매직 잉크 서명 기법을 이용한 전자 현금 프로토콜 설계

[0]백종현, 염흥열
순천향대학교 전기전자공학부

Electronic Cash Protocol Using the Magic Ink Signature

Jong-Hyun Baek, Heung-Youl Youm
Dept. of Electrical and Electronic Engineering, SoonChunHyang Univ.
E-mail:jhwhite@unitel.co.kr, hyyoum@asan.sch.ac.kr

Abstract

Chaum's blind signature scheme is the typical withdrawal procedure of electronic cash. In blind signature scheme, a signer sign a document while he never knows the content of the signed document. Yung and Jakobsson[3] presented new signature scheme with which the content of document can be unblinded if unusual activity is detected. This signature is referred to as the magic ink signature. In this paper, we analyze the magic ink signature, and present two new magic ink signature schemes using KCDSA and Schnorr signature algorithm. We propose two types of the efficient electonic cash system using these magic ink signature schemes. One is the electronic cash system with a single server magic ink signature scheme, and the other is the electronic cash system with a distributed magic ink signature scheme.

제1장 서 론

전자 현금 시스템에서의 인출 과정은 D.Chaum이 제안한 은닉 서명 방식을 많이 사용하였다. 은닉 서명 방식은 수신자만이 서명된 문서를 가질 수 있고, 서명자는 자신이 서명한 문서의 내용을 알 수 없도록 구성되어 있다. 이 단점을 보완하기 위한 서명 기법이 매직 잉크 서명 방식이다. 그러나 사용자가 범죄 행위나 불법 행동을 하면 서명자는 서명문의 내용을 확인할 필요가 있다. 매직 잉크 서명 방식은 서명자가 서명문의 내용을 노출(unblind)하고 싶으면 적법한 절차를 거쳐 매직 잉크를 현상하여 서명된 문서의 내용을 확인할 수 있도록 구성된다. 매직 잉크 서명 방식은 비밀 공유와 임계치(threshold) 기법이 결합된 다중 서명자에 의해 서명되는 분배된 매직 잉크 서명 기법(distributed magic ink signature scheme)

으로 확장될 수 있다. 여기서는 정족수 이상의 서버들이 협력해야만 서명문을 생성할 수 있고 또 필요한 경우에 서명문의 내용을 노출(unblind)할 수 있다. 본 논문에서는 기존의 매직 잉크 서명 기법과 매직 잉크 서명 방식에서 사용된 보안 기법 및 툴을 분석하고, Schnorr 서명 기법과 KCDSA(Korea Certificate-based Digital Signature Algorithm)을[13] 사용한 새로운 매직 잉크 서명 방식을 제안한다. 그리고 제안된 매직 잉크 서명 방식을 사용한 지불 과정을 포함하는 전자 현금 프로토콜을 제시한다.

본 논문의 2장에서는 매직 잉크 서명 기법의 개념과 단일 서버를 채용한 매직 잉크 서명방식과 다수 서버가 요구되는 분배된 매직 잉크 서명 방식을 제시하고, 3장에서는 매직 잉크 서명 생성 알고리듬에 사용되는 보안 기법과 보안 툴들을 제시한다. 그리고 4장에서는 Schnorr 서명 기법과 KCDSA 을 사용한 새로운 매직 잉크 서명문 생성 프로토콜을 제시한다. 5장에서는 새로운 매직 잉크 서명 방식을 사용한 두 종류의 전자 현금 시스템을 설계한다.

제2장 매직 잉크 서명 방식의 개념 및 알고리듬

2.1 매직 잉크 서명 방식

매직 잉크 서명 방식[3]은 다음과 같은 단계를 거쳐 서명이 수행된다. 먼저 브라인드 서명 방식과 같이 문서와 묵지를 봉투에 넣는다. 여기서 문서는 아직 쓰여지지 않는 상태이다. 수신자는 봉투 위에 매직 잉크를 사용하여 문서를 쓴다. 매직 잉크는 현상이 된 후에만 내용을 알 수 있는 특수 잉크이다. 그리고 서명자는 봉투에 서명한다. 묵지 복사에 의해 문서의 내용이 내부 종이에 쓰여진다. 수신자는 내부 종이를 갖고 서명자는 봉투를 갖는다. 따라서 수신자는 서명문을 얻을 수 있다. 서명자는 문서의 노출(unblind)이 필요하면 매직 잉크를 현상하여 서명문의 내용을 구한다. 정당한 수신자의 서명문은 안전하지만, 범죄 행위자나 다른 부정한 행위자들의 서명문은 노출(unblind) 가능한 서명 기법이다.

2.2 단일 서버 매직 잉크 서명 방식

단일 서버 매직 잉크 서명 방식은 다음과 같다.

① 수신자 R은 서명을 원하는 해쉬된 메시지(H(M))인 $m \in Z_q$ 을 선택한다. 그리고 두 개의 은닉 요소 $a, b \in_u Z_q$ 를 생성하고, m 의 브라인딩($\mu = ma \bmod q$)을 계산한다. 그리고 수신자는 서명문 생성 서버 S로 μ를 보낸다.

② S는 랜덤 비밀 세션키 $k' \in_u Z_q$ 를 생성하고, $r' = g^{k'} \bmod p$ 를 계산하여 서명문 수신자 R 로 전송한다.

③ R은 $r = [r'^{b} \bmod p] \bmod q$ 를 계산한다. 그리고 r의 브라인딩 요소

$\rho = ra \bmod q$ 를 계산하여 서명문 생성 서버로 전송한다.

④ S는 추후 서명문 및 사용자 노출을 위해 요구되는 *tag* 와, 공개 세션키 ρ 를 사용하여 메시지 μ의 DSS(digital signature standard) 서명문 σ 를 생성한다.[3] 여기서 *tag* 가 먼저 계산되어지고, 나중에 $\sigma = k'(\mu + x\rho) \bmod q$ 가 계산된다. 서버는 R로 σ 를 전송한다.

⑤ 수신자 R은 서명문 s 를 계산한다. 여기서 $s = (\sigma a^{-1}b^{-1}) \bmod q = k'(m + xr)b^{-1}$ 이다.

2.3 분배된 매직 잉크 서명 방식

본 절에서는 단일 서버 매직 잉크 서명 방식에 분배(distributed) 개념을 추가한 분배된 매직 잉크 서명방식을 분석한다. 단일 서버 매직 잉크 서명 방식과는 달리 분배된 매직 잉크 서명 방식에서는 서버 정족수의 협력이 있어야 서명문을 생성할 수 있고 정족수가 협력하면 서명문을 노출할 수 있다. 분배된 매직 잉크 생성 프로토콜은 다음과 같다. 여기서 Q를 S1 ,..., S_n 내의 t 개의 서버들의 집합인 정족수라 한다.

① 서명문 수신자 R은 서명을 원하는 메시지 $m \in Z_q$ 를 갖는다. 수신자는 두 개의 은닉 요소 $a, b \in_u Z_q$ 를 생성하고, m의 브라인딩 $\mu = ma \bmod q$ 를 구하고 μ 의 (t,n) 비밀 공유 $(\mu_1, \cdots, \mu_n)$ 과 공개 정보 $(g^{\mu_1}, \cdots, g^{\mu_n})$ 를 계산한다. R은 서명문 생성 서버 S_i 에게 비밀 채널을 통해 μ_i 를 전송한다.

② 서버의 집합 $S_i \mid i \in Q$ 는 분배적으로 랜덤 비밀 세션키 $k' \in_u Z_q$를 생성한다. 각 서버 S_i는 공유 k'_i를 갖는다. 그리고 서버 S_i는 $g^{k'_i} \bmod p$ 를 공개하고, 서명문 수신자 R 에게만 보내는 $r' = g^{k'^{-1}} \bmod p$ 을 계산한다.

③ 서명문 수신자 R은 $r = [[r'^{b} \bmod p] \bmod q]$를 계산하고, r을 $\rho = ra \bmod q$ 로 은닉한다. R은 공개정보가 $(g^{\rho_1}, \cdots, g^{\rho_n})$인 ρ의 (t,n) 비밀 공유 $(\rho_1, \cdots, \rho_n)$를 계산한다. R은 S_i에게 ρ_i를 보낸다.

④ 서버의 집합 $S_i \mid i \in Q$는 분배적으로 *tag*와 메시지 μ 의 DSS 서명문 σ 를 생성한다. 여기서 *tag*를 먼저 계산하고 그 후에 다음과 같은 σ 를 계산한다. 즉 S_i는 $\sigma_i = \overline{k_i}\,(\mu_i + x_i\,\rho_i) \bmod q$ 를 계산한다. 그때 $\sigma = \overline{k}\,(\mu + x\,\rho) \bmod q$ 는 비밀의 곱셈 방법[10] 을 사용하여 σ_i 로부터 interpolate 된다. 서버 S_i는 R로 σ 를 보낸다.

⑤ 서명문 수신자 R은 서명문 s를 계산한다. 여기서 $s = \sigma\, a^{-1}b^{-1} \bmod q$ 이다. 여기서 (r,s) 는 m의 정당한 DSS 서명문이다.

제3장 매직 잉크 방식에 사용된 기법 및 툴

3.1 비밀 공유 기법[2]

Shamir는 임계치를 갖는 매직 잉크 서명 방식을 적용하기 위하여 유한체상의 다항식 방정식을 이용하였다. (k,n)-threshold 비밀 공유 기법은 n 참여자들에게 비밀에 관한 부분 정보를 분배하는 신뢰성있는 dealer와 나머지 n 명의 참여자간에 수행된다. 여기서는 k 보다 작은 참여자들의 그룹은 비밀에 관한 어떠한 정보도 얻을 수 없지만 적어도 k 참여자들의 그룹은 다항시간(polynomial time) 내에 비밀을 계산할 수 있다. 예를 들어, M을 재구성하는데 3명의 참여자가 필요한 k=3 인 (3,n)-threshold 기법을 원한다면, k-1 차수인 2차 다항식을 생성한다.

$$F(x) = (ax^2 + bx + M) \bmod p$$

여기서 p는 다른 계수보다 큰 랜덤 소수이다. 계수 a와 b는 랜덤하게 선택되어진다. 계수 a와 b는 분배 과정 동안만 비밀스럽게 유지되고 있다가 각 참여자의 부분 비밀인 shadow가 분배된 후 버려진다. M은 메시지이고, 소수 p는 공개된다. shadow는 아래와 같이 n개의 다른점들에서 다항식 값을 구함으로서 얻어진다.

$$k_i = F(x_i)$$

첫 번째 shadow 인 k_1 은 상기 다항식에 x=1를 대입한 결과값이고, 두 번째는 x=2 인 값을 대입한 결과값이다. 2차 다항식은 3개의 미지 계수 a, b, M 을 갖기 때문에, 3개 shadow를 이용하면 3개의 미지 계수들을 결정할 수 있다. 예를 들어 M을 11이라 가정한다. (3,5)-threshold 기법을 구성하는데 먼저 아래 식과 같은 2차 방정식을 생성한다. ($a = 7$ 과 $b = 8$이 랜덤하게 선택된다)

$$F(x) = (7x^2 + 8x + 11) \bmod 13$$

5개의 shadow는 아래와 같다.

$$k_1 = F(1) = 7 + 8 + 11 \equiv 0 (\bmod 13)$$

$$k_2 = F(2) = 28 + 16 + 11 \equiv 3 (\bmod 13)$$

$$k_3 = F(3) = 63 + 24 + 11 \equiv 7 (\bmod 13)$$

$$k_4 = F(4) = 112 + 32 + 11 \equiv 12 (\bmod 13)$$

$$k_5 = F(5) = 175 + 40 + 11 \equiv 5 (\bmod 13)$$

3개의 shadow (k_2, k_3, k_5) 로부터 M을 재구성하는 것은 아래와 같이 선형방정식을 이용하

여 구한다.

$$a \cdot 2^2 + b \cdot 2 + M \equiv 3 (\mathrm{mod}\, 13)$$
$$a \cdot 3^2 + b \cdot 3 + M \equiv 7 (\mathrm{mod}\, 13)$$
$$a \cdot 5^2 + b \cdot 5 + M \equiv 5 (\mathrm{mod}\, 13)$$

결과는 $a = 7,\ b = 8,\ M = 11$이다. 따라서 비밀 정보 M을 구할 수 있다. 이 공유 기법은 큰 수들에도 쉽게 실현되어진다. 만약 6명이 모여서 메시지를 구할 수 있는 30개의 부분으로 메시지를 분배하고 싶다면, 30명에게 각각 다음 식과 같은 5차 다항식의 shadow 값을 분배한다.

$$F(x) = ax^5 + bx^4 + cx^3 + dx^2 + ex + M \ (\mathrm{mod}\, p)$$

6명은 M를 포함한 6개의 알려지지 않은 값을 구할 수 있다. 5명은 M에 대한 어떠한 것도 알 수 없다.

3.2 검증할 수 있는 비밀 공유 (VSS)[6],[10]

Shamir의 비밀 공유 기법과 같이 이 비밀 공유 기법도 각 참여자 P_i에게 비밀 공유 s_i를 분배한다. 하지만 이 검증할 수 있는 비밀 공유 기법에서는 각 참여자가 자신의 공유의 정확성을 검증할 수 있다. 먼저 dealer는 $f(0) = s$ 를 만족하는 k-1 차수의 다항식 f(x)를 선택하고 $(s_1, \ldots, s_n) \xleftrightarrow{(t,n)} s \bmod q$ 가 되는 공유 s_i를 생성한다.

여기서, $f(x) = \sum_j a_j x^j$ 이 되면, dealer는 $\alpha_j = g^{a_j} \bmod p$를 참여자로 각각 전파한다. 그리고 $g^{s_i} = \prod \alpha_j^{x^j}$ 를 검사하는 것으로 s_i가 실제 정의된 비밀인지를 검사한다.

3.3 연합 난수 비밀 공유(J_RSS: Joint Randomized Secret Sharing)[6],[7]

연합 난수 비밀 공유기법에서는 참여자들이 랜덤값의 (t,n)-비밀 공유에 대응되는 공유를 총괄적으로 각각 선택한다. 즉 자신의 비밀 공유 α_i 를 선택하여, (t,n)-비밀 공유 기법을 이용하여 각 참여자에게 부분 비밀 α_{ij} 을 분배한다. 또 각 참여자들은 수신된 부분 정보를 합하여 비밀 s 에 대응되는 자신의 부분 비밀 공유 s_i 를 계산한다. 이렇게 함으로서 각 참여자들은 $(s_1, \ldots, s_n) \xleftrightarrow{(t,n)} s \bmod q$ 가 되는 비밀 공유 s_i 를 갖는다. 일반적인 (t,n)-비밀 공유 기법에서 처럼 비밀 s 값은 각 참여자는 물론 t-1 참여자들이 연합하여도 비밀이 유지된다. 이 프로토콜에서, 모든 참여자들은 자신들이 선택하는 랜덤 지역 비밀에 대한 dealer 들처럼 행동해야 한다. 최종 공유 s_i (i = 1,....,n)는 각 참여자에 의해 P_i에게 분배된 부분 비밀 공유의 합이다. 따라서, 공동 비밀 s 는 모든 분배된 비밀들의 합과 같다.

3.4 공동 영(Zero) 비밀 공유[3],[10]

이 프로토콜은 비밀값이 영인 총괄적으로 공유를 생성한다. 이 프로토콜은 3.3절의 공동 난수 비밀공유 프로토콜과 비슷하지만 지역 랜덤 비밀 대신 각 참여자들이 영값의 공유를 분배한다. 영값에 대한 분배의 정당성은 각 분배 다항식들의 상수항 p_0 가 '0' 이라는 것을 검사하여 수행된다. 즉, $g^{p_0} = 1$ 인가를 검사한다. 비밀 s 의 현행 공유에 영 공유를 추가하는 것으로, 비밀을 바꾸지 않고, 비밀 s 의 공유의 랜덤화를 얻을 수 있다.

3.5 역원 계산[10]

$P_1, \ldots, P_n$ 인 n 참여자들 사이에 공유되는 주어진 비밀 $k \bmod q$는 k 와 k^{-1}을 보이지 않고 $k^{-1} \bmod q$값의 공유를 생성한다. 각 참여자 P_i 는 k 의 (t,n) 비밀공유에 대응되는 공유 k_i 를 갖는다. 즉, $(k_1, \ldots, k_n) \xleftrightarrow{(t,n)} k$ 이다. k^{-1} 을 위한 공유 계산은 다음과 같다.

① 참여자는 공동 난수 비밀 공유 프로토콜을 사용하여 랜덤 요소 a의 (t,n) 비밀공유를 공동으로 계산한다. $a_1, \ldots, a_n$ 에 의한 결과 공유를 $(a_1, \ldots, a_n) \xleftrightarrow{(t,n)} a$ 로 표시한다.

② 참여자들은 (2t,n) 공동 영 비밀공유 프로토콜을 수행한 후, 각 참여자 P_i 는 비밀 '0'의 공유 b_i 를 갖는다. Interpolation 다항식은 2t 차수이다.

③ 참여자들이 $k_i a_i + b_i$ 값을 전파하고, 2t 차수 다항식에 대응되는 interpolating으로 $\mu = ka$ 값을 재구성한다.

④ 각 참여자들은 $u_i = \mu^{-1} a_i \bmod q$ 로서 k^{-1} 의 공유 u_i 를 계산한다. 단계 1의 $(a_1, \ldots, a_n) \xleftrightarrow{(t,n)} a$ 에 의해 k^{-1} 값을 재구성할 수 있다.

$$(k^{-1}a^{-1})a_i = (k^{-1}a^{-1})a = k^{-1}$$

위의 프로토콜을 이용하면 k 와 k^{-1}을 보이지 않고 k 의 역원 k^{-1} 을 구할 수 있다.

3.6 두 비밀의 곱셈[10]

참여자들 사이에 공유된 주어진 두 비밀 u와 v 가 있다. 이 두 비밀의 곱 uv 를 계산한다. 이때 이 두 실제 비밀 값은 비밀스럽게 유지된다. 각각 t 차수의 다항식에 의해 공유된 주어진 u, v 는 각 참여자들이 자신들의 u, v 공유를 지역적으로 곱한다. 결과는 2t 차수 다항식의 uv 의 공유가 될 것이다. 따라서, 값 uv 는 2t+1의 정당한 공유들의 집합으로부터 재구성된다. t = 3인 경우, u와 v 는 3차 다항식이 되고, 이 두 비밀의 곱은 6차 다항식이 된다. 즉 7 개의 정당한 공유들의 집합으로 비밀이 재구성된다. 공동 영 비밀공유 프로토콜을 사용하는 추가적인 재랜덤화 절차는 곱해진 비밀들의 비밀을 보호하는데 필요하다. 이

랜덤화는 t 차수의 두 다항식의 곱인 2t 차수의 다항식이 랜덤 다항식이 아니고, u와 v에 대한 정보를 드러낼 수 있기 때문에 필수적이다.

제4장 새로운 매직 잉크 서명방식

본 장에서는 한국 표준 인증서-기반 디지털 서명 알고리듬과 Schnorr 디지털 서명 알고리듬을 사용한 매직 잉크 서명방식에 대해 설명한다.

4.1 단일 서버 KCDSA 매직 잉크 서명 방식

본 절에서는 참고문헌 [13]를 변경한 KCDSA를 이용한 단일 서버 매직 잉크 서명 기법을 제시한다. 여기서, [x]p 는 x mod p를 의미하고, Z 는 공개키 증명서의 해쉬값이다. 프로토콜의 수행으로 얻어지는 서명문은 (R,S) 이다.

R		S
		$k \in_R Z_q$
$a \in_R Z_q$	$\xleftarrow{W}$	$W = [g^k]_p$
$M = h(m)$		
$H = h(Z, M)$		
$H' = [H/a]_q$		
$R = h(W^a)$		
$R' = [R/a]_q$	$\xrightarrow{H', R'}$	tag $[c(H'/R')]_q$
		$E' = H' \oplus R'$
	$\xleftarrow{S'}$	$S' = [x(k-E')]_q$
$S = [S'a]_q$		

4.2 단일서버 Schnorr 매직 잉크 서명방식

본 절에서는 Schnorr 서명 기법을 이용한 단일 서버 매직 잉크 서명 기법을 제시한다. 프로토콜의 수행으로 얻어지는 서명문은 (E,S) 이다.

R		S
		$k \in_R Z_q$
	$\leftarrow R'$	$R' = [g^k]_p$
$a \in_R Z_q$		
$M = h(m)$		
$\mu = [M/a]_q$		
$R = [[R'^a]_p]_q$		
$E = h(R, M)$		
$\epsilon = [E/a]_q$	$\rightarrow \mu, \epsilon$	
		tag 계산
	$\leftarrow S'$	$S' = [k + x\epsilon]_q$
$S = [S'a]_q$		

4.3 분배된 KCDSA 매직 잉크 서명 방식

본 절에서는 KCDSA를 이용한 분배된 다수 서버 매직 잉크 서명 기법을 제시한다. 프로토콜의 수행으로 얻어지는 서명문은 (R,S) 이다. k_i 는 J_RSS 방식으로 공유된다.

R		S_i
		공동 k_i 생성
	$\leftarrow W$	$W = [g^k]_p$
$a \in_R Z_q$		
$M = h(m)$		
$H = h(Z, M)$		
$H' = [H/a]_q$		
H'의 비밀공유 $(H'_1, \cdots, H'_n)$		
$R = h(W^a)$		
$R' = [R/a]_q$		
R'의 비밀공유 $(R'_1, \cdots, R'_n)$	$\rightarrow H'_i, R'_i$	tag 계산
		$E' = H' \oplus R'$
		$S'_i = [x_i(k_i - E'_i) + c_i]_q$
		$S' = [x(k - E')]_q$
	$\leftarrow S'$	
$S = [S'a]_q$		

4.4 분배된 Schnorr 매직 잉크 서명 방식

본 절에서는 Schnorr 서명 기법을 이용한 분배된 다수 서버 매직 잉크 서명 기법을 제시

한다. 프로토콜의 수행으로 얻어지는 서명문은 (E,S) 이다.

R		S_i
		공동 k_i 생성
$a \in_R Z_q$	$\xleftarrow{R'}$	$R' = [g^k]_p$
$M = h(m)$		
$\mu = [M/a]_q$		
μ의 (t,n)비밀공유 $(\mu_1, \cdots, \mu_n)$		
$R = [[R'^b]_p]_q$		
$E = h(R, M)$		
$\varepsilon = [E/a]_q$		
ε의 (t,n) 비밀공유 $(\varepsilon_1, \cdots, \varepsilon_n)$	$\xrightarrow{\mu_i, \varepsilon_i}$	
		tag 계산
		$S'_i = [(k_i + x_i\varepsilon_i) + c_i]_q$
	$\xleftarrow{S'}$	$S' = [k + x\varepsilon]_q$
$S = [S'a]_q$		

제5장 매직 잉크 서명기법을 이용한 전자현금 프로토콜

5.1 시스템의 셋업

은행은 랜덤하게 다섯 가지 요소 (g, h, g_1, g_2, d) 를 선택한다.[8] 사용자(U)는 은행에 계정을 열고, 랜덤 수 u_1, u_2 를 생성하고, $I = g_1^{u_1} g_2^{u_2}$ 를 계산한다. 은행은 (u_1, u_2) 를 알지 못한 상태에서 I 와 함께 사용자의 식별자와 계정 번호를 저장한다. 사용자가 이중 사용을 한다면, 은행은 (u_1, u_2)를 찾을 수 있고, $I = g_1^{u_1} g_2^{u_2}$ 의 계산으로 사용자의 계정과 식별자를 확인할 수 있다.

5.2 단일 서버 KCDSA 매직 잉크 서명 방식을 사용한 발행 프로토콜

① 은행은 적절한 돈을 사용자의 계정으로부터 빼고, 랜덤 비밀 세션키 $k \in_R Z_q$ 를 생성하고, $W = [g^k]_p$ 을 계산하여 사용자로 보낸다.

② 사용자는 랜덤수 $s \in_R Z_q$을 선택하고, $m = I \cdot d$ 을 s 멱승한 $m' = m^s = g_1^{u_1 s} g_2^{u_2 s} d^s$을 계산한다. m' 을 $A = g_1^{x_1} g_2^{y_1} d^{z_1}$와 $B = m^s / A$ 두 부분으로 나눈다.[8] 그리고 해쉬 함수를 사용하여 $H = h(Z, m', A)$ 를 계산한다. 랜덤수 a 를 선택하고, $H' = (H/a) \bmod q$, $R = h(W^a)$, 그리고 $R' = (R/a) \bmod q$를 계산한다. 사용자는 은행에게 H', R' 을 보낸다.

③ 은행은 tag를 계산하고, $E' = H' \oplus R'$과 $S' = [x(k - E')]_q$를 계산한다. 이때 은행은 사용자에게 S' 을 보낸다.

④ 사용자는 S' 을 사용하여 S 를 계산한다. 여기서 $S = S'a \bmod q$ 이다. 사용자는 3쌍 $(A, B, sign(m') = (R, S))$을 갖는 것으로 발행 프로토콜을 완수한다.

5.3 단일 서버 Schnorr 매직 잉크 서명 방식을 사용한 발행 프로토콜

① 은행은 적절한 돈을 사용자의 계정으로부터 인출하고, 랜덤 비밀 세션키 $k \in_R Z_q$를 생성하고, $R' = [g^k]_p$ 을 계산한다. 이때 은행은 사용자에게 R' 을 보낸다.

② 사용자는 랜덤수 $s \in_R Z_q$를 선택하고, m 을 s 멱승한 $m' = m^s = g_1^{u_1 s} g_2^{u_2 s} d^s$ 을 계산한다. 그리고 m'을 A, B 두 부분으로 나눈다. 랜덤 수 $a \in_R Z_q$를 선택하여, $\mu = [m'/a]_q$ 와 $R = [R'^a]_p]_q$ 을 계산한다. 그리고 $E = h(R, m', A)$와 $\varepsilon = [E/a]_q$ 를 계산하여 은행에게 보낸다.

③ 은행은 tag를 계산한다. 그리고 $S' = [k + x\varepsilon] \bmod q$ 를 계산하여 사용자에게 보낸다.

④ 사용자는 S'을 사용하여 S를 계산한다. 여기서 $S = S'a \bmod q$ 이다. 사용자는 3쌍 $(A, B, sign(m') = (E, S))$ 을 갖는 것으로 발행 프로토콜을 완수한다.

5.4 분배된 KCDSA 매직 잉크 서명 방식을 사용한 발행 프로토콜

Q를 B_1 ,..., B_n 내의 t 개의 서버들로 구성된 정족수라 한다. 초기값으로 B_i 는 t 차수의 다항식 $F(\cdot)$을 통해 공유된 비밀값 x 의 공유 x_i 를 갖는다.

① 은행은 적당량의 돈을 사용자의 계정으로부터 인출하고, 은행 서버의 집합 $B_i \in Q$는 J_RSS을 이용하여 랜덤 비밀 정보 $k \in_R Z_q$를 생성한다. 이때 각 서버 B_i 는 공유 k_i 를 갖는다. 그리고 은행 서버의 집합 $B_i \in Q$는 $W = [g^k]_p$ 을 계산하고 사용자로 W 을 보낸다.

② 사용자는 랜덤수 $s \in_R Z_q$ 을 선택하고, m 을 s 멱승한 $m' = m^s = g_1^{u_1 s} g_2^{u_2 s} d^s$ 을

계산한다. 사용자는 m'을 A, B, 두 부분으로 나눈다. a 를 랜덤하게 선택하고, $H=h(Z, m', A)$, $H' = [H/a]_q$, $R = h(W^a)$, 그리고 $R' = [R/a]_q$ 를 계산한다. 이때 사용자는 분배적으로 공개정보가 $(g^{H'_1}, \cdots, g^{H'_n})$ 인 H'의 (t,n)-비밀 공유 $(H'_1, \cdots, H'_n)$와 공개 정보가 $(g^{R'_1}, \cdots, g^{R'_n})$ 인 R' 의 (t,n)-비밀공유 $(R'_1, \cdots, R'_n)$를 계산한다. 이때 사용자는 각 B_i로 H'_i, R'_i 을 보낸다.

③ 모든 $B_i \in Q$ 는 분배적으로 tag와 ε의 서명문 $S' = [x(k-E')] \bmod q$ 를 생성한다. 그리고 사용자로 S' 을 보낸다.

④ 사용자는 S' 을 사용하여 S 를 계산한다. 여기서, $S = S'a \bmod q$ 이다.

사용자는 3쌍 $(A,\ B,\ sign(m')=(R, S))$을 갖는 것으로 발행 프로토콜을 완수한다.

5.5 분배된 Schnorr 매직 잉크 서명 방식을 사용한 발행 프로토콜

초기값으로 B_i는 t 차 다항식 $F(\cdot)$을 이용하여 공유된 비밀값 x 의 부분 비밀 공유 x_i를 갖는다.

① 은행은 적절한 돈을 사용자의 계정으로부터 빼고, 은행 서버의 집합 $B_i \in Q$는 랜덤 비밀 세션키 $k \in_R Z_q$를 생성한다. 이때 각 서버 B_i 는 공유 k_i를 갖는다. 그리고 은행 서버의 집합 $B_i \in Q$는 $R' = [g^k]_p$ 을 계산하고 사용자로 R' 을 보낸다.

② 사용자는 랜덤수 $s \in_R Z_q^*$을 선택하고, m 을 s 를 사용하여 m' 으로 바꾼다. 여기서 $m' = m^s = g_1^{u_1 s} g_2^{u_2 s} d^s$ 이다. 사용자는 m'을 A, B 두 부분으로 나눈다. a 를 랜덤하게 선택하여, $\mu = [m'/a]_q$, $R = [[R'^a]_p]_q, E = h(R, m', A)$, 그리고 $\varepsilon = [E/a]_q$를 계산한다. 이때 사용자는 분배적으로 공개정보가 $(g^{\mu_1}, \cdots, g^{\mu_n})$인 μ의 (t,n)-비밀공유 $(\mu_1, \cdots, \mu_n)$와, 공개 정보가 $(g^{\varepsilon_1}, \cdots, g^{\varepsilon_n})$인 ε의 (t,n)-비밀 공유 $(\varepsilon_1, \cdots, \varepsilon_n)$ 을 계산한다. 이때 사용자는 B_i에게 μ_i, ε_i 을 보낸다.

③ B_i는 분배적으로 tag와 ε의 서명문 $S' = [k + x\varepsilon] \bmod q$ 를 생성한다. 그리고 사용자에게 S' 을 보낸다.

④ 사용자는 S' 을 사용하여 S 를 계산한다. 여기서 $S = S'a \bmod q$ 이다.

사용자는 3쌍 $(A,\ B,\ sign(m')=(E, S))$을 갖는 것으로 발행 프로토콜을 끝낸다.

5.6 KCDSA 매직 잉크 서명방식을 사용한 지불 프로토콜

① 사용자는 3쌍 $(A,\ B,\ sign(m'))$ 을 상점으로 전송한다.

② 상점은 먼저 $A \cdot B \neq 1$을 검증한다. 이 관계가 만족하면 상점은 수신한 서명 (R, S)를 다음과 같이 검증한다. 검증이 맞다면 challenge c 를 사용자에게 보낸다.

$$W' = y^S g^E \bmod q$$
$$h(W') = R$$

③ 사용자는 challenge의 응답으로 $r_1 = x_1 + cx_2 \bmod q$, $r_2 = y_1 + cy_2 \bmod q$, 그리고 $r_3 = z_1 + cz_2 \bmod q$ 를 계산하고, 상점에게 r_1, r_2, r_3를 보낸다.

④ 상점은 $g_1^{r_1} g_2^{r_2} d^{r_3} \stackrel{?}{=\!=} AB^c$ 를 검증하고 맞으면 정당한 전자 현금으로 받아들인다.

5.7 Schnorr 매직 잉크 서명 방식을 사용한 지불 프로토콜

① 사용자는 3쌍 $(A,\ B,\ sign(m'))$을 상점으로 전송한다.

② 상점은 먼저 $A \cdot B \neq 1$을 검증한다. 맞으면 수신한 서명문 (E, S)을 다음과 같이 검증한다.

$$V = g^S y^E \bmod p$$
$$E \stackrel{?}{=\!=} h(V, m')$$

검증이 맞다면 challenge c를 사용자에게 보낸다.

③ 사용자는 challenge의 응답으로 $r_1 = x_1 + cx_2 \bmod q$, $r_2 = y_1 + cy_2 \bmod q$, 그리고 $r_3 = z_1 + cz_2 \bmod q$ 를 계산하고, 상점에게 r_1, r_2, r_3를 보낸다.

④ 상점은 $g_1^{r_1} g_2^{r_2} d^{r_3} \stackrel{?}{=\!=} AB^c$ 를 검증하고 맞으면 코인을 받아들인다.

5.8 이체 프로토콜 (Deposit Protocol)

상점은 일정한 시간이 경과되어 은행으로 사용자로부터 수신된 모든 지불 정보들 $(A, B, sign(m'), c, r_1, r_2, r_3)$ 을 보낸다. 은행은 상점으로부터 수신된 정보의 정당성을 확인하고 해당 금액을 상점의 계좌로 지불한다.

사용자가 현금을 이중으로 사용하면 challenge에 대한 응답, $x_1, x_2, y_1, y_2, z_1, z_2$를 은행은 구할 수 있다. 이를 이용하여 은행은 이중 사용자의 u_1, u_2을 다음 식과 같이 계산할 수 있다.[8]

$$u_1 = (x_1+x_2)(z_1+z_2)^{-1} \bmod q$$
$$u_2 = (y_1+y_2)(z_1+z_2)^{-1} \bmod q$$

은행은 $I = g_1^{u_1} g_2^{u_2}$ 를 계산하여 이중 사용자의 신원을 확인할 수 있다.

제6장 결 론

본 논문은 기존의 전자 현금의 인출 과정에서 많이 사용되던 은닉 서명 기법의 단점을 보완한 매직 잉크 서명 기법을 분석하였고, 새로운 매직 잉크 서명기법을 제안하였다. 그리고 분배 개념과 비밀 공유 분산 기법을 이용한 분산화된 서버를 갖는 매직 잉크 서명 방식을 제시하였다. 또한 본 논문에서는 매직 잉크 서명에 적용 가능한 다양한 툴과 기법들을 분석하고, 이를 이용한 두가지 방법의 효율적인 전자현금 시스템을 제안하였다. 두가지 전자현금 시스템은 단일 서버 전자현금 시스템과 분배된 전자 현금 시스템이 있다. 본 논문의 전자현금 시스템에는 분할 사용이 가능하지 않고 양도 기능이 없기 때문에, 추후에는 분할 사용이 가능하고 양도가 가능한 전자 현금으로 확장할 계획이다.

-참고문헌-

[1] Man Young Rhee, "Cryptography and Secure Communications", Mcgraw-Hill, 1992.

[2] Bruce Schneier, "Applied Cryptography", Wiley, 1996.

[3] Markus Jakosson, Moti Yung, "Distributed Magic Ink Signature", Advances in cryptology - Proceedings of Eurocrypt '97, pp. 450-464.

[4] Markus Jakosson, Moti Yung, "Revokable and Versatile Electronic Money", 3rd ACM conference on Computer and Communications Security, 1996, pp. 76-87.

[5] P. Feldman, "A Practical Scheme for Non-Interactive Verifiable Secret Sharing", FOCS '89, pp. 427-437, 1989.

[6] T. Pedersen, "Distributed provers with applications to undeniable signatures", Advances in Cryptology -Proceedings of Eurocrypto '91, 1991.

[7] T. Pedersen, "Non-interactive and information-theretic secure verifiable secret sharing", Advances in Cryptology-Proceedings of Crypto '91, pp. 129-140.

[8] S. Brands, "An Efficient Off-line Cash Systems Based on the Representation Problem", C.W.I. Technical Report CS-T9323, The Netherlands.

[9] C. Schnorr. "Efficient signature generation by smart cards", Jounal of Cryptology 4,

pp.161-174, 1991.

[10] R. Gennaro, S. Jarecki, H. Krawczyk, T. Rabin,"Robust Threshold DSS Signature", Adances in Cryptology-Proceedings of Eurocrypt '96, pp. 354-371, 1996.

[11] S. Brands, "Untraceable Off-line Cash in Wallet with Observers", Advances in Cryptology-Proceedings of Crypto '93, pp.302-318.

[12] N. Ferguson, "Single-term Off-line Coins", Advances in Cryptology-Proceedings of Eurocrypto '93, 1993.

[13] 이필중, "매직 잉크 서명 방식," private communications, 1997.6.

[14] KISA, "KCDSA(Korea certificate-based Digital Signature Standard)" 1997.6.