사용자가 직접 제어 가능한 임계 암호 시스템

양종필* · 이경현**

1. 서 론

인터넷 환경에서의 전자 상거래를 위한 트랜잭션이 빈번해지면서, 기업과 개인들은 자신의 비밀키를 전자 서명을 수행하기 위해서 사용하고 있다. 전자서명을 위한 기업과 개인의 비밀키는 인터넷 환경에서 악의적 공격자들의 주요한 공격 목표가 되고 있기 때문에, 안전한 전자 상거래를 위하여 이러한 비밀키의 보호하기 위한 기술은 가장 중요한 연구 이슈가 되고 있다.

보다 안전한 비밀키의 보호를 위하여, 1979년 Shamir는 단일 비밀키를 다수의 서버들에게 분산시켜서 보관하는 비밀 분산 기법(Secret sharing scheme)을 제안하였다[2]. 하지만, 이러한 비밀 분산 기법은 분산된 비밀키를 통한 전자서명을 수행하기 위해서는 어느 한 장소에서 비밀키를 복구한 후에 전자서명을 수행해야하며, 비밀키의 복구가 이루어지는 장소가 공격을 당할 경우에 비밀 분산 자체를 통한 비밀키의 보호에 대한 이득이 사라지게 된다. 따라서, 많은 연구자들은 비밀키 자체의 분산 보안이 아닌 비밀키를 사용하는 함수에 대한 비밀 분산인 임계 전자서명 기법을 제안하였다. 임계 전자서명 알고리즘들은 기존의 전자 상거래에 핵심적인 요소기술인 전자서명을 위한 능력을 단일 시스템이 아닌 다중 시스템에 분산시킴으로써, 전자서명에 사용되는 비밀키를 통한 서명 능력을 안전하게 보전하는 것이 주된 목적이라 할 수 있다. 하지만, 안전한 전자서명을 위해서 각 기업 또는 사용자가 자신들을 위한 분산 시스템을 독자적으로 구축하는 것은 시스템의 구축 및 유지보수를 위한 많은 비용을 초래하게 된다. 따라서, 전자서명에 사용되는 비밀키의 보안이 매우 민감한 개인 사용자와 기업을 위하여 비밀키들을 분산된 방법으로 다수의 서버 시스템에 안전하게 보존하며, 분산된 그 비밀키에 의한 암호학적인 연산을 안전하게 대행해 주는 서비스 및 시스템이 요구된다. 하지만, 이러한 환경에서 분산된 서버 시스템에서는 암호학적 연산을 단지 활성화가 가능한 악의적 사용자는 분산된 비밀키의 소유자의 비밀키를 모르는 상황에서도 그 비밀키의 소유자의 암호학적인 연산을 대신 수행할 수 있는 위험이 존재한다. 즉, 기존의 임계 암호시스템을 통한 분산된 시스템의 구축을 통해서는, 사용자의 동의 없이 비밀키를 통한 악의적인 전자서명 및 암호문의 복호화 수행을 효율적으로 방지하지 못한다. 본 논문은 위와 같은 문제를 해결하기 위하여 사용자는 다수의 서버시스템에 비밀 분산된 자신의 비밀키를 통한 암호 연산 수행에 대한 직접적인 제어권을 소유하는 기법을 제안한다.

* 부경대학교 전자계산학과 박사과정
** 부경대학교 전자컴퓨터정보통신공학부 교수
※ 삼가 이 논문을 故 박지환 교수님 영전에 올립니다.

본 논문의 구성은 다음과 같다. 제 2장에서는 관련 연구 및 본 논문을 위한 통신 모델에 대해서 언급하며, 제 3장에서는 본 논문에서 제안하는 기법을 소개한다. 또한, 제 4장에서는 제안 기법의 보안성에 대해서 논한다. 그리고, 5장에서는 결론을 맺는다.

2. 관련 연구

2.1 전자서명 암호 기술

■ 전자서명 기법(Signature Scheme)

일반적인 전자서명 기법은 다음과 같이 세 가지 알고리즘으로 구성된다: $OS=(SO.Init, OS.Sig, OS.Ver)$. 보안 파라메터 α를 입력으로, 확률적 키 생성 알고리즘 $OS.Init$은 공개키와 비밀키 쌍 (Y, X)을 출력한다. 메시지 M과 비밀키 X를 입력으로, 서명 알고리즘 $OS.Sig$은 전자서명문 σ를 출력한다. 어떤 전자서명문 σ, 공개키 Y, 메시지 M을 입력으로, 검증 알고리즘 $OS.Ver$은 σ가 유효하면 TRUE를, 유효하지 않으면 FALSE를 출력한다.

■ 임계 전자서명 기법(Threshold Signature Scheme)

(k, l)-임계 전자서명 알고리즘은 l개의 서버 중에서 k개의 서버로 이루어진 어떤 부분집합은 유효한 전자서명을 생성할 수 있지만, k보다 적은 수의 서버들이 참여할 때는 유효한 전자서명을 생성할 수 없는 시스템이다. 결과적으로, 비밀 분산과는 달리 사용자는 자신의 비밀키에 대응되는 서명 함수(Signing Function)를 l개의 서버에 분산시키게 된다. 따라서, 비밀 분산기법에서의 비밀키의 노출에 대한 위험이 없어지게 된다. 일반적인 임계 전자서명 알고리즘은 다음과 같은 세 가지의 알고리즘으로 구성된다: $TS=(TS.Init, TS.Sig, TS.Ver)$. 여기서, t를 공격자에게 침입을 허용한 서버들의 수라고 두자. 그러면, 임계 전자서명 알고리즘의 세 가지 알고리즘은 다음과 같이 동작된다.

- $TS.Init$: 보안 파라메터 α, 허용되는 침입된 서버들의 수 t, 서버들의 전체 수 l을 입력으로, $TS.Init$는 기반하는 공개키 암호 시스템을 위한 공개키 Y와 이에 대응되는 비밀키 X를 출력한다. 그리고, 출력된 비밀키 X는 비밀 분산 기법을 통하여 l개의 서버들 $S_1, ..., S_l$ 에게 안전하게 분배된다.

  $X \rightarrow^{(k,l)} (X^{(1)}, ..., X^{(l)})$

서버 S_i는 자신의 비밀 분배키으로 $X^{(i)}$를 소유하게 된다. 여기서, $1 \le i \le l$ 이다.

- $TS.Sig$: 어떤 메시지 M을 전자서명하기 위해서, S_{i_j}는 자신의 비밀 분배키를 사용하여 아래와 같은 부분 서명문 $\sigma^{(i_j)} = g_1(M, X^{(i_j)})$ 를 계산한다. 그리고, k개 이상의 부분 서명문을 통하여, 누구나 유효한 전자서명문 $\sigma = g_2(\sigma^{(i_1)}, ..., \sigma^{(i_k)})$을 계산 가능하다. 여기서, $1 \le j \le k$에 대하여 $t+1 \le k \le l$, $1 \le i_j \le l$이다. 또한, g_1과 g_2는 기반하는 전자서명 알고리즘에 의존하는 알고리즘이다.
- $TS.Ver$: 임계 전자서명 시스템에 의해서 생성된 전자서명문의 검증 절차는 기반하는 전자서명 알고리즘의 검증절차와 동일하다.

V.Shoup[15], D.Boneh[3,5,14], P. Fougque[7]과 T.Rabin[13]는 RSA 알고리즘을 기반한 임계 전자서명 기법을 제안했으며, R.Gennaro[10,11]

는 이산 대수 문제에 기반한 임계 전자서명 기법을 제안하였다.

2.2 통신 모델

본 논문에서 제안하는 기법을 위한 참여자들은 아래와 같다.

- 사용자(User) : 자신의 비밀키를 l개의 서버들에게 안전하게 비밀 분산하여 보관하고 싶어한다. 또한, 적어도 $k \geq t+1$개의 서버들이 그 사용자를 위한 전자서명문을 생성하기 위해서 서명 함수를 수행할 수 있으며, 사용자는 자신을 위한 서명 함수의 활성화에 대한 제어권을 소유한다. 어떤 기업, 개인 또는 인증기관(CA)등이 사용자가 될 수 있다.
- 서버(Server) : 제안된 기법에서는 l개의 서버들 $\{S_1, \ldots, S_l\}$이 존재한다. 모든 서버들은 각 사용자의 비밀 정보에 대한 분배값(share)들을 안전하게 보관하고 있으며, 각 사용자를 위한 서명 함수를 대신하여 수행한다.

본 논문에서는 어떤 공격자는 계산적으로 제한되어져서, k개 이상의 서버들에 대한 공격을 성공하지 못함을 가정한다. 그리고, 서버들은 전용 브로드캐스트 채널에 접근할 수 있음을 가정한다. 따라서, 만약 어떤 서버 S_i가 어떤 메시지를 전송하면, 그 메시지는 다른 모든 서버들에 의해서 수신되며, S_i로부터 보내어진 메시지임을 알 수 있다.

3. 사용자 제어 가능한 전자서명 기법

본 장에서는 사용자가 다수의 서버에게 분산되어진 자신의 서명 함수들의 활성화에 대한 제어권을 가지기 위한 기법을 제안한다.

3.1 제안 기법의 구성

본 논문에서 제안하는 사용자 제어 가능한 임계 전자서명(User controllable threshold signature : UCTS) 기법은 사용자가 전체 l개의 서버들에게 분배된 자신의 비밀키로부터 암호학적 계산을 위해서 사용되는 연산(함수)들의 활성화를 제어할 수 있다. 사용자 제어 가능한 임계 전자서명 기법은 기존의 임계 전자서명 기법과 동일한 구조로 $UCTS = (UCTS.Init, UCTS.Sig, UCTS.Ver)$의 세 가지 알고리즘으로 구성된다.

- $UCTS.Init$: 사용자는 비밀키를 생성하고, 생성된 비밀키 정보를 l개의 서버에 분배하는 단계이다.
  1. 보안 파라메터 α, 허용되는 침입된 서버들의 수 t, 서버들의 전체 수 l을 입력으로하여, $UCTS.Init$는 기반하는 공개키 암호시스템을 위한 공개키 Y와 이에 대응되는 비밀키 X를 출력한다.
  2. 사용자는 기반하는 전자서명 알고리즘의 비밀키 생성 규칙에 부합되는 랜덤값 δ를 생성하여, $SK = X \circ \delta$를 계산한다. 여기서, δ를 제어 파라메터라고 하며, 생성된 SK를 가상 비밀키라고 한다. 또한, 연산자 $\circ$는 기반 알고리즘에 의존적으로 결정된다.
  3. 사용자는 SK를 Shamir의 (k, l)-비밀 분산기법을 사용하여 l개의 서버들 $S_1, \ldots, S_l$에게 비밀 분산한다.

$$SK \rightarrow^{(k,l)} (SK^{(1)}, \ldots, SK^{(l)})$$

서버 S_i는 가상 비밀 분배키으로 $SK^{(i)}$를 소유하게 된다. 여기서, $1 \leq i \leq l$ 이다.

- $UCTS.Sig$: 메시지 M에 대한 전자서명을 수행한다.
  1. 사용자는 $\varepsilon = g_1(M, \delta)$를 계산하고, M

을 l개의 서버들에게 전달한다. 여기서, g_1은 기반하는 전자서명 알고리즘에 의존적으로 생성된다.

2. S_{i_j}는 자신의 비밀 분배키를 사용하여 아래와 같은 가상 부분서명문 $\xi^{(i_j)} = g_2(M, SK^{(i_j)})$를 계산한다. 그러면, k개 이상의 가상 부분서명문을 통하여, 누구나 유효한 가상 전자서명문 $\xi = g_3(\xi^{(i_1)}, \ldots, \xi^{(i_k)})$를 계산 가능하다. 여기서, $1 \le j \le k$에 대하여 $t+1 \le k \le l$, $1 \le i_j \le l$이다. g_2와 g_3는 기반하는 전자서명 알고리즘에 의존하는 알고리즘이다.
3. ε을 소유하고 있는 사용자는 가상 서명문 ξ로부터 유효한 전자서명문을 $\sigma = g_4(\xi, \varepsilon)$을 유도 가능하다. 여기서, g_4는 기반하는 전자서명 알고리즘에 의존하는 알고리즘이다. 더욱이, 필요에 따라서 g_3와 g_4는 함께 구현되어질 수 있다.

- *UCTS.Ver* : 임계 전자서명 시스템에 의해서 생성된 전자서명문의 검증 절차는 기반하는 전자서명 알고리즘의 검증절차와 동일하다.

가상 비밀키를 생성하기 위한 $\circ$ 연산자는 기반 전자서명 알고리즘에 따라서 여러 가지 방법이 존재할 수 있다. 하지만, 본 논문에서는 $+$ 연산자를 기준으로서 서술하겠다. 또한, 지금부터 제안 기법을 UCTS로 표기한다.

3.2 RSA 기반의 임계 전자서명 알고리즘으로의 적용

인수 분해 문제의 어려움에 기반한 전자서명 알고리즘으로 RSA가 있다. 따라서, RSA 기반의 임계 전자서명 기법을 UCTS로 변환하기 위하여 다음과 같은 수행한다[13,15]. 먼저, 어떤 메시지 M를 Z_n^*의 원소에 매핑하기 위해서, 어떤 해쉬 함수 H를 사용하는 것으로 가정한다. 여기서, $m = H(M)$으로 둔다.

1. d를 비밀키, δ를 제어 파라메터, SK를 가상 비밀키로 설정하자. 그러면, 사용자를 위한 가상 비밀키는 $SK = d + \delta$가 된다.
2. g_2와 g_3에 의해서 계산되어진 메시지 M에 대한 가상 전자서명문은 m^{SK}가 된다.
3. 사용자는 메시지 M에 대한 유효한 전자서명문을 아래와 같이 g_4를 사용하여 유도한다.

$$g_4(m^{SK}, g_1(m, \delta)) = g_4(m^{SK}, m^{\delta}) = \frac{m^{SK}}{m^{\delta}} = m^d$$

m^{δ}는 단지 그 사용자에 의해서만 계산 가능하기 때문에, 그 사용자만이 가상 전자서명문 m^{SK}로 부터 유효한 전자서명문 m^d를 유도 가능하다. 또한, k개 이상의 가상 서명문들과 대응되는 전자 서명문을 도청한 공격자는 이산대수의 어려움으로 인하여 제어 파라메터 δ를 얻을 수 없다. RSA에 기반한 UCTS의 실질적인 예는 부록 A에서 소개된다.

3.3 이산대수 문제 기반의 임계 전자서명 알고리즘으로의 적용

본 절에서는 Schnorr 전자서명 알고리즘을 기반한 임계 전자서명 알고리즘을 사용자 제어 가능하도록 변경한다[10]. Schnorr 전자서명 기법을 간략히 기술하면 다음과 같다. p와 q를 $q|p-1$인 큰 소수라고 두자. 그리고, g는 Z_p^*상의 위수 q를 가지는 생성자로 두고 (p, q, g)를 공개한다.

서명자는 $1 \le x \le q$를 만족하는 비밀키를 임의적으로 선택하고, $y = g^x \bmod p$를 공개키로 한다. $H: \{0,1\}^* \rightarrow Z_p$를 이상적인 랜덤 오라클로 간주되는 해쉬함수라고 둔다. 어떤 메시지 M을 서명하기 위해서, 서명자는 단명 비밀값 $k \in_R Z_q^*$를 선택하고, 아래의 수식 (1)과 같이 전자서명 (r, s)를 계산한다[1].

$$s = k + cx \bmod q, \quad c = H(M, r), \quad r = g^k \bmod p \tag{1}$$

Schnorr 전자서명 알고리즘을 기반한 UCTS은 다음과 같은 구조를 가지고 있다.

1. Schnorr 전자서명 알고리즘을 위한 가상 비밀키를 $SK = x + \delta$로 두자. 여기서, x는 실질적인 비밀키이며, $1 \le SK \le q$를 만족한다.
2. 알고리즘 g_2와 g_3에 의해서 계산되는 메시지 M에 대한 가상 전자서명문은 $vs = k + c \cdot SK \bmod q$가 된다.
3. 사용자는 g_4 알고리즘을 사용하여,

$$s = g_4(vs, g_1(M, \delta)) = g_4(vs, c \cdot \delta) = vs - c \cdot \delta \bmod q = k + cx \bmod q$$

유효한 전자서명문 (r, s)을 계산한다. 여기서, r 값은 Schnorr 전자서명 알고리즘과 동일한 값이다.

하지만, c값이 공개값이며 $c \cdot \delta = vs - s$이기 때문에, k개 이상의 가상 부분서명문들과 대응되는 전자서명문을 도청한 공격자는 제어 파라메터 δ값을 획득 가능하게 된다. 따라서, UCTS에서 제어 파라메터를 보호하기 위하여, 본 논문에서는 제어 파라메터의 노출을 방지하기 위해서[8,9]에서와 같은 2자간의 Schnorr 전자서명 기법을 사용하고자 한다. 본 논문에서 이를 위하여 새로운 2자간의 Schnorr 전자서명 기법을 제안하며, 이 기법을 사용자 제어 가능한 2자간 Schnorr 전자서명(User Controllable Two-Party Schnorr signature : UCTPS)라고 한다. 제안되는 기법은 [8, 9]에서 소개된 기법과는 다른 접근법을 가지며, [4]에서 안전한 대리 서명키를 생성하기 위해서 사용된 기법을 변형하였다.

제안된 UCTPS는 $UCTPS = (UCTPS.Init, UCTPS.Sig, UCTPS.Ver)$로 구성된다.

- $UCTPS.Init$: 서명자는 $1 \le x \le q$를 만족하는 비밀키를 임의적으로 선택하고, $y = g^x \bmod p$를 공개키로 한다. 사용자는 랜덤한 제어 파라메터 δ를 생성하고, 가상 비밀키 $SK = x + \delta$를 계산한다. 사용자는 서버에게 SK를 안전하게 전송하고, 자신의 비밀키를 시스템 메모리로부터 삭제한다.
- UCTPS.Sig : 메시지 M을 전사서명하기 위해서,
  1. 사용자는 $\tilde{k} \in_R Z_q$를 생성하고, $\tilde{r} = g^{\tilde{k}} \bmod p$를 계산한다. 사용자는 $\tilde{r}, M$을 서버에게 전송한다.
  2. 서버는 $k \in_R Z_q$를 생성하고, $r = g^k \bmod p$와 $R = \tilde{r} \cdot r = g^{\tilde{k}+k} \bmod p$를 계산한다. 서버는 가상 전자서명문 $vs = k + H(M, R) \cdot SK \bmod q$을 계산하고, 사용차에게 (r, vs)를 전송한다.
  3. 사용자는 $R = r \cdot \tilde{r}$을 계산하고, 전자서명문을 수식 (2)를 사용하여 유도한다.

$$s = vs + (\tilde{k} - H(M, R) \cdot \delta) \bmod q \tag{2}$$

즉, 사용자의 메시지 M에 대한 전자 서명문은 (R, s)가 된다. 여기서, $s = (k + \tilde{k}) + H(M, R) \bmod q$이다.

- $UCTS.Ver$: 일반적인 Schnorr 전자서명의

검증과 동일하게 수행된다.

Schnorr를 기반한 임계 전자서명 기법을 사용자 제어 가능하게 만들기 위해서, UCTPS를 UCTS에 쉽게 적용이 가능하다. 즉, UCTPS.Init에서 생성한 가상 비밀키를 UCTS.Init와 같은 형태로 l개의 서버들에게 비밀 분산시킨다. 그리고, UCTPS.Sig에서 단일 서버에서 계산된 가상 전자서명문은 UCTS.Sig와 같이 다수의 서버들에 의한 협력을 통하여 계산하게 된다. UCTPS를 통한 UCTS의 구현을 통해서, k개 이상의 가상 부분서명문들과 대응되는 전자서명문을 도청한 공격자는 사용자가 선택한 γ_k 값을 모르기 때문에 제어 파라메터를 구할 수 없다. 따라서, 악의적인 도청자로 부터 제어 파라메터의 노출을 방지할 수 있다. Schnorr 전자서명에 기반한 UCTS의 실질적인 예는 부록 B에서 소개된다.

4. 보안성

본 논문에서 제안된 기법은 아래와 같은 보안성을 제공한다.

1. 위조 불가성(Unforgeability) : 제안 기법은 암호학적으로 안전한 현존하는 전자서명 기법과 임계 전자서명 기법을 기반으로 하였다. 따라서, 만약 사용되는 임계 전자서명 기법이 위조 불가하면, 제안 방안 또한 위조 불가하다.
2. 사용자 제어가능성(User controllability) : 제안 기법은 기존의 임계 전자서명 알고리즘에서 사용자의 동의 없이 비밀키를 통한 악의적인 전자서명 및 암호문의 복호화 수행을 효율적으로 방지하지 못한 점을 해결하였다. 따라서, 사용자가 임계 전자서명의 수행을 직접적으로 제어 가능하다.
3. 빠른 취소능력(Fast revocation) : 사용자는 자신이 원할 시에 l개로 구성된 서버들의 서명 능력을 즉시 취소가 가능하다. 즉, 자신의 제어 파라메터를 통한 연산을 수행하지 않음으로써, l개의 서버들의 서명 능력을 간단히 취소 가능하다.
4. 확장성 : 제안 기법은 기존의 임계 전자서명 알고리즘에 사용자의 제어 능력을 추가한 것이다. 따라서, 기존에 제안되었던 임계 전자서명 알고리즘뿐만 아니라 새로이 제안될 임계 전자서명 알고리즘에도 쉽게 적용이 가능하다.

5. 결 론

본 논문에서 제안한 사용자 제어 가능한 임계 전자서명 기법(UCTS)은 사용자가 다수의 서버에 분산된 자신의 서명 함수들에 대한 활성화를 직접 제어 가능한 기법이다. 또한, 제안된 UCTS는 소인수 분해에 기반한 임계 전자서명 기법뿐만 아니라 이산 대수 문제에 기반한 전자서명 기법에도 적용 가능하다. 더욱이, Schnorr 전자서명을 기반한 임계 전자서명의 좀 더 안전한 응용을 위하여, 새로운 2자간의 Schnorr 전자서명 기법(UCTPS)을 제안하였다. 제안된 UCTPS는 기 제안된 2자간 전자서명과 같이 여러 응용에서도 적용 가능하다.

참 고 문 헌

[1] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, "Handbook of Applied Cryptography", CRC Press.
[2] A. Shamir, "How to Share a Secret", C. ACM, 22(1):612-613, 1979.
[3] D. Boneh, M. Franklin, "Efficient generation of

shared RSA keys", in Proceedings Crypto's 97, pp.425-439.

[4] H. Petersen and P. Horster, "Self-certified keys - Concepts and Applications", In Proc. Communications and Multimedia Security'97, pp.102-116, Chapman and Hall, 1997.

[5] M. Malkin, T. Wu, D. Boneh, "Experimenting with shared RSA key generation", Proceedings of the Internet Society's 1999 Symposium on Network and Distributed System Security (SNDSS), pp. 43-56.

[6] P.Feldman, "A Practical Scheme for Non-Interactive Verifiable Secret Sharing", In Proc. 28th FOCS, pp.427-437, IEEE, 1987.

[7] P. Fouque, J. Stern, "Fully Distributed Threshold RSA under Standard Assumptions", ASIACRYPT 2001, pp. 310-330.

[8] P. MacKenzie, M. Reiter, "Networked Cryptographic Devices Resilient to Capture", IEEE Security and Privacy'01, May 14-16, 2001.

[9] P. MacKenzie and M. Reiter. "Two-Party Generation of DSA Signatures", Crypto'01, LNCS 2139, pp.137-154, 2001.

[10] R. Gennaro, S. Jarecki, H. Krawczyk, "Revisiting the Distributed Key Generation for Discrete-Log Based Cryptosystems", RSA Security' 03, April 2003.

[11] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin. "Robust threshold DSS signatures", In Information and Computation 164, pp.54-84, 2001.

[12] Torben Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing", In Crypto '91, pp.129-140, 1991.

[13] Tal Rabin, "A Simplified Approach to Threshold and Proactive RSA", In H. Krawczyk, editor, Advances in Cryptology-CRYPTO'98, LNCS 1462, pp. 89-104, 1998.

[14] T. Wu, M. Malkin, and D. Boneh, "Building intrusion tolerant applications", In proceedings of the 8th USENIX Security Symposium, pp. 79-91, 1999.

[15] Victor Shoup, "Practical threshold signatures", In Proc. Eurocrypt 2000, LNCS 1807, pp. 207-220. 2000.

양 종 필

- 1999년 2월 부경대학교 전자계산학과 졸업
- 2001년 8월 부경대학교 전자계산학과 석사
- 2002년 3월~현재 부경대학교 전자계산학과 박사과정
- 관심분야 : 네트워크 및 시스템 보안 기술, 공개키 기반 구조, 비밀 분산

이 경 현

- 1982년 2월 경북대학교 수학 교육과 졸업
- 1985년 2월 한국과학기술원 응용수학과 석사
- 1992년 8월 한국과학기술원 수학과 박사
- 1985년 2월~1993년 2월 한국전자통신연구소 연구원, 선임연구원
- 1995년 7월~1996년 7월 호주 애들레이드 대학, Post Doc.
- 1999년 7월~1999년 8월 일본 동경대학 생산기술연구소, Visiting Scholar
- 2001년 7월~2002년 8월 미국 Univ. of California, Irvine 방문교수
- 2002년 9월~2003년 7월 필리핀 콜롬보기술 교육자대학, Faculty Consultant
- 1999년 12월~현재 한국멀티미디어학회, 학술이사, 운영위원 역임, (현) 재무이사, 논문지 편집위원
- 1999년 12월~현재 한국정보보호학회 논문지 편집위원, 국제이사
- 1993년 3월~현재 부경대학교 전자컴퓨터정보통신공학부 교수
- 관심분야 : 암호이론, 암호프로토콜, 네트워크보안, 이동네트워크, 그룹키 관리

부 록

A. 사용자 제어 가능한 V.Shoup의 기법

Victor Shoup이 [15]에서 발표한 RSA 알고리즘을 기반한 임계 전자서명 알고리즘을 사용자 제어 가능하도록 적용하면 아래와 같다.

UCTS.Init

1. 사용자는 랜덤한 두 개의 큰 소수 p와 q를 생성한다. 여기서, $p=2p'+1$, $q=2q'+1$이며, p'와 q' 또한 소수이다. 그리고, $n=p\cdot q$과 $m=p'\cdot q'$을 계산한다.
2. 사용자는 소수인 RSA 공개키로 임의적으로 e를 선택하고, $e\cdot d\equiv 1 \bmod m$를 만족하는 $d\in Z$를 계산하여 RSA 비밀키로 취한다.
3. 사용자는 $a_0=d$로 설정하고, 제어 파라메터 δ를 설정한다. 그리고, $\{0,\cdots,m-1\}$범위에서 $k-1$개의 a_i 값을 랜덤하게 선택한다. 선택된 $k-1$개의 a_i값들과 δ값을 사용하여 다항식을 아래와 같이 구성한다.

$$F(X) = (a_0+a_1x+a_2x^2+\cdots+a_{k-1}x^{k-1})+\delta \in Z[x]$$
$$= \left(\sum_{i=0}^{k-1} a_ix^i \right)+\delta \in Z[x]$$

4. $L(n)$을 n의 비트 길이로 두자. 그리고, 사용자는 $1\le i\le l$인 모든 $sk_i=F(i) \bmod m$를 계산한다. sk_i는 서버 i의 가상 비밀 분배키이다.
5. Q_n을 Z_n^* 상에서 제곱한 수로 이루어진 부분군(subgroup)으로 두자. 사용자는 랜덤한 $v\in Q_n$을 선택하고 $1\le i\le l$에 대해서 $v_i=v^{sk_i}\in Q_n$을 계산한다. 따라서, i번째 서버를 위한 검증키를 $VK=v$와 $VK_i=v_i$로 설정한다. 여기서, $\Delta=l!$로 두자. k 개의 서버로 구성된 어떤 서버 집합 S가 존재하며, 어떤 $i\in\{0,\cdots,l\}\backslash S$ 와 $j\in S$에 대하여 수식 (3)를 정의 가능하다.

$$\lambda_{i,j}^S = \Delta\frac{\prod_{j'\in S\backslash\{j\}}(i-j')}{\prod_{j'\in S\backslash\{j\}}(j-j')} \in Z \quad (3)$$

그리고, Lagrange 보간 다항식으로부터, 수식 (4)를 얻을 수 있다.

$$\Delta\cdot F(i) = \sum_{j\in S}\lambda_{i,j}^S F(j) \bmod m \quad (4)$$

UCTS.Sig : 전자서명하기 위한 어떤 메시지 M을 Z_n^*상의 요소로 매핑하기 위하여, 일방향 해쉬함수 H를 사용한다. 만약 $x=H(M)$이면, M에 대한 유효한 전자서명 $y^e=x$를 만족하는 $y\in Z_n^*$가 될 것이

다.

1. 사용자는 메시지 M을 전자서명 하기 위해서, $x=H(M)$와 $\varepsilon=x^{\delta} \bmod n$을 계산하고, x를 서버 S_i에게 전송한다. 여기서, $1\le i\le l$이다.
2. 각 S_i는 가상 부분서명문 $x_i=x^{2\Delta sk_i}\in Q_n$ 을 계산한다. 또한, 자신이 생성한 가상 부분서명문의 유효성 검증을 위해서 "올바름의 증명"을 계산한다.
   ① $\tilde{x}=x^{4\Delta}$로 두자. S_i는 어떤 랜덤한 수 $r\in\{0,\cdots,2^{L(n)+2L_1}-1\}$를 선택하고, $v'=v^r$, $x'=\tilde{x}^{r}$, $c=H'(v,\tilde{x},v_i,x_i^2,v',x')$, $z=sk_ic+r$ 를 계산한다. 여기서, H'는 L_1 비트를 출력하는 일방향 해쉬함수이다.
   ② 따라서, 각 S_i에 대한 "올바름의 증명"은 (z, c)로 구성된다.
3. 각 S_i는 자신의 가상 부분서명문 x_i와 (z, c)를 사용자에게 전송한다.
4. 사용자는 가상 부분서명문을 결합하고, 최종적으로 유효한 전자서명문을 아래의 단계를 거쳐서 유도한다.
   ① "올바름의 증명"을 검증하기 위해서, 사용자는 $c\,?=H'(v,\tilde{x},v_i,x_i^2,v^zv_i^{-c},\ \tilde{x}^zx_i^{-2c})$를 검사한다.
   ② 사용자는 위의 검사를 통과한 서버들의 집합 $S=\{i_1,\cdots,i_k\}\subset\{1,\cdots,l\}$를 구성한다.
   ③ 사용자는 아래의 수식를 계산한다.

$$w=\frac{x_{i_1}^{2\lambda_{0,i_1}^{S}}\cdots x_{i_k}^{2\lambda_{0,i_k}^{S}}}{x^{4\Delta^2\delta}}=x^{4\Delta^2d}$$

여기서, λ는 수식 (3)에서 정의된 정수이다. 그리고, 수식 (4)로부터 $e'=4\Delta^2$일 때, $w^e=x^{e'}$임을 알 수 있다.

   ④ $\gcd(e', e)=1$이기 때문에, $y^e=x$를 만족하는 전자 서명문 y를 계산 가능하다. 즉, $y=w^ax^b$라고 두고, 여기서 a와 b는 $e'a+eb=1$을 만족하는 수, 확장 유클리드 알고리즘을 통해서 계산 가능하다.

UCTS.Ver : 일반적인 RSA 전자서명문과 동일하다.

B. 사용자 제어 가능한 R.Gennaro의 기법

R.Gennaro가 제안한 [10]에서는 사용자의 비밀키는 l 개의 서버들에게 덧셈적으로 비밀 분산(additive secret sharing)된다. 따라서, (k, l)-임계 전자서명을 구성하기 위해서, [10]과 [12]에서 제안된 분산된 키 생성(Distributed Key Generation : Ped-DKG)와 [6]에서 제안된 검증 가능한 비밀 분산(Verifiable Secret Sharing : Fel-VSS) 기법들을 사용한다. [10]에서 제안된 기법을 사용자 제어 가능하게 만들기

위해서, 3.2절에서 소개된 UCTPS를 사용한다. 유효한 서명문을 생성하기 위한 절차는 아래와 같다.

UCTS.Init

생성된 비밀키 x를 비밀 분산하기 이전에, 사용자는 가상 비밀키, $SK = x + \delta$, 를 계산하여, 덧셈적 비밀 분산을 수행한다. 즉, 각 서버 S_i는 SK에 대한 덧셈적 가상 비밀 분배키 sk_i와 대응되는 공개 정보인 $y_i = g^{sk_i}$ 값을 소유하게 된다.

UCTS.Sig : 어떤 메시지 M을 전사서명하기 위해서,

1. 사용자는 $\tilde{k} \in {}_R Z_q$를 생성하고, $\tilde{r} = g^{\tilde{k}} \bmod p$를 계산한다. 사용자는 $\tilde{r}$, M을 서버에게 전송한다.
2. 서버는 Ped-DKG를 수행하여, 각 서버는 단명 비밀값 k에 대한 덧셈적 분배값 k_i를 소유하게 되며, 각 k_i값들은 Fel-VSS 기법을 통하여 또 다시 비밀 분산된다. 따라서, 최종적으로 각 S_i에 대한 공개 값으로 $r = g^r$, $R = \tilde{r} \cdot r$ 과 $r_i = g^{k_i}$가 생성된다.
3. 각 서버는 지역적으로 $c = H(M, R)$을 계산한다. 그리고, 각 서버 S_i는 덧셈적 가상 부분서명문 $vs_i = k_i + c \cdot sk_i \bmod q$를 계산하고, 사용자에게 전송한다.
4. 각 덧셈적 가상 부분서명문들은 $g^{vs_i} = r_i \cdot y_i^c$를 만족하는 가를 검사 받는다. 만약 검사가 성공적이면, 사용자는 가상 서명문을 $vs = vs_1 + \cdots + vs_l$과 같이 계산한다. 최종적으로 사용자는 메시지 M에 대한 유효한 전자서명문 (R, s)를 아래와 같이 계산한다.

$$s = vs + (\tilde{k} - H(M, R) \cdot \delta) \bmod q, \quad R = r \cdot \tilde{r}$$

만약, 위의 검사가 실패하면, sk_i와 k_i는 재구성되어, vs_i가 공개적으로 계산된다.

UCTS.Ver : 일반적인 Schnorr 전자서명 기법의 검증절차와 동일하다.