RSA 기반의 Designated Confirmer Undeniable 전자 서명

박주환, 염대현, 이필중

포항공과대학교 전자전기공학과

{jhpark, daehyun}@oberon.postech.ac.kr, pjl@postech.ac.kr

Designated Confirmer Undeniable Signatures Based on RSA

Ju Hwan Park, Dae Hyun Yum, Pil Joong Lee

Dept. of Electronic and Electrical Eng., POSTECH, Korea

{jhpark, daehyun}@oberon.postech.ac.kr, pjl@postech.ac.kr

요 약

D. Chaum이 제안한 Undeniable Signatures[1]을 시작으로 undeniable signature에 대한 많은 scheme이 제안되었다. 일반적인 전자 서명에서는 누구나 검증을 할 수 있는데 비해, undeniable signature를 검증하기 위해서는 서명자의 협동이 필수적이다. 한편, confirmer를 따로 지정해서 confirmer에게 서명 검증 능력을 부여할 수 있는 Designated Confirmer Signatures[4]가 제안되었다. 본 논문에서는 T. Miyazaki가 제안한 RSA-Based Convertible Undeniable Signature Scheme[6]을 변형하여, Designated Confirmer Undeniable Signature System Based on RSA를 제안하고, 제안된 프로토콜을 분석한다.

I. 서론

오늘날 공개키 시스템은 암호화뿐만 아니라, 전자서명을 가능하게 해 줌으로써 아주 중요한 역할을 하고 있다. 그 중 전자서명은 전자적 문서의 무결성, 서명자 인증, 부인 방지 등의 중요한 암호학적 서비스를 제공한다. 일반적인 전자서명에서는 어느 누구라도 서명자의 공개키를 이용하여 서명값을 검증함으로써 서명자의 서명 여부를 확인할 수가 있다(self-authenticating property or universal verifiability). 하지만, 이러한 특징이 항상 바람직한 것은 아니다. 어떤 소프트웨어 공급업체에서 제품을 출시하고, 돈을 지불한 고객만이 그 소프트웨어가 진짜이고, 변경되지 않았다는 것을 확인할 수 있도록 허락하고 싶을 경우가 있다고 가정하자. 이런 경우에 undeniable signature[1, 2, 3, 5, 6, 8]을 이용하면 오직 돈을 지불한 고객만이 서명값을 검증하여, 공급업체가 그 소프트웨어에 대한 책임을 지고 있다는 확신을 할 수 있게 된다. D. Chaum의 scheme[1]을 시작으로, undeniable signature에 관계된 scheme들이 많이 제안되었다. 1990년에는 서명자가 갖고 있는 검증에 필요한 정보를 공개하면, 일반적인 전자서명으로 변환되어 누구든지 서명을 검증할 수 있게 되는 Convertible Undeniable Signatures[3]가 J. Boyar에 의해 제안되었다. 그 후, D. Chaum은 서명자가 confirmer에게 서명 검증 능력을 부여하는 Designated Confirmer Signatures[4]를 제안하였다. 이산 대수 문제의 어려움(Discrete Logarithm Problem)에 기반한 위의 scheme들에 이어서, 소인수 분해 문제의 어려움(Integer Factorization Problem)에 기반한 undeniable signature가 R. Gennaro, H. Krawczyk와 T. Rabin에 의해 제안되었다[5]. T. Miyazaki는 R. Gennaro의 scheme이 선택적으로(selectively) convertible하지 못하고, hidden verifier attack[7]에 대해 취약하며, denial 프로토콜이 결정적이지 못한(undeterministic) 점을 지적하면서 좀 더 향상된 RSA-Based Convertible Undeniable Signatures[6]을 제안하였다. 본 논문에서는 T. Miyazaki의 scheme[6]을 바탕으로, RSA-Based Designated Confirmer Undeniable Signature

Scheme을 제안하고, 프로토콜을 분석한다. 논문의 구성은 다음과 같다. 2장에서는 T. Miyazaki의 RSA-Based Convertible Undeniable Signature Scheme[6]에 대하여 설명하고, 3장에서 RSA-Based Designated Confirmer Undeniable Scheme을 제안한다. 그리고, 4장에서 제안된 scheme을 분석하고, 5장에서 결론을 맺는다.

II. RSA-Based Convertible Undeniable Signature[6]

1. Key Generating System

우선, 시스템이 다음과 같이 법(modulus)의 집합을 생성한다.

$$N = \{ n | n = pq,\ p < q,\ p = 2p' + 1, q = 2q' + 1 \}$$

(p, q, p', q' : large primes)

각각의 사용자는 다음의 매개변수(parameter)를 생성한다.

① $n \in N$인 n을 선택

② $L = LCM(p-1, q-1) = 2p'q'$을 계산

③ 홀수 $e, d_2 (3 \le e, d_2 \le L-1)$을 선택

④ $d_1 = (ed_2)^{-1} \pmod{L}$을 계산

⑤ $d = d_1 d_2 \pmod{L}$을 계산

⑥ $w \in Z_n^*$ $(w \neq 1)$를 선택하고,
$S_w = w^{ed_1} \bmod n$을 계산

사용자의 공개키는 (e, n, w, S_w)가 되고, 이 값들은 공개한다. 그리고, 비공개키 (d_1, d_2, d)는 비밀리에 보관한다. 여기서, 공개키 e와 비공개키 d의 관계는 일반적인 RSA에서의 키 쌍과 동일하다. 키 생성과정에서 단지 역원 계산이 일반적인 RSA의 키 생성과정보다 더 필요하므로, 비용(cost)면에서 큰 차이는 없다.

$$\forall M \in Z_n ;\ M^{ed} \equiv M^{ed_1 d_2} \equiv M \pmod{n}$$

2. 일반적인 RSA Signature의 생성

1) Signature의 생성

$$S = m^d \pmod{n}$$

2) Signature의 검증

$$m = S^e \pmod{n}$$

3. Undeniable Signature의 생성

메시지 m에 해당하는 undeniable signature S_m을 생성하기 위해서 서명자는 공개키 e와 비공개키 d_1을 이용해서 다음과 같이 계산한다.

$$S_m = m^{ed_1} \pmod{n} \qquad (1)$$

4. Signature Confirmation

서명자를 P, 검증자를 V라고 하자. P는 II.1절과 같은 매개변수를 갖고 있어야 하나, V는 그런 매개변수를 생성할 필요는 없다. V는 단지 p_v와 q_v의 곱인 n_v를 계산해야 하며, 이 값을 공개하도록 한다. P와 V가 보유한 매개변수는 [표 1]과 같다.

	Public Information	Private Information
P	n_p, e_p, w_p, S_{wp}	$p_p, q_p, d_p, d_{1p}, d_{2p}$
V	$n_v, (e_v, w_v, S_{wv})$	$p_v, q_v, (d_v, d_{1v}, d_{2v})$

[표 1] P와 V의 매개변수

여기서 ()의 매개변수는 이 scheme에서는 필요가 없는 것들이다. V는 이미 메세지 m과 수신한 서명값 $\widehat{S_m}$의 쌍 $(m, \widehat{S_m})$을 갖고 있다. P가 [표 1]에 제시된 매개변수를 이용하여, 메시지 m에 서명한 유효(valid)한 서명값 S_m은 식(1)에 의해 다음과 같이 계산된다.

$$S_m = m^{e_p d_{1v}} \pmod{n_p} \qquad (2)$$

V가 검증을 요청하는 서명값 $\widehat{S_m}$이 P에 의해 메세지 m에 서명된 유효한 값인지는 [그림 1]의 Signature Confirmation 프로토콜을 통해 증명할 수 있다. 이 프로토콜은 다음의 식을 사용하여 서명을 검증한다.

P : Prover	V : Verifier
	1.임의의 $i,j(1\le i,j\le n)$를 선택하여 Q를 계산 $Q \overset{\triangle}{\equiv} \widehat{S_m}^{i} S_{wp}{}^{j} \pmod{n_p}$
	$\Leftarrow Q$
2. Q로부터 A를 계산 $a \overset{\triangle}{\equiv} Q^{d_{2p}} \pmod{n_p}$ $A=\begin{cases} a & (\text{if } a \text{ is odd}) \\ n_p-a & (\text{if } a \text{ is even}) \end{cases}$	
3. $2\le x\le(n_v-2)$인 x를 선택하고 C를 계산 $C \overset{\triangle}{\equiv} x^{A} \pmod{n_v}$	
	$C \Rightarrow$
	$\Leftarrow i,\ j$
4. $Q \neq \widehat{S_m}^{i} S_{wp}{}^{j} \pmod{n_p}$이면, Q, i, j는 올바르지 않은 값이고, 프로토콜을 멈춤	
	$A,\ x \Rightarrow$
	5. $C \overset{?}{\equiv} x^{A} \pmod{n_v}$ $A^2 \overset{?}{\equiv} (m^{i} w_p{}^{j})^2 \pmod{n_p}$ 두 등식이 모두 성립하면, V는 $\widehat{S_m}$을 P가 m에 서명한 유효한 서명값으로 받아들임

[그림 1] Signature Confirmation 프로토콜

$$(M^{e_p d_{1v}})^{d_{2v}} \equiv M^{e_p d_{1v} d_{2v}} \equiv M^{e_p d_p} \equiv M \pmod{n_p}$$

만약 V가 갖고 있는 서명값 $\widehat{S_m}$이 유효한 값이면, 단계 2에서 P는 다음과 같이 a값을 계산해 낸다.

$$\begin{aligned} a &\equiv Q^{d_{2p}} \pmod{n_p} \\ &\equiv \widehat{S_m}^{id_{2p}} S_{wp}{}^{jd_{2p}} \pmod{n_p} \\ &\equiv m^{ie_p d_{1v} d_{2p}} w_p^{je_p d_{1v} d_{2p}} \pmod{n_p} \\ &\equiv m^{i} w_p^{j} \pmod{n_p} \end{aligned}$$

[그림 1]의 단계 2에서 P가 A의 값을 홀수로 만들기 위해 $\pm a \pmod{n}$로 하는 이유는 mod L_v 연산에서 역원을 존재하도록 함으로써, resistance protocol[6]이 존재한다는 것을 보장하기 위해서이다. 이는 결국 hidden verifier attack[7]을 피하기 위함이다. 단계 4와 단계 5의 과정은 각기 V와 P가 단계 1과 단계 3에서 생성한 값을 올바르게 전송하고 있는지를 확인하기 위한 과정이다.

5. Denial of Signature

서명자는 V가 검증을 요청하는 서명값이 유효하지 않은(invalid) 값임을 Denial 프로토콜을 통해서 증명할 수 있다.

$\{i_1, j_1, Q_1, a_1, A_1, x_1, C_1\}$을 사용한 confirmation 프로토콜
V는 이 매개변수들로써 서명값이 유효한지 여부를 검사
1-1. $C_1 \overset{?}{\neq} x_1^{A_1} \pmod{n_v}$이면, A_1, x_1은 옳지 않은 값이고 멈춤
1-2. $A_1{}^2 \overset{?}{=} (m^{i_1} w_p{}^{j_1})^2 \pmod{n_p}$이면, 서명값은 유효하고 멈춤
$\{i_2, j_2, Q_2, a_2, A_2, x_2, C_2\}$을 사용한 confirmation 프로토콜
V는 이 매개변수들로써 서명값이 유효한지 여부를 검사
2-1. $C_2 \overset{?}{\neq} x_2^{A_2} \pmod{n_v}$이면, A_2, x_2은 옳지 않은 값이고 멈춤
2-2. $A_2{}^2 \overset{?}{=} (m^{i_2} w_p{}^{j_2})^2 \pmod{n_p}$이면, 서명값은 유효하고 멈춤
V는 confirmation 프로토콜이 올바르게 수행되었는지 검사
3. $(A_1 w_p{}^{-j_1})^{2i_2} \overset{?}{=} (A_2 w_p{}^{-j_2})^{2i_1} \pmod{n_p}$이면 서명값은 유효하지 않음

[그림 2] Signature Denial 프로토콜

메시지 m에 대해서 유효하지 않은 서명을 $\widehat{S_m}$이라고 하자($\widehat{S_m} \neq m^{ed_1} \pmod{n}$). P는 이 프로토콜을 수행함으로써, V가 검증을 요청한 서명값 $\widehat{S_m}$이 P에 의해서 메시지 m에 서명된 유효한 값이 아님을 보여줄 수 있다. 따라서, P가 이 프로토콜을 수행할 수 없다면, V가 제시한 서명값이 유효하지 않은 값임을 증명할 수 없게 된다. 이 프로토콜은 세 부분으로 구성되는데, 그 첫 번째 부분과 두 번째 부분에서 P와 V는 각각 임의의 값들을 생성해서 [그림 1]의 Signature Confirmation 프로토콜을 두 차례 수행하게 된다. 그리고, 마지막 부분에서 V는 P가 Signature Confirmation 프로토콜을 올바르게 수행하였는지를 검사하게 된다. 여기서, Signature Confirmation 프로토콜은 서명값이 유효한 서명값이 아님을 확인하기 위해서 사용한다. 따라서 첫 번째 부분과 두 번째 부분의 Signature Confirmation 프로토콜에서 서명값이 유효하다고 판정되면 프로토콜을 종료해야 한다. 이 프로토콜이 올바르게 수행되면 단계 3에서의 등식은 다음과 같이 성립한다.

$$(A_1 w_p{}^{-j_1})^{2i_2} \equiv (\pm \widehat{S_m}^{d_{2p}i_1} w_p{}^{j_1} w_p{}^{-j_1})^{2i_2} \pmod{n_p}$$

$$\equiv \widehat{S_m}^{2d_2 i_1 i_2} \pmod{n_p}$$

$$(A_2\, w_p^{-i_2})^{2i_1} \equiv (\pm\ \widehat{S_m}^{d_2 i_2}\, w_p^{i_2}\, w_p^{-i_2})^{2i_1} \pmod{n_p}$$

$$\equiv \widehat{S_m}^{2d_2 i_1 i_2} \pmod{n_p}$$

III. RSA-Based Designated Confirmer Undeniable Signatures

[6]에서는 서명자의 참여 없이는 검증이 불가능하다. 따라서, 서명자가 어떠한 이유에서 검증자의 검증요구에 응답할 수 없을 경우에는 그 어느 누구라도 서명값을 검증할 수 없게 된다. 이러한 분제점을 해결하기 위한 방법으로, 서명자가 confirmer를 지정(designation)하여 서명 검증에 필요한 정보를 제공함으로써, confirmer 역시 서명의 유효성을 검증할 수 있도록 하는 Designated Confirmer Signatures[4]가 제안되었다. Confirmer는 서명을 confirmation할 수는 있으나, confirmation에 사용한 비밀정보를 이용해서 서명자의 신분으로 서명을 할 수는 없다(unforgeable). 본 장에서는 [6]을 변형하여 RSA기반에서의 Designated Confirmer Undeniable Signature Scheme을 제안한다. 이는, RSA 서명을 사용하는 표준화된 여러 통신 프로토콜에 바로 적용할 수 있는 장점이 있다. 편의상 서명자를 P, 검증자를 V, confirmer를 C라고 한다.

1. Key Generating System

II.1절과 동일

2. 일반적인 RSA Signature의 생성

II.2절과 동일

3. Undeniable Signature의 생성

Designated confirmer undeniable signature는 아래와 같이 생성한다.

① 임의의 $r(3 \le r \le L-1)$을 선택

② $K=(ed_1)\cdot r \pmod{L}$를 계산

③ 메시지 m과 K를 연접(concatenation)한 값의 *Hash*함수[9] 출력을 구함

④ $S_1 = H(m\|K)^{ed_1} \pmod{n_p}$을 계산

서명값 S_m을 다음과 같이 생성하여 V에게 보내준다.

$$S_m = (S_1, K) \tag{3}$$

4. Confirmer의 지정(designation)

P는 r을 안전한 전송 채널(secure channel)을 통하여 C에게 전송함으로써, P대신 C가 서명 검증 능력을 갖도록 한다. P로부터 r을 받은 C는 r이 서명 검증에 필요한 올바른 값인지를 검사하기 위하여, V로부터 K를 받아 아래와 같이 비교한다.

$$S_w^r \stackrel{?}{=} w^K \pmod{n_p} \tag{4}$$

위의 식이 등식을 만족하지 않으면, P가 전송한 r이 V가 제시할 서명을 검증해 줄 수 있는 올바른 값이 아니므로, P가 유효하지 않은 r을 전송하였음을 C는 알 수 있다. 식(4)가 등식을 만족할 경우에 C는 [그림 3]의 Designation의 유효성 검증 프로토콜을 수행함으로써, 서명을 검증해 줄 수 있는 유효한 r을 C가 갖고 있음을 V에게 증명할 수 있다.

C : Confirmer	V : Verifier
	1. 임의의 $i(1 \le i \le n)$를 선택하여 C를 계산 $C \stackrel{\Delta}{=} S_w^i \pmod{n_p}$
	$\Leftarrow C$
2. $R \stackrel{\Delta}{=} C^r \pmod{n_p}$ 을 계산	
	$R \Rightarrow$
	3. $R \stackrel{?}{=} w^{iK} \pmod{n_p}$ 등식이 성립하면 서명값 $\widehat{S_m}$을 검증할 수 있는 유효한 r을 C가 갖고 있음이 확인됨.

[그림 5] Designation의 유효성 검증 프로토콜

5. 서명자에 의한 Signature Confirmation

P가 [표 1]에 제시된 매개변수를 가지고, 메시지 m에 서명한 서명값 S_m은 다음과 같다.

$$K=(e_p d_{1p})\cdot r \pmod{L} \quad (5)$$

$$S_1=H(m\|K)^{e_p d_{1p}} \pmod{n_p} \quad (6)$$

$$S_m=(S_1, K) \quad (7)$$

서명값 S_m이 P가 메세지 m에 서명한 유효한 값인지는 [그림 4]의 서명자에 의한 Signature Confirmation 프로토콜을 통해 증명한다. 이 프로토콜은 다음의 식을 사용한다.

$$(H(M\|K)^{e_p d_{1p}})^{d_{2p}} \equiv H(M\|K)^{e_p d_p} \equiv H(M\|K) \pmod{n_p}$$

S : Signer	V : Verifier
	1.임의의 $i, j (1\le i, j\le n)$를 선택하여 Q를 계산 $Q \triangleq \widehat{S_1}^{i} S_{wp}^{j} \pmod{n_p}$
	$\Leftarrow Q$
2. A를 계산 $a \triangleq Q^{d_{2p}} \pmod{n_p}$ $A=\begin{cases} a & (\text{if } a \text{ is odd}) \\ n_p - a & (\text{if } a \text{ is even}) \end{cases}$	
3. $2\le x\le (n_v-2)$인 x를 선택하고 C를 계산 $C \triangleq x^A \pmod{n_v}$	
	$C \Rightarrow$
	$\Leftarrow i, j$
4. $Q \ne \widehat{S_1}^{i} S_{wp}^{j} \pmod{n_p}$이면 Q, i, j는 올바르지 않은 값이고, 프로토콜을 멈춤	
	$A, x \Rightarrow$
	5. $C \stackrel{?}{=} x^A \pmod{n_v}$, $A^2 \stackrel{?}{=} (H(m\|K)^{i} w_p^{j})^2 \pmod{n_p}$ 두 등식이 모두 성립하면, V는 $\widehat{S_m}$을 P가 m에 서명한 유효한 서명값으로 받아들임

[그림 4] 서명자에 의한 Confirmation 프로토콜

만약 수신한 서명값 $\widehat{S_m}$이 유효하면, 단계 2에서 P는 다음과 같이 a값을 계산해 낸다.

$$a \equiv Q^{d_{2p}} \pmod{n_p}$$

$$\equiv \widehat{S_1}^{id_{2p}} S_{wp}^{jd_{2p}} \pmod{n_p}$$

$$\equiv H(m\|K)^{ie_p d_{1p} d_{2p}} w_p^{je_p d_{1p} d_{2p}} \pmod{n_p}$$

$$\equiv H(m\|K)^{i} w_p^{j} \pmod{n_p}$$

6. 서명자에 의한 Denial of Signature

서명자에 의한 Denial 프로토콜은 V가 제시한 서명값이 유효하지 않은 값임을 P가 증명하고자 할 때 사용된다. 이 프로토콜이 올바르게 수행되면 단계 3에서의 등식은 다음과 같이 성립한다.

$\{i_1, j_1, Q_1, a_1, A_1, x_1, C_1\}$을 사용한 confirmation 프로토콜
V는 이 매개변수들로써 서명값이 유효한지 여부를 검사
1-1. $C_1 \stackrel{?}{\ne} x_1^{A_1} \pmod{n_v}$이면, A_1, x_1은 옳지 않은 값이고 멈춤
1-2. $A_1^2 \stackrel{?}{=} (H(m\|K)^{i_1} w_p^{j_1})^2 \pmod{n_p}$이면, 서명값은 유효하고 멈춤
$\{i_2, j_2, Q_2, a_2, A_2, x_2, C_2\}$을 사용한 confirmation 프로토콜
V는 이 매개변수들로써 서명값이 유효한지 여부를 검사
2-1. $C_2 \stackrel{?}{\ne} x_2^{A_2} \pmod{n_v}$이면, A_2, x_2은 옳지 않은 값이고 멈춤
2-2 $A_2^2 \stackrel{?}{=} (H(m\|K)^{i_2} w_p^{j_2})^2 \pmod{n_p}$이면, 서명값은 유효하고 멈춤
V는 confirmation 프로토콜이 올바르게 수행되었는지 검사
3. $(A_1 w_p^{-j_1})^{2i_2} \stackrel{?}{=} (A_2 w_p^{-j_2})^{2i_1} \pmod{n_p}$이면, 서명값은 유효하지 않음

[그림 5] 서명자에 의한 Signature Denial 프로토콜

$$(A_1 w_p^{-j_1})^{2i_2} \equiv (\pm \widehat{S_1}^{d_{2p} j_1} w_p^{j_1} w_p^{-j_1})^{2i_2} \pmod{n_p}$$

$$\equiv \widehat{S_1}^{2d_{2p} j_1 i_2} \pmod{n_p}$$

$$(A_2 w_p^{-j_2})^{2i_1} \equiv (\pm \widehat{S_1}^{d_{2p} j_2} w_p^{j_2} w_p^{-j_2})^{2i_1} \pmod{n_p}$$

$$\equiv \widehat{S_1}^{2d_{2p} j_1 i_2} \pmod{n_p}$$

7. Confirmer에 의한 Signature Confirmation

P가 V의 서명 검증 요구에 응답할 수 없을 때, V는 C와 [그림 6]의 Confirmer에 의한 Confirmation 프로토콜을 수행함으로써 서명을 검증할 수 있다. 프로토콜을 수행하여 단계 5의 등식을 만족하면, V가 제시한 서명값 S_m이 유효한 서명값임을 알 수 있다. 이 프로토콜은 다음의 식을 사용한다.

$$(H(M\|K)^{e_p d_{1p}})^{r_p} \equiv H(M\|K)^{e_p d_{1p} r_p} \equiv H(M\|K)^{K} \pmod{n_p}$$

C : Confirmer	V : Verifier
	1.임의의 $i, j(1 \le i, j \le n)$를 선택하여 Q를 계산 $Q \triangleq \hat{S}_1^{\,i} S_{wp}^{j} \pmod{n_p}$
	$\Leftarrow Q$
2. A를 계산 $a \triangleq Q^r \pmod{n_p}$ $A = \begin{cases} a & (\text{if } a \text{ is odd}) \\ n_p - a & (\text{if } a \text{ is even}) \end{cases}$ 3. $2 \le x \le (n_v - 2)$인 x를 선택하고 C를 계산 $C \triangleq x^A \pmod{n_v}$	
	$C \Rightarrow$
	$\Leftarrow i, j$
4. $Q \neq \hat{S}_1^{\,i} S_{wp}^{j} \pmod{n_p}$이면 Q, i, j는 올바르지 않은 값이고, 프로토콜을 멈춤	
	$A, x \Rightarrow$
	5. $C \stackrel{?}{=} x^A \pmod{n_v}$ $A^2 \stackrel{?}{=} (H(m\|\|K)^i w_p^j)^{2K} \pmod{n_p}$ 두 등식이 모두 성립하면, V는 $\hat{S}_m$을 P가 m에 서명한 유효한 서명값으로 받아들임

[그림 6] Confirmer에 의한 Confirmation 프로토콜

만약 수신한 서명값 $\hat{S}_m$이 유효하면, 단계 2에서 C는 다음과 같이 a값을 계산해 낸다.

$$a \equiv Q^r \pmod{n_p}$$
$$\equiv \hat{S}_1^{\,ir} S_{wp}^{\,jr} \pmod{n_p}$$
$$\equiv H(m\|K)^{ie_1d_1r} w_p^{je_1d_1r} \pmod{n_p}$$
$$\equiv (H(m\|K)^i w_p^j)^K \pmod{n_p}$$

8. Confirmer에 의한 Denial of Signature

Confirmer에 의한 Denial 프로토콜은 V가 제시한 서명값이 유효하지 않은 값임을 C가 증명하고자 할 때 사용된다. 이 프로토콜이 올바르게 수행되면 단계 3에서의 등식은 다음과 같이 성립한다.

$$(A_1 w_p^{-j_1K})^{2i_2} \equiv (\pm \hat{S}_1^{\,i_1r} S_{wp}^{\,j_1r} w_p^{-i_1K})^{2i_2} \pmod{n_p}$$
$$\equiv \hat{S}_1^{\,2i_1i_2r} \pmod{n_p}$$

$$(A_2 w_p^{-j_2K})^{2i_1} \equiv (\pm \hat{S}_1^{\,i_2r} S_{wp}^{\,j_2r} w_p^{-i_2K})^{2i_1} \pmod{n_p}$$
$$\equiv \hat{S}_1^{\,2i_1i_2r} \pmod{n_p}$$

$\{i_1, j_1, Q_1, a_1, A_1, x_1, C_1\}$을 사용한 confirmation 프로토콜
V는 이 매개변수들로써 서명값이 유효한지 여부를 검사
1-1. $C_1 \stackrel{?}{\neq} x_1^{A_1} \pmod{n_v}$이면, A_1, x_1은 옳지 않은 값이고 멈춤
1-2. $A_1^2 \stackrel{?}{=} (H(m\|K)^{i_1} w_p^{j_1})^{2K} \pmod{n_p}$이면, 서명값은 유효하고 멈춤
$\{i_2, j_2, Q_2, a_2, A_2, x_2, C_2\}$을 사용한 confirmation 프로토콜
V는 이 매개변수들로써 서명값이 유효한지 여부를 검사
2-1. $C_2 \stackrel{?}{\neq} x_2^{A_2} \pmod{n_v}$이면, A_2, x_2은 옳지 않은 값이고 멈춤
2-2. $A_2^2 \stackrel{?}{=} (H(m\|K)^{i_2} w_p^{j_2})^{2K} \pmod{n_p}$이면, 서명값은 유효하고 멈춤
V는 confirmation 프로토콜이 올바르게 수행되었는지 검사
3. $(A_1 w_p^{-j_1K})^{2i_2} \stackrel{?}{=} (A_2 w_p^{-j_2K})^{2i_1} \pmod{n_p}$이면, 서명값은 유효하지 않음

[그림 7] Confirmer에 의한 Denial 프로토콜

IV. 분석

본 논문에서 제시된 프로토콜 중 S와 V간에 이루어지는 프로토콜은 T. Miyazaki[6]의 논문에 제시된 프로토콜과 비교해서 다음 한 가지만 다를 뿐 모든 점이 동일하다. 즉, 메시지와 K를 연접(concatenation)한 값에 해쉬 함수[9]를 적용시켜 서명한다는 점이다. 해쉬 함수의 적용은 security 측면에서 아무런 나쁜 영향을 미치지 않기에, S와 V가 수행하는 프로토콜은 [6]에서 제안된 프로토콜과 동일한 security level을 갖는다. 따라서, 여기서는 C를 지정하여 검증에 필요한 정보를 제공함으로써 발생할 수 있는 문제점을 살펴본다.

1. Unforgeability

서명을 위조할 수 있는 가장 위협적인 공격자는 C이다. 하지만, C는 r과 K를 P로부터 전송 받아 알고 있지만, 식(5)의 계산에 필요한 L값을 알 수 없다. r과 K만으로는 ed_1값을 구해낼 수가 없으므로, C는 식(6)과 같이 서명자의 서명을 만들어 낼 수 없다.

2. 서명자의 dishonesty

P가 C를 속이기 위해서 검증에 적합하지 못한 r값을 C에게 전송할 경우, C는 designation과정 중에 식(4)를 계산해서 비교해 봄으로써, P의 거짓 여부를 확인할 수 있다.

3. Confirmer의 dishonesty

C가 V를 속이는 과정은 [6]에서 P가 V를 속이는 과정과 동일하다. 따라서, C가 서명 검증 과정에 유효하지 않은 값을 사용할 경우 V는 [6]에서와 같이 C의 거짓 여부를 확인할 수 있다.

V. 결론

본 논문에서는 RSA-Based Convertible Undeniable Signature Scheme[6]을 사용하여, RSA 기반에서의 Designated Confirmer Undeniable Signature Scheme을 제안했다. 서명자가 서명 검증 과정에 참여할 수 없는 경우에는 confirmer가 서명자의 역할을 대신할 수 있으므로, 검증자는 보다 유연하게 전자 서명을 검증할 수 있고, RSA 서명을 사용하는 표준화된 여러 통신 프로토콜에 바로 적용할 수 있는 특징이 있다.

VI. 참고문헌

[1] D. Chaum and H. V. Antwerpen, "Undeniable Signatures", Proc. *CRYPTO'89* pp. 212-217, 1990.

[2] D. Chaum, "Zero-Knowledge Undeniable Signatures", Proc, *EUROCRYPTO'90* pp. 458-464, 1990.

[3] J. Boyar, D. Chaum, I. Damgard and T. Pedersen, "Convertible Undeniable Signatures", Proc. *CRYPTO'90* pp. 189-205, 1991.

[4] D. Chaum, "Designated Confirmer Signatures", Proc. *EUROCRYPTO'94* pp. 86-91, 1994.

[5] R. Gennaro, H. Krawczyk and T. Rabin, "RSA-Based Undeniable Signatures", Proc. *CRYPTO'97* pp. 132-149, 1997.

[6] T. Miyazaki, "An Improved Scheme of the Gennaro-Krawczyk-Rabin Undeniable Signature System Based on RSA", *The 3rd International Conference on Information Security and Cryptology*, Lecture Notes in Computer Science, Springer-Verlag, LNCS 2015, pp. 139-154, 2000.

[7] Y. Desmedt and M. Yung, "Weaknesses of Undeniable Signature Schemes", Advances in Cryptology Proceedings of *EUROCRYPTO'91.*

[8] S. J. Park, K. H. Lee and D. H. Won, "An Entrusted Undeniable Signature", Proceedings of the 1995 *Japan-Korea Workshop on Information Security and Cryptography*, pp. 120-126, Jan. 1995.

[9] *TTAS.KO*-12.0011/R1 해쉬함수표준-제2부 : 해쉬함수알고리즘표준(HAS-160).