情報保護學會論文誌
第11券 第3號, 2001. 6

Feige-Fiat-Shamir 은닉전자서명에 기반한 추적 가능한 전자화폐 시스템의 설계*

박 왕 석**, 박 창 섭***

Design of traceable Electronic Cash System based on Feige-Fiat-Shamir blind digital signature*

Wang-Suk Park**, Chang-Seop Park***

요 약

전자상거래는 실생활의 물질적인 상거래에 비해 많은 장점을 가지고 있지만, 인터넷이라는 가상공간을 통해 상거래가 이루어지므로 상호간의 신용문제, 지불 방법 등 개선되어야 할 부분이 많이 있다. 이를 위한 초기의 전자화폐는 사용자의 개인정보에 대한 유출을 막기 위해 은닉전자서명을 이용하여 설계되었으며, 사용자의 완전한 익명성이 보장되었다. 본 논문에서는 계산적으로 효율성이 매우 좋은 Feige-Fiat-Shamir 은닉전자서명을 제안하였고, 제안한 Feige-Fiat-Shamir 은닉전자서명을 이용하여, 전자화폐가 불법적으로 사용될 경우 전자화폐를 추적할 수 있는 추적 가능한 전자화폐 시스템을 설계하였다.

ABSTRACT

E-commerce has various advantages such as saving the cost and no constraint on time and space, unlike real-world commerce. On the other hand, e-commerce has some important issues to solve since the commerce is conducted on the cyberspace. The issues are a mutual confidence of parties participating in the commerce as well as a method of payments. On early days, electronic cash was designed using blind digital signature to protect the personal information from being exposed and to provide the perfect anonymity for user. In this paper, a new blind signature scheme based on Feige-Fiat-Shamir digital signature is proposed, which is very efficient compared with the other schemes in terms of the computational complexity. Also, a traceable Electronic Cash System which is based on the proposed blind digital signature is designed, which has a nice feature of identifying the spender in case of using the money illegally.

keyword : *Electronic Cash, Traceability, Blind Signature, Feige-Fiat-Shamir Signature,*

Ⅰ. 서 론

인터넷을 통한 전자상거래는 현재 많이 사용하고 있는 실생활의 상거래 보다 시간과 공간의 제약을 거의 받지 않는다는 편리함과 유용성을 가지고 있으며, 사용자들에게 중간 유통단계를 줄여 주어서 비용을 절감시킬 수 있는 많은 장점을 가지고 있다. 반면 실생활에서의 물리적인 상거래는 주로 사용자들 간의 직접적인 대면을 통해 이루어지지만, 인터넷에서의 상거래는 인터넷이라는 가상적인 공간에서

* 본 연구는 단국대학교 2000년도 교내 연구비에 의해 지원되었음
** MIsecurity 주임 연구원 (kingston@misecurity.com)
*** 단국대학교 전자.컴퓨터학부 교수 (csp0@ns.anseo.dankook.ac.kr)

상거래가 이루어지므로 상호간의 신용문제와 지불 방법 등의 문제를 가지고 있다. 특히 지불 방법의 경우 비밀성이 유지되지 않는 개방된 인터넷에서 이루어져야 되므로, 거래 정보의 위조나 복사를 통한 부당 이익 등 많은 논쟁의 여지를 가지고 있다. 이러한 문제점들을 해결하기 위한 방법으로 인터넷 상에서 지불이 가능한 전자화폐 시스템이 제시되었다[5~8~9~15]. 전자화폐 시스템은 실생활에서 화폐를 사용할 때와 마찬가지로 사용자의 익명성을 제공하여야 한다. 즉, 상점이 자신과 거래한 사용자의 신상과 화폐의 일련번호 등을 기록하는 방법 등을 사용하지 않는 한, 사용자가 사용한 화폐만으로는 사용자의 거래 내역을 포함하여 사용자의 신상에 대한 정보를 알 수 없어야 한다. 이러한 특성을 위해 1982년 David Chaum은 은닉전자서명을 처음으로 제안하였는데,[8] 이는 전자서명의 변형된 형태[6~10~11~12]로 사용자의 익명성을 제공하여 준다. 그 후 은닉전자서명을 이용한 많은 전자화폐 시스템이 제시되었으나, 은닉전자서명은 사용자의 완전한 은닉성을 제공하기 때문에 돈 세탁, 불법적인 상품(무기, 마약) 구매에 사용될 경우, 사용자에 대한 추적이 불가능하게 된다. 이러한 문제점을 보완하기 위해 합법적인 경우에는 익명성을 제어할 수 있는 방법들이 연구 중에 있다. 즉, 실생활에서는 합법적인 거래에 사용된 전자화폐에 대한 사용자의 익명성은 지켜지며, 불법적인 거래에 대해서는 법률 등과 같은 일정한 기준을 두어 사용자를 추적할 수 있는 전자화폐 시스템이 필요하다.

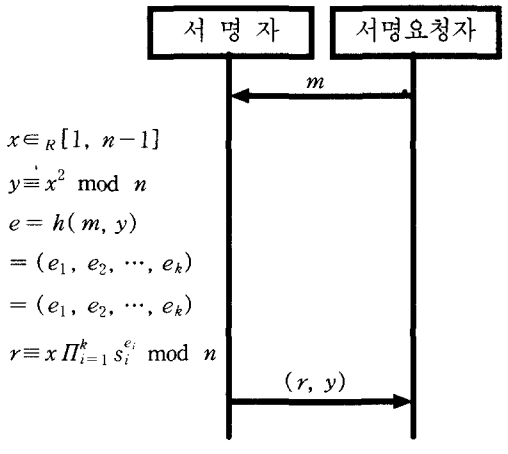
본 논문에서는 기존에 제시되었던 Feige-Fiat-Shamir 전자서명[1]을 이용하여 새로운 은닉전자서명을 제안하였으며, 이를 전자화폐 시스템에 적용하였다. 본 논문에서 제안한 전자화폐 시스템은 사용자의 합법적인 전자화폐 사용시에는 사용자에게 익명성을 제공하며, 불법적인 전자화폐의 사용시에는 은행과 신용기관의 협력으로 사용자를 추적할 수 있는 익명성 제어 기능을 가지고 있다.

Ⅱ. 새로운 Feige-Fiat-Shamir 은닉전자서명

본 장에서는 서명 기법 중, 두 개의 큰 소수 p, q로 이루어진 법 $n=pq$에 대한 제곱근을 구하는 문제의 어려움에 안전성의 기반을 두고 있는 Feige-Fiat-Shamir(F-F-S) 서명에 대해 설명하고, F-F-S 서명을 기반으로 새로운 은닉전자서명인 F-F-S 은닉전자서명을 설계, 분석하였다.

2.1 Feige-Fiat-Shamir 전자서명

F-F-S 전자서명은 Fiat-Shamir 인증 프로토콜에 기반을 두어, 이전에 Fiat와 Shamir에 의해 제시되었던 전자서명 방법의 변형된 형태로, 임의의 길이의 메시지를 k비트의 해쉬값으로 변형시켜주는 해쉬함수 $h: \{0,1\}^* \rightarrow \{0,1\}^k$가 사용된다[3]. F-F-S 서명을 사용하기 위해서는 우선 신뢰할 수 있는 제3자(trusted third party(TTP))가 비밀키로 보관하는, 임의의 서로 다른 큰 소수 p와 q를 생성하여, 공개키로 사용하는 법(modulus) $n(=pq)$을 계산한다. 이러한 사전 작업은 TTP가 아닌 서명자 개인에 의해서도 이루어 질 수 있다. 서명자는 생성된 n을 이용하여 다음과 같이 개인키와 공개키를 만듦으로써 서명에 사용할 키를 생성한다. 우선 임의의 양의 정수 k를 선택하고, Z_n^*의 원소인 k개의 서로 다른 임의의 정수 $s_1, s_2, \cdots, s_k \in Z_n^*$를 선택한다. 이때 k를 보안 파라미터라 하며, 보통 $k=128$이다. 선택되어진 임의의 정수 $\{s_1, s_2, \cdots, s_k\}$가 서명자의 개인키가 되며, 개인키 각각의 원소 s_i를 이용하여 $v_i \equiv s_i^{-2} \bmod n \ (1 \le i \le k)$을 계산하고, 계산되어진 $\{v_1, v_2, \cdots, v_k\}$가 서명자의 공개키가 된다. TTP에 의해 계산된 법 n과 서명자에 의해 생성된 개인키, 공개키를 이용하여 그림 1과 같은 방법으로 메시지 m에 대한 F-F-S 전자서명이 이루어진다.



〔그림 1〕 F-F-S 전자서명 생성 프로토콜

서명 요청자는 서명을 받고자하는 메시지 m을 서명자에게 전송한다. 서명자는 임의의 정수 x를 선택하여 $y \equiv x^2 \bmod n$을 계산한 후, 전달받은 메시지 m과 y에 해쉬함수 h를 적용하여 $e = h(m, y) = (e_1, e_2, \cdots, e_k)$를 계산한다. 이때 사용되어지는 해쉬함수 h는 임의의 길이의 메시지를 k비트의 해쉬값으로 변형시켜주는 해쉬함수 $h: \{0,1\}^* \rightarrow \{0,1\}^k$가 사용된다. 계산한 e를 이용하여 $r \equiv x \Pi_{i=1}^{k} s_i^{e_i} \bmod n$을 계산한다. 이러한 과정을 통해 메시지 m에 대한 서명 (r, y)가 생성되며, 생성된 서명 (r, y)를 서명 요청자에게 전송한다. 서명 요청자는 전송받은 메시지 m에 대한 서명 (r, y)가 올바른 서명인가를 알아보기 위하여 $r^2 \Pi_{i=1}^{k} v_i^{e_i} \equiv y \bmod n$ 인지를 확인한다. 만약 (r, y)가 메시지 m에 대한 올바른 서명이라면 다음과 같은 계산식에 의해 등식이 성립한다.

$$r^2 \Pi_{i=1}^{k} v_i^{e_i} = x^2 \Pi_{i=1}^{k} s_i^{2e_i} \Pi_{i=1}^{k} v_i^{e_i} \quad (1)$$
$$= x^2 \Pi_{i=1}^{k} s_i^{2e_i} s_i^{-2e_i}$$
$$= x^2 \equiv y \bmod n$$

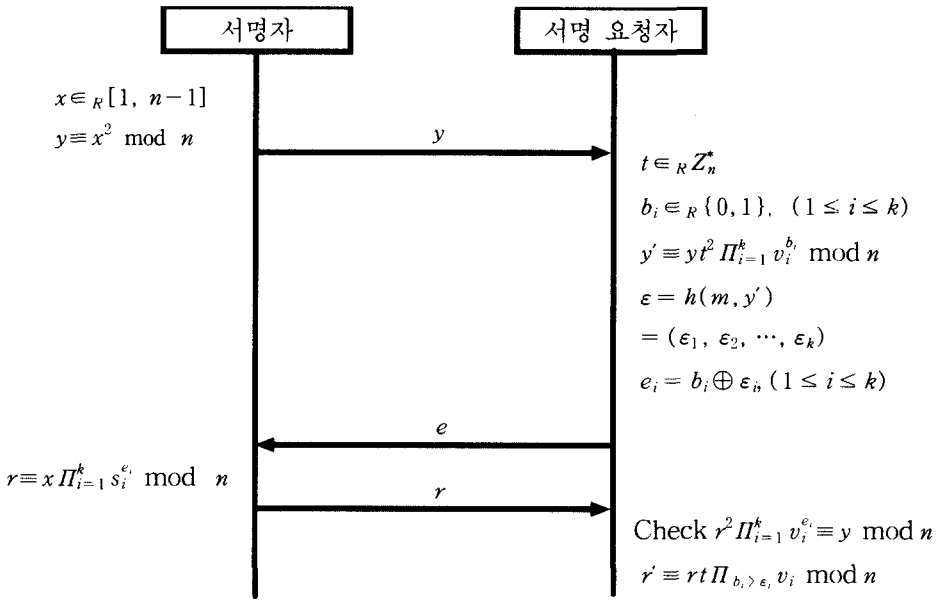
F-F-S 서명은 법 $n = pq$에 대해서 제곱근(square root) $s_i \equiv v_i^{-1/2} \bmod n$ 계산의 어려움에 그 이론적 기반을 두고 있다. 이러한 법 n에 대한 제곱근을 구하는 문제는 법 n을 소인수 분해하는 문제와 그 난이도가 동일하다. F-F-S 서명은 RSA를 이용한 전자 서명보다 법에 대한 곱셈 연산의 수가 매우 적다. 예를 들어, RSA의 경우 길이가 768인 법 n을 사용할 경우, 평균 1152번의 법에 대한 곱셈 연산을 필요로 한다. 반면, F-F-S 서명은 선택되어진 적당한 k에 대해 약 $k/2$번의 법에 대한 곱셈 연산을 필요로 한다. 즉, 법 n의 길이가 768이고, k가 128일 때, 필요한 법에 대한 곱셈 연산은 평균 66번으로 약 6%정도이다. 따라서, 서명에 걸리는 시간이 매우 적다는 장점을 가지고 있다.

2.2 새로운 Feige-Fiat-Shamir 은닉전자서명

본 장에서는 F-F-S 전자서명에 은닉요소(blinding factor) $(b_1, b_2, \cdots, b_k)$를 이용하여 설계한 F-F-S 은닉전자서명의 프로토콜, 안전성, 효율성에 대해 알아본다.

2.2.1 프로토콜

F-F-S 은닉전자서명은 F-F-S 서명과 마찬가지로 신뢰할 수 있는 제3자(TTP)가 비밀키로 보관하는 임의의 서로 다른 큰 소수 p와 q를 생성하여, 공개키로 사용하는 법(modulus) $n(=pq)$을 계산



〔그림 2〕 Feige-Fiat-Shamir 은닉전자서명

한다. 이러한 사전작업은 F-F-S 서명과 마찬가지로 서명자에 의해 이루어질 수도 있다. 서명자는 n을 이용하여 서명에 사용하는 개인키와 서명을 확인하는 공개키를 생성한다. 우선 서명자는 개인키를 생성하기 위해 적당한 크기의 양의 정수 k를 선택하고, Z_n^*의 원소인 k개의 서로 다른 임의의 정수 $s_1, s_2, \cdots, s_k \in Z_n^*$를 선택한다. 선택된 임의의 정수 $\{s_1, s_2, \cdots, s_k\}$가 서명자의 개인키가 되며, 개인키 각각의 원소 s_i를 이용하여 $v_i \equiv s_i^{-2} \bmod n\ (1 \le i \le k)$을 계산하고, 계산되어진 $\{v_1, v_2, \cdots, v_k\}$가 서명자의 공개키가 된다. 계산된 법 n과 서명자에 의해 생성된 개인키, 공개키, 은닉요소를 이용하여 그림 2와 같은 방법으로 주어진 메시지 m에 대한 F-F-S 은닉전자서명이 이루어진다.

프로토콜의 첫 번째 단계에서 서명자는 1과 $n-1$ 사이의 임의의 난수 x를 선택하여 $y \equiv x^2 \bmod n$을 계산하고, y를 서명 요청자에게 전송한다. 난수 x는 프로토콜의 안전성을 위해 중복을 피하여 선택되어져야 한다. 두 번째 단계에서 서명 요청자는 Z_n^*상의 임의의 난수 t와 은닉요소 $b = (b_1, b_2, \cdots, b_k)$, $b_i \in \{0, 1\}$를 선택하고, 서명자의 공개키 $\{v_1, v_2, \cdots, v_k\}$를 이용하여 $y' \equiv yt^2 \Pi_{i=1}^k v_i^{b_i} \bmod n$을 계산한다. 계산되어진 y' 및 메시지 m에 임의의 길이의 입력값을 k비트의 해쉬값으로 변형시켜주는 해쉬함수 h를 이용하여 $\varepsilon = h(m, y') = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_k)$를 계산한다. 계산되어진 ε과 은닉요소 b의 i번째 비트에 대한 배타적 논리합(exclusive or)을 통해 $e = (e_1, e_2, \cdots, e_k)$를 계산한다. 즉, $e_i = b_i \oplus \varepsilon_i$이며 i는 1부터 k까지이다. 서명 요청자는 위의 과정을 통해 계산되어진 e를 서명자에게 전송한다. 세 번째 단계에서 서명자는 전송 받은 e와 자신이 선택한 난수 x 그리고, 자신의 개인키 s_i를 이용하여 $r \equiv x \Pi_{i=1}^k s_i^{e_i} \bmod n$을 계산하여 서명 요청자에게 전송한다. 네 번째 단계에서 서명 요청자는 등식 $r^2 \Pi_{i=1}^k v_i^{e_i} \equiv y \bmod n$이 성립되는지 확인하고, 만약 등식이 성립할 경우 $r' \equiv rt \Pi_{b_i > \varepsilon_i} v_i \bmod n$을 계산한다. 위의 과정을 통해 서명요청자는 메시지 m에 대한 서명자의 F-F-S 은닉전자서명 (m, r', y')을 얻을 수 있다. 네 번째 단계에서 서명 요청자가 서명을 생성하기 전에 확인하는 등식 $r^2 \Pi_{i=1}^k v_i^{e_i} \equiv y \bmod n$은 서명을 생성하고 있는 서명자가 올바른 서명을 생성하고 있는지에 대한 확인을 위한 등식이다. 만약 올바른 서명을 생성하고 있다면 수식(1)을 통해 등식은 성립한다.

은닉전자서명을 통해 생성된 메시지 m에 대한 서명은, 서명에 참여하지 않은 제 3자에 의해서도 확인이 가능하여야 한다. F-F-S 은닉전자서명을 통해 생성된 메시지 m에 대한 서명 (m, r', y')은 다음과 같은 과정을 통해 확인이 가능하다. 확인자는 메시지 m과 서명의 y'을 이용하여 $\varepsilon = h(m, y')$을 계산하고, 서명자의 공개키 v_i를 이용하여 등식 $r'^2 \Pi_{i=1}^k v_i^{\varepsilon_i} \equiv y' \bmod n$을 만족하는지 확인하면 된다. 만약, (m, r', y')이 메시지 m에 대한 올바른 F-F-S 은닉전자서명이라면 다음과 같은 수식을 통해 등식이 성립한다.

$$
\begin{aligned}
\text{좌변} : \ & r'^2 \Pi_{i=1}^k v_i^{\varepsilon_i} = r^2 t^2 \Pi_{b_i > \varepsilon_i} v_i^2 \Pi_{i=1}^k v_i^{\varepsilon_i} \qquad (2)\\
& = x^2 t^2 \Pi_{i=1}^k s_i^{2e_i} v_i^{\varepsilon_i} \Pi_{b_i > \varepsilon_i} v_i^2\\
& = x^2 t^2 \Pi_{i=1}^k v_i^{-e_i} v_i^{\varepsilon_i} \Pi_{b_i > \varepsilon_i} v_i^2\\
& \equiv yt^2 \Pi_{i=1}^k v_i^{\varepsilon_i - e_i} \Pi_{b_i > \varepsilon_i} v_i^2 \bmod n
\end{aligned}
$$

$$
\text{우변} : \ y' \equiv yt^2 \Pi_{i=1}^k v_i^{b_i} \bmod n
$$

위의 좌변과 우변이 성립하기 위해서는 모든 i $(1 \le i \le k)$에 대하여 $yt^2 \Pi_{i=1}^k v_i^{\varepsilon_i - e_i} \Pi_{b_i > \varepsilon_i} v_i^2 \equiv yt^2 \Pi_{i=1}^k v_i^{b_i} \bmod n$이 성립하면 된다. 각각의 i에 대한 좌변과 우변의 값을 표로 도시하면 표 1과 같다. 표 1에서 볼 수 있듯이 정당한 서명일 경우 모든 i에 대하여 등식이 성립하게 되며 서명에 참여하지 않은 제 3자가 서명자의 정당한 서명인지를 확인할 수 있다.

b_i	ε_i	e_i	좌변	우변
0	0	0	1	1
0	1	1	1	1
1	0	1	v_i	v_i
1	1	0	v_i	v_i

〔표 1〕 Feige-Fiat-Shamir 은닉전자서명의 유효성

2.2.2 안전성

많은 암호학적 기법에 사용되는 프로토콜은 랜덤 오라클 모델(random oracle model)을 기반으로 프로토콜의 안전성을 증명한다[14]. 랜덤 오라클 모

델 기반이란 프로토콜에 사용되어 지는 해쉬함수들이, 각각의 입력값들에 대해 랜덤하게 결과값들을 출력한다는 것을 의미하며 따라서, 이러한 해쉬함수들은 암호학적으로 안전하다는 것을 의미한다. 본 논문에서 사용되어지는 모든 해쉬함수들은 임의의 입력값에 대해 k비트 길이의 출력값을 갖는 랜덤 오라클 모델이라 가정한다. 은닉전자서명의 안전성은 사용자에 대한 은닉성(blindness) 보장과 서명에 대한 위조 불가능성(unforgeability)의 두 가지 측면으로 나누어 생각해 볼 수 있다.

사용자에 대한 은닉성 보장은 서명자가 자신이 서명할 메시지의 내용을 알 수 없다는 특성과, 서명 후에 메시지와 메시지에 대한 자신의 서명을 보았을 때, 언제 누구에게 해준 서명인지를 알 수 없어야 한다는 특성을 만족하여야 한다. F-F-S 은닉전자서명에서 서명 요청자가 서명자에게 메시지 m에 대한 서명을 요청할 때, 다른 은닉전자서명과 마찬가지로 서명 요청자는 메시지 m을 전송하지 않고, 임의의 난수 t와 은닉요소 b를 사용하여 $y' \equiv yt^2 \Pi_{i=1}^{k} v_i^{b_i} \bmod n$을 계산하고 메시지 m과 함께 해쉬함수를 이용하여 $\varepsilon = h(m, y')$를 계산한 후 이를 이용하여 e를 계산한다. 이때 사용되어지는 해쉬함수 h는 랜덤 오라클 모델이므로, 두 번째 단계에서 서명자에게 전송되어지는 인수 e를 통해서 서명요청자의 메시지 m을 계산해 내는 것은 계산적으로 불가능하다. 또한, 서명자가 서명 후에 생성되어진 서명 (m, r', y')을 사용하여 언제 누구를 위한 서명인지를 판별하기 위해서는 서명 (m, r', y')으로 부터 서명시 사용되어진 인수들 즉, y, e, r 중 하나와의 연결성을 도출할 수 있다면 가능하다. 그러나, 서명 생성시 계속적으로 변하는 임의의 난수 t와 은닉요소 b, 해쉬함수 h를 사용하여 서명을 생성하므로, 서명 (m, r', y')으로부터 y, e 또는 r과의 연결성을 찾는 것은 계산적으로 불가능하다. 그러므로, F-F-S 은닉전자서명은 사용자에 대한 은닉성을 보장한다.

위조 불가능성(unforgeability)이란 l개의 메시지 $m_1, m_2, \cdots, m_l$에 대한 은닉전자서명 (m_1, r_1', y_1'), (m_2, r_2', y_2'), $\cdots, (m_l, r_l', y_l')$을 알고 있어도 이를 이용하여 $l+1$번째의 메시지 m_{l+1}에 대한 정당한 은닉전자서명 $(m_{l+1}, r_{l+1}', y_{l+1}')$을 계산해 낼 수 없어야 한다는 것이다. 이것은 은닉전자서명을 이용한 대표적 응용분야인 전자지불 시스템에서, 사용자가 은행이 발급한 전자화폐 이외의 불법적인 위조 화폐를 만들 수 없어야 한다는 것과 같은 의미이다. 위조 불가능성에 대한 안전성은, 위조를 시도하는 공격자가 서명에 참여하고 있는 사람인지, 아니면 서명에 참여하고 있지 않은 제 3자인지에 따라 내부공격과 외부공격으로 구분할 수 있다. 우선, 위조에 대한 안전성을 설명하기 위해 외부공격에 대한 안전성을 고려해보자. 외부공격에 대한 안전성은 프로토콜에 사용되는 모든 해쉬함수들이 랜덤 오라클이라는 가정에 의해 서명자와 서명요청자 간에 전송되는 메시지를 통한 공격은 불가능하다. 그러므로, 외부공격자는 서명자에 대해 공개되어져 있는 공개키 v_i로부터 개인키 s_i를 도출하거나, 서명할 임의의 메시지 m에 대해 $r'^2 \Pi_{i=1}^{k} v_i^{\varepsilon_i} \equiv y' \bmod n$을 만족하는 인수 r', b, y'을 계산해 내는 공격 방법이 있다. 첫 번째로 공개키를 이용한 개인키 도출에 의한 공격은 F-F-S 은닉전자서명이 F-F-S서명과 마찬가지로 법 $n = pq$에 대해서 제곱근(square root), $s_i \equiv v_i^{-1/2} \bmod n$을 계산하는 것이 매우 어렵다는데에 그 이론적 기반을 두고 있다. 이러한 법 n에 대한 제곱근을 구하는 문제는 법 n을 소인수 분해하는 문제와 그 난이도가 동일한 것으로 알려져 있으므로, 서명자의 공개키로부터 개인키를 도출하여 공격하는 것은 계산적으로 불가능하다. 두 번째로, 서명할 임의의 메시지 m에 대해 $r'^2 \Pi_{i=1}^{k} v_i^{\varepsilon_i} \equiv y' \bmod n$을 만족하는 인수 r'^2, b, y'을 계산해 내는 것은 가능하나, 서명에는 r'^2이 아닌 r'을 제공하여야 하므로 이러한 공격 방법 또한 계산적으로 불가능하다. 위조를 위한 내부공격은 서명요청자가 서명자의 서명 개인키를 도출하기 위해 또는 임의의 메시지 m에 대한 서명자의 정당한 서명을 생성하기 위해 시도되어 진다. 만약 내부공격자가 서명자의 서명 개인키를 도출하기 위한 공격을 시도한다고 가정하자. 이 경우, 내부공격자도 외부공격자와 마찬가지로 공개키 v_i로부터 개인키 s_i를 직접적으로 계산할 수는 없다. 그러므로 내부공격자는 임의의 메시지 m에 대한 서명을 위해 서명자와 자신 사이에 오가는 메시지들을 통해 개인키를 도출하려 시도할 것이다. 즉, 내부공격자는 의도적으로 조작된 e를 통해 얻어지는 r을 이용하여 개인키 s_i를 도출하려 할 것이다. 만약 공격자가 프로토콜의 첫 번째 단계에서 사용되어지는 난수 x를 계산해 낼 수 있다면, 의도적으로 조작된 e를 통해 한 번의 공격마

다 서명자의 개인키의 일부분인 하나의 s_i를 계산해 낼 수도 있다. 그러나, $y \equiv x^2 \bmod n$으로부터 난수 x를 계산해 내는 것 또한 공개된 y로부터 y의 제곱근을 구하는 문제이며, 이러한 x는 매번 서명자가 서명을 할 때마다 중복을 피하여 선택하게 되므로, 위와 같은 방법을 통한 개인키의 도출은 계산적으로 불가능하다. 내부공격자에 의해 시도되어 질 수 있는 또 하나의 공격은, 공격자가 l개의 메시지 $m_1, m_2, \cdots, m_l$에 대한 은닉전자서명 (m_1, r_1', y_1'), (m_2, r_2', y_2'), $\cdots$, (m_l, r_l', y_l')과 각각의 서명에 사용되어진 인수 y, e, r을 이용하여 $l+1$번째의 메시지 m_{l+1}에 대한 은닉전자서명 $(m_{l+1}, r_{l+1}', y_{l+1}')$을 생성하려는 공격이다. 공격자가 위의 데이터들을 통해 $r'^2 \Pi_{i=1}^{k} v_i^{\varepsilon_i'} \equiv y' \bmod n$을 만족하는 $(m_{l+1}, r_{l+1}', y_{l+1}')$을 얻을 수 있다면 이러한 공격은 성공하게 된다. 이러한 공격이 성공하기 위해서는 두 번째 단계에서 계산되어지는 $\varepsilon = h(m_{l+1}, y_{l+1}')$과 $e_i = b_i \oplus \varepsilon_i$ $(1 \le i \le k)$가 $r_{l+1}'^2 \Pi_{i=1}^{k} v_i^{\varepsilon_i'} \equiv y_{l+1}' \bmod n$을 만족하여야 한다. 그러나, 프로토콜에 사용되어 지는 해쉬함수들은 랜덤 오라클이라는 가정에 의해 위의 조건을 만족하는 m_{l+1}과 y_{l+1}'을 계산하는 것은 계산적으로 불가능하므로 l개의 정당한 서명을 이용한 $l+1$번째의 위조 서명은 불가능하며 따라서, F-F-S 은닉전자서명은 위조 불가능성을 만족한다.

위에서 설명한 바와 같이 F-F-S 은닉전자서명은 은닉성 보장과 위조 불가능성을 모두 만족하게 되며, 그러므로 은닉전자서명에서 제공되어야 할 안전성을 모두 충족하게 된다.

2.2.3 효율성

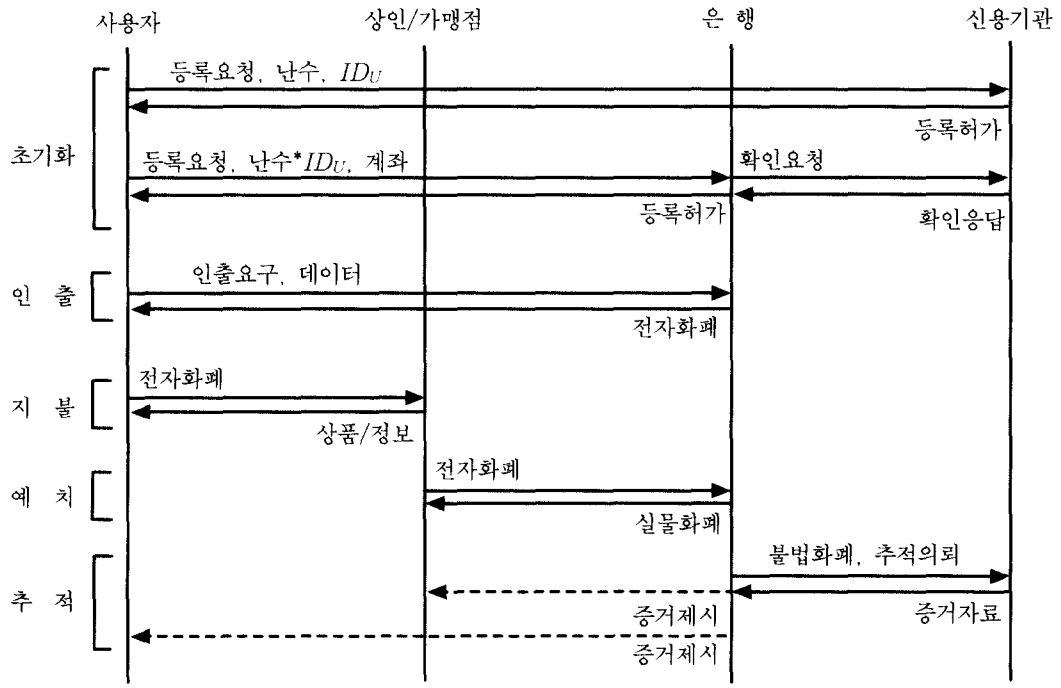
암호화 시스템에 사용되어지는 공개키 기반 프로토콜의 효율성은 계산비용, 메시지 전송수, 공개키, 개인키 등의 키의 길이 등이 고려되어져야 한다. 그러나 많은 부분들이 하드웨어의 급속한 발달로 인하여 예전에 비해 그 중요성이 떨어지게 되었다. 즉, 저장 장치의 발달 등으로 인해 공개키와 개인키 등 저장하여야 할 데이터의 길이는 그 중요성이 감소하게 되었다. 그러므로, 프로토콜의 효율성은 계산비용에 의해 크게 좌우되고, 이러한 계산비용은 법(modulus)에 대한 지수승 연산, 곱셈 연산, 역원 연산이 대부분을 차지하며, 법 계산에 비해 적은 비용을 차지하는 해쉬함수 계산이 포함된다. 법으로 사용되어지는 n의 길이는 보통 512 bit에서 1024 bit를 사용하고 있으며, 이러한 법에 대한 계산을 위해 많은 효율적인 알고리즘들이 나오고 있지만, 아직까지 많은 메모리와 연산량을 필요로 하고 있다[7~18].

〔표 2〕 은닉전자서명의 계산비용 비교

	(1)	(2)	(3)	(4)
지수승 연산	6	5	8	0
곱셈 연산	10	7	8	$1.8k$
역원 연산	4	2	0	0
해쉬함수 계산	0	0	2	1

본 장에서는 기존에 발표되었던 DSA 은닉전자서명(1), Nyberg-Rueppel 은닉전자서명(2), Okamoto-Schnorr 은닉전자서명(3)과 본 논문에서 제시한 F-F-S 은닉전자서명(4)의 효율성을, 법에 대한 지수승 연산, 곱셈 연산, 역원 연산 그리고, 해쉬함수 계산으로 나누어 비교해 보았다. 이때, 각각의 은닉전자서명의 동일한 안전성을 위해 법으로 사용되는 n과 p는 모두 1024 bit라 가정하고 Feige-Fiat-Shamir은닉전자서명에서 사용되는 보안 파라미터 $k=128$이라 가정한다. 모든 프로토콜들의 키 생성은 사전 계산이 가능하므로 효율성의 비교에서 제외시켰다. 표 2는 각각의 은닉전자서명의 생성과 확인 단계에 필요한 계산들을 법에 대한 지수승 연산, 곱셈 연산, 역원 연산 그리고, 해쉬함수 계산의 수에 대해 비교한 표이다.

2.2.1절의 프로토콜에서 볼 수 있듯이 F-F-S 은닉전자서명에서 사용되어지는 지수승 연산의 지수는 1 또는 2이므로 실질적으로 법에 대한 곱셈 연산으로 간주하여도 무방하다. 현재까지 발표되어진 법에 대한 지수승 연산의 효율적인 알고리즘에서 길이가 l인 법에 대한 지수승을 계산하기 위해서는 평균 $0.3246 \times l$번의 법에 대한 곱셈 연산을 필요로 하며 또한 많은 메모리를 요구한다[7]. 〔법 n이 1024 bit일 경우 한번의 지수승을 계산하기 위해서는 평균적으로 약 332번의 법에 대한 곱셈연산을 해야한다는 계산이 나온다. 반면, F-F-S 은닉전자서명이 기존의 은닉전자서명과 동일한 안전성을 갖기 위해 보안 파라미터 k를 128($k=128$)이라 할 때, 평균 1번의 해쉬함수와 233번의 곱셈 연산만이 필요하다. 따라서, 여러 번의 법에 대한 지수승 연산을 필요로 하는 기존의 은닉전자서명에 비해 연산비용에 있어 좋은 효율성을 가지고 있다.



〔그림 3〕 전체적인 프로토콜의 구조

III. Feige-Fiat-Shamir 은닉전자서명에 기반한 추적 가능한 전자화폐 시스템

본 장에서는 앞에서 제시한 F-F-S 은닉전자서명을 이용하여 추적 가능한 전자화폐 시스템을 설계하고, 새로운 전자화폐 시스템의 안전성 및 효율성에 대해 알아본다.

3.1 프로토콜

F-F-S 은닉전자서명을 이용한 전자화폐 시스템은 화폐의 발행과 결제를 담당하는 은행(Bank), 화폐를 발행 받아 사용하는 사용자(User), 사용자와 거래하는 상인(Merchant), 그리고 분쟁 발생 시 해결을 위한 신용기관(TTP)으로 구성된다. 전체적인 프로토콜은 각각 개체들의 초기화 프로토콜, 사 용자가 은행에서 화폐를 인출하는 인출 프로토콜, 사용자가 상인에게 화폐를 지불하는 지불 프로토콜, 상인이 은행에게 화폐를 제시하여 자신의 계좌로 해당 금액의 예치를 요구하는 예치 프로토콜, 그리고 분쟁 발생 시 신용기관이 개입하여 해당 화폐에 대한 사용자를 추적하는 추적 프로토콜로 구성된다. 전체적인 프로토콜의 구조는 그림 3과 같다.

3.1.1 표기법(Notation)

TTP : 신뢰할 수 있는 신용기관
U : 사용자(User)
B : 은행(Bank)
M : 상인(Merchant)
ID_A : A의 식별자(identity)
T_E : 화폐의 만기일(expire time)
T_S : 사용자가 화폐를 사용한 시간
p, q : TTP가 정한 두 개의 큰 소수(prime)
m_1, m_2 : 메시지 m_1과 m_2의 결합(concatenate)
$h_k(m)$: 메시지 m을 입력받아 길이가 k비트이고 각각의 원소가 0 또는 1을 출력하는 해쉬함수
$Sig_A(m)$: 메시지 m에 대한 A의 서명
$(v_1, v_2, \cdots, v_k)$: 은행의 F-F-S 은닉전자서명 공개키
$(s_1, s_2, \cdots, s_k)$: 은행의 F-F-S 은닉전자서명 개인키
$(b_1, b_2, \cdots, b_k)$: 사용자의 은닉요소(blinding factor) 모든 i에 대해 b_i는 0 또는 1

3.1.2 초기화(setup) 프로토콜

신용기관(TTP)

신용기관은 F-F-S 은닉서명을 위한 안전한 두 개

의 큰 소수 p, q를 선택하여 안전하게 보관하고 $n=pq$를 계산하여 공개한다.

은행(Bank)

은행은 신용기관이 계산한 n을 이용하여 F-F-S 은닉전자 서명의 개인키 $(s_1, s_2, \cdots, s_k)$를 생성하여 안전하게 보관하고 공개키 $(v_1, v_2, \cdots, v_k)$를 계산하여 공개한다. 이때, 공개키와 개인키는 은행이 발행할 전자화폐의 금액에 따라 각각 다르게 결정한다.

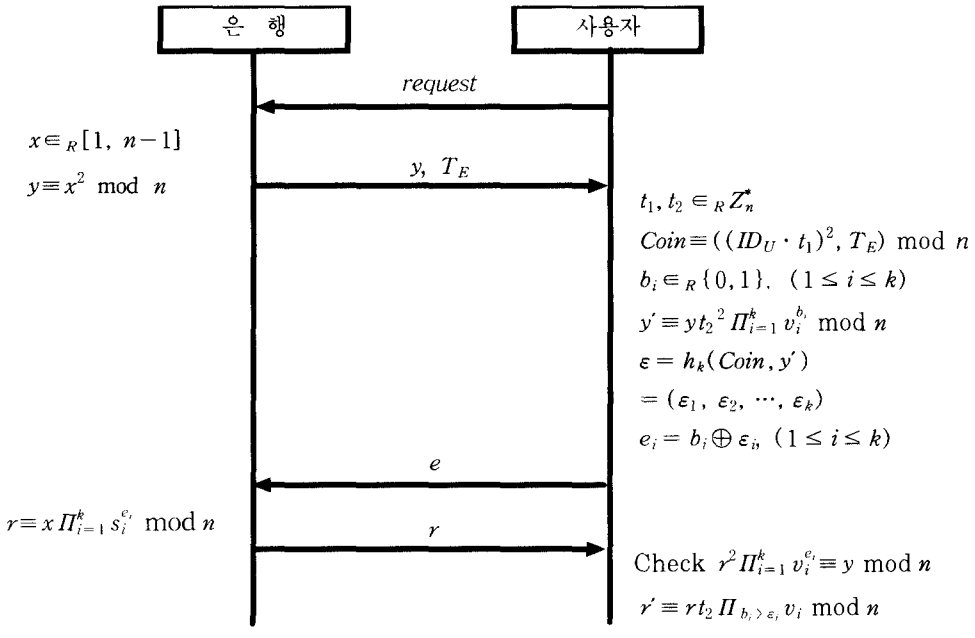
사용자(User)

사용자는 자신의 식별자 ID_U와 난수 a를 선택하여 신용기관에 등록한다. 이때 신용기관은 서로 다른 사용자에 대한 $a \cdot ID_U \bmod n$값이 같아지지 않도록 한다. 사용자는 은행에 자신의 계좌를 개설할 때 $a \cdot ID_U \bmod n$을 전달한다. 은행은 신용기관에 사용자의 $a \cdot ID_U \bmod 3n$이 올바른지를 확인하는데 이때, 신용기관은 사용자의 a와 ID_U를 은행에게 알려주지 않고 $a \cdot ID_U \bmod n$이 올바른 값 인지만을 은행에 통보한다.

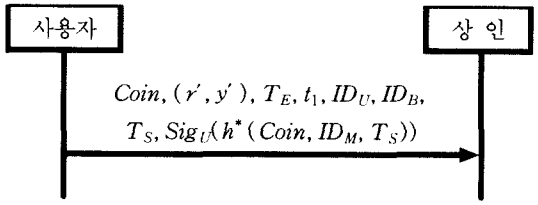
3.1.3 인출(withdrawal) 프로토콜

인출 프로토콜을 통해 사용자는 은행으로부터 전자화폐 $Coin$을 발급받게 되며, 이때 사용자의 프라이버시를 보장하기 위해 F-F-S 은닉전자서명을 사용한다. 만약 은행이 부정한 전자화폐를 발견하였을 경우, 이를 추적하기 위해 $Coin$에 추적인자 ID_U를 사용하게 된다. 만약 사용자가 올바르지 않은 ID_U를 사용하였을 경우 이는 전자화폐를 사용할 때 상인으로부터 지불 거절을 당하게 된다. 인출 프로토콜의 전체적인 구조는 그림 4와 같다.

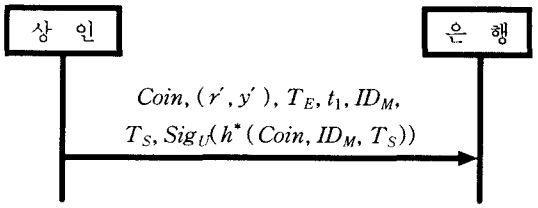
사용자는 전자화폐를 발행하는 거래은행에 전자화폐를 발급해 줄 것을 요청하는 request를 보낸다. request에는 사용자의 신원을 은행이 인증 할 수 있는 데이터와 자신의 거래 계좌번호, 발급 받고자 하는 전자화폐의 액수 등이 포함된다. 사용자의 신원에 대한 인증은 패스워드를 이용한 인증 프로토콜들을 사용할 수 있다[2]. 전자화폐의 인출을 요청받은 은행은 임의의 난수 $x \in_R [1, n-1]$를 선택하여 $y \equiv x^2 \bmod n$을 계산한 후, 계산한 y를 발행할 전자화폐의 만기일 T_E와 함께 사용자에게 전송한다. 사용자는 두 개의 난수 $t_1, t_2 \in Z_n^*$와 은닉요소(blinding factor) $b_i \in_R \{0,1\}, (1 \le i \le k)$을 선택하고 자신의 확인자 ID_U와 은행으로부터 전송 받은 y, T_E, 그리고, 자신이 발행 받고자하는 액수에 해당하는 은행의 전자화폐의 공개키 v_i 등을 이용하여



〔그림 4〕 인출 프로토콜



〔그림 5〕 지불 프로토콜



〔그림 6〕 예치 프로토콜

$Coin \equiv ((ID_U \cdot t_1)^2, T_E)$, mod n $y' \equiv y t_2{}^2 \Pi_{i=1}^{k} v_i^{b_i}$ mod n를 계산한다. 계산된 y'과 $Coin$을 이용하여 $\varepsilon = h_k(Coin, y') = (\varepsilon_1, \varepsilon_2, \cdots, \varepsilon_k)$를 계산하고, 마지막으로 계산된 ε_i와 은닉요소 b_i을 사용하여 $e=(e_1, e_2, \cdots e_k)$, $e_i = b_i \oplus \varepsilon_i$를 계산하여 e를 은행에 전송한다. 은행은 전송 받은 e와 자신의 개인키 s_i 등을 이용하여 $r \equiv x \Pi_{i=1}^{k} s_i^{e_i}$ mod n을 계산하고 r을 사용자에게 전송한다. 사용자는 전송 받은 r과 은행의 공개키 v_i 등을 이용하여 $r^2 \Pi_{i=1}^{k} v_i^{e_i} \equiv y$ mod n인지를 확인하고, 등식이 만족할 경우 $r' \equiv r t_2 \Pi_{b_i > \varepsilon_i} v_i$ mod n을 계산한다. 위와 같은 프로토콜을 통해 사용자는 $Coin \equiv ((ID_U \cdot t_1)^2, T_E)$ mod n에 대한 은행의 은닉전자서명 (r', y')을 얻을 수 있다. 인출 프로토콜이 이상 없이 진행되었을 경우 은행은 사용자의 계좌에서 발행한 전자화폐의 금액을 감한다.

3.1.4 지불(spending) 프로토콜

사용자가 상인에게 상품 또는 정보에 대한 대금으로 전자화폐를 지불하는 프로토콜이며, 전체적인 지불 프로토콜은 그림 5와 같다. 사용자는 상인이 제공하는 상품 또는 정보에 대한 대금을 지불하기 위해 그림 5와 같은 데이터를 상인에게 전송한다. 상인은 전송 받은 데이터 중 ID_U, T_E, t_1을 이용하여 $Coin \equiv ((ID_U \cdot t_1)^2, T_E)$ mod n이 성립하는지를 확인하고, ID_B에 대한 은행의 공개키를 이용하여 (r', y')이 $Coin$에 대한 정당한 F-F-S 은닉전자서명인지를 확인한다. 또한, $Sig_U(h^*(Coin, ID_M, T_S))$가 사용자의 정당한 서명인지를 확인한다. 지불 프로토콜의 데이터들에 이상이 없을 경우, 상인은 해당 전자화폐 $Coin$을 지불 대금으로 받고 상품 또는 정보를 사용자에게 제공한다.

3.1.5 예치(withdrawal) 프로토콜

예치 프로토콜은 상인이 상품 또는 정보의 대금으로 사용자로부터 받은 전자화폐를 은행에게 실물화폐로 교환하는 기능을 가진 프로토콜이며 전체적인 예치 프로토콜은 그림 6과 같다. 은행은 사용된 전자화폐를 해당 전자화폐의 만기일까지 데이터베이스에 저장함으로써 전자화폐의 이중 사용을 알아내고 추적 프로토콜을 통해 이중 사용한 사용자 또는 상인을 밝혀낼 수 있다. 상인은 자신이 상품 또는 정보의 대금으로 받은 전자화폐를 은행에게 실물화폐로 교환하기 위해 그림 6과 같은 데이터를 은행에게 전송한다. 은행은 $Coin$이 이중 사용된 전자화폐인지를 사용된 전자화폐를 저장해 놓은 데이터베이스를 통해 확인한다. 데이터베이스에서 해당 $Coin$을 발견해 내면 이중 사용된 전자화폐로 간주하고 추적 프로토콜을 진행한다. 만약 데이터베이스에 해당 $Coin$이 없을 경우 $Coin$의 뒷부분이 전자화폐의 만기일 T_E와 일치하는지, (r', y')이 $Coin$에 대한 자신의 F-F-S 은닉전자서명인지를 확인한다. 예치 프로토콜의 데이터들에 이상이 없을 경우 은행은 전자화폐의 해당 금액을 상인의 계좌에 예치한다.

3.1.6 추적(tracing) 프로토콜

추적 프로토콜은 예치 프로토콜을 통해 상인이 은행에게 전자화폐를 제시한 이후 해당 전자화폐가 이중 사용되었거나, 또는 의심스러운 전자화폐에 대해 수사기관이 합법적인 절차를 통해 사용자를 추적하고자 할 경우 은행이 예치 프로토콜을 통해 상인으로부터 전송 받은 데이터를 신용기관에게 제시함으로써 수사기관이 은행과 신용기관의 협조를 얻어 사용자를 추적할 수 있다. 이 경우, 신용기관에 의해 개인의 신상이 노출될 수도 있으므로, 추적 프로토콜을 수행하기 위해서는 수사기관이 법원의 영장 발급과 같은 법적인 절차를 거쳐 추적 프로토콜 수행의 권리를 부여받은 후 실행할 수 있다. 추적 프로토콜은 기존에 제시되었던 대부분의 추적가능 전자화폐 시스템의 추적 프로토콜과 마찬가지로 사용자가 전자화폐 인출 시 만든 사용자의 식별자에 의해 추적된다. 이중 사용은 사용자의 이중 사용, 상인의 이중 사용, 그리고 사용자와 상인의 공모에 의한 이

중 사용으로 나누어 생각할 수 있다. 이중 사용 모두의 경우 먼저 예치 프로토콜을 실행한 전자화폐에 대해서는 정상적인 예치가 진행되며, 그 이후에 같은 전자화폐에 대한 예치 프로토콜은 이중사용으로 간주되어 추적 프로토콜이 진행된다. 추적 프로토콜에서 신용기관은 n에 대한 소인수 p, q를 알고 있으므로 전자화폐 $Coin \equiv ((ID_U \cdot t_1)^2, T_E) \bmod n$에서 $ID_U \cdot t_1$을 계산해 낼 수 있고, 따라서 ID_U를 알 수 있다. ID_U를 알아낸 신용기관은 ID_U의 서명 공개키로 $Sig_U(h^*(Coin, ID_M, T_S))$가 사용자 ID_U의 $h^*(Coin, ID_M, T_S)$의 올바른 서명인지를 확인한다. 만약 사용자 ID_U의 올바른 서명일 경우, 임의의 데이터에 대한 올바른 서명은 정당한 사용자만이 생성할 수 있기 때문에 이는 사용자의 이중 사용으로 간주한다. 또한 서명이 ID_U의 올바른 서명이 아닐 경우 이는 상인의 이중 사용으로 간주하게 되는데, 이는 지불 프로토콜에서 사용자의 서명을 확인하게 되어 있기 때문이다. 만약 사용자와 상인의 공모에 의한 이중 사용일 경우에도 위의 경우 중 한 경우에 해당되며, 공모한 사용자나 상인 중 적어도 한 명 이상은 이중 사용자로 밝혀지기 때문에 공모는 일어나지 않게 된다. 이중 사용으로 인한 추적 이외에도 수사기관이 영장 발급 등과 같은 합법적인 절차를 통해 전자화폐의 사용자에 대한 추적을 의뢰할 경우, 신용기관은 위와 같은 방법을 통해 전자화폐의 사용자 ID_U를 알아 낼 수 있다.

3.2 안전성 및 효율성

전자화폐는 화폐 가치를 포함하고 있는 디지털 형태로 이루어져 있으므로 가치 정보의 조작과 위조가 용이하다. 따라서, 전자화폐의 안전성 측면에서 전자화폐에 대한 조작과 위조가 불가능하여야 한다. 본 논문에서 제안한 전자화폐의 인출 프로토콜은 II장에서 제안한 F-F-S 은닉전자서명에 기반하여 설계되었으며, II장 2.2 절에서 F-F-S 은닉전자서명의 위조 불가능성을 보였다. 따라서, 본 논문에서 제안한 전자화폐는 은행만이 발급할 수 있으며 전자화폐에 대한 조작이나 위조는 불가능하다. 또한 전자화폐는 사용자의 프라이버시를 위해 익명성을 제공하는 것이 바람직하다. 인출 프로토콜은 F-F-S 은닉전자서명을 사용하였으므로 은행은 사용자에게 자신이 발급하는 전자화폐 $Coin$의 값을 알지 못한 상태에서 발급하게 되며, 또한 예치 프로토콜에서 은행이 상인으로부터 전송 받는 데이터를 통해서도 사용자의 어떠한 정보도 알아낼 수 없으므로 사용자의 익명성을 제공한다. 또한 이중 사용과 같은 불법적인 전자화폐에 대해서는 이중 사용을 한 사용자 또는 상인을 밝혀 낼 수 있으므로 이중 사용에 대한 문제점을 해결할 수 있다. 전자화폐에서 익명성을 제공하기 위한 방법으로 가장 많이 사용되는 것은 은닉전자서명 기법이다. 본 논문에서 제안한 F-F-S 은닉전자서명은 II장 2.3절에서 분석한 것과 같이 기존의 다른 은닉전자서명에 비해 계산비용 측면에서 그 효율성이 우수함을 알 수 있으며, 따라서, 이러한 F-F-S 은닉전자서명을 기반으로 설계한 전자화폐는 기존의 소인수 분해문제[1~16]나 이산대수문제[1~16] 기반의 은닉전자서명을 이용한 전자화폐 시스템[13~17]보다 계산비용 측면에서 효율성이 좋다. 또한 전자화폐에서 완전한 익명성의 제공으로 인해 돈 세탁, 불법 구매 자금으로의 사용 가능성이 있었지만, 본 논문에서 제안한 전자화폐는 합법적인 경우 추적 프로토콜을 통해 익명성 취소(anonymity revocation) 기능을 제공하여 전자화폐가 불법적인 용도로 사용된 경우 사용자를 알아낼 수 있게 설계되었다.

Ⅳ. 결 론

전자상거래에서 지불을 위한 전자화폐 시스템이 사용자의 익명성을 제공하지 못한다면, 사용자의 프라이버시는 보호받지 못하게 되며, 이 경우 사용자의 개인 신상 누출에 의한 많은 피해가 발생할 것이다. 전자화폐 시스템에서 현재까지 사용자의 익명성을 제공하기 위한 가장 적절한 방법은 1982년 David Chaum이 제안한 은닉전자서명이다. 그러나, 초기에 제안되었던 전자화폐 시스템들은 사용자의 프라이버시를 위해 사용자의 완전한 익명성을 제공하게 설계되었고, 이로 인해 불법적인 전자화폐 사용시, 이를 사용한 사용자를 추적하여 처벌할 방법이 없었다. 이러한 이유로 최근에는 이를 제어하기 위해 익명성 취소라고 하는 익명성 제어의 기능을 가진 전자화폐 시스템의 연구가 활발히 진행 중에 있다. 즉, 전자화폐의 합법적인 사용자에 대해서는 사용자의 익명성을 제공하여 주는 반면, 불법적인 사용자에 대해서는 해당 전자화폐에 대한 추적을 통해 사용자를 알아낼 수 있도록 한다.

본 논문에서는 기존에 제시되었던 은닉전자서명보다 계산적으로 효율이 좋은 새로운 Feige-Fiat-

Shamir 은닉전자서명을 설계하였고, 이를 전자화폐 시스템에 적용시킴으로써, 전자화폐 시스템의 계산적인 효율성을 증대시켰다. 또한 전자화폐 생성시 전자화폐 내부에 추적인자 ID_U를 포함시켜서, 불법적으로 사용된 전자화폐에 대해 신용기관이 전자화폐를 통해 ID_U를 계산하여 사용자를 추적할 수 있게 하였다. 이러한 전자화폐에 대한 추적 기능의 추가로, 전자화폐에 대한 불법적인 사용과 전자화폐의 이중 사용을 사전에 억제할 수 있을 것으로 기대된다.

참 고 문 헌

[1] U.Feige, A.Fiat, and A.Shamir, "Zero-Knowledge Proofs of Identity", Journal of Cryptology, 1, pp. 77-94, 1988.

[2] 박왕석, 정종필, 박창섭, 이동훈, "패스워드를 이용한 인증 프로토콜들에 대한 고찰", 통신정보보호학회지, 제 9권 4호, pp. 51-63, 1999

[3] A.Fiat, A.Shamir, "How to Prove Yourself : practical solutions of identification and signature problems", Crypto '86 pp 186-194, 1986

[4] A.J.Menezes, P.C.van Oorschot and S.A.Vanstone "Handbook of Applied Cryptography" CRC Press, 1996

[5] A.Chan, Y.Frankel and Y.Tsiounis, "An efficient off-line electronic cash scheme as secure as RSA", Research Report NU-CCS-96-03, Northeastern University, Boston, 1995

[6] A.Lysyanskaya and Z.Ramzan, "Group Blind Digital Signatures: A Scalable Solution to Electronic Cash", Editor Proceedings of the Second International Conference on Financial Cryptography 1998

[7] C.Y.Chen, C.C.Chang and W.P.Yang, "Hybrid method for modular exponentiation with precomputation", Electron Lett., vol.32, no.6, pp. 540-541, 1996

[8] D.Chaum, "Blind signatures for untraceable payments", Advances in Cryptology -Crypto '82, Plenum Press, pp. 199-203, 1983

[9] D.Chaum, A.Fiat and M.Naor, "Untraceable Electronic cash", Advances in Cryptology - Crypto '88, Springer Verlag, pp. 319-327, 1988

[10] D.Chaum and E.Heijst, "Group signatures", Eurocrypt '91, pp. 257-265, 1991

[11] D.Chaum and H.V.Antwerpen "Undeniable signatures", Crypto '89 pp. 212-217, 1989

[12] D.Pointcheval and J.Stern, "Provably secure blind signature schemes", Asiacrypt '96 pp. 252-265, 1996

[13] K.Q.Nguyen, Y.M.Vijay, V.Varadharajan "A New Digital Cash Scheme Based on Blind Nyberg-Rueppel Digital Signature", In Information Security-Proceedings of First International Workshop, ISW'97, pp. 313-320, 1997

[14] M.Bellare and P.Rogaway, "Random Oracles are Practical a Paradigm for Designing Efficient Protocols", Proc. of the 1st CCCS ACM press pp. 62-73, 1996

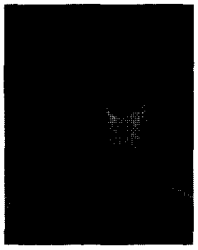
[15] N.Ferguson, "Single Term Off-Line Coins", Advances in Cryptology - Eurocrypt '93, Springer Verlag, pp. 318-328, 1993

[16] N.Koblitz "A Course in Number Theory and Cryptography", Springer-Verlag 2nd Edition, 1994

[17] S.Brands "Untraceable off-line cash in wallets with observers", In Advances in Cryptology-Crypto'93, pp. 302-318, 1993

[18] V.Dimitrov and T.Cooklev, "Two algorithms for modular exponentiation using nonstandard arithmetics", IEICE Trans. vol. E78-A, no.1, pp. 82-87, 1995

〈著者紹介〉



박 왕 석 (Wang-Suk Park) 학생회원
1998년 2월 : 단국대학교 수학과 학사
2000년 8월 : 단국대학교 전자계산학 석사
2000년 6월 : MISecurity 무선인터넷 보안연구소 주임 연구원
〈관심분야〉 전자서명, 무선보안

박 창 섭(Chang-Seop Park) 종신회원
1983년 : 연세대학교 경제학사
1983년 : 한국 IBM System Administrator
1987년 : LEHIGH Univ. 전산학 석사
1990년 : LEHIGH Univ. 전산학 박사
1990년 : 단국대학교 전자계산학과 조교수
2000년 : 단국대학교 전자.컴퓨터학부 교수
〈관심분야〉 암호이론, 네트워크 보안