익명성 조절이 가능한 전자화폐 프로토콜에 관한 연구

천윤정*, 송주석*

*연세대학교 컴퓨터과학과

A Study on anonymity-controllable Electronic Cash Protocols

Yoon-jung Chun*, Joo-seok Song*

*Department of Computer Science, Yonsei University

요 약

최근 인터넷의 급속한 발달로 인해, 인터넷을 이용한 전자상거래가 급증하고 있다. 그 결과로 전자화폐, 전자지갑, 전자수표와 같은 안전한 전자지불시스템에 대한 연구가 계속되고 있다. 그러나 현재 쓰이고 있는 전자화폐 프로토콜들은 대부분 사용자의 익명성 보장을 해주지 못하고 있다. 본 논문에서는 안전하고, 익명성을 보장해주는 효과적인 지불 프로토콜을 제안하였다. 사용자가 은행으로부터 전자화폐를 발급 받고 난 다음, 상점에 그 전자화폐를 사용하기 전에 더 높은 익명성을 원한다면 TTP (AP)를 거쳐서 사용자의 정보를 감추는 것이다. 이 TTP는 사용자로부터 받은 전자화폐의 유용성을 검증하고, 다시 암호한 데이터를 인증해준다. 이러한 새로운 방법을 통해 사용자는 자신의 시간, 연산과 비용 등을 고려하여 적절한 단계의 익명성을 제공받을 수 있다. 효율성 증가를 위해서, DLP 기반에서 ECDLP 기반으로 옮겼으며, 이중사용을 방지하기 위해 ECC를 이용한 Schnorr 서명을 사용하였다.

I. 서론

1991년 월드 와이드 웹(WWW)의 등장은 짧은 시간 내에 인터넷 이용 인구의 수를 급증시켰고, 그와 더불어서 사용자는 시간과 공간에 구애받지 않고 물건을 구매할 수 있는 인터넷을 통한 전자상거래를 많이 이용하게 되었다. 따라서 전자상거래는 기존에 화폐로 표현되던 가치정보를 가상 공간에서 표현할 수 있어야 했고, 그에 따라서 각종 전자지불 시스템이 등장하게 되었다.

전자화폐 시스템들은 많이 제안되었지만, 대부분 익명성의 문제를 잘 고려하지 않고 있다. 1983년 David Chaum에 의해서 제안된 on-line 지불 시스템의 경우 처음으로 익명성을 제공하여 주었으나[1], 이것은 상점과 은행에 대하여 약간의 익명성을 제공해 주는 것으로, 은행은 사용자와 그 화폐를 위한 매우 큰 DB를 유지하여야만 했다. 그 외에 온라인 지불 시스템인 CyberCoin은 상점이 지불을 승인할 때마다 은행에 직접 확인을 해보는 방법을 사용하였다. 이것은 은행이 모든 구매를 확인할 수 있으므로, 완전한 익명성을 제공하였다고 보기는 어렵다. 또 신용카드에 적합한 SET 프로토콜은 다른 단순한 온라인 프로토콜과는 달리 연산이나 통신을 이용하여 보안에 신경을 썼지만, 여전히 완벽한 익명성 제공이나, 부인 방지 등은 제공하지 못하고 있다.

최초의 익명성이 보장되는 오프라인 전자화폐는 Chaum 등이 제안하였는데[5], 분할선택 방식을 이용하여 전자화폐의 이중 사용자 추적을 실현하였지만, 이 방식은 저장할 데이터의 양과 통신량이 증가하는 단점이 있다. 또, Chan, Frankel, Tsiounis 등은 RSA의 보안을 이용한 오프라인 전자화폐 시스템을 제안하였는데, 여기서의 사용자 익명성은 RSA의 보안에 의존한다.[7] 그리고 2000년, David Pointcheval는 다시 암호화하는 방식을 이용하여 사용자의 익명성을 조절할 수 있는 지불 시스템을 제안하였으며[9] , 2002년 Wang 등은 앞서 제안한 방법을 조금 더 개선한 익명성 조절이 가능한 전자화폐 시스템을 제안하였다.[8]

본 논문에서는 오프라인 전자화폐 시스템을 목표로, 사용자 자신이 원하는 정도의 익명성을 얻을 수 있고, 불법 사용에 대해서 사용자 추적이 가능한 전자지불 프로토콜을 제안하였다. 본 논문에서는 Wang, Cao, Kambayashi의 전자화폐 시스템[8]을 기본 모델로 삼아, 타원곡선 암호 기법을 적용하여 더욱 효용성을 증가시켰다.

II. 본문

1. 사용되는 암호학적 기법

1) 타원곡선 암호법

타원곡선 암호 시스템은 타원곡선 위의 점 P를 x번 더하는 계산이 주를 이룬다. 즉, $Q=xP$를 구하는 더하기 연산, 혹은 스칼라 곱 연산이 바로 그것이다. 여기서 점 Q와 점 P를 알고 있어도, 점 P를 몇 번 더해야 점 Q가 되는지를 알 수 없다는 문제가 바로 타원곡선 이산대수 문제(ECDLP)이다. 타원곡선 암호법은 바로 ECDLP에 기반한다. 그림 1은 본 논문에서 사용할 타원곡선

암호법이다.

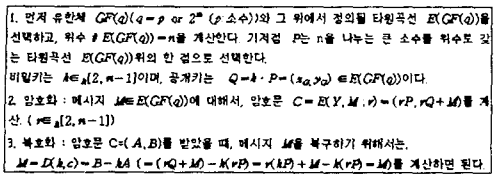

그림 1 : 타원곡선 암호법

2) Schnorr 서명

부인방지를 위한 서명 방법은 1990년에 Chaum과 van Antwerpen에 의해 제안되었다.[2] 본 논문에서 사용할 서명 방법은 1991년에 제안한 Schnorr의 서명방법[4]의 ECC버전을 쓰려고 한다.

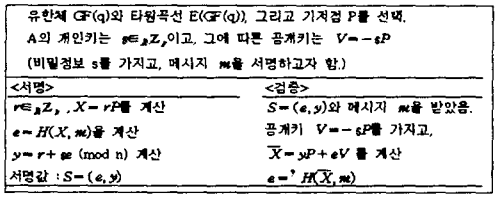

그림 2 : ECC버전의 Schnorr 서명

3) 전자화폐의 유효성 검증 방법

$Y=x_BP$는 은행의 공개키이고, $I=x_uP$는 사용자의 공개키이며, $H(x, y)$는 위와 마찬가지로 해쉬 함수이다. 여기에서의 전자화폐 c는 역시 I로 암호화된 것이다 ; $c=(A, B)=(rP, rY+sI)$, $r, s\in_R Z_n^*$. 은행은 전자화폐 c에 대해서 유효하다는 인증서 $Cert_c$를 함께 제공한다. 여기서 전자화폐 c에 대한 소유를 증명하기 위해서는 자신의 개인키 x_u와 임의로 선택되어진 수 r, s를 확인하여야 한다.

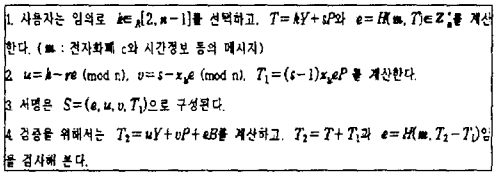

그림 3 : $c=(rP, rY+sI)$의 유효성 검증

2. 익명성 제어 가능한 전자화폐

전자 화폐 시스템은, 은행(B), 고객(C), 상점(S)의 3가지 구성요소로 이루어져 있으며, 주요 프로토콜로는 인출 프로토콜, 예치 프로토콜, 지불 프로토콜의 3가지가 있다. 여기서 고객과 상점은 은행에 계좌를 가지고 있고, 먼저 고객이 은행에 '인출 프로토콜'을 이용하여, 전자화폐를 발급 받은 후, 상점에 '지불 프로토콜'을 이용하여 물건을 구매하고, 이후 상점은 은행으로부터 그에 합당한 돈을 '예치 프로토콜'을 이용하여 자신의 계좌에 입금 받는 형태를 취한다.

그러나 새로운 전자화폐 프로토콜은 기존의 시스템에서 AP(Anonymity Provider)가 추가되는 형태를 지닌다. 여기서 사용자는 은행으로부터 인출 받은 전자화폐를 그냥 사용할 수도 있고, 사용자가 좀더 높은 익명성을 제공받기를 원한다면, AP를 통해서, 추적 불가능한 익명성이 제공되는 전자화폐를 얻을 수 있다.

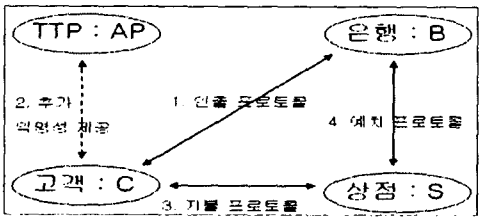

그림 4 : 새로운 전자화폐 프로토콜

1) 시스템 초기화

먼저 사용할 유한체 $GF(q)$ ($q=p$ or 2^m)를 선택하고, 그에 따른 타원곡선 $E(GF(q))$를 선택한다. 그리고 그 타원곡선의 위수 # $E(GF(q))=n$을 계산하여, 기저점 P는 n을 나누는 큰 소수를 위수로 갖는 타원곡선 $E(GF(q))$위의 한 점으로 선택한다.

은행은 임의로 $x_B\in_R Z_n^*$를 선택하고, 그에 따른 공개키 $Y=x_BP$를 등록한다. 그리고 적절한 해쉬 함수 H를 선택한다. 사용자 역시 임의로 비밀키 $x_u\in_R Z_n^*$를 선택하고, 자신의 공개키 $I=x_uP$를 등록한다.

2) 인출 과정

사용자는 은행의 공개키 $Y=x_BP$와 자신의 공개키인 $I=x_uP$와 임의로 2와 n-2사이의 정수에서 선택되어진 값 r, s를($r, s\in_R Z_n^*$) 가지고 전자화폐 $c=(A, B)=(rP, rY+sI)$를 만든다. 그리고 나서 전자화폐 c에 날짜 정보 등을 포함하여, 자신의 개인키 x_u를 가지고 서명 S를 한다. 그 후 그 둘을 r, I와 함께 은행측에 보낸다. 은행은 바른 암호화인지 확인한 후 사용자의 계좌를 수정하고, 사용자에게 전자화폐 c에 대한 인증서 $Cert_c$를 발행하여 보내준다. 사용자는 ($r, s, Cert_c$)를 저장한다.

3) 익명성 제공(AP) 과정

사용자가 익명성이 제공되는 전자화폐를 원한다면, 은행이 보낸 이 전자화폐를 그냥 사용해서는 안된다. 왜냐하면 사용자의 정보인 I와 그에 따른 $Cert_c$의 정보가 은행에 저장되어 있기 때문이다. 따라서 이 문제를 해결하기 위해서 AP가

필요하게 되었다.

먼저 사용자는 $\rho \in_R Z_n^*$ 를 선택하고, 전자화폐 $c=(A,B)$를 $c'=(A',B')=(\rho P+A, \rho Y+B)$ 으로 다시 암호화한다. 그 다음, 사용자의 비밀키 x_u를 이용하여 $M=\rho U$(U : AP에 의해 공개된 점)에 서명 S'을 한다. (여기서 $k, r, s \in_R Z_n^*$)

$$S'=(H(M, kY+sP), k-(r+\rho)e, s-x_u e, T_1)$$

바로 이 서명 때문에, 사용자가 ρ를 알고 있다는 사실로 서명을 부인할 수 없게 된다. 게다가 사용자의 비밀키 x_u를 이용하여 암호화하였기 때문에, 은행은 물론이고, 그 어느 누구도 이 단계에서 사용자로 위장할 수 없다. 이렇게 계산된 c, c', S'와 M을 AP에게 보내면, AP는 $Cert_c$를 가지고 c와, 메시지 M에의 S'의 서명의 유효성을 확인한 다음, 새로운 전자화폐 c'에 대한 새로운 인증서 $Cert_{c'}$을 만들어 사용자에게 전달한다. 그리고 c와 c'의 연계를 위해 AP는 (c, c', M, S')를 저장한다.

4) 지불 과정

사용자가 전자화폐를 이용하여 물건에 대한 지불을 상점에게 하는 것으로, 전자화폐의 유용성 검증은 전자화폐 c 또는 c'과 연관된 비밀키 (x_u, s)를 알고 있느냐를 보이는 것으로 증명한다. 새 인증서 $Cert_{c'}$, 구입, 날짜 등등에 대해서 상점에게 그 전자화폐와 관련된 비밀키 (x_u, s)로 서명 $S=(e, u, v, T_1)$을 하는 것으로 증명이 되는 것이다. 서명에 들어가는 메시지에 시간정보가 들어있기 때문에, 만약 서명의 내용을 고치기 위해서는 비밀키 (x_u, s)가 필요하다.

5) 예치 과정

이 시스템은 off-line이기 때문에, 상점은 나중에 지불 정보를 은행에 보내게 된다. 그 정보는 전자화폐 c와 서명정보 $S=(e, u, v, T_1)$ (또는 c'과 S'), 그리고 그 지불이 일어난 시간정보 등으로 구성된다. 은행은 올바른 지불정보가 맞는지를 검증한 다음, 적절한 금액을 상점의 계좌에 이체해주면 된다.

6) 불법 사용에 대한 익명성 철회

만약 같은 전자화폐를 2번 사용하였다고 하면, 사용자의 신원정보가 드러나게 된다. 만약 두 상점에 같은 전자화폐의 정보가 전달되었다고 하면, 은행은 쉽게 그 전자화폐가 이전에 사용되었던 것임을 알아낼 수 있다.

3. 안전성 분석

사용자가 같은 전자화폐를 두 번 사용했다고 가정해보자. 즉, c_1과 c_2는 같은 전자화폐를 다르게 표현한 것이라고 한다면, 그 화폐는 둘 다 모두는 사용되어질 수 없다. 먼저 사용자가 c_1의 전자화폐를 인증서 $Cert_{c_1}$를 가지고 사용했다고 하면, 이때 이와 관련된 서명 $S_1=(e_1, u_1, v_1, T_{11})$ 또한 상점으로 보내게 된다. 또 만약 사용자가 또 다른 전자화폐 c_2를 인증서 $Cert_{c_2}$와 서명 $S_2=(e_2, u_2, v_2, T_{12})$를 가지고 또 사용하였다고 하면,

$$u_1=k_1-(r_1+\rho)e_1, \quad v_1=s-x_u e_1 \pmod n,$$

$$u_2=k_2-(r_2+\rho)e_2, \quad v_2=s-x_u e_2 \pmod n$$

이므로, $(v_2-v_1)/(e_1-e_2)=x_u$이고, 따라서 사용자의 비밀키가 드러나게 된다.

그렇다면, 예전의 전자화폐와 익명성을 제공받은 후의 새 전자화폐를 사용하고자 한다고 해보자. 여기서도 이것들과 관련된 서명 $S_1=(e_1, u_1, v_1, T_{11})$과 $S_2=(e_2, u_2, v_2, T_{12})$을 상점에 보내야 하며,

$$u_1=k_1-r_1 e_1, \quad v_1=s-x_u e_1 \pmod n,$$

$$u_2=k_2-(r_2+\rho)e_2, \quad v_2=s-x_u e_2 \pmod n$$

이므로, 역시 $(v_2-v_1)/(e_1-e_2)=x_u$이고, 따라서 사용자의 비밀키가 드러나게 된다. 그러므로, 이 전자화폐 시스템은 재사용이 불가능하다.

또, n번의 지불 정보를 가지고 있다고 해도, $n+1$번째의 유효한 다른 지불정보에 대해서는 알아낼 수 있는 p.p.t TM이 존재하지 않아야 한다. 그 이유는 적법한 사용자의 경우에는 비밀키 x_u와 s가 절대로 드러나지 않기 때문이다. 게다가 임의의 수로 선택되는 s값은 각각의 전자화폐마다 다르기 때문에 n번 모두 다르게 된다. 따라서 $n+1$번째의 유효한 전자화폐를 추측해낼 수 없다.

세 번째로, 위조 가능성에 대해 생각해보자. 유효한 전자화폐를 만들어내기 위해서는 I로 암호화하는 전자화폐 $c=(rP, rY+sI)$와 전자화폐 c와 시간 정보에 대한 서명 S에 사용되는 사용자의 개인키 x_u이다. 은행에서는 I로 암호화한 전자화폐 c를 얻을 수는 있지만, 사용자의 개인키 x_u를 얻을 수는 없기 때문에 전자화폐를 위조해낼 수가 없다. AP의 경우 AP가 알고 있는 것은 c, c', M, S'인데 여기서 M에 대한 서명값 S'은 사용자의 비밀키 x_u를 이용하여 암호화한 것이므로 알아낼 수 없기 때문에, 역시 유효한 전자화폐를 위조해낼 수 없다.

마지막으로, 사용자가 전자화폐를 만들 때 자신을 제외하고는 그 누구도 알 수 없는 자신의 비밀키인 x_u와 s를 사용하기 때문에, 아무도 그 화폐로부터 사용자를 추적해낼 수도 없다.

III. 결론

지금까지 사용자의 환경에 따라서 다른 정도의 익명성을 제공해 줄 수 있는 전자화폐 프로토콜에 대해서 논의하였다. 사용자는 소액의 전자화폐를 사용하거나, 급한 경우 익명성을 제공받지 않은 상태에서도 전자화폐를 사용할 수 있으며, 원할 때에는 자신의 개인정보 노출 없이도 AP를 이용하여 더 높은 수준의 익명성을 제공받을 수 있다. 또한 통신과 계산에 필요한 부하가 적고, 익명성 조절이 되고, 이중 사용되었을 경우에는 익명성 취소도 가능하며, 위조·위장 등의 공격에 안전한 오프라인 전자화폐 시스템이다. 마지막으로, 기존에 제안되었던 [8], [9]과 비교해 보았을 때, 기존의 전자화폐 시스템들은 모두 DLP문제에 기반하였으나, 본 논문은 ECDLP문제에 기반하여, 메모리 요구와 연산 부하를 줄였다.

참고문헌

[1] D. Chaum, "Blind Signature for untraceable payments", In Advances in Cryptology: Porc. of CRYPTO'82, Plenum Press, 1983

[2] D. Chaum , H. Van Antwerpen, "Undeniable signatures", In Advances in Cryptology - Crypto'89, Springer-Verlag, 1990.

[3] D. Chaum, A. Fiat, and M. Noar, "Untraceable Electronic Cash", In Advances in Cryptology-Proc. of CRYPTO'88, LNCS Vol.403, Springer-Verlag, 1989

[4] C. P. Schnorr, "Efficient Signature Generation by Smart Cards", Journal of Cryptology, 1991.

[5] M. Franklin, M. Yung, "Secure and Efficient off-line digital money", In Proceedings of the 20th International Colloquium on Automata, Languages and Programming, Springer-Verlag, 1993.

[6] S. von Solms and D. Naccache, "On Blind Signatures and Perfect Crimes", Computers and Security, 1992

[7] A. Chan, Y. Frankel, and Y. Tsiounis, "An efficient off-line electronic cash scheme as secure as RSA", Research report nu-ccs-96-03, Northeastern Univ, 1995

[8] H. Wang, J. Cao, and Y. Kambayashi, "Building a consumer scalable anonymity payment protocol for Internet purchases", In Proceedings of the 12th International Workshop on research issues in Data Engineering : Engineering e-Commerce/e-Business Systems, IEEE, 2002.

[9] D. Pointcheval, "Self-scrambling anonymizers", In Proceedings of Financial Cryptography, Springer-Verlag, 2000.

[10]Y. Frankel Y. Tsiounis, and M. Yung "Fair off-line cash made easy", In Advances in Cryptology-Asiacrypt'98, Springer-Verlag, 1998