

# 분배된 비밀 공유 기법을 이용한 KCDSA 매직 잉크 서명 방식

류 영규\*, 윤 호선\*, 염 흥열\*

## A KCDSA Magic Ink Signature Scheme Using a Distributed Verifiable Secret Sharing Method

Young-Kyu Lyu\*, Ho-Sun Yoon\*, Heung-Youl Youm\*

### 요 약

전자현금은 기본적으로 은행의 서명문이다. 전자 현금 시스템에 분배 개념을 추가함으로써 전자현금의 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 추적 할 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 제안하고, DSS 방식과 KCDSA 매직 잉크 서명 방식의 계산량을 서로 비교 분석하였다.

### Abstract

Electronic cash is a digital signature issued by bank. If the concept of the distributed secret sharing and magic ink signature is introduced in the existing electronic cash system, we can increase the security level and the availability of electronic cash system, and trace the electronic cash itself and the owner of electronic cash which was issued anonymously to a user in case of illegal usage of electronic cash by users. If the trust is concentrated on one bank system, the problem of misuse of bank can be occurred. To solve this problem, the distributed secrete sharing scheme need to be introduced in electronic cash system.

In this paper, We propose a DSS(Digital Signature Standard) distributed magic ink signature scheme and a KCDSA(Korea Certificate-based Digital Signature Algorithm) distributed magic ink signature scheme using a verifiable secret sharing method, and we compare two methods with respect to the required computation amount for the generation of magic-ink signature.

## I. 서론

전세계적 인터넷 망의 확산으로 인하여 전자 상거래시스템의 구축을 가속화하고 있다. 따라서 전자 상거래에서의 보안이 매우 중요하게 되었다. 지금까지의 전자 상거래에서는 대부분이 신용카드를 사용하였다. 그러나 신용카드를 사용하는 전자 상거래는 개인의 프라이버시를 침해하는 문제가 있다. 이를 해결할 수 있는 한 방식이 전자 현금시스템이다. 지금까지의 대표적인 전자 현금시스템에서는 D. Chaum이 제안한 은닉 서명 기법(Blind Signature)[1]이다. 그러나 은닉 서명 방식은 사용자의 프라이버시는 보호하지만, 사용자가 불법으로 현금을 사용하는 경우 은행의 현금 추적 권한은 보호하지 못하며, 사용자에게 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 자신이 서명한 서명문을 알고 싶으면, 매직 잉크를 현상하는 방법처럼 자신의 서명문을 확인할 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 추가함으로써 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 II장에서는 매직 잉크에 사용되는 비밀 공유 방식을 기술하고, III장에서는 단일 서버인 경우에 대하여 기술하고, IV장에서는 II장에서 기술된 기술들을 분배된 DSS 매직 잉크 서명 기법과 분배된 KCDSA[3] 매직 잉크 서명 기법에 적용해 본다. V장에서는 적용된 두 매직 잉크 서명 기법을 비교 분석한다.

## II. 매직 잉크에 사용된 비밀 공유 방식 기술

이 장에서는 비밀 공유를 위해서 적용되는 기본 암호 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.

### 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 역승에서의 보간법을 다룬다.

#### 1.1. 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. 이러

한 비밀 공유 기법은  $t$ 보다 작은 참여자들의 그룹이 비밀에 관한 어떠한 정보도 얻을 수 없으며, 적어도  $t$  참여자들의 그룹은 다항 시간(Polynomial Time)에 비밀을 계산할 수 있다.

1.2. Lagrange 보간 다항식

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

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

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

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

1.3. 역승 보간법

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

$$\begin{aligned} g^a &= \prod_{i=1}^n (g^{a_i})^{c_i} \quad (c_i = \prod_{1 \leq j \leq t, j \neq i} \frac{x_j}{x_j - x_i}) \\ &= (g^{a_1})^{c_1} \cdot (g^{a_2})^{c_2} \cdot \dots \cdot (g^{a_t})^{c_t} \\ &= (g^{a_1})^{\frac{x_2}{x_2-x_1} \cdot \frac{x_3}{x_3-x_1} \dots \frac{x_t}{x_t-x_1}} \dots \\ &\quad (g^{a_t})^{\frac{x_1}{x_1-x_t} \dots \frac{x_{t-1}}{x_{t-1}-x_t}} \end{aligned} \tag{3}$$

이러한 보간 방법을 이후 Exp-Interpolate 로[6] 표현한다.

2. 검증 가능한 비밀 공유 방식

1절에서 다룬 비밀 공유 방식은 모든 참여자들이 정직하다는 가정 하에서 유효한 프로

토콜이다. 하지만 실제 응용에서는 악의 있는 적을 고려할 필요가 있다. 이 절에서는 공유된 비밀의 정당성을 검증할 수 있는 프로토콜들과 이러한 프로토콜들을 이용한 곱셈 프로토콜과 덜러가 필요 없는 연합 난수 비밀 공유 방식을 다룬다. 이러한 검증 가능한 비밀 공유 방식을 VSS(Verifiable Secret Sharing)라고 한다.

2.1. Feldman 검증 가능한 비밀 공유 방식

이 프로토콜은 덜러를 포함해서  $(n-1)/2$ 의 악의 있는 장해를 견딜 수 있다. Shamir의 방식과 유사하게, 각 행위자  $P_i$ 에 대해서  $(\sigma_1, \dots, \sigma_n) \leftarrow (t, n) \rightarrow \sigma \pmod{q}$ 인 공유  $\sigma_i$ 를 생성한다. 만약  $f(x) = \sum_j a_j x^j$  이면, 덜러는  $a_j = g^{a_j} \pmod{p}$ 를 공개한다. 이것은 행위자들이 식 (4)에서와 같이  $g^{\sigma_i} = \prod a_j^{i^j}$ 을 검사함으로써 값  $\sigma_i$ 의 정당성을 검사하도록 한다. 또한 재구성할 때에 잘못된 공유  $\sigma_i'$ 을 검출하도록 한다. 비밀의 값이 오직 계산적으로 안전하다는 것에 주의하라. 즉, 비밀 값  $g^{a_0} = g^{\sigma} \pmod{p}$ 이 누설된다.

$$\begin{aligned} g^{\sigma_i} &= a_0 \cdot a_1^{i^1} \cdot a_2^{i^2} \cdot a_3^{i^3} \dots a_t^{i^t} \\ &= g^{a_0} \cdot g^{a_1 i} \cdot g^{a_2 i^2} \dots g^{a_t i^t} \\ &= g^{a_0} \cdot g^{a_1 i} \cdot g^{a_2 i^2} \dots g^{a_t i^t} \\ &= g^{\sigma_i} \end{aligned} \tag{4}$$

$$(\because \sigma_i = a_0 + a_1 i + a_2 i^2 \dots + a_t i^t)$$

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

2.2. Pedersen의 검증 가능한 비밀 공유 방식

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

로토클은 다음과 같다.

먼저,  $g, h \in G_q$ 를 정의하자. 여기서  $G_q$ 는  $q$ 차의  $Z_q^*$ 의 유일한 부분집합이고,  $g$ 는  $G_q$ 의 생성자이다. 만약 요소  $a \in Z_q^*$ 가  $G_q$ 에 있다면  $a \in G_q \leftrightarrow a^q = 1$ 이다.  $Z_q$ 는 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  $F(0) = s$ 를 만족하는 최소한  $k-1$ 차의 다항식  $F \in Z_q[x]$ 를 선택하며, 그리고 나서  $s_i = F(i)$  (for  $i = 1, \dots, n$ )를 계산한다. 여기서,  $F(x) = s + F_1 x + \dots + F_{k-1} x^{k-1}$ 라 하자. 또한 딜러는 임의적으로  $G_1, \dots, G_{k-1} \in Z_q$ 를 선택하고  $i = 1, \dots, k-1$ 에 대해서  $F_i$ 에 위탁을 수행할 때  $G_i$ 를 사용한다. 즉, 딜러는  $E_i = E(F_i, G_i)$ 를 공개한다.
- 3)  $G(x) = t + G_1 x + \dots + G_{k-1} x^{k-1}$ 이라 하고,  $t_i = G(i)$  ( $i = 1, \dots, n$ )라 하자. 그리고 나서 딜러는 비밀스럽게  $(s_i, t_i)$ 를  $P_i$ 에게 전송한다. 여기서  $i = 1, 2, \dots, n$ 이다.

$P_i$ 가 그의 공유  $(s_i, t_i)$ 를 수신하면,  $P_i$ 는 식 (5)를 검증한다.

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

위의 식 (5)를 만족하면  $P_i$ 는 정당한 비밀의 몫을 공유한 것이고, 그렇지 않으면 잘못된 비밀의 몫을 수신한 것이다. 이것의 증명은 식 (6)과 같다.

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

$$\begin{aligned} \prod_{j=0}^{k-1} E_j^{i^j} &= E_0^{i^0} E_1^{i^1} \dots E_{k-1}^{i^{k-1}} \\ &= E(s, t) \cdot [E(F_1, G_1)]^i \dots [E(F_{k-1}, G_{k-1})]^{i^{k-1}} \\ &= g^s h^t \cdot [g^{F_1} h^{G_1}]^i \dots [g^{F_{k-1}} h^{G_{k-1}}]^{i^{k-1}} \\ &= g^{s + F_1 i + F_2 i^2 + \dots + F_{k-1} i^{k-1}} h^{t + G_1 i + G_2 i^2 + \dots + G_{k-1} i^{k-1}} \end{aligned} \quad (6)$$

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

### 2.3. 연합 난수 비밀 공유 방식

지금까지 논의된 비밀 공유 방식은 모든 참여자가 신임하는 딜러를 요구한다. 하지만 이번 절에서 논의하는 비밀 공유 방식은 참여자 각자가 딜러처럼 행동함으로써 딜러가 필요 없으며, 각 참여자는 임의의 난수를 공유하는 프로토콜이다. 즉, 자신의 비밀 공유  $a_i$ 를 선택하여,  $(t, n)$ -비밀 공유 기법을 이용하여 각 참여자에게 부분 비밀  $a_{ij}$ 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  $(s_1, \dots, s_n) \xleftrightarrow{(t, n)} s \pmod q$ 가 되는 비밀 공유  $s_i$ 를 갖는다. 일반적인  $(t, n)$ -비밀 공유 기법에서처럼 비밀 값  $s$ 는 모든 참여자들로부터 그리고 심지어는  $t-1$  참여자들의 연합으로부터도 비밀이 유지된다.

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

- 1)  $P_i$ 는  $a_i$ 의 분배를 위한  $(t, n)$ -비밀 공유 기법을 가지고 있다고 가정한다.
- 2) 각  $P_i$ 는  $a_i \in Z_q$ 를 임의로 선택한다.
- 3) 각  $P_i$ 는  $(t, n)$ -비밀 공유 기법을

이용해  $a_i$ 에 대한 부분 공유인  $a_{ij}$ 를 계산한다.

4) 각 참여자인  $P_j$ 에게  $a_{ij}$ 를 전송한다.

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

$$a_{11} = a_1x_1^2 + b_1x_1 + a_1, \quad a_{12} = a_1x_2^2 + b_1x_2 + a_1, \quad \dots, \quad a_{1n} = a_1x_n^2 + b_1x_n + a_1$$

$$a_{21} = a_2x_1^2 + b_2x_1 + a_2, \quad a_{22} = a_2x_2^2 + b_2x_2 + a_2, \quad \dots, \quad a_{2n} = a_2x_n^2 + b_2x_n + a_2$$

⋮

$$a_{n1} = a_nx_1^2 + b_nx_1 + a_n, \quad a_{n2} = a_nx_2^2 + b_nx_2 + a_n, \quad \dots, \quad a_{nn} = a_nx_n^2 + b_nx_n + a_n$$

6) 각  $P_i$ 는  $a_{1i}, \dots, a_{ni}$ 를 수신하고, 부분 비밀 공유들  $s_i = a_{1i} + \dots + a_{ni}$ 를 계산한다. 예를 들어서,  $P_1$ 은  $a_{11}, \dots, a_{n1}$ 을 계산할 수 있고,  $s_1 = a_{11} + \dots + a_{n1}$ 을 계산할 수 있다. 각 참여자들이 계산하는 부분 비밀 공유는 다음과 같다.

$$s_1 = a_1x_1^2 + b_1x_1 + a_1 + \dots + a_nx_1^2 + b_nx_1 + a_n = (a_1 + \dots + a_n)x_1^2 + (b_1 + \dots + b_n)x_1 + (a_1 + \dots + a_n)$$

⋮

$$s_n = a_1x_n^2 + b_1x_n + a_1 + \dots + a_nx_n^2 + b_nx_n + a_n = (a_1 + \dots + a_n)x_n^2 + (b_1 + \dots + b_n)x_n + (a_1 + \dots + a_n)$$

7) 비밀  $s = \sum_{i=1}^n a_i$ 의 (t,n)-임계치 기법의 부분 비밀 공유는  $(s_1, \dots, s_n)$ 이다.

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,  $a_{1n}$ 의 값을 검증하기 위해서  $P_1$ 은  $g^{a_1}, g^{b_1}, g^{a_1}$ 을 공개한다.

이렇게 공개된 값들을 이용해서 식 (4)에 적용하면  $a_{1n}$ 을 수신하는 참여자  $P_n$ 은 수신된  $a_{1n}$ 이 정당한 값인지를 검증할 수 있다. 식 (7)을 참고하라.

$$g^{a_{1n}} = g^{a_1} \cdot g^{b_1x_n} \cdot g^{a_1x_n^2} = g^{a_1} \cdot g^{b_1x_n} \cdot g^{a_1x_n^2} = g^{a_{1n}} \quad (7)$$

$$(\because a_{1n} = a_1 + a_1x_n + b_1x_n^2)$$

이러한 방법을 이후 Feldman-JR-VSS 이라고 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 다항식의 상수항이 '0'이라는 것을 검사하는 것에 의해 수행된다. 즉, 2절에서, 각 참여자  $P_i$ 는 비밀 공유  $a_i$ 의 값이 '0'이 되도록 선택한다. 만약 모든 참여자들이 이러한 프로토콜을 따른다면, 비밀 s의 값인 다항식의 상수항  $a_1 + \dots + a_n$ 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.

#### 2.4. 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  $u \times v$ 를 계산하는 프로토콜이다.

각각 t 차수의 다항식에 의해 공유된 주어진 u, v는 각 참여자들이 자신들의 u, v공유를 지역적으로 곱한다. 결과는 2t 차수 다항식의  $u \times v$  공유가 될 것이다. 따라서, 값  $u \times v$ 는 2t+1의 정당한 공유들의 집합으로부터 재구성된다.

t=3인 경우, u와 v는 3차 다항식이 되고,

이 두 비밀의 곱은 6차 다항식이 된다. 즉, 7개의 적당한 공유들의 집합으로 비밀이 재구성된다. 연합 영 비밀 공유 프로토콜을 사용하는 추가적인 재 랜덤화 절차는 곱해진 비밀들의 비밀을 보호하는데 필요하다. 이후 Multi-Zero라 표기한다.

III. 단일 서버 매직 잉크 서명 프로토콜

단일 서버 매직 잉크 서명 프로토콜은 유일한 하나의 서버가 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.

1. 단일 서버 DSS 매직 잉크서명 프로토콜

1.1. 파라미터

DSS 서명 방식의 시스템 파라미터는 다음과 같다.

- 1)  $p$ : 소수  $|p| = 512 + 64i, i=0, \dots, 8$
- 2)  $q$ : 160 비트 소수,  $q | p-1$ .
- 3)  $g$ :  $g = h^{(p-1)/q} \text{ mod } p$   
( $g > 1, 1 < h < p-1$ )
- 4)  $H$ : 160 비트 해쉬 함수(SHA)
- 5)  $x$ : 비밀키 ( $0 < x < q$ )
- 6)  $y$ : 공개키 ( $y = [g^x]_p$ )

1.2. 서명 프로토콜

단일 서버 방식의 서명 프로토콜은 그림1과 같다. 여기서  $m, r, s$ 가 메시지  $M$ 의 서명문 쌍이 된다.

1.3. 서명문 검증

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

$$\begin{aligned}
 r &= g^{ms^{-1}} y^{rs^{-1}} \\
 &= g^{(m+xs)^{-1}} \\
 &= g^{k^{-1}b} = \bar{r}^b
 \end{aligned}$$

2. 단일 서버 KCDSA 매직 잉크 서명 프로토콜

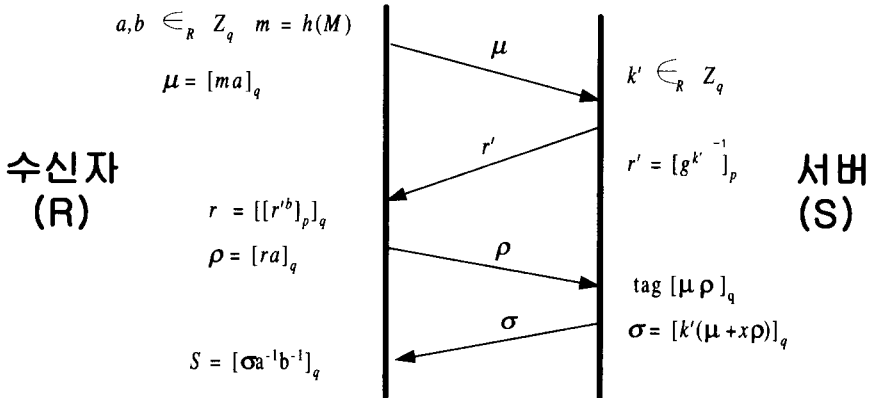


그림 1 단일 서버 DSS 서명 프로토콜

2.1. 파라미터

KCDSA의 시스템 파라미터는 다음과 같다.

- 1)  $p$ : 소수  $|p| = 512 + 256i, i = 0, \dots, 6$
- 2)  $q$ : 소수  $q | p-1, |q| = 128 + 32j, j = 0, \dots, 4$

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

$$R \triangleq h((W^a)')$$

IV. 분배된 매직 잉크 서명 프로토콜

분배된 개념의 매직 잉크 서명 프로토콜은 시스템의 안정화와 유효성을 증가시키고, 믿음의 집중을 분산시켜 일정의 정족수가 모

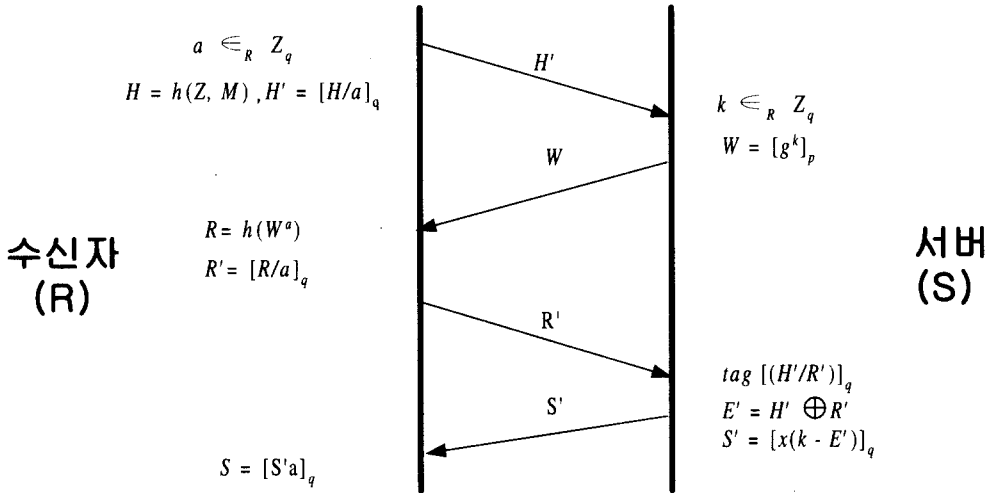


그림 2 단일 서버 KCDSA 서명 프로토콜

- 3)  $g$ : GF(p) 상에서 위수가 q인 수
- 4)  $h$ :  $|q|$  길이의 출력값을 갖는 충돌 회피성 해쉬함수
- 5)  $x$ : 비밀키 ( $0 < x < q$ )
- 6)  $y$ : 공개키 ( $y = [g^{x^{-1}}]_p$ )
- 7)  $Z$ : Cert\_Form의 해쉬 값

2.2. 서명 프로토콜

KCDSA에 기초인 단일 서버 서명 프로토콜은 그림2와 같다.

2.3. 검증

서명문의 검증은 다음의 관계식을 이용한다.

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

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

여야 서명과 추적이 가능하다.

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 기법도 적용 가능하다.

1. tag 계산

분배된 DSS 매직 잉크 서명 프로토콜에서 tag 계산은 다음과 같다.

$S_1, \dots, S_n$  사이에 공유된 비밀  $\rho \pmod q$  가 R에 의해 주어지면,  $\rho$ 와  $\rho^{-1}$  상에 정보를 밝히지 않고  $\rho^{-1} \pmod q$  공유를 생성한다.

각 서버  $S_i$ 는  $\rho$ 의 (t, n) 비밀 공유에 대

응하는 공유  $\rho_i$ 를 갖는다. 즉,

$(\rho_1, \dots, \rho_n) \xrightarrow{(t, n)} \rho$ 이다.  $\rho^{-1}$ 에 대한 공유들의 계산은 아래와 같이 수행된다.

- 1) Feldman-JR-VSS 프로토콜을 이용하여 랜덤 요소  $d \in Z_q$ 의  $(t, n)$  공유를 생성한다. 즉,  $(d_1, \dots, d_n) \xrightarrow{(t, n)} d$ 이다.
- 2) 각 서버들은  $(2t-1)$ 차 항  $[\rho_i d_i]_q$ 을 Multi 수행한 후에, 각 서버  $S_i$ 는 "secret" 0의 공유  $b_i$ 를 갖는다.
- 3) 각 서버들은  $\rho_i d_i + b_i$ 를 공개하고 대응하는  $(2t-1)$ 차 다항식을 Interpolate  $(\rho_1 d_1 + b_1, \dots, \rho_n d_n + b_n) \xrightarrow{(2t, n)} \rho d$ 을 계산한다.
- 4) 각 서버들은  $a_i \Leftarrow \alpha^{-1} d_i \pmod q$ 를 설정함으로써  $\rho^{-1}$ 의 공유  $a_i$ 를 계산한다.
- 5) 각 서버들은 자신의  $[(\mu_1 a_1, \dots, \mu_n a_n)]_q$ 를 Multi로 수행하여 tag를 각각 가지고 있다. 그러므로 서 추후에 서명 쌍들이 사용될 경우  $[(\mu_i a_i)]_q \xrightarrow{(2t, n)} \mu \alpha$ 인 tag를 계산하며, 서명 쌍으로 온 것과

비교함으로써 서명 쌍을 추적한다.

분배된 KCDSA 매직 잉크 서명 프로토콜의 tag 계산은 위 과정에서  $\rho$ 를  $R'$ 으로,  $\mu$ 를  $H'$ 으로 치환하여 계산하면 된다.

2. 분배된 DSS 매직 잉크 서명 프로토콜

2.1. 키 생성

분배된 DSS 매직 잉크 서명 프로토콜을 수행하기 전 각 서버들은 Feldman-JR-VSS를 수행하여 서버  $S_i$ 가 비밀값  $x$ 에 대한 그의 몫인 비밀 입력  $x_i$ 을 유지한다. 각 서버들은 수신된 비밀 정보에 대해  $g^{x_i}$  하여 공개한다. 각 서버  $S_i$ 는 Exp-Interpolate를 이용하여 공개키  $y = g^x$ 을 계산하여 공개한다.

2.2. 서명 프로토콜

- 1) Feldman-VSS를 이용하여  $R$ 은 서버  $S_i$ 에 대한  $\mu$ 에 대한 그의 몫  $\mu_i$ 를  $S_i$ 에게 전송한다.
- 2) Feldman-JR-VSS를 수행하여 서버  $S_i$ 가 랜덤 값  $k'$ 에 대한 그의 몫인 비밀

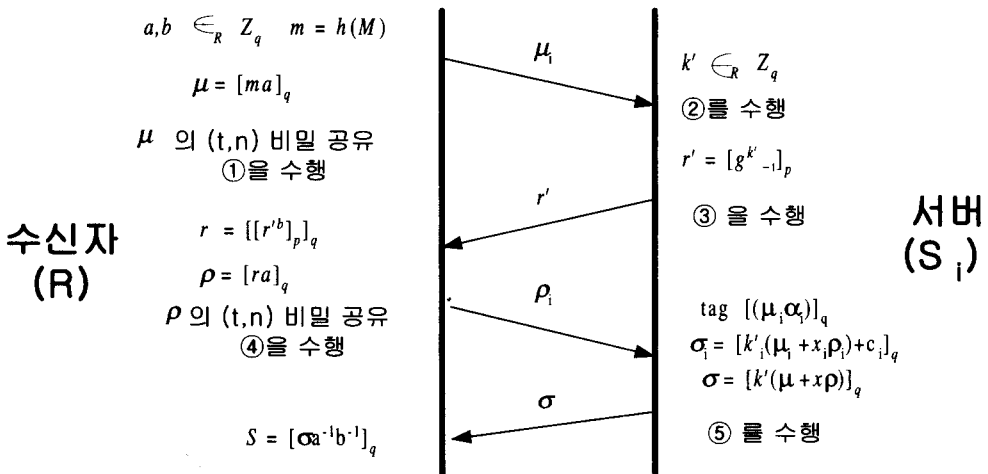


그림 3 분배된 DSS 서명 프로토콜



입력  $\bar{k}_i$  을 유지하고,  $g^{\bar{k}_i}$  을 공개한다.

- 3) Feldman-JR-VSS을 이용하여  $d$ 을 공유하고 각 서버는  $g^{d_i}$ 을 공개한다.  $(2t-1)$ 차 다항식에 의존하는  $[e_i = \bar{k}_i d_i + c_i]_q$ 의 Multi-Zero를 수행하여  $e_i$ 을 구하고,  $Interpolate(e_i, \dots, e_n)$

$\xrightarrow{(2t, n)}$   $e$ 을 계산하고, 각 서버  $S_i$ 는 지역적으로  $\bar{r} = [g^{\bar{k}^{-1}}]_q$ 을 계산한다.

- 각 서버는  $d$ 를 Exp-Interpolate  $\rightarrow g^d$
- $Interpolate(e_i, \dots, e_n) \xrightarrow{(2t, n)}$   $e$
- $e^{-1} \rightarrow \bar{k}^{-1} d^{-1}$
- $(g^d)^{e^{-1}} \bmod p \bmod q \rightarrow g^{\bar{k}^{-1}} (\bar{r})$ 을 계산하여 각 서버들은 R에게 전송한다.

- 4) Feldman-VSS을 이용하여 R은 서버  $S_i$ 에 대한  $\rho$ 에 대한 그의 몫  $\rho_i$ 을  $S_i$ 에게 전송한다.

- 5) 서버들 간에  $[x_i \rho_i + c_i]_q$ 을 Multi-Zero 프로토콜을 수행한 후  $Interpolate(x_1 \rho_1, \dots, x_n \rho_n) \xrightarrow{(2t, n)}$   $x\rho$  값을 계산한 후, 각 서버  $S_i$ 는  $(2t-1)$ 차  $\sigma_i$ 을 Multi-Zero로 수행하여 각  $S_i$ 는 다항식에 의존하는  $[\bar{k}_i(\mu_i + x\rho) + c_i]_q \rightarrow$

$\sigma_i$ 를 계산하고,  $Interpolate(\sigma_1, \dots, \sigma_n)$

$\xrightarrow{(2t, n)}$   $\sigma$  하여 서버는  $\sigma$ 을 R에게 전송한다.

### 3. 분배된 KCDSA 매직 잉크 서명 프로토콜

#### 3.1. 키 생성

분배된 KCDSA 매직 잉크 서명 기법을 수행하기 전 각 서버들은 Feldman-JR-VSS을 수행하여 서버  $S_i$ 가 비밀값  $x$ 에 대한 그의 몫인 비밀 입력  $x_i$ 를 유지한다.

서명 프로토콜 전에 Feldman-JR-VSS을 이용하여  $d$ 을 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  $e_i$ 을 구하고,  $Interpolate(e_i, \dots, e_n) \xrightarrow{(2t, n)}$   $e$ 을 계산하고, 각 서버  $S_i$ 는 지역적으로  $y = [g^{x^{-1}}]_q$ 을 계산한다.

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

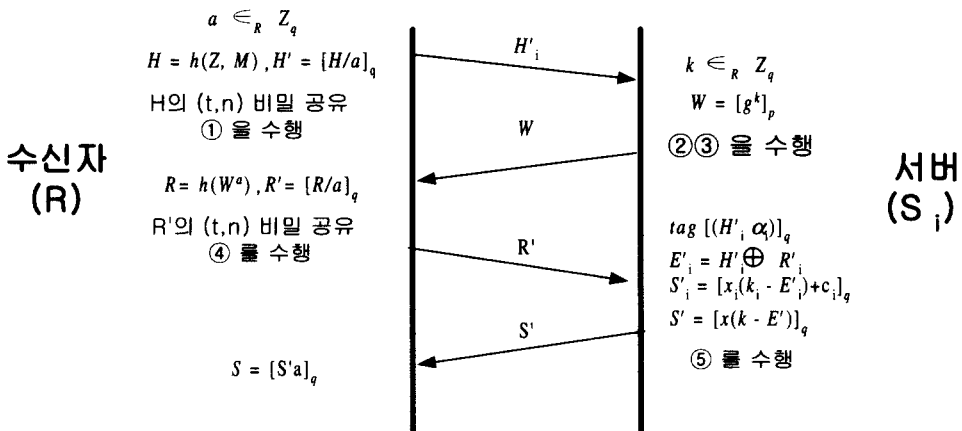


그림 4 분배된 KCDSA 서명 프로토콜

3.2. 서명문 생성

- 1) R은 Feldman-VSS를 이용하여  $H$ 의 공유된 값  $(H_1, \dots, H_n)$ 을 서버  $S_i$ 에 대하여 그의 몫  $H_i$ 을  $S_i$ 에게 전송한다.
- 2) Feldman-JR-VSS를 수행하여 서버  $S_i$ 가 랜덤 값  $k$ 에 대한 그의 몫인 비밀 입력  $k_i$ 을 유지하고,  $g^{k_i}$ 을 공개한다.
- 3) 각 서버  $S_i$ 은 Exp-Interpolate를 이용하여  $W=[g^{k_i}]_p$ 을 계산하여 R에게 전송한다.
- 4) R은 Feldman-VSS를 이용하여  $R'$ 의 공유된 값  $(R'_1, \dots, R'_n)$ 을 서버  $S_i$ 에 대하여 그의 몫  $R'_i$ 을  $S_i$ 에게 전송한다.
- 5)  $(2t-1)$ 차 항의  $[x_i(k_i - E_i) + c_i]_q$ 을 Multi-Zero로 수행하므로서 각 서버  $S_i$ 는  $S'_i$ 을 계산하고,  $Interpolate(S'_1, \dots, S'_n) \xrightarrow{(2t, n)} S'$  하여  $S'$ 을 R에게 전송한다.

V. 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  $W$ 을 계산할 때, Feldman의 검증 가능한 비밀 공유 방식을 적용함으로써  $k$ 의 값을 밝히지 않고  $W=g^k \bmod p$ 를 계산하도록 했다. 표 1은 분배된 DSS 매직 잉크

서명 프로토콜과 분배된 KCDSA 매직 잉크 서명 프로토콜에서 필요한 비밀 공유 기법들의 횟수를 비교한 것이다.

과정 서명 공유	키생성		서명문생성	
	DSS	KCDSA	DSS	KCDSA
Feldman-VSS	0	0	2	1
Feldman-JR-VSS	1	2	2	1
Multi	0	1	3	1
Interpolate	0	1(2t-1차)	3(2t-1차)	1(2t-1)
Exp-Interpolate	1	1	1	1

표 1 DSS 비밀 공유 방식과 KCDSA 비밀 공유 방식의 비교

표1에서 보듯이, 분배된 KCDSA 매직 잉크 서명 방식이 키 생성 단계에서는 필요한 비밀 공유 방식이 더 많이 필요함으로 비효율적이지만, 서명문 생성 단계에서는 상당히 효율적인 것을 알 수 있다.

VI. 결론

본 논문에서는 단일 서버 DSS와 KCDSA 매직 잉크 서명에 다중 서버를 갖는 분배 개념들을 추가함으로써 시스템의 안전성을 증가시키고, 서비스의 유용성을 증가시킨 분배에 대한 기술을 분석했으며 이를 분배된 DSS와 KCDSA 매직 잉크 서명에 적용하였다. 두 방식을 서로 비교한 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] 백중현, 염홍열 "매직 잉크 서명 방식을 이용한 전자현금 프로토콜 설계", CISC'97, pp 354-367, 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

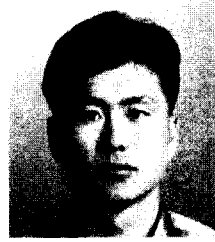
[8] T.Pedersen, "Distributed provers with applications to undeniable signatures", In Proc. EUROCRYPTO 91, 1991

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

□ 著者紹介

류 영 규 (Young-Kyu Lyu)

학생회원



1998년 2월 : 순천향대학교 전자공학과 졸업  
1998년 3월 ~ 현재 : 순천향대학교 전자공학과 석사과정

〈관심분야〉 전자상거래, 부호이론, 네트워크 보안

윤 호 선 (Ho-Sun Yoon)

정회원



1997년 2월 : 순천향대학교 전자공학과 졸업  
1999년 2월 : 순천향대학교 대학원 전자공학과 석사  
1999년 3월 ~ 현재 : 순천향대학교 산업기술연구소 연구원

〈관심분야〉 분산처리시스템, 전자상거래, 암호이론

염 홍 열 (Heung-Youl Youm)

종신회원



1981년 : 한양대학교 전자공학과 졸업  
1983년 : 한양대학교 대학원 전자공학과 석사  
1990년 : 한양대학교 대학원 전자공학과 박사  
1982년 12월 ~ 1990년 9월: 한국전자통신연구소 선임연구원

1990년 3월 ~ 현재 : 순천향대학교 공과대학 전기전자공학부 부교수

1997년 3월 ~ 현재 : 순천향대학교 산업기술연구소 소장

1997년 3월 ~ 현재 : 한국통신정보보호학회 총무이사

〈관심분야〉 암호이론, 부호이론, 이동통신분야