거스름의 재사용이 가능한 새로운 오프라인 수표시스템

(A New Offline Check System with Reusable Refunds)

김 상 진 † 최 이 화 †† 오 희 국 †††
(Sangjin Kim) (Ihwa Choi) (Heekuck Oh)

요 약 오프라인 수표시스템에서는 수표의 액면가와 지불대금이 다르면 거스름이 발생한다. 대부분의 오프라인 수표시스템에서는 거스름을 지불에 사용하지 못하고 이를 돌려받기 위한 환불 프로토콜을 진행하여야 한다. 그러나 거스름을 지불에 사용할 수 있다면 수표방식은 환불이 필요 없는 편리한 지불 수단이 된다. 이 논문에서는 거스름을 지불에 사용할 수 있는 새로운 오프라인 수표시스템을 제안한다. 새 시스템에서는 상점이 거스름 수표를 발행하도록 하여 이를 지불에 사용할 수 있도록 개선하였다. 상점은 자신의 개인키를 이용해서 은행이 발행하는 수표와 동일한 형태의 거스름 수표를 발행하고, 이것이 유통되어 은행에 입금되면 수표의 금액만큼 해당 상점에게 청구하는 방식이다. 이 시스템에서 수표의 형태는 기존 시스템에 비해 매우 간단하며 액면가 표현에 있어서도 제한이 없다. 또한 지불에 사용한 수표와 거스름 수표의 연결이 어려우며, 거스름 수표도 익명으로 지불할 수 있다. 그밖에 수표의 불법적인 사용을 막기 위해 화폐 추적, 인출자 추적이 가능한 조건부 익명성을 제공하고 트랜잭션의 원자성을 보장한다.

키워드 : 전자화폐, 오프라인 전자수표, 거스름의 재사용

Abstract In offline check systems, a client does not have to pay the exact amount. Instead, a client refunds the difference between the check value and the paid amount. In most offline systems, clients can not spend the remainder. But if the refund can be made spendable, it would provide a more convenient payment method. In this paper, we present a new offline system, which allows refunds to be reused as payments. In our system, the shop issues a new check using its private key for the difference. This new check, called the refund check, can be spent in the same way as checks issued by the bank. If the refund check is deposited to, or refunded at the bank, the bank charges the issuer of the check for the amount. The form of a check in this system is much simpler than previous check systems. It also uses a more flexible and efficient denomination method. The refund check is unlinkable to the check used in the payment where the refund check was issued. This system provides coin and owner tracing mechanisms to reinforce controls on illegal use of anonymous checks and was designed with consideration to the atomicity of transactions.

Key words : electronic cash, offline electronic check, reusable refund

1. 서 론

전자화폐는 지불 방법에 따라 동전방식[1], 분할방식[2,3], challenge-semantic 방식[4], 수표방식[5-12] 등으로 분류할 수 있다. 동전방식은 실물화폐처럼 각 동전이 100원, 500원 등의 고정된 액면가를 가지는 방식이다. 일반적으로 고객은 지불대금과 일치하도록 여러 개의 동전을 조합하여 지불하며, 지불 비용은 사용하는 동전 개수에 비례하여 증가한다. 또한 고객이 가지고 있는 동전을 조합하여 지불대금을 만들 수 없으면 지불할 수 없다. 분할방식은 인출 받은 화폐를 액면가보다 작은 여러 개의 화폐로 나누어 사용하는 방식이다. Challenge-semantic 방식은 하나의 화폐를 가지고 다양한 금액으로 여러 번 지불할 수 있는 방식이다. 이 두 방식은 지

† 정 회 원 : 한국기술교육대학교 인터넷미디어공학부 교수
sangjin@kut.ac.kr
†† 비 회 원 : 소프트포럼 연구원
inchoi@softforum.com
††† 종신회원 : 한양대학교 전자컴퓨터공학부 교수
hkoh@cse.hanyang.ac.kr
논문접수 : 2001년 12월 17일
심사완료 : 2002년 9월 3일

불대금에 맞춰서 금액을 자유롭게 사용할 수 있다는 장점이 있지만 같은 화폐를 사용한 여러 지불이 서로 연결된다는 문제가 있다. 물론 최근에 서로 연결할 수 없는 분할 가능한 화폐가 제안되었지만 지불 과정에서 cut-and-choose 기법을 사용하기 때문에 계산비용이 많이 드는 시스템이다[3]. 수표방식은 수표의 액면가 한도 내에서 어떤 금액에 대해서도 지불할 수 있는 방식이다. 이 방식에서는 수표의 액면가와 지불대금이 일치하지 않으면 차액만큼의 거스름을 돌려 받는다.

전자수표는 실세계에서의 수표와 특성이 조금 다르다. 서양의 수표는 신용 기반의 지불 방식이다. 즉, 수표에 지불대금을 명시하여 지불하면 유통 후에 그 금액만큼 고객에게 청구되는 후불 방식이다. 이러한 신용 기반의 후불 방식은 디지털 서명을 사용하면 쉽게 구현할 수 있다. 그러나 전자화폐로서의 전자수표는 실세계의 수표를 구현해 놓은 것이 아니다. 전자수표는 실세계 수표와 다르게 선불 방식의 화폐이며 익명이 보장되고 지불 후에 거스름이 발생할 수 있다. 대부분의 문헌[5-12]에서 이러한 특성을 지닌 화폐를 전자수표라 표현하고 있어서 이 논문에서도 이 용어를 사용한다.

수표방식은 여러 개의 화폐를 지불해야 하는 동전방식과 비교해 편리한 지불 수단이지만 거스름 처리가 복잡하고 효율적인 거스름 메커니즘을 제공하기가 어렵기 때문에 많은 주목을 받지 못하고 있다[5-10]. 온라인 방식에서는 수표의 발행기관인 은행이 지불에 참여하기 때문에 지불 과정에서 발생하는 거스름을 쉽게 만들어 줄 수 있으며 이것을 지불에 사용할 수도 있다[11]. 그러나 오프라인 시스템에서는 은행이 지불에 참여하지 않으므로 온라인 시스템에서와 같은 거스름 처리가 어렵다. 이런 문제 때문에 기존 시스템에서는 인출 단계부터 수표를 두 부분으로 구성하여 하나는 지불에 사용하고 다른 하나는 거스름을 돌려 받기 위해 사용한다. 이때의 거스름은 지불에 사용할 수 없고 항상 환불 프로토콜을 이용하여 돌려 받아야 한다[5-8].

이 논문은 거스름을 지불에 사용할 수 없다는 기존 오프라인 수표시스템의 문제점을 해결한 새로운 수표시스템을 제안한다. 지불에 사용할 수 있도록 만든 수표 형태의 거스름을 이 논문에서는 "거스름 수표"라 한다. 새 시스템에서는 이 거스름 수표를 은행 대신에 상점이 발행하도록 하였다. 은행이 고객에게 수표를 발행하는 과정과 동일하게 상점은 자신의 서명키로 거스름 수표를 발행한다. 이 수표는 은행이 발행한 수표처럼 지불에 사용할 수 있으며, 실세계의 상품권과 개념적으로 유사하다. 다만, 그 상품권을 발행한 상점뿐만 아니라 다른 상점에서도 사용할 수 있다는 점이 다르다. 거스름 수표가 유통되어 은행에 입금되면 수표를 발행한 상점에게 수표의 액면가만큼을 청구한다. 수표의 발행권한을 상점에게 위임하는 경우, 상점은 이 권한을 남용할 수 있다. 그러나 새 시스템은 불법적으로 거스름 수표를 발행한 상점을 밝혀낼 수 있도록 고안하였다.

이 논문에서 제안하고 있는 오프라인 수표시스템의 특성을 간단히 요약하면 다음과 같다.

- 상점이 거스름 수표를 발행하는 시스템이며, 거스름 수표에 대해서는 후불방식이다.
- 분할방식이나 challenge-semantic 방식과는 달리 지불에 사용된 수표와 거스름으로 발행된 수표가 서로 연결되지 않는다.
- 액면가에 따라 생성자 튜플의 길이가 증가하는 기존 시스템과 달리 하나의 생성자를 사용하여 수표의 액면가를 표현한다.
- 돈세탁, 협박, 불법구매와 같은 범죄에 대응하기 위해 필요하면 익명성을 철회할 수 있는 조건부 익명성을 제공한다.
- 인출 프로토콜, 지불 프로토콜, 입금 프로토콜의 원자성을 보장한다.

이 논문의 구성은 다음과 같다. 2장에서는 새 시스템과 관련된 연구에 대해 살펴보고 3장에서는 이 논문에서 제안하는 오프라인 수표시스템을 서술한다. 4장에서는 새 시스템의 안전성과 원자성에 대해 살펴보며, 기존 수표시스템과 비교를 한다. 끝으로 5장에서는 결론과 향후 연구 방향에 대해 서술한다.

2. 관련 연구

온라인 수표시스템은 1989년 Chaum[9]에 의해 처음 소개되었다. RSA 비밀지수로 서명하여 수표의 액면가를 표현하는 이 시스템은 고정된 액면가의 수표만을 인출할 수 있으며 거스름을 축적하기 위해 쿠키통(cookie-jar)을 사용한다. 이것을 지불에 사용할 수 있지만 액면가 한도 내에서 자유롭게 사용할 수 없고 일부 금액에 대해서만 지불이 가능하다. Deng 등은 온라인 지불의 단점을 보완하기 위해 다중 지불세션이라는 새로운 방식의 지불 프로토콜을 제안하였다[10]. 이 프로토콜은 하나의 수표로 한 상점과 오프라인으로 여러 번 거래할 수 있다. 이 시스템은 일회성 공개키를 수표의 일련번호로 사용하여 지불의 안전성을 높였고, 고정된 하나의 공개 지수를 이용하여 수표의 액면가를 표현한다. 그러나 이 시스템은 거스름과 수표 자체가 전혀 다른 형태여서 거스름을 지불에 사용하기 어렵다는 문제점이 있다. 최

근에 이러한 기존 온라인 수표시스템의 거스름 재사용 문제를 해결한 수표시스템이 발표되었다[11]. 이 시스템은 부분은닉서명(partially blind signature)[13]을 사용하여 수표의 액면가를 임의로 표현할 수 있으며 거스름의 형태가 수표와 같아서 이를 지불에 사용할 수 있다. 또한 Deng 등의 시스템처럼 다중 지불세션 프로토콜을 제공한다. 그러나 이 시스템도 온라인이라는 근본적인 한계를 벗어나지 못했다.

Chaum 등[5]은 처음으로 오프라인 방식의 수표시스템을 발표하였으며, Hirschfeld[6]는 이 시스템을 보완한 시스템을 발표하였다. 이 두 시스템은 인출 단계에서 고객이 전달한 값에 은닉서명을 하기 위해 cut-and-choose 기법을 사용한다. 고객은 은행으로부터 받은 값을 두 부분으로 나누어 그 중 하나는 지불에 사용하고, 다른 하나는 남은 잔액을 환불받기 위해 사용한다. 그러나 두 시스템 모두 cut-and-choose 기법을 사용하기 때문에 인출할 때와 지불할 때 계산량과 정보교환량이 많아서 시스템의 효율이 떨어진다. Brands[7]는 cut-and-choose 기법을 사용하지 않는 오프라인 수표시스템을 처음으로 발표하였다. 이 시스템은 cut-and-choose 기법을 사용하지 않아도 유효한 서명을 얻을 수 있는 제한적 은닉서명기법(restrictive blind signature)을 이용하며, 표현 문제(representation problem)를 이용해서 수표의 형태를 구성하였다. Solages와 Traore가 제안한 시스템[8]은 Brands가 제안한 시스템에 익명성 제어 기능을 추가하였고, 요구되는 연산의 양을 줄여서 효율성을 향상시켰다.

오프라인 수표시스템에서는 지불 과정에 은행이 참여하지 않으므로 거스름을 처리하기가 어렵다. 이 때문에 앞서 설명한 오프라인 수표시스템[5-8]은 수표를 인출 단계부터 지불을 위한 부분과 환불받기 위한 부분으로 구성한다. 또한 기존 시스템들은 액면가 표현방법의 한계와 고객의 익명성을 보장하기 위해 고정된 금액의 수표만을 인출할 수 있다. 그러나 인출한 수표를 한번 밖에 사용할 수 없기 때문에 항상 남은 잔액을 환불받아야 하는 불편함이 있으며, 인출할 수 있는 수표의 액면가가 고정되어 있으므로 지불액과 환불액 간에 보수관계(complementary relationship)가 성립하여 고객 익명성에 나쁜 영향을 준다. 최근에 이러한 기존 시스템의 단점을 다소 해결하여 남은 금액을 지불에 사용할 수 있는 오프라인 수표시스템이 발표되었다[12]. 이 시스템은 하나의 수표를 인출할 때 두 개의 값에 은행의 은닉서명을 받아서 거스름에 대해 한번 더 지불할 수 있도록 개선하였지만 거스름을 재사용할 수 있는 횟수에 제한이 있다는 문제점이 있다. 이 논문에서는 상점이 거스름 수표를 발행하도록 하여 오프라인 수표시스템이 가지고 있는 거스름 문제를 해결한 새로운 시스템을 제안한다.

3. 제안하는 오프라인 수표시스템

이 장에서는 오프라인 수표시스템에서 발생하는 거스름을 지불에 사용할 수 있는 새로운 시스템을 제안한다. 이 시스템은 Solages와 Traore[8]의 오프라인 수표시스템을 기본 골격으로 사용하고 있다. 따라서 이 시스템도 표현 문제[1]를 이용하여 수표를 구성하며, Solages와 Traore의 제한적 은닉서명과 이들이 제안한 익명성 제어 메커니즘을 활용한다. 그러나 새 시스템은 Solages와 Traore의 시스템과 달리 이중구조의 수표를 사용하지 않고, 수표의 형태가 간단하며, 다양한 액면가를 제공한다. 또한 거스름을 지불에 사용할 수 있으며, 일어나는 트랜잭션의 원자성을 고려하여 시스템의 각종 프로토콜을 설계하였다. 이 시스템의 특징을 나타내는 시스템의 흐름도는 그림 1과 같다. 이 흐름도는 이 시스템에서 일어날 수 있는 여러 지불 유형 중 하나이다.

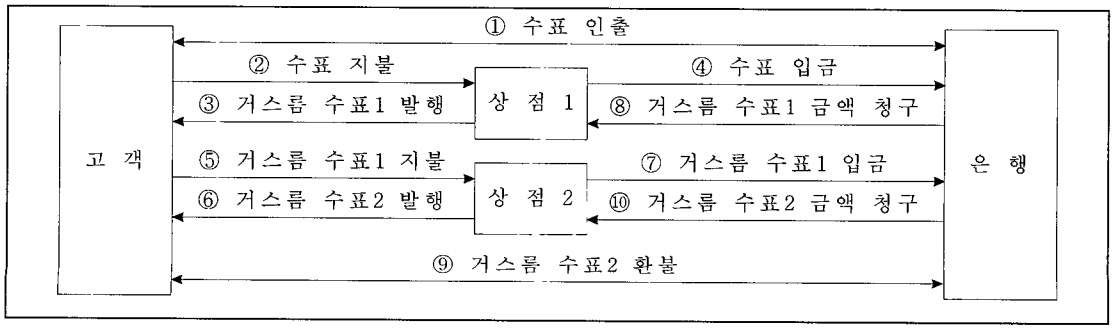

그림 1 새 시스템의 흐름도

3.1 시스템 설정

이 논문에 있는 모든 수학 연산은 군(group)의 위수(order)가 매우 큰 소수 q인 G_q군에서 이루어진다. 이 군은 먼저 큰 소수 p를 선택하고 p-1의 소인수 중 하나인 q를 선택하여 구성한다. G_q군은 곱셈군 Z^*_p의 부분군(subgroup)으로 1을 제외한 모든 원소는 법 p에서 G_q군의 생성자가 된다. 시스템의 안전성은 이 군에서 이산대수(discrete logarithm)를 구하는 것이 계산적으로 어렵다는 것에 기반한다. 이 논문에서 지수 요소와 관련된 연산은 법 q에서 이루어지고 나머지 연산은 모두 법 p에서 이루어진다. 이후, 논문에서는 법 p와 q에 대한 연산 표기를 생략한다. 이 시스템에서 사용하는 해쉬함수 H_q 는 임의의 길이의 입력을 받아 Z_q의 원소를 출력하는 충돌회피(collision-resistant) 해쉬함수이다.

시스템에 참여하는 구성원은 신뢰기관, 은행, 상점, 고객이다. 시스템 설정과 계정 개설에 필요한 구성원들의 파라미터는 표 1에 요약되어 있다. 고객과 상점은 다음 절에 설명된 방법으로 은행에 계정을 개설하여야 지불에 참여할 수 있다. 신뢰기관은 프로토콜 상에서 발생할 수 있는 여러 분쟁을 중재하고, 필요한 경우 추적 메커니즘을 통해서 수표의 사용 경로나 수표 사용자의 신원을 밝히는 역할을 한다. 신뢰기관은 G_q에 속하는 생성자 g_T를 선택하고, 화폐 추적을 위한 개인키 $x_{CT} \in Z^*_q$와 인출자 추적을 위한 $x_{OT} \in Z^*_q$를 선택한다. 그 다음에 대응되는 공개키 $y_{CT} = g_T^{x_{CT}^{-1}}$와 $y_{OT} = g_T^{x_{OT}^{-1}}$을 계산하고, g_T와 두 개의 공개키를 공개한다. 은행은 표 1에 설명되어 있는 G_q군에 속하는 다섯 개의 생성자 g_B, g_V, g_U, g_S, g_D를 임의로 선택한다. 그 다음 수표를 발행할 때 사용할 개인키 $x_B \in Z^*_q$를 임의로 선택하고 대응되는 공개키 $y_B = g_B^{x_B}$을 계산한다. 또한 거스름 수표 교환 프로토콜에서 필요한 $y_B' = g_D^{x_B}$도 계산한다. 은행은 p, q, g_B, g_V, g_U, g_S, g_D, y_B, y'_B을 공개한다. 신뢰기관의 공개키 쌍과 은행이 사용하는 공개키 쌍의 형태가 다른 것은 신뢰기관의 개인키는 서명할 때 사용되는 키가 아니라 화폐 추적과 인출자 추적에 사용되는 키이기 때문이다. 3.7절에 추적 메커니즘을 보면 이런 형태의 키를 사용하는 이유를 쉽게 알 수 있다.

이 시스템에서 사용되는 수표의 형태는 표현 문제를 이용하여 다음과 같이 구성한다.

$$C_B = g_U^{x_U} g_V^{\nu} g_T^{r}$$

여기에서 x_U는 고객의 비밀신원정보이고, ν는 수표의 액면가이며, r은 고객의 신원 정보를 감추고 추적할 때 사용하는 요소이다. 생성자 튜플 (g_U, g_V, g_T)에 대한

표 1 시스템 설정과 계정 개설에 필요한 파라미터

신뢰기관	g_T	신뢰기관의 공개키를 위한 생성자
	$x_{OT} \in_R Z^*_q, y_{OT} = g_T^{x_{OT}^{-1}}$	인출자 추적(owner tracing)을 위한 신뢰기관의 공개키쌍
	$x_{CT} \in_R Z^*_q, \ y_{CT} = g_T^{x_{CT}^{-1}}$	화폐 추적(coin tracing)을 위한 신뢰기관의 공개키쌍
은행	g_B	은행의 공개키를 위한 생성자
	g_V	수표 구성에서 액면가 정보를 표현하기 위한 생성자
	g_U	고객의 신원 정보를 표현하기 위한 생성자
	g_S	상점의 신원 정보를 표현하기 위한 생성자
	g_D	거스름 수표 발행을 위한 생성자
	$x_B \in_R Z^*_q, \ y_B = g_B^{x_B}$	수표 발행을 위한 공개키쌍
	$y'_B = g_D^{x_B}$	거스름 수표 교환 프로토콜에 사용되는 공개정보
상점	$x_S \in_R Z^*_q, \ y_S = g_S^{x_S}$	상점의 비밀신원정보와 공개신원정보 상점의 거스름 수표 발행을 위한 공개키쌍
	$y'_S = g_D^{x_S}$	거스름 수표 발행을 위한 공개정보
고객	$x_U \in_R Z^*_q, \ y_U = g_U^{x_U}$	고객의 비밀신원정보와 공개신원정보

C_B의 표현은 $(x_U,\ v,\ r)$이 된다. 표현 문제는 G_q에서 이산대수를 구하는 것이 계산적으로 어렵다고 가정할 때 C_B의 표현을 찾는 다항시간 알고리즘이 없다는 것에 바탕을 두고 있다[1].

3.2 계정 개설

고객은 은행에 계정을 개설하기 위해 자신의 비밀신원정보 $x_U \in Z_q^*$를 임의로 선택하고 대응되는 공개신원정보 $y_U = g_U^{x_U}$을 계산하여 y_U를 은행에 전달한다. 그 다음 기저 g_U에 대한 y_U의 이산대수 $\log_{g_U} y_U$를 알고 있음을 비상호작용으로 영지식 증명을 한다[14]. 이 증명은 그림 2에 기술되어 있으며, 이 논문에서는 ZK Proof ($Log_{g_U} y_U$)로 표기한다. 이산대수를 모를 때 이 증명을 하는 것은 계산적으로 어렵다. 그림 2에서의 T_U는 고객의 요청 시간을 나타내며, 이 값 때문에 증명은 매번 달라진다. 은행은 증명을 확인하고 y_U를 고객의 식별자로 기록해 놓는다. 시스템에 참여하는 상점도 고객과 유사하게 $x_S \in Z_q^*$를 임의로 선택하여 $y_S = g_S^{x_S}$을 계산하고 y_S를 은행에 전달하여 계정을 개설한다. 상점은 또한 거스름 수표 발행 과정에서 고객의 익명성을 보장할 때 사용하는 $y'_S = g_D^{x_S}$을 계산해서 은행에 전달한다. 그 다음 고객과 같은 방법으로 $\log_{g_S} y_S$를 알고 있음을 증명한다. 고객과 다른 점은 y_S는 상점의 식별자 역할뿐만 아니라 거스름 수표를 발행할 때 공개키로도 사용되며, 이 때 발행키는 x_S이다. 은행은 상점의 공개키와 y'_S을 함께 공개한다.

3.3 인출 프로토콜

고객은 자신의 신원을 증명하고, 수표의 추적이 가능하도록 추적 정보를 전달한 다음에 수표를 발행 받는다. 이 시스템의 인출 프로토콜은 그림 3에 기술되어 있다.

3.3.1 고객 신원 증명

고객은 자신의 공개신원정보 y_U를 은행에 전달하고 그림 2의 ZKProof ($\log_{g_U} y_U$) 과정을 진행하여 x_U를 알고 있음을 영지식으로 증명한다. 이 증명은 이산대수 영지식 증명이므로 x_U를 알고 있는 고객만이 증명을 구성할 수 있다. 더욱이 매번 새로운 인출시간 T_U를 포함하므로 공격자는 다른 인출에서 사용된 증명을 이용하여 신원을 가장할 수 없다. 은행은 이 과정을 통해서 고객의 신원을 확인한다.

3.3.2 수표 추적 정보 전달

고객은 수표에 사용할 은닉요소 $r \in Z_q^*$를 임의로 선택

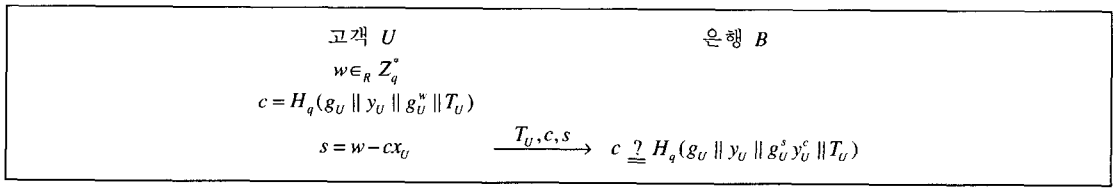

그림 2 이산대수 영지식 증명 프로토콜 ZKProof ($\log_{g_U} y_U$)

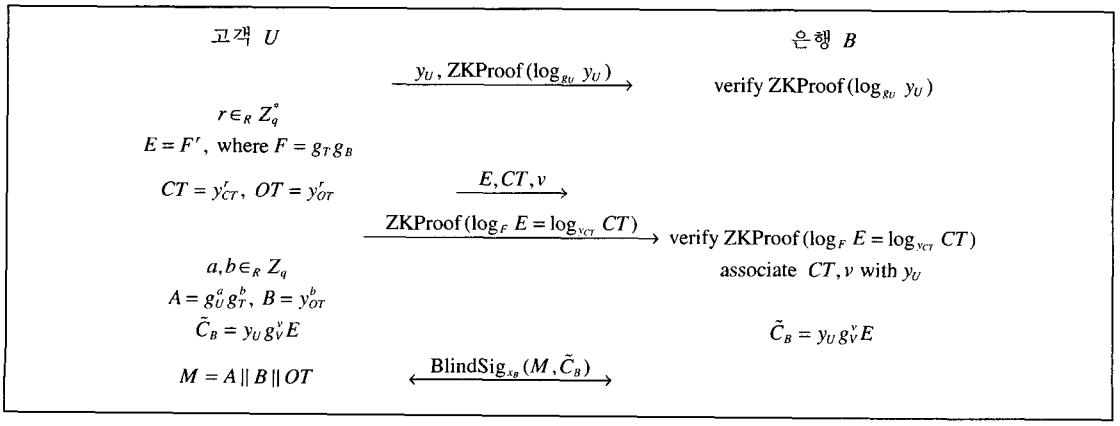

그림 3 수표 인출 프로토콜

하고 E, CT, OT를 계산한다. E는 은닉서명의 입력 $\tilde{C}_B$를 은행이 직접 계산할 때 사용되는 값이며, CT는 화폐 추적을 할 때 사용되는 값이고, OT는 인출자 추적을 할 때 사용되는 값이다. 고객은 E, CT, 인출하고자 하는 수표의 액면가 ν, ZKProof($\log_F E = \log_{y_{CT}} CT$)를 은행에 전달한다. ZKProof($\log_F E = \log_{y_{CT}} CT$)는 $\log_F E$와 $\log_{y_{CT}} CT$가 같음을 영지식으로 증명하는 비상호작용 증명이다[15]. 이 증명은 그림 4에 기술되어 있으며, 은행은 이 증명을 통해 E의 구성이 올바르고, CT를 이용하여 화폐 추적을 할 수 있음을 확인한다. 확인이 끝나면 은행은 인출 데이타베이스에 고객과 CT, ν를 연관시켜 놓는다.

3.3.3 수표 인출

고객과 은행은 각자 은닉서명을 할 $\tilde{C}_B = y_U g_1^{\nu} E$를 계산한다. 고객은 추가로 A와 B도 계산한다. A는 지불 과정에서 고객이 수표 C_B의 표현을 알고 있음을 상점에게 증명할 때 사용되며, B는 인출자 추적을 위한 OT의 유효성을 증명할 때 사용된다. A와 B를 서명에 포함하여 고정시킴으로써 고객은 지불 과정에서 이 때 포함한 값만을 이용하여 도전에 대한 응답을 할 수 있다. 고객과 은행은 M, $\tilde{C}_B$에 대해 그림 5에 기술된 제한적 은닉서명 프로토콜을 수행한다. 이 논문에서 사용하는 제한적 은닉서명은 Solages와 Traore[8]이 제안한 서명으로 BlindSig로 표기한다. 인출 프로토콜이 종료되면 고객은 C_B에 대한 은행의 서명 $\mathrm{Sig}_{x_B}(C_B) = (z, \gamma, \sigma)$을 얻게 되며, $z = C_B^{x_B}, \gamma, \sigma$ 은 다음 식을 만족한다.

$$\gamma = H(M \| C_B \| z \| g_B^{\sigma} y_B^{\gamma} \| C_B^{\sigma} z^{\gamma})$$

3.4 지불 프로토콜

이 시스템의 지불 프로토콜은 그림 6과 같이 다섯 단계로 구성되어 있으며, 실제 지불 프로토콜은 그림 7에 기술되어 있다. 고객이 구매하는 상품이나 서비스는 전자적인 형태라고 가정한다.

3.4.1 수표 지불

단계 1에서 고객은 수표를 상점에게 지불하고 수표의 유효성을 증명한다. 고객은 먼저 $C_B, A, B, OT, \mathrm{Sig}_{x_B}(C_B)$, 수표의 액면가 ν, 구매하려는 상품의 식별자 I를 상점에게 전달한다. 상점은 수표의 서명을 확인하고 현재 시간 T_s를 고객에게 전달한다. 상점은 시간 정보뿐만 아니라

고객 U		은행 B
$w \in_R Z_q^*$		
$c = H_q(F \| E \| y_{CT} \| CT \| F^w \| y_{CT}^w)$		
$s = w - cr$	$\xrightarrow{c,\ s}$	$c \stackrel{?}{=} H_q(F \| E \| y_{CT} \| CT \| F^s E^c \| y_{CT}^s (CT)^c)$

그림 4 이산대수 등가 영지식 증명 프로토콜 $\mathrm{ZKProof}(\log_F E = \log_{y_{CT}} CT)$

고객 U		은행 B
		$\tilde{z} = \tilde{C}_B^{x_B}$
		$\omega \in_R Z_q^*$
$\mu, \nu \in Z_q$	$\xleftarrow{\tilde{z}, \tilde{\alpha}, \tilde{\beta}}$	$\tilde{\alpha} = g_B^{\omega}, \tilde{\beta} = \tilde{C}_B^{\omega}$
$z = \tilde{z} / y_B^{\nu}$		
$\alpha = \tilde{\alpha}^{\mu} g_B^{\nu}, \beta = \tilde{\beta}^{\mu} \tilde{C}_B^{\nu} / \alpha^{r}$		
$\gamma = H_q(M \| C_B \| z \| \alpha \| \beta)$		
$\tilde{\gamma} = \gamma / \mu$	$\xrightarrow{\tilde{\gamma}}$	$\tilde{\sigma} = \omega - \tilde{\gamma} x_B$
$\tilde{\alpha} \stackrel{?}{=} g_B^{\tilde{\sigma}} y_B^{\tilde{\gamma}}, \tilde{\beta} \stackrel{?}{=} \tilde{C}_B^{\tilde{\sigma}} \tilde{z}^{\tilde{\gamma}}$	$\xleftarrow{\tilde{\sigma}}$	
$\sigma = \mu \tilde{\sigma} + \nu$		

그림 5 제한적 은닉서명 $\mathrm{BlindSig}(M, \tilde{C}_B)$

상품 정보와 같은 다른 여러 정보를 결합하여 T_S를 만들어 같은 T_S가 사용될 확률이 매우 적도록 할 수도 있다. 고객은 비상호작용으로 수표의 표현을 알고 있음을 증명하기 위해 C_B, A, B, T_S, y_S를 해쉬함수에 적용하여 도전값 c를 만든다. 그 다음 생성자 튜플 (g_U, g_T)에 대한 A의 표현 (a, b)를 이용하여 도전에 대한 응답 s_1과 s_2를 계산하여 상점에게 전달한다. 상점은 c와 $C = C_B / g_V^v$를 계산하고, A와 $g_U^{s_1} g_T^{s_2} C^c$를 비교한다. 이 비교를 통해 수표

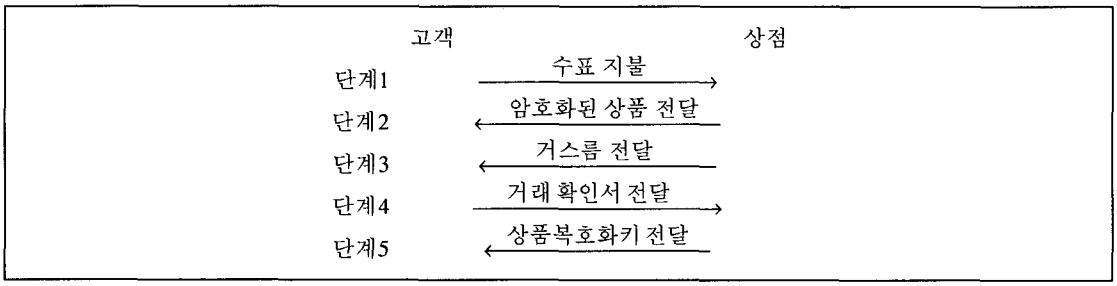

그림 6 지불 프로토콜의 5단계

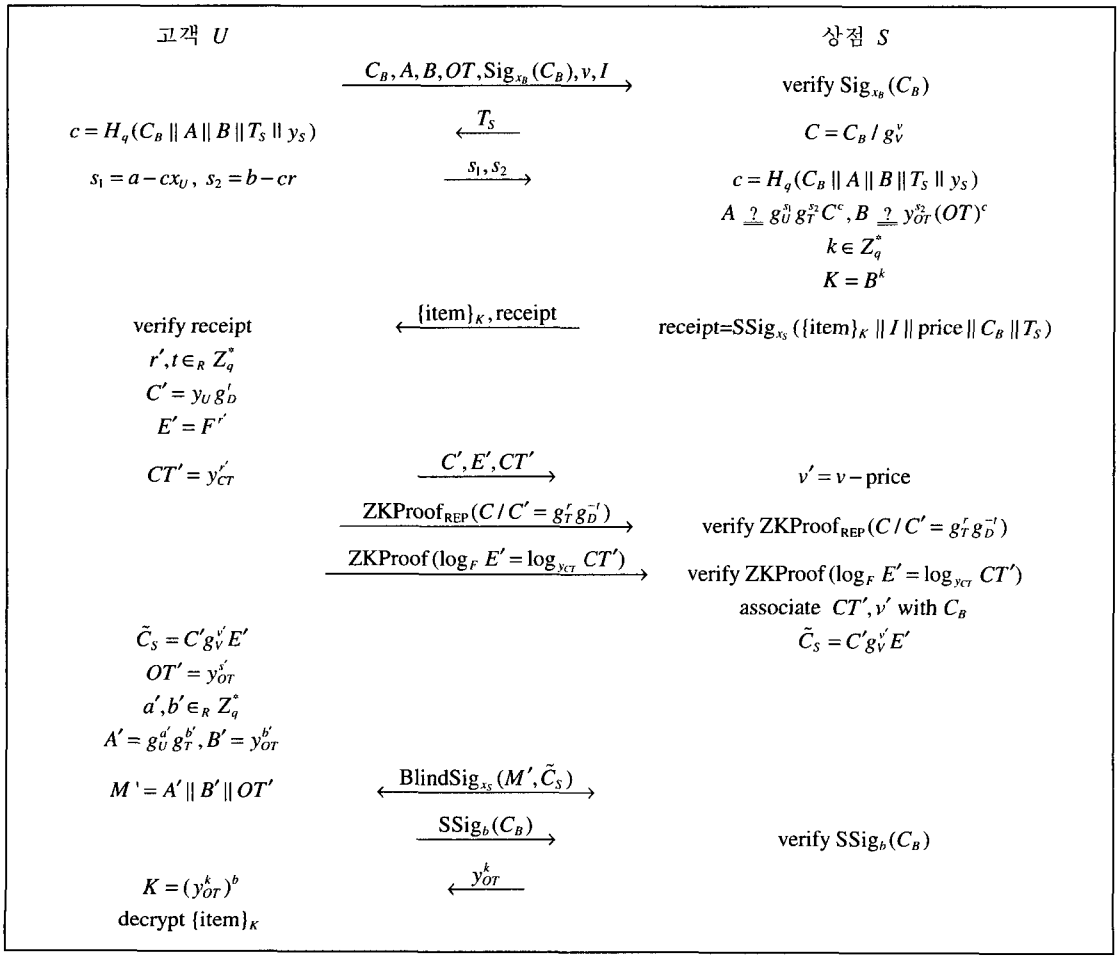

그림 7 지불 프로토콜

의 액면가와 지불자가 수표 C_B의 표현을 알고 있는지 확인한다. 그 다음 B와 $y_{OT}^{s_3}(OT)^c$을 비교하여 OT를 통해 C_B에 대한 인출자 추적을 할 수 있음을 확인한다.

3.4.2 암호화된 상품 전달

단계 2에서 상점은 Diffie-Hellman 비밀키[16] $K = B^k = y_{OT}^{bk}$을 생성하고, 고객이 구매하려는 상품을 K로 암호화하여 고객에게 전달한다. 상품을 암호화하여 전달하는 이유는 지불 프로토콜의 원자성을 보장하기 위한 것이다. 이와 관련된 내용은 4.3절에서 더 자세히 논의한다. 상점은 암호화된 상품과 영수증 $\mathrm{SSig}_{x_S}(\{\mathrm{item}\}_K \| I \| \mathrm{price} \| C_B \| T_S)$를 함께 전달한다. 영수증은 현재 프로토콜 수행과 바인딩되도록 암호화된 상품, I, 상품가격 price, C_B, 지불시간 T_S를 상점의 개인키 x_S를 이용하여 그림 8과 같이 Schnorr 서명[14]하여 만든다. 상점이 잘못된 상품이나 다른 복호화 키를 주는 경우 고객은 이 영수증을 이용하여 상점의 부정행위를 증명할 수 있다. 이 논문에서는 제한적 은닉서명의 결과와 구분하기 위해 Schnorr 서명의 결과는 SSig로 표기한다.

3.4.3 거스름 전달

단계 3에서 상점은 수표의 액면가와 지불대금의 차액 v'에 대한 거스름 수표 C_S를 발행하여 준다. 이를 위해 상점과 고객은 인출과 유사한 과정을 수행한다. 인출 과정에서 고객은 실명으로 참여하지만 여기서는 익명으로 참여하며, 고객의 익명성이 계속 유지되어야 한다. 고객은 익명으로 거스름 수표를 받기 위해 은닉요소 $t \in Z_q$를 임의로 선택하여 자신의 신원 정보 y_U를 은닉하고, 은닉된 결과 값 $C' = y_u g_D^t$을 상점에게 전달한다. 고객은 전달된 C'에 포함되어 있는 신원 정보와 지불한 수표 C_B에 포함되어 있는 신원 정보가 같음을 영지식으로 증명한다. 이 증명은 생성자 튜플 (g_T, g_D)에 대한 C/C'의 표현 $(r, -t)$를 알고 있음을 증명하는 비상호작용 영지식 증명이다. 이 증명은 그림 9에 기술되어 있으며, $\mathrm{ZKProof}_{\mathrm{REP}}(C/C' = g_T^r g_D^{\tilde{r}})$로 표기한다.

이 때문에 인출 때와는 달리 고객과 상점이 각자 만드는 은닉된 $\tilde{C}_S$의 형태는 $g_U^{xv} g_V^{v'} E' g_D^t$이 된다. $\tilde{C}_S$에는 두 개의 은닉요소가 포함되어 있으므로 이들을 제거하기 위해서는 그림 5에 기술된 제한적 은닉서명 프로토콜을 그대로 사용할 수 없다. 상점은 그림 5에서 $\tilde{z} = \tilde{C}_S^{x_S}, \tilde{\alpha}, \tilde{\beta}$를 전달할 때 $\tilde{\alpha}' = g_D^w$을 추가로 주며, 고객은 z와 β를 계산할 때 그림 5와 달리 다음을 수행한다.

$$
\begin{aligned}
z &= \tilde{z}/(y_S^{r'} y_S^{r''}) = (g_U^{xv} g_V^{v'} g_T^{r'})^{x_S} = C_S^{x_S} \\
\alpha &= \tilde{\alpha}^{\mu} g_B^{v} \\
\alpha' &= \tilde{\alpha}'^{\mu} g_D^{v} \\
\beta &= \tilde{\beta}^{\mu} \tilde{C}_S^{v} / (\alpha^{r'} \alpha'^{r''})
\end{aligned}
$$

은닉서명을 마치면 고객은 $C_S = g_U^{xU} g_V^{v'} g_T^{r'}$과 거스름 수표에 대한 상점의 서명 $\mathrm{Sig}_{x_S}(C_S)$를 얻는다. 상점은 화폐 추적을 위해 필요한 CT', v'을 C_B와 연관시켜 데이타베이스에 저장한다. 만약 수표의 액면가와 지불대금이 일치하면 이 단계는 생략할 수 있다.

고객은 거스름 수표 C_S를 그림 7과 동일한 지불 프

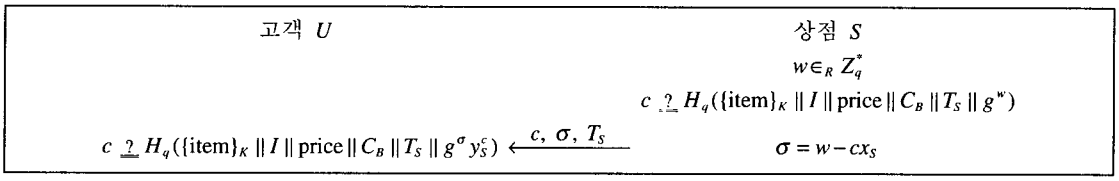

그림 8 영수증 $\mathrm{SSig}_{x_S}(\{\mathrm{item}\}_K \| I \| \mathrm{price} \| C_B \| T_S)$

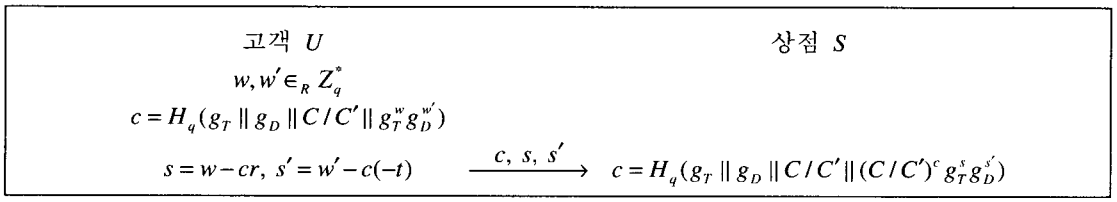

그림 9 $\mathrm{ZKProof}_{\mathrm{REP}}(C/C' = g_T^s g_S^{\tilde{r}'})$ 표현 영지식 증명 프로토콜

로토콜을 진행하여 다른 지불에 사용할 수 있다. 이 때, 수표를 발행해 준 상점의 식별자나 인증서를 함께 보내서 상점의 서명 $Sig_{x_S}(C_S)$을 확인할 수 있도록 한다. 거스름 수표는 은닉서명을 통해 발행되므로 고객의 익명성이 보장되며, 은행으로부터 인출할 수 있는 수표의 액면가가 고정되어 있지 않으므로 액면가 정보를 이용하여도 지불한 수표와 연관하는 것은 어렵다.

3.4.4 거래 확인서 전달

단계 4에서 고객은 자신만이 알고 있는 기저 y_{OT}에 대한 B의 이산대수 b를 키로 사용해서 그림 10에서와 같이 C_B에 Schnorr 서명을 한다. 이 서명은 $\mathrm{SSig}_b(C_B)$로 표기하며 이것을 "거래 확인서"라 한다. 이 확인서는 고객이 상품과 거스름을 받았다는 증거가 되며, 상점은 거래 확인서가 있어야 지불 받은 수표를 은행에 입금할 수 있다. 이 단계에서 b는 Schnorr 서명을 위한 고객의 개인키 역할을 하며, 대응되는 공개키는 $B\ (=y_{OT}^b)$이다. 상점은 B를 이용하여 고객의 서명을 확인한다.

3.4.5 상품복호화키 전달

단계 5에서 상점은 상품복호화키를 고객에게 전달한다. 이 때 Diffie-Hellman 비밀키 동의 프로토콜에 따라 상점이 고객에게 y_{OT}^k을 주면 고객은 b를 이용하여 상품을 복호화할 수 있는 K를 생성할 수 있다. 만일 공격자가 프로토콜 중간에서 암호화된 상품 정보와 y_{OT}^k를 가로채더라도 b를 알지 못하므로 고객이 구매한 상품을 불법적으로 얻을 수 없다. 고객은 K를 이용하여 상품을 확인한다. 만일 복호화한 상품이 요청한 상품이 아니거나 복호화가 제대로 되지 않으면 단계 2에서 받은 영수증을 제시하여 신뢰기관에게 중재요청을 한다.

3.5 입금 프로토콜

상점은 지불받은 수표를 입금하기 위해 고객으로부터 받은 수표 $C_B, A, B, OT, \mathrm{Sig}_{x_B}(C_B), v$, 도전과 응답 과정을 확인하기 위한 s_1, s_2, T_S, y_S, 거래 확인서 $\mathrm{SSig}_b(C_B)$를 은행에 전달한다. 은행은 c를 직접 만들고 상점이 지불 과정에서 했던 것과 같은 방법으로 지불 트랜스크립트의 유효성을 확인한다. c를 직접 계산하는 것은 청구할 수 있는 상점이 맞는지 확인하기 위함이다. 그 다음 입금 데이타베이스를 검사하여 수표의 이중사용 여부를 확인하고 입금된 수표의 액면가만큼을 상점의 계좌에 입금하여 준다. 입금된 수표가 특정 상점이 발행한 수표라면 월말이나 일정 기간이 지난 후에 그 때까지 입금된 수표를 모아서 상점과 정산 과정을 진행한다.

만일 입금된 수표가 이중사용 되었다면, 은행은 다음과 같은 절차를 통해 수표를 이중사용한 고객의 신원을 밝혀낼 수 있다. 같은 수표를 지불 받은 상점 S와 S'이 지불 프로토콜에서 받은 도전, 응답값이 각각 $c, s_1\ (=a-cx_U)$와 $c', s_1'\ (=a-c'x_U)$이라면, 다음과 같은 계산을 통해 고객의 신원 정보를 밝혀낼 수 있다.

$$\frac{s_1 - s_1'}{c' - c} = x_U$$

3.6 환불 프로토콜

더 이상 사용하지 않을 수표는 지불 프로토콜의 단계 1을 은행과 진행하여 환불받을 수 있다. 만일 환불 요청한 수표가 특정 상점이 발행한 거스름 수표라면 은행은 어느 상점이 발행한 것인지 알 수 있다. 그러나 거스름 수표는 은닉서명을 통해 발행되기 때문에 발행한 상점과 은행이 공모하더라도 이전 지불 정보와는 연결할 수 없다. 또한 수표의 액면가가 고정되어 있지 않기 때문에 거스름 금액을 가지고 지불 정보와 연결하는 것 역시 어렵다.

만일 환불 요청한 수표가 어느 상점으로부터 인출 받은 거스름 수표인지 감추고 싶다면 은행과 수표 교환 프로토콜을 진행하여 은행이 발행하는 수표로 바꾼 후에 환불받을 수 있다. 이 때 교환 프로토콜은 지불 프로토콜의 단계 1과 단계 3을 은행과 수행하여 이루어진다. 만일 교환할 수표의 액면가가 특이한 금액이면 평범한 액면가의 여러 개의 수표로 교환할 수도 있다.

3.7 익명성 철회

고객 U		상점 S
$w \in_R Z_q^*$		
$c = H_q(C_B \parallel y_{OT}^w)$		
$\sigma = w - cb$	$\xrightarrow{c,\ \sigma}$	$c \stackrel{?}{=} H_q(C_B \parallel y_{OT}^{\sigma} B^c)$

그림 10 거래 확인서 $\mathrm{SSig}_b(C_B)$

대부분의 지불시스템은 은닉서명을 이용하여 고객의 익명성을 보장한다. 그러나 완전한 익명성이 보장되는 시스템은 돈세탁, 불법적인 지불 등과 같은 범죄 행위에 악용될 수 있다. 이러한 이유로 최근에는 고객의 익명성은 보장하되, 필요한 경우 익명성을 철회할 수 있는 조건부 익명성을 대부분 제공하고 있다[8,17]. 이 논문에서는 Solages와 Traore[8]에서의 기법을 적용하여 화폐의 불법적인 사용이 의심되는 지불에 대해 익명성을 철회할 수 있는 기능을 제공한다.

3.7.1 화폐 추적

화폐 추적은 특정 인출 과정에서 발행된 수표가 범죄에 악용되었다고 의심될 때 그것의 사용 경로를 추적하기 위해 사용되는 메커니즘이다. 이 시스템에서는 상점이 거짓 누명을 받을 경우에 결백함을 증명하기 위해서도 사용되며, 이것에 대해서는 4.2절에서 논한다. 은행은 인출 데이타베이스에서 추적하고 싶은 화폐의 CT를 신뢰기관에게 전달한다. 신뢰기관은 법원의 허가가 있으면 화폐 추적을 위한 개인키 x_{CT}를 가지고 $(CT)^{x_{CT}} = g_T^r$를 계산하여 은행에게 돌려준다. 은행은 다음과 같은 계산을 통해서 유통 후에 수표가 은행에 입금되면 이것을 식별할 수 있다.

$$C_B = y_U g_V^v (CT)^{x_{CT}} = y_U g_V^v g_T^r$$

3.7.2 인출자 추적

인출자 추적은 특정 수표로부터 그 수표의 인출자 신원을 밝혀내는 것이다. 은행은 입금 데이타베이스에서 추적하고 싶은 화폐의 OT를 신뢰기관에게 전달한다. 신뢰기관은 법원의 허가가 있으면 인출자 추적을 위한 개인키 x_{OT}를 가지고 $(OT)^{x_{OT}} = g_T^r$를 계산하여 은행에게 돌려준다. 은행은 다음과 같은 계산으로 수표 C_B의 인출자를 알아낼 수 있다.

$$y_U = C_B / (g_V^v g_T^r)$$

4. 시스템 분석

이 장에서는 새 시스템의 안전성과 원자성에 대해 분석하고, 기존 수표시스템과의 비교를 통해 새 시스템의 효율성에 대해 논한다. 또한 상점에게 수표 발행 권한을 주었을 때 발생할 수 있는 문제점을 제시하고 그것에 대한 해결책을 살펴본다.

4.1 시스템의 안전성

정리 1. G_q에서 이산대수를 계산하는 것이 어렵다고 가정하면 임의로 선택한 생성자 튜플 $(g_1,\ldots,g_l)$을 입력하였을 때, 어느 정도 성공할 수 있는 확률을 가지고 어떤 수 $y \in G_q$의 명백하지 않는 표현(non-trivial representation)[1]을 찾는 다항시간 알고리즘은 존재하지 않는다.

따름정리 1. G_q에서 이산대수를 계산하는 것이 어렵다고 가정하면 임의로 선택한 생성자 튜플$(g_1,\ldots,g_l)$을 입력하였을 때, 어느 정도 성공할 수 있는 확률을 가지고 어떤 $y \in G_q$와 그것의 서로 다른 두 가지 표현을 찾는 다항시간 알고리즘은 존재하지 않는다.

정리 1과 따름정리 1에 대한 증명은 이산대수 가정을 이용하여 쉽게 증명할 수 있다[1].

가정 1. 어떤 기저에 대한 어떤 값의 이산대수를 모를 때 그 값의 이산대수를 알고 있음을 비상호작용으로 영지식 증명을 하는 것은 계산적으로 어려우며, 이런 증명에서 확인자는 증명하는 이산대수에 대한 어떤 정보도 얻을 수 없다[14].

가정 2. 어떤 생성자 튜플에 대한 어떤 값의 표현을 모를 때 그 값의 표현을 알고 있음을 비상호작용으로 영지식 증명을 하는 것은 계산적으로 어려우며, 이런 증명에서 확인자는 증명하는 표현에 대한 어떤 정보도 얻을 수 없다[7].

정리 2. Solages와 Traore의 제한적 은닉서명에서 수신자가 프로토콜을 충실하게 수행하면 서명자는 서명한 메시지와 그것의 결과 서명에 대한 어떤 정보도 얻을 수 없다[8].

가정 3. Solages와 Traore의 제한적 은닉서명을 다항번(병행으로 또는 순차적으로) 순행하여도 이 서명을 위조하는 것은 계산적으로 어렵다[8].

가정 1, 2, 3의 정당성에 관한 논의는 이 논문의 범위를 벗어나며, 각 가정의 참고문헌을 참조하면 보다 자세한 내용을 얻을 수 있다. 정리 2는 [15]에서 Chaum과 Pedersen의 은닉서명의 은닉성을 증명한 것과 같은 방법으로 쉽게 증명할 수 있다.

정리 3. 이 시스템에서 수표를 위조하는 것은 계산적으로 어렵다.

증명. 이 시스템에서 수표는 Solages와 Traore의 제한적 은닉서명 프로토콜을 통해 발행된다. 만약 서명키를 모르는 공격자가 직접 서명을 위조하였다면 이는 기저 g_B에 대한 은행의 공개키 y_B의 이산대수를 계산하였음을 의미한다. 그러나 이것은 이산대수 가정에 위배

1) $y=1$일 때 $(0,0,\ldots,0)$을 y의 명백한 표현이라 한다.

된다. 만약 서명키를 모르는 공격자가 인출 과정을 병행으로 또는 순차적으로 여러 번 수행하여 위조하였다면 이것은 가정 3에 위배된다. 이 논리는 상점이 발행하는 거스름 수표에도 적용된다. 따라서 이 시스템에서 수표를 위조하는 것은 계산적으로 어렵다. □

정리 4. 이 시스템에서 수표를 조작하는 것은 계산적으로 어렵다.

증명. 이 시스템에서 수표는 고객 정보, 액면가 정보, 추적 정보로 구성되어 있다. 정리 3에 의해 수표를 위조할 수 없으며, 따름정리 1에 의해 C_B의 여러 표현을 알 수 없으므로 인출된 후에 수표를 조작하는 것은 계산적으로 어렵다. 따라서 인출 과정에서 조작이 가능하지 않다면 이 시스템의 수표는 조작 공격에 대해 안전한다. 그런데 수표에 포함된 정보는 같은 형태로 수표에 표시되어 있으므로 이 중 한 가지에 대한 조작 공격이 가능하면 나머지도 같은 방법으로 조작할 수 있다. 따라서 이 증명에서는 고객 정보에 대해서만 조작할 수 없음을 증명한다. 은행은 은닉서명의 입력을 스스로 만들어 사용하므로 이런 공격을 하기 위해서는 $\mathrm{ZKProof}(\log_F E = \log_{y_{CT}} CT)$ 나 제한적 은닉서명을 공격하여야 한다. 각 경우에 대해 살펴보면 다음과 같다.

- $\mathrm{ZKProof}(\log_F E = \log_{y_{CT}} CT)$에 대한 공격: 기존 E 대신에 $E' = Eg_U^x$을 전달하여 증명에 성공할 수 있으면 최종으로 얻게 되는 수표는 $C_B = g_U^{x_U} g_V^v g_T^r$이 아니라 $C_B' = g_U^{x_U + x} g_V^v g_T^r$을 얻게 된다. 공격자는 그림 5에서 $c = H_q(F \| E' \| y_{CT} \| CT \| F^w g_U^{cx} \| \delta')$를 만족하는 c를 찾아야 공격에 성공할 수 있다. 그러나 이것은 H_q의 일방향성 때문에 계산적으로 어렵다.
- BlindSig에 대한 공격: 만약 공격자가 $\log_{g_B} g_U = x$를 알고 있으면 다음과 같이 공격에 성공할 수 있다. 먼저, $r' \in Z_q$와 $x' \in Z_q$를 임의로 선택하여 은닉요소 r를 $r' + xx'$로 설정하여 사용하고, 나중에 서명 과정에서 g_B^r를 은닉요소로 사용하지 않고 $g_B^{r'}$을 사용하면 결과 수표는 $C_B' = g_U^{x_U + x'} g_V^v g_T^r$이 된다. 그러나 $\log_{g_B} g_U$를 계산하는 것은 이산대수 가정에 의해 계산적으로 어려우므로 이런 공격은 가능하지 않다.

따라서 이 시스템에서 수표를 조작하는 것은 계산적으로 어렵다. □

정리 5. 이 시스템에서 은행이 발행된 수표를 추적하는 것은 계산적으로 어려우며, 이것은 상점과 공모를 하여도 마찬가지이다. 뿐만 아니라 같은 고객이 인출한 두 수표를 서로 연결하거나 지불한 수표와 그 과정에서 받은 거스름 수표를 서로 연결하는 것도 계산적으로 어렵다.

증명. 정리 2에 의해 은행은 서명 과정에서 얻은 정보로부터 메시지나 서명 결과에 대해 어떤 정보도 얻을 수 없다. 뿐만 아니라 이산대수 가정과 정리 1에 의해 은닉서명 전에 수신한 E, CT, $\mathrm{ZKProof}(\log_F E = \log_{y_{CT}} CT)$로부터 나중에 C_B를 식별할 수 있는 어떤 정보도 얻을 수 없다. 또한 은닉서명에 포함한 A, B, OT를 보지 못하므로 이 정보를 이용하여 수표를 추적할 수 없다. 뿐만 아니라 익명으로 이루어지면 상점은 지불자에 대한 어떤 정보도 얻지 못하므로 은행은 상점과 공모를 하여도 지불에 사용된 수표의 인출자를 알 수 없다. 만일 특이한 액면가의 수표를 인출 받는다면 액면가 정보를 이용하여 인출자를 추적할 수 있으므로 수표의 액면가를 몇 가지로 제한하는 것이 바람직하다.

수표를 구성하는 정보 중 신원 정보를 제외한 나머지 정보는 수표를 인출할 때마다 임의로 선택되므로 같은 고객의 두 수표를 연결하기 위해 사용할 수 없다. 정리 4에 의해 고객은 수표를 조작할 수 없으므로 같은 고객이 인출한 수표에는 항상 같은 신원 정보가 포함된다. 그러나 앞서 논한 바와 같이 인출 과정과 그 과정에서 발행된 수표를 연관할 수 없으며, 정리 1에 의해 수표의 표현을 얻는 것은 계산적으로 어려우므로 다른 인출 과정에서 인출된 같은 고객의 수표를 서로 연관할 수 없다.

상점은 은닉서명을 통해서 수표를 발행하기 때문에 지불에 사용한 수표와 거스름 수표를 연결하는 것은 어렵다. 이렇게 발행되므로 거스름 수표를 익명으로 다른 상점에게 지불하면 상점끼리 공모를 하여도 지불 수표와 거스름 수표를 연관할 수 없으며, 나중에 은행에 입금되어도 상점과 공모를 통해 은행이 거스름 수표와 지불 수표를 연관할 수 없다. 수표의 액면가가 여러 종류이므로 특정한 금액의 거스름 수표가 발행되어도 액면가 정보를 이용하여 거스름 수표와 그것의 지불 수표를 연관하는 것 역시 어렵다. 다만, 이것이 보장되기 위해서는 충분한 양의 수표가 유통되어야 한다. □

보조정리 1. 이 시스템에서 수표 C_B(또는 C_S)를 지불하기 위해서는 생성자 튜플 (g_U, g_V, g_T)에 대한 C_B(또는 C_S)의 표현을 알아야 한다.

증명. 지불 과정에서 고객은 C_B를 전달한 다음에 C_B의 표현 중 g_V에 해당하는 값은 공개하고 나머지 색인에 대해서는 표현에 관한 영지식 증명을 비상호작용으로 한다. 가정 2에 의해 C_B의 표현을 모르면 할 수 없으므로 이 보조정리는 성립한다. □

정리 6. 이 시스템에서 수표를 이중사용하면 은행은 그 사실을 발견할 수 있을 뿐만 아니라 이중사용한 고객을 밝혀낼 수 있다. 하지만 은행이 이 권한을 남용하여 고객에게 이중사용에 대해 누명을 씌우는 것은 계산적으로 어렵다.

증명. 은행은 정상적으로 입금된 지불 트랜스크립트를 입금 데이타베이스에 저장하므로 이중사용되면 그 사실을 알 수 있다. 이 시스템도 입금된 수표가 이미 입금 데이타베이스에 있는지 검사하여 이중사용을 판단한다. 이 시스템에서는 C_B 뿐만 아니라 A, B, OT 값도 같아야 같은 수표로 간주한다. 그러므로 서로 다른 고객이 인출한 수표가 우연히 일치하여 이중사용으로 오인될 확률은 매우 적다. 이 시스템에서도 정리 1에 의해 고객을 제외한 어느 누구도 수표의 표현을 알 수 없으므로 보조정리 1에 의해 상점은 수표를 스스로 사용할 수 없고, 고객에게 누명을 씌울 수도 없다. 지불할 때마다 상점은 매번 다른 T_S 값을 주므로 고객은 같은 상점에게 같은 트랜스크립트가 만들어지도록 두 번 지불할 수도 없다. 고객은 수표의 표현을 알고 있음을 영지식으로 증명할 때 인출 과정에서 수표에 포함한 A의 표현을 이용하여야 한다. 만약 고객이 A의 여러 표현을 알거나 수표의 여러 표현을 알면 이중사용하여도 자신의 신원이 드러나지 않도록 할 수 있다. 그러나 따름정리 1에 의해 어떤 값의 서로 다른 여러 개의 표현을 알 수 없으므로 이런 방법으로 공격하는 것은 계산적으로 어렵다. 은행은 정리 1에 의해 지불받은 수표의 표현을 알 수 없으므로 보조정리 1에 의해 고객에게 이중사용하였다고 누명을 씌울 수 없다. □

정리 7. 이 시스템에서 상점이 부정을 하지 않으면 고객은 지불 과정에서 받는 거스름 수표를 조작할 수 없다.

증명. 거스름 수표의 발행 과정은 $\mathrm{ZKProof}_{\mathrm{REP}}(C/C' = g_T^{r} g_D^{-t})$과 제한적 은닉서명 때 은닉요소를 하나 더 제거한다는 것을 제외하고는 인출 과정과 같다. 변형된 제한적 은닉서명에서도 정리 4와 같은 논리에 의해 기저 g_D에 대한 수표를 구성하는 생성자들에 대한 이산대수를 모르면 공격할 수 없다. 따라서 정리 4와 변형된 은닉서명의 안전성에 의해 거스름 수표도 $\mathrm{ZKProof}_{\mathrm{REP}}(C/C' = g_T^{r} g_D^{-t})$을 공격할 수 없으면 조작할 수 없다. 고객은 $\mathrm{ZKProof}_{\mathrm{REP}}(C/C' = g_T^{r} g_D^{-t})$을 통해 자신의 공개신원정보 y_U를 은닉한 C'에 포함된 신원 정보와 지불한 수표에 포함된 신원 정보가 같음을 영지식으로 증명한다. 이 증명은 표현에 관한 영지식 증명이다. 고객은 다른 고객이 인출한 수표의 표현을 알 수 없으므로 정리 4, 보조정리 1, 가정 2에 의해 이 과정을 공격하여 자신의 신원 정보가 아닌 다른 값으로 만든 C'을 이용하여 증명에 성공할 수 없다. 따라서 상점과 공모를 하지 않는 한 고객은 거스름 수표를 조작할 수 없다. □

정리 8. 이 시스템에서는 수표를 인출한 고객만이 그 수표를 환불받을 수 있으며, 사용한 후에 환불받거나 환불한 후에 사용할 수 없다. 또한 수표를 이중으로 환불받을 수도 없다.

증명. 수표에 포함되어 있는 고객 정보를 은행이 확인함으로 제3자가 전송되는 메시지를 공격하여 다른 사람을 대신하여 환불받을 수 없다. 환불을 먼저 받은 후에 다시 사용하거나 사용한 후에 환불받는 것은 이중사용하는 것과 같다. 더욱이 환불은 실명으로 이루어지는 것이므로 이렇게 한 고객을 식별하는 것은 쉽다. 환불은 실명으로 이루어지므로 환불이 된 수표를 기록해 놓으면 이중환불을 쉽게 발견할 수 있으며, 그렇게 하는 고객을 쉽게 식별할 수 있다. 고객은 수표의 액면가만큼을 환불하게 되며 정리 4에 의해 수표의 액면가 정보를 변경할 수 없으므로 받아야 하는 금액이상으로 환불받을 수 없다. □

4.2 거스름 수표의 안전성

상점이 발행하는 거스름 수표는 유통된 후에 발행한 상점에게 액면가만큼 청구되는 후불 방식이므로 가짜 수표를 만들어 사용하더라도 상점은 별다른 이득을 얻을 수 없다. 그런데 이렇게 만든 가짜 수표를 여러 번 사용할 수 있으면 상점은 불법적인 이득을 얻을 수 있다. 예를 들어 그림 11처럼 상점 1이 임의의 $x \in Z_q^*$가 포함된 C_S를 만들어 이것을 이중으로 사용하였다고 하자. 이 경우 은행은 신원 정보 x를 계산할 수 있지만 x는 존재하지 않는 거짓 신원 정보일 확률이 매우 높다. 이 시스템에서는 이중사용된 유효한 두 지불 트랜스크립트를 이용하여 계산한 인출자 정보가 거짓 정보이면 그 수표를 발행한 상점에게 그 책임을 묻는다. 이것은 오직 상점 1만이 유효한 C_S를 만들 수 있으며, 고객은 상점이 발행한 수표를 조작할 수 없기 때문이다. 그

러나 이것으로 이런 공격을 모두 방어할 수 없다.

그림 12에서처럼 C_S를 상점 2에게 지불하고 그 상점이 발행하는 거스름 수표 C'_S를 여러 번 사용하는 경우에는 전혀 잘못이 없는 상점 2가 부정한 것으로 간주된다. 이것은 상점 2가 C_S가 위조된 것임을 알지 못해서 거짓 신원 정보 x가 포함된 거스름 수표를 발행해 주었기 때문이다. 이런 경우, 상점 2는 자신의 결백을 증명할 수 있어야 한다. 이것을 위해 상점 2는 이중사용된 거스름 수표와 같은 액면가를 가진 수표 정보를 데이타베이스에서 모두 찾는다. 이 때, 다른 상점이 발행한 수표를 지불 받아 거스름을 발행한 것만 찾는다. 다음과 같이 총 n개의 수표 정보를 찾았다고 하자.

$$(v',CT_1,C_{S_1}),\ \ldots\ ,(v',CT_i,C_{S_i}),\ \ldots\ ,(v',CT_n,C_{S_n})$$

여기서 $i \neq j$인 S_i와 S_j는 같을 수 있다. 상점은 문제가 된 C'_S과 은행이 이중사용 검출작업을 통해 알아낸 x, 액면가 v', 데이타베이스에서 찾은 $CT_1, \ldots, CT_n$을 신뢰기관에게 전송하고 화폐 추적을 요청한다. 신뢰기관은 개인키 x_{CT}를 이용해서 $(CT_i)^{x_{CT}} = g_T^{r_i}$을 계산하고 C'_S과 $g_U^x g_V^{v'} g_T^{r_i}$을 비교한다. 만약 일치하는 것을 찾으면 S_i가 S'보다 먼저 x를 신원 정보로 사용한 거스름 수표를 발행한 것이 된다. 만일 S_i도 결백하면 같은 방법으로 입증한다. 결국에는 이러한 화폐 추적을 거쳐서 부정을 한 상점을 밝혀낼 수 있다.

이런 결백 입증 과정이 효율적이지는 못하지만 이런 방법이 존재하는 것만으로 많은 상점은 이와 같은 부정을 행하지 않을 것이다. 또한 상점이 유지하여야 하는 데이타베이스의 크기는 은행이 유지하는 데이타베이스의 크기와 마찬가지로 수표의 유효기간을 이용하여 적당한 크기로 유지할 수 있다. 수표의 유효기간은 기존 수표의 형태에서 하나의 생성자를 더 사용하여 쉽게 나타낼 수 있다.

상점이 스스로 수표를 위조하지 않고 지불 프로토콜에서 신원 증명 과정을 생략하여 고객에게 가짜 신원 정보가 포함된 거스름 수표를 발행 받도록 할 수도 있다. 그러나 상점은 이런 부정으로 아무런 이득을 얻을 수 없으므로 이러한 부정은 하지 않을 것이다.

4.3 원자성

지불시스템에서 원자성(atomicity)이란 네트워크나 시스템의 오류, 참여자의 부정이나 부인 등으로 인해 다른 참여자가 손해 보는 일이 없도록 하는 것이다. 여기에서 은행은 부정을 하지 않는다고 가정한다. 새 시스템에서는 [18]에서 제안한 기법을 적용하여 시스템의 원자성을 보장한다. 참여자들은 프로토콜이 중단되어 문제가 발생하면 해당 프로토콜을 다시 진행하여 복구를 시도한다. 이를 복구 프로토콜(recovery protocol)이라 한다. 고객

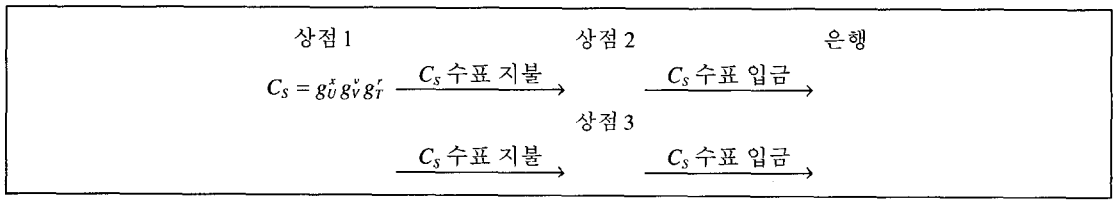

그림 11 상점의 공격 방법 1

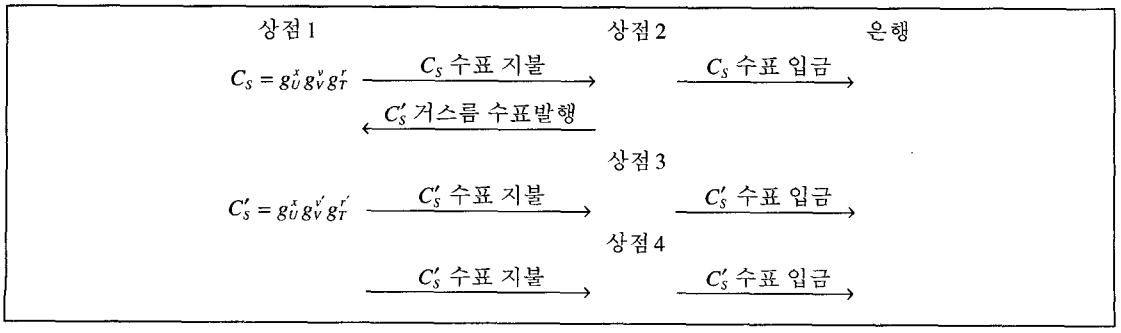

그림 12 상점의 공격 방법 2

은 프로토콜이 중간에 중단되면 다시 은행이나 상점에게 복구를 시도할 수 있지만 반대의 경우는 어렵다. 따라서 이것을 고려하여 프로토콜을 설계하여야 한다. 이런 이유에서 상품을 암호화하여 전달하고 나중에 복호화키를 주는 것이다. 손해를 볼 수 있는 상황에서 복구가 되지 않으면 신뢰기관에게 중재를 요청하여 문제를 해결하는 분쟁 해결 프로토콜(resolve protocol)을 수행한다.

4.3.1 인출 프로토콜의 원자성

인출 프로토콜에서 고객은 은행이 발행한 수표를 전달받지 못할 수 있다. 혹은, 정상적으로 프로토콜을 진행하여 유효한 수표를 받았지만 고객이 이것을 부인할 수 있다. 고객이 오류가 발생하였다고 주장하면 그 주장의 진실성 여부와 상관없이 이미 발행한 수표에 대해서는 화폐 추적을 하여 그 수표를 블랙리스트에 등록하여 사용할 수 없도록 만들고, 새 수표를 발행하여 주거나 환불해 준다.

4.3.2 지불 프로토콜의 원자성

단계 1/단계 2 진행 후: 상점이 이 단계까지 진행하고 중단하면 고객은 그 수표를 다른 지불에 사용할 수 없다. 다른 상점에게 지불할 경우, 같은 수표를 받은 두 상점이 공모하면 단계 1에서의 도전, 응답값을 이용하여 고객의 신원 정보를 알 수 있기 때문이다. 이러한 문제는 도전-응답 프로토콜을 이용하여 이중 사용을 검출하는 기존 시스템에서도 존재한다[1,7,8]. 그래서 이와 같은 문제가 발생하면 고객은 환불 프로토콜을 진행하여 수표의 금액만큼 되돌려 받아야 한다. 상점은 거래 확인서가 없으므로 이 상태에서는 수표를 입금할 수 없다.

단계 3 진행 후: 거스름을 받은 후에 고객이 이 단계에서 중단하면 상품을 얻을 수는 없지만 합법적인 거스름은 얻을 수 있다. 또한 지불에 사용한 수표는 환불 또는 교환하면 추가적인 이득을 얻을 수 있다. 상점은 이 단계에서 종료되면 지불 받은 수표를 입금도 하지 못하고 거스름만 준 상태가 된다. 그러므로 상점은 고의로 이 과정에서 중단을 하지 않을 것이며, 올바른 거스름을 줄 것이다. 반면 고객이 고의로 중단한 경우에 상점은 단계 1과 단계 3의 트랜스크립트를 신뢰기관에 전달하여 중재요청을 하여야 한다. 그러면 화폐 추적을 통해 고객에게 발행한 거스름 수표를 알아내어 블랙리스트에 등록하여 그 수표를 다시 사용할 수 없도록 하고, 지불받은 수표에 대해서는 인출자 추적을 하여 고의로 중단한 고객을 밝혀내어 책임을 추궁한다.

단계 4 진행 후: 이 단계에서 상점이 중단하면 고객은 상품복호화키를 받지 못한다. 이 때 고객은 신뢰기관에게 지불한 수표를 제시하고 그것의 소유자임을 입증한다. 신뢰기관은 은행을 통해 해당 수표가 입금되었는지 확인한다. 만일 입금된 후라면 상점에게 상품복호화키를 주도록 지시하고 상점이 이러한 지시에 응하지 않을 경우에는 법적인 책임을 받게 한다. 반대로 입금 전이면 입금을 거부하도록 하고 상점에 접촉하여 문제의 원인을 알아내고 적절한 조치를 취한다.

단계 5 진행 후: 상점이 보내준 상품이 요청한 것이 아니거나 복호화가 제대로 되지 않는 경우도 있을 수 있다. 고객은 상점이 전송한 영수증을 가지고 있으므로 이것을 증거로 제시하여 상점의 부정을 입증할 수 있다. 영수증은 상점의 개인키로 서명이 되어 있으므로 상점은 부인할 수 없고 고객은 상점의 개인키를 알지 못하므로 거짓 영수증을 만들어 허위 요청을 할 수 없다.

4.3.3 입금/환불 프로토콜의 원자성

입금 프로토콜이 진행 중에 중단되면 고객은 은행에게 필요한 정보를 보내고 다시 입금 요청을 한다. 은행은 요청한 수표가 이미 입금 처리되었는지 여부를 확인한다. 아직 입금이 안 된 수표이면 해당 상점의 계좌로 수표 금액을 입금한다. 입금 된 수표인 경우에는 이미 입금되었음을 통보하여 이중으로 지급되지 않도록 한다. 환불 프로토콜의 원자성 역시 입금 프로토콜에서와 동일하다.

4.4 기존 오프라인 수표시스템과의 비교

기존의 Chaum 등[5]의 시스템과 Hirschfeld[6]의 시스템은 인출 과정에서 cut-and-choose 방식을 사용하므로 계산량이 많고, 프로토콜 수행 과정에서 많은 정보가 교환된다. 새 시스템은 제한적 은닉서명을 이용하여 수표를 인출하므로 [5]와 [6]의 시스템보다 계산량과 정보교환량 측면에서 효율적이다.

Brands[7]의 시스템과 Solages와 Traore[8]의 시스템은 둘 다 $2^{l}-1$의 액면가를 표현하기 위해 길이가 l인 생성자 튜플을 사용한다. 이에 비해 새 시스템에서는 액면가를 표현하기 위해 단일 생성자를 사용한다. 따라서 새 시스템은 기존 시스템보다 수표의 형태가 간단하며, 지불 과정을 제외한 나머지 과정에서 소요되는 계산량과 정보교환량이 적다. 대부분의 기존 시스템[5-8]은 고정된 액면가의 수표만을 인출할 수 있으며, 다른 액면가를 가진 수표를 만들려면 생성자의 수를 늘려야 한다. 반면 액면가를 표현하기 위해 하나의 생성자만 사용하는 새 시스템은 액수에 제약이 없어서 다양한 액면가를 쉽게 제공할 수 있다. 이 특성과 거스름을 지불에 사용할 수 있는 특성 때문에 기존 시스템에 있는 고객의 익명성에 나쁜 영향을 주는 지불액과 환불액 간에 보수관

계가 새 시스템에는 없다. 따라서 새 시스템은 기존 시스템에 비해 고객의 익명성이 크게 향상되었다.

새 시스템은 지불 과정에서 발생한 거스름을 상점이 발행하며, 이 거스름을 지불에 사용할 수 있다. 하지만 기존 시스템에는 없는 거스름 수표 발행 과정이 있어 지불 비용은 오히려 기존 시스템이 우수할 수 있다. 그러나 거스름이 필요 없는 경우를 고려하면 새 시스템이 더 우수하다. 또한 기존 시스템에서는 남은 잔액을 환불받아야 하지만 새 시스템은 새롭게 인출하지 않고 남은 잔액을 다른 지불에 사용할 수 있으므로 이런 관점에서는 새 시스템이 더 효율적일 수도 있다. 예를 들어 1500원인 수표를 700원과 800원으로 지불하고자 하면 기존 시스템에서는 1500원짜리 수표를 두 개 인출하고, 이것을 각각 700원과 800원으로 지불하여야 하고, 그 다음에 800원과 700원을 환불받아야 한다. 반면에 새 시스템에서는 1500원짜리 수표를 하나 인출하여 700원 지불하고, 상점으로부터 받은 800원짜리 거스름 수표를 다시 지불하면 된다. 그러나 모든 경우에 새 시스템이 좋은 것은 아니다. 새 시스템은 지불 과정에 거스름 수표 발행 과정이 있어 지불 비용이 많다는 단점이 있지만 거스름의 재사용성, 액면가 표현방법 등, 이 단점을 상쇄할 수 있는 여러 장점을 지니고 있다.

5. 결 론

수표시스템에서는 수표의 액면가와 지불대금이 항상 일치하지 않기 때문에 거스름이 발생한다. 기존의 오프라인 수표시스템은 이러한 거스름 처리를 위해 수표를 두 부분으로 나누어, 한 부분은 지불에 사용하고 다른 한 부분은 환불을 받기 위해 사용한다. 그래서 거스름이 발생하는 경우는 반드시 환불 프로토콜을 진행하여 돌려 받아야 한다. 이 논문에서는 거스름 수표를 상점이 발행하도록 하여 받은 거스름을 지불에 사용할 수 있는 편리한 오프라인 수표시스템을 제안하였다. 상점이 거스름 수표를 발행하도록 하면 기존에 사용하던 이중구조의 수표를 사용할 필요가 없어 수표의 형태가 매우 간단하며 고객이 원하는 액면가의 수표를 인출 받을 수 있다. 따라서 지불액과 환불액 간에 보수 관계를 통해 지불과 환불의 연결이 가능하였던 기존 시스템에 비해 고객의 익명성도 향상되었다. 물론 지불 과정에서 기존에 없는 거스름 발행 과정이 추가되어 지불 비용이 증가하였지만 이 부분을 제외하고는 정보교환량이나 계산량 측면에서 기존보다 효율적이다. 새 시스템은 또한 트랜잭션의 원자성을 고려하여 프로토콜을 설계하였으며, 익명 전자화폐의 불법적인 사용을 막기 위해 화폐 추적, 인출자 추적이 가능한 조건부 익명성을 제공하였다.

이 논문의 가장 큰 특징은 상점이 거스름을 발행하는 것이다. 오프라인 방식에서는 은행이 지불 프로토콜에 참여하지 않기 때문에 화폐로서 통용될 수 있는 거스름을 발행하는 것이 어렵다. 그래서 상점이 거스름을 발행하도록 하여 은행의 역할을 대신하도록 하였다. 이 경우 상점이 발행한 수표에 대해서는 후불방식이다. 새 시스템은 조건부 익명성을 제공하기 때문에 상점이 불법적인 수표를 발행하여 이득을 얻을 수 없다. 또한 새 시스템에서는 수표가 이중사용 된 경우 그 사용자를 밝혀낼 수 있으며, 트랜잭션의 원자성을 보장하는 등 지불 프로토콜에서 만족되어야 하는 기본적인 안전성을 보장한다.

새 시스템에서는 거스름 수표를 상점이 발행하도록 허용하기 때문에 이에 대한 신뢰는 상점의 신용에 많이 의존하며, 시스템의 수용성 측면에서 널리 받아들여지기 위해서는 정책적인 지원이 필요하다. 특히 대량의 거스름 수표를 발행하고 그것에 대한 채무를 이행하지 않는 불량 상점이 발생할 수 있으므로 상점의 신용을 제한하는 효율적인 방법이 요구된다. 그러나 시스템의 효율에 나쁜 영향을 미치지 않으면서 신용을 제한할 수 있는 방법을 아직 찾지 못하였으며, 앞으로 이에 대한 연구가 필요하다. 만일 상점이 수표를 발행하지 않고도 거스름을 지불에 사용할 수 있는 거스름 메커니즘이 개발된다면, 후불방식의 문제점도 없는 더욱 효율적인 오프라인 지불 방식이 될 것이다.

참 고 문 헌

[1] Brands, S., *"Untraceable Off-line Cash in Wallets with Observers," Advances in Cryptology, Crypto 1993,* Springer Verlag, LNCS 773, pp. 302-318, Aug. 1993.

[2] Chan, A.H., Frankel, Y., and Tsiounis, Y., *"Easy Come-Easy Go Divisible Cash," Advances in Cryptology, Eurocrypt 1998,* Springer Verlag, LNCS 1403, pp. 561-575, May 1998.

[3] Nakanishi, T. and Sugiyama, Y., *"Unlinkable Divisible Electronic Cash," Proc. of the 3rd Int. Workshop on Information Security, ISW 2000,* Springer Verlag, LNCS 1975, pp. 121-134, Dec. 2000.

[4] Jakobsson, M. and Yung, M., *"Revokable and Versatile Electronic Money," Proc. of the 3rd ACM Conf. on Computer and Communications Security,* pp. 76-87, Mar. 1996.

[5] Chaum, D., den Boer, B., van Heyst, E., Mjoelsnes, S.F., and Steenbeek, A., *"Efficient Offline Electronic Checks," Advances in Cryptology, Eurocrypt 1989,* Springer Verlag, LNCS 434, pp. 294-301, Apr. 1989.

[6] Hirschfeld, R., *"Making Electronic Refunds Safer,"*

Advances in Cryptology, Crypto 1992, Springer Verlag, LNCS 740, pp. 106-112, Aug. 1992.

[7] Brands, S., "An Efficient Off-Line Electronic Cash System based on the Representation Problem," CWI(Centrum voor Wiskunde en Informatica) Technical Report, CS-R9323, 1993.

[8] de Solages, A. and Traore, J., *"An Efficient Fair Off-Line Electronic Cash System with Extensions to Checks and Wallets with Observers," Proc. of the 2nd Int. Conf. on Financial Cryptography, FC 1998*, Springer Verlag, LNCS 1465, pp. 275-295, Feb. 1998.

[9] Chaum, D., *"Online Cash Checks," Advances in Cryptology, Eurocrypt 1989*, Springer Verlag, LNCS 434, pp. 288-293, Apr. 1989.

[10] Deng, R.H., Han, Y., Jeng, A.B., and Ngair, T., *"A New On-Line Cash Check Scheme," Proc. of the 4th ACM Conf. on Computer and Communications Security*, pp. 111-116, Apr. 1997.

[11] Kim, S. and Oh, H., *"A New Electronic Check System with Reusable Refunds,"* Int. J. of Information Security, Vol. 1, No. 3, pp. 175-188, Springer Verlag, 2002.

[12] 김상진, 오희국, "남은 금액을 재사용할 수 있는 오프라인 전자수표시스템," 한국정보보호학회 논문지, 제11권, 제6호, pp. 27-40, 2001년 12월.

[13] Abe, M. and Camenisch, J., *"Partially Blind Signature Schemes," Proc. of 1997 Symp. on Cryptography and Information Security Workshop*, 1997.

[14] Schnorr, C.P., *"Efficient Signature Generation by Smart Cards," J. of Cryptology*, Vol. 4, No. 3, pp. 161-174, Nov. 1991.

[15] Chaum, D. and Pedersen, T.P., *"Wallet Databases with Observers," Advances in Cryptology, Crypto 1992*, Springer Verlag, LNCS 740, pp. 89-105, Aug. 1992.

[16] Diffie, W. and Hellman, M.E., *"New Directions in Cryptography," IEEE Tran. on Information Theory*, Vol. 10, No. 6, pp. 644-654, Nov. 1976.

[17] Frankel, Y., Tsiounis, Y., and Yung, M., *"Fair Off-Line e-cash Made Easy," Advances in Cryptology, Asiacrypt 1998*, Springer Verlag, LNCS 1514, pp. 257-270, Oct. 1998.

[18] Xu, S., Yung, M., Zhang, G., and Zhu, H., *"Money Conservation via Atomicity in Fair Off-Line E-Cash," Proc. of the 2th Int. Workshop on Information Security, ISW 1999*, Springer Verlag, LNCS 1729, pp. 14-31, Nov. 1999.

김 상 진
정보과학회논문지 : 정보통신
제 30 권 제 2 호 참조

최 이 화
2000년 2월 한양대학교 전자계산학과 졸업
2002년 2월 한양대학교 컴퓨터공학과 석사
2002년 1월~현재 (주)소프트포럼 연구원
관심분야는 정보보호, 전자화폐, 네트워크 보안, PC 보안

오 희 국
정보과학회논문지 : 정보통신
제 30 권 제 2 호 참조