안전한 비밀키 갱신이 가능한 Schnorr 형 서명 기법

김중만, 김광조

*한국정보통신대학원대학교, 공학부

Intrusion-Resilient Key-Evolving Schnorr Signature Scheme

JoongMan Kim, Kwangjo Kim

*Department of Computer Science Information & Communications Univ.

요 약

보안 시스템에서 비밀키 노출은 가장 심각한 문제 중 하나이며, 피할 수 없는 문제로 여겨진다. 최근에 키 전개(key-evolving) 패러다임이 비밀키 노출이 가져올 수 있는 피해를 최소화시키기 위한 수단으로 도입되었다. 이 패러다임에서는 프로토콜 전 구간이 여러 시간 구간으로 나누어진다. 시간 구간 i에서 서명자는 비밀키 SK_i 를 가지게 되며, 이를 주기적으로 갱신하게 된다. 반면 공개키 PK는 프로토콜 전 구간동안 변함이 없다. 이러한 개념을 기반으로 하여 본 논문에서는 안전한 비밀키 갱신이 가능한 Schnorr 형 서명 기법을 제시한다. 이 서명 기법은 다음과 같은 특징을 갖는다. 만약 모든 시간 구간에 있는 비밀키들이 노출되지 않는다면, 아직 노출되지 않은 시간 구간의 서명을 위조할 수 없다. 즉, 모든 시간 구간의 비밀키를 알아야 어떤 시간 구간의 서명을 위조할 수 있다. 이 서명 기법은 위조 불가능하게 안전하며, 이산 로그 (Discrete Logarithm Problem)의 어려운 문제에 기반한 Schnorr 서명 기법에서 유도 되었다.

I. 서론

비밀키 노출은 피할 수 없는 문제로 여겨진다. 특히, 전반적으로 비밀키 노출을 방지하는 방법들은 비용이 많이 들며, 대부분의 상용 애플리케이션에 있어서 비실용적이다. 따라서 비밀키 노출이 가져올 수 있는 부정적인 영향을 최소화하는 것이 중요하다. 이러한 문제와 관련하여 많은 연구가 이루어져 왔다.

전방 보안(Forward-secure) 서명 기법 [1], [2]에서는 서명자는 비밀키를 소유하며, 매 시간 주기마다 키가 갱신된다. 이러한 서명 기법은 현재 시간 구간의 비밀키가 노출되더라도 과거 시간 구간 안에 있는 서명의 안전성은 보장된다.

문턱(Threshold) 서명 기법 [3], [9] 에서는 비밀키가 n개의 디바이스들 간에 공유된다. 가령, n개의 디바이스들 중 t개의 디바이스들이 공모를 하여도 공격자는 그 기법을 깰 수 없다.

Key-insulated 서명 기법 [4] 에서 공격자는 과거 뿐 아니라 미래의 서명도 위조할 수 없다. 이러한 서명 기법은 서명자가 주기적으로 통신하는 서버의 안전한 저장 공간을 가정하고 있다.

최근 제안된 intrusion-resilient 서명 기법 [5]은 key-insulated 서명 기법과 같이 두 모듈을 가정하고 있다. 하나는 "서명자" 이고, 또 하나는 "base"이다. intrusion-resilient는 두 모듈내의 비밀키 노출이 동시에 이루어지지 않는다면, 과거와 미래 시간 구간 안에 있는 서명의 안전성이 보장된다. 또한 두 모듈 내 비밀키 노출이 동시에 이루어진다면, 과거 시간 구간 안에 있는 서명의 안전성만 보장된다.

Tzeng 등은[10] 전방 보안(forward-secure) 서명 기법과 비슷한 키 전개 패러다임을 제안했다. 그들은 공개키 기반 구조에서의 비밀키 노출 문제를 다루었지만, 서명 기법에서도 비슷할 것이다. 이 패러다임에서는 우선 프로토콜 전 구간을 여러 시간 구간으로 나눈다. 서명자의 공개키 PK는 프로토콜 전 구간동안 고정 될 것이다. 시간 구간 i에서의 서명자의 비밀키는 $SK_i, i \geq 0$ 이다. 시간 구간이 i에서 $i+1$로 진행 하게 되면, 서명자는 자신의 비밀키를 SK_i에서 SK_{i+1}로 갱신을 한다. 그리고 SK_i을 바로 삭제한다. 이 때 TA(trusted agent)의 도움을 받을 수도 있다. 만약 공격자가 시간 구간 i에서 서명자의 시스템을 침입해서 그 시간 구간에서의 비밀키 SK_i을 얻는다하더라도 그 이전 키들은 지웠기 때문에 직접적으로 이전 서명을 얻을 수는 없다. 또한 서명자가 자신의 키가 도난당했다는 사실을 모른다 하더라도 그는 바로 그 시간 구간안의 서명만 위조될 것이라는 사실을 확신할 수 있을 것이다. 본 논문에서는 이러한 패러다임을 이용하여 intrusion-resilient의 개념을 제안한다. 이 intrusion-resilient는 Itkis 등이[5] 제안한 개념과는 다른 개념으로 선형 연립방정식과 Threshold 암호의 성질을 이용해서 얻어 질 것이다.

본 논문의 구성은 다음과 같다: 본문의 첫 번째 장에서는 intrusion-resilient 개념을 제시하고, 두 번째 장에서는 제안 프로토콜을 소개하며, 세 번째 장에서는 제안 프로토콜의 안전성 분석을 한다. 마지막으로 결론을 정리한다.

II. 본문

1. 정의

이 장에서는 키 전개 프로토콜에 [6] 기반한 서명 프로토콜 정의를 제시한다. 이 정의에서는 여러개의 TA들이 있어 서명자의 비밀키를 갱신하기 위한 비밀키 부분값(share)을 가지고 있다고 가정한다. 그리고 나서는 intrusion-resilient 개념을 제시한다.

1) 표기법

키 전개 서명 기법은 4개의 알고리즘 -키 생성 알고리즘, 키 갱신 알고리즘, 서명 생성 알고리즘, 서명 검증 알고리즘-으로 구성 되어 있다. 본 논문에서는 $\{0,1\}^*$을 유한길이의 이진 문자열들의 집합으로 정의하고, 다음과 같이 두개의 해쉬 함수 $h_1 : \{0,1\}^* \rightarrow \{0,1\}^t$과 $h_2 : \{0,1\}^* \rightarrow Z_q$를 정의한다. 또한 해쉬 함수 h_1의 출력값을 비트 표기법으로 나타내어 벡터 값을 생성하는 벡터 함수 B를 정의한다. 벡터 함수 B는 키 생성 알고리즘에서 정의된다. 또한 본 논문에서 사용되는 t는 제안 프로토콜의 전 구간을 나타내는데 사용된다. 본 논문에서는 TA의 존재를 가정한다. 실제로 각각의 TA_j는 서명자의 비밀키 s의 부분값인 s_j를 공유하게 된다. 그리고 서명자는 TA들이 가지고 있는 값과 자신이 가지고 있는 값들을 가지고 Shamir의 (k,n) 문턱 서명 기법을 [9] 통해서 안전한 방식으로 비밀키 SK_i를 계산하게 된다.

2) 보안 요구 사항

키 전개 프로토콜이 갖는 특성은 다음과 같다. [6]

- 전방 안전성 (forward-secret)

현재 시간 구간의 비밀키를 이용해서 과거 시간 구간의 비밀키를 알 수 없다.

- 후방 안전성 (backward-secret)

현재 시간 구간의 비밀키를 이용해서 미래 시간 구간의 비밀키를 알 수 없다.

- 키 독립성 (key-independent)

적절한 시간 구간의 비밀키를 이용해서 어떠한 시간 구간의 비밀키를 알 수 없다. 즉, 전방 안전성과 후방 안전성을 모두 만족한다.

3) Intrusion-Resilient 개념 정의

Tzeng 등은 [10] 공개키 기반 구조에서 "resilience"의 개념을 제안했다. 이 개념은 서명 기법에서도 다음과 같이 똑같이 적용 될 것이다.

[정의 1] 서명 기법을 위한 보안 모델 하에서 만약 공격자가 z개의 비밀키 $SK_{i_1}, SK_{i_2}, \ldots, SK_{i_z}$ 들을 획득 할 지라도 그가 서명 기법을 깰 수 없다면 이런 키 전개 서명 기법을 z-resilient하다고 정의 한다.

위 정의에 의하면 공격자는 z 시간 구간들의 비밀키 $SK_{i_1}, SK_{i_2}, \ldots, SK_{i_z}$를 획득 할 지라도 아직 노출이 안 된 다른 시간 구간 안에 있는 비밀키를 구할 수 없다. 실제로 제안 프로토콜은 $(t-1)$-resilient한 프로토콜이다. 그래서 다음과 같은 정의를 제시한다.

[정의 2] 키 전개 프로토콜이 $(t-1)$-resilient하다면 그 프로토콜을 intrusion-resilient하다고 정의한다.

이 정의에 의하면 제안 프로토콜은 intrusion-resilient 기법이다. 제안 프로토콜의 안전성 분석에서 이를 보일 것이다.

2. 제안 프로토콜

이 장에서는 Shamir의 (k,n) 문턱 서명 기법을 이용한 제안 프로토콜을 기술한다. 이 프로토콜은 위조 불가능하게 안전하며, 이산 로그의 어려운 문제에 기반한 Schnorr 서명 기법에서 유도 되었다. [8]

- 키 생성 알고리즘

a) q가 $p-1$의 약수인 큰 소수 q와 p를 선택한다.

b) 차수가 $q \in Z_p^*$인 순환군 G_q에서 생성원 g를 생성한다.

c) 비밀키 정보 $S=(s_1, s_2, ..., s_t)$를 G_q 안에서 생성한다.

d) 비밀키 정보 S를 이용하여 공개키를 계산한다.

$$PK = (g^{s_1}, g^{s_2}, ..., g^{s_t})$$

e) 두 개의 충돌 회피 해쉬 함수를 사용한다.

$h_1 : \frac{1}{2}0,1\frac{3}{4}^* \rightarrow \frac{1}{2}0,1\frac{3}{4}^l$과 $h_2 : \frac{1}{2}0,1\frac{3}{4}^* \rightarrow Z_q$

f) 벡터 함수를 정의한다.

$B(x) := h_1(x)$의 바이너리 표현

$:= (e_1, e_2, ..., e_t)$

여기서 e_i들은 비트들이다.

g) t개의 $t-1$차수의 다항식을 Z_p^* 안에서 생성한다.

$$f_1(x) \equiv s_1 + \sum_{i=1}^{t-1} \square_{1,i} x^i \bmod q$$

$$f_2(x) \equiv s_2 + \sum_{i=1}^{t-1} \square_{2,i} x^i \bmod q$$

$$\vdots$$

$$f_t(x) \equiv s_t + \sum_{i=1}^{t-1} \square_{t,i} x^i \bmod q$$

이 다항식들은 비밀키 정보 S의 각 원소들을 공유하기 위해 사용 될 것이다.

h) 서명자는 자신과 TA들에게 다중 부분값 (multiple share) $MS_l, 1 \le l \le n$, 을 분배한다.

$$MS_1 = (f_1(x_1), f_2(x_1), ..., f_t(x_1))$$

$$MS_2 = (f_1(x_2), f_2(x_2), ..., f_t(x_2))$$

$$\vdots$$

$$MS_n = (f_1(x_n), f_2(x_n), ..., f_t(x_n))$$

이때 j 개의 TA들이 있다고 가정하면 서명자는 이 TA들에게 MS_1에서 MS_j까지 분배하고, 나머지는 자신이 소유한다.

i) 다중 부분값의 분배가 끝나면 비밀키 정보 S를 바로 삭제한다. 서명자의 공개키는 PK이다.

- 키 갱신 알고리즘

a) 각 $TA_{r_1}, 1 \le r_1 \le j$, 들은 비밀키 갱신을 위해 다음을 계산한다.

$$SKU_{r_1} = (f_1(x_{r_1}), f_2(x_{r_1}), ..., f_t(x_{r_1})) \cdot B(i+1) \bmod q$$

여기서 $\cdot$ 는 내적을 의미한다. 이 SKU_{r_1}를 서명자에게 보낸다.

b) 서명자는 자신이 가지고 있는 다중 부분값들 중 $(k-j)$개를 랜덤하게 선택한 후 다음을 계산한다.

$$SKU_{r_2} = (f_1(x_{r_2}), f_2(x_{r_2}), ..., f_t(x_{r_2})) \cdot B(i+1) \bmod q$$

여기서 $d_1 \le r_2 \le d_{k-j}$ 이다.

c) 서명자는 다음과 같이 Lagrange 보간법을 사용하여 비밀키를 갱신한다.

$$SK_{i+1} = \sum_{r_1=1}^{j} SKU_{r_1} \cdot \left(\prod_{t_1 \le l \ne r_1 \le t_k} \frac{x_l}{x_l - x_{r_1}} \right)$$

$$+ \sum_{r_2 = d_1}^{d_{k-j}} SKU_{r_2} \cdot \left(\prod_{t_1 \le l \ne r_2 \le t_k} \frac{x_l}{x_l - x_{r_2}} \right) \bmod q$$

여기서 $(t_1, t_2, ..., t_k) = (1, ..., j, d_1, ..., d_{k-j})$ 이다.

- 서명 생성 알고리즘

a) 난수 $k \in Z_q^*$ 를 선택한다.

b) $r = g^k \bmod p$

$e = h_2(m, r)$

$z = (SK_i) \cdot e + k \bmod q$

c) 메시지 m에 대한 서명값은 (z, e, i) 이다.

- 서명 검증 알고리즘

a) 검증자는 서명자의 인증된 공개키 PK를 얻는다.

$PK = (g^{s_1}, g^{s_2}, ..., g^{s_t})$

b) $v = g^z \cdot (g^{(s_1, s_2, ..., s_t) \cdot H(i)})^{-e} \bmod p$

$e0 = h_2(m, v)$

c) $e = e0$이 만족하는지 확인하여 검증한다.

키 생성 알고리즘에서 서명자가 다중 부분값들을 분배하는 과정에서 서명자가 가지게 될 양과 TA의 수와의 관계는 다음과 같다. (k는 임계 값이다.)

$$n - j < k \text{ 와 } n < 2k - 2$$

이 두개의 부등식들은 오로지 서명자만이 또는 TA들만이 비밀키를 갱신할 수 없도록 하기 위함이다. 즉, 서명자와 TA들은 반드시 서로 공모해야 한다. 분배 후 intrusion-resilient를 보장하기 위해서 비밀키 정보 S를 삭제한다.

또한 키 갱신 알고리즘에서 서명자가 각 $TA_l, 1 \le l \le j$, 들로부터 올바른 값을 받았는지를 검증하기 위해 각 TA_l들은 $g^{f_1(x_l)}, g^{f_2(x_l)}, ..., g^{f_t(x_l)}$ 값들을 공개한 후, 서명자는 다음을 계산한다.

$$g^{SKU_i} \equiv g^{(f_1(x_i), f_2(x_i), ..., f_t(x_i)) \cdot B(i)} \bmod p$$

이때 $g^{(f_1(x_i), f_2(x_i), ..., f_t(x_i)) \cdot B(i)}$ 은 각 TA_l들이 공개한 값들을 시간 구간 i의 해쉬 값에 기반하여 곱한 값을 의미한다.

3. 제안 프로토콜의 안전성 분석

1) 복잡도 및 효율성 분석

제안 프로토콜은 사이즈가 $O(t)$인 공개키를 갖는다. 키 생성 알고리즘에서는 처음에 비밀키 정보 S를 생성한 후 그것을 이용하여 공개키 PK를 계산한다. 이 과정은 t번의 모듈 멱승(modular exponentiation)을 요구한다. 두번째로 키 갱신 알고리즘은 합과 곱만으로 이루어져 있다. Shamir의 (k, n) 문턱 서명 기법을 사용하기 때문에 키 갱신 과정은 $(k \cdot \frac{t}{2})$번의 모듈 합(modular summation) 과 k번의 모듈 곱(modular multiplication)을 필요로 한다. 여기서 $\frac{t}{2}$가 의미하는 것은 이상적인 충돌 회피 해쉬 함수 h_1은 그 출력값을 바이너리 표현으로 나타냈을 때 평균적으로 $\frac{t}{2}$개의 '1' 비트들을 생성함을 말한다. 마지막으로 서명 생성 알고리즘은 Schnorr 서명 기법을 사용하기 때문에 Schnorr 서명 기법과 똑같은 복잡도를 갖는다. 하지만 제안 프로토콜은 공개키 PK의 원소를 이용하여 시간 구간 i에서 검증을 위해 사용 되어질 값 $g^{(s_1, s_2, ..., s_t) \cdot B(i)}$ 을 구하는 과정이 수행된다. 이 과정은 검증 하기 전 검증자에 의해 미리 계산이 가능하다. 따라서 서명 검증 과정에서 $\frac{t}{2}$번의 모듈 곱을 요하는 계산 과정이 더 요구된다.

2) 안전성 분석

Pointcheval 과 Stern은 [7] Schnorr 서명 기법이 선택된 메시지 공격에 대하여 위조 불가능성(Unforgeable Against Chosen Message Attack)을 증명하였다. 따라서 제안 프로토콜 또한 메시지 공격에 대하여 위조 불가능성이 됨을 다음 정리에서 보인다.

[정리 1] Schnorr 서명 기법이 위조 불가능하면

제안 프로토콜 또한 위조 불가능하다.

[증명] 우리는 이 정리의 대우를 증명할 것이다. 즉, 제안된 프로토콜이 위조 가능하면 Schnorr 서명 기법 또한 위조 가능함을 보이자. 위조자를 시뮬레이션 할 것이다. 위조자는 서명자로 하여금 또 다른 타당한 서명을 생성할 수 있다고 하자. 이 때 서명자에게 주어지는 모든 정보는 메시지 m과 서명값 (z, e, i)이다. 결국 방정식 $z = (SK_i) \cdot e + k$ 은 성립된다. 이것은 메시지 m에 대한 Schnorr 서명임을 알 수 있다. 따라서 위조자는 서명자를 이용하여 Schnorr 서명 메카니즘을 깰 수 있다. □

제안 프로토콜이 *intrusion-resilient* 키 전개 Schnorr 서명 기법임을 다음 정리에서 보일 것이다. 제안된 프로토콜이 intrusion-resilient함을 보이기 위해 선형 연립 방정식의 성질을 이용한다.

[정리 2] 랜덤 오라클 모델에서 제안 프로토콜은 intrusion-resilient 하다.

[증명] 공격자는 비밀키 정보 S의 각 원소들이 미지수가 되고, 각 시간 구간의 비밀키들이 선형 방정식을 이루는 선형 연립 방정식을 생성 할 수 있다. 비밀키 정보 S의 모든 원소들의 수와 모든 시간 구간의 수는 t이므로 선형 연립 방정식의 성질에 의해 공격자는 과거든 미래든 간에 아직 노출 되지 않은 시간 구간의 서명을 얻기 위해 모든 시간 구간의 비밀키들을 알아야 한다. 이것은 제안된 프로토콜이 $(t-1)$-resilient함을 의미한다. 따라서 제안 프로토콜은 intrusion-resilient하다.□

제안 프로토콜이 intrusion-resilient함을 보였다. intrusion-resilient가 전방 안전성(forward-secret)을 만족하는 개념임을 다음 정리에서 보일 것이다.

[정리 3] 랜덤 오라클 모델에서 제안 프로토콜이 intrusion-resilient 하면 전방 안전성(forward-secret)을 만족한다.

[증명] 우리는 이 정리의 대우를 증명할 것이다. 즉, 제안된 프로토콜이 전방 안전성(forward-secret)을 만족하지 않으면, intrusion-resilient를 만족하지 않음을 보이자. 공격자는 현재 시간 구간 i의 비밀키 SK_i 를 얻은 상태에서 그 이전 시간 구간 j의 비밀키 $SK_j\,(j < i)$ 를 서명자로부터 얻을 수 있다. 이것은 t개의 선형 방정식들과 t개의 서로 다른 미지수들이 이루는 선형 연립 방정식의 성질을 깨는 것이다. 즉 미지수가 t개인 선형 연립 방정식에서 최소한 t개 이상의 선형 방정식을 알아야 그 선형 연립 방정식의 해를 구할 수 있다. 따라서 이것은 제안된 프로토콜이 $(t-1)$-resilient하지 않음을 의미한다. 따라서 제안 프로토콜은 intrusion-resilient를 만족하지 않는다. □

[정리 3]과 비슷하게 제안 프로토콜이 후방 안정성(backward-secret)을 만족함을 보일 수 있다. 따라서 제안 프로토콜은 키 독립성 (key-independent)을 만족한다.

III. 결론

본 논문에서 안전한 비밀키 갱신이 가능한 Schnorr 형 서명 기법이 제안 되었다. 우리는 *intrusion-resilient*가 비밀키 노출 문제에 대하여 최고의 안전성을 보장 할 것이라 생각 된다. intrusion-resilient를 얻기 위해 제안된 프로토콜은 선형 연립 방정식의 성질을 이용하였고, 키 갱신은 Shamir의 (k, n) 문턱 서명 기법을 사용하였다. 또한 *intrusion-resilient*가 전방 안전성(forward-secret)을 만족함을 제안 프로토콜의 안전성 분석에서 보였다. 그리고 제안 프로토콜은 Schnorr 서명 기법을 사용하기 때문에 서명 메카니즘은 Schnorr 서명 기법과 똑같은 효율성을 갖는다 할 수 있다. 제안된 프로토콜에서 Shamir의 (k, n) 문턱 서명 기법을 사용하기 위해 여러 TA 들을 사용했지만, 실제로는 TA 가 아닌 일반적인 에지전트를 사용 할 수 있다는 것을 쉽게 보일 수 있다. 따라서 앞으로 TA 없는 프로토콜을 만드는 것도 상당히 흥미진진한 일이라 할 수 있을 것이다.

제안된 프로토콜은 이산 로그의 어려운 문제에 기반하기 때문에 이 문제에 기반한 다른 암호 기법들 즉, ElGamal, DSS와 같은 일반적인 서명 기법에 폭 넓게 사용 될 수 있는 장점을 가지고 있다. 또한 Schnorr 서명 기법이 스마트 카드에 적합한 기법인 만큼 제안된 프로토콜도 이러한 애플리케이션에 적용되어 개발이 가능 할 것이다.

참고문헌

[1] R. Anderson, "Two Remarks on Public-Key Cryptology", Invited lecture, Fourth Annual Conference on Computer and Communications Security, ACM, 1997.

[2] M. Bellare and S. Miner, "A Forward-Secure Digital Signature Scheme", In

Michael Wiener, editor, Advances in Cryptology - Crypto '99, Springer-Verlag, 15-19 August 1999.

[3] Y. Desmedt and Y. Frankel, "Threshold cryptosystems", In G.Brassard, editor, Advances in Cryptology - Crypto '89, Springer-Verlag, LNCS 435, pp. 307-315, 1990.

[4] Y. Dodis, J. Katz, S. Xu, and M. Yung, "Key-Insulated Public-Key Cryptosystems", In Lars Knudsen, editor, Advances in Cryptology - Eurocrypt 2002, Springer - Verlag, LNCS, 28 April-2 May 2002.

[5] G. Itkis and L. Reyzin, "SiBIR: Signer-Base Intrusion-Resilient Signatures", In Moti Yung, editor, Advances in Cryptology - Crypto 2002, Springer-Verlag, LNCS, 18-22 August 2002.

[6] C. -F. Lu and S. W. Shieh, "Secure Key-Evolving Protocols for Discrete Logarithm Schemes", In Proceedings of The Cryptographer's Track at the RSA Conference 2002., Springer-Verlag 2002, LNCS 2271, pp. 300-310.

[7] D. Pointcheval and J. Stern, "Security Proofs for Signature Schemes",In Ueli Maurer, editor, Advances in Cryptology - Eurocrypt '96, Springer-Verlag, LNCS 1070, pp. 387-398, 12-16 May 1996.

[8] C. P. Schnorr, "Efficient Identification and Signatures for Smart Card", Advances in Cryptology - Eurocrypt '89, Springer-Verlag, pp. 235-251.

[9] A. Shamir, "How to share a secret", Communications of the ACM, 22(11), pp. 612-613, 1979.

[10] W. -G. Tzeng and Z. -J. Tzeng, "Robust Key-Evolving Public-Key Encryption Schemes", Record 2001/009, Cryptology ePrint Archive, 2001.