情報保護學會論文誌
第13券 第2號, 2003. 4

확인 가능한 암호기법을 사용한 지불의 원자성 보장 방법

최 형 섭*, 김 상 진**, 오 희 국***

Providing Payment Atomicity Using Verifiable Encryption

Hyungsup Choi, Sangjin Kim, Heekuck Oh

요 약

확인 가능한 암호화 기법(verifiable encryption)은 암호문을 해독하지 못해도 어떤 것이 암호화되어 있는지 확인할 수 있는 기법이다. 이 기법은 공정한 교환(fair exchange)에서 암호화된 물건을 먼저 상대방에게 제시하여 상대방이 그것을 받을 수 있다는 확신을 갖도록 하는데 사용된다. 그러나 지금까지의 공정한 교환은 참여자의 익명성을 고려하지 않았다. 이 논문에서는 표현문제(representation problem)를 사용하는 익명의 화폐시스템에 이러한 확인 가능한 암호화 기법을 적용하여, 지불의 원자성을 제공하는 방법을 제안한다. 이 방법은 기존의 방법과는 달리 분쟁을 해결할 때 신뢰기관이 은행으로부터 상점의 입금여부를 확인할 필요가 없어서 효율적이다. 또한 상점의 입금시한이 없으며, 고객과 상점 모두 시간적인 제약 없이 분쟁해결을 요청할 수 있는 유연한 방식이다. 반면에 지불과정에서 확인 가능한 암호화를 적용하기 위한 증명이 요구되므로 추가비용이 들어간다. 새 시스템의 안전성과 기존의 원자성 보장 방법과의 비교분석도 논의한다.

ABSTRACT

Verifiable encryption is an encryption technique with which one can verify what has been encrypted even if one can not decrypt the ciphertext. This technique can be used in fair exchange to convince the counterpart of his or her receiving an item by presenting an encrypted form in advance. In this paper, a method that can guarantee the payment atomicity is proposed by applying verifiable encryption to an electronic cash system based on the representation problem. With the new method, the process of dispute settlement is improved in the fact that the trusted third party do not have to interact with the bank to resolve disputes. This method is also flexible in a sense that clients and shops can request for dispute settlement regardless of any deadline constraint. However, additional proof is necessary to apply verifiable encryption during payment. We discuss the security and the atomicity of our method, and compare ours with others.

Keyword : *Payment atomicity, Fair Exchange, Verifiable Encryption*

Ⅰ. 서 론

전자화폐에서 지불과정은 고객이 상점에게 대금을 지불하고, 상점이 고객에게 상품을 전달하는 두 단계로 이루어진다. 네트워크를 통한 전자상거래에서 고객과 상점은 서로를 신뢰하지 않는 관계이다. 언제 네트워크가 끊어질지 모르며 누군가가 고의로 종료할 수 있다. 따라서 고객과 상점 간에 이루어지는 지

* (주)위세아이텍(hschoi@wise.co.kr)
** 한국기술교육대학교 인터넷미디어공학부(sangjin@kut.ac.kr)
*** 한양대학교 컴퓨터공학과(hkoh@cse.hanyang.ac.kr)

불과정은 원자성(atomicity)이 보장되어야 한다. 만약 원자성을 보장하지 않으면 고객이 대금을 지불하고도 상품을 받지 못하거나 상점이 상품을 주고도 대금을 받지 못할 수 있다. 지불과정에서 원자성을 보장하는 것을 지불의 원자성이라 하며, Tygar가 처음 이 문제를 제기하였다[1]. 그러나 Tygar가 제안한 방법은 신뢰기관이 항상 지불에 참여해야하는 방식이었다. 이후에 Boyd와 Foo는 문제가 발생했을 때만 신뢰기관이 참여하는 낙관적인(optimistic) 접근방법으로 지불의 원자성을 제공하였다[2]. 이 방법은 전환가능한 서명(convertible signature)을 사용하므로 익명성을 제공하지 못한다. Xu 등은 익명성을 제공하고 낙관적인 접근을 사용해 오프라인 화폐에 적합한 지불의 원자성 보장 방법을 제안하였다[3].

지불의 원자성에 관한 직접적인 연구 외에 공정한 교환(fair exchange)이라는 연구 분야가 있다[4,5]. 공정한 교환은 네트워크상의 두 참여자가 서로의 물건을 교환할 때, 서로가 손해보지 않는다는 것을 보장하는 교환방식이다. 공정한 교환은 원자성을 제공한다는 측면에서 지불의 원자성과 유사한 성격을 가지기 때문에 많은 사람들이 공정한 교환으로 쉽게 지불의 원자성을 제공할 수 있을 것으로 생각한다. 하지만 지금까지의 공정한 교환은 참여자의 익명성을 고려하지 않았다. 또한 그 연구가 주로 참여자 자신의 서명을 공정하게 교환하는 데 집중되어 있다[5,6]. 따라서 은행이 서명하여 발행한 화폐와 디지털 상품을 교환하는 지불과정에 적용하기가 쉽지 않다.

공정한 교환에는 확인 가능한 암호화(verifiable encryption)를 사용하는 방법이 있다[5,6]. 확인 가능한 암호화는 어떤 지정된 다른 사람의 공개키로 메시지를 암호화한 다음에 그 암호문에 무엇이 암호화되어있는지 그것을 해독할 수 없는 참여자가 확인할 수 있도록 해주는 암호기법이다. 공정한 교환에서는 이 기법을 사용하여 신뢰기관의 공개키로 물건을 암호화하여 상대방에게 전달한다. 이렇게 하면 암호문을 받은 사람은 암호문을 해독할 수 없어도 어떤 물건이 암호화되어 있는지 확인할 수 있다. 또한 이 암호문은 신뢰기관이 해독할 수 있으므로 물건을 받을 수 있다는 확신을 갖게 된다. 만약 물건을 받지 못하면 신뢰기관에게 해독을 요청하여 물건을 받을 수 있다.

현재의 화폐시스템은 대부분 고객의 익명을 보장하는 오프라인 방식이다. 이런 화폐시스템에서 지불의 원자성을 보장하는 방법은 다음의 세 가지를 고려해야 한다.

- 고객은 익명으로 지불에 참여한다.
- 분쟁이 발생했을 경우에만 신뢰기관이 참여하도록 프로토콜을 구성해야 한다.
- 고객과 상점 이외에 제 3자가 상품을 얻을 수 없어야 한다.

이 논문에서는 표현문제(representation problem)[7]를 사용하는 화폐시스템에 효율적인 확인 가능한 암호화 기법을 적용하여 지불의 원자성을 제공하는 방법을 제안한다. 사용하는 확인 가능한 암호화 기법은 Naccache 공개키 시스템[8]과 다른 군에서의 이산대수 등가 증명을 사용한다. 기존의 다른 군에서의 이산대수 등가 증명[9]은 보통 범위증명[10]까지 같이 해야했다. 하지만 여기서는 범위증명이 필요 없고 이산대수가 가질 수 있는 모든 범위에서 증명이 가능하다. 따라서 지불과정에서 확인 가능한 암호화를 적용하기 위한 증명이 추가로 필요하지만 그 비용이 많지 않다. 제안하는 원자성 보장 방법은 기존의 방법과는 달리 분쟁을 해결할 때 신뢰기관이 은행으로부터 상점의 입금여부를 확인할 필요가 없다. 또한 상점의 입금시한이 없어 고객과 상점 모두 시간적인 제약 없이 분쟁해결을 요청할 수 있는 유연한 방식이다.

이 논문의 구성은 다음과 같다. 2장에서는 가정하는 전자화폐와 기존의 원자성 보장 방법에 대해 설명하고, 3장에서는 이 논문의 이해를 돕기 위한 수학적인 배경을 설명한다. 4장에서는 확인 가능한 암호화 기법과 이를 이용해서 지불의 원자성을 보장하는 방법을 제안한다. 5장에서는 제안하는 방법의 원자성과 안전성을 분석하고, 다른 방법과 비교한다. 마지막으로 6장에서는 결론과 향후 연구방향에 대해 서술한다.

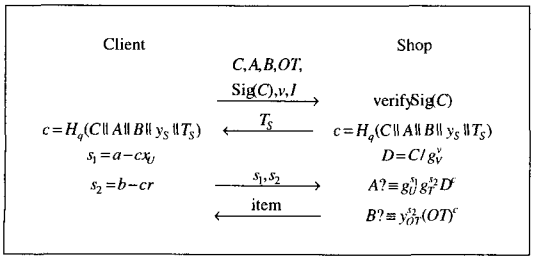
II. 관련 연구

이 논문에서 사용되는 대부분의 수학 연산은 G_q 군을 사용한다. p가 큰 소수이고, q는 $q|p-1$인 큰 소수일 때 G_q군은 곱셈군 Z_p^*의 위수(order)가 q인 부분군(subgroup)이다. 따라서 지수요소와 관련된 연산은 법 q에서 이루어지고, 일반적인 연산은 모두 법 p에서 이루어진다. 이후 논문에서는 법 p와 q에 대한 연산 표기는 생략하고 이와 다른 군에서의 연산에만 법을 표기한다. 또한 나머지 연산을 하지 않을 경우 이를 명시하기 위해 (in Z) 로 표기한다. 정수의 범위에 관련된 표기로 범위 (a, b)는 열린 구간으로 a보다 크고 b보다 작은 범위를 의미한다.

2.1 화폐시스템

이 논문에서 가정하는 화폐시스템은 기본적으로 Solages와 Traore의 화폐[11]와 같다. 이 화폐는 표현문제를 사용하여 동전을 $C=g_U^{x_U}g_V^{v}g_T^{r}$과 같이 구성한다. 여기서 g_U, g_V, g_T는 G_q군의 생성자(generator)이고, x_U는 고객의 비밀신원정보이고, v는 화폐의 액면가이며, r은 고객이 사용한 은닉요소이다. r은 화폐추적과 인출자추적 기능을 제공하기 위해 사용된다. 고객은 제한적 은닉서명(restricted blind signature) 프로토콜을 수행하여 화폐를 인출한다. 화폐는 $A=g_U^{a}g_T^{b}$, $B=y_{OT}^{b}$, $C, OT=y_{OT}^{r}$, Sig(A ‖ B ‖ OT ‖ C)가 된다. 여기서 a와 b는 Z_q에서 임의로 선택된 난수이며 y_{OT}는 인출자 추적을 위한 공개키이다. '‖'는 비트 결합(bitwise concatenation)을 나타내며, Sig(A ‖ B ‖ OT ‖ C)는 $(A \| B \| OT \| C)$를 신뢰기관이 서명한 것으로 이후에는 간단하게 Sig(C)로 표기하기로 한다. A는 지불과정에서 고객이 C의 표현을 알고있는지 확인하는데 사용된다. B와 OT는 C의 인출자를 추적할 수 있다는 것을 증명할 때 사용된다.

여러 동전을 사용하여 지불하는 것이 보다 일반적인 지불과정이지만 여기서는 문제를 간단하게 하기 위해 지불대금과 동전의 액면가는 항상 같다고 가정한다. 즉 지불과정에서 동전 하나와 상품 하나를 교환한다고 가정한다. 이 화폐의 지불과정은 [그림 1]과 같다. I는 상품의 식별자, y_S는 상점의 식별자, T_S는 구매시간을 나타내며, $H_q: \{0,1\}^* \rightarrow Z_q$는 충돌회피 해쉬함수(collision-resistant hash function)이다. 고객이 상품을 구매하려면, 고객은 상점에게 화폐, 상품의 식별자 I, 가격 v를 준다. 상점은 고객에게 도전(challenge) c를 주고, 고객은 해당하는 응답(response) s_1과 s_2를 계산하여 상점에게 준다. 이 도전/응답을 통해서 상점은 고객이 화폐의 올바른 사용자임을 알 수 있다. 도전/응답은 상점이 입금할 때 반드시 은행에 제시해야 하고, 은행은 이를 이용하여 고객이 이중사용했을 때 고객의 신원을 알 수 있다. 즉, 상점은 화폐를 받은 것만으로는 의미가 없고, 도전/응답과정을 마쳐야만 지불이 완료되었다고 볼 수 있다. 상점은 지불이 완료되면 해당하는 상품을 고객에게 전달한다.



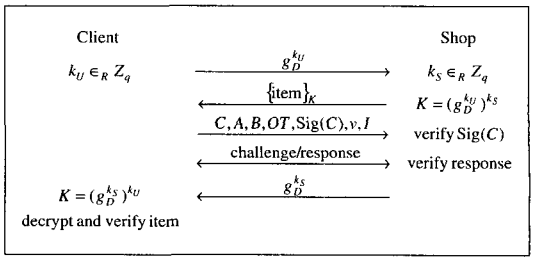
〔그림 1〕 가정하는 화폐의 지불과정

2.2 Xu 등의 원자성 보장 방법

Xu 등은 오프라인 화폐시스템에서 원자성을 보장하는 방법을 제안하였다[3]. 이들은 분쟁이 발생했을 때만 신뢰기관이 참여하는 낙관적인 접근방법을 사용한다. 자세한 지불과정은 [그림 2]와 같다. 상점이 먼저 상품을 암호화하여 고객에게 주면 고객은 상점에게 대금을 지불한다. 상점은 지불을 확인하고 상품의 해독키를 고객에게 준다. 대금을 지불한다는 것은 화폐를 주고 도전/응답까지 수행한 것을 말한다. 상점은 고객에게 받은 대금을 입금시한 내에 은행에 입금한다.

이 방법에서는 상점이 먼저 대금을 받기 때문에 상점은 손해보는 경우가 없다. 이렇게 하는 것은 고객이 익명으로 참여하기 때문이다. 반대로 고객은 대금을 지불하고도 상품을 받지 못하는 상황이 발생할 수 있다. 이 경우 고객은 입금시한이 지난 후에 상품을 받지 못했다고 신뢰기관에 분쟁해결을 요청한다. 신뢰기관은 은행에 접촉하여 상점이 입금했는지 알아본 후, 입금 여부에 따라 분쟁을 해결한다. 만약 입금이 되었다면 상점이 응답을 받고도 해독키를 주지 않은 것이고, 입금이 되지 않았다면 상점이 응답을 받지 못한 것이다. 입금시한이 필요한 이유는 신뢰기관이 분쟁을 해결할 때 고려해야 하는 경우의 수를 줄이기 위함이다. 이로 인해서 상점의 입금시한이 지날 때까지 고객은 분쟁해결을 받을 수가 없다. 이런 점은 고객에게 불편한 요소가 될 수 있다.

입금시한을 없애고 분쟁해결을 받도록 할 수 있



〔그림 2〕 Xu 등의 지불 프로토콜

다. 하지만 입금시한이 없으면 다음과 같은 과정을 거쳐야 한다. 고객이 신뢰기관에 분쟁해결을 요청하면, 신뢰기관은 은행에 입금여부를 묻는다. 만약 입금이 된 상태라면 상점이 대금을 받았다는 것이 명확하다. 그렇지 않다면 상점의 입금을 막고, 상점에 문의해야 한다. 상점이 고객에게 대금을 받지 못했다고 주장한다면 고객에게 거래가 취소되었음을 증명하는 진술서(affidavit)를 작성하여 준다. 반대로 상점이 대금을 받았다고 하면 상점으로부터 해독키를 받아 고객에게 전달하고 은행에게 상점의 입금을 허가하도록 하여야 한다. 다시 말해 입금시한을 없애면 상점이 입금하지 않았을 때 신뢰기관이 상점의 상태를 알 수 없으므로 상점에 거래할 의사가 있는지 문의해야 한다.

고객이 해독키를 받아 상품을 풀어보았는데 원하는 상품이 아닐 수 있다. 이 때 고객은 신뢰기관에 분쟁해결을 요청하고 상점의 부정을 증명해야 한다. Xu 등의 방법은 도전값을 만들 때 상품의 식별자를 넣어서 이를 증명하고자 하였다. 하지만 신뢰기관은 상품의 식별자만으로 고객이 무엇을 받았는지 확인할 수 없다. 또한 이 도전값은 은행에 입금되었을 때 은행이 확인해야 한다. 그래서 은행이 화폐가 사용된 용도를 알게되므로 거래행위의 일부가 노출된다. 이 방법을 개선하여 송장(invoice)을 사용하는 방법이 있다[12]. 상점은 암호화된 상품의 해쉬값, 화폐, 거래조건에 서명하여 송장을 작성하며, 고객에게 암호화된 상품을 줄 때 함께 준다. 분쟁이 발생하면 신뢰기관은 고객이 제시한 송장을 이용하여 상점의 부정을 확인할 수 있다. 송장을 쓰지 않고 도전값을 생성할 때 암호화된 상품의 해쉬값을 넣는 방법이 있다. 그러나 이 방법은 상점이 은행에 입금한 경우에만 송장과 같은 기능을 하게 된다. 따라서 이 논문에서는 송장을 사용한다

Ⅲ. 수학적 배경

3.1 Naccache 암호시스템

Naccache 암호시스템은 Naccache와 Stern이 제안한 공개키 암호시스템이다[8]. 이 암호시스템은 RSA 법에서 다차잉여(higher residue)를 계산하는 것이 어렵다는 것에 기반하고 있다. 이들은 결정적인(deterministic) 방법과 확률적인(probabilistic) 방법, 두 가지를 제안하고 있으며, 이 논문에서는 결정적인 방법을 사용한다.

결정적인 방법에서 공개키는 $n=PQ$ 와 $g\in Z_n^*$이며, 개인키는 P, Q, σ이다. 여기서 P와 Q는 큰 소수이며, 그것의 곱인 n의 길이는 적어도 768 비트 이상이어야 한다. σ는 중복되는 소인수를 가지지 않는 B- 매끄러운(B-smooth)[1] 홀수인 160비트 정도의 정수이고, 이 때 B는 10 비트 정도의 작은 정수이다. 이 σ는 $\sigma \mid \phi(n)$과 $\gcd(\phi(n)/\sigma, \sigma)=1$을 만족해야 한다. g의 위수는 σ의 배수이어야 한다. 이런 조건을 만족하는 n, σ, g는 다음과 같이 생성할 수 있다.

먼저 σ를 생성하기 위해 짝수 k개의 서로 다른 홀수인 소수 p_i를 선택하고, $u=\prod_{i=1}^{k/2} p_i$ 와 $v=\prod_{i=k/2+1}^{k} p_i$를 계산한다. 그 다음 u와 v를 이용하여 $\sigma=uv=\prod_{i=1}^{k} p_i$ 를 계산한다. P와 Q는 $P=2au+1$과 $Q=2bv+1$을 소수로 만드는 a와 b 선택하여 생성한다. g는 임의로 선택한 다음에 위수가 $\phi(n)/4$인지를 검사하여 생성한다.

암호화는 한번의 지수연산만 하면 된다. 메시지 $m(<\sigma)$을 공개키 (n, g)로 암호화 하려면 $g^m \bmod n$을 계산한다. 암호해독은 중국인의 나머지 정리(Chinese remainder theorem)를 사용한다. σ의 k개의 소인수는 각각 p_i이고, 이 때 m_i를 법 p_i에서 m과 합동인 수라 하자. k개의 m_i를 구하면 중국인의 나머지 정리를 사용하여 메시지 m을 구할 수 있다. 암호문 $c=g^m \bmod n$에서 m_i를 구하기 위해서는 $c_i=c^{\phi(n)/p_i} \bmod n$를 계산한다. c_i는 $y_i=(m-m_i)/p_i$일 때수식 (1)의 유도에 의해서 법 n에서 $g^{m_i \phi(n)/p_i}$과 합동이며, 이 식을 만족하는 m_i는 0부터 p_i-1까지 대입해서 찾을 수 있다.

$$c_i = c^{\frac{\phi(n)}{p_i}} = g^{\frac{m\phi(n)}{p_i}} = g^{\frac{(m_i+y_i p_i)\phi(n)}{p_i}} = g^{\frac{m_i\phi(n)}{p_i}} g^{y_i\phi(n)} = g^{\frac{m_i\phi(n)}{p_i}} \bmod n \quad (1)$$

이 암호시스템은 개인키 (P, Q, σ)를 알고 있는 사람만이 기저 g에 대한 c의 이산대수 m을 계산할 수 있다. 이와 같은 방식의 암호시스템으로 Okamoto와 Uchiyama의 암호시스템[13]이 있다. 이 논문에서는 Naccache의 시스템을 사용하고 있지만 Okamoto와 Uchiyama의 암호시스템을 사용할 수도 있다.

1) 수가 B 매끄럽다는 것은 B보다 큰 소인수를 가지지 않는 정수임을 의미한다.

3.2 다른 군에서의 이산대수 등가 증명

이 절에서는 확인 가능한 암호기법에 필요한 이산대수 등가 증명을 설명한다. 이 기법은 다른 군에서의 이산대수 등가 증명이 필요하다. 이 증명에서는 두 군의 위수가 다르므로 이산대수의 범위도 같이 증명해야 한다.

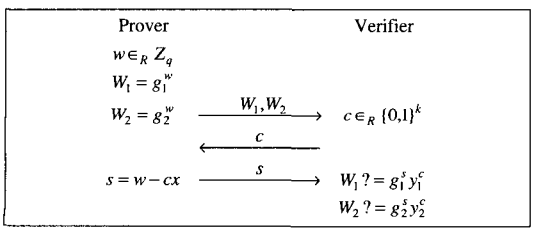
3.2.1 이산대수의 등가 증명

같은 군에서 기저 g_1과 g_2에 대한 $y_1 \in G_q$와 $y_2 \in G_q$의 이산대수가 같음을 증명하는 프로토콜은 [그림 3]과 같다[14]. 이 증명을 이 논문에서는 PKProof ($\log_{g_1} y_1 = \log_{g_2} y_2$)로 표기한다.

다른 군에서의 이산대수 등가 증명은 [그림 3]에서 s를 나머지 연산을 하지 않고 확인자에게 주어 증명을 할 수 있다고 생각할 수 있다. $G_1 = \langle g_1 \rangle$과 $G_2 = \langle g_2 \rangle$는 각각 군의 위수가 q_1과 q_2인 서로 다른 군으로, $q_1 < q_2$를 만족하고 $\gcd(q_1, q_2) = 1$이라고 가정하자 $y_1 = g_1^x$, $y_2 = g_2^{w'}$, $W_1 = g_1^w$, $W_2 = g_2^{w'}$, $x \neq x'$일때 증명자가 G_1과 G_2의 위수를 모두 안다고 하면 증명자는 아래 두 식을 만족하는 s를 중국인의 나머지 정리를 이용하여 구할 수 있다.

$$s \equiv w - cx \pmod{q_1}$$
$$s \equiv w' - cx' \pmod{q_2}$$

이 s는 $Z_{q_1 q_2}$에서 유일하다. $s < q_1$일 때 $w - cx \bmod q_1$과 $w' - cx' \bmod q_2$는 반드시 같아야 한다. 하지만 증명자가 c를 예측할 확률은 $1/2^k$이므로 증명자가 $w - cx \bmod q_1 = w' - cx' \bmod q_2$을 만족하는 w'을 선택할 확률은 무시할 수 있다. 따라서 s의 범위를 제한한다면, 이를 다른 군에서의 이산대수 등가 증명으로 사용할 수 있다. s의 범위를 제한하기 위해서는 x의 범위와 w의 범위를 제한해야 한다. 이는 이산대수의 범위 증명을 통해서 할 수 있다.



〔그림 3〕 이산대수 등가 증명

3.2.2 이산대수의 범위 증명

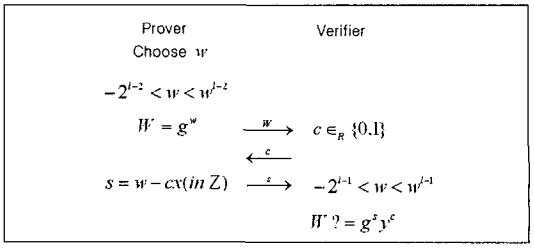
이산대수 범위 증명(interval proof)은 x를 밝히지 않고 y의 이산대수를 알고 있음을 증명하면서 동시에 x가 어떤 범위 안에 있음을 증명하는 것이다. 이진(binary) 도전을 사용하는 이산대수의 범위 증명[10]은 [그림 4]의 프로토콜을 k번 수행하여 이루어진다. 이 증명을 통해서 x가 정해진 범위 $(-2^l, 2^l)$에 속해있음을 증명한다. 여기서 l은 $2^{l+l'}$이 q보다 작도록 설정해야 한다. 하지만 실제로 x의 범위가 $(-2^{l-2}, 2^{l-2})$ 이어야 증명에 성공할 수 있다. 확인자는 응답 s의 범위를 확인해서 s의 범위를 알게 되므로 증명자는 s계산에서 나머지 연산을 하지 않는다. 이 증명을 PKProof($\log_g y \wedge -2^{l'} < \log_g y < 2^{l'}$)이라고 표기한다. 확인자가 s가 범위 $(-2^{l-1}, 2^{l-1})$인지 검사하여 x가 범위 $(-2^l, 2^l)$에 있다고 믿게 되는 이유는 이 증명의 지식추출기(knowledge extractor)가 계산하는 x의 범위 때문이다. 표준 rewinding 기술을 이용하면 지식추출기는 같은 W를 사용하지만 c와 s값이 다른 두 트랜스크립트를 얻을 수 있다.

따라서 $W = g^s y^c = g^{s'} y^{c'}$ $(c \neq c', s \neq s')$식으로 부터 $x = (s' - s)/(c - c')$를 계산할 수 있다. 그런데 $(c - c')$은 항상 1 또는 -1이 되어 $(s' - s)$를 나누므로 x는 범위 $(-2^l, 2^l)$에 속한다.

이 법위증명은 이진도전을 사용하기 때문에 효율성이 떨어진다. Camenisch와 Michels는 강한 RSA 가정(strong RSA assumption) 하에서 효율적인 이산대수 범위 증명 방법과 다른 군에서의 이산대수등가 증명을 제안하였다[9].

강한 RSA 가정. 충분히 큰 RSA 법 n과 Z_n^*의 임의의 원소 z가 주어졌을 때, $e > 1$과 $z = u^e \bmod n$을 만족하는 $(u, e) \in Z_n^* \times Z$를 구하는 것은 어렵다.

[9]의 범위증명은 이진도전을 사용하지 않고 $c = \{0,1\}^k$을 사용한다. 그러나 이와 같은 도전을 사



〔그림 4〕 이진 도전을 사용하는 이산대수 범위증명

용하면 지식추출기 $x=(s'-s)/(c-c')$에서 $(c'-c)$가 항상 $(s-s')$을 나누어야 x의 범위를 증명할 수 있다. $d=\gcd(s'-s, c-c')$이라고 할 때, 확장 유클리드 알고리즘(extended Euclidean algorithm)을 이용하여 다음을 만족하는 u와 v를 계산할 수 있다.

$$u\frac{c-c'}{d}+v\frac{s'-s}{d}=1$$

따라서 다음이 성립한다.

$$g=g^{u\frac{c-c'}{d}+v\frac{s'-s}{d}}=g^{u\frac{c-c'}{d}}(g^{s'-s})^{\frac{v}{d}}=g^{u\frac{c-c'}{d}}(y^{c-c'})^{\frac{v}{d}}=(g^u y^v)^{\frac{c-c'}{d}}$$

그런데 $d<c-c'$이면 1이 아닌 g의 루트를 계산할 수 있다. 그러나 이것은 강한 RSA 가정에 위배되므로 $d=c-c'$이어야 한다. 따라서 $(c-c')\mid(s'-s)$이다. 그러므로 RSA 법을 사용하면 이진 도전을 사용하지 않아도 x의 범위 증명이 가능하다. 지식추출기로 확인자가 알게되는 x의 범위를 구해보면 이 된다. 하지만 s계산에서 cx가 $(-2^{l-2}, 2^{l-2})$에 속해야 하므로 s가 k비트일 때 증명 가능한 x의 범위는 $(-2^{l-2-k}, 2^{l-2-k})$이 된다.

강한 RSA 가정을 사용하여 법위증명을 하게되면 효율적으로 다른 군에서 이산대수 등가 증명을 할 수 있다. 그러나 증명할 수 있는 x의 범위가 도전의 길이만큼 줄어들게 된다. 따라서 $(-2^{l-2-k}, 2^{l-2-k})$에 속하지 않으면 증명을 하지 못하는 문제점이 있다. 이 문제는 직접 x를 선택하여 증명을 한다면 상관없지만 만약 어떤 계산에 의해 생성된 x를 증명해야 하는 경우에는 사용하기 힘들다. 이 논문에서는 이 문제를 극복하기 위해 범위증명 없이 효율적으로 확인 가능한 암호화를 하는 방법을 제시한다.

Ⅳ. 확인 가능한 암호화를 이용한 원자성 보장 방법

이 장에서는 기존 지불의 원자성을 보장하는 방법에 확인 가능한 암호화를 적용하여 견고하고 분쟁해결이 명확한 원자성 보장 방법을 제안한다. 확인 가능한 암호화를 사용함으로써 기존의 고객이 먼저 지불하고 나중에 상품을 받는 방식에서 반대로 고객이 먼저 상품을 받고 나중에 지불하는 방식으로 전환하였다. 화폐를 확인 가능하게 암호화하기 위해서 Naccache 암호시스템과 다른 군에서의 이산대수 등가증명을 사용하였다.

4.1 시스템 설정

제안하는 방법의 시스템 설정은 분쟁을 해결해주는 신뢰기관의 공개키 설정이 추가로 필요하다는 것을 제외하고는 가정하는 화폐시스템의 초기 설정과 같다. 신뢰기관은 Naccache 암호시스템의 공개키 (n, g)를 생성하여 공개한다. 이 때 g의 위수가 g_U의 위수보다 커야한다. g_U의 위수는 q이므로 신뢰기관은 공개키 (n, g)를 생성할 때 곱셈군 Z_n^*에서 g의 위수가 q보다 큰지 확인해야 한다. 정확하게 말하면 Naccache 암호시스템에서 σ보다 작은 값을 암호화하므로 σ가 q보다 커야한다. 고객과 상점은 지불에 참여하기 전에 신뢰기관의 올바른 Naccache 공개키를 알고있어야 한다.

4.2 확인 가능한 암호화를 이용한 지불

이 논문은 가정한 화폐시스템의 지불과정에서 고객의 지불을 확인 가능한 암호기법으로 암호화한다. 그러면 상점은 나중에 대금을 받을 수 있다는 확신을 갖게 되어 기존과 달리 고객보다 먼저 프로토콜을 완료할 수 있다. 지불은 두 가지 방법으로 암호화를 할 수 있다. 하나는 은행이 서명하여 발행한 동전 $\mathrm{Sig}(C)$를 암호화하는 것이고, 다른 하나는 응답을 암호화하는 것이다. 응답을 암호화하여도 되는 이유는 앞서 말한 것처럼 도전/응답 과정이 끝나야 지불이 완료되는 것이기 때문이다. 만약 $\mathrm{Sig}(C)$를 암호화한다고 하면 각각의 서명기법마다 다른 확인 가능한 암호기법이 필요하다. 하지만 응답을 암호화하면 은행이 어떤 서명기법을 사용하더라도 이산대수 기반 도전/응답을 하는 시스템이면 동일한 암호화기법을 적용할 수 있다. 그래서 이 논문에서는 응답을 암호화하는 방법을 선택하였다.

가정한 화폐의 지불 과정인 [그림 1]에서 응답은 s_1과 s_2로 이루어져 있다. 이 두 개중 s_1만을 확인 가능한 암호기법으로 암호화한다. 고객은 $E=g_U^{s_1}$과 $V=g^{s_1} \bmod n$을 계산하여 (E, V)를 상점에게 준다. V는 s_1을 신뢰기관의 공개키로 암호화한 것이

다. 그러면 (E, V)는 신뢰기관의 공개키로 s_1을 확인할 수 있는 암호문이 된다. 상점은 $A? = Eg_T^{s_2}D^c$을 만족하는지 검사하여 도전에 대한 올바른 응답인지 확인한다. 그리고 고객과 상점은 각각 증명자와 확인자로 [그림 5]와 같이 PKProof($\log_{g_U}E = \log_g V$)를 수행한다. 이 증명으로 고객은 기저 g와 g_U에 대한 E와 V의 이산대수가 같다는 것을 상점에게 증명하게 된다. 또한 고객이 s_1을 알고 있다는 것도 동시에 증명하게 된다. 이렇게 하면 상점은 올바른 응답이 암호화되어 있으며, 이 암호문을 신뢰기관이 풀 수 있다는 것을 확인할 수 있다. 이 논문에서는 이와 같은 암호화 기법을 $VE(\cdot)$라고 표기한다.

3.2절에서는 증명자가 두 군의 위수를 안다고 가정했다. 그러나 여기서는 고객이 G_q의 위수는 알지만 Z_n^*의 위수는 알지 못한다. 이 때 다른 군에서의 이산대수 등가 증명에서 이산대수의 범위 증명이 필요 없다는 것을 아래와 같이 증명할 수 있다.

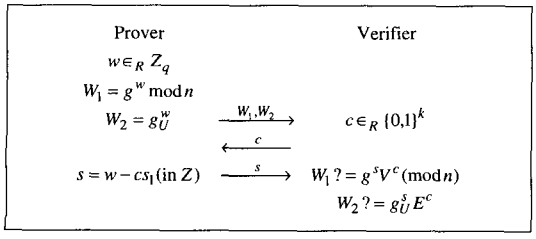
정리 1. 만약 증명자가 한 군의 위수를 알지 못하고 계산할 수 없다면, 다른 군의 이산대수 등가 증명에서 범위 증명을 하지 않아도 안전하다.

증명. $G_1 = \langle g_1 \rangle$과 $G_2 = \langle g_2 \rangle$는 각각 위수가 q_1, $q_2 (q_1 < q_2)$인 서로 다른 군이라고 하자. 증명자는 q_1은 알지만 q_2는 알지 못한다고 가정하자. $y_1 = g_1^x$, $y_2 = g_2^{x'}$, $w_1 = g_1^w$, $w_2 = g_2^{w'}$ 이고 $x \neq x'$이고 x와 x'은 법 q_1에서 합동이 아니라 하자. 또한 증명자는 도전 c를 예측할 수 없다고 하자. 이 때 증명자는 다음 두 식 (2)와 (3)을 동시에 만족하는 s를 생성하면 다른 x로 증명을 성공할 수 있다.

$$s \equiv w - cx \pmod{q_1} \quad (2)$$

$$s \equiv w' - cx' \pmod{q_2} \quad (3)$$

증명자는 s를 계산하는 상황에서 이미 x, x', w, w'을 각각 y_1, y_2, W_1, W_2를 사용하여 고정(commit)한 상태이다. 만약 증명자가 q_2를 안다면 중국인의 나머지 정리를 사용하여 s를 계산해 낼 수 있지만 증명자는 q_2를 알지 못하므로 이 방법으로는 증명에 성공할 수 없다. 증명자는 임의로 c를 예측하여 증명을 성공할 수 있지만 그 확률은 $1/2^k$ 이다. 또한 증명자는 $s = w - cx (\text{in } Z)$를 계산할 수 있으며, 이 s는 (2)식을 만족한다. 하지만 s가 (3)식을 우연히 만족할 확률은 $1/q_1$ 이다. $x \neq x'$이고 $x \equiv x' \pmod{q_1}$인 x와 x'을 사용하여 증명할 수 있지만 이 논문에서는 법 q_1에 대해서 x가 등가인 것만 증명하면 되기 때문에 문제가 되지 않는다. □



[그림 5] PKProof ($\log_{g_U} E = \log_g V$)

한 가지 더 고려해야할 점은 상점이 응답의 유효성을 A와 $Eg_T^{s_2}D^c$가 법 p에서 같은지를 비교하여 확인한다는 것이다. s_1을 넣어서 $g_U^{s_1}g_T^{s_2}D^c$을 계산하지 않고 E로 대신하여 검사하여도 $\log_{g_U}E$를 알고있음을 보이면 안전성에 문제가 되지 않는다는 것을 아래와 같이 증명할 수 있다.

정리 2. 상점이 s_1대신에 E를 사용하여 A와 $g_U^{s_1}g_T^{s_2}D^c$가 법 p에서 등가인지 증명하여도 $\log_{g_U}E$를 알고있음을 보이면 안전하다.

증명. 화폐 C의 표현을 알지 못하는 사람도 s_2'을 임의로 정하고 $E' = g_U^{s_1'} = Ag_T^{-s_2'}D^{-c}$를 계산하여 E'을 구할 수 있으므로 표현 증명을 할 수 있다. 하지만 지불과정에서 표현 증명과 더불어 E와 V의 이산대수 등가 증명을 해야 하는데, 이렇게 계산한 E'을 이용해서는 $\log_{g_U}E'$을 계산할 수 없으므로 증명할 수 없다. 그러므로 제안한 방식으로 표현 증명을 하여도 문제가 발생하지 않는다. □

4.3 지불과정

제안하는 지불과정은 [그림 6]과 같다. 고객은 상점에게 화폐와 고객의 Diffie-Hellman(DH) 키 $g_D^{k_U}$을 준다. 상점은 암호화키 K를 생성하여 상품을 암호화하고 상점의 DH 키를 해쉬한 값 $H(g_D^{k_S})$과 함께 고객에게 전달한다. 암호화된 상품, 송장, 상점의 DH 키를 해쉬한 값 $H(g_D^{k_S})$, 거래 시간 T_S를 고객에게 전달한다. 송장은 암호화된 상품, 상품 가격 v, 사용한 화폐, 상점 식별자 y_S, 거래 시간, 암호화에 사

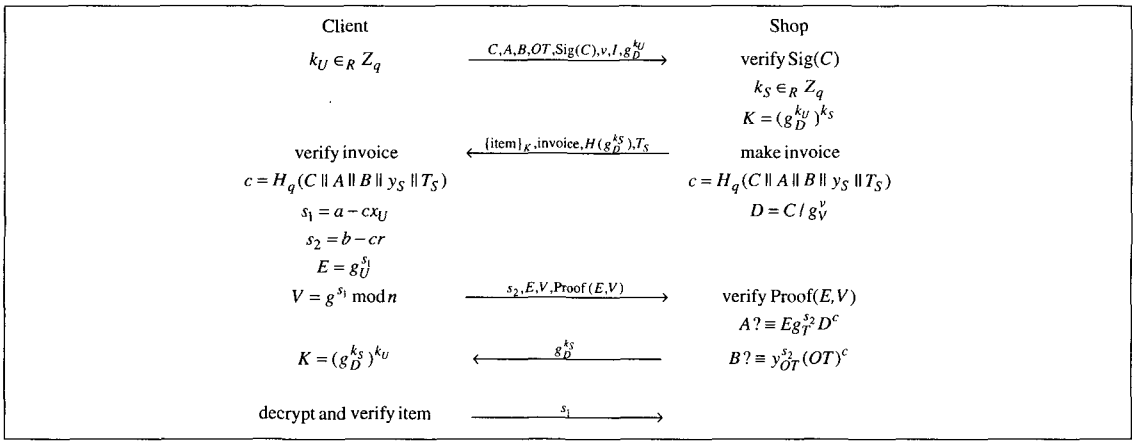
용된 키를 서명하여 만든다. 송장에 $H(g_D^{k_S})$를 넣어 자신의 키를 고정(commit)하고, $g_D^{k_U}$를 넣어 고객의 키와 연결하는 이유는 나중에 분쟁이 발생하였을 때 실제 사용된 키가 아닌 다른 키를 제시할 수 없도록 하기 위함이다. 암호화된 상품을 포함하는 이유는 상점이 어떤 암호화키로 어떤 상품을 고객에게 주었는지 고객이 신뢰기관에 송장을 통해서 증명할 수 있도록 하기 위해서이다. 고객은 상점의 송장이 올바른지 확인한다. 거래하기로 한 내용이 맞으면 고객은 확인할 수 있게 암호화된 응답 $VE(s_1)$과 S_2를 상점에게 준다. 상점은 검증과정을 통해서 고객이 화폐의 정당한 사용자인지 그리고 신뢰기관이 응답을 해독할 수 있는지를 확신할 수 있게 된다. 문제가 없다면 상점은 고객에게 상점의 키 $g_D^{k_S}$을 주고, 고객은 해독키 K를 계산하여 암호화된 상품을 푼다. 고객이 원하는 상품을 얻었다면 고객은 상점에게 응답을 주어 지불을 완료한다.

제안하는 지불과정은 Xu 등의 방법과는 달리 입금시한을 사용하지 않는다. 그 이유는 입금시한이 없어도 신뢰기관은 은행에 접촉할 필요가 없고 고려해야 하는 경우의 수도 복잡하지 않다. 만약 제안하는 방법에서 입금시한을 사용하면 분쟁해결을 더 용이하게 할 수 있다. 즉 신뢰기관은 분쟁해결과정에서 해결을 요청한 참여자와의 통신만으로 모든 분쟁을 해결할 수 있다. 다만 고객은 이 시한이 지나야만 분쟁해결을 요청할 수 있다. 이 방법에 대해서는 이 논문에서는 설명하지 않는다.

4.4 분쟁해결과정

상점과 고객이 지불과정을 정직하게 수행한다면 문제없이 거래가 이루어진다. 하지만 어느 한쪽이 정직하게 수행하지 않거나 네트워크 오류가 발생하면 분쟁이 일어날 수 있다. 제안하는 지불방법에서는 고객이 올바르게 암호화된 응답을 주기 전에 중단되면 서로 원하는 것을 얻을 수 없으므로 분쟁이 발생하지 않는다. 그러나 그 이후에 중단되면 다음과 같은 세 가지 분쟁이 발생할 수 있으며, 각 분쟁이 발생하면 다음과 같이 해결한다.

분쟁 발생 유형 1. 고객이 올바른 암호화된 응답을 주었는데도 상점이 고의로 상점의 키 $g_D^{k_S}$을 주지 않거나 네트워크 오류 등의 문제로 $g_D^{k_S}$을 받지 못할 수 있다. 두 경우 모두 아래와 같이 해결한다. 암호화된 응답을 신뢰기관이 상점에게 풀어줄 수 있기 때문에 고객은 반드시 신뢰기관에 해결을 요청해야 한다. 분쟁을 해결하는 프로토콜은 [그림 7]과 같다. 고객은 신뢰기관에 지불 트랜스크립트(transcript)를 주고 정당한 화폐의 사용자임을 표현 증명 PKProof($C = g_U^{x_U} g_V^{v} g_T^{r}$)으로 증명한다. 지불과정에서는 정해진 값을 이용하여 표현 증명을 하였지만 여기서는 임의의 값을 이용하여 증명한다. 따라서 익명으로 신뢰기관에 접촉이 가능하다. 만약 상점이 이미 해결을 요청해서 응답을 받아간 경우라면 신뢰기관은 그 때 저장해둔 상점의 키를 고객에게 전달한다. 그렇지 않은 경우라면 신뢰기



$\mathrm{invoice} = \mathrm{Sig}_S(H(I \| \{\mathrm{item}\}_K) \| v \| C \| y_S \| T_S \| H(g_D^{k_S}) \| g_D^{k_U})$

$\mathrm{Proof}(E,V) = \mathrm{PKProof}(\log_{g_U} E = \log_g V)$

[그림 6] 제안하는 지불과정

관은 상점에 요청하여 상점의 키를 받아 고객에게 전달하고, 상점에게는 응답을 준다. 이 과정은 온라인으로 이루어지기 힘들어서 고객이 차후에 다시 문의해야 한다. 만약 상점이 키를 주기를 거부하면 법적인 책임을 물게 된다.

분쟁 발생 유형 2. 상점이 키를 주고도 응답을 받지 못할 수 있다. 이는 고의로 고객이 응답을 주지 않는 경우와 네트워크 오류 등의 문제로 상점이 응답을 받지 못한 경우에 발생한다. 두 경우 모두 아래와 같은 방법으로 해결한다. 상점은 이 응답이 있어야 입금을 할 수 있으므로 반드시 신뢰기관에 해독을 요청해야 한다. 이 때 사용하는 프로토콜은 [그림 8]과 같다. 상점은 지불 트랜스크립트, 상점의 키 $g_D^{k_S}$를 신뢰기관에 주고 분쟁해결을 요청한다. 신뢰기관은 트랜스크립트의 도전값이 정확한지 계산하여 확인한다. 이 과정에서 신뢰기관은 상점이 올바른 해독키를 주었는지도 알 수 있다. 도전값이 맞으면 상점에게 응답을 해독해 준다. 신뢰기관은 상점에게 받은 트랜스크립트와 상점의 키를 저장해 둔다. 그리고 분쟁 발생 유형 1에서처럼 고객이 신뢰기관에 해결을 요청하였을 때, 신뢰기관은 고객의 정보를 확인하고 고객에게 상점의 키를 전달한다.

분쟁 발생 유형 3. 상점이 준 상품이 거래하기로 한 상품이 아닐 수 있다. 이 때 사용하는 분쟁 해결 프로토콜은 [그림 9]와 같다. 고객은 신뢰기관에 지불 트랜스크립트, 암호화된 상품, 상점의 키 $g_D^{k_S}$와 해독키를 생성할 수 있도록 k_U를 준다. 이 때 제 3자가 상품을 얻지 못하도록 k_U의 비밀성을 보장하면서 증명으로 밝힌다. 신뢰기관은 고객이 준 값들이 올바른지 송장을 통해서 확인한다. 신뢰기관은 해독키 K를 $(g_D^{k_S})^{k_U}$으로 계산하여 암호화된 상품을 풀어보고 상품 식별자에 해당하는 상품인지를 비교한다. 이 비교는 전자적으로 할 수 없다. 만약 올바른 상품이 아니라면 고객에게 지불취소를 증명하는 진술서(affidavit)를 서명하여 작성해준다. 고객은 익명성의 문제로 이 화폐를 다른 상점에 사용하기 어렵기 때문에 환불을 받아야한다. 고객은 은행에 화폐와 진술서를 제시하고 화폐의 정당한 사용자임을 증명하여 해당하는 금액을 환불받거나 다시 인출하면 된다. 그리고 신뢰기관은 상점에게 해당하는 책임을 지게 한다.

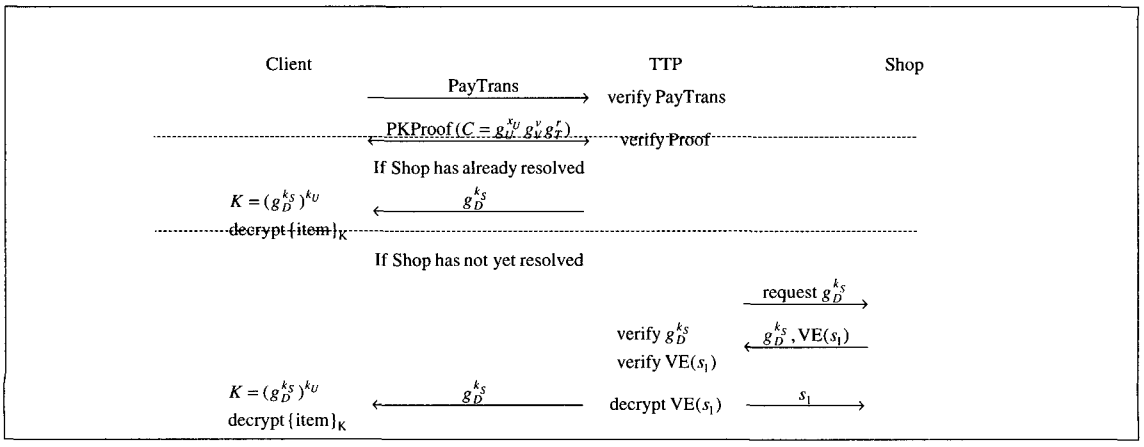
Ⅳ. 시스템 분석

이 장에서는 제안하는 방법의 원자성과 안전성에 대해 논한다. 또한 1장에서 언급한 기본 요구사항들을 만족하는지 보인다. 그리고 기존 Xu 등의 방법과 비교한다.

5.1 원자성

가정 1 (가정하는 화폐시스템의 안전성). 가정하는 화폐시스템은 고객의 신원을 노출하지 않으며, 고객이 이중사용했을 때 고객의 신원을 드러낸다.

가정 2 (Naccache 암호시스템의 안전성). Naccache 암호시스템은 RSA법 n의 인수분해를 알지 못하고는 암호문을 해독하는 것이 계산적으로 어렵다.



PayTrans = $\{C \| A \| B \| OT \| \mathrm{Sig}(C) \| v \| H(g_D^{k_S}) \| g_D^{k_U} \| y_S \| T_S \| (H(I \| \{\mathrm{item}\}_K) \| \mathrm{invoice} \| s_2 \| E \| V \| \mathrm{Proof}(E,V))$

[그림 7] 분쟁 해결 프로토콜 1

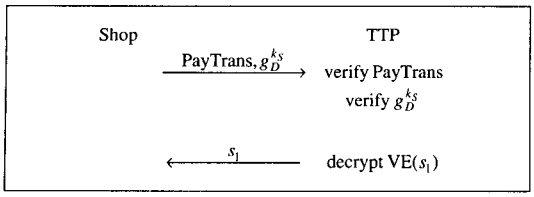
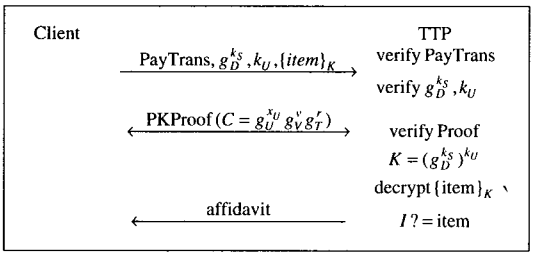


그림 8. 분쟁 해결 프로토콜 2



〔그림 9〕 분쟁 해결 프로토콜 3

정리 3. 고객은 대금을 지불하지 않고 상품을 얻을 수 없다.

증명. 고객이 상점으로부터 암호화된 상품과 상점의 키를 해쉬한 값을 받았을 때 고객이 $H(g_D^{k_S})$으로부터 $g_D^{k_S}$를 구한다면 암호화된 상품을 해독할 수 있다. 하지만 $H(\cdot)$는 일방향 해쉬함수이므로 고객이 $g_D^{k_S}$를 구하는 것은 계산적으로 어렵다. 고객이 상품의 해독키 $g_D^{k_S}$을 받은 후에 응답을 주지 않고 중단해도 상점은 $VE(s_1)$를 이미 가지고 있으므로 [그림 8]과 같이 신뢰기관으로부터 응답 s_1을 받을 수 있다. 만약 고객이 이 화폐를 다른 상점에 사용하거나 은행에서 환불받았다면, 가정 1의 이중사용 검출에 의해서 고객의 신원이 노출된다. 따라서 고객은 대금을 지불하지 않고는 상품을 얻을 수 없다. □

정리 4. 상점은 고객에게 상품을 주지 않고 대금을 받을 수 없다.

증명. 상점은 고객으로부터 응답을 받기 전까지는 은행에 화폐를 입금하지 못한다. 상점은 $VE(s_1)$을 받은 후에 신뢰기관에 요청하여 응답 s_1을 얻을 수 있다. 하지만 상점은 해독을 요청할 때 신뢰기관에게 자신의 올바른 DH 키를 제시해야한다. 따라서 이 경우에 고객은 [그림 7]과 같이 상점의 키를 얻을 수 있다. 만약 상점이 스스로 $VE(s_1)$을 해독할 수 있으면 공격이 가능하다. 상점은 (E, V) 중에 하나의 이산대수를 알아내야 하지만 G_q군과 Z_n^*군에서 이산대수를 계산하는 것은 이산대수 가정과 가정 2에 의해 계산적으로 어렵다. 상점은 잘못된 상품을 암호화해서 고객에게 줄 수 있지만 송장에 암호화된 상품과 상점의 키가 들어 있기 때문에 이러한 잘못을 행하였음을 부정할 수 없다. 따라서 상점은 올바른 상품을 주지 않으면 대금을 받을 수 없다. □

정리 5. 제안하는 방법은 지불과정의 원자성을 보장한다.

증명. 다음과 같은 4가지 경우만이 발생할 수 있다.

- 고객과 상점은 올바르게 화폐와 상품을 교환한다. 이는 지불의 원자성을 위배하지 않는다.
- 고객은 불합리한 이득을 얻지 못한다. 주장 3에 의해서 이런 경우는 발생하지 않는다.
- 상점은 불합리한 이득을 얻지 못한다. 주장 4에 의해서 이런 경우는 발생하지 않는다.
- 고객과 상점은 모두 교환에 실패한다. 이는 지불의 원자성을 위배하지 않는다.

위의 4가지 경우 모두 원자성을 위배하지 않으므로 제안하는 방법은 지불의 원자성을 보장한다. □

5.2 안전성

정리 6. 지불프로토콜과 분쟁해결 프로토콜에서 고객의 신원이 노출되지 않는다.

증명. 가정하는 화폐의 지불과정은 가정 1에 의해 익명성이 보장된다. 원자성을 보장하기 위해 추가로 교환되는 (E, V)와 PKProof($\log_{g_U} E = \log_g V$)에는 고객의 신원을 드러내는 정보가 없다. 분쟁 해결 1과 3에서 고객은 화폐의 표현을 영지식 증명으로 증명하지만 이 또한 고객의 신원을 노출하지 않는다. 분쟁 해결 3에서 고객이 진술서를 가지고 은행으로부터 환불을 받을 때도 같은 증명을 사용하므로 고객의 신원이 드러나지 않는다. □

정리 7. 고객과 상점 이외에 제 3자는 상품을 얻을 수 없다.

증명. 제 3자가 상품을 얻지 못하는 것은 DH 키 분배 프로토콜을 사용해서 만족한다. 상점과 고객은 상품의 암호화키를 DH 키 분배 프로토콜을 통해서 공유하므로 Diffie-Hellman 가정에 의해 제 3자는 상품을 얻을 수 없다. 신뢰기관 역시 분쟁 발생 유형 3에서 고객이 직접 해독키를 주는 경우를 제외하고는 이 해독키를 얻을 수 없어서 상품을 얻지 못한다.□

5.3 기존 지불의 원자성 보장 방법과의 비교

지불의 원자성을 보장하는 방법에 관한 연구는 많이

이루어져 있지 않다. 또한 공정한 교환과는 직접적인 비교가 불가능하므로 여기서는 대표적 방법인 Xu 등의 방법하고만 비교한다. 비교에서는 모든 도전/응답을 상호작용(interactive)으로 한다고 가정한다. 메시지 전송은 최적화 시켰으며, 연산은 비용이 많이 드는 지수연산을 고려하였다. 비용을 계산할 때 예를 들어 $g_1^a g_2^b g_3^c$은 3번의 지수연산이 필요한 것으로 가정하였다. 이 논문에서 사용하는 송장을 만들고 확인하는데는 지수연산이 필요하다. 하지만 Xu 등의 방법에서 같은 기능을 제공하기 위해서는 필요한 연산이므로 송장에 관한 연산은 고려하지 않는다. 제안하는 방법은 DH 키 동의를 위해서 고객과 상점이 각각 2회의 지수연산을 한다. 확인 가능한 암호화에는 고객이 2번, 상점이 4번의 연산이 필요하며, 상점이 응답을 확인하는데 5번과 Sig(C)를 확인하는데 4번의 지수연산이 필요하다. 전체적으로 고객이 6번, 상점이 15번으로 총 21번의 지수연산을 한다. [표 1]을 보면 제안하는 방법이 지불과정에서 2번의 메시지 전송과 7번의 지수연산이 더 필요한 것을 알 수 있다. 이것은 제안하는 방법이 지불과정에 확인 가능한 암호화를 위한 다른 군에서의 이산대수 등가 증명을 해야하기 때문이다.

Xu 등의 방법은 분쟁이 발생하면 항상 신뢰기관이 은행에 접촉하여 상점이 입금했는지를 확인해야 했다. 반면에 제안하는 방법은 신뢰기관이 은행에 접촉할 필요가 없다. 그 이유는 분쟁 발생 유형 2와 3이 발생하면 신뢰기관은 상점이 응답을 받았는지를 알게 되기 때문이다. Xu 등의 방법에서는 분쟁해결에서 일어날 수 있는 경우의 수를 줄이기 위해서 상점이 일정 시한 안에 입금을 하도록 했다. 그리고 상점의 입금시한이 지나야 고객은 분쟁해결을 요청할 수 있었다. 제안하는 방법은 상점이 입금시한을 가지지 않고 고객과 상점 모두 시간의 제약 없이 분쟁해결을 신뢰기관에게 요청할 수 있다.

V. 결 론

이 논문에서는 확인 가능한 암호화 기법을 사용하여 지불의 원자성을 제공하는 방법을 제안하였다. 제안하는 방법은 지불과정에서 화폐의 표현을 영지식으로 증명할 때 응답을 확인 가능하게 암호화하였으며, 이 암호화에서는 Naccache 공개키 시스템과 다른 군의 이산대수 등가 증명을 사용하였다. 이 증명은 기존의 다른 군의 이산대수 등가 증명과 달리 이산대수의 범위증명을 하지 않으며, 이산대수가 가질 수 있는 모든 범위에서 증명이 가능하다. 따라서 기존 증명보다 효율적이고, 표현문제를 사용하는 화폐시스템의 수정을 최소화할 수 있는 장점을 가지고 있다. 뿐만 아니라 도전에 대한 응답을 지수로 사용하여 확인하는 모든 화폐시스템에도 쉽게 적용할 수 있다.

확인 가능한 암호화를 사용하면 상점은 입금시한을 가지지 않고, 고객과 상점 모두 시간적이 제약 없이 분쟁해결을 요청할 수 있다. 또한 신뢰기관이 분쟁을 해결할 때 은행에 상점의 입금여부를 확인하는 절차가 필요 없어서 은행과 독립적으로 지불의 원자성을 제공할 수 있다. 반면에 모든 지불마다 확인 가능한 암호화를 위한 추가적인 증명을 해야하는 단점을 가진다. 또한 이 논문은 문제를 간단히 하기 위해 하나의 지불에 한 동전을 사용한다고 가정하였다. 따라서 여러 동전으로 지불할 때 효율적으로 지불의 원자성을 보장할 수 있는 방법에 대한 연구가 향후 필요하다.

참 고 문 헌

[1] D. Tygar, "Atomicity in Electronic Commerce," *Proc. of the 15th ACM Symp. on Principles of Distributed Computing*, pp. 8~26, ACM Press, 1996.

[2] C. Boyd and E. Foo, "Off-line Fair Payment Protocols using Convertible Signatures," *Advances in Cryptology, Asiacrypt* 1998, LNCS 1514, pp. 271~285, Springer-Verlag, 1998.

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

[4] N. Asokan, M. Schunter, and M. Waidner, "Optimistic Protocols for Fair Exchange," *Proc. of the 4th ACM Conf. on Computer and Communications Security*, pp. 6~17, ACM Press, 1997.

[5] G. Atenise, "Efficient Verifiable Encryption (and Fair Exchange) of Digital Signatures," *Proc. of the 6th Conf. on Computer and Communications Security*, pp. 138~146, ACM Press, 1999.

[6] N. Asokan, V. Shoup, and M. Waidner, "Optimistic Fair Exchange of Digital Signatures," *IEEE J. on Selected Areas in Communications*, Vol. 18, No. 4,

pp. 593~610, IEEE Press, 2000.

[7] S. Brands, "Untraceable Off-line Cash in Wallets with Observers," *Advances in Cryptology, Crypto 1993*, LNCS 773, pp. 302~318, Springer-Verlag, 1994.

[8] D. Naccache and J. Stern, "A New Public Key Cryptosystem Based on Higher Residues," *Proc. of the 5th ACM Conf. on Computer and Communications Security*, pp. 59~66, ACM Press, 1998.

[9] J. Camenisch and M. Michels, "Separability and Efficiency for Generic Group Signature Schemes," *Advances in Cryptology, Crypto 1999, LNCS 1666*, pp. 106~121, Springer-Verlag, 1999.

[10] J. Camenisch and M. Michels, "A Group Signature Scheme Based on an RSA-Variant," BRICS, Tech. Rep. RS~98~27, 1998.

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

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

[13] T. Okamoto and S. Uchiyama, "A New Public-Key Cryptosystem as secure as Factoring," *Advances in Cryptology, Eurocrypt 1998*, LNCS 1403, pp. 308~318, Springer-Verlag, 1998.

[14] D. Chaum and T.P. Pedersen, "Wallet Databases with Observers," *Advances in Cryptology, Crypto 1992*, LNCS 740, pp. 89~105, Springer-Verlag, 1993.

〈著者紹介〉

최 형 섭 (Hyungsup Choi) 학생회원
2001년~한양대학교 전자컴퓨터공학부(학사)
2003년~한양대학교 컴퓨터공학과(석사)
<관심분야> 암호기술 응용, 전자화폐
URL: http://infosec.hanyang.ac.kr/~hschoi

김 상 진 (Sangjin Kim) 정회원
1995년~한양대학교 전자계산학과(학사)
1997년~한양대학교 전자계산학과(석사)
2002년~한양대학교 컴퓨터공학과(박사)
2003년~현재 : 한국기술교육대학교 인터넷미디어공학부 전임강사
<관심분야> 암호기술 응용, 전자화폐, 전자선거
URL: http://infosec.kut.ac.kr/sangjin

오 희 국 (Heekuck Oh) 종신회원
1983년~한양대학교 전자공학과(학사)
1989년~아이오와주립대학 전자계산학과(석사)
1992년~아이오와주립대학 전자계산학과(박사)
1993년~1994년 : 한국전자통신연구원 선임연구원
1995년~현재 : 한양대학교 컴퓨터공학과 부교수
<관심분야> 암호기술 응용, 분산컴퓨팅
URL: http://infosec.hanyang.ac.kr/~hkoh