자 U_{i+1}에게 원 서명자 U_0의 서명 생성 능력을 위임

하고자 한다면 다음의 단계를 수행한다.

① U_i는 서명 생성 키

$\lambda_i \equiv \sigma_i + x_i y_i \pmod{p-1}$를 계산한다.

② U_i는 난수 $k_i \in Z_{p-1}$을 선택한 후

$K_i \equiv g^{k_i} \pmod{p}$를 계산한다.

③ U_i는 대리서명용 키 σ_i를 계산한다.

$\sigma_i \equiv (\sigma_{i-1} + x_i) y_i + k_i \pmod{p-1}$

④ U_i는 대리서명용 키 σ_i를 U_{i+1}에게 전송하고, K_i는 신뢰센터에 등록한다.

(2) 대리 서명용 키 검증

대리서명자 U_i는 U_{i-1}에게 받은 σ_i와 U_{i-1}의 공개키 y_{i-1}와 대리 서명용 공개정보 K_{i-1}키를 이용하여 다음 식을 계산하고 대리서명용 키를 검증한다.

$g^{\sigma_{i-1}} \equiv ((y_0 K_0^{K} y_1)^{y_1} K_1) \cdots y_{i-1})^{y_{i-1}} K_{i-1} \pmod{p}$

위 식이 검증되면, U_i는 U_0의 대리서명용 키 λ_i, σ_i를 생성할 수 있다. 여기에서 λ_i는 서명용 키이고, σ_i는 대리서명용 키이다.

(3) 서명 생성 및 검증

U_i는 ElGamal 서명과 Nyberg-Ruppel 서명과 같은 일반 서명 방식을 이용하여 $SIG_{U_i}(m, \lambda_i)$ 대리서명을 생성할 수 있다. 또한 이 서명을 받은 검증자도 다음 식과 같이 대리서명 공개키를 검증할 수 있다.

$$\rho_i \equiv g^{\lambda_i} \pmod{p}$$

$$\equiv (((y_0 K_0^{K_0} y_1)^{y_1} K_1 y_2)^{y_2} \cdots)^{y_{i-1}} K_{i-1} y_i^{y_i}$$

그리고 $Ver(Sig_{U_i}(m, \lambda_i), \rho_i)$을 이용하여 대리서명을 검증할 수 있다.

Araki의 다단계 대리서명 방식은 Mambo 방식을 이용한 다단계 대리서명 방식에서 발생하는 문제점을 해결하였으나 다음과 같은 또 다른 문제점을 가지고 있다[9].

- 원 서명자가 지정한 대리인이 아닌 다른 사람이 원 서명자를 대신하여 서명할 수 있다.
- 원 서명자가 어떤 불법적인 의도에 의해서 위임 부인을 할 경우가 발생할 수 있다.
- 다단계 확장 시 대리서명자가 또 다른 대리서명자에게 권한 위임을 하는 과정에서 제3자에 의한 불법적인 변조가 있을 수 있다.
- 원 서명자가 사후 자신의 위임 서명에 대한 결과를 알 수가 없으므로 원 서명자의 보호가 이루어지지 않는다.

III. 다단계 대리서명을 위한 권한위임 프로토콜

3.1 프로토콜 개요

그림 1은 이 논문에서 제안하는 서명권한 위임 과정의 흐름을 보여준다. 원서명자 S_0는 S_1에게 권한위임을 의뢰하고 S_1이 이를 승낙하면 승낙사실을 TC(Trusted Center)에 알린다. 그런 다음 TC는 S_1에게 서명권한대행을 허가하는 메시지를 보냄으로써 S_1는 대리서명을 수행할 수 있다.

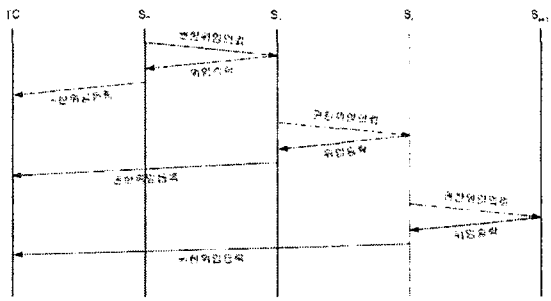

그림 1. 프로토콜 개요
Fig. 1 Protocol Outline

원서명자 S_0가 S_1에게 서명권한을 위임하기 위해서는 위임개시 및 만료일자, 지정대리인 등을 담은 권한위임정보를 S_1에게 주어야 할 것이다. 권한위임정보의 구성은 그림 2와 같다. startingTime과 expirationTime은 각각 위임개시 및 만료일자를 의미하며 proxyCandidates는 지정대리인의 리스트, designatedProxy는 대리인을 나타낸다.writtenBy는 권한 위임을 의뢰하는 위임자를 의미한다.

```
delegationInfo ::= SEQUENCE
    startingTime        INTEGER
    expirationTime      INTEGER
    proxyCandidates     STRING    OPTIONAL
    designatedProxy     STRING
    writtenBy           STRING
```

그림 2. 권한 위임 정보 구성
Fig. 2 Authority Delegation Information

대리서명자 S_i가 다시 S_{i+1}에게 S_0를 대신하여 서명을 수행하도록 하고자할 때는 그림 3과 같은 권한변경정보를 전송한다.

```
updatedInfo ::= SEQUENCE
    startingTime        INTEGER
    designatedProxy     STRING
    writtenBy           STRING
```

그림 3. 권한변경정보
Fig. 3 Authority Update Information

제안하는 프로토콜에서 사용되는 표기법은 그림4와 같다.

Notations

- p : 임의의 소수
- r_i 임의의 난수, $r_i \in Z_{p-1}$
- $R_i = g^{r_i}$
- DM_i : S_i에 의해 작성된 delegationInfo,
  $DM_i = DM_{i-1} \| UD_i$
- UD_i : S_i에 의해 작성된 updatedInfo
- $H_i = hashFunction(DM_i)$

그림 4. 프로토콜 표기법
Fig. 4 Notations

3.2 서명권한 위임 프로토콜

[권한위임의뢰]

1. 원서명자 S_0는 S_1에게 서명권한위임 메시지 $D_0 = (DM_0, R_0, d_0)$를 전송한다.

$$r_0 \in Z_{p-1}$$
$$R_0 = g^{r_0} \ (mod\, p)$$
$$d_0 = x_0 + r_0 R_0 H_0 \ (mod\, p)$$

2. S_1은 D_0가 S_0에 의해 생성되었음을 다음과 같이 확인할 수 있다

$$g^{d_0} = y_0 R_0^{R_0 H_0}$$

3. D_0가 유효하면 S_1은 S_0에게 수신확인 및 위임수락 메시지 A_1을 전송한다.

$$A_1 = x_1 + r_1 R_1 + d_0$$

4. S_0는 TC에 A_1을 전송한다.

5. TC는 S_1에게 서명권한수행 허가 메시지를 전송한다.

[권한위임변경의뢰]

1. 권한을 위임받은 S_i가 다시 S_{i+1}에게 서명권한을 위임하고자할 때 S_i는 S_{i+1}에게 서명권한위임 메시지 $D_i = (DM_i, R_0, R_1, .. R_i, d_i)$를 전송한다.

$$d_i = d_{i-1} + x_i + r_i R_i H_i$$
$$DM_i = DM_{i-1} \| UD_i$$

2. S_{i+1}는 D_i가 S_i에 의해 생성되었음을 다음과 같이 확인할 수 있다

$$g^{d_i} = y_0 R_0^{R_0 H_0} y_1 R_0^{R_1 H_1} \ldots\ldots y_i R_0^{R_i H_i}$$

3. S_{i+1}은 S_i에게 수신확인 및 위임수락 메시지 A_{i+1}을 전송한다.

$$A_{i+1} = x_{i+1} + r_{i+1} R_{i+1} + d_i$$

4. S_i는 TC에 A_{i+1}을 전송한다.

5. TC는 S_{i+1}에게 서명권한수행 허가 메시지를 전송한다.

[서명생성]

1. 대리서명자 S_i는 M에 대하여 다음과 같이 자신의 개인키로 서명을 수행한다. 서명방법은 Shnorr의 서명방법 등을 이용할 수 있다.

$$Sign_{x_0}(M \| D_{i-1})$$

2. 검증자는 먼저 D_i를 검증하고 유효하면 메시지 $(M \| D_i)$에 대한 서명 결과를 검증한다.

[서명권한위임철회]

1. 원서명자 S_0는 TC에 권한위임 철회의사를 밝힌다.

2. TC는 현재 서명권한을 수행중인 S_i에 권한수행 중단을 요구한다.

3.2 프로토콜 분석

제안된 프로토콜은 다음과 같은 특징을 갖는다.

1. secure 채널이 필요치 않다.

이는 위임장을 갖더라도 권한수행이 불가능하다는 것을 의미한다. 또한 암복호를 위한 컴퓨팅파워를 절약할 수 있다는 의미이기도하다.

2. 권한위임에 대한 부인을 할 수 없다.

원서명자나 대리서명자가 권한을 위임한 사실을 추후에 부인할 수 없도록 함으로써 대리서명자를 보호한다.

3. 권한수락에 대해서 부인할 수 없다.

위임을 허락한 대리서명자가 위임 사실을 부인하는 불법행위가 불가능하도록 설계됨으로써 원서명자 및 권한 위임자를 보호한다.

4. 지정된 서명자 이외에 제3자는 서명을 생성하

거나 위조할 수 없다.

5. 원한다면 원서명자는 현재 누구에 의해 서명이 권한이 수행되고 있는지 알 수 있다. 권한 위임 사실이 TC에 등록되어지도록 설계되었기 때문에 누구에게 권함이 위임되었는지를 확인할 수 있다.

6. 필요하다면 원서명자는 권한위임 만료 기간 내에라도 서명권한 위임을 철회할 수 있다.

IV. 결 론

대리서명은 서명자가 대리서명자에게 서명 권한을 위임하여 대리 서명자가 원 서명자를 대신하여 서명을 수행할 수 있도록 하는 전자서명 응용 기법 중 하나이다. 대리서명 방식은 위조 불가, 검증 가능성, 대리서명자 식별, 서명 부인 방지, 오용 방지 등의 기능을 가져야한다. 이를 해결하기 위한 대리서명 기법들이 제안되었으나 대리서명자가 다시 서명권한을 위임하여야하는 다단계 대리서명 방식을 만족시키기에는 문제점이 있었다. 안전한 다단계 대리서명을 위해서는 원 서명자의 위임 부인을 방지하여 대리서명자를 보호할 수 있어야 하며, 원 서명자에게는 자신이 위임한 서명이 실제로 이루어진 상황을 알 수 있게 하여 원 서명자의 서명에 대해 보호할 수 있어야 한다. 이 논문에서는 다단계 대리서명을 위한 서명권한 위임 프로토콜을 설계하였다. 이 프로토콜은 원서명자와 대리서명자 모두를 보호할 수 있다. 또한 보안채널을 요구하지 않으며 권한위임 및 위임수락 부인이 불가능하고 지정된 대리서명자 이외에 제3자에 의한 서명위조가 불가능하며 권한위임기간 만료이전이라도 위임을 철회할 수 있다는 장점을 갖는다.

참고문헌

[1] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature : Delegation of the power to sign message", IEICE Transaction on Fundamentals, vol. E79-A, no. 9, pp. 1338~1354, 1996.

[2] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signature for delegating signing operation", Proc. Third ACM Conf. on Computer and Communications Security, pp. 48~57, 1996.

[3] L.Yi, G.Bai and G.Xiao,"Proxy multi signature scheme: A new type of proxy signature scheme," Electronics Letters, Vol.36 No.6,2000

[4] T.ElGamal, "A public key crytosystem and signature scheme based on discrete logarit hms," IEEE Tran. Information Theory, Vol.31, No.4, 1985.

[5] 박소영, 이상호, "비밀분산법과 Diffie-Hellman 문제에 기반한 동적 멀티 대리서명 프로토콜", 한국정보과학회논문지(시스템 및 이론) 제31권 제8호, pp.465-472, 2004.8.

[6] S. Araki and K. Imamura, "An application of Mambo - Usuda - Okamoto Proxy Signature Schemes", Proc. of ISITA, 2000.

[7] 박세준, 오해석,"위임등록 프로토콜을 이용한 대리서명 기법", 정보처리학회논문지C 제11-C권 제 1호, pp.1-10, 2004.2.

[8] S. Kim, S. Park and D. Won, "Proxy signatures, revisited", Proc. of ICICS'97, LNCS 1334, pp. 223~232, 1997.

[9] 남기희, 이여진, 김성열, 정일용, "위임인증서를 기반으로 한 대리서명 방식 프로토콜의 설계", 한국정보과학회 제30회 춘계학술발표회 논문집, pp.431-433, 2003.4.

저자소개

김성열(Seong-yeol Kim)

1994년 조선대학교 전자계산학과(이학사)

1996년 조선대학교대학원 전자계산학과(이학석사)

2000년 조선대학교대학원 전자계산학과(이학박사)

2002년 ~ 현재 울산과학대학 컴퓨터정보학부 조교수

※관심분야 : 정보보안, 분산시스템, 전자상거래, 무선인터넷,정보가전, 임베디드 시스템