검증 가능한 비밀 공유 기법을 이용한 매직 잉크 서명 방식

류 영규, 윤 호선, 류 종호, 염 흥열
순천향대학교 전기전자공학부

A Magic Ink Signature Scheme Using Verifiable Secret Sharing Method

Young-Kyu Lyu, Ho-Sun Yoon, Jong-ho You, Heung-Youl Youm,
Dept. of Electrical and Electronic Eng., Soonchunhyang Univ.
E-mail : young@elec.sch.ac.kr

요약

전자현금은 기본적으로 은행의 서명문이다. 전자 현금 시스템에 분배 개념을 추가함으로써 은행 서명 시스템의 안전성을 증가시킬 수 있고, 익명으로 발행된 전자 현금을 추적 할 수 있다. 매직 잉크 방식은 사용자에 익명으로 서명문을 발급할 수 있고 추후에 사용자가 불법적으로 전자 현금을 사용하는 경우 이를 추적할 수 있는 서명 기법이다. 그러나 매직 잉크 서명 기법에도 믿음을 집중하는 문제가 제기되며, 이 경우 분배된 매직 잉크 서명 기법(Distributed Magic Ink Signature Scheme)이 사용되어야 한다. 본 논문에서는 분배 개념을 적용한 DSS 매직 잉크 서명을 기술하고, 분배 개념을 적용한 KCDSA 매직 잉크 서명을 제시하고 DSS 방식과 KCDSA 매직 잉크 서명 방식의 계산량을 서로 비교 분석하였다.

제1장 서론

전세계적으로 인터넷의 발달로 인한 전자 상거래가 활발히 이루어지고 있다. 따라서 전자 상거래에서의 보안이 매우 중요하게 되었다. 지금까지의 전자 상거래에서는 대부분이 신용카드를 사용하였다. 그러나 신용카드를 사용하는 전자 상거

래는 개인의 프라이버시를 침해하는 문제가 있다. 이를 해결 할 수 있는 한 방식이 전자 현금시스템이다. 지금까지의 전자 현금 시스템에서는 D. Chaum이 제안한 은닉 서명 기법(Blind Signature)[1]이 대표적인 것이었다. 그러나 은닉 서명 방식은 사용자의 프라이버시는 보호하지만 은행의 프라이버시는 보호하지 못하고 사용자에 의한 돈 세탁과 불법 이중 사용을 가능케 하는 단점이 있다. 즉, 은닉 서명 기법은 수신자만이 서명된 문서를 가질 수 있고, 서명자는 서명된 문서가 어떤 것인지 알 수가 없었다. 이러한 단점을 보완한 것이 매직 잉크 서명 기법이다. 이 서명 기법은 서명자가 서명문의 노출(Unblind)을 필요로 하면 매직 잉크를 현상하는 것으로 문서를 볼 수 있는 기법이다. 하지만 이 노출은 범죄 행위나 부정한 행동이 발견될 때만 수행된다. 최근에 독립적인 개체들간의 다양한 분배 개념들이 제시되었다. 전자 현금 시스템에 분배 개념들을 추가함으로서 시스템의 안전성을 증가시키고, 서비스의 유용성을 증가시킨다. 이러한 이유에서 매직 잉크 서명 기법에도 분배된 개념을 사용한 분배된 매직 잉크 서명 기법(Distributed Magic Ink Signature Scheme)[2]이 사용된다.

본 논문의 2장에서는 매직 잉크에 사용되는 비밀 공유 방식을 기술하고, 3장에서는 단일 서버인 경우에 대하여 기술하고, 4장에서는 2장에서 기술된 기술들을 분배된 DSS 매직 잉크 서명 기법과 분배된 KCDSA[3] 매직 잉크 서명 기법에 적용해 본다. 5장에서는 적용된 두 매직 잉크 서명 기법을 비교 분석한다.

제2장 매직 잉크에 사용된 비밀 공유 방식 기술

이 장에서는 비밀 공유를 위해서 적용되는 기술들에 대해서 서술한다. 이러한 기술들은 분배된 DSS 매직 잉크 서명 프로토콜과 분배된 KCDSA 매직 잉크 서명 기법에 적용될 기반 기술들이다. 또한 기타 많은 응용 분야를 가질 수 있을 것이다.

2.1. 기본적인 비밀 공유 방식

처음으로 비밀 공유 방식에 대해서 언급한 Shamir는 유한체의 다항식을 이용했으며, Lagrange는 다항식을 보간하는 방법을 발표했다. 이 절에서는 Shamir의

비밀 공유 기법, Lagrange의 보간 다항식, 그리고 멱승에서의 보간법을 다룬다.

가. Shamir의 비밀 공유 기법

Shamir[4]는 임계치 기법의 구성을 위해 유한체의 다항식을 사용하였다. (t, n) 임계치 비밀 공유 기법은 n 참여자들에게 비밀에 관한 부분 정보를 분배하는 딜러(dealer)가 있는 n 참여자들 사이의 프로토콜이다. 이러한 비밀 공유 기법은 t보다 작은 참여자들의 그룹이 비밀에 관한 어떠한 정보도 얻을 수 없으며, 적어도 t 참여자들의 그룹은 다항 시간(polynomial time)에 비밀을 계산할 수 있다.

나. Lagrange 보간 다항식(Interpolation Polynomial)

n 지점($y_1=f(x_1)$, $y_2=f(x_2),\ldots,y_n=f(x_n)$)을 지나는 n-1 차의 보간 다항식(interpolation polynomial)[5]이 식 (1)과 같이 주어졌다고 하자.

$$P(x)=\sum_{j=1}^{n}P_j(x) \qquad (1)$$

여기서 $P_j(x)=\prod_{\substack{k=1\\k\neq j}}^{n}\frac{x-x_k}{x_j-x_k}y_j$ 이다. 즉, 식 (1)은 식 (2)으로 표현할 수 있다.

$$P(x)=\sum_{j=1}^{n}y_j\prod_{\substack{k=1\\k\neq j}}^{n}\frac{x-x_k}{x_j-x_k} \qquad (2)$$

다. 멱승 보간법

$(a_1,\ldots,a_n)\xleftarrow{(t,n)}a \bmod q$인 a_i가 각 참여자들에게 분배되고, 각 참여자들은 $g^{a_i} \bmod p$로부터 $g^a \bmod p$를 계산하려고 한다고 가정하자. 이러한 계산을 수행하기 위해서 식 (3)을 수행하면 된다.

$$g^a=\prod_{i=1}^{t}(g^{a_i})^{c_i} \qquad \left(\text{여기서, } c_i=\prod_{1\le j\le t,\,j\neq i}\frac{x_j}{x_j-x_i}\right)$$

$$=(g^{a_1})^{c_1}\cdot(g^{a_2})^{c_2}\cdot\cdots\cdot(g^{a_t})^{c_t}$$

$$=(g^{a_1})^{\frac{x_2}{x_2-x_1}\cdot\frac{x_3}{x_3-x_1}\cdots\frac{x_t}{x_t-x_1}}\cdots(g^{a_t})^{\frac{x_1}{x_1-x_t}\cdots\frac{x_{t-1}}{x_{t-1}-x_t}} \qquad (3)$$

이러한 보간 방법을 이후 Exp-Interpolate로[6] 표현한다.

2.2. 검증 가능한 비밀 공유 방식

2.1절에서 다룬 비밀 공유 방식은 모든 참여자들이 정직하다는 가정 하에서 유효한 프로토콜이다. 하지만 실제 응용에서는 악의 있는 적을 고려할 필요가 있다. 이 절에서는 공유된 비밀의 정당성을 검증할 수 있는 프로토콜들과 이러한 프로토콜들을 이용한 곱셈 프로토콜과 딜러가 필요 없는 연합 난수 비밀 공유 방식을 다룬다. 이러한 검증 가능한 비밀 공유 방식을 VSS(Verifiable Secret Sharing)라고 한다.

가. Feldman의 검증 가능한 비밀 공유 방식

이 프로토콜은 딜러를 포함해서 $\frac{n-1}{2}$의 악의 있는 장해까지 견딜 수 있다. Shamir의 방식과 유사하게, 각 행위자 P_i에 대해서 $(\sigma_1, \ldots, \sigma_n) \xleftarrow{(t,n)} \sigma \bmod q$인 공유 σ_i를 생성한다. 만약 $f(x) = \sum_j a_j x^j$이면 딜러는 $\alpha_j = g^{a_j} \bmod p$를 공개한다. 이것은 행위자들이 식 (4)에서와 같이 $g^{\sigma_i} = \prod \alpha_j^{i^j}$을 검사함으로써 값 σ_i의 정당성을 검사하도록 한다. 또한 재구성할 때에 잘못된 공유 σ_i'을 검출하도록 한다. 비밀의 값이 오직 계산적으로 안전하다는 것에 주의하라. 즉, 비밀 값 $g^{a_0} = g^{\sigma} \bmod p$이 누설된다.

$$
\begin{aligned}
g^{\sigma_i} &= \alpha_0 \cdot \alpha_1^{i^1} \cdot \alpha_2^{i^2} \cdot \alpha_3^{i^3} \cdots \alpha_t^{i^t} \\
&= g^{a_0} \cdot g^{a_1 i} \cdot g^{a_2 i^2} \cdots g^{a_t i^t} \\
&= g^{a_0} \cdot g^{a_1 i} \cdot g^{a_2 i^2} \cdots g^{a_t i^t} \\
&= g^{\sigma_i} (\because \sigma_i = a_0 + a_1 i + a_2 i^2 \cdots + a_t i^t) \qquad (4)
\end{aligned}
$$

이러한 비밀 공유 방식을 이후 Feldman-VSS[7]라고 표현한다.

나. Pedersen의 검증 가능한 비밀 공유 방식

Feldman의 VSS의 안전성은 이산 대수의 어려움에 근거하며, Pedersen의 VSS는 정보 이론적으로 안전한(Information-Theoretic Secure) VSS 이다. 자세한 프로토콜은 다음과 같다.

먼저, g, h $\in$ G_q를 정의하자. 여기서 G_q는 q차의 Z_q^*의 유일한 부분집합이고, g는 G_q의 생성자이다. 만약 요소 $a \in Z_q^*$가 G_q에 있다면 $a \in G_q \leftrightarrow a^g = 1$이다. Z_q는 필드이고, 딜러는 s$\in Z_q$를 다음과 같이 분산한다.

① 딜러는 s 위탁을 공개한다 : 임의적으로 선택된 t$\in Z_q$에 대해서 $E_0 = E(s,t)$ 이다. 여기서 $E(s,t) = g^s h^t$ 이다.

② 딜러는 F(0)=s를 만족하는 최소한 k-1 차의 다항식 $F \in Z_q[x]$를 선택하며, 그리고 나서 $s_i = F(i)$ (for $i = 1, \ldots, n$)를 계산한다. 여기서, $F(x) = s + F_1 x + \cdots + F_{k-1} x^{k-1}$라 하자. 또한 딜러는 임의적으로 $G_1, \cdots, G_{k-1} \in Z_q$를 선택하고 i=1, ⋯ , k-1에 대해서 F_i에 위탁을 수행할 때 G_i를 사용한다. 즉, 딜러는 $E_i = E(F_i, G_i)$를 공개한다.

③ $G(x) = t + G_1 x + \cdots + G_{k-1} x^{k-1}$이라 하고 $t_i = G(i)$ $(i = 1, \ldots, n)$라 하자. 그리고 나서 딜러는 비밀스럽게 (s_i, t_i)를 P_i에게 전송한다. 여기서 i=1, 2, ... , n 이다.

P_i가 그의 공유 (s_i, t_i)를 수신하면, P_i는 다음 식을 검증한다.

$$E(s_i, t_i) = \prod_{j=0}^{k-1} E_j^{i^j} \qquad (5)$$

위의 식 (5)이 만족하면 P_i는 정당한 비밀의 몫을 공유한 것이고, 그렇지 않으면 잘못된 비밀의 몫을 수신한 것이다. 이것의 증명은 식 (6)과 같다.

$$E(s_i, t_i) = g^{s_i} h^{t_i}$$
$$= g^{s + F_1 i + F_2 i^2 + \cdots + F_{k-1} i^{k-1}} h^{t + G_1 i + G_2 i^2 + \cdots + G_{k-1} i^{k-1}}$$

$$\prod_{j=0}^{k-1} E_j^{i^j} = E_0^{i^0} E_1^{i^1} \cdots E_{k-1}^{i^{k-1}}$$

$$= E(s,t) \cdot [E(F_1, G_1)]^i \cdots [E(F_{k-1}, G_{k-1})]^{i^{k-1}}$$

$$= g^s h^t \cdot [g^{F_1} h^{G_1}]^i \cdots [g^{F_{k-1}} h^{G_{k-1}}]^{i^{k-1}}$$

$$= g^{s+F_1 i+F_2 i^2+\cdots+F_{k-1} i^{k-1}} h^{t+G_1 i+G_2 i^2+\cdots+G_{k-1} i^{k-1}} \quad (6)$$

이러한 비밀 공유 방식을 이후 Pedersen-VSS라고 표현한다. 이러한 비밀 공유 방식은 무조건적으로 비밀의 기밀성을 보호하지만, 공유의 정당성은 계산적인 가정에 의존한다.

다. 연합 난수 비밀 공유 방식

지금까지 논의된 비밀 공유 방식은 신임 받는 딜러가 존재해야만 한다. 하지만 이번 절에서 논의하는 비밀 공유 방식은 참여자 각자가 딜러처럼 행동함으로써 딜러가 필요 없으며, 각 참여자는 임의의 난수를 공유하는 프로토콜이다. 즉, 자신의 비밀 공유 α_i를 선택하여, (t, n)-비밀 공유 기법을 이용하여 각 참여자에게 부분 비밀 α_{ij}을 분배한다. 또 각 참여자들은 수신된 부분 정보를 합하여 비밀 s에 대응되는 자신의 부분 비밀 공유 s_i를 계산한다. 이렇게 함으로서 각 참여자들은 $(s_1, \ldots, s_n) \xleftrightarrow{(t,n)} s \bmod q$ 가 되는 비밀 공유 s_i를 갖는다. 일반적인 (t, n)-비밀 공유 기법에서처럼 비밀 s값은 모든 참여자들로부터 그리고 심지어는 t-1참여자들의 연합으로부터도 비밀이 유지된다.

최종 공유 s_i (i=1,...,n)은 각 참여자에 의해 P_i에게 분배된 공유의 합으로 계산된다. 따라서, 공동 비밀 s는 모든 분배된 비밀들의 합과 같다. 연합 난수 비밀 공유 기법[8][9]의 예가 다음에 있다.

① P_i는 α_i의 분배를 위한 (t, n)-비밀 공유 기법을 가지고 있다고 가정

② 각 P_i는 $\alpha_i \in Z_q$를 임의로 선택

③ 각 P_i는 (t, n)-비밀 공유 기법을 이용해 α_i에 대한 부분 공유인 α_{ij}를 계산

④ 각 참여자인 P_j에게 α_{ij}를 전송

⑤ $P_1 : (\alpha_{11}, \alpha_{12}, \cdots, \alpha_{1n})$, $P_2 : (\alpha_{21}, \alpha_{22}, \cdots, \alpha_{2n}), \cdots, P_n : (\alpha_{n1}, \alpha_{n2}, \cdots, \alpha_{nn})$

$$\alpha_{11} = a_1x_1^2 + b_1x_1 + \alpha_1,\ \alpha_{12} = a_1x_2^2 + b_1x_2 + \alpha_1,\ \cdots,\ \alpha_{1n} = a_1x_n^2 + b_1x_n + \alpha_1$$

$$\alpha_{21} = a_2x_1^2 + b_2x_1 + \alpha_2,\ \alpha_{22} = a_2x_2^2 + b_2x_2 + \alpha_2,\ \cdots,\ \alpha_{2n} = a_2x_n^2 + b_2x_n + \alpha_2$$

$$\vdots \qquad\qquad \vdots \qquad\qquad \vdots$$

$$\alpha_{n1} = a_nx_1^2 + b_nx_1 + \alpha_n,\ \alpha_{n2} = a_nx_2^2 + b_nx_2 + \alpha_n,\ \cdots,\ \alpha_{nn} = a_nx_n^2 + b_nx_n + \alpha_n$$

⑥ 각 P_i는 $\alpha_{1i}, \cdots, \alpha_{ni}$를 수신하고, 부분 비밀 공유들 $s_i = \alpha_{1i} + \cdots + \alpha_{ni}$을 계산한다. 예를 들어서, P_1은 $\alpha_{11}, \cdots, \alpha_{n1}$을 계산할 수 있고, $s_1 = \alpha_{11} + \cdots + \alpha_{n1}$을 계산할 수 있다. 각 참여자들이 계산하는 부분 비밀 공유는 다음과 같다.

$$\begin{aligned} s_1 &= a_1x_1^2 + b_1x_1 + \alpha_1 + \cdots + a_nx_1^2 + b_nx_1 + \alpha_n \\ &= (a_1 + \cdots + a_n)x_1^2 + (b_1 + \cdots + b_n)x_1 + (\alpha_1 + \cdots + \alpha_n) \end{aligned}$$

$$\vdots$$

$$\begin{aligned} s_n &= a_1x_n^2 + b_1x_n + \alpha_1 + \cdots + a_nx_n^2 + b_nx_n + \alpha_n \\ &= (a_1 + \cdots + a_n)x_n^2 + (b_1 + \cdots + b_n)x_n + (\alpha_1 + \cdots + \alpha_n) \end{aligned}$$

⑦ 비밀 $s = \sum_{i=1}^{n} a_i$의 (t, n)-임계치 기법의 부분 비밀 공유: $(s_1, \cdots, s_n)$

따라서 t명 이상의 참여자 등이 연합하면 비밀 정보 s를 구할 수 있음을 알 수 있다.

이러한 비밀 공유 방식을 이후 Joint-RSS라고 표현한다.

이러한 연합 난수 비밀 공유 방식을 검증하기 위해서 Feldman의 비밀 공유 방법을 적용할 수 있다. 예를 들어서, α_{1n}의 값을 검증하기 위해서 P_1은 g^{a_1}, g^{b_1}, g^{α_1}을 공개한다. 이렇게 공개된 값들을 이용해서 식 (4)에 적용하면 α_{1n}을 수신하는 참여자 P_n은 수신된 α_{1n}이 정당한 값인지를 검증할 수 있다. 식 (7)을 참고하라.

$$g^{\alpha_{1n}} = g^{\alpha_1} \cdot g^{b_1 x_n} \cdot g^{a_1 x_n^2}$$

$$= g^{a_1} \cdot g^{b_1 x_n} \cdot g^{a_1 x_n^2}$$

$$= g^{a_{1n}} (\because \alpha_{1n} = \alpha_1 + a_1 x_n + b_1 x_n^2) \quad (7)$$

이러한 방법을 이후 Feldman-JR-VSS이라고 표현한다.

연합 영 비밀 공유(Joint Zero Secret Sharing) 프로토콜은 비밀 값이 영인 총괄적인 공유를 생성한다. 이 프로토콜은 연합 난수 비밀 공유 프로토콜과 비슷하지만 지역적인 비밀 대신 각 참여자들이 영 값의 공유를 분배한다. 영 값의 정당한 분배는 각 분배 다항식들의 상수항이 '0'이라는 것을 검사하는 것에 의해 수행된다. 즉, 2.2절에서, 각 참여자 P_i는 비밀 공유 α_i의 값이 '0'이 되도록 선택한다. 만약 모든 참여자들이 이러한 프로토콜을 따른다면, 비밀 s의 값인 다항식의 상수항 $\alpha_1 + \cdots + \alpha_n$의 값은 '0'이 된다. 비밀 s의 현행 공유에 영 공유를 추가하는 것으로, 비밀을 바꾸지 않고, 비밀 s의 공유의 랜덤화를 얻을 수 있다. 앞에서 서술되었듯이 검증 가능한 비밀 공유 방식을 적용하면 검증 가능한 연합 영 비밀 공유 방식을 생성할 수 있다.

다. 두 비밀들의 곱셈

이 절에서 다룰 프로토콜은, 참여자들 사이에 공유된 주어진 두 비밀 u와 v가 있을 때, 이 두 비밀을 밝히지 않고 두 비밀의 곱 uv를 계산하는 프로토콜이다.

각각 t 차수의 다항식에 위해 공유된 주어진 u, v는 각 참여자들이 자신들의 u, v공유를 지역적으로 곱한다. 결과는 2t 차수 다항식의 uv 공유가 될 것이다. 따라서, 값 uv는 2t+1의 정당한 공유들의 집합으로부터 재구성된다.

t=3인 경우, u와 v는 3차 다항식이 되고, 이 두 비밀의 곱은 6차 다항식이 된다. 즉, 7개의 정당한 공유들의 집합으로 비밀이 재구성된다. 연합 영 비밀 공유 프로토콜을 사용하는 추가적인 재 랜덤화 절차는 곱해진 비밀들의 비밀을 보호하는데 필요하다. 이후 Multi-Zero라 표기한다.

제3장 단일 서버 매직 잉크 서명 프로토콜

단일 서버 매직 잉크 서명 프로토콜은 유일한 하나의 서버가 서명고 서명과 관련된 비밀정보를 한 서버에만 보관함으로써 믿음이 집중되며, 서버가 원하면 언제든지 서명문을 추적할 수 있는 특징이 있다.

이 장에서는 단일 서버인 DSS 매직 잉크 서명 프로토콜과 단일 서버인 KCDSA 매직 잉크 서명 프로토콜을 기술한다. 단일 서버 매직 잉크 서명 프로토콜을 수행하는 동안에 tag를 계산 해 서버가 가지고 있어 추후에 서명 쌍들이 사용될 경우 tag와 비교하여 서명 쌍을 추적한다.

3.1. 단일 서버 DSS 매직 잉크 서명 프로토콜

가. 파라메터

DSS 서명 방식의 시스템 파라메터는 다음과 같다.

a. p: 소수 $|p| = 512 + 64i, i=0, \cdots, 8$

b. q: 160 비트 소수, $q \mid p-1$.

c. g: $g = h^{(p-1)/q} \bmod p$ ($g>1$, $1<h<p-1$)

d. H: 160 비트 해쉬함수(SHA)

e. x: 비밀키 ($0 < x < q$)

f. y: 공개키 ($y = [g^x]_p$)

나. 서명 프로토콜

단일 서버 방식의 서명 프로토콜은 다음과 같다.

R		S
m = H(M), $a, b \in_R Z_q$		
$\mu = [ma]_q$ 계산	$\xrightarrow{\mu}$	$\bar{k} \in_R Z_q$ 선택
		$\bar{r} = [g^{\bar{k}^{-1}}]_p$ 계산
$r = [[\bar{r}^b]_p]_q$ 계산	$\xleftarrow{\bar{r}}$	

$\rho = [ra]_q$ 계산 $\xrightarrow{\rho}$ $tag[(\mu\rho^{-1})]_q$ 계산

$\sigma = [\overline{k}(\mu + x\rho)]_q$ 계산

$s = [\sigma a^{-1} b^{-1}]_q$ $\xleftarrow{\sigma}$

여기서 m,r,s가 메시지 M의 서명문 쌍이 된다.

다. 검증

위에서 구한 (m,r,s)을 수신한 송신자는 다음의 관계식을 이용하여 서명문을 검증한다.

$$r = g^{ms^{-1}} y^{rs^{-1}}$$
$$= g^{(m+xr)s^{-1}}$$
$$= g^{k^{-1}b} = \overline{r}^{b}$$

3.2. 단일 서버 KCDSA 매직 잉크 서명 프로토콜

가. 파라메터

KCDSA의 시스템 파라메터는 다음과 같다.

a. p: 소수 $|p| = 512 + 256i, i = 0, \cdots, 6$

b. q: 소수 $q \mid p-1, |q| = 128 + 32j, j = 0, \cdots, 4$

c. g: GF(p) 상에서 위수가 q인 수

d. H: $|q|$ 길이의 출력값을 갖는 충돌회피성 해쉬함수

e. x: 비밀키 $(0 < x < q)$

f. y: 공개키 $(y = [g^{x^{-1}}]_p)$

h. Z: Cert_Form의 해쉬값

나. 서명 프로토콜

KCDSA에 기초인 단일 서버 서명 프로토콜은 다음과 같다.

R		S_i
$a \in_R Z_q$		
$H = h(Z, M),\ H' = [H/a]_q$	$\xrightarrow{H'}$	
		$\overline{k} \in_R Z_q$ 선택
	$\xleftarrow{W}$	$W = [g^k]_p$
$R = h(W^a), R' = [R/a]_q$		
	$\xrightarrow{R'}$	$tag[(H'R'^{-1})]_q$ 계산
		$E' = H' \oplus R'$
$S = [S'a]_q$	$\xleftarrow{S'}$	$S' = [x(k - E')]_q$

다. 검증

서명문의 검증은 다음의 관계식을 이용한다.

$$(W^a)' = g^{H \oplus R} \cdot y^S$$

$$g^{ka} = g^{H \oplus R} y^S$$

$$= g^{H \oplus R} g^{x^{-1}S}$$

$$R \overset{?}{=} h((W^a)')$$

제4장 분배된 매직 잉크 서명 프로토콜

분배된 개념의 매직 잉크 서명 프로토콜은 시스템의 안정화와 유효성을 증가 시키고, 믿음의 집중을 분산 시켜 일정의 정족수가 모여야 서명과 추적 할 수 있다. 이 장에서는 분배된 DSS 매직 잉크 서명 프로토콜과 분배된 KCDSA 매직 잉크 서명 프로토콜에 공유 기법들을 적용하고, 이를 비교 분석한다. 본 논문에서는 Feldman의 기법만을 이용하여 적용하였다. Feldman 기법 대신에 Pedersen의 기

법도 적용 가능하다.

4.1. tag 계산

분배된 DSS 매직 잉크 서명 프로토콜에서 tag 계산은 다음과 같다.

$S_1, \cdots, S_n$ 사이에 공유된 비밀 $\rho \bmod q$ 가 R에 의해 주어지면, ρ와 ρ^{-1} 상에 정보를 밝히지 않고 $\rho^{-1} \bmod q$ 공유를 생성한다.

각 서버 S_i는 ρ의 (t, n) 비밀 공유에 대응하는 공유 ρ_i를 갖는다. 즉, $(\rho_1, \cdots, \rho_n) \xrightarrow{(t,n)} \rho$이다. ρ^{-1}에 대한 공유들의 계산은 아래와 같이 수행된다.

a. Feldman-JR-VSS(혹은 Pedersen-JR-VSS)를 프로토콜을 이용해서 랜덤 요소 $d \in Z_q$의 (t, n) 공유를 생성한다. 즉, $(d_1, \cdots, d_n) \xrightarrow{(t,n)} d$이다.

b. 각 서버들은 (2t-1)차 항 $[\rho_i d_i]_q$을 Multi 수행한 후에, 각 서버 S_i는 "secret" 0의 공유 b_i를 갖는다.

c. 각 서버들은 $\rho_i d_i + b_i$를 공개하고 대응하는 (2t-1)차 다항식을 Interpolate $(\rho_1 d_1 + b_1, \cdots, \rho_n d_n + b_n) \xrightarrow{(2t,n)} \alpha = \rho d$ 을 계산한다.

d. 각 서버들은 $\alpha_i \triangleq \alpha^{-1} d_i \bmod q$를 설정함으로써 ρ^{-1}의 공유 α_i을 계산한다.

e. 각 서버들은 자신의 $[(\mu_i \alpha_i)]_q$를 Multi로 수행하여 tag를 각각 가지고 있는다.

그러므로 서 추후에 서명 쌍들이 사용될 경우 $[(\mu_i \alpha_i)]_q \xrightarrow{(2t,n)} \mu\alpha$ 인 tag를 계산하며, 서명 쌍으로 온 것과 비교하므로 서 서명 쌍을 추적한다.

분배된 KCDSA 매직 잉크 서명 프로토콜의 tag 계산은 위 과정에서 $\rho \rightarrow R'$로 $\mu_i \rightarrow H'_i$ 로 치환하여 계산하면 된다.

4.2. 분배된 DSS 매직 잉크 서명 프로토콜

가. 키 생성

분배된 DSS 매직 잉크 서명 프로토콜을 수행하기 전 각 서버들은 Feldman-JR-VSS를 수행하여 서버 S_i가 비밀값 x에 대한 그의 몫인 비밀 입력 x_i을 유지한다. 각 서버들은 수신된 비밀 정보에 대해 g^{x_i} 하여 공개한다. 각 서버 S_i는 Exp -Interpolate를 이용하여 공개키 y= g^x을 계산하여 공개한다.

나. 서명 프로토콜

① Feldman-VSS을 이용하여 R은 서버 S_i에 대한 μ에 대한 그의 몫 μ_i를 S_i에게 전송한다.

② Feldman-JR-VSS을 수행하여 서버 S_i가 랜덤 값 $\overline{k}$에 대한 그의 몫인 비밀 입력 $\overline{k_i}$ 을 유지하고, $g^{\overline{k_i}}$을 공개한다.

③ Feldman-JR-VSS을 이용하여 d을 공유하고 각 서버는 g^{d_i}을 공개한다. (2t-1)차 다항식에 의존하는 $[e_i = k_i d_i + c_i]_q$의 Multi-Zero를 수행하여 e_i을 구하고, Interpolate $(e_i, \cdots, e_n) \xrightarrow{(2t,n)} \epsilon$을 계산하고, 각 서버 S_i는 지역적으로 $\overline{r} = [g^{\overline{k}^{-1}}]_q$을 계산한다.

- 각 서버는 d을 Exp-Interpolate → g^a
- Interpolate $(e_i, \cdots, e_n) \xrightarrow{(2t,n)} \epsilon$
- e^{-1} → $k^{-1}d^{-1}$
- $(g^d)^{e^{-1}}$ mod p mod q → $g^{k^{-1}}$ ($\overline{r}$)을 계산하여 각 서버들은 R에게 전송한다.

④ Feldman-VSS을 이용하여 R은 서버 S_i 에 대한 ρ에 대한 그의 몫 ρ_i을 S_i에게 전송한다.

⑤ 서버들 간에 $[x_i \rho_i + c_i]_q$을 Multi-Zero 프로토콜을 수행한 후 Interpolate $(x_1 \rho_1, \cdots, x_n \rho_n) \xrightarrow{(2t,n)} x\rho$ 값을 계산 한 후, 각 서버 S_i는 (2t-1)차 다항

식에 의존하는 $[\overline{k_i}(\mu_i + x\rho) + c_i]_q \rightarrow \sigma_i$을 Multi-Zero로 수행하여 각 S_i는 σ_i를 계산하고, Interpolate $(\sigma_1, \cdots, \sigma_n) \xrightarrow{(2t,n)} \sigma$ 하여 서버는 σ을 R에게 전송한다.

R		S_i
m = h(M), $a, b \in_R Z_q$		
$\mu = [ma]_q$		
μ의 (t,n) 비밀공유 ①을 수행		
	$\xrightarrow{\mu_i}$	$\overline{k} \in_R Z_q$ ②를 수행
		$[g^{\overline{k_i}}]_p$ 공표
$r = [[\overline{r}^b]_p]_q$	$\xleftarrow{\overline{r}}$	$\overline{r} = [g^{\overline{k}^{-1}}]_p$ ③를 수행
$\rho = [ra]_q$		
ρ의 (t,n) 비밀 공유 $(\rho_1 \cdots \rho_n)$ ④을 수행	$\xrightarrow{\rho_i}$	$tag[(\mu_i a_i)]_q$ 계산
		$\sigma_i = [\overline{k_i}(\mu_i + x_i\rho_i)]_q$
$s = [\sigma a^{-1} b^{-1}]_q$	$\xleftarrow{\sigma}$	$\sigma = [\overline{k}(\mu + x\rho)]_q$ ⑤를 수행

4.3. 분배된 KCDSA 매직 잉크 서명 프로토콜

가. 키 생성

분배된 KCDSA 매직 잉크 서명 기법[10]을 수행하기 전 각 서버들은 Feldman-JR-VSS을 수행하여 서버 S_i가 비밀값 x에 대한 그의 몫인 비밀 입력

x_i 를 유지한다.

서명 프로토콜 전에 Feldman-JR-VSS을 이용하여 d을 공유하고, 각 서버는 g^{d_i}을 공개한다. (2t-1)차 다항식에 의존하는 $[e_i = x_i d_i + c_i]_q$의 Multi-Zero를 수행하여 e_i을 구하고, Interpolate $(e_i, \cdots, e_n) \xrightarrow{(2t,n)} e$을 계산하고, 각 서버 S_i는 지역적으로 $y=[g^{x^{-1}}]_q$을 계산한다.

- 각 서버는 d을 Exp-Interpolate → g^a
- Interpolate $(e_1, \cdots, e_n) \xrightarrow{(2t,n)} e$
- $e^{-1} \rightarrow x^{-1}d^{-1}$
- $(g^d)^{e^{-1}} \bmod p \bmod q \rightarrow g^{x^{-1}}$(=y)을 계산하여 공개한다.

나. 서명문 생성

① R은 Feldman-VSS을 이용하여 H'의 공유된 값 $(H'_1, \cdots, H'_n)$을 서버 S_i에 대하여 그의 몫 H'_i을 S_i에게 전송한다.

② Feldman-JR-VSS을 수행하여 서버 S_i가 랜덤 값 k에 대한 그의 몫인 비밀 입력 k_i을 유지하고, g^{k_i}을 공개한다.

③ 각 서버 S_i은 Exp-Interpolate를 이용하여 $W=[g^k]_p$을 계산하여 R에게 전송한다.

④ R은 Feldman-VSS을 이용하여 R'의 공유된 값 $(R'_1, \cdots, R'_n)$을 서버 S_i에 대하여 그의 몫 R'_i을 S_i에게 전송한다.

⑤ (2t-1)차 항의 $[x_i(k_i - E'_i) + c_i]_q$을 Multi-Zero로 수행하므로 서 각 서버 S_i는 S'_i을 계산하고, Interpolate $(S'_1, \cdots, S'_n) \xrightarrow{(2t,n)} S'$ 하여 S'을 R에게 전송한다.

R	S_i
$a \in_R Z_q$	

$H = h(Z, M)$, $H' = [H/a]_q$

Secret Sharing

$(H'_1, \cdots, H'_n)$ ①를 수행 $\xrightarrow{H'_i}$ $k \in_R Z_q$ 선택

$\xleftarrow{r}$ $W = [g^k]_p$ ②,③를 수행

$R = h(W^a), R' = [R/a]_q$

Secret Sharing

$(R'_1, \cdots, R'_n)$ ④을 수행 $\xrightarrow{R'_i}$ $tag[(H'_i a_i^{-1})]_q$계산

$E' = H' \oplus R'$

$S'_i = [x_i(k_i - E'_i) + c_i]_q$

$s = [S'a]_q$ $\xleftarrow{S'}$ $S' = [x(k - E')]_q$ ⑤를 수행

제5장 비교 및 분석

DSS와 KCDSA의 매직 잉크 서명 방식은 공통점이 상당히 많다. 그러므로 비밀 공유 방식에 있어서도 서로 유사한 점이 상당히 많지만, KCDSA가 계산적인 면에서 약간 효율임을 알 수 있고, KCDSA의 비밀 공유 방식에서는 $g^{x^{-1}} \bmod p$의 계산을 키 생성 단계에서만 수행하고, 서명문 생성 단계에서는 수행을 하지 않기 때문에 이러한 계산을 위한 많은 계산적인 부하를 줄일 수 있다. 반대로 DSS의 비밀 공유 방식에서는 $g^{k^{-1}} \bmod p \bmod q$의 계산을 서명문 생성 단계에서 수행하고, 대신 공개키 생성 단계에서는 수행하지 않는다. 또한 KCDSA의 비밀 공유 방식에서 W을 계산할 때, Feldman의 검증 가능한 비밀 공유 방식을 적용함으로써 k의 값을 밝히지 않고 $W = g^k \bmod p$를 계산하도록 했다. 표 1은 각 서

명 방식에서 필요로 하는 비밀 공유 기법들을 비교한 것이다.

표 1 DSS 비밀 공유 방식과 KCDSA 비밀 공유 방식의 비교

과정	공유방식 / 서명방식	Feldman-VSS	Feldman-JR-VSS	Multi	Interpolate	Exp-Interpolate
키 생성	DSS	0	1	0	0	1
	KCDSA	0	1	1	1(2t-1차)	1
서명문 생성	DSS	2	2	3	3(2t-1차)	1
	KCDSA	2	1	1	1(2t-1차)	1

표1에서 보듯이, 분배된 KCDSA 매직 잉크 서명 방식이 키 생성 단계에서는 비효율적이지만, 서명문 생성 단계에서는 서명문 생성 단계에서는 상당히 효율적인 것을 알 수 있다.

제6장 결론

본 논문에서는 단일 서버 DSS와 KCDSA 매직 잉크 서명에 다중 서버를 갖는 분배 개념들을 추가함으로서 시스템의 안전성을 증가시키고, 서비스의 유용성을 증가시킨 분배에 대한 기술을 분석했으며 이를 분배된 DSS와 KCDSA 매직 잉크 서명에 적용하였다. 두 방식을 서로 비교 한 결과 키 생성 과정에서는 분배된 KCDSA 매직 잉크 서명이 연산이 더욱 복잡하지만 실제 서명 프로토콜 동안은 분배된 KCDSA 매직 잉크 서명이 상당히 효율적인 것을 알 수 있었다. 추후 연구 분야는 보다 빠르고 간단한 검증 가능한 공유 방식을 적용해야 할 것이다.

참고문헌

[1] D.Chaum, "Blind Signatures for Untraceable Payments", Advances in cryptology-Proceedings of Crypto '82, pp.199-203. 1983.

[2] M. Jakosson, Moti Yung, "Distributed Magic Ink Signature", Advances in Cryptology-Proceedings of Eurocrypt '97, pp. 450-464. 1997.

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

[4] A. Shamir, "How to Share a Secret", Communications of the ACM, 1979

[5] 백종현, "Electronic Cash Protocols Design Using the Magic Ink Signature Sharing", 순천향대학교 석사 논문, 1997

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

[7] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing", In Proc of the 28th IEEE Symposium on the Foundations of Computer Science, pp 427-437, 1987

[8] T.Pedersen, "Distributed provers with applications to undeniable signatures", In Proc. EUROCRYPTO 91, 1991

[9] T.Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing", In Proc. CRYPTO 91, pp 129-140, 1991

[10] 백종현, 염흥열 "Electronic Cash Protocol using the Magic Ink Signature", CISC'97, pp 354-367, 1997