확인 가능한 암호화를 사용한 지불의 원자성 보장 방법

최형섭*, 김상진*, 오희국*

*한양대학교, 컴퓨터공학과

Providing Payment Atomicity Using Verifiable Encryption

Hyungsup Choi*, Sangjin Kim*, Heekuck Oh*

*Department of Computer Science and Engineering, Hanyang Univ.

요 약

확인 가능한 암호화는 암호문을 풀어보지 못해도 어떤 것이 암호화되어 있는지 확인할 수 있는 암호화 기법이다. 이 기법은 공정한 교환에서 암호화된 물건을 우선 제시하여 상대방이 물건을 받을 수 있다는 확신을 가지게 하는데 사용된다. 지불의 원자성을 보장하기 위해 공정한 교환을 적용해 볼 수 있다. 이 논문에서는 확인 가능한 암호화 기법을 표현문제를 사용하는 화폐시스템에 적용하여, 지불의 원자성을 제공하는 방법을 제안한다. 확인 가능한 암호화를 사용하면 분쟁이 발생했을 때 신뢰기관이 은행으로부터 상점의 입금여부를 확인할 필요가 없이 분쟁해결이 간단하다. 반면에 지불과정에서 확인 가능한 암호화를 적용하기 위한 증명이 추가로 필요하다.

I. 서론

전자화폐에서 지불과정은 고객이 상점에게 화폐를 지불하고 상점이 고객에게 상품을 전달하는 두 단계로 이루어진다. 고객과 상점은 서로를 신뢰하지 않는 관계이므로 지불과정은 반드시 원자성을 보장해야 한다. 만약 원자성을 보장하지 않으면 고객이 지불을 하고도 상품을 받지 못하거나 상점이 상품을 주고도 지불을 받지 못하는 문제가 발생할 수 있다. 이러한 지불의 원자성(payment atomicity)은 Tygar에 의해서 처음으로 제기되었다 [1]. 그러나 Tygar가 제안한 방법은 신뢰기관이 항상 지불에 참여해야하는 방식이었다. 이후에 Xu 등은 문제가 발생했을 때만 신뢰기관을 사용하는 낙관적인(optimistic) 접근 방법으로 지불의 원자성을 제공하였다 [2].

지불의 원자성에 관한 직접적인 연구 외에 공정한 교환(fair exchange)이라는 연구 분야가 있다 [3,4]. 공정한 교환은 네트워크상의 두 참여자가 서로의 물건을 교환할 때, 서로가 손해보지 않는다는 것을 보장하는 교환방식이다. 공정한 교환은 원자성을 제공한다는 측면에서 지불의 원자성과 유사한 형태를 가지며, 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 교환하는 데 집중되어 있다. 따라서 은행이 서명하여 발행한 화폐와 전자적인 상품을 교환하는 지불과정에 적용이 쉽지 않다.

공정한 교환에는 확인 가능한 암호화(verifiable encryption)를 사용하는 방법이 있다 [3,4]. 확인 가능한 암호화는 암호문을 풀어보지 못하더라고 암호문 안에 무엇이 암호화되어 있는지를 확인할 수 있는 암호화 기법이다. 공정한 교환에서는 신뢰기관의 공개키로 물건을 확인 가능한 암호화하여 상대방에게 전달한다. 이렇게 하면 암호문을 받은 사람은 물건을 받을 수 있다는 확신을 갖게 된다. 만약 물건을 받지 못하면 신뢰기관에게 암호문의 복호화를 요청하여 물건을 받을 수 있다.

현재의 화폐시스템은 대부분 익명이고 오프라인 방식이며 [5], 이런 화폐시스템에서 지불의 원자성 보장하기 위해 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 확인 가능한 암호화를 표현문제를 사용하는 화폐시스템에 적용하여, 지불의 원자성을 제공하는 방법을 제안한다

이 논문의 구성은 다음과 같다. 2장에서는 가정하는 전자화폐와 기존의 원자성 보장 방법에 대해서 설명하고, 3장에서는 확인 가능한 암호화와 제안하는 지불의 원자성 보장 방법을 설명하고 장단점을 분석한다. 끝으로 4장에서는 결론과 향후 연구방향에 대해 서술한다.

II. 관련 연구

1. 가정하는 전자화폐

가정하는 화폐시스템은 기본적으로 Solages와 Traore의 화폐[5]와 같다. 이 화폐는 표현문제를 사용하며, 동전은 $C = g_U^{x_U} g_V^{v} g_T^{r}$ 과 같이 구성되어 있다. x_U는 고객의 비밀신원정보이고, v는 화폐의 액면가이며, r은 고객이 사용한 은닉요소로 화폐추적과 인출자추적 기능을 제공하기 위해 사용된다. 고객은 제한적 은닉서명 프로토콜을 수행하여 화폐를 인출한다. 인출된 화폐는 $A = g_U^{a} g_T^{b}$, $B = y_{OT}^{b}$, $OT = y_{OT}^{r}$, $\mathrm{Sig}(C)$가 된다.

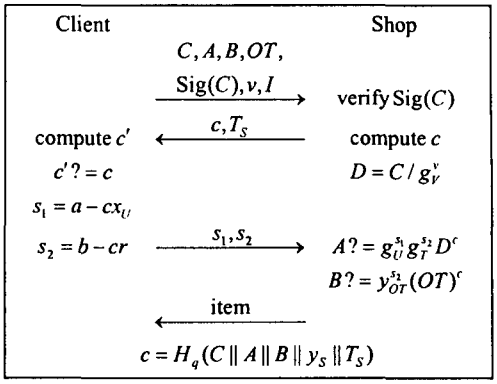
이 화폐의 지불과정은 그림 1과 같다. 고객은 상점에게 화폐, 사려는 상품의 식별자 I, 가격 v를 준다. 고객은 상점과 도전/응답(challenge/response)과정을 거쳐서 고객이 올바른 사용자임을 증명한다. 여기서 도전/응답은 상점이 입금할 때 반드시 은행에 제시해야 하는 정보이며, 은행은 이를 이용하여 고객이 이중사용했을 때 고객의 신원을 얻을 수 있다. 즉 화폐를 받은 것만으로는 의미가 없고 도전/응답을 마쳐야만 지불이 완료되었다고 볼 수 있다.

지불 프로토콜에서는 지불대금과 동전의 액면가가 같아서 하나의 동전을 이용하여 지불한다고 가정한다.

2. 기존의 지불의 원자성 보장 방법

Xu 등은 상점이 암호화된 상품을 먼저 고객에게 주고 고객으로부터 상품 대금을 받으면 상품의 복호화키를 주는 방법을 사용하였다 [2]. 상점의 입금시한이 있어서 상점은 상품의 대금을 입금시한 내에 반드시 입금해야 한다. 고객이 복호화키를 받지 못했을 때 고객은 상점의 입금시한이 지난 후에만 신뢰기관에게 해결을 요청할 수 있다. 이런 점은 고객에게 불편한 요소가 될 수 있다. 또한 해결과정에서 신뢰기관은 은행에 접촉해서 입금여부를 확인해야 한다. 이런 입금시한과 입금여부의 확인이 필요한 이유는 상점이 응답을 받았는지를 신뢰기관이 알 수 없기 때문이다. 상점이 입금시한 안에 입금을 하지 않으면 신뢰기관은 상점이 응답을 받지 못한 것으로 처리할 수 있다. 이 방법은 도전값을 만들 때 상품의 식별자를 넣어서 상점의 부정을 증명하려고 한다. 하지만 상점이 어떤 상품을 고객에게 주었는 지를 신뢰기관은 알 수 없고, 화폐가 어떤 상품을 사는데 사용되었는지 은행이 알게 되는 문제가 있다.

위의 방법을 개선하여 송장(invoice)을 사용하는 방법이 있다. 송장은 상점이 어떤 상품을 보냈는지를 명시하여 고객에게 주는 것이다. 이는 암호화된 상품의 해쉬값, 화폐, 거래 조건들에 상점이



<그림 1> 가정하는 화폐의 지불과정

서명하여 작성한다. 그러면 분쟁이 발생했을 때 고객이 신뢰기관에 송장을 제시하여 상점의 부정을 증명할 수 있다. 입금시한을 없애고 프로토콜을 설계할 수도 있지만 상점이 응답을 받았는지 여부를 신뢰기관이 알 수 없어서 분쟁해결이 복잡해진다.

III. 확인 가능한 암호화를 사용한 지불의 원자성 보장 방법

1. 확인 가능한 암호화

신뢰기관은 Naccache 등의 암호시스템[6]의 공개키를 설정하여 공개한다. 이 암호시스템의 공개키는 (g_T, n)이며, 메시지 m을 암호화할 때 $c = g_T^{m} \bmod n$ 을 계산한다. 개인키를 알고 있는 사람만이 암호문의 이산대수를 계산하여 메시지를 복호화할 수 있다.

가정하는 화폐에서는 s_1, s_2를 응답으로 사용했다. 이 응답을 확인 가능한 암호화 기법으로 암호화하기 위해서 응답 대신에 $g_U^{s_1}$, s_2, $c(= g_T^{s_1} \bmod n)$를 상점에게 준다. c는 s_1을 신뢰기관의 공개키로 암호화한 것이다. 고객은 상점에게 $g_U^{s_1}$과 c의 이산대수가 같다는 것을 증명한다. 이렇게 하면 상점은 암호문 안에 올바른 응답이 들어있음을 알 수 있고, 신뢰기관은 암호문을 풀어 응답을 복호화 할 수 있다. 상점은 $A? = g_U^{s_1} g_T^{s_2} D^c$을 확인할 때 $g_U^{s_1}$을 직접 넣어서 계산한다. 따라서 화폐 C의 표현을 알지 못하는 사람도 임의로 s_2'를 정하고 $g_U^{s_1} = A g_T^{-s_2'} D^{-c} \bmod p$를 계산하여 $g_U^{s_1}$을 구할 수 있다. 하지만 고객은 이산대수 가정에 의해 $g_U^{s_1}$으로부터 s_1을 계산할 수 없다. 따라서 고객은 $g_U^{s_1}$과 c의 이산대수 등가 증명을 통

해 $g_U^{s_1'}$의 이산대수가 s_1'이고 이 이산대수를 알고 있다는 것을 증명해야 하는데 이것을 할 수 없다. 그러므로 이런 방식으로 표현증명을 하여도 문제가 발생하지 않는다.

$g_U^{s_1}$과 c의 이산대수가 같음을 증명하는 것은 다른 군에서 이산대수 등가 증명으로 Camenish 등이 제안한 증명을 사용한다 [7]. 이 증명방법은 일반적인 이산대수 등가증명에 이산대수의 범위 증명(interval proof)을 추가적으로 하는 방식이다. 이 증명에서는 확인자와 증명자 사이에 처음 한 번의 설정단계가 필요하다. 설정단계는 비용이 많이 들지만 한 확인자가 여러 증명자에게 똑같이 사용할 수 있다. 그래서 이 논문에서는 신뢰기관과 상점(확인자)이 미리 설정단계를 수행하여 신뢰기관이 상점에게 인증서를 발급해 주게 하였다. 이렇게 하면 고객(증명자)은 상점(확인자)의 인증서를 확인하는 것만으로 설정단계를 마칠 수 있다.

2. 제안하는 원자성 보장 방법

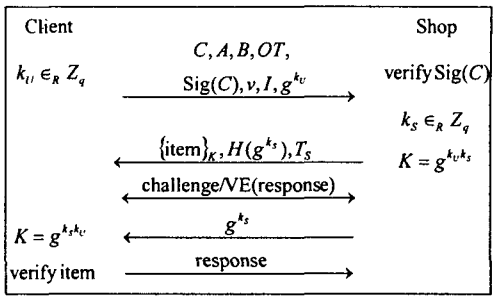
제안하는 지불과정은 그림 2와 같다. 고객은 상점에게 화폐와 고객의 Diffie-Hellman 키 k_U를 준다. 상점은 암호화키 K를 생성하여 상품을 암호화하고 상점의 Diffie-Hellman 키를 해쉬한 값과 함께 고객에게 전달한다. 상점은 다음과 같이 도전값 c를 만든다.

$$c = H_q(C \| A \| B \| y_S \| T_S \| H(g^{k_S}) \| g^{k_U} \| H(I \| \{\text{item}\}_K))$$

이때 $H(g^{k_S})$를 넣어서 자신의 키를 커밋(commit)하게 하고, g^{k_U}를 넣어서 고객의 키와 연결시킨다. 또한 $H(I \| \{\text{item}\}_K)$를 넣어서 상품의 식별자와 암호화된 상품을 연결하여 커밋한다. 고객은 상점의 도전값이 고객이 생성한 도전값과 같은지를 비교하여 상점이 커밋한 내용을 확인한다. 거래하기로 한 내용이 맞으면 고객은 확인 가능한 암호화된 응답 VE(response)를 상점에게 준다. 상점은 확인 가능한 암호화된 응답을 통해서 고객이 화폐의 정당한 사용자인지 그리고 신뢰기관이 응답을 복호화할 수 있는지를 확신할 수 있게 된다. 문제가 없다면 상점은 고객에게 g^{k_S}를 주고, 고객은 복호화키 K를 계산하여 암호화된 상품을 푼다. 고객이 원하는 상품을 얻었다면 고객은 상점에게 응답을 주고 지불을 완료한다.

상점과 고객이 위의 지불과정을 정직하게 수행한다면 문제없이 지불이 이루어진다. 하지만 어느 한쪽이 정직하게 수행하지 않으면 분쟁이 발생하게 되고 이 경우는 다음과 같은 분쟁의 유형에 따라 해결한다.

분쟁 발생 유형 1. 고객이 암호화된 응답을 주었는데도 상점이 상점의 키를 주지 않을 수 있다. 고객은 신뢰기관에 지불 트랜스크립트를 주고 정당한 화폐의 사용자임을 표현 영지식 증명으로 증명한다. 만약 상점이 이미 신뢰기관에 해결을 요청해서 응답을 받아간 경우라면 그 때 저장해 둔 상점의 키를 고객에게 전달한다. 그렇지 않은 경우라면 신뢰기관은 상점에 요청하여 상점의 키를 받아 고객에게 전달하고, 상점에게는 응답을 준다.



<그림 2> 제안하는 지불과정

분쟁 발생 유형 2. 상점이 키를 주고도 응답을 받지 못할 수 있다. 상점은 지불 트랜스크립트, $H(I \| \{\text{item}\}_K)$, 상점의 키를 신뢰기관에 주고 분쟁해결을 요청한다. 신뢰기관은 트랜스크립트의 도전값이 정확한지 확인하고 맞으면 상점에게 응답을 복호화하여 준다. 신뢰기관은 상점에게 받은 정보를 저장해 둔다.

분쟁 발생 유형 3. 상점이 준 상품이 거래하기로 한 상품이 아닐 수 있다. 고객은 신뢰기관에 지불 트랜스크립트, 암호화된 상품, 상점의 키 g^{k_S}와 복화화키를 생성할 수 있도록 k_U를 준다. 또한 화폐의 정당한 사용자임을 표현 영지식 증명으로 밝힌다. 신뢰기관은 고객이 준 값들이 올바른지 도전값을 직접 생성하여 확인한다. 신뢰기관은 복호화키를 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 화폐, 진술서를 제시하고 화폐의 정당한 사용자임을 증명하여 해당하는 금액을 환불받거나 다시 인출하면 된다. 그리고 신뢰기관은 상점에게 해당하는 책임을 지게 한다.

제안하는 지불의 원자성 보장 방법의 장점은 다음과 같다.

- 기존 방법과 달리 분쟁이 발생했을 때 신뢰기관이 은행에 접촉하여 상점이 입금했는지를 확인할 필요가 없다.
- 상점이 입금시한을 가지지 않고, 고객도 시간의 제약 없이 분쟁해결을 신뢰기관에게 요청할 수 있다.

• 고객은 상점이 올바른 상품을 주지 않았음을 신뢰기관에게 증명할 수 있어서 신뢰기관이 상점의 부정을 정확히 알 수 있다.

제안하는 지불의 원자성 보장 방법의 단점은 다음과 같다.

• 지불 프로토콜에서 응답을 확인 가능한 암호화할 때 추가적인 증명이 필요하다.

• 원자성 보장을 위한 신뢰기관의 공개키 설정과 상점의 인증서 발급이 필요하다.

IV. 결론

이 논문에서는 확인 가능한 암호화 기법을 사용하여 지불의 원자성을 제공하는 방법을 제안하였다. 확인 가능한 암호화를 사용하면 분쟁이 발생했을 때 입금시한을 가지지 않아 유연하면서도 분쟁해결이 명확하다. 또한 신뢰기관이 은행에 입금여부를 확인하는 절차가 필요 없다. 반면에 모든 지불마다 확인 가능한 암호화를 위한 추가적인 증명을 해야하는 단점을 가진다. 따라서 응답을 효율적으로 암호화할 수 있는 기법에 대한 연구가 필요하다. 또한 이 논문은 하나의 지불에 한 동전을 사용한다고 가정하였다. 여러 동전으로 지불할 때 효율적으로 지불의 원자성을 보장할 수 있는 방법에 대한 연구도 필요하다.

참고문헌

[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] Shouhuai Xu, Moti Yung, Gendu Zhang, and Hong 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, 1999.

[3] N. Asokan, V. Shoup, and M. Waidner, "Optimistic Fair Exchange of Digital Signatures," IEEE Journal on Selected Areas in Communications, Volume 18, Issue 4, pp. 593-610, IEEE Press, 2000.

[4] 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.

[5] 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, 1998.

[6] 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.

[7] J. Camenish and M. Michel, "Separability and Efficiency for Generic Group Signature Schemes," Proc. of the 19th Annual Int. Cryptology Conf., LNCS 1666, pp. 106-121, Springer, 1999.