제18회 한국정보처리학회 추계학술발표대회 논문집 제9권 제2호 (2002.11)

익명성을 갖는 효율적인 1회용 대리서명

김소진*, 최재귀**, 박지환*
*부경대학교 전자계산학과
**부경대학교 정보보호학과
e-mail:sojin97@shannon.pknu.ac.kr

Efficient One-time Proxy Signature with Anonymity

So-Jin Kim*, Jae-Gwi Choi**, Ji-Hwan Park*
*Dept of Computer Science, PuKyong University

요 약

모발 에이전트는 고객을 대신하여 호스트에서 고객의 서명을 대리 수행하도록 하는 기능을 가지고 있다. 이러한 모발 에이전트를 이용한 전자상거래 환경에서 안전성을 강화하기 위한 1회용 대리서명 기법이 제안되어 있다[4]. 그러나, 전자상거래와 같은 환경에서 대부분의 사용자가 신분노출을 꺼려하는 익명성이 제공되지 않는 문제점이 있다. 따라서 본 논문에서는 무선 환경에서의 계산량과 통신량을 줄이면서 익명성을 제공할 수 있는 안전하고 효율적인 1회용 대리서명 기법을 제안한다.

1. 서론

대리서명 기법은 원 서명자가 지정한 사람이 대신해서 서명을 하는 방식으로 다음과 같은 조건을 가져야 한다[1].

- **위조 불능성**
  원 서명자가 지정한 서명자만이 정당한 대리서명을 생성할 수 있다. 제 3자는 위임서명을 할 수 없어야 한다.
- **검증 가능성**
  위임 서명을 확인하는 검증자는 대리 서명이 원 서명자가 인정한 대리 서명자에 의하여 서명되었음을 확인할 수 있어야 한다.
- **신분 확인성**
  누구든지 대리서명한 대리자의 신분을 확인할 수 있어야 한다.
- **부인 불가성**
  원 서명자를 대신하여 대리서명한 대리자는 이에 대해 부정할 수 없어야 한다.

Mambo[1]는 본인이 부재시 대신 서명을 할 수 있는 대리서명 방식을 최초로 제안하였다. 이후 KPW[2], OKW[3] 등이 Mambo 방식을 개선하였다. 그리고 KBLK 방식[4]에서는 모발 에이전트를 이용한 전자상거래에서 모발 에이전트의 안전성을 강화하기 위해 1회용 대리서명을 이용한 기법을 제안하였다. 이 제안 방식은 사용자의 신분을 보호할 수 있는 기능이 없고, 각 단계마다 사용되는 키의 수가 많아 키 관리의 문제점이 있다.

따라서 본 논문은 1회용 대리서명에서 사용되는 키의 수를 줄여 계산량과 통신량을 낮추고, 익명성을 추가한 개선된 1회용 대리서명 기법을 제안한다.

2. 1회용 대리서명[4]

모발 에이전트를 이용하는 전자상거래 환경에서는 암호화된 서명 함수 기법에서 함수의 수행권한이 호스트에 전적으로 부여된다. 때문에 호스트의 부정이 발생할 수 있으므로 KBLK 방식[4]에서는 실패-중단 서명기법을 응용한 1회용 대리서명 기법을 제안하였다. 이 기법은 오직 한 개의 메시지에 대해서만 주어진 위임키로 대리서명을 수행하도록 하는 방식으로 그 개요를 간략히 나타낸다.

- **표기법**
  - x, y, y': 사용자의 비밀키, 공개키
    $x \in Z_q^*$, $y \equiv \alpha^x$, $y' \equiv \beta^x \pmod{p}$
  - a_i , b_i: 호스트의 비밀키, 공개키

$a_i \in Z_q^*$, $i \in \{1,2,3,4\}$

$b_j \equiv \alpha^{a_j}$, $j \in \{1,3\}$, $b_k \equiv \beta^{a_k}$, $k \in \{2,4\}$

- C: 사용자의 ID, S: 호스트의 ID
- req_c: 사용자의 주문조건(요구사항)
- bid_s: 호스트의 주문정보(판매)
- msg: $h(C, S, req_c, bid_s)$, 서명대상 메시지

· 신뢰기관 T의 초기화

- p : 512비트 이상의 큰 소수, 공개
- q : $q|p-1$인 큰 소수, 공개
- α : $\alpha \in Z_p^{order\ q}$, 공개
- β: $a \in Z_q^*$, $\beta \equiv \alpha^a \pmod p$, 공개

가. 사용자는 로컬 환경에서 다음을 수행한다.

① 비밀키 $k_1, k_2, k_3, k_4 \in Z_q^*$ 선택

② r_1, r_2, r_3, r_4 계산

$r_i \equiv \alpha^{k_i}$, $i \in \{1,3\}$, $r_j \equiv \beta^{k_j} \pmod p$, $j \in \{2,4\}$

③ s_1, s_2, s_3, s_4 위임키 생성

$s_i \equiv x \cdot hash(req_c, r_i) + k_i \pmod p$, $i \in \{1,2,3,4\}$

모발 에이전트는 사용자의 요구 조건에 유효한 주문정보를 가진 호스트를 만나면, 사용자의 C, req_c, (r_1, r_2, r_3, r_4), (s_1, s_2, s_3, s_4)을 전달한다.

나. 호스트는 주문정보 bid_s를 이용하여 서명할 메시지 $msg = h(C, S, req_c, bid_s)$를 계산한다. 그리고 나서, 다음의 값들을 계산한다.

① 에이전트가 전달한 키값들을 검증

$\alpha^{s_i} \equiv y^{hash(req_c, r_i)} \cdot r_i$, $i \in \{1,3\}$

$\beta^{s_j} \equiv y'^{hash(req_c, r_j)} \cdot r_j$, $j \in \{2,4\}$

② 대리서명을 위한 서명키 생성

$x_i \equiv s_i + a_i \cdot hash(req_c, r_i)$, $i \in \{1,2,3,4\}$

③ 대리 서명에 대한 공개키 β_1, β_2 계산

$\beta_1 \equiv \alpha^{x_1}\beta^{x_2}, \beta_2 \equiv \alpha^{x_3}\beta^{x_4}, \pmod p$

④ 메시지 msg에 대한 서명 σ_1, σ_2 생성

$\sigma_1 \equiv x_1 + msg \cdot x_2 \pmod q$

$\sigma_2 \equiv x_3 + msg \cdot x_4 \pmod q$

호스트는 모발 에이전트를 통해 (S, C, bid_s), msg, (β_1, β_2), (σ_1, σ_2)를 사용자에게 전달한다.

다. 사용자는 전달받은 값들을 검증하여 호스트가 생성한 서명에 대한 인증을 수행한다.

① $m = h(C, S, req_c, bid_s)$를 계산, $m = msg$ 검사

② 서명 공개키를 검증하여 호스트가 서명키를 정당하게 생성했는지 검증

$\beta_1 \equiv (y \cdot b_1)^{hash(req_c, r_1)} \cdot (y' \cdot b_2)^{hash(req_c, r_2)} \cdot r_1 \cdot r_2 \pmod p$

$\beta_2 \equiv (y \cdot b_3)^{hash(req_c, r_3)} \cdot (y' \cdot b_4)^{hash(req_c, r_4)} \cdot r_3 \cdot r_4 \pmod p$

③ 서명 검증

$\beta_1 \beta_2^{msg} \equiv \alpha^{\sigma_1} \cdot \beta^{\sigma_2} \pmod p$

위의 방식은 사용자측의 계산량을 줄여주기 위해 호스트측에서 대리서명을 수행하는 기법이다. 그러나 대리서명을 위한 서명키 생성 과정에서 많은 키들이 계산되고, 사용된다. 그러므로 키의 사용을 줄이고, 사용자 신분을 보호하기 위한 익명성 기능을 추가한 1회용 대리서명을 제안한다.

3. 익명성을 갖는 효율적인 1회용 대리서명

제안 방식은 KBLK 방식[4]처럼 4개의 서명 비밀키가 필요하고, 서명에 대한 공개키 값들을 생성하여 공개하기 위한 등록센터를 가정하며, 사용자의 요구조건인 req_c는 시간정보를 포함한 값이므로 유일한 값으로 설정됨을 전제한다. 그리고 사용자의 익명성을 보장하기 위한 임시 비밀키/공개키 쌍은 등록센터를 통해 생성하고, 등록한다.

· 표기법

- x_c, y_c: 사용자의 비밀키, 공개키
  $x_c \in Z_q^*$, $y_c \equiv g^{x_c} \pmod p$
- $x_{h_1}, x_{h_2}, x_{h_3}, x_{h_4}, y_{h_1}, y_{h_2}, y_{h_3}, y_{h_4}$: 호스트의 비밀키, 공개키
  $x_{h_i} \in Z_q^*$, $i \in \{1,2,3,4\}$
  $y_{h_j} \equiv g^{x_{h_j}}$, $j \in \{1,2\}$, $y_{h_k} \equiv y_r^{x_{h_k}}$, $k \in \{3,4\}$
- msg: $h(\text{임시}ID_c, ID_h, req_c, bid_h)$, 서명 메시지

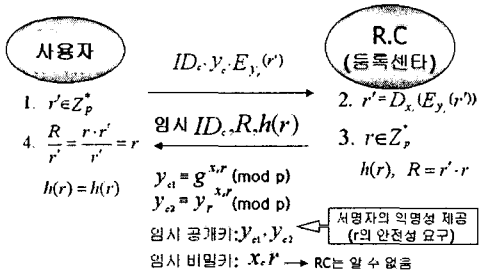
· 등록센터(R.C)의 초기화

- x_r, y_r: 등록센터의 비밀키, 공개키
  $x_r \in Z_q^*$, $y_r \equiv g^{x_r} \pmod p$
- p : 512비트 이상의 큰 소수, 공개
- q : $q|p-1$인 큰 소수, 공개
- g : $\alpha \in Z_p^{order\ q}$, 공개

가. 등록

사용자는 임시 비밀키/공개키 쌍을 얻기 위해 그림1과 같은 과정을 수행하여 임시ID_c를 얻고, 자신이 계산한 $h(r)$ 값과 전송받은 $h(r)$ 값이 같으면, y_{c_1}, y_{c_2}, $x_c r$를 생성한다. 이때 r의 유효 기간에 따라

등록 횟수는 달라진다. 만약 r의 유효 기간을 설정하지 않는다면, 등록은 한번만 하면 된다. 그러면 다음 등록 단계는 생략하고, 대리 서명 위임 단계를 수행하면 된다. 등록센터는 자신의 DB에 ID_c, y_c, $E_{y_r}(r)$를 저장하고, 모든 임시 공개키는 공개한다.



<그림1> 임시 비밀키/공개키 생성 과정

나. 사용자의 위임키 생성

등록 단계에서 생성한 임시ID_c, y_{c_1}, y_{c_2}, $x_c r$를 가지고 원 서명자는 아래와 같이 대리서명을 위한 위임키를 생성하여 모발 에이전트를 통해 호스트에게 전송한다.

① $k \in Z_{p-1}$, $K_1 \equiv g^k$, $K_2 \equiv y_r^k \pmod p$

② 임시ID_c, req_c, K_1, K_2로 다음을 계산

$e = h(\text{임시}ID_c, req_c, K_1, K_2)$

③ 자신의 임시 비밀키 $x_c r$로 위임키 생성

$(s \equiv x_c r \cdot e + k \bmod p) \bmod q$

④ 모발 에이전트를 통해 호스트에게 임시ID_c, K_1, K_2, s, req_c을 전송

다. 위임키 검증 및 대리서명 생성

호스트는 주문정보 bid_h를 이용하여 서명할 메시지 $msg = h(\text{임시}ID_c, ID_h, req_c, bid_h)$를 계산한 후, y_{c_1}, y_{c_2}, K_1, K_2, s, req_c를 이용하여 다음과 같이 위임키를 검증한다.

① $g^s \equiv y_{c_1}^{h(\text{임시}ID_c, req_C, K_1, K_2)} \cdot K_1 \pmod p$

$y_r^s \equiv y_{c_2}^{h(\text{임시}ID_c, req_C, K_1, K_2)} \cdot K_2 \pmod p$

위의 식이 검증되지않으면 수행중단.

y_{c_1}, y_{c_2}의 정당성은 등록센터에서 확인 가능.

② 대리 서명키 s_1, s_2, s_3, s_4 생성

$s_1 \equiv s + x_{h_1}$, $s_2 \equiv s + x_{h_2} \pmod q$

$s_3 \equiv s + x_{h_3}$, $s_4 \equiv s + x_{h_4} \pmod q$

④ 대리 서명에 대한 공개키 β 계산

$\beta \equiv g^{(s_1 + s_2)} \cdot y_r^{(s_3 + s_4)} \pmod p$

⑤ 메시지 msg에 대한 서명 σ_1, σ_2 생성

$\sigma_1 \equiv (s_1 + s_2)/(msg + x_{h_1} - x_{h_2}) \pmod q$

$\sigma_2 \equiv (s_3 + s_4)/(msg + x_{h_3} - x_{h_4}) \pmod q$

호스트는 모발 에이전트를 통해 (ID_h, bid_h), msg, (β), (σ_1, σ_2)를 사용자에게 전달한다.

라. 서명 검증

사용자는 전달받은 값들을 검증하여 호스트가 생성한 서명에 대한 인증을 수행한다.

① $m = h(\text{임시}ID_c, ID_h, req_c, bid_h)$로 $m = msg$ 검사

② 서명 공개키를 검증하여 호스트가 서명키를 정당하게 생성했는지 검증

$\beta \equiv y_{c_1}^{2e} \cdot K_1^2 \cdot y_{h_1} \cdot y_{h_2} \cdot y_{c_2}^{2e} \cdot K_2^2 \cdot y_{h_3} \cdot y_{h_4}$

③ 서명 검증

$\beta \equiv (y_{h_1} \cdot y_{h_2}^{-1} \cdot g^{msg})^{\sigma_1} \cdot (y_{h_3} \cdot y_{h_4}^{-1} \cdot y_r^{msg})^{\sigma_2}$

4. 제안 기법의 안전성 분석 및 고찰

▶ 안전성

[정리1] 원 서명자만이 위임키를 생성할 수 있다.

$s \equiv x_c r \cdot e + k$

- 위임키에 대한 안전성은 이산대수 문제의 어려움에 기반하여 원 서명자의 비밀키를 모르면 생성할 수 없다.

[정리2] 원 대리서명자만이 대리서명을 할 수 있다.

$s_1 \equiv s + x_{h_1}$, $s_2 \equiv s + x_{h_2}$

$s_3 \equiv s + x_{h_3}$, $s_4 \equiv s + x_{h_4}$

- 서명키의 안전성도 이산대수 문제의 어려움에 기반함으로 대리자의 비밀키를 모르면 키를 생성할 수 없으며, 서명도 불가능하다.

[정리3] 대리 서명자의 서명이 1회성임을 보장한다.

- 2번 이상 사용하면 그림2와 같이 비밀키가 노출된다.

[정리4] 서명단계는 실패-중단 서명기법의 안전성과 동일하다.

- 이산대수 문제의 어려움에 기반함으로 서명의 위조는 등록센터의 비밀키 x_r를 알아야 한다.

$$(g^{x_{h_1}} \cdot g^{-x_{h_2}} \cdot g^{msg})^{(\sigma_1 - r_1)} = (y_r^{x_{h_1}} \cdot y_r^{-x_{h_2}} \cdot y_r^{msg})^{(r_2 - \sigma_2)}$$
$$= (g^{x_{h_1}} \cdot g^{-x_{h_2}} \cdot g^{msg})^{x_r(r_2 - \sigma_2)}$$

[정리5] 원 서명자의 익명성을 보장한다.

- 이산대수 문제의 어려움에 기반한다.
- 임시 공개키 $y_{c_1} \equiv g^{x_c r}$, $y_{c_2} \equiv y_r^{x_c r}$에서 $x_c r$은 오직 원 서명자만 알고, 원 공개키 y_c는 등록센터만 안다.

(1) σ_1: 증명

$\sigma_1=(s_1+s_2)/(msg+x_{k1}-x_{k2})$ $\sigma_1'=(s_1+s_2)/(msg'+x_{k1}-x_{k2})$

$\sigma_1 msg+\sigma_1 x_{k1}-\sigma_1 x_{k2}=s_1+s_2$ $\sigma_1' msg'+\sigma_1' x_{k1}-\sigma_1' x_{k2}=s_1+s_2$

$\sigma_1 msg+\sigma_1 x_{k1}-\sigma_1 x_{k2}=\sigma_1' msg'+\sigma_1' x_{k1}-\sigma_1' x_{k2}$ ①

①의 식을 계산(식 2개, 미지수 2개)하면, x_{k1}, x_{k2} 노출

(2) σ_2: 증명

$\sigma_2=(s_3+s_4)/(msg+x_{k3}-x_{k4})$ $\sigma_2'=(s_3+s_4)/(msg'+x_{k3}-x_{k4})$

$\sigma_2 msg+\sigma_2 x_{k3}-\sigma_2 x_{k4}=s_3+s_4$ $\sigma_2' msg'+\sigma_2' x_{k3}-\sigma_2' x_{k4}=s_3+s_4$

$\sigma_2 msg+\sigma_2 x_{k3}-\sigma_2 x_{k4}=\sigma_2' msg'+\sigma_2' x_{k3}-\sigma_2' x_{k4}$ ②

②의 식을 계산(식 2개, 미지수 2개)하면, x_{k3}, x_{k4} 노출

그러므로 서명키를 2번 이상 사용할 경우,
- 대리자 자신의 비밀키가 노출되고 대리 서명키 (s_1, s_2, s_3, s_4)도 노출된다.

<그림2> 1회용 대리서명임을 증명

▶ 제안방식 고찰

(1) 익명성 - 사용자의 신분 보호

(2) 인증성 - 사용자는 서명 공개키를 검증하고, 대리자는 위임키를 검증하여 정당성을 확인

(3) 효율성 - 대리자가 사용자를 대신하여 서명함으로 사용자측의 계산량을 줄여 효율성 확보

(4) 부인봉쇄- 대리서명 생성시 사용자와 대리자의 비밀정보를 포함하므로 서명생성에 대한 서로의 부인방지 가능

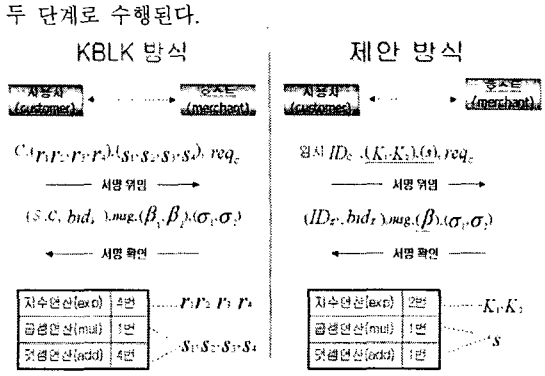
(5) 안전성 - 이산대수 문제의 어려움에 기반

표1과 같이 제안 방식에서 사용되는 키는 KBLK 방식에 비해 총 8개가 감소한다.

(표1) 1회용 대리서명 비교

	KBLK 방식	제안 방식
통신 횟수	2회	2회
사용자의 원비밀키 (원공개키)	1개 (2개)	1개 (2개-익명성)
대리자의 원비밀키 (원공개키)	4개 (4개)	4개 (4개)
사용자의 위임키	4개	1개
사용자의 임시 비밀키	4개	1개
사용자의 임시 공개키	4개	2개
대리자의 대리 서명키	4개	4개
1회성	○	○
익명성	x	○

대리서명을 수행하여 검증하는 과정은 그림3과 같이 두 단계로 수행된다.



<그림3> 통신량/계산량 비교

제안 기법은 호스트측의 서명 생성 단계에서 계산량이 조금 증가하지만, 사용자측에서의 통신량과 계산량은 상대적으로 낮음을 알 수 있다. 또한 KBLK 방식은 위임키 4개, 서명 공개키 2개가 사용되지만, 제안 방식은 위임키 1개, 서명 공개키 1개가 사용됨으로 위임키와 서명 공개키에 대한 검증 단계도 KBLK 방식보다 간단하다.

5. 결론

본 논문에서는 KBLK 방식을 개선한 1회용 대리서명 기법을 제안하였다. 대리자가 악의를 가지고 위임키를 2번이상 사용할 경우, 정리3에 의해 대리자의 비밀키가 노출되기 때문에 재사용이 불가능하다. 그리고 사용자측의 계산량을 상대적으로 감소시켜 이동통신에 적용 가능하다. 또한 무선 환경은 유선 환경에 비해 도청자나 그 밖의 위조나 불법적 변경 등과 같은 위협들에 매우 취약하므로 사용자들의 신분을 보호할 수 있게 익명성을 추가하였다.

참고문헌

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

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

[3] 오수현, 김현주, 원동호, " 이동 통신 환경에서의 전자 상거래에 적용할 수 있는 Proxy-signcryption 방식", 통신정보보호학회논문지 제10권 제2호, 2000. 6.

[4] 김희선, 백준상, 이병천, 김광조, "대리서명을 이용한 모발일 에이전트의 안전성 강화 방법", 한국정보보호학회, 종합 학술발표 논문집, Vol. 10, No. 1, pp. 424~437, 2000.