논문 99-24-3A-14 한국통신학회논문지 '99-3 Vol.24 No.3A

랜덤화된 은닉서명 방식에 기반을 둔 효율적인 전자화폐 시스템

정회원 최 영 철*, 김 승 주*, 원 동 호**

Efficient Electronic Cash System Based on Randomized Blind Signature

Youngchul Choi*, Seungjoo Kim**, Dongho Won** *Regular Members*

요 약

N. Ferguson은 Eurocrypt'93에서 랜덤화된 은닉서명 방식에 기반을 둔 전자화폐 시스템을 제안하였다. Ferguson의 시스템은 최초로 challenge-and-response 기술을 사용하여 그 동안 전자화폐 시스템의 가장 큰 문제점이던 통신량 오버헤드의 문제를 해결하는 획기적인 전기를 마련하였다. 그러나, Ferguson의 시스템은 전자화폐 시스템의 가장 기본적인 요구사항인 안전성에 치명적인 문제점을 가진다. 즉, 전자화폐의 발행자인 은행이 시스템의 구성원인 상점과 공모할 경우, 정당한 사용자는 공모공격으로 인하여 불법적인 전자화폐의 이중 사용자로 몰릴 수 있다는 것이며, 이것은 근본적으로 법적 효력을 갖는 전자화폐 시스템을 구성할 수 없게 만든다. 본 논문에서는 Ferguson의 시스템이 갖고 있는 문제점을 분석하고, 이를 해결할 수 있는 새로운 방안들을 제시한다.

ABSTRACT

N.Ferguson presented an efficient electronic cash system based on a randomized blind signature at Eurocrypt'93. Ferguson's system played an important role to solve the problem that is to have much overhead of communication, which is the worst problem in electronic cash system until that time, using the challenge-and-response technique. But his system has a fatal problem that breaks the security requirement, which is the one of basic requirements in the electronic cash system. That is, if a bank, the issuer of electronic cash, colludes with a shop, the participant of electronic cash system, it could frame up an honest user as a double-spender by collusion attack. This problem make the electronic cash system to have no legal force. In this paper, we analyze the problem of Ferguson's system and propose the countermeasures against it.

I. 서 론

N. Ferguson은 RSA 서명 방식[1]을 이용한 랜덤화된 은닉서명(randomized blind signature)에 기반을 둔 효율적인 전자화폐 시스템을 Eurocrypt'93에 발표하였고[2], Crypto'93에는 관찰자(observer)와 n회 사용가능 하도록 확장한 시스템을 발표하였다.[3] Ferguson의 시스템은 기존의 전자화폐 시스템들이 사용하던 cut-and-choose 방법 대신 challenge-and-response 기술을 최초로 사용함으로써 전자화폐 시스템의 실제 구현시 가장 큰 문제점으로 등장했던 통신량 과중의 문제점을 해결한 전자화폐 시스템이다. Ferguson이 제안한 전자화폐는 두 부분으로 나뉘어지며 각각 RSA 서명으로 이루어진다. 발행 받은 전자화폐를 상점에 지불하는 경우, 사용자는 단지 상점으로부터 전송되는 challenge에 대한 모듈라

* 한국정보보호센터(ycchoi@kisa.or.kr)
** 성균관대학교
논문번호 : 98032-0119, 접수일자 : 1998년 1월 29일

곱셈의 결과와 전자화폐의 두 부분을 모듈라 곱셈한 결과를 상점에 전송하면 되므로, Ferguson 시스템의 지불 프로토콜은 매우 간단하며 효율적이다. Ferguson의 시스템은 전자화폐 시스템의 여러 가지 요구사항들 중 다음과 같은 기본적인 요구사항들을 기반으로 하고 있다.

- 안전성 (security) : 전자화폐 시스템의 모델은 일반적으로 사용자, 상점 그리고 은행으로 구성된다. 만약, 어떤 구성원들이 서로 공모하여 나머지 구성원을 위해(危害)하는 경우, 공격받는 구성원은 어떠한 공모공격(collusion attack)에도 안전해야만 한다.
- 이중사용의 방지 (prevention of double-spending) : 전자화폐는 그 자체가 하나의 가치 있는 디지털 정보이다. 디지털 정보는 종이 문서와는 달리 복사본의 생성이 쉬우며, 더 불어 그것의 원본 및 사본의 구별이 불가능하게 된다. 결국, 이중사용의 의미는 악의(惡意)의 사용자가 전자화폐를 불법 복제하여 무단으로 재사용 하는 것을 의미하는 것이며, 이것은 전자화폐 시스템에서 반드시 방지되어야 한다.
- 프라이버시 (privacy) : 전자화폐 시스템의 가장 큰 특징은 사용자 프라이버시가 보장된 다는 것이다. 즉, 전자화폐는 실제 현금과 같이 사용자의 거래 내역이 추적되지 않는다. 이러한 사용자 거래의 불추적성을 일반적으로 사용자의 프라이버시라고 일컫는다. 사용자 프라이버시의 보장은 전자화폐 시스템의 가장 큰 장점이 되며, 프라이버시 보장 강도에 따라 허위 프라이버시(fake privacy)와 프라이버시(privacy)로 나뉜다. [2]
- 오프라인 (off-line) : 전자화폐 시스템은 온라인(on-line) 방식과 오프라인(off-line) 방식으로 대별될 수 있다. 온라인 전자화폐 시스템은 사용자와 상점의 거래시 은행의 개입이 필요한 시스템으로서, 바꾸어 말하자면, 사용자가 상점에 전자화폐를 지불하는 경우 네트워크 상으로 은행의 개입이 있어야 한다는 것이다. 반대로 오프라인 전자화폐 시스템은 사용자와 상점의 거래시 은행의 개입이 필요치 않은 것이다. 일반적으로 전자화폐 시스템은 오프라인 방식을 채택하고 있는데, 이것은 물리적 화폐의 기본 성질에 따른 것이며, 더불어 컴퓨터 네트워크를 통해서 뿐만 아니라 일반 상점의 오프라인 단말기를 통해서도 거래가 가능토록 하는 장점을 주기 때문이다.

Ferguson의 전자화폐 시스템은 상기에서 설명된 요구사항을 기반으로 하기 때문에 그의 시스템은 상기의 기본적인 요구사항을 만족해야 한다. 그러나, 불행히도 Ferguson의 시스템은 상기의 기본 요구사항중 안전성 요구사항에 위배되는 커다란 문제점을 지니고 있다.[5] 전자화폐의 발행자인 은행이 상점과 공모할 경우, 은행은 정당한 사용자를 불법적인 전자화폐의 이중 사용자로 모함할 수 있다. 바꾸어 말하면, 불법적인 이중 사용자가 적발될 경우, 신뢰기관은 그 사람이 정말로 악의를 갖는 불법 이중 사용자인지, 아니면 은행과 상점에 의해 이중 사용자로 공모 당한 정당한 사용자인지 구별할 수 없다는 문제가 발생할 수 있다는 것이며,. 이것은 결국 Ferguson의 전자화폐 시스템이 법적 효력을 가질 수 없다는 것을 의미하는 것이다. 본 논문에서는 이러한 Ferguson의 시스템의 문제를 최초로 보이며, 이에 따른 해결책들을 제안한다.

Ⅱ. Ferguson 전자화폐 시스템의 문제점

1. Ferguson의 전자화폐 시스템

〔전자화폐의 구조〕

Ferguson 시스템의 전자화폐는 두 부분으로 구성되며, 첫 번째 부분은 $(C^kA)^{1/v}$이고 두 번째 부분은 $(C^uB)^{1/v}$이다. 여기서 v는 은행의 공개키이며 소수이다(이것은 전자화폐의 금액단위(denomination) 표시 역할을 한다). 그리고, k는 랜덤 수로서 사용자만이 알고 있는 전자화폐의 비밀정보이며, U는 사용자의 식별자(identity)이다. C, A, B는 공개된 안전한 일방향 함수(one-way function)와 base number c, a, b를 이용하여 각각 $C=f_c(c)$, $A=f_a(a)$, $B=f_b(b)$로 계산된다.

〔지불 프로토콜〕

그림 1에서 Alice는 상점에게 c, a, b를 전송하며, 상점은 그것을 이용하여 C, A, B를 생성한다. 그리고 나서, 상점은 자신의 0이 아닌 랜덤수 x(challenge)를 Alice에게 전송한다. Alice는 x를 수신한 후, $r=kx+u \pmod v$과 $R=(C^rA^xB)^{1/v} \pmod n$을 계산하여 상점에게 전송하며, 상점은 수신된 r, R을 이용하여 $R^v \doteq (C^rA^xB) \pmod n$임을 검증한다.

참고 : 상기 프로토콜에서 모든 지수부분 계산은 모듈라 v상에서 이루어진다. 또한, Alice는 상점측이 R^v 계산이 $(C^rA^xB) \pmod n$가 아니라 $(C^{r \pmod v}A^xB)$

*(mod n)*와 같은 결과를 갖도록 하기 위해서, R에 대한 적절한 정정인자(correction factor)를 r,R과 함께 전송해야만 한다. 이 정정인자는 r *(mod v)*에 대한 몫 값이며, 상점측에서는 이것을 R의 지수부분으로 하여, R^v를 이 결과로 나누어주면 지수부분이 *(mod v)* 상에서 동작하는 것과 같은 결과를 갖는다.

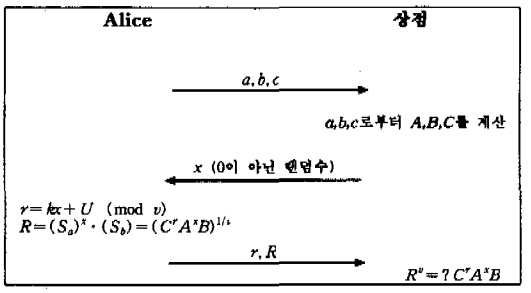

그림 1. 지불 프로토콜
Fig. 1 The Payment Protocol

〔랜덤화된 은닉서명 방식〕

랜덤화된 은닉서명 방식은 논문 [2]에서 D. Chuam과 Ferguson에 의해 제안된 것으로서, 전자화폐 시스템의 인출 프로토콜을 설계하는데 있어서 가장 중요한 부분이 된다. 랜덤화된 은닉서명 방식은 다음과 같은 시스템 파라메터를 갖는다.

- n : RSA 서명 시스템에서 사용되는 모듈라 값으로서 큰 두 소수의 곱으로 구성되며, 그 소수들은 은행만이 알고 있음
- v : RSA 서명 시스템에서 공개키로 사용되며, 적절히 큰 소수(약 128비트 이상)
- g : Z_n^*에 포함되며 소모적 공격(exhaustive attack)에 견딜 수 있는 큰 위수(order)를 갖는 공개된 수
- f : Z_n^*에서 Z_v^*로 사상(mapping)되며 공개되는 안전한 일방향 함수
- p : $p \equiv 1 \bmod n$ 인 소수
- h : 위수 n을 갖는 공개된 수로서 체(field) F_p에 포함

그림 2에서 Alice는 랜덤 수 a_1과 2개의 은닉 인자 δ와 γ를 선택한다. 그리고, $\gamma^v a_1 g^{\delta} \pmod n$를 계산하여 은행에 전송한다. 은행은 랜덤수 a_2를 선택하여 $h^{a_2} \pmod p$를 계산한 후, 그것을 Alice에게 전송하며, Alice는 그것을 이용하여 $e = f(h^{a_1 a_2}) - \delta \pmod v$를 계산한 후, 은행에 전송한다. 이어서 은행은 A값의 은닉된 형태 $\overline{A} = (\gamma^v a_1 g^{\delta}) \cdot (a_2) \cdot (g^e) \pmod n$를 계산한 후 그것을 Alice에게 전송한다. 마지막으로 Alice는 $\overline{A}$를 은닉 인자 γ로 나누어 A를 취하며, 결국 서명쌍으로서 $(a, (ag^{f(h^a)})^{1/v})$를 얻는다. 즉, 메시지 a에 대한 서명은 $(ag^{f(h^a)})^{1/v}$가 되는 것이다. 이후 이러한 서명이 제3자 검증자에게 검증 되는 경우 Alice는 검증자에게 메시지 a와 그것에 대한 서명 $S_A = (ag^{f(h^a)})^{1/v}$를 전송하게 되며, 검증자는 그것들을 수신한 후, 공개정보 $g, f(\cdot), h$를 이용하여 $S_A{}^v = ag^{f(h^a)}$임을 확인함으로써 Alice의 서명을 검증할 수 있다.

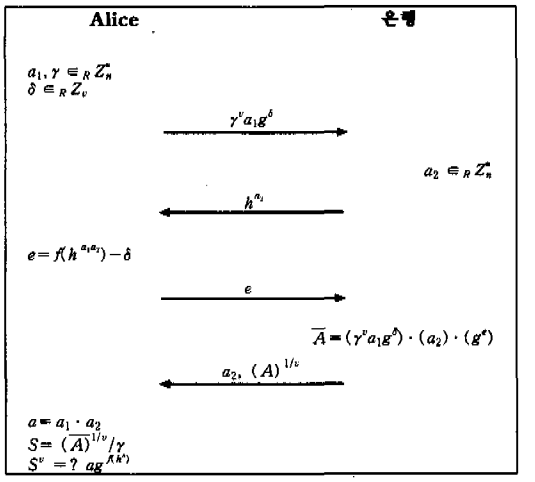

그림 2. 랜덤화된 은닉서명 프로토콜
Fig. 2 The Randomized Blind Signature Protocol

상기 프로토콜 수행과정에서 주의할 것은 완전한 은닉형태를 만들기 위해서 지수를 포함한 모든 계산은 모듈라 v 상에서 이루어져야만 한다는 것이다. 예를 들면 e는 $e = f(h^{a_1 a_2}) - \delta \pmod v$로 계산된다. 이러한 연산은 검증자측에서 Alice의 메시지 a를 가지고 검증하는경우 잘못된 연산결과를 도출시키기 때문에 Alice는 은행으로부터 서명값 $(ag^{f(h^a)})^{1/v}$를 받은 후에 반드시 정정절차를 거쳐야 한다. 이것은 Alice가 은행으로부터 받은 최종 서명값 $(ag^{f(h^a)})^{1/v}$에 정정인자 $g^{(f(a)-\delta) \, div \, v}$를 곱함으로써 해결될 수 있다. 이후 이 논문의 나머지 부분에서 나오는 모든 서명은 이러한 규칙을 묵시적으로 따른다고 가정한다.

상기의 랜덤화된 은닉서명 스킴은 RSA 서명 시스템에 그 기반을 두고 있으며, a를 구성함에 있어서 Alice는 a_1을 은행은 a_2를 제공하므로 a의 결과는 항상 랜덤하게 구성된다. 그러므로, Alice는 자

신만이 원하는 형태에 대한 서명을 얻을 수 없으며, 결국 랜덤화된 은닉서명은 아래의 요구조건 3가지를 만족하게 된다.

- 요구사항 1 : Alice는 특별한 형태의 수에 RSA 서명을 받는다. (Alice 독자적으로는 그 수를 만들 수 없다.)
- 요구사항 2 : 은행은 서명되어지는 수가 랜덤하게 선택되어졌음을 확신할 수 있다.
- 요구사항 3 : 은행은 Alice가 받은 서명에 관련된 어떠한 정보도 가지지 못한다.

〔인출 프로토콜〕

Ferguson 시스템의 인출 프로토콜은 상기에서 설명한 랜덤화된 은닉서명에 기반을 두어 구성된다. 그림 3에서 Alice는 은행과 공동으로 자신에게만 알려질 수 있는 방식으로 base number c, a, b를 생성한다. 또한, Alice는 각 전자화폐가 가지고 있는 고유한 랜덤 수 k를 은행과 공동으로 생성하는데, 이것은 Alice만이 알고 있는 비밀정보가 되며, 지불 프로토콜에서 response r을 계산하는데 사용된다.

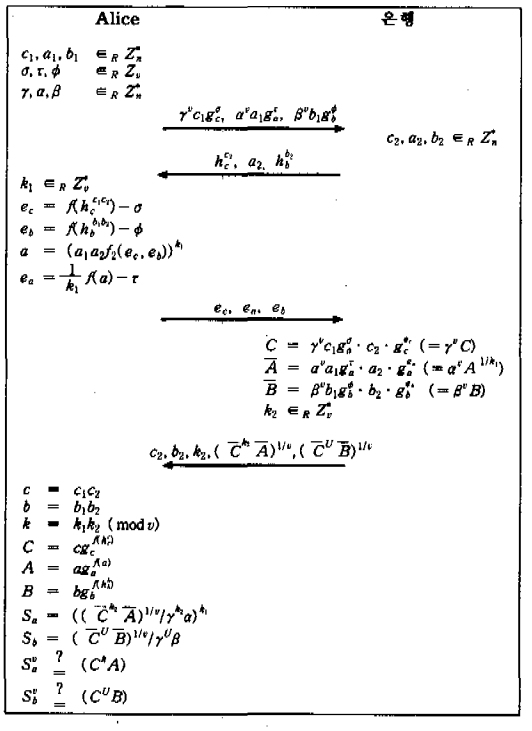

그림 3. 인출 프로토콜
Fig. 3 The Withdrawal Protocol

단계 1) Alice는 랜덤 수 $c_1, a_1, b_1, \sigma, \tau, \phi, \alpha, \beta, \gamma$를 선택한다. 여기서 c_1, a_1, b_1은 베이스 수를 구성하기 위한 Alice의 제공 인자들이고, σ, τ, ϕ는 지수 부분을 위한 은닉 인자들이며, 나머지 α, β, γ 곱셈 부분을 위한 은닉 인자들이다. Alice는 선택한 수들을 이용하여, $\gamma^v c_1 g_c^\sigma \pmod n$, $\alpha^v a_1 g_a^\tau \pmod n$, $\beta^v b_1 g_b^\phi \pmod n$를 계산한 후, 그 결과들을 은행에게 전송한다.

단계 2) 은행은 베이스 수를 랜덤하게 하기 위한 랜덤 수 c_2, a_2, b_2를 선택한 후, Alice가 직접적인 c_2, b_2 값들을 볼 수 없도록 하기 위해서 $h_c^{c_2} \pmod p$, a_2, $h_b^{b_2} \pmod p$와 같이 계산하여 Alice에게 전송한다.

단계 3) Alice는 전자화폐의 고유한 비밀정보 k를 구성하기 위한 랜덤 수 k_1을 선택한다. 그리고, 은행으로부터 수신된 수들과 선택한 수를 이용하여 $e_c = f(h_c^{c_1 c_2}) - \sigma \pmod v$, $e_b = f(h_b^{b_1 b_2}) - \phi \pmod v$를 계산한다. 또한, e_a는 k를 지수로 가져야 하기 때문에 $a = (a_1 a_2 f_2(e_c, e_b))^{k_1} \pmod n$로 두어 $e_a = (1/k_1) f(a) - \tau \pmod v$를 계산한다. Alice는 계산된 인자들 e_c, e_a, e_b를 은행에 전송한다. 여기서, f_2는 랜덤화된 은닉 서명의 시스템 파라메터 f와 같은 일방향 함수이며, 구별을 위해 다른 표기를 사용하였다.

단계 4) 은행은 C, A, B의 은닉된 값들을 $\overline{C} = \gamma^v c_1 g_c^\sigma \cdot c_2 \cdot g_c^{e_c} \pmod n$와 같이 계산하며, 이것은 $\gamma^v c g_c^{f(k)}$ (단, $c = c_1 c_2$)과 같다. $\overline{A}$, $\overline{B}$도 역시 유사한 방식으로서 계산되며, 결국 $\overline{C} = \gamma^v C \pmod n$, $\overline{A} = \alpha^v A^{1/k_1} \pmod n$, $\overline{B} = \beta^v B \pmod n$와 같은 결과가 생성된다. 그리고, 은행은 k를 랜덤하게 하기 위해 랜덤 수 k_2를 선택하고, 마지막으로 c_2, b_2, k_2, $(\overline{C}^{k_2} \cdot \overline{A})^{1/v} \pmod n$, $(\overline{C}^U \cdot \overline{B})^{1/v} \pmod n$를 계산하여 Alice 에게 전송한다.

단계 5) Alice는 c_2, b_2를 이용하여 $c = c_1 c_2 \pmod n$, $b = b_1 b_2 \pmod n$를 계산한 후, $C = c g_c^{f(k)} \pmod n$, $A = a g_a^{f(a)} \pmod n$, $B = b g_b^{f(k)} \pmod n$를 구한다. 그리고, 전자화폐를 구성하는 두 부분에 대한 은행의 서명 $S_a = ((\overline{C}^{k_2} \cdot \overline{A})^{1/v} / \gamma^{k_2} \alpha)^{k_1} \pmod n$, $S_b = (\overline{C}^U \cdot \overline{B})^{1/v} / \gamma^U \beta \pmod n$를 계산한다. 마지막으로 Alice는 계산된서명값의 정확성을 확인하기 위해서 $S_a^v \stackrel{?}{=} C^k A \pmod n$, $S_b^v \stackrel{?}{=} C^U B \pmod n$를 계산하여 비교한다.

참고 : 상기 프로토콜 전제에서 베이스 수들은 모듈라 n 상에서 계산이 이루어지며, 지수부분은 모두 모듈라 v 상에서 계산이 이루어진다. 또한, 공개 인자 g 값들을 g_a, g_c, g_b와 같이 서로 다른 인자로 사용하는 이유는 C, A, B를 계산하는 경우, 지수 값들이 서로 혼합되지 않고 구별되기 위함이다.

〔예치 프로토콜〕

상점은 사용자로부터 받은 전자화폐 $(C^rA^xB)^{1/v}$와 그것에 따르는 거래 사본 (c,a,b,r,R,x)를 은행에 예치한다. 이 때 은행은 c,a,b로부터 C,A,B를 계산하여 그 결과를 데이터베이스에 기 저장되어 있는 다른 C,A,B들과 비교를 한다. 만약 계산된 C,A,B가 이미 데이터베이스에 저장되어 있다면, 이것은 부정 사용자의 이중 사용을 의미하는 것이며, 은행은 상기에서 설명된 이중 사용자 검출 방법을 통해 이중 사용자를 색출한다. 만약 계산된 C,A,B가 데이터베이스에 저장되어 있지 않다면, 은행은 상점으로부터 받은 파라메터들을 데이터베이스에 저장한 후, 예치된 전자화폐에 해당하는 금액만큼 출금시켜 주거나 상점의 해당 계좌로 예치시켜 준다.

2. Ferguson 전자화폐 시스템의 문제점

Ferguson 전자화폐 시스템의 문제점은 앞서 언급한 바와 같이 은행과 상점이 공모할 경우 정당한 사용자를 이중 사용자로 모함할 수 있게 된다는 것이다. 이러한 문제점이 나타나는 이유는 지불 프로토콜에서 single-term을 이용함과 동시에 사용자의 식별자를 직접적으로 사용하였기 때문이다. 이러한 공모공격에 대한 구체적인 내용을 보이기 전에, 이해를 돕기 위해서 부정한 이중 사용자의 식별자가 어떻게 도출되는지를 먼저 고찰하겠다.

〔이중 사용자의 식별자 추출〕

만약 어떤 사용자가 자신의 전자화폐를 복제하여 그것을 불법적으로 이중 사용한다고 가정하자. 이중 사용자는 이미 자신의 전자화폐를 한 번 사용하였기 때문에 상점이 그 전자화폐를 은행에 예치했다면, 은행의 데이터베이스 내에는 이미 사용된 전자화폐의 base number (c,a,b)가 존재하게 된다. 그러므로, 이중 사용자가 base number로서 (c,a,b)를 갖는 전자화폐를 이중 사용하게 되는 경우, 상점이 그것을 은행에 예치하는 시점에서 은행은 자신의 데이터베이스를 검색함으로써 그 전자화폐가 이중 사용된 것임을 쉽게 알 수 있다. 이렇게 해서 어떤 전자화폐가 이중사용 된 것임이 밝혀지게 되면, 은행은 공개 정보 (r,r',x,x')로부터 아래와 같은 방식으로 이중 사용자의 식별자를 도출시킬 수 있다. 이 방식은 Shamir가 논문 [4]에서 제안한 비밀 공유 방식(secret sharing scheme)에 기반을 둔 것이다.

$$r = kx + U,\quad r' = kx' + U \tag{1}$$

$$k = (r - r')/(x - x') \tag{2}$$

상기 두 식 (1)과 (2)로부터 U는 쉽게 도출될 것이며, 은행은 도출된 U를 이용하여 자신의 데이터베이스를 검색하여 이중 사용자의 신원을 밝힐 수 있게 된다.

〔Ferguson 시스템의 문제점〕

그러나, 불행히도 Ferguson의 시스템은 상기에서 설명한 이중 사용자의 식별자 도출방법이 역기능으로 작용하여 전자화폐의 고유 비밀정보인 k를 유출 가능케 한다는 문제가 발생하게 된다. 즉, 만약 은행이 어떤 정당한 사용자로부터 사용된 전자화폐의 고유 비밀정보인 k값을 알게 된다면, 은행은 자신의 서명 시스템을 이용하여 이미 사용된 전자화폐와 동일한 형태의 전자화폐를 생성할 수 있게 되며, 이것은 결국 은행이 상점과 공모만 하게되면 정당한 사용자를 쉽게 불법적인 이중 사용자로 만들 수 있게 되는 것이다. 이러한 시나리오는 Ferguson 전자화폐 시스템의 기본적인 요구사항과 가정하에 충분히 설정될 수 있으며, 이것은 결국 자신이 설정한 요구사항과 가정에 모순되는 것을 의미하는 것이다. 시나리오는 다음과 같다.

단계 1) 상점은 자신의 주거래 고객인 Alice와 거래를 시작한다. 즉, 상점은 Alice가 주거래 고객이기 때문에 Alice에 대한 중요한 신상정보(예를 들면, 이름, 주소, 주민등록번 호 등)를 알고 있다고 가정한다.

단계 2) 상점은 거래 후 Alice가 지불한 전자화폐를 은행에 예치하는 과정에서, 고객의 중요 한 신상정보를 은행에게 알려준다. 은행은 상점이 제공한 고객의 신상정보를 이용 하여 해당 고객에 대한 식별자 U를 검출한다.

단계 3) 은행은 검출된 Alice의 식별자 U와 상점이 제공한 공개 정보 (r,x)로부터, 식 $r=kx+U$를 이용하여 k를 도출한다. 결론적으로 은행은 Alice가 사용한 전자화폐의 비밀정보 k를 알게 되고, 더불어 사용된 전자화폐의 base number (c,a,b)를 알고 있기 때문에 사용된 전자화폐와 동일한 전자화폐를 생성할 수가 있게 된다.

단계 4) 만약 은행이 상기에서 위조한 전자화폐

를 다른 상점과 공모하여, 마치 Alice가 사용 한 것처럼 시뮬레이션 하게 되면, 결국 정당한 고객 Alice는 부정한 이중 사용자로 몰릴 수 있게 되며, 동시에 신뢰기관은 이중사용이 Alice에 의해 일어난 것인지 은행 과 상점에 의한 조작인지를 구별할 수 가 없게 된다.

상기에서 살펴본 시나리오는 현실적으로 충분히 발생할 수 있는 것이며, 이것은 Ferguson 전자화폐 시스템의 기본적인 요구사항들중 안전성 조건에 위배되는 것이다. 더욱이, 시나리오의 마지막 항에서 언급한 것처럼 신뢰기관은 불법적인 이중사용이 은행에 의한 공모인지 불법적인 이중 사용자에 의한 것인지를 구별할 수 없기 때문에, Ferguson의 전자화폐 시스템은 잠재적으로 법적효력을 갖지 못한다는 심각한 문제점을 가지게 되는 것이다.

Ⅲ. 디지털 서명을 이용한 해결방안

본 절에서는 II 절에서 설명된 Ferguson 전자화폐 시스템의 문제점을 해결할 수 있는 첫 번째 방안을 제안한다.[5] 제안하는 방안의 기본적인 개념은 전자화폐에 k이외의 다른 비밀정보를 삽입하여 사용자가 전자화폐를 사용하는 경우, 그 비밀정보를 이용하여 지불 프로토콜에서 주고받은 정보들에 대한 디지털 서명을 만들어 상점에 건네줌으로써 은행의 부정을 방지한다는 것이다. 이 경우 은행과 상점이 공모할지라도 사용자의 디지털 서명을 위조할 수 없기 때문에 거래 정보를 위조할 수 없게 된다.

〔전자화폐의 구조〕

제안하는 해결방안의 전자화폐 구조는 II 절에서 설명된 Ferguson의 전자화폐 구조와 같다. 단지 전자화폐에 디지털 서명을 검증할 수 있는 정보를 넣기 위해 base number의 설계를 다르게 하였으며, 그 구조는 아래와 같다.

- $C=f_c(c)(=cg_c^{f(k)}(\bmod n))$ (단, $c=g^{c_1+c_2}$ $(c_1, c_2 \in_R Z_v^*)$)
- $A=f_a(a)(=ag_a^{f(a)}(\bmod n))$ (단, $a=g^{a_1+a_2}$ $(a_1, a_2 \in_R Z_v^*)$)
- $B=f_b(b)(=bg_b^{f(b)}(\bmod n))$ (단, $b=g^{b_1+b_2}$ $(b_1, b_2 \in_R Z_v^*)$)

상기에서 사용되는 시스템 파라메터들은 Ferguson의 시스템과 동일하다. 제안하는 방법에서는 base number (c,a,b)를 g를 이용하여 설계하고, 여기에서 사용된 지수들$(c_1,c_2,a_1,a_2,b_1,b_2)$은 지불 프로토콜에서 사용할 디지털 서명의 비밀키로 사용되며, 상점에서는 이러한 서명의 검증을 위해서 base number들의 곱, $c \cdot a \cdot b = g^{c_1+c_2+a_1+a_2+b_1+b_2} mod\ n$을 공개키로 사용한다.

〔지불 프로토콜〕

전자화폐의 구조에서 살펴본 바와 같이 제안하는 해결방안은 지불 프로토콜에서 전자화폐에 삽입된 비밀정보를 이용하여 사용자가 거래 정보에 대해 디지털 서명을 수행하는 것이다. 이 경우 비밀키는 사용자만이 알고 있는 $c_1+c_2+a_1+a_2+b_1+b_2$ $(mod\ v)$가 되며 공개키는 상점이 수신된 base number들을 곱함으로서 쉽게 얻을 수 있는 $g^{c_1+c_2+a_1+a_2+b_1+b_2}$ $(mod\ n)$ 이다. 이 경우 한 가지 주의할 점이 있다. 우리는 이미 Ferguson 전자화폐 시스템의 지불 프로토콜에서 서명의 완전한 검증을 위해 response r을 $(mod\ v)$ 상에서 계산하고 그 결과를 전송하는 경우, r에 대한 몫을 함께 전송한다는 것을 보았다. 마찬가지로 상기에서 설명한 비밀키와 공개키도 그러한 부수적인 추가조치를 취하는 것으로 가정한다. 즉, 검증자가 검증을 위해 c,a,b를 각각 곱함으로써 공개키를 생성하는 경우 $c_1+c_2+a_1+a_2+b_1+b_2$ $(mod\ v)$의 결과를 g의 지수로 갖는 것과 다르게 되기 때문에, 서명자는 항상 $c_1+c_2+a_1+a_2+b_1+b_2$ $(mod\ v)$에 대한 몫을 서명문과 함께 전송해주는 것으로 묵시적 가정을 한다.

사용되는 디지털 서명 시스템은 경우에 따라서 여러 가지를 사용할 수 있으나, 본 논문에서는 Nyberg-Ruppel 방식의 서명 시스템[6]을 사용하는 것으로 가정하며, 자세한 스킴의 소개대신 *SIGN*과 *VERIFY*라는 기호로 표시하겠다. 수행과정은 다음과 같다.

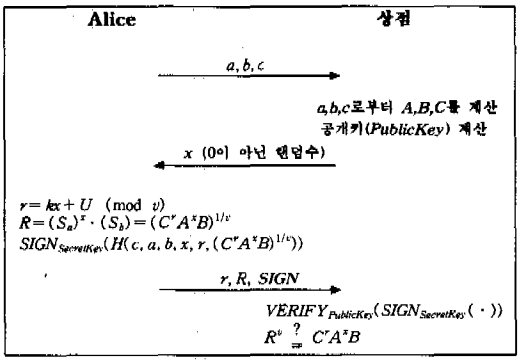

그림 4. 지불 프로토콜
Fig. 4 The Payment Protocol

그림 4와 같이 Alice는 상점에게 c, a, b를 전송하며, 상점은 그것을 이용하여 C, A, B를 생성함과 동시에 서명의 검증을 위한 공개키를 베이스 수를

곱함으로써 만든 후, 자신의 x를 Alice에게 전송한다. Alice는 x를 수신한 후, $r=kx+U \pmod v$과 $R=(C^rA^xB)^{1/v} \pmod n$을 계산하고, 더불어 자신만이 알고 있는 비밀키를 이용하여 거래 정보에 대한 서명, $SIGN_{SecretKey}(H(c,a,b,x,r,(C^rA^xB)^{1/v}))$을 생성하여 그것들을 상점에 전송한다(여기서 H는 안전한 해쉬(hash) 함수이다). 상점은 수신된 r, R을 이용하여 검증식 $R^v \stackrel{?}{=} (C^rA^xB)$을 계산하여 비교한다. 그리고 나서, 마지막으로 거래정보에 대한 검증을 위하여 앞서 계산한 공개키를 이용하여 $VERIFY_{PublicKey}(SIGN_{SecretKey}(H(c,a,b,x,r,(C^rA^xB)^{1/v})))$를 수행한다.

〔인출 프로토콜〕

제안하는 해결 방안의 인출 프로토콜은 Ferguson 시스템의 인출 프로토콜과 거의 유사하며, 다른 점은 디지털 서명에 사용될 비밀키 및 공개키 쌍을 생성하기 위해 단지 베이스 수만 c_1,a_1,b_1 대신 g^{c_1}, g^{a_1}, g^{b_1} 로 사용한다는 것이다. 물론 은행이 생성하는 c_2,a_2,b_2 도 마찬가지이며 프로토콜의 세부적인 단계별 내용은 그림 5와 같다.

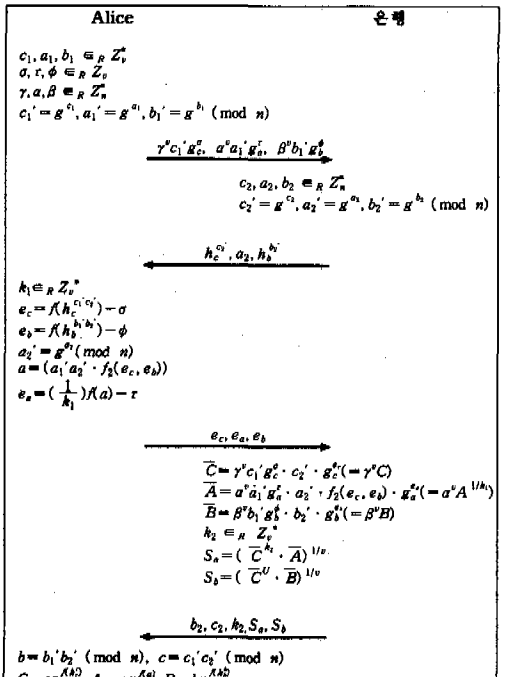

그림 5. 인출 프로토콜
Fig. 5 The Withdrawal Protocol

단계 1) Alice는 랜덤 수 $c_1, a_1, b_1, \sigma, \tau, \phi, \alpha, \beta, \gamma$를 선택한다. 여기서, c_1, a_1, b_1은 베이스 수의 지수로 사용되는 Alice의 제공 인자들이고, 이것을 이용하여 베이스 수 $c_1' = g^{c_1}$, $a_1' = g^{a_1}$, $b_1' = g^{b_1} \pmod n$를 계산한다. σ, τ, ϕ는 지수 부분을 위한 은닉 인자들이며, 나머지 α, β, γ는 곱셈 부분을 위한 은닉 인자들이다. Alice는 선택한 수와 계산된 결과들을 이용하여, $\gamma^v c_1' g_c^\sigma \pmod n$, $\alpha^v a_1' g_a^\tau \pmod n$, $\beta^v b_1' g_b^\phi \pmod n$를 계산한 후, 그 결과들을 은행에게 전송한다.

단계 2) 은행은 베이스 수의 지수 인자들 c_2, a_2, b_2를 선택한 후, 베이스 수 $c_2' = g^{c_2}$, $a_2' = g^{a_2}$, $b_2' = g^{b_2} \pmod n$를 계산한다. 그리고 나서, Alice가 직접적인 c_2', b_2'의 값들을 볼 수 없도록 하기 위해서 $h_c^{c_2'} \pmod p$, a_2', $h_b^{b_2'} \pmod p$를 계산한 후, 그것들을 Alice에게 전송한다.

단계 3) Alice는 전자화폐의 고유한 비밀정보 k를 구성하기 위한 랜덤 수 k_1을 선택한다. 그리고, 은행으로부터 수신된 수들과 선택한 수를 이용하여 $e_c = f(h_c^{c_1' c_2'}) - \sigma \pmod v$, $e_b = f(h_b^{b_1' b_2'}) - \phi \pmod v$를 계산한다. 또한, e_a는 k를 지수로 가져야 하기 때문에 $a = (a_1' a_2' f_2(e_c, e_b))^{k_1} \pmod n$와 같이 계산된 a를 이용하여 $e_a = (1/k_1) f(a) - \tau \pmod v$를 계산한다. Alice는 계산된 인자들 e_c, e_a, e_b를 은행에 전송한다. 여기서, f_2는 랜덤화된 은닉 서명의 시스템 파라메터 f와 같은 일방향 함수이며, 구별을 위해 다른 표기를 사용하였다.

단계 4) 은행은 C, A, B의 은닉된 값들을 $\overline{C} = \gamma^v c_1' g_c^\sigma \cdot c_2' \cdot g_c^{e_c} \pmod n$와 같이 계산하며, 이것은 $\gamma^v c g_c^{f(h_2')}$ (단, $c = c_1' c_2'$)과 같다. $\overline{A}$, $\overline{B}$도 역시 유사한 방식으로서 계산되며, 결국 $\overline{C} = \gamma^v C \pmod n$, $\overline{A} = \alpha^v A^{1/k_1} \pmod n$, $\overline{B} = \beta^v B \pmod n$와 같은 결과가 생성된다. 그리고, 은행은 k를 랜덤하게 하기 위해 랜덤 수 k_2를 선택하고, 마지막으로 c_2, b_2, k_2, $(\overline{C}^{k_2} \cdot \overline{A})^{1/v} \pmod n$, $(\overline{C}^U \cdot \overline{B})^{1/v} \pmod n$를 계산하여 Alice에게 전송한다.

단계 5) Alice는 c_2, b_2를 이용하여 $b_1' = g^{b_1} \pmod n$, $c_1' = g^{c_1} \pmod n$를 계산한 후 $c = c_1' c_2' \pmod n$, $b = b_1' b_2' \pmod n$를 계산한다. 그리고 나서, $C = c g_c^{f(h_2')} \pmod n$, $A = a g_a^{f(a)} \pmod n$, $B = b g_b^{f(h_2')} \pmod n$ 를 구한 후, 전자화폐를 구성하

는 두 부분에 대한 은행의 서명, $S_a = ((\overline{C}^{h_2} \cdot \overline{A})^{1/v} / \gamma^{h_3} a)^{h_1} \pmod n$, $S_b = (\overline{C}^{U} \cdot \overline{B})^{1/v} / \gamma^{U} \beta \pmod n$를 계산한다. 마지막으로 Alice는 계산된 서명값의 정확성을 확인하기 위해서 $S_a^v \stackrel{?}{=} C^t A \pmod n$, $S_b^v \stackrel{?}{=} C^U B \pmod n$를 계산하여 비교한 후, 그 결과들을 전자화폐로써 취한다.

참고 : 상기 프로토콜 전체에서 베이스 수들은 모듈라 n 상에서 계산이 이루어지며, 지수부분은 모두 모듈라 v 상에서 계산이 이루어진다. 또한, 공개 인자 g 값들을 g_a, g_c, g_b와 같이 서로 다른 인자로 사용하는 이유는 C, A, B를 계산하는 경우, 지수 값들이 서로 혼합되지 않고 구별되기 위함이다. 반대로, 베이스 수를 구성하기 위한 g 값을 g_a, g_c, g_b와 같이 다르게 사용하지 않고 일반적인 g로 통일시킨 이유는 비밀키와 공개키의 관계를 유지시키기 위함이다.

〔예치 프로토콜〕

II. 1절에서 설명된 Ferguson 시스템의 예치 프로토콜과 같다.

Ⅳ. Brands 전자화폐 시스템의 식별자 표현을 이용한 해결방안

본 절에서는 III 절에서 제안된 해결방안보다 효율적인 방식의 해결방안을 제안하고자 한다. 디지털 서명을 이용하는 해결방안의 장점은 매우 간단하다는 것으로서, Ferguson 전자화폐 시스템에 어떤 큰 변화를 주지 않아도 시스템의 문제점을 해결할 수 있게 된다. 그러나, 첫 번째 해결방안은 지불 프로토콜에서 사용자가 전자화폐를 제출함과 동시에 디지털 서명 생성과정을 수행해야 하기 때문에, 지불 단계시 계산량과 통신량이 늘어나며, 더불어 키관리의 어려움이 발생하게 된다. 두 번째 제안하는 해결방안은 이러한 첫 번째 해결방안의 효율성을 개선한 것으로서 Ferguson의 시스템을 다소 수정한 방식을 취한다.

Ferguson 시스템이 갖는 문제점의 발생원인은 앞서 언급한 바와 같이 지불단계시 response로서 single-term r만을 사용하고, 더불어 식별자 U를 직접적으로 사용하기 때문이다. 즉, Ferguson 시스템의 문제점을 근본적으로 해결하기 위해서는 이러한 발생 원인을 배제시켜야 한다. 제안하는 해결방안에서는 하나의 response 대신 두 개의 response를 두었으며, 더불어 Ferguson이 사용한 직접적인 식별자 U 대신, Brands가 제안한 전자화폐 시스템[7]에서 사용했던 방식의 식별자 표현방법을 사용하였다.

〔시스템 파라메터 설정〕

II 절에서 설명한 Ferguson 시스템의 파라메터와 같다.

〔전자화폐의 구조〕

제안하는 시스템의 전자화폐 구조는 Ferguson의 시스템에 기반을 두지만 표현방법은 다소 다르다. Base number는 Ferguson 시스템과 마찬가지로 (c,a,b)를 사용하지만, 원래 시스템이 가지고 있던 전자화폐의 비밀정보 k는 사용하지 않는다. 그리고, 사용자의 식별자 U를 직접적으로 사용하는 것 대신에 U는 $g_d^{u_1} g_e^{u_2}$로 나타내며, 지불 프로토콜에서 response를 구성하는 경우에는 사용자만이 알고 있는 (u_1,u_2)를 사용한다. 그렇게 함으로써 은행과 상점이 공모를 하더라도, 은행은 U의 값은 알 수 있지만 그것을 구성하고 있는 지수 (u_1,u_2)를 모르기 때문에 사용자의 response를 위조할 수 없게 된다. 제안하는 방법의 전자화폐 구조는 아래와 같다.

- $coin_1 = (CAg_d^d g_e^e)^{1/v} \pmod n$ (단, $C = cg_c^{f(k_2)} \pmod n$, $A = ag_a^{f(a)} \pmod n$)
- $coin_2 = (CBg_d^{u_1} g_e^{u_2})^{1/v} \pmod n$ (단, $B = bg_b^{f(k_3)} \pmod n$, $U = g_d^{u_1} g_e^{u_2} \pmod n$) $\pmod n$)

Ferguson 시스템의 전자화폐 구조와 비교해 보면, 원래의 시스템에서는 존재하던 전자화폐의 고유 비밀정보 k는 사용하지 않는 대신, 기존에는 없던 $g_d^d g_e^e$과 $g_d^{u_1} g_e^{u_2}$을 전자화폐의 두 구성요소에 각각 추가시켰다. 여기서, d와 e는 (u_1,u_2)을 이용한 response 들을 생성하기 위해 필요한 인자들이다.

〔지불 프로토콜〕

그림 6에서 Alice는 상점에게 base number c, a, b를 전송한다. 상점은 수신된 base number들을 이용하여 C, A, B를 계산한 후, x를 Alice에게 전송한다. Alice는 x를 수신한 후, 이것을 이용하여 response $r_1 = d + u_1 x \pmod v$, $r_2 = e + u_2 x \pmod v$ 그리고 $R = (C^{x+1} AB^x g_d^{d+u_1 x} g_e^{e+u_2 x})^{1/v} \pmod n$을 각각 계산하고 그것들을 상점에게 전송한다. 마지막으로 상점은 수신된 response들을 이용하여 전자화폐의 정확성을 검증한다.

참고 : 상기 프로토콜에서 모든 지수부분 계산은 모듈라 v상에서 이루어진다. 또한, Alice는 상점측이 R^v 계산이 $(C^{x+1} AB^x g_d^{d+u_1 x} g_e^{e+u_2 x})^{1/v}$가 아니라

$(C^{x+1}AB^x g_d^{d+u_1x(\bmod v)} g_e^{e+u_2x(\bmod v)})^{1/v}$와 같은 결과를 갖도록 하기 위해서, R에 대한 적절한 정정인자(correction factor)를 r,R과 함께 전송해야만 한다. 이 정정인자는 $r_1 \pmod v$과 $r_2 \pmod v$에 대한 몫 값이며, 상점측에서는 이것을 R의 지수부분으로 하여, R^v를 이 결과들로 나누어주면 된다.

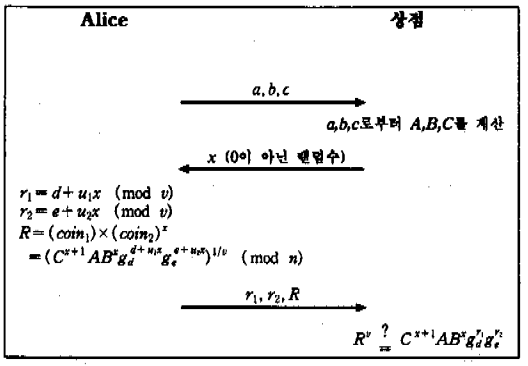

그림 6. 지불 프로토콜
Fig. 6 The Payment Protocol

〔사용자 식별자 U의 개설〕

Alice는 은행으로부터 전자화폐를 발급받기 전에 계좌개설을 먼저 수행한다. 물론 이때 Alice는 자신의 식별자 U를 생성하게 되며, 과정은 다음과 같다. Alice는 은행과 함께 Z_v^* 상에서 (u_1,u_2)를 랜덤하게 생성하고, 이것을 이용하여 $U = g_d^{u_1} g_e^{u_2} (mod\ n)$를 계산한다. 이때, 은행은 (u_1,u_2)의 값에 대해서는 전혀 모르며, 단지 U 값만을 알게 된다. 은행은 이후 U와 Alice의 신상정보를 함께 자신의 데이터베이스에 저장한다(U의 세부적인 생성 방법은 논문 [7]을 참조한다.).

〔인출 프로토콜〕

제안하는 두 번째 해결방안도 역시 Ferguson 시스템의 랜덤화된 은닉서명 방식에 기반을 둔다. 상세한 프로토콜 내용은 그림 5와 같다.

단계 1) Alice는 랜덤 수 $c_1, a_1, b_1, \sigma, \tau, \phi, d_1, e_1, \delta, \alpha, \beta, \gamma$들을 선택한다. 여기서 c_1, a_1, b_1는 베이스 수를 생성하기 위한 것이며, σ, τ, ϕ는 지수값에 대한 은닉 인자이다. 그리고, d_1, e_1은 기존의 전자화폐에 있는 k와 비슷한 역할을 수행하는 인자로서, (u_1,u_2)와 함께 response 들을 생성하기 위해 필요한 인자들이다. 그리고, $\delta, \alpha, \beta, \gamma$는 곱셈을 위한 은닉인자들이다. Alice는 선택한 인자들을 이용하여 $\alpha^v a_1 g_a^{\tau} \pmod n$, $\beta^v b_1 g_b^{\phi} \pmod n$, $\gamma^v c_1 g_c^{\sigma} \pmod n$, $\delta^v g_d^{d_1} g_e^{e_1} \pmod n$를 각각 계산한 후, 식별자 U와 함께 은행에 전송한다.

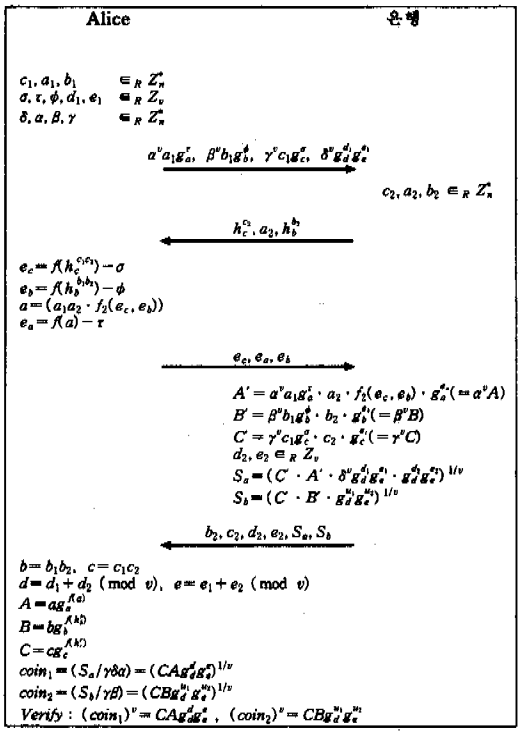

그림 7. 인출 프로토콜
Fig. 7 The Withdrawal Protocol

단계 2) 은행은 Alice가 전송한 인자들을 수신한 후, c_2, a_2, b_2를 생성하여 $h_c^{c_2} \pmod p$, a_2, $h_b^{b_2} \pmod p$를 계산한다. 그리고, 그 결과들을 Alice에게 전송하다.

단계 3) Alice는 수신된 값들을 이용하여 $e_c = f(h_c^{c_1c_2}) - \sigma \pmod v$와 $e_b = f(h_b^{b_1b_2}) - \phi \pmod v$를 계산한다. 또한, $a = a_1a_2 \cdot f_2(e_c, e_a) \pmod n$를 계산하고, 이것을 이용하여 $e_a = f(a) - \tau \pmod v$를 계산한다. 그리고 Alice는 e_c, e_a, e_b를 은행에 전송한다.

단계 4) 은행은 수신된 e_c, e_a, e_b를 이용하여 C, A, B가 은닉된 형태인 $C' = \gamma^v c_1 g_c^{\sigma} \cdot c_2 \cdot g_c^{e_c} \pmod n$, $A' = \alpha^v a_1 g_a^{\tau} \cdot a_2 \cdot f_2(e_c, e_b) \cdot g_a^{e_a} \pmod n$, $B' = \beta^v b_1 g_b^{\phi} \cdot b_2 \cdot g_b^{e_b} \pmod n$를 계산한다. 그리고, d와 e를 생성하기 위해서 랜덤 수 d_2, e_2를 선택한 후, 계산된 C', A', B'를 이용하여 $S_a = (C' \cdot A' \cdot \delta^v g_d^{d_1} g_e^{e_1} \cdot g_d^{d_2} g_e^{e_2})^{1/v} \pmod n$, $S_b = (C' \cdot B' \cdot g_d^{u_1} g_e^{u_2})^{1/v} \pmod n$를 계산한다. 여기서, $g_d^{u_1} g_e^{u_2}$는 은행에게 알

려져 있는 Alice의 공개된 식별자 U이다. 계산 후, 은행은 $b_2, c_2, d_2, e_2, S_a, S_b$를 Alice에게 전송한다.

단계 5) Alice는 수신된 수로부터 $b = b_1 b_2 (\bmod n)$, $c = c_1 c_2 (\bmod n)$, $d = d_1 + d_2 (\bmod v)$, $e = e_1 + e_2 (\bmod v)$를 각각 계산한다. 그리고 $C = cg_c^{f(k_2')} (\bmod n)$, $A = ag_a^{f(a)} (\bmod n)$, $B = bg_b^{f(k_2')} (\bmod n)$를 계산한 후, $coin_1$과 $coin_2$를 각각 $S_a / \gamma\delta\alpha = (CAg_d^d g_e^e)^{1/v} (\bmod n)$, $S_b / \gamma\beta = (CBg_d^{u_1} g_e^{u_2})^{1/v} (\bmod n)$로 계산한다. 그리고, 마지막으로 $coin_1$과 $coin_2$를 검증하기 위해 각각 $(coin_1)^v \stackrel{?}{=} CAg_d^d g_e^e (\bmod n)$, $(coin_2)^v \stackrel{?}{=} CBg_d^{u_1} g_e^{u_2} (\bmod n)$를 계산하여 비교한다.

참고 : 상기 프로토콜에서 Ferguson 시스템과 마찬가지로 베이스 수들은 모듈라 n 상에서 계산이 이루어지며, 지수부분은 모두 모듈라 v 상에서 계산이 이루어진다.

상기의 인출 프로토콜은 Ferguson의 인출 프로토콜에 비해 계산량이 모듈라 멱승 연산 3번, 모듈라 곱셈 감소가 5번 더 많지만 실질적으로 비교하면 그것보다는 다소 줄어들 수 있다. 왜냐하면, Ferguson 시스템은 인출 프로토콜에서 은행의 독자적인 부정행위를 방지하기 위해서, 사용자가 인출 프로토콜 중간에 그 동안 송수신 되었던 정보들에 대해 디지털 서명을 생성하여 그 결과를 은행에 전송하는 것이 필요하기 때문이다. 즉, 은행이 사용자가 전자화폐를 인출하지 않았음에도 불구하고, 사용자의 식별자 U를 무단으로 이용하여 사용자가 인출한 것처럼 은행이 시뮬레이션 할 수 있다는 것이다. Ferguson 시스템과 같이 은행이 인출 프로토콜에서 얻은 정보만을 가지고 사용자인 것처럼 지불 프로토콜을 수행할 수 있는 경우에는, 은행의 부정행위를 방지하기 위하여 일반적으로 인출 프로토콜 중간에 사용자의 디지털 서명을 추가하게 된다. 본 논문에서 제안하는 두 번째 해결방안은 인출 프로토콜에 이러한 사용자의 디지털 서명이 필요없게 된다. 왜냐하면 비록, 은행이 스스로 시뮬레이션 하여 어떤 사용자 식별자 U를 가지고 전자화폐를 발급하였다 하더라도, U를 구성하고 있는 지수 (u_1, u_2)의 값을 모르기 때문에 지불 프로토콜을 수행할 수 없게 된다.

결국, 자신이 부정하게 발급한 전자화폐를 사용할 수 없기 때문에, 이러한 부정행위는 자동적으로 방지되게 된다. 그러므로, Ferguson의 인출 프로토콜에 직접적으로 나타나 있지 않은 계산량은 줄어들 수 있게 되므로, 인출 프로토콜에서의 계산량은 비교적 허용할 만한 것이 된다.

[예치 프로토콜]

II. 1절에서 설명된 Ferguson 시스템의 예치 프로토콜과 같다.

V. 결 론

Ferguson의 전자화폐 시스템은 효율성 면에서 매우 뛰어난 개념을 갖는 전자화폐 시스템이다. 그러나, 불행히도 그의 시스템은 시스템의 기본적인 요구사항인 안전성을 위배한다는 중요한 문제점을 지니고 있었다. 본 논문에서는 이러한 문제점을 지적하였고, 또한 그러한 문제점을 해결할 수 있는 두 가지 해결방안을 제안하였다. 첫 번째 제안한 방안의 장점은 Ferguson의 전자화폐 시스템을 크게 수정할 필요없이 간단한 수정만 가하여 문제를 해결할 수 있다는 것이다. 그러나, 사용자가 지불 단계에서 전자화폐를 지불하는 프로토콜을 수행하는 것과 더불어 서명 프로토콜을 함께 수행해야 한다는 점에서 오버헤드를 가지게 되었다. 이러한 시스템의 효율성 측면을 고려하여 제안한 것이 두 번째 해결방안이다. 이 해결책은 Ferguson 시스템이 내포하고 있는 문제점의 근본 원인을 고려한 것으로서, Brands의 식별자 표현방법을 Ferguson의 시스템에 맞도록 변경한 것이다. 이것은 첫 번째 해결방안에 비해 매우 효율적인 방식이며, 기존의 Ferguson 시스템과 거의 비슷한 계산량을 가진 상태로도 시스템의 문제점을 해결할 수 있다는데 큰 의의를 갖는다.

참 고 문 헌

[1] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems", *Communications of the Association for Computing Machinary*, vol. 21, no. 2, pp. 120-126, 1978.

[2] N. Ferguson, "Single Term Off-line Coins", *Advances in Cryptology, Proc. of EUROCRYPT'93*, pp. 318-328, 1993.

[3] N. Ferguson, "Extensions of Single Term Coins", *Advances in Cryptology, Proc. of*

CRYPTO'93, pp. 292-301, 1993.

[4] A. Shamir, "How to Share A Secret", *Communications of the Association for Computing Machinary*, vol. 22, no. 11, pp. 612-613, 1979.

[5] Y.C. Choi, S.J. Kim and D.H. Won, "The Problem of Single Term Off-line Coins and the Countermeasures", *ITC-CSCC97(International Technical Conference on Circuits/Systems, Computers and Communications)*, vol. 1 pp. 587-590. 1997.

[6] K, Nyberg and R.A. Rueppel, "A New Signature Scheme Based on the DSA Giving Mesaage Recovery", *Proc. 1st of ACM Conference on Computer and Communications Security*, pp. 58-61,1993.

[7] S. Brands. "An Efficient Off-line Electronic Cash Systems Based on the Representation Problem", *C.W.I. Technical Report CS-T9323*, The Netherlands, 1993.

[8] D. Chaum, "Blind Signatures for Untraceable Payments", *Proc. of CRYPTO'82*, Plenum Press, pp. 199-203, 1983.

최 영 철(Young-Chul Choi) 정회원

1970년 4월 5일생

1996년 2월 : 성균관대학교 정보공학과 졸업 (공학사)

1998년 2월 : 성균관대학교 대학원 정보공학과 졸업 (공학석사)

1997년 10월~현재 : 한국정보보호센터 연구원

김 승 주(Seung-Joo Kim) 정회원

1971년 9월 22일생

1994년 2월 : 성균관대학교 정보공학과 졸업 (공학사)

1996년 2월 : 성균관대학교 대학원 정보공학과 졸업 (공학석사)

1996년 3월~현재 : 성균관대학교 대학원 정보공학과 박사과정

원 동 호(Dong-Ho Won) 정회원

1949년 9월 23일생

1976년 2월 : 성균관대학교 전자공학과 졸업 (공학사)

1978년 2월 : 성균관대학교 대학원 전자공학과 졸업 (공학석사)

1988년 2월 : 성균관대학교 대학원 전자공학과 졸업 (공학박사)

1978년 4월~1980년 3월 : 한국전자통신연구소 연구원

1985년 9월~1986년 8월 : 일본 동경공대 객원연구원

1982년 3월~현재 : 성균관대학교 공과대학 정보공학과 교수

1995년 3월~1997년 2월 : 성균관대학교 교학처장

1996년 4월~1998년 4월 : 정보화추진위원회 자문위원

<주관심분야> 암호이론, 정보이론