논문 00-25-4A-8 한국통신학회논문지 '00-4 Vol.25 No.4A

검증가능한 자체 인증 공개키 방식

정회원 양 형 규*

Verifiable Self-Certified Public Keys Scheme

Hyung-Kyu Yang* *Regular Member*

요 약

Girault에 의해서 처음 제안된 자체인증 공개키 방식은 공개키의 인증이 키의 사용동안에 명백히 검증이 가능토록 한 방식이다. 본 논문에서는 처음으로 검증가능한 자체인증 공개키 에 대한 개념과 방식을 제안하고 이러한 조건을 만족시키는 프로토콜을 제안한다. 검증가능한 자체인증 공개키 방식은 인증기반 방식과 Girault의 자체인증 공개키 방식의 장점을 결합한 방식이다. 더욱이 본 논문에서는 Petersen의 익명이 가능한 자체인증 키 방식에 대해 분석하고 더욱 안전한 방식을 제안한다.

ABSTRACT

Self-certified public keys, introduced by Girault allow the authenticity of public keys to be verified implicitly during the use of the keys. This paper first presents new concept of verifiable self-certified public keys and provides concrete examples satisfying our conditions. Verifiable self-certified public keys combine the benefit of certification-based schemes and Girault's self-certified public keys. Furthermore, we also cryptanalyze Petersen's pseudonymous self-certified keys and present the more secure protocol.

Ⅰ. 서 론

Diffie와 Hellman은 논문 [1]에서 공개키 암호 방식을 제안하였다. 위 방식에서 모든 사용자는 한 쌍의 키 즉, 공개키 P와 비밀키 s를 소유하고 공개키는 공개한다. 그러나 이러한 공개는 공개키를 능동공격에 취약하도록 만든다.

이러한 부정한 공격에 대한 명백한 해결법은 신뢰센터에 의해서 P가 정말로 사용자 I의 공개키라는 것을 확인시키는 인증자 W를 사용하는 것이다. 이러한 방식의 가장 간단한 방법으로 "Certificate"라 불리는 W를 (I, P)에 대한 디지털서명의 형태로 취하면 된다. 이러한 방식은 Girault의 신뢰수준(trust-level) 3을 만족시키지만, 저장용량과 P와 W에 대한 계산량과 같은 추가적인 비용이 소요된다.

Shamir에 의해서 제안된[2] "identity-based" 방식은 위의 결점을 해결하였다. 이 방식의 이점은 P는 I이외의 정보는 제공하지 않으며, W는 s를 제외하고는 어떠한 것도 제공하지 않는다. 그런데 역시 다음과 같은 결점이 있다. 즉, s는 I와 신뢰센터에 의해서 생성된 trapdoor로부터 계산되기 때문에 사용자의 비밀키를 알게 된다. Girault는 논문[3]에서 "self-certified public keys"라 불리는 좀더 진보한 기술을 제안하였다. 이것은 인증기반(certification-based)과 개인식별기반(identity-based) 방식의 중간 정도이다. 위 방식에서 개인식별기반 방식과는 대조적으로 각각의 사용자 I는 자신의 비밀키 s를 선택하고, 자신의 공개키 P'를 계산한다. 이 때 신뢰센터는 (P',I)로부터 trapdoor에 대한 지식 없이는 계산할 수 없는 witness W를 계산한다. 그리고 인증기반 방식과 반대로 witness W는 공개키 P 자체에

* 강남대학교 이공대학 산업전공학부 전자계산전공(hkyang@kns.kangnam.ac.kr)
논문번호 : 99460-1116, 접수일자 : 1999년 11월 16일

포함되지만 분리된 값의 형태를 취하진 않는다. 이 방식은 신뢰수준 2,3을 만족한다.

논문[4,5]에서는 인증기반 방식과 개인식별기반 방식의 장점을 결합한 "self-certified identity-based schemes"을 제안했다. 위 방식은 개인식별기반 방식처럼 P = I = W이고 인증기반 방식처럼 각각의 사용자는 자신의 비밀키 s를 선택한다.[6,7] 표1은 제안된 방식들의 특성을 보여준다.

그런데 자체인증 기반 공개키 방식은 부인을 할 수 있다는 단점이 있다.[8] 인증기반 방식에서 공개키 P와 인증은 witness W를 안 후 곧바로 검증되지만 자체인증 기반 방식에서 인증은 비밀키가 암호화, 서명 검증, 키 분배 혹은 어떤 다른 암호적 사용을 할 때만 검증된다. 예를 들면, 만약 디지털 서명의 검증이 자체인증 공개키를 사용해서 실패했다면, 서명 혹은 공개키가 틀렸는지 확실하지 않다.

본 논문에서는 위 방식의 단점을 해결한 검증가능한 자체인증 공개키에 대한 개념과 방식을 제안하였다. 제안한 방식은 인증기반 방식보다 계산적인 면에서 효율적이고 자체인증 공개키 방식과는 달리 검증이 가능하다. 또한 H. Petersen과 P. Horster이 제안한 pseudonymous 자체인증 키 방식은 신뢰수준 3을 만족하지 못한다는 것을 증명하였고 이것을 보완한 방식을 제안하였다.

Ⅱ. Girault의 자체인증기반 방식

본 장에서는 간단히 RSA/Rabin 디지털서명 방식을 이용한 방식만을 설명한다. 초기 단계에서, CA(Certificate Authority)는 RSA 모듈러 $n = p \cdot q$ (단 p, q는 큰 소수)를 선택하고, p-1과 q-1에 서로소인 정수 e를 생성하고 e moulo $(p-1)\cdot(q-1)$의 역원 d를 계산한다. 이 때 CA는 가환 그룹 $(Z/nZ)^*$의 최대 위수인 정수 α를 계산한다. CA는 n, e 그리고 α를 공개하고 p, q 그리고 d를 비밀로 한다.

키생성 단계는 두 개의 단계로 구성된다. 첫 번째로, 각각의 사용자는 랜덤하게 비밀키 x를 선택하고 자신의 공개키 $y = \alpha^{-x} \pmod n$를 계산하고 y를 CA에게 준다. 이 때 사용자는 CA에게 자신은 x를 안다는 사실을 x에 대한 정보를 누설하지 않고서 증명한다.

CA는 α^{-x} 와 사용자 개인정보의 모듈러 감산에 대한 RSA 서명을 하므로써 witness를 다음과 같이 계산한다.

$$w = (y - I)^d \pmod n$$

결과 다음의 방정식을 얻는다.

$$w^e + I = y \pmod n \qquad (1)$$

그러나 Girault의 자체인증 방식에서는 모든 사용자는 known-key attack만으로도 식(1)을 만족하는 쌍 (y, w)를 얻을 수 있다.

$$w \in_R Z_n,$$

$$y = w^e + I \pmod n$$

그래서 만약 디지털서명에서 검증이 자체인증 공개키를 사용해서 실패했다면 서명이 틀렸는지 공개키가 잘못됐는지 확실치가 않다.

표 1. 제안된 방식들의 특성 비교

I : user's identification string, (s, P): user's key-pair(secret key, public key),
G : witness that P is really the public key of user I

	System	Public key (generation)	Secret key (generation)	Witness (generation)	Trust level	Verify
Public-key	(s, P)	P:user	s : user			
Certification-based schemes	(I, s, P, W)	I,P : user W : authority	s : user	authority's sig. on (I,P)	3	explicit
Identity-based schemes	(I, s) P = I, W = s	I	s:authority	authority	1	(implicit)
Self-certified public keys	(I, s, P) W = P	(I, P)	s : user	authority	2,3	implicit
Self-certified identity	(I, s) P = I = W	I	s : user	authority	2,3	(implicit)

Ⅲ. 제안한 검증가능한 자체인증 공개키 방식

정의 1. (검증가능한 자체인증 공개키) 검증가능한 자체인증 공개키 방식은 다음의 두 조건을 만족한다.

1. (자체인증) witness는 공개키와 같은 속성을 가진다. 사용자의 속성들 즉, 비밀키, 공개키 등은 계산적으로 위조할 수 없는 관계를 만족한다. 이 것은 명백히 임의의 프로토콜에서 키들의 적절한 사용동안 검증되어진다.

2. (검증) 더욱이 필요하다면, witness를 안 후 공개키에 대한 인증을 검증할 수 있는 효율적인 방법이 있다.

3.1 RSA 디지털 서명 방식을 이용한 방식

3.1.1 사전 단계

단계 1. 두 개의 크기가 거의 같은 큰 소수 p와 q의 곱으로써 $n \geq 2^{512}$을 선택. 단, $p = 2p' + 1$, $q = 2q' + 1$ 이고 p'와 q' 역시 소수

단계 2. 위수가 $r = p' \cdot q'$ (즉, $a^r = 1 \pmod n$)인 기저 $a \neq 1$

단계 3. 큰 정수 $u < r$ 을 선택

단계 4. 2^{128}와 p'(그리고 q')사이에 있는 홀수를 출력하는 일방향 함수 h를 선택. 이것은 만약 출력이 짝수면 최하위 비트를 홀수로 변경하므로써 쉽게 구현된다.

단계 5. CA는 n, a, u, f를 공개하고 p와 q를 비밀로 한다.

3.1.2 자체인증 가능한 키 생성 단계

CA를 방문한 엘리스는 만약 CA가 자신을 합법적인 사람으로 간주한다면 witness w_A를 얻는다. witness w_A는 다음처럼 생성된다.

단계 1. 엘리스는 자신의 비밀키 $x_A < u$를 랜덤하게 선택해서 다음처럼 자신의 공개키를 계산한 후 CA를 방문해서 y_A를 CA에게 전송한다.

$$y_A = a^{-x_A} \pmod n$$

단계 2. 엘리스의 신분을 검사한 후 CA는 대응되는 ID_A를 준비해서 다음을 계산한다.

$$w_A = (y_A - ID_A)^{h(y_A)^{-1}} \pmod n$$

CA는 엘리스에게 w_A를 송신한다.

단계 3. 이 때 엘리스의 검증가능한 자체인증 공개키는 w_A와 $e_A = h(y_A)$이다.

지금, 위의 사전단계는 엘리스가 명백히 공개키 y_A에 대한 인증을 검사하는 것을 가능케한다. 왜냐하면 (y_A, w_A)가 주어지면, 다음과 같은 식이 만족된다.

$$w^{h(y_A)} + ID_A = y_A \pmod n$$

Alice		Bob
$e_A = h(y_A)$		
	w_A, c_A →	
		$e_B = h(y_B)$
	← w_B, c_B	
$K_{AB} = (w_B^{e_B} + ID_B)^{x_A}$		$K_{AB} = (w_A^{e_A} + ID_A)^{x_B}$

그림 1. 검증가능한 자체인증 키 교환 프로토콜

3.1.3 키 교환 프로토콜

(ID_A, x_A, w_A, c_A)를 엘리스의 속성이라 하고 (ID_B, x_B, w_B, c_B)를 밥의 속성이라 하자. 이들은 간단히 다음과 같은 값을 계산하므로써 상호 인증된 키를 교환한다.(그림 1 참조)

$$K_{AB} = (w_B^{e_B} + ID_B)^{x_A} = (w_A^{e_A} + ID_A)^{x_B} = a^{x_A \cdot x_B} \pmod n$$

이러한 프로토콜은 Girault의 방식과 연관성을 가진다. 그러나 그것과는 다르게 만약 키 교환 프로토콜이 실패하면, 각각의 사용자는 다음과 같이 계산해서 공개키의 타당성을 검증할 수 있다.

$$\tilde{y}_{A(B)} = w_{A(B)}^{e_{A(B)}} + ID_{A(B)} \pmod n$$

$$e_{A(B)} = h(\tilde{y}_{A(B)})$$

3.2 Schnorr디지털 서명 방식을 이용한 방식

3.2.1 사전 단계

단계 1. $|p| \geq 512$, $|q| \geq 140$ 그리고 q|(p-1)을 만족하는 큰 소수 p, q를 선택

단계 2. 위수 q인 Z_q^*의 교환 그룹에 속해있는 생성자 α($\alpha^q = 1 \pmod q$)를 선택

단계 3. 출력 크기가 128 비트보다 큰 일방향 함수 선택

단계 4. CA는 랜덤 수 $x_{CA} \in_R Z_q^*$를 자신의 비밀키로서 선택하고 자신의 공개키를 다음과 같이 계산

$$y_{CA} = \alpha^{x_{CA}} \pmod p$$

CA는 p, q, α, h을 공개하고 x_{CA}을 비밀리에 보관한다.

3.2.2 검증가능한 자체인증 키 생성 단계

단계 1. 엘리스를 인증한 후, CA는 랜덤 수 $\tilde{k}_A \in_R Z_q^*$을 생성하고 다음과 같이 계산

$$\tilde{r}_A = \alpha^{\tilde{k}_A} \pmod p$$

CA는 $\tilde{r}_A$을 엘리스에게 전송

단계 2. 엘리스는 랜덤 수 $a \in_R Z_q^*$을 선택해서 다음과 같이 계산

$$r_A = \tilde{r}_A \cdot \alpha^a \pmod p$$

엘리스는 ID_A와 r_A를 CA에게 전송

단계 3. CA는 서명을 다음과 같이 계산

$$\tilde{s}_A = x_{CA} \cdot h(ID_A, r_A) + \tilde{k}_A \pmod q$$

계산된 값 $\tilde{s}_A$은 엘리스에게 전송

단계 4. 엘리스는 $x_A = \tilde{s}_A + a \pmod q$을 비밀키로서 보관

단계 5. 엘리스는 $k'_A \in_R Z_q^*$을 이용해서 $r'_A = \alpha^{k'_A} \pmod p$ 을 계산하고 다음처럼 (e'_A, s'_A)을 계산한다.

$$e'_A = h(r'_A)$$
$$s'_A = k'_A - x_A \cdot e'_A \pmod q$$

지금, 엘리스의 검증가능한 자체인증 공개키는 r_A, e'_A, s'_A이다. 이 때 공개 파라메타 y_{CA}, ID_A, r_A이 주어지면 다음과 같은 식이 만족된다.

$$y_A = \alpha^{x_A} = y_{CA}^{h(ID_A, r_A)} \cdot r_A \pmod p$$

더욱이 만약 암호화, 서명 검증, 키 교환, 혹은 다른 암호 프로토콜이 실패했다면, (e'_A, s'_A)이 주어진 상태에서 사용자는 다음과 같이 검사하므로써 공개키에 대한 인증을 검증한다.

$$e'_A = h(\alpha^{s'_A} \cdot (y_{CA}^{h(ID_A, r_A)} \cdot r_A)^{e'_A}) \pmod p$$

계산량 관점에서 고찰해보면 제안한 방식은 인증기반 방식에 비해 계산량을 줄일 수 있다. 즉, 만약 낮은 멱승(예, e = 3)을 인증기반 방식에서 사용한다면 본 방식은 $w_A = (y_A - ID_A \| h(y_A))^e \pmod n$ 혹은 $w_A = (y_A - ID_A \oplus h(y_A))^e \pmod n$를 사용할 수 있다. 안전성에서도 본 방식은 자체인증 공개 키 방식과는 다르게 명백히 검증할 수 있다.(표2와 3을 참고)

표 2. RSA 방식을 이용한 검증가능한 자체인증 공개키에 대한 성능 비교 :

(|n| = 512, |e| = |h()| = 128) (number : the amount of work to perform modular multiplication in 512 bits, WI(b) : the amount of work to perform b-bit modular inversion, WH(b) : the amount of work to compute a hash function with a b-bit long input [10])

	Total length	computational work	
		Verify (implicit)	Verify (explicit)
Certification-based schemes [13]	1024	not available	160 + WH(|n|+|ID|)
Self-certified public keys [2]	512	160	not available
Ours	640	160	(optional) WH(|n|)

표 3. Schnorr 방식을 이용한 검증가능한 자체인증 공개키에 대한 성능 비교

(|p| = 512, |q| = 140, |h()| = 128)

	Total length	computational work	
		Verify (implicit)	Verify (explicit)
Certification-based schemes [13]	780	not available	217 + WH(|p|+|ID|)
Self-certified public keys [2]	512	161 + WH(|p|+|ID|)	not available
Ours	780	161 + WH(|p|+|ID|)	(optional) 217 + WH(|p|)

Ⅲ. Pseudonymous 자체인증 공개키 방식

논문[3]에서 Girault는 다양한 형태의 인증 방식들을 신뢰 수준으로 나누어 분석했으며(즉, 신뢰 수준 1, 2, 3), 논문 [8]에서 Petersen은 신뢰 수준을 "pseudonymous self-certified public keys"라 불리는 단계 4까지 확장했다.

정의 2. (pseudonymous 자체인증 공개키) 신뢰센터는 자체인증 공개키(self-certified public keys)를 pseudonymous PS와 함께 사용자에게 발급한다. 단, 사용자의 진정한 신분은 신뢰센터에게 숨긴다. 그렇치않으면 잘못 pseudonym을 사용하는 모든 연산은 제삼자와 연관될 수 있다.

그래서 pseudonymous 자체인증 공개키 방식은 전자 현금과 같은 응용에 사용된다. Petersen은 blind Schnorr 서명 방식[11]을 사용해서 pseudonymous 자체인증 공개키 발급 프로토콜을 제안했다.(그림 2 참조) 여기서 엘리스는 자신의 고유 신분 ID_A 대신에 pseudonym PS_A를 사용한다. 그리고 CA는 다른 한 쌍의 인증키 ($\tilde{x}_{CA}$, $\tilde{y}_{CA}$)을 위의 프로토콜과 구별하기 위해서 사용한다.

그런데 신뢰 수준 4를 만족하도록 Petersen의 프로토콜을 재구성하면 악의의 제삼자는 자신의 pseudonym 대신에 엘리스의 pseudonym PS_A를 발각되지 않고 사용할 수 있는 단점이 있다. 그리고 Saeednia의 논문에서도 재판관이 악의의 CA와 악의의 사용자를 구별할 수 없으며 재구성된 Petersen의 방식은 신뢰 수준 3조차도 만족시키지 못한다.

악의의 제삼자가 잘못 pseudonym을 사용해서 자체인증 공개키를 얻는 것을 방지하기 위해서 본 논문에서는 키 생성 함수를 CA와 RA(Registration

Alice		CA
		$\tilde{k}_A \in_R Z_q^*$ $\tilde{r}_A = a^{\tilde{k}_A} \pmod p$
	$\tilde{r}_A$ ←	
$a, b \in_R Z_q^*$ $r_A = \tilde{r}_A \cdot a^a \cdot y_{CA}^b \pmod p$ $e_A = h(PS_A, r_A)$ $\tilde{e}_A = e_A + b \pmod q$		
	$\tilde{e}_A$ →	
		$\tilde{s}_A = \tilde{x}_{CA} \cdot \tilde{e}_A + \tilde{k}_A$
	$\tilde{s}_A$ ←	
$x_A = \tilde{s}_A + a \pmod q$		

그림 2. Petersen's 키 발급 프로토콜

Authority)로 분리해서 만든다. RA는 사용자 신분, pseudonym 등에 관한 정보를 등록하고 CA의 서명을 blind한다.

4.1 Pseudonymous자체인증 키 생성 단계

단계 1. CA는 $k_A \in_R Z_q^*$을 선택한 후 다음과 같이 계산

$$\tilde{r}_A = a^{k_A} \pmod{p}$$

$\tilde{r}_A$를 엘리스에게 전송

단계 2. 엘리스는 $a \in_R Z_q^*$을 랜덤하게 선택하고 ID_A, PS_A, $\tilde{r}_A \cdot a^a \pmod{p}$을 안전한 방법으로 RA에게 송신

단계 3. 엘리스의 신분을 검사한 후, RA는 PS_A를 등록하고 다음을 계산

$$b \in_R Z_q^*,$$

$$r_A = (\tilde{r}_A \cdot a^a) \cdot y_{CA}^b \pmod{p},$$

$$\tilde{e}_A = h(PS_A, r_A) + b \pmod{q}$$

그리고 RA는 $\tilde{e}_A$와 이것에 대한 서명을 엘리스에게 전송

단계 4. 엘리스는 $\tilde{e}_A$와 서명을 CA에게 제출

단계 5. RA의 서명을 검사한 후, CA는 $\tilde{s}_A = \tilde{x}_{CA} \cdot \tilde{e}_A + k_A \pmod{q}$을 엘리스에게 송신

단계 6. 엘리스는 자신의 비밀키로써 다음을 계산

$$x_A = \tilde{s}_A + a \pmod{q}$$

그리고 자신의 대응되는 공개키를 다음과 같이 계산

$$y_A = a^{x_A} = \tilde{y}_{CA}^{h(PS_A, r_A)} \cdot r_A \pmod{p}$$

검출되지 않으면서 같은 pseudonym을 악의의 제삼자가 사용하는 것을 방지하기 위한 방법으로 RA는 위의 pseudonymous 자체인증 키를 생성하는데 참여한다. 제안된 프로토콜은 정직한 사용자에 대해서는 익명성을 제공하지만 반대의 경우 RA의 협조에 의해서 사용자의 신분이 노출된다.

V. 결 론

본 논문에서는 검증가능한 자체인증 공개키에 대한 개념을 제안하였다. 검증가능한 자체인증 키 방식은 인증기반 방식보다 계산적인 면에서 효율적이고 자체인증 공개 키 방식과는 달리 검증이 가능하다.

더욱이 본 논문은 H. Petersen과 P. Horster이 제안한 pseudonymous 자체인증 키 방식은 신뢰수준 3을 만족하지 못한다는 것을 증명하였고 이것을 보완한 방식을 제안하였다.

참 고 문 헌

[1] W.Diffie and M.Hellman, "New Directions in Cryptography," *IEEE Trans. Inform. Theory,* vol. 22, pp.644-654, 1976.

[2] A.Shamir, "Identity-based Cryptosystems and Signature Schemes," *Advances in Cryptology (Proceedings of Cryto'84),* Lecture Notes in Computer Science, vol. 196, Springer-Verlag, pp. 47-53, 1985.

[3] M.Girault, "Self-certified Public Keys," *Advances in Cryptology(Proceedings of Euro Crypt '91),* Lecture Notes in Computer Science, vol.547, Springer- Verlag, pp. 490-497, 1991.

[4] Sungjun Park, Chungryong Jang, Kyungsin Kim and Dongho Won, "A Paradoxical Identity-based Scheme Based on The Υ^{th}-residuosity Problem and Discrete Logarithm Problem," *An International Conference on Numbers and Forms, Cryptography and Codes'94,* 1994.

[5] Sungjun Park and Dongho Won, "A "Paradoxical" Identity-based Scheme Based on The Υ^{th}-residuosity Problem and Discrete Logarithm Problem," *Journal of The Korean Institute of Information Security and Cryptology,* vol.4/No. 2, pp.113-118, 1994.

[6] C.H.Lim and P.J.Lee, "Authentication and Digital Signature Schemes using One-time Self-certified Public Keys", *Proc.of JCCI'95, Joint Conference on Communication Information,* pp.33-36, Apr. 1995,

[7] C.H.Lim and P.J.Lee, "A Method for Obtaining Authentication and Digital Signature Schemems using One Secret Key", Korea patient, Serial No. 95-10019 27/4/1995.

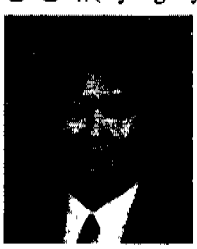
[8] M.Mambo, K Usuda, and E. Okamoto, "Proxy Signatures : Delegation of the Power to Sign Messages", *IEICE Trans. Fundamental*, vol.E79-A, no.9, pp.1338-1354, 1996.

[9] C.P. Schnorr, "Efficient Identification and Signatures for Smart Cards," *Advances in Cryptology (Proceedings of Crpyto'89)*, Lecture Notes in Computer Science, vol. 435, Springer-Verlag, pp. 239-252, 1989.

[10] B.S.Kaliski, "A response to DSS," Nov. 1991.

[11] T.Okamoto, "Provable Secure and Practical Identification Schemes and Corresponding Signature Schemes," LNCS 740, *Advances in Cryptology: Proc. Crypto'92*, Springer, pp.31-53, 1992.

양 형 규(Hyung-Kyu Yang) 정회원



1983년 2월 : 성균관대학교 전자공학과 학사

1995년 2월 : 성균관대학교 전자공학과 석사

1994년 8월 : 성균관대학교 정보공학과 박사

1982년 12월~1990년 2월 : 삼성전자 컴퓨터 부문 선임 연구원

1995년 3월~현재 : 강남대학교 이공대학 산전전공학부 전자계산전공 조교수

<주관심 분야> 암호 이론 및 응용, 네트워크 보안, 전자화폐, 정보 은닉 등