유효 기간을 갖는 포워드-시큐어 대리 서명 방법
(A Time-limited Forward-secure Proxy Signature Scheme)

김 상 희 † 조 태 남 †† 이 상 호 ††† 채 기 준 ††† 박 원 주 †††† 나 재 훈 ††††
(Sanghee Kim) (Taenam Cho) (Sang-Ho Lee) (Kijoon Chae) (Wonjoo PARK) (Jaehoon Nah)

요 약 대리 서명이란 원 서명자가 대리 서명자에게 서명 권한을 위임하여, 대리 서명자가 원 서명자를 대신해서 서명을 생성하는 것이다. 일반적으로 대리 서명자가 위임받은 권한은 유효 기간을 가지며, 이를 위해 위임 정보에 위임 기간을 포함시키는 방법이 있다. 그러나 일반적인 방법에서는, 대리 서명자의 서명 생성 시간을 알 수 없기 때문에 유효 기간이 만료된 대리 서명자의 서명 위조를 막을 수 없고, 위임 기간 중에 대리 서명키가 노출되었을 경우 정당한 대리 서명자가 과거에 생성한 서명의 타당성을 보호하지 못한다. 본 논문에서는 위임 기간이 만료된 대리 서명자의 서명 위조를 막고, 현재의 서명키가 노출되더라도 과거 서명의 타당성이 보호되는 새로운 서명 방법을 제안한다. 본 논문에서 제안한 유효 기간을 갖는 포워드-시큐어 대리 서명 프로토콜은 원 서명자가 시간 관련 파라미터를 제어하므로 정확한 시간 정보를 필요로 하지 않는다. 또한, 어플리케이션의 특성이나 정책에 따라 설정된 서명키 갱신 구간별로 포워드-시큐어한 속성을 만족시킨다.

키워드 : 대리 서명, 포워드-시큐어 서명, 위임 기간

Abstract Proxy signature scheme is a cryptographic protocol that an original signer delegates her signing capability to a proxy signer, and then the proxy signer is able to create signatures on behalf of the original signer. In general, there is time-limit for which the signing capability of the proxy signer is valid. One of methods to limit the valid delegation time is to make public delegation information contain the expiration date of the delegation. However, in this method we cannot prevent the proxy signer from signing after the valid delegation is expired because no one knows the exact time when the proxy signer signed a message. The validity of the past legal signatures cannot be preserved in case that the proxy signer's key is compromised during the delegation period.

In this paper, we propose a new scheme, time-limited forward-secure proxy signature protocol, which prevents the proxy signer from signing after the valid delegation is expired and which preserves the validity of the past legal signatures even if the signing key is compromised. The proposed scheme does not require the exact time-information by making an original signer control time-related parameters and satisfies the forward-security property in each update-period of the proxy signing key. The time-period is determined according to the application characteristics or security policies.

Key words : proxy signature, forward-secure signature, delegation

· 본 연구는 한국전자통신연구원 정보보호연구본부 네트워크보안연구부 위탁연구과제에 의한 것임
† 비 회 원 : 이화여자대학교 컴퓨터학과
kshee78@nate.com
†† 정 회 원 : 이화여자대학교 컴퓨터학과
tncho@ewha.ac.kr
††† 종신회원 : 이화여자대학교 컴퓨터학과 교수
shlee@ewha.ac.kr
kjchae@ewha.ac.kr
†††† 비 회 원 : 한국전자통신연구원 연구원
wjpark@etri.re.kr
jhnah@etri.re.kr
논문접수 2002년 11월 15일
심사완료 2003년 5월 19일

1. 서 론

전자 결재 시스템을 사용하는 회사나 기관에서 상위자의 부재시 하위자가 부재 기간동안 업무를 대행할 수 있도록 하는 체계적 방법이 필요하다. 또한, 인터넷을 통하여 서버가 인증한 데이터나 소프트웨어를 분배하는 시스템에서 사용자가 많은 경우에 중앙 서버의 로드를 줄이기 위해서는 지역 서버가 중앙 서버의 인증 및 데이터와 소프트웨어의 분배 역할을 분담할 수 있어야 한다. 이러한 권한 위임의 방법으로서 대리 서명 방법을 사용할 수 있다.

대리 서명이란 원 서명자가 대리 서명자에게 서명 권한을 위임하여, 대리 서명자가 원 서명자를 대신해서 서명을 생성하는 것이다. 수신자는 대리 서명을 검증하는 과정에서 대리 서명자의 정당한 서명임을 검증할 수 있어야 할 뿐만 아니라 원 서명자의 위임 동의를 확인할 수 있어야 한다. 이를 위해서 원 서명자는 대리 서명자에게 위임 정보를 전송해 주고, 대리 서명자는 이 위임 정보로부터 대리 서명키를 만들어 그 키로 대리 서명을 생성한다. 대리 서명키는 원 서명자가 생성한 위임 정보로부터 유도되었기 때문에, 수신자는 서명 검증 과정에서 원 서명자의 위임 동의를 확인할 수 있다.

대리 서명자의 대리 서명 권한은 원 서명자의 부재 기간이나 원 서명자가 위임한 기간동안만 유효하다. 위임 기간이 만료되면, 원 서명자는 권한 취소 프로토콜을 사용하여 위임했던 권한을 취소할 수 있다[1]. 원 서명자가 권한을 위임할 당시에 위임 정보에 위임 기간을 포함시켜서 대리 서명자가 그 기간동안만 서명을 생성할 수 있도록 하면 별도의 권한 취소 프로토콜을 사용하지 않아도 된다[2,3]. 그러나 위임 정보에 위임 기간을 명시하더라도 대리 서명자가 서명을 생성한 시간을 알 수 없기 때문에, 위임 기간이 만료된 대리 서명자의 서명 위조를 막을 수 없다. 또한, 서명키가 노출되었을 경우에 과거에 생성된 정당한 서명과 위조된 서명을 구분할 수 없다. 이러한 서명키 노출 문제를 해결하기 위해서 연구되고 있는 포워드 시큐어(forward-secure) 서명은 현재의 서명키가 노출되더라도 이로부터 과거의 서명키를 유도해내지 못하게 함으로써 과거에 생성된 정당한 서명을 보호하는 방식이다. 위임 기간이 만료된 대리 서명자의 서명 위조를 막기 위해서 타임-스탬프(time stamp)를 이용한 대리 서명 방법은[4] 포워드 시큐어한 속성은 만족하지만 서명에 시간 정보가 포함되기 때문에 대리 서명자와 수신자가 정확한 시간 정보를 얻을 수 있는 메커니즘이 필요하며, 매 서명마다 타임-스탬프를 생성하여 인증된 저장 공간에 공고해야 한다.

서명키 노출의 문제는 원 서명자뿐만 아니라 대리 서명자에게도 해결해야 할 문제이다. 따라서 위임 기간동안 대리 서명키가 노출되더라도 합법적인 서명은 보호될 수 있는 방법이 필요하다. 본 논문에서는 원 서명자가 위임 기간을 확인할 수 있는 정보의 공고를 제어함으로써 대리 서명자와 수신자가 별도의 시간 정보를 사용하지 않고 위임의 유효성을 검증할 수 있으며 시간 구간별로 포워드 시큐어한 대리 서명 스킴을 제안한다. 본 논문은 6장으로 구성된다. 2장에서는 관련 연구를 기술하며, 3장에서는 유효 기간을 갖는 포워드 시큐어 대리 서명을 제안한다. 4장에서는 제안한 방법의 안전성과 효율성을 분석하며, 5장에서는 기존의 프로토콜과 안전성 및 효율성을 비교하고, 실험 결과를 분석한다. 그리고 6장에서 결론을 맺는다.

2. 관련 연구

2.1 포워드-시큐어 서명

일반적인 전자 서명 방식에서는 서명키가 노출되었을 때, 정당한 서명자가 과거에 생성한 서명과 공격자에 의해 위조된 서명을 구분할 수 없기 때문에 모든 서명을 무효화해야 한다. 이러한 키 노출 문제를 해결하기 위해서 많은 연구가 진행되고 있으며[5-10], 그 중에서 포워드 시큐어 서명은 현재의 서명키가 노출되더라도 정당한 서명자가 과거에 생성한 서명의 타당성을 유지시켜 주는 서명 기법이다.

일반적인 전자 서명 스킴을 그대로 이용하면서 포워드-시큐어한 속성을 만족시키기 위한 방법으로서, 신뢰된 타임-스탬핑 기관(trusted time-stamping authority)을[11] 이용해서 문서에 시간 정보를 포함시키는 방법이 있다. 그러나 제3의 신뢰 기관을 이용하는 것은 스킴의 제약 사항이 되고 비용도 많이 들기 때문에[5], 지금까지 포워드 시큐어 서명 스킴에 대한 연구는 신뢰 기관의 개입을 배제하고 있다. 신뢰 기관을 사용하지 않는 방법으로서, 일정 시간 구간마다 서명키를 갱신하는 방법들이 제시되었다[5-8]. 이는 현재의 서명키로부터 과거의 서명키를 유도할 수 없도록 함으로써 과거의 정당한 서명을 보호하고, 서명 검증키는 변동시키지 않도록 함으로써 사용자의 편의를 제공한다.

포워드 시큐어 서명 스킴들은 효율성 측면에서 저장 공간이나 계산량 등에서 상충 관계(trade-off)가 있지만, 서명 검증키를 고정된 시간 구간동안 사용한다면 기본적으로 주요 파라미터인 서명키, 서명 검증키 및 서명의 크기가 시간 구간의 수에 독립적이어야 한다[5-7]. Fiat-Shamir 스킴을 이용한 방법은[5] 주요 파라미터의 크기가 시간 구간의 수에 독립적인 최초의 스킴으로서, 소인수 분해 문제의 어려움에 기반하여 키 갱신 알고리즘을 설계하였다. 그러나 이 알고리즘이 기반하고 있는 Fiat-Shamir 서명 스킴의 계산량이 많기 때문에, 이 서명 스킴 역시 계산량이 많다. Guillou-Quisquater 서명을 이용한 방법은[6] 강력한 RSA 가정 즉, 두 소수의 곱 n과 Z_n^*상의 값 α가 주어졌을 때, $\beta^\gamma = \alpha$ ($\gamma > 1$)를 만족하는 $\beta \in Z_n^*$를 찾기 어렵다는 사실에 기반한 서명 스킴이다. 이 방법은 서명키를 갱신하는 횟수보다 서명

을 생성하는 횟수가 더 빈번하다고 가정하고, 서명키 생성과 갱신의 계산량을 시간 구간의 수에 비례하게 하는 대신에 서명 생성과 검증의 계산량을 최적화시켰다. 또한, 특정 서명 스킴에 포워드-시큐어한 속성을 부가한 것이 아니라 마스터 공개키를 이용하여 특정 시간 구간에 사용할 공개키들을 체인(chain)의 형태로 인증함으로써 임의의 서명 스킴을 포워드-시큐어하게 만드는 방법도[7] 제안되었다. 그러나 이 방법은 키 생성 계산량과 부가적인 저장 공간이 시간 구간의 수에 비례한다.

2.2 대리 서명

[1]에서는, 제안한 대리 서명 스킴이 대리 서명자가 무한하게 서명을 생성할 수 있기 때문에 부정직한 대리 서명자의 권한 남용 가능성이 있음을 지적하였다. 이를 방지하기 위해 대리 서명자의 서명 권한을 취소하기 위한 방법으로서 서명자 취소 목록을 공개하는 방법과 원 서명자가 새로운 위임 정보를 생성하여 정직한 대리 서명자에게 재분배하는 방법을 제시하고 있다. [2]에서 제안한 대리 서명 스킴은 위임 정보에 위임 기간을 포함시킴으로써 별도의 권한 취소 프로토콜을 필요로 하지 않는다. 그러나 위임 정보에 위임 기간을 명시적으로 표시하더라도, 대리 서명자의 서명 생성 시간을 알 수 없기 때문에 위임 기간이 만료된 대리 서명자의 서명 위조 가능성이 있다[2,3].

Sun의 대리 서명 방법은[4] 타임-스탬프를 이용하여 대리 서명자가 위임 기간동안만 서명을 생성할 수 있도록 설계되었다. 이 스킴은 포워드-시큐어한 속성을 제공하므로 본 논문에서 제안하는 스킴과의 비교 분석을 위하여 프로토콜을 그림 1에 기술하였다. 그림 1에서 사용되는 용어들은 다음과 같으며, 3장 이후에서도 같은 용어를 사용한다.

- p, q : $q|p-1$을 만족하는 큰 소수
- g : 위수가 q인 Z_p^*상의 서브 그룹 생성자
- $h()$, $H()$: 충돌 회피성(collision-resistant) 일방향(one-way) 해쉬 함수

단계	원 서명자	단계	대리 서명자	단계	수신자
1 위임 정보 생성	• $k_1 \in_R Z_q^*$ 선택 • $K_1 = g^{k_1}$과 $s = x_O \cdot y_O + k_1 \cdot h(m_w, K_1)$ 계산 • 대리 서명자에게 (m_w, K_1, s) 전송	2 위임 정보 검증	• $g^s = y_O^{y_O} \cdot K_1^{h(m_w, K_1)}$ 등식 검증	4 서명 요청	• $k_3 \in_R Z_q^*$ 선택 • $K_3 = g^{k_3}$ 계산 • 대리서명자에게 (m, K_3) 전송
		3 대리 서명키 생성	• $x_a = s + x_P \cdot y_P$ 계산		
		5 대리 서명 생성	• c 생성 • $k_2 \in_R Z_q^*$ 선택 • $K_2 = g^{k_2}$ 계산 • $L_n = h(n, m, c, K_1, K_2, K_3, m_w, ID_r, L_{n-1}, L_{f(n)})$ 계산 • $\sigma = x_a + k_2 \cdot h(m, c, K_1, K_2, K_3, m_w, ID_r, L_n)$ 계산 • 수신자에게 $(m, \sigma, K_1, K_2, m_w, c, L_n)$ 전송	6 대리 서명 검증	• c 검증 • $g^\sigma = y_O^{y_O} \cdot K_1^{h(m_w, K_1)} \cdot y_P^{y_P} \cdot K_2^{h(m, c, K_1, K_2, K_3, m_w, ID_r, L_n)}$ 등식 검증

m_w : 원 서명자의 신원, 대리 서명자의 신원, 위임 권한의 유효 기간 등이 표시된 보증 정보
c : 대리 서명 생성 시간에 대한 정보
L_n : 이진 연결 스킴(binary linking scheme)에서의[12] n번째 연결 정보(linking information)
ID_r : 수신자의 신원

그림 1 Sun의 대리 서명 프로토콜

- x_O , $y_O = g^{x_O} \bmod p$: 원 서명자 O의 개인키와 공개키
- x_P , $y_P = g^{x_P} \bmod p$: 대리 서명자 P의 개인키와 공개키
- $S()$: 이산 대수 문제의 어려움에 기반한 서명 생성 알고리즘
- $V()$: 이산 대수 문제의 어려움에 기반한 서명 검증 알고리즘

연결 정보 L_n은 상대적이고 일시적인 인증(RTA : Relative Temporary Authentication)을 제공하기 위해서 대리 서명자가 인증된 저장 공간에 공고하며, 위임 기간이 만료되면 마지막 연결 정보 L_z를 이용하여 종료(ending) 연결 정보 $L_{end} = h(z+1, L_z, L_{f(z+1)})$를 계산하여 공고한다.

위임 기간이 만료된 대리 서명자와 수신자의 공모가 의심될 때에는 이진 연결 스킴의 검증 과정에 따라서 공고된 연결 정보를 검증한다. 현재의 연결 정보에는 과거 연결 정보의 해쉬 함수 값이 포함되기 때문에 과거의 서명을 위조할 수 없으므로 포워드-시큐어하다. 그러나 서명 생성 및 검증시에 대리 서명자와 수신자가 정확한 시간 정보 c를 생성하고 이를 검증할 수 있는 메커니즘이 필요하며, 이진 연결 스킴의 함수 $f(n)$을 계산하고 매 서명마다 연결 정보를 생성하여 인증된 저장 공간에 공고해야 하는 단점이 있다.

3. 유효 기간을 갖는 포워드-시큐어 대리 서명 (TFPS, Time-limited Forward-secure Proxy Signature)

3.1 프로토콜의 요구 사항

본 논문에서는 다음과 같은 요구 사항을 만족하는 TFPS를 제안한다.

- 보안 요구 사항

대리 서명에 대한 보안 요구사항을[1,3] 만족해야 한다.

① 강력한 위조 불가능성(strong unforgeability): 원 서명자에 의해 지명된 대리 서명자만이 대리 서명을 생성할 수 있어야 한다. 원 서명자를 포함한 어떤 제3자도 대리 서명자를 가장하여 타당한 서명을 생성할 수 없어야 한다.

② 검증성(verification): 검증자는 대리 서명으로부터 원 서명자의 위임 동의를 확인할 수 있어야 한다.

③ 강력한 신원 확인성(strong identifiability): 누구나 대리 서명으로부터 대리 서명자의 신원을 확인할 수 있어야 한다.

④ 강력한 부인 불가능성(strong undeniability): 대리 서명자는 자신이 서명한 사실을 부인할 수 없어야 한다.

⑤ 오용 방지(prevention of misuse): 대리 서명자는 원 서명자로부터 위임받은 권한 이외의 목적으로 대리 서명키를 사용할 수 없어야 한다. 만약, 대리 서명키 오용의 문제가 발생하면 대리 서명자의 책임이 명시적으로 드러나야 한다.

또한, 대리 서명 생성을 위임 기간으로 제한하고, 서명키 노출 문제를 해결하기 위해서는 다음과 같은 요구 사항을 만족해야 한다.

⑥ 대리 서명자는 원 서명자가 설정한 위임 기간동안 타당한 대리 서명을 생성한다.

⑦ 위임 기간이 만료된 대리 서명자는 수신자와 공모하여 서명을 생성할 수 없다.

⑧ 대리 서명자는 시간 구간별로 포워드-시큐어 서명을 생성한다. 즉, 현재 시간 구간의 서명키로부터 과거 시간 구간의 서명키를 유도해 내지 못한다.

- 효율성 요구 사항

① 대리 서명자는 효율적인 포워드-시큐어 서명을 생성한다. 즉, 주요-파라미터인 대리 서명키, 대리 서명 검증키, 대리 서명의 크기가 시간 구간의 수에 독립적이다. 또한, 포워드-시큐어 서명을 위한 부가적인 저장 공간을 필요로 하지 않는다.

② 대리 서명자와 수신자는 정확한 시간 정보를 필요로 하지 않는다.

3.2 TFPS

TFPS 프로토콜은 다섯 단계로 구성된다(그림 2 참조). 위임 정보 생성 단계에서는 원 서명자가 보안 강도에 따라 보안 파라미터인 대리 서명키 갱신 주기와 위임 기간을 설정하고, 이에 대한 위임 정보를 생성하여 대리 서명자에게 전송한다. 그리고 위임 정보와 시간 구간에 관련된 파라미터들을 공고한다. 위임 정보 검증 단계에서는 대리 서명자가 위임 정보의 타당성을 검증하고, 대리 서명키 생성 및 갱신 단계에서는 매 시간 구간마다 대리 서명키를 갱신한다. 대리 서명 생성 단계에서는 해당 시간 구간의 대리 서명키를 이용하여 효율적인 포워드-시큐어 대리 서명을 생성하고, 이를 수신자에게 전송한다. 마지막으로 대리 서명 검증 단계에서는 수신자가 대리 서명자의 정당한 서명임을 검증하고, 원 서명자의 위임 동의와 함께 위임 권한의 유효 기간을 확인한다.

위임 기간이 만료되면, 원 서명자는 공고했던 파라미터들을 삭제하고 대리 서명자는 위임 기간동안 생성한 서명의 해쉬값들을 원 서명자에게 전송한다.

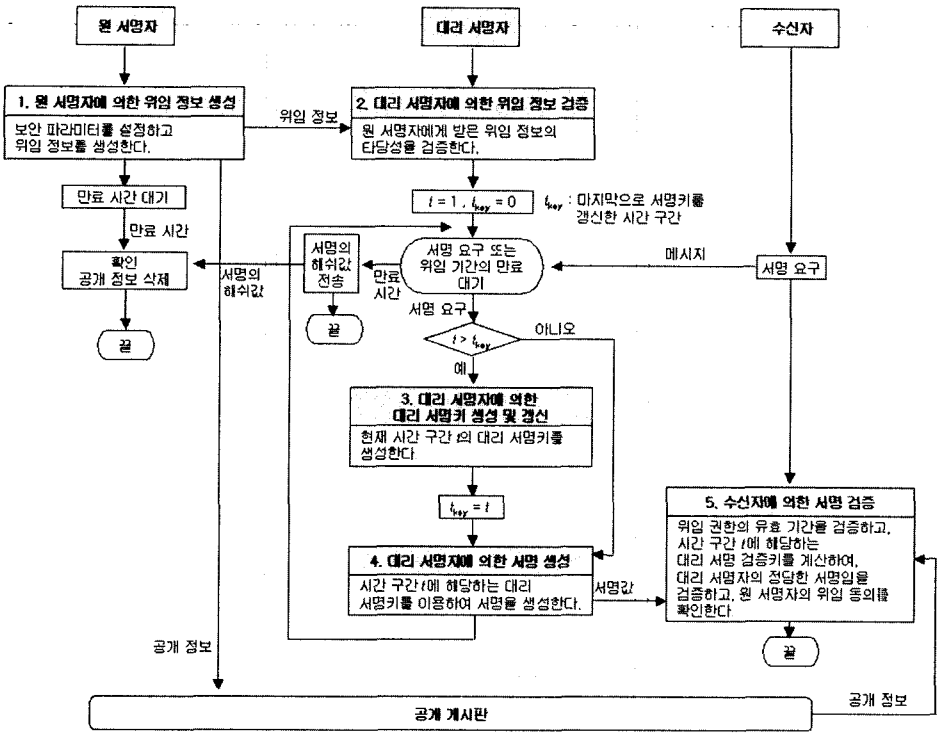


그림 2 TFPS 프로토콜

① 원 서명자에 의한 위임 정보 생성

1	보안 파라미터로써 대리 서명키 갱신 주기 u, 위임 기간 $L(=u \cdot T)$을 설정하고(그림 3 참조), 보증 정보 m_w를 생성한다.
2	해쉬 체인의 초기 입력값 $A \in_R Z_q^*$를 선택한다.
3	위임 권한이 만료되는 시간에 대한 해쉬값 $H^{T+1}(A)=E$를 계산한다.
4	$k \in_R Z_q^*$를 선택하고, $r=g^k \pmod p$과 $s=x_O \cdot h(m_w, r, E)+k \pmod q$를 계산한다.
5	r, T, E를 공고한다.
6	대리 서명자에게 (m_w, s, A, r, E)를 전송한다.

보증 정보 m_w에는 원 서명자의 신원, 대리 서명자의 신원 또는 가능한 대리 서명자의 집합, 위임 기간 L, 갱신 주기 u와 대리 서명키 갱신 횟수 T에 대한 설명, 서명 권한에 대한 세부적인 사항 등이 포함된다. 보증 정보 m_w에 대리 서명자의 신원이 명시되기 때문에 s를 공개된 채널로 전송하더라도 공격자가 이 정보를 이용하여 정당한 대리 서명자임을 가장할 수 없다.

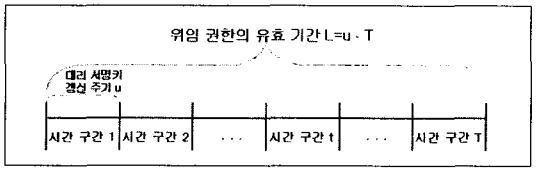


그림 3 보안 파라미터

② 대리 서명자에 의한 위임 정보 검증

1	$g^s = y_O^{h(m_w, r, E)} \cdot r \pmod p$의 등식이 성립하는지 확인함으로써 위임 정보의 타당성을 검증한다.

③ 대리 서명자에 의한 대리 서명키 생성 및 갱신

매 시간 구간 t $(1 \le t \le T)$마다 다음을 수행한다.

1	$H^t(A)=H(H^{t-1}(A))$를 계산하여 시간 구간을 갱신한다.
2	$b_t \in_R Z_q^*$를 선택하고, $B_t=g^{b_t} \pmod p$를 계산한다.
3	b_{t-1}를 삭제하여 포워드-시큐어한 속성을 만족시킨다.
4	시간 구간 t동안 사용할 대리 서명키 x_t를 생성한다. $x_t=s+x_P \cdot h(H^t(A), B_t, r)+b_t \pmod q$

④ 대리 서명자에 의한 서명 생성

1	보증 정보 m_w를 통해서 메시지 m이 서명 위임된 사항인지 확인한다.
2	현재 시간 구간 t의 대리 서명키 x_t로 메시지 m에 대한 서명 $\sigma = S(x_t, m)$을 생성한다.
3	수신자에게 대리 서명 $(m, \sigma, m_w, r, E, t, H^t(A), B_t)$를 전송한다.

⑤ 수신자에 의한 서명 검증

1	보증 정보 m_w를 통해서 메시지 m이 서명 위임된 사항인지 확인한다.
2	$H^{T-t+1}(H^t(A)) = E$인지 확인함으로써 위임 권한의 유효 기간을 검증한다.
3	대리 서명 검증키 y_t를 계산한다. $y_t = y_O^{h(m_w, r, E)} \cdot r \cdot y_P^{h(H^t(A), B_t, r)} \cdot B_t \pmod p$
4	현재 시간 구간 t의 대리 서명 검증키 y_t로 $V(y_t, m, \sigma) = true$인지 검증한다.

위임 기간이 만료되면, 원 서명자는 공고했던 파라미터들을 삭제하고 대리 서명자는 위임 기간동안 생성한 서명의 해쉬값들을 서명하여 원 서명자에게 전송한다. 위임 기간 만료 후 서명을 검증하고자 할 때에는 원 서명자에게 서명의 타당성 여부를 확인할 수 있다.

4. 프로토콜의 안전성 및 효율성 분석

4.1 안전성

- 강력한 위조 불가능성 : $S()$, $V()$가 안전하다고 가정하였을 때, 대리 서명을 위조하려면 대리 서명키를 위조해야 한다. 이것은 대리 서명 검증키 y_t로부터 $y_t = g^{x_t}$를 만족하는 x_t를 알아내는 문제와 같으며, 이산 대수 문제의 어려움에 기반하여 계산상 불가능하다. 또한, 대리 서명키 $x_t = s + x_P \cdot h(H^t(A), B_t, r) + b_t$는 대리 서명자의 개인키 x_P를 알아야만 계산할 수 있고, 보증 정보 m_w에 대리 서명자 P가 정당한 서명자임이 명시되어 있는 s를 이용해야 한다. 따라서, 합법적인 대리 서명자만이 대리 서명을 생성할 수 있고 원 서명자를 포함한 어떤 제 3자도 타당한 서명을 생성할 수 없다.
- 검증성 : 수신자가 대리 서명을 검증하기 위해서 대리 서명 검증키 $y_t = y_O^{h(m_w, r, E)} \cdot r \cdot y_P^{h(H^t(A), B_t, r)} \cdot B_t$를 계산할 때, 원 서명자의 공개키 y_O를 통하여 원 서명자의 위임 동의가 검증된다.
- 강력한 신원 확인성 : 수신자가 대리 서명을 검증하기 위해서 대리 서명 검증키 $y_t = y_O^{h(m_w, r, E)} \cdot r \cdot y_P^{h(H^t(A), B_t, r)} \cdot B_t$를 계산할 때, 대리 서명자의 공개키 y_P를 통하여 대리 서명자의 신원이 확인된다.
- 강력한 부인 불가능성 : 강력한 위조 불가능성에 의해서 대리 서명자는 자신이 서명한 사실을 부인할 수 없다.
- 오용 방지 : 보증 정보 m_w에 명시되지 않은 목적으로 대리 서명키가 사용되었다면, 강력한 위조 불가능성에 의해서 대리 서명자 P의 책임이 명시적으로 드러난다.
- 위임 기간 : 위임 기간이 만료되면, 원 서명자는 시간 구간에 관련된 파라미터들을 삭제한다. 따라서, 수신자는 원 서명자가 공고한 파라미터를 확인해 봄으로써 위임 권한의 유효 기간을 검증할 수 있다.
- 공모 불가능성 : 위임 기간이 만료되면, 대리 서명자는 위임 기간동안 생성한 서명의 해쉬값들을 원 서명자에게 전송하기 때문에 위임 기간이 만료된 대리 서명자와 수신자의 공모는 불가능하다.
- 포워드-시큐어 : 시간 구간 t의 서명키 $x_t = s + x_P \cdot h(H^t(A), B_t, r) + b_t$가 노출되더라도 공격자는 x_P, b_t 값을 알아낼 수 없고, 이들이 노출되더라도 b_j $(1 \le j < t)$는 b_t와 연관성이 없는 난수이기 때문에 이로부터 b_j를 알아낼 수 없다. 또한, 과거의 B_j $(1 \le j < t)$들이 공개되어있다 하더라도 이산 대수 문제의 어려움에 근거하여 B_j로부터 b_j $(1 \le j < t)$를 알아내는 것은 계산상 불가능하다. 따라서, 시간 구간 t의 서명키 x_t로부터 시간 구간 j $(1 \le j < t)$의 서명키 x_j를 유도해 낼 수 없기 때문에 정당한 서명자에 의해 과거에 생성된 서명은 보호된다.

4.2 효율성

- 효율적인 포워드-시큐어 서명 : 주요-파라미터인 대리 서명키, 대리 서명 검증키 및 대리 서명의 크기가 다음과 같이 대리 서명키 갱신 횟수 T에 독립적이다.

  대리 서명키

  $$x_t = s + x_P \cdot h(H^t(A), B_t, r) + b_t \pmod q$$

  대리 서명 검증키

  $$y_t = y_O^{h(m_w, r, E)} \cdot r \cdot y_P^{h(H^t(A), B_t, r)} \cdot B_t \pmod p$$

  서명 $\sigma = S(x_t, m)$

또한, 수신자의 대리 서명 검증시 해쉬 함수 계산을 제외하고는 대리 서명키 생성 및 갱신, 대리 서명 생성의 계산량이 대리 서명키 갱신 횟수 T에 독립적이며, 포워드-시큐어한 속성을 제공하기 위한 부가적인 저장 공간을 필요로 하지 않는다.

- 시간 정보 : 원 서명자는 위임 기간을 설정하고 그와 관련된 파라미터들을 공고한다. 그리고 위임 기간이 만료되면 공고했던 파라미터들을 삭제한다. 따라서, 대리 서명자와 수신자는 정확한 시간 정보를 생성할 필요없이 공고된 시간 관련 파라미터를 확인함으로써 위임 기간을 검증할 수 있다.

5. 프로토콜 비교 및 실험 결과 분석

5.1절에서는 기존 프로토콜과 TFPS의 효율성을 비교하고, 5.2절에서는 Sun의 스킴과[4] TFPS의 계산적인 효율성을 분석적 방법과 실험 결과를 통하여 비교한다.

5.1 기존 프로토콜과의 비교 분석

현재는 포워드-시큐어한 속성을 만족시키기 위해 설계된 대리 서명이 없으므로, 기존의 대리 서명 방식을 이용하여 서명 권한을 위임한 후 대리 서명키 생성 및 갱신, 서명 생성 및 검증시에 포워드-시큐어 서명 방식들을[5-7] 적용했을 때와 TFPS의 효율성을 비교한다. 그러나 대리 서명과 포워드-시큐어 서명을 별도로 수행하는 경우에, 수신자가 원 서명자의 위임 동의와 함께 대리 서명자의 정당한 서명임을 검증하기 위해서는 포워드-시큐어한 속성을 만족하는 서명키를 생성할 때 그 키에 위임 정보를 포함시키는 부가적인 프로토콜을 수행하거나, 대리 서명자가 포워드-시큐어 서명을 생성한 후에 위임 정보가 포함된 별도의 대리 서명키로 이중 서명하는 방법 등을 사용해야 한다.

포워드-시큐어 서명 스킴들은 서명키 생성, 서명키 갱신, 서명 생성, 서명 검증 단계로 구성되며, 대리 서명 스킴은 위임 정보 생성, 위임 정보 검증, 대리 서명키 생성, 대리 서명 생성, 대리 서명 검증 단계로 구성되므로, 이 단계별로 지수, 곱셈, 덧셈, 해쉬 함수의 수행 횟수를 분석하여 대표적인 연산을 중심으로 표 1에 나타내었다. 대리 서명은 보안 요구사항을[1,3] 만족하는 [3]의 스킴을 사용하였다.

표 1 포워드-시큐어 서명 스킴들의 효율성 비교

	포워드-시큐어 대리 서명	대리 서명	포워드-시큐어 서명		
	TFPS	[3]	[5]	[6]	[7]
주요 파라미터의 크기 (서명키, 검증키, 서명)	T에 독립적	-	T에 독립적	T에 독립적	T에 독립적
위임 정보 생성	지수 1 해쉬 $\theta(T)$	지수 1 해쉬 1	-	-	
위임 정보 검증	지수 2	지수 2	-	-	
서명키 생성	지수 1	덧셈 1	지수 l 곱셈 l	곱셈 $\theta(T)$	$KG()$ $\theta(T)$ $SIGN()$ $\theta(T)$
서명키 갱신	지수 1	-	곱셈 l	곱셈 $\theta(T)$	$KG()$ 1
서명 생성	$S()$ 1	$S()$ 1	지수 1 곱셈 $\theta(l)$	지수 2	$SIGN()$ 1
서명 검증	지수 2 $V()$ 1 해쉬 $\theta(T)$	지수 1 $V()$ 1	지수 1 곱셈 $\theta(l)$	지수 2	$VER()$ 2
부가적인 저장 공간	-	-	-	-	$\theta(T)$

※ [6]은 부가적인 저장 공간을 $\theta(\log T)$로 늘리는 대신, 서명키 생성과 갱신의 계산량을 $\theta(\log T)$로 줄일 수 있다.
※ [7]은 서명 생성의 계산량을 늘리고 서명의 크기를 $\theta(\log T)$로 하는 대신, 부가적인 저장 공간을 줄일 수 있다.

l : Fiat-Shamir 스킴의[13] 보안 파라미터
$KG()$: 임의의 서명키 생성 알고리즘
$SIGN()$: 임의의 서명 생성 알고리즘
$VER()$: 임의의 서명 검증 알고리즘

표 2 TFPS와 Sun의 스킴의 계산량 비교

단계		연산	TFPS	Sun의 스킴
1	원 서명자에 의한 위임 정보 생성	지수	1	1
		해쉬	$T+2$	1
		곱셈	1	2
		덧셈	1	1
2	대리 서명자에 의한 위임 정보 검증	지수	2	3
		해쉬	1	1
		곱셈	1	1
		덧셈		
3	대리 서명자에 의한 대리 서명키 생성 (및 갱신)	지수	1	
		해쉬	2	
		곱셈	1	1
		덧셈	2	1
4	대리 서명자에 의한 서명 생성	지수	1	2
		해쉬	1	2
		곱셈	1	1
		덧셈	1	1
		부가적인 작업		시간 정보 생성 및 생성된 연결 정보 공고
5	수신자에 의한 서명 검증	지수	4	5
		해쉬	3	2
		곱셈	4	3
		덧셈		
		시간 정보	해쉬 $T-t+1$	시간 정보 생성 및 검증
인증된 저장 공간			r, T, E	서명 생성 횟수에 비례하는 L_n

표 1에서 보는 바와 같이 [3]의 대리 서명 스킴은 TFPS에 비해 서명키 생성과 서명 검증시 지수 계산이 1번씩 더 적다. 그러나 포워드-시큐어한 속성을 만족시키기 위해서 [5-7]의 스킴을 이용하면 서명 생성시 부가적인 연산이 필요하고, 서명 검증시에도 지수 계산이 1번 이상 필요하다. 뿐만 아니라, 보증 정보에 위임 기간을 명시하는 방법으로는[3] 대리 서명자의 위임 기간과 포워드 시큐어 서명을 생성한 시간 구간 사이의 연관성을 지어 주지 못하기 때문에, 위임 권한의 유효 기간을 완전하게 반영하지 못한다.

5.2 Sun의 스킴과[4] 비교

Sun의 스킴은 서명에 시간 정보를 포함시킴으로써 대리 서명자가 위임 기간동안만 서명을 생성할 수 있도록 설계되었다. 이 대리 서명 방법은 포워드-시큐어한 속성을 제공하므로, 5.2.1에서는 TFPS와 Sun의 스킴의 효율성을 분석적으로 비교하고, 5.2.2에서는 두 프로토콜을 구현한 실험 결과를 분석한다.

5.2.1 효율성 분석

TFPS와 Sun의 스킴의 각 단계별 계산량은 표 2와 같다. TFPS에서 서명 알고리즘 $S()$, $V()$로 Schnorr의 서명 스킴을[14] 이용한다고 가정한다. 표 2에서 나타난 바와 같이 TFPS는 1, 3 및 5단계에서 해쉬 함수, 곱셈, 덧셈 계산이 많고, Sun의 스킴은 2, 4, 및 5단계에서 지수 계산이 많다. square-and-multiply 알고리즘을[15] 사용할 때 k승($k>1$) 지수 계산은 $\log k$번의 곱셈 계산량과 같으므로, TFPS가 효율적이라 할 수 있다.

5.2.2 실험 결과

Pentium Ⅳ, RAM 256MB에서 MS Visual C 6.0과 OpenSSL 암호 라이브러리를 사용하여 TFPS와 Sun의 스킴을 구현하였다. 프로토콜에 사용되는 해쉬 함수는 MD5를 사용하였다. 대리 서명자의 서명 생성 횟수를 n이라 하고, TFPS의 대리 서명키 갱신 횟수를 T라고 할 때, 각 단계별로 1000번 이상 수행하여 측정한 평균 수행 시간은 표 3과 같다. Sun의 스킴의 수행 시간은

표 3 n번 서명시 수행 시간

(단위 : msec)

	TFPS	Sun의 스킴
시스템 파라미터 설정	779	
원 서명자에 의한 위임 정보 생성	17 + 0.014 T	18
대리 서명자에 의한 위임 정보 검증	10	26
대리 서명자에 의한 대리 서명키 생성	16 T	0.007 n
대리 서명자에 의한 서명 생성	15 n	35 n
수신자에 의한 서명 검증	19n + 0.007 nT	48 n
전체 시간	806 + 16.014 T + 34 n + 0.007 nT	823 + 83.007 n

n의 함수로 나타나고, TFPS는 원 서명자가 서명키 갱신 횟수 T를 조절할 수 있으므로 n과 T의 함수로 나타난다.

Sun의 스킴에서는 매 서명마다 연결 정보 L_n을 공고하여 포워드-시큐어하게 만들기 때문에, TFPS에서 매 서명마다 서명키를 갱신하는 것과 같은 효과를 낳는다. TFPS에서 매 서명마다 새로운 대리 서명키를 사용하도록 $T=n$으로 설정하고 n=1000, 2000, 3000, 4000, 5000일 때의 수행 시간을 Sun의 스킴과 비교하면 그림 4와 같다. Sun의 스킴은 지수 계산이 많고 TFPS는 해쉬 계산이 많기 때문에 TFPS가 더 효율적이지만, TFPS의 1과 5단계의 해쉬 함수 계산은 시간 구간의 수 T에 비례하므로 해쉬 함수의 계산량이 지수 계산에 비해 미비하다고 하더라도 T에 따라 계산량이 누적된다. 그림 4에 나타난 바와 같이 $T=n<5000$인 경우에는 TFPS가 효율적이고 $T=n\geq5000$이 되면 해쉬 함수 계산량의 누적으로 TFPS의 효율성이 저하됨을 볼 수 있다.

그러나 TFPS에서는 원 서명자가 대리 서명키 갱신 횟수 T를 설정함으로써 보안 레벨과 효율성의 상충관

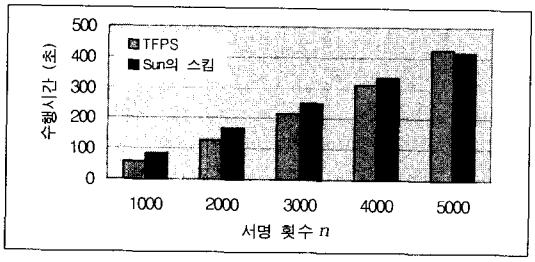


그림 4 서명 횟수에 따른 수행 시간

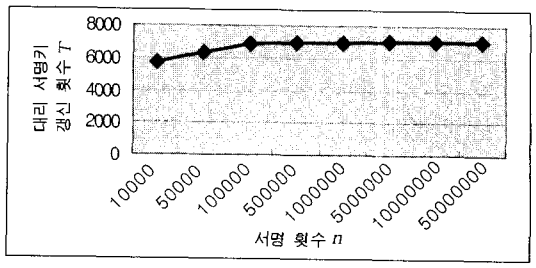


그림 5 Sun의 스킴보다 효율적인 T의 최대값

계를 조절할 수 있다. 그림 5는 $n\geq5000$일 경우에도 TFPS가 효율적인 $T(\leq7000)$의 값을 보여준다. 따라서, TFPS는 $n<5000$이거나 $T\leq7000$인 경우에 더 효율적인 것으로 분석된다.

6. 결 론

대리 서명은 대리 서명자가 원 서명자의 권한을 위임받아 대리 수행하는 것으로서, 일반적으로 그 위임은 유효 기간을 갖기 때문에 위임 기간 만료 이후의 대리 서명자의 서명 위조 방지는 중요한 문제이다. 위임 기간 만료 후의 대리 서명자의 서명 위조를 막기 위하여 위임 기간을 위임 정보에 명시할 수 있지만, 대리 서명자의 서명 생성 시간을 알 수 없기 때문에 완전한 해결책이 되지 못한다. 또한, 일반 서명에서와 마찬가지로 대리 서명에서도, 서명키 노출로 인한 과거 서명에 대한 타당성 시비는 해결해야 할 문제이다. 그러나 대리 서명에서는 이 문제를 해결하기 위한 연구가 시도되지 않고 있다.

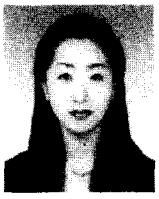
본 논문에서는 위임 기간이 만료된 대리 서명자의 서명 위조를 막고, 서명키 노출 문제를 해결하기 위한 포워드-시큐어 서명을 생성하는 방법을 제안하였다. 본 논문에서 제안한 TFPS는 원 서명자가 시간 관련 파라미터를 공고하여 제어하므로 대리 서명자와 수신자가 정확한 시간 정보를 필요로 하지 않는다. 또한, 어플리케이션 특성이나 정책에 따라 원 서명자가 서명키 갱신 횟수를 조절하고 시간 구간별로 포워드-시큐어한 속성을 만족시킴으로써 효율성을 높일 수 있다.

참 고 문 헌

[1] M. Mambo, K. Usuda and E. Okamoto, "Proxy Signatures: Delegation of the Power to Sign Message," IEICE Trans. Fundamentals, Vol. E79 A, No. 9, 1996.

[2] S. Kim, S. Park and D. Won, "Proxy Signatures, Revisited," Proc. of ICICS 97, 1997.

[3] B. Lee, H. Kim and K. Kim, "Strong Proxy Signature and its Applications," Proc. of SCIS 2001, 2001.
[4] H. M. Sun, "Design of Time Stamped Proxy Signature with Traceable Receivers," Proc. of IEE Computers and Digital Techniques, Vol. 147, No. 6, 2000.
[5] M. Bellare and S. Miner, "A Forward Secure Digital Signature Scheme," Crypto'99, 1999.
[6] G. Itkis and L. Reyzin, "Forward Secure Signatures with Optimal Signing and Verifying," Crypto'01, 2001.
[7] H. Krawczyk, "Simple Forward Secure Signatures from any Signature Scheme," 7th ACM Conference on Computer and Communication Security, 2000.
[8] T. Malkin, D. Micciancio, and S. Miner, "Efficient Generic Forward-Secure Signatures with an Unbounded Number of Time Periods," Eurocrypt'02, 2002.
[9] Y. Dodis, J. Katz, S. Xu and M. Yung, "Key Insulated Public Key Cryptosystems," Eurocrypt'02, 2002.
[10] G. Itkis and L. Reyzin, "SiBIR: Signer Base Intrusion-Resilient Signatures," Crypto'02, 2002.
[11] S. Haber and W. Stornetta, "How to Time Stamp a Digital Document," Proc. of Crypto'90, Vol. 537, 1990.
[12] A. Buldas, P. Laud, H. Lipmaa and J. Villemson, "Time Stamping with Binary Linking Schemes," Proc. of Crypto'98, 1998.
[13] A. Fiat and A. Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems," Proc. of Crypto'86, Vol. 263, 1986.
[14] C. P. Schnorr, "Efficient Signature Generation by Smart Cards," Journal of Cryptology, Vol. 4, 1991.
[15] A. J. Menezes, P. C. Oorschot and S. A. Vanstone, Handbook of Applied Cryptography, CRC, 1997.



김 상 희
2001년 이화여자대학교 컴퓨터학과 학사 2003년 이화여자대학교 과학기술대학원 컴퓨터학과 석사. 관심분야는 정보보호 암호 프로토콜, 네트워크 보안

조 태 남
1986년 이화여자대학교 전자계산학과 학사. 1988년 이화여자대학교 대학원 전자계산학과 석사. 1988년~1996년 한국전자통신연구원 선임연구원. 1998년~현재 이화여자대학교 과학기술대학원 컴퓨터학과 박사과정. 관심분야는 정보보호, 알고리즘 설계, 네트워크 보안

이 상 호
1979년 서울대학교 계산통계학과 학사 1981년 한국과학기술원 전산학과 석사 1987년 한국과학기술원 전산학과 박사 1983년~현재 이화여자대학교 컴퓨터학과 교수. 관심분야는 알고리즘 설계, 정보보호, 바이오인포매틱스

채 기 준
1982년 연세대학교 수학과 이학사. 1984년 미국 Syracuse University 컴퓨터학과 이학석사. 1990년 미국 North Carolina State University 컴퓨터공학과 공학박사. 1990년~1992년 미군 해군사관학교 컴퓨터학과 조교수. 1992년~현재 이화여자대학교 컴퓨터학과 교수. 관심분야는 네트워크 보안, 액티브 네트워크 보안 및 관리, 인터넷/무선통신망/고속통신망 프로토콜 설계 및 성능분석

박 원 주
1998년 충남대학교 정보통신공학과 학사 2000년 충남대학교 대학원 정보통신공학과 석사. 2000년~현재 한국전자통신연구원 연구원. 관심분야는 IPsec, IPv6, VPN, 멀티캐스팅, 네트워크 보안

나 재 훈
1985년 중앙대학교 컴퓨터공학과 학사 1987년 중앙대학교 대학원 컴퓨터공학과 석사. 1987년~현재 한국전자통신연구원 책임연구원. 관심분야는 IPsec, Mobile IP, IPv6, 네트워크 보안