International Journal of Cryptology Research 3(1): 15-26 (2011)

Secure Electronic Cash Using Elliptic Curve Cryptography Based on Zero Knowledge Proof

[1,2]Khalid. O. Elaalim and [2]Shoubao Yang

[1]Department of Statistic and Computer Science
Faculty of Applied Sciences, Red Sea University, Port Sudan, Sudan

[2]School of Computer Science and Technology
University of Science and Technology of China, Hefei, Anhui, China

Email: [1]osman64@mail.ustc.edu.cn and [2]syang@ustc.edu.cn

ABSTRACT

The security of electronic cash is an important research topic to push electronic cash into practice. However, the double-spending problem will cause a great loss to the bank and spend electronic cash anonymity will be problem to the user. In this paper we present new electronic cash system using elliptic curve cryptography based on zero knowledge proof. In this scheme the user generates two secret keys such that one of them can be revealed by the bank when double spending is occurred and another cannot be revealed. The benefit of this scheme is that, the user can not open an account again when the coin is spent more than once because the bank can reveal only one secret key, which is not enough to get another secret key of the user. Another benefit of this scheme is secure blind electronic cash. Since electronic cash is secure blind then the bank cannot get any information about electronic coin (zero knowledge proof). This scheme is also unforgeability and unlinkability.

Keywords: electronic cash, elliptic curve, blind signature, zero knowledge proof, protect double spending.

INTRODUCTION

Electronic cash

Electronic cash is a specific kind of electronic payment scheme, defined by certain cryptographic properties. The simple electronic cash system consists of three parties: banks, clients, and merchants. Clients and merchants have accounts at banks. The money is transferred from the client's account to the merchant's account by using three cryptographic protocols: a withdrawal protocol with which the client withdraws coins against his account at the bank, a payment protocol with which the client pays coins to the merchant, and a deposit protocol with which the merchant deposits coins to the bank. Electronic cash system has been widely discussed in recent years. The first

e-cash scheme was proposed in 1982 by Chaum. Generally, the efficient electronic cash must satisfy the following properties:

1. Unforgeability: it means that only legal signer can generate valid signatures. In the electronic cash system means that only bank can issue electronic coins and only legal user can withdraw electronic coins from his account and deposit electronic cash into his account.
2. Unlinkability: it means that it is very hard to decide whether two different valid signatures were computed by the same signer. In electronic cash that means it is difficult for the bank to determine whether any pair of payments is executed by the same customer, unless the payments cause over spending.
3. Anonymously payment: it means that given a valid signature, it is computationally infeasible to find the identity of the signer without knowing the secret key. In electronic cash means that bank has no way of tracing electronic coin.
4. Protect double spending: it means that spend electronic coin more than one time from the users or from bank misused is prohibited

Recently, many electronic cash systems were proposed (Chaum *et al.* (1988); Chaum (1989); Brands (1993); Okamoto and Ohta (1991); Okamoto (1995)). All of them work to provide anonymous electronic cash and protect double spending, some of them are efficient and others are not.

Blind signatures

Blind signatures are variants of digital signature scheme that can not allow signer to know any information about the message actually he signed. A secure blind signature scheme must satisfy the unforgeability and unlinkability properties. The blind signature provides both anonymity and unlinkability. The first blind signature scheme was proposed by Chaum (1982). Chaum *et al.* (1988) proposed untraceable electronic cash using cut and choose method for blind electronic coin. Okamoto and Ohta (1991) were the first to attempt an improvement on this system. They modified the model by moving the most complex part of the functionality of the withdrawal protocol, namely the zero-knowledge proof of the user's identity, to the user setup (account establishment protocol, which was executed much less frequently).

Protect electronic cash from double spending

In recent years a lot of schemes protecting double spending of electronic cash have been proposed (Camenisch *et al.* (1995); Popescu (2006); Lee

(2003)). Most schemes that proposed preventing double spending would reveal user identity. However, sometimes double spending occurs due to bank misused. Nyang and Song (1999) proposed digital cash that can prevent double spending without revealing user identity. In their work the user has two secret keys, one can be revealed when double spending is occurred but another can not. The benefit of Nyang and Song (1999) is that the user does not need to open an account again in the bank.

Our contribution

In this paper we will propose new electronic cash based on elliptic curve cryptography using zero knowledge proof to provide user with anonymity payment electronic cash and protect double spending from dishonest user. We will propose secure blind signature that provides anonymously payment and unlinkability electronic cash.

NOTION AND BUILDING BLOCK

Through this paper, we denote an elliptic curve E defined over Z_q where q is a prime number greater than three and $Z_q = \{0,1,2,...,q-1\}, Z_q^* = Z_q / \{0\}$. The number of points in $E(Z_q)$ should be divisible by a large prime n (Johnson and Menezes (2000)). Two hash functions H, H_0 are employed. H is used for the construction and verification of signature of bank, and H_0 is used for challenge and must be computed in payment protocol. $\|$ denotes concatenation $G, G_1, G_2 \in E(F_q)$.

Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given an elliptic curve E defined over a finite field Z_q, a point $P \text{ and } Q \in E(Z_q)$ of order n, find the integer $x \in [0, n-1]$ such that $Q = xP$. The integer x is called the discrete logarithm of Q to the base P denoted as $x = \log_P Q$. If x is sufficient large, then it is infeasible to compute it.

ZERO KNOWLEDGE PROOF (ZKP) OF ELLIPTIC CURVE

First Zero knowledge was proposed by Goldwasser *et al.* (1989).

Chaum and Pedersen (1993) proposed Wallet Databases with Observers, which used zero knowledge proof based on RSA algorithm. Camenisch *et al.* (1995) proposed electronic commerce using elliptic curve cryptography based on ZKP.

ZKP is an interactive proof that allows a prover to prove secret information to a verifier without revealing it. By general agreement, a zero-knowledge proof should satisfy three basic properties:

1. Completeness: A prover who knows the secret information can prove it with probability 1.
2. Soundness: If a prover does not know the secret information, then the probability of verifier accepting is $\leq \frac{1}{3}$.
3. Zero-knowledge: if the information is true, no cheating verifier learns anything other than this fact.

In the zero-knowledge proof, let $G_1, G_2 \in E(F_q)$ and $x \in Z_q^*$, $P_1 = xG_1$ and $P_2 = xG_2$. The prover and the verifier perform the following zero-knowledge proof:

$$ZKP\{(x) : P_1 = xG_1 \; and \; P_2 = xG_2\}.$$

The details of the zero-knowledge proof are as follows:

Step1: the prover chooses $s \in Z_q^*$ and computes

$t_1 = sG_1 \quad and \; t_2 = sG_2$

the prover sends t_1, t_2 to the verifier.

Step2: the verifier chooses $u \in Z_q^*$ as challenge and sends it to the prover.

Step3: the prover computes and sends $r = s + ux$ to the verifier.

Step4: the verifier accepts r if and only if $t_1 \stackrel{?}{=} rG_1 - uP_1 \; and \; t_2 \stackrel{?}{=} rG_2 - uP_2$

ELECTRONIC CASH SYSTEM

The setup system

The simple electronic cash system consists of three parties: banks, clients, and merchants.

Bank setup protocol

The bank selects random number $x_B \in Z_q^*$ as private key and also chooses three points (G, G_1, G_2) on elliptic curve, computes $P = x_B G, P_1 = x_B G_1$ and $P_2 = x_B G_2$ and publishes $(E(F_q), n, q, G, G_1, G_2, P, P_2, H)$.

User setup protocol and opening an account

The user selects two random number $x_1, x_2 \in Z_q^*$ as private keys, the first one for opening account but can not be revealed when double spending is occurred, and another one can be revealed when the user does something illegal. And then the user computes two public keys $I_1 = x_1 G \text{ and } I_2 = x_2 G.$ When user wants to open an account with bank, the user sends public keys with identification (identity card, identity license or passport) to the bank. The bank keeps the user public keys ($I_1 \text{ and } I_2$) with identification in its database. Bank computes $z_1 = x_B(I_1 + G_1) \text{ and } z_2 = x_B(I_2 + G_1)$ and sends them back to the user.

Note: $z_1 \text{ and } z_2$ can be also computed by user self as $z_1 = x_1 P + P_1 \text{ and } z_2 = x_2 P + P_1.$

Merchant setup

The merchant selects random number $x_s \in Z_q^*$ as private key and calculates public key $P_s = x_s G.$

Withdrawal protocol

The withdraw protocol involves user and bank, in which user withdraws an electronic coin from bank. When the user wants withdraw electronic coin from bank, following steps must be performed:

Step1: The bank selects random number $w \in Z_q^*$ and computes $a = wG,$ $b_1 = w(I_1 + G_1) \text{ and } b_2 = w(I_2 + G_2).$ The bank sends a, b_1, b_2 to user.

Step2: User generates $s_1, s_2, u_1, u_2 \in Z_q^*$ and calculates

$A = s_1(I_1 + G_1), \quad B = u_1 G_1 + u_2 G_2, \quad C = s_2(I_2 + G_2), \quad Z_1' = s_1 z_1$
$and\ Z_2' = s_2 z_2$

Users also generates $u, v \in Z_q^*$, computes

$a_1 = ua + vG,\ F = s_2 G + uvG_2,\ w_1 = s_1 u b_1 + vA$ and $w_2 = s_2 u b_2 + vC$
and then computes

$$c = H(A, B, C, Z_1', Z_2', F, a_1, w_1, w_2)$$

User sends $c' = c / u$ as challenge to the bank.

Step3: The bank sends response $r = c'w + x_B a$ to the user.

Step4: The user accepts response $rG \stackrel{?}{=} (c' + P)a$ and $r(I_1 + G_1) \stackrel{?}{=} c'b_1 + az_1$,
$r(I_2 + G_2) \stackrel{?}{=} c'b_2 + az_2$

If they hold, then the user computes $r' = ru + v$.

Proposition 1. During the withdrawal protocol, User can accepts response if $rG \stackrel{?}{=} (c' + P)a$ and $r(I_1 + G_1) \stackrel{?}{=} cb_1 + az_1$, $r(I_2 + G_2) \stackrel{?}{=} cb_2 + az_2$.

Proof:

$$\begin{aligned} rG &= (c'w + x_B a)G \\ &= c'wG + ax_B G \\ &= c'a + aP \\ &= (c' + P)a \end{aligned}$$

$$\begin{aligned} r(I_1 + G_1) &= (c'w + x_B a)(I_1 + G_1) \\ &= c'w(I_1 + G_1) + ax_B(I_1 + G_1) \\ &= c'b_1 + az_1 \end{aligned}$$

$$\begin{aligned} r(I_2 + G_2) &= (c'w + x_B a)(I_2 + G_2) \\ &= c'w(I_2 + G_2) + ax_B(I_2 + G_2) \\ &= c'b_2 + az_2 \end{aligned}$$

□

Payment protocol

The payment protocol involves user and merchant, in which user pays an electronic coin to merchant. When the user wants to pay for goods from

merchant, following steps must be performed:

Step1: User generates $y_1, y_2 \in Z_q^*$ and computes $D = y_1G_1 + y_2G_2$
User sends A, B, C, D, F to merchant

Step2: Merchant computes challenge d and sends to user
$d = H_0(A, B, C, D, I_s, amount, amount\,type, date / time)$

Step3: User calculates responses

$$r_1 = s_1 + dy_1, \qquad r_2 = d(s_2 + y_2)$$
$$r_3 = s_1x_1 + ds_2x_2, \qquad r_4 = d(s_1x_1 + s_2)$$
$$r_5 = ds_1 + u_1, \qquad r_6 = duv + u_2$$

User sends them to merchant.

Step4: Merchant accepts if and only if

$$r_1G_1 + r_2G_2 + r_3G \stackrel{?}{=} A + d(D + C)$$
$$r_4G + r_5G_1 + r_6G_2 \stackrel{?}{=} d(A + F) + D$$

Proposition 2. During the payment protocol, shop can accepts response if $r_1G_1 + r_2G_2 + r_3G \stackrel{?}{=} A + d(D + C)$ and $r_4G + r_5G_1 + r_6G_2 \stackrel{?}{=} d(A + F) + D$

Proof:
$$\begin{aligned} r_1G_1 + r_2G_2 + r_3G &= (s_1 + dy_1)G_1 + (d(s_2 + y_2)G_2) + (s_1x_1 + ds_2x_2)G \\ &= s_1G_1 + dy_1G_1 + ds_2G_2 + dy_2G_2 + s_1x_1G + ds_2x_2G \\ &= s_1(x_1G + G_1) + ds_2(x_2G + G_2) + d(y_1G_1 + y_2G_2) \\ &= A + dC + dD \end{aligned}$$

$$\begin{aligned} r_4G + r_5G_1 + r_6G_2 &= (d(s_1x_1 + s_2))G + (ds_1 + u_1)G_1 + (duv + u_2)G_2 \\ &= ds_1x_1G + ds_2G + ds_1G_1 + u_1G_1 + duvG_2 + u_2G_2 \\ &= ds_1(x_1G + G_1) + d(s_2G + uvG_2) + u_1G_1 + u_2G_2 \\ &= dA + dF + B \end{aligned}$$

□

Deposit protocol

The deposit protocol involves the merchant and the bank as following:

Step1: The merchant sends $A, B, C, D, F, r_1, r_2, r_3, r_4, r_5$ *and* r_6 to the bank.

Step2: The bank verifies validity of the electronic coin as proposition 2.

Step3: The bank checks whether the coin has been spent. If the coin was not spend yet then the bank accepts the electronic coin and deposits it to the merchant account.

SECURITY ANALYSIS

In this section, we will discuss the security properties of the electronic cash system: anonymity,unforgeability, unlinkability and protect double spending.

Anonymity

The user can make anonymous payment to the merchant; in this scheme only the bank knows the identity of the E-cash, which is confidential to the merchant. In the payment protocol, the merchant receives e-cash from the user. The merchant can only verify the validity of signatures, but could not determine the identity of signer. Then the user can spend electronic coin anonymously.

Proposition 3. It is impossible for the bank to get any information about electronic coin if it knows secrete key x_1.

Proof: If the user wants to make coin, he needs to select random numbers s_1, s_2, u_1, u_2, u *and* v and calculates A, B, D, F, a_1, w_1 *and* w_2.
This is impossible for bank to get them, because they are selected by the user.□

Unforgeability and unlinkability

To be able to forge a coin, any coalition user must be able to give proofs of knowledge without knowing the witness. However this contradicts the properties of the signature scheme of the bank, and of the non-interactive ZKP. In withdrawal protocol no one can withdraw e-coin except the user who is the account owner, see proposition 5.

Proposition 4. Only the bank is able to issue electronic cash.

Proof: When user wants to withdraw coin from the bank, the bank generates

random number $w \in Z_q^*$ and then computes and sends $a, b_1 \text{ and } b_2$ to the user. Since w is elliptic curve discrete logarithm, so it is very hard to calculate w. Furthermore, there is also another elliptic curve discrete logarithm, which is the bank private key x_B.

No one is able to get $w \text{ and } x_B$, then only bank can be able to issue electronic coin. □

Proposition 5. No one can allege to be the other users and withdraw e-cash from bank (even the bank).

Proof: In the withdrawal protocol, the user needs to implement an authentication protocol with the bank while no other one knows the user's private keys $x_1 \text{ and } x_2$, , if any other user including the bank wants to withdraw e-cash, he needs $x_1 \text{ and } x_2$ which is computationally infeasible to calculate $x_1 \text{ and } x_2$ from $I_1 = x_1 G \text{ and } I_2 = x_2 G$ respectively.

Then only the exact user can take out e-cash from the bank. □

Proposition 6. If the blind signature is secure then electronic cash is unlinkability.

Proof: In the withdrawal protocol, if the user sends two challenges $c' \text{ and } c_1'$ to the bank, then the bank can not decide these two challenges made by the same user, because the bank has no information about them (zero knowledge). □

Protect Double Spending

Double spending is to spend the coin more than once. In the deposit protocol the merchant sends the transcript of the execution of the payment protocol to the bank which verifies that coin. If this is successful, the bank checks double spending. If the bank finds the coin in deposit database then the bank can trace the user, see following proposition 7. Not all of double spending is made by user but some times it occurs due to the bank misused. Also we should protect the user from the bank misused.

Proposition 7. If the user spends coin two times then the bank can trace the user.

Proof: In the payment protocol after the merchant sends challenge d to the user, the user calculates responses $r_1, r_2, r_3, r_4, r_5 \text{ and } r_6$ and sends them to the merchant. If the user wants to spend the same coin twice then the merchant should send to user another challenge d' and user calculates other responses $r_1', r_2', r_3', r_4', r_5' \text{ and } r_6'$ to the merchant. In the deposit protocol the merchant sends $r_1, r_2, r_3, r_4, r_5 \text{ and } r_6$ and $r_1', r_2', r_3', r_4', r_5' \text{ and } r_6'$ to the bank, then the bank can reveal user private key x_1 as following:

$$r_4 = d(s_1 x_1 + s_2) \tag{1}$$

$$r_4' = d'(s_1 x_1 + s_2). \tag{2}$$

By subtracting equation (2) from (1), we get

$$r_4 - r_4' = s_1 x_1 (d - d') \tag{3}$$

and

$$r_5 = d s_1 + u_1 \tag{4}$$

$$r_5' = d' s_1 + u_1 \tag{5}$$

By subtracting equation (5) from (4), we get

$$r_5 - r_5' = s_1 (d - d')$$

$$s_1 = (r_5 - r_5') / (d - d') \tag{6}$$

From (3) and (6) we get

$$x_1 = (r_5 - r_5') / s_1 (d - d')$$

Then x_1 is one private key of the user. □

Proposition 8. Electronic cash can protect user from the bank misused.

Proof: If the bank knows one secret key of the user, the bank is not able to withdraw electronic coin because he needs another secrete key.

CONCLUSION

We have proposed secure electronic cash using elliptic curve

cryptography based on zero knowledge proof, and each part is fully described with proving the important equations. In this work user generates two random secret key, one can be revealed when double spending occurs and another can't. This scheme achieves protecting double-spending. We use zero knowledge protocol to provide user anonymity and unlinkability. Analysis shows that this protocol has good security anonymity, unforgeability, unlinkability and prevents double-spending. We hope this scheme is suitable for the development of the e-cash.

REFERENCES

Brands, S. (1993). Untraceable Off-line Cash in Wallets with Observers. *Advances in Cryptology-CRYPTO'93, LNCS 773*. Springer-Verlag: 302-318.

Camenisch, J. L., Piveteau, J. M. and Stadler, M. A. (1995). Fair blind signatures. Advances in Cryptology, EUROCRYPT'95, *Lecture Notes in Computer Science*. Springer-Verlag. **921**: 209–219.

Chaum, D. (1982). Blind Signature for untraceable Payments. *Advances in Cryptology-Crypto' 82*. New York: Plenum Press: 199-203.

Chaum, D., Fiat, A. and Naor, M. (1988). Untraceable Electronic Cash. *Advances in Cryptology - CRYPTO '88, LNCS 403*. Springer Verlag: 319-327.

Chaum, D. (1989). Online Cash Checks. *Advances in Cryptology-EUROCRYPT'89, LNCS 434*. Springer-Verlag: 288-293.

Chaum, D. and Pedersen, T. (1992). Wallet Databases with Observers. *Advances in Cryptology - CRYPT0 '92, LNCS 740*. Springer-Verlag Berlin Heidelberg: 89-105.

Goldwasser, S., Micali, S. and Rackoff, C. (1989). The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*. **18** (1): 186–208.

Johnson, Don B. and Menezes, Alfred J. (2000). Elliptic Curve DSA (ECDSA): An Enhanced DSA. *Available (http://www.certicom.com/).*

Lee, H., Choi, M. and Rhee, C. (2003). Traceability of double spending in secure electronic cash system. *Proceeding of the 2003 International Conference on Computer Networks and Mobile Computing (ICCNM'03).*

Nyang, DaeHun and Song, JooSeok. (1999). Preventing Double-Spent Coins from Revealing User's Whole Secret. *ICISC'99, LNCS 1787*, Springer-VerlagBerlin Heidelberg: 30–37.

Okamoto, T. and Ohta, K. (1991). Universal Electronic Cash. *Advances in Cryptology - CRYPTO '91, LNCS 576*. Springer-Verlag: 324-337.

Okamoto, T. (1995). An Efficient Divisible Electronic Cash Scheme. *Advances in Cryptology - CRYPTO'95, LNCS 963*. Springer-Verlag: 438-451

Popescu, C. (2006). An Electronic Cash System Based on Group Blind Signatures. *INFORMATICA*. **17**(4): 551–564.

Sultan Almuhammadi, Sui, Nien T. and McLeod, D. (2004). Better Privacy and Security in E-Commerce: Using Elliptic Curve-Based Zero-Knowledge Proofs. *Proceedings of the IEEE International Conference on E-Commerce Technology*.