The Electronic Cash Protocol Based on Dynamic Group Signature

Jian Xu[1*], Yuxi Li[2], Jingwei Miao[3], and Fucai Zhou[1]
[1] *Software College, Northeastern University, Shenyang, Liaoning, China*
{xuj, fczhou}@mail.neu.edu.cn
[2] *College of Information Science and Engineering, Northeastern University Shenyang, Liaoning, China*
eliyuxi@gmail.com
[3] *University of Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, F-69621, France*
jingwei.miao@insa-lyon.fr

Abstract

Due to the increasingly serious information attacks in network times, the electronic commerce is facing additional challenges in the aspects of security and efficiency. Therefore, in this paper, we propose a new practical e-cash protocol based on dynamic group signature in BSZ model, using the Non-Interactive Zero-Knowledge proofs system. And it is proved that our protocol satisfies anonymity which resists chosen-ciphertext attack, unforgeability, traceability, and no double-spending without random oracles. Compared with the existing protocols, our protocol not only permits dynamic enrollment of members and non-interactive during transmission, but also offers lower storage and communication cost.

Keywords: electronic cash, group signature, BSZ model, Non-Interactive Zero-Knowledge

1 Introduction

The definition of electronic cash (e-cash) is introduced by Chaum in 1982 [1], aiming at solving the limitations of traditional money in transaction and effectively protecting users' anonymity. General protocols on e-cash includes three entities and four protocols: the entities in e-cash are *bank*, *merchant* and *user*, and the protocols involved are *open*, *withdraw*, *spend* and *deposit*. In these protocols, digital signature schemes are widely exploited. The first e-cash protocol, proposed by Chaum in 1982 [1], is based on blind signatures which make e-cash anonymous and unlinkable.

Due to the increasingly serious information attack in network times, the complete anonymity in the schemes with blind signatures result in the wantonly growing of illegality, such as laundering, illegal transaction, blackmail and so on [2]. Moreover, there is only one bank entity in charge of the distribution of all money in the most existing e-cash protocols, which is too difficult to match the multiple banks' situation in real life. For the above reasons, in 1998, Lysyanskaya and Ramzan proposed the first multiple banks e-cash protocol in the Financial Cryptography [3]. They utilized blind signature and group signature in their protocol, and opened up a new direction for further research. In group signatures, the group member is permitted to sign messages on behalf of the whole group in the meantime his personal identity is anonymous. That is, the verifier can verify whether a signature is generated by the group member or not, without learning the personal identity of the signer. Hence, compared with blind signatures, group signatures can better satisfy the security requirement of e-cash in reality.

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, volume: 4, number: 4, pp. 65-79
*Corresponding author: Information Security Institute Software College, Northeastern University, Shenyang, Lane 3, Wen-Hua Road, HePing District, Shenyang, Liaoning, 110819, China, Tel: +86-13804011518

It is worthy to note that in many e-cash protocols, it is concerned little about the insider attacks in untrusted third party, even malicious insider. Therefore, Ferguson[4] proposed a method to prevent such attacks by sign data exchanged at withdrawl protocol. However, in this method, both the user and the bank need a large storage space to refrain from disputes. In 2011, Nishide and Fukuoka[5]formally analyzed the security against insider threats in some classical e-cash protocols.

1.1 Related Works

With the continuous progress on group signatures, many cryptographic researchers use it to design e-cash. In 2001, a fair e-cash protocol was proposed by Maitland and Boyd [6], which is based on a coalition-resistance group signatures scheme. After two years, another efficient fair e-cash protocol is proposed by Canard which is a breakthrough in the research history of e-cash [7]. Except other essential security properties in standard model, it is proved traceable for double-spending. In recent years, there is a tendency that e-cash protocols are possessing specific features for specific requirements. For instance, in 2007, Canard maintained that it should be possible for e-cash to be divisible in circulation, and he proposed a divisible e-cash without TTP (trusted third party), using binary tree and limited accumulators. It allows user to deposit a specified amount of e-cash [8], which can be divided into multiple, smaller values in spend protocol. However, this method would bring about security risk, it is linkable for every small values divided from the same e-cash. Moreover, in 2008, with the cryptography tools of zero-knowledge proofs and verifiable encryption, Blanton proposed an efficient transferable e-cash protocol based on CL group signatures [9]. However, the anonymity in its deposit protocol is not strong enough. More recently, in 2011, a new off-line e-cash protocol, proposed by Eslami and Talebi, achieved not only anonymity and traceability, but perfectly fraud control [10]. And what is worth mentioning is that it make bank processing data more effectively by attaching expiration date to every e-cash, so the bank can abolish the outdated e-cash directly.

1.2 Our Contribution

Many existing e-cash protocols are interactive. Fortunately, Non-Interactive Zero-Knowledge (NIZK) proofs can improve this situation to some extent. But due to the complexity of NIZK, most researchers tend to treat it as a black box, but ignore its proof details and verification methods. Therefore it is hard to apply in reality.

Aiming at the above problems, in this paper, based on dynamic group signature with NIZK proofs system, we propose a new practical electronic cash protocol. Compared with the existing related works, we have three contributions as follows:

(1) We propose a new security model which can capture the security demands of e-cash. And give the security proof for the proposed e-cash protocol in the standard model that it is with CCA anonymity, unforgeability, traceability and no double-spending.

(2) We construct the specific proof procedure of NIZK for the proposed protocol. What is worth mentioning is that we prove and sign the blocks of messages instead of limiting the proved message to only one bit (0 or 1), which improve the proof efficiency.

(3) In terms of practical applicability, a register protocol is added in e-cash: it permits dynamic enrollment of members, and preserves anonymity of a group signature even if the adversary can see arbitrary key exposures or arbitrary openings of other group signatures; secondly, concerning the problems of the signature size, we use Groth protocol to generate keys, minimized e-cash signature size to constant value.

In the end of the paper, we compare our protocol with other existing e-cash protocols on security and performance, it turns out that ours has advantages of security and efficiency than others.

2 Preparations

2.1 Groth-Sahai Proofs System

The definition of NIZK was first introduced by Blum in 1988 [11]. It can make verifier be sure of whether a statement is true or not without disclosing any information and multiple interactive between both sides. In addition, in 2008, an efficient paring-based NIZK proofs system was constructed by Groth and Sahai in EUROCRYPT [12], which made NIZK practical and transferrable.

There are some common basic concepts of Groth-Sahai Proofs System: Commitment values $\{C_m\}_m = 1...M$ hide $\{x_m\}_{m=1...M} \in G_1$, $\{b_q\}_{q=1...Q} \in G_2$ by selecting random $\{a_q\}_{q=1...Q} \in G_1$, $\{b_q\}_{q=1...Q} \in G_2$, $\{\alpha_{q,m}\}_{q=1...Q,m=1...M \in Zp}$, $\{\beta q,n\}_{q=1...Q,n=1...N \in Zp}$, to compute $\{C_m = aq\prod_{m=1}^{M} x_m{}^{\alpha_{q,m}}\}_{m=1...M}$. The statement s is consists of all the commitments and bilinear paring product equations:

$$\prod_{q=1}^{Q} e(a_q \prod_{m=1}^{M} x_m^{\alpha_{q,m}}, b_q \prod_{n=1}^{N} y_n^{\beta_{q,n}}) = t \tag{1}$$

If given a proof π, which is related to corresponding statement s, it means to show that the paring product equations have the solutions, and the system can extract x_m and y_n ,which satisfying the equations of the statement s, more formally it can be expressed as follows:

$$\begin{aligned}\pi = \text{NIZK}\{((c_1 : x_1), ..., (c_M : x_M), (d_1 : y_1), ...(d_N : y_N)) : \\ \prod_{q=1}^{Q} e(a_q \prod_{m=1}^{M} x_m^{\alpha_{q,m}}, b_q \prod_{n=1}^{N} y_n^{\beta_{q,n}}) = t\}\end{aligned} \tag{2}$$

2.2 Group Signature Schemes Based on BSZ Model

The first dynamic group signature scheme, proposed by Bellare, Shi and Zhang [13], is available for members to join dynamically, and proved secure in standard model. In this scheme, they formally defined a strong security model (called BSZ model) for group signature.

A dynamic group signature scheme based on BSZ model consists of the following algorithms: Setup, Join, Sign, Verify and Trace. The Setup algorithm produces a pair of signing and verifying keys, a pair of encryption and decryption keys. To join a group, a user should produce a personal key pair, and obtains a certificate from the issuer, in other word a signature under the issuer's key. Any group member can generate a group signature simply by signing the message with his personal signing key, encrypting his certificate, verifying key, and this signature, and then producing those ciphertexts together with a NIZK proof that the certificate and signature in the plaintext are indeed valid. The opening is done by decrypting the ciphertexts, where the verifying key gives the user's identity and the signature corresponds to the unforgeable proof.

3 Protocol Design

3.1 Entity Constitution

In this paper, three types of entities constitute the entire e-cash protocol: a central bank, local banks and users; in addition users are divided into customers and merchants. The circulation of e-cash is as follows: The central bank issues the certificates for local banks, and records the information of the legal local banks and users for registration, for manage and revoke them. And the local bank can create e-cash and sign it anonymously to customers. Then customers send e-cash to merchants who offer goods and services. And merchants can deposit the e-cash to the bank only if it is legal. If there is a dispute between the the entities concerned, either can apply to the central bank for extracting the identity of the other one and execute the arbitration.

3.2 Security Definition

The proposed e-cash protocol satisfies correctness and secure properties as anonymity, unforgeability, traceability and no double-spending. The Definitions of anonymity and unforgeability are formally described by the interactive games between the simulator S and adversary A.

In such games, the adversary's ability to attack the target e-cash protocols is simulated by some encryption services. The adversary gain access to these services by simulator S. The properties are elaborated on as follows.

Correctness

Correctness refers to the group signature produced by the legal group member (local bank or user) is: validity, namely the verification of the signature can be done by the receiver; legality, namely any specified group member can be traced in tracing protocol; consistency, means it is sure that a group signature does belong to the group member who really generated it. So in the case that all group members are legal and customer has enough e-cash, the customer's e-cash will always be accepted by merchant, and merchant's one will be always accepted by the local bank.

Anonymity

Anonymity refers that it is hard for the group members to calculate the private key of the central bank or recover the user's identity from any e-cash.

Suppose that adversary A, not in possession of the user's secret key, performs two phases with simulation S: probing phase and challenge phase. In probing phase, on input of the bank's secret key and public parameters , A perform a bounded number of queries in polynomial time to the simulation S in an adaptive manner to such an extent to obtain the identity of the user or extract the secret key. Then in challenge phase, A output two legal identities and e-cash m to S, S run the e-cash protocols to output a signature on m. In the consequence A will find it hard to tell the signature is signed by which identity.

Unforgeability

Unforgeability refers that it is hard for any group member to forge signatures of other members.

Suppose that adversary A, not in possession of the user's secret key, performs two phases with simulation S: probing phase and output phase. In probing phase, A perform a bounded number of hash queries and signature queries in polynomial time to the simulation S in an adaptive manner. When performs hash query per time, A chooses randomly, then obtain from s; When performs signature query per time, A query S for , then S outputs by simulating the signature process; in output phase, A will find it hard to output a signature which can be accepted by the verification.

Traceability

Traceability is divided into two sides: on one side, if there exists a dispute, the central bank will definitely extract the identity of the entity in response On the other side, If the illegal users try to forge e-cash, the central bank cannot extract the identity of the legal members.

No double-spending

No double-spending refers that it is not available for user to spend the same e-cash in any two transactions, that is to say, the e-cash serial numbers in any two transactions should be different.

3.3 General Description

3.3.1 Setup Protocol

Step1: Calls BilinearSetup (1^K) to generate system parameters , where $e : G \times G \to G_T$, $G = \langle g \rangle$ and the prime p is the order of G, G_T. Choose $r, x, y \leftarrow Z_p$ randomly, set $f = g^x, h = g^y, \Omega = g^r$; and pick $(r_u, s_v) \leftarrow {Z_p}^2$, $z \leftarrow Z_p^*$ randomly, next calculate the triple $(u, v, w) = (f^{r_u}, h^{s_v}, g^{r_u + s_v + z})$ then make it public. In the end, choose random vectors $(\vec{u}, \vec{v}, \vec{w}) \in G^3$, where $\vec{u} := (u_1, u_2, u_3, ..., u_n)$, $\vec{v} := (v_1, v_2, v_3, ..., v_n)$, $\vec{w} := (w_1, w_2, w_3, ...w_n)$ define a hush function as $H : \{0,1\}^* \to \{0,1\}^n$.

Step2: The central bank choose $\alpha \leftarrow Z_n$ as its secret key, then use it to calculate its public key $\omega = (\omega_1, \omega_2) = (g^{\alpha}, g^{\alpha^2})$. $tk = (x, y, z)$ is its extract key for tracing entity if necessary. The secret key of the local banks group is $k \leftarrow Z_q^*$, and its public key is $PK = g^k$. Local bank choose $r' \leftarrow Z_q^*$ randomly to calculate its secret key $d = \{d_1, d_2\} = \{g^{\alpha r'}, g^{\alpha k} g^{r'}\}$.

3.3.2 Register Protocol

Register protocol consists of bank registration and user registration. The entities included are the central bank and the local bank/user. Any local bank/user, who wants to join in the protocol, must perform this interactive protocol with the central bank. For instance, if it is bank registration, the phase is shown as follows:

Step1: $Bank_i : (k, g^k) \leftrightarrow CentralBank : (g^k)$

By running the Groth protocol [14], bank i obtains its public key $pk := g^k$and secret key $pk := g^k$, while at the same time the central bank only obtain bank i's public key $pk := g^k$.

Step2: $CentralBank \rightarrow Bank_i : (cert_{B_i})$

The central bank choose $id \in_n^*$randomly, then generate bank i's certificate $cert_{\mathrm{B_i}} = (\sigma_1, \sigma_2)$, where $\sigma_1 = (h \cdot pk)^{1/(r+id)} \sigma_2 = g^{id}$; calculate $c_{ID} = H(\sigma_1, \sigma_2)$ and record it in the corresponding reg[ID], finally send $cert_{B_i} = (\sigma_1, \sigma_2)$ to bank i.

The central bank record it in the corresponding reg[ID].

Step3: $Bank_i \rightarrow CentralBank : (cert_{B_i})$

After obtained the certificate, bank i judge the correctness of the certificate by calculating whether the equation $e(\sigma_1, \Omega\sigma_2) = e(g,h)e(g, pk)$ is right or not. If not passed, bank i will calculate the hash value of (σ_1, σ_2), and send $c = H(\sigma_1, \sigma_2)$ to the central bank. The central bank will compare it with c_{ID} in the corresponding reg[ID], then determine whether the certificate has been tampered with in the process of transfer.The register protocol of user is the same as the phase above-mentioned, so it's no need to state more.

3.3.3 Open Protocol

The participators are local banks and users in this phase. The bank i should affirm the user's legal identity before allow him open an account in it. This phase contains the following stages:

Step1: $User \rightarrow Bank_i : (\sum)$

First, user must commit to his certificate $c_{ID} \in \{0,1\}^n$ to make sure that his certificate is unforgeable. We can assume that the length of is n, and c_i is the i-th bit in c_{ID}, then we choose the random number $(r, s) \leftarrow Z_p \times Z_p$ to calculate the commitment value of the certificate $C = (C_1, C_2, C_3) = (f^r \prod_{i=1}^n u_i{}^{c_i}, h^s \prod_{i=1}^n v_i{}^{c_i}, g^{r+s} \prod_{i=1}^n w_i{}^{c_i})$, next ,we choose $t \leftarrow Z_p$ to produce non-interactive zero-knowled ge proof (π_1, π_2, π_3) on C, as the following show:

$$\pi_1 = \begin{pmatrix} \vec{\pi}_{1,1} \\ \vec{\pi}_{1,2} \\ \vec{\pi}_{1,3} \end{pmatrix} = \begin{pmatrix} (f^r \prod_{i=1}^{n} u_i^{2c_i-1})^r \\ h^{rs-t} \prod_{i=1}^{n} v_i^{(2c_i-1)r} \\ g^{(r+s)r+t} \prod_{i=1}^{n} w_i^{(2c_i-1)r} \end{pmatrix} \tag{3}$$

$$\pi_2 = \begin{pmatrix} \vec{\pi}_{2,1} \\ \vec{\pi}_{2,2} \\ \vec{\pi}_{2,3} \end{pmatrix} = \begin{pmatrix} f^{rs+t} \prod_{i=1}^{n} u_i^{(2c_i-1)s} \\ (h^s \prod_{i=1}^{n} v_i^{2c_i-1})^s \\ g^{(r+s)r-t} \prod_{i=1}^{n} w_i^{(2c_i-1)s} \end{pmatrix} \tag{4}$$

$$\pi_3 = \begin{pmatrix} \vec{\pi}_{3,1} \\ \vec{\pi}_{3,2} \\ \vec{\pi}_{3,3} \end{pmatrix} = \begin{pmatrix} \vec{\pi}_{1,1} & \vec{\pi}_{2,1} \\ \vec{\pi}_{1,2} & \vec{\pi}_{2,2} \\ \vec{\pi}_{1,3} & \vec{\pi}_{2,3} \end{pmatrix} = \begin{pmatrix} f^{r(r+s)+t} \prod_{i=1}^{n} u_i^{(2c_i-1)(r+s)} \\ h^{s(r+s)-t} \prod_{i=1}^{n} v_i^{(2c_i-1)(r+s)} \\ g^{2r(r+s)} \prod_{i=1}^{n} w_i^{(2c_i-1)(r+s)} \end{pmatrix} \tag{5}$$

Finally, the user sends the statement $\sum := NIZK\{\pi_1, \pi_2, \pi_3, \mathrm{C}\}$ to the bank.

Step2: $Bank_i \rightarrow User : (valid/invalid)$

If the bank wants to verify the reality and legality of the identity (certificate) of a user, it means to judge whether the proof of Σ is legal, which is to verify whether the pairing-based Bilinear equation are equal:

$$e_{11} = e(f, \pi_{11}) \text{and} e_{12} = e(C_1, C_1 \prod_{i=1}^{n} u_i^{-1}) \tag{6}$$

$$e_{13} = e(f, \pi_{22}) \text{and} e_{21} = e(C_2, C_2 \prod_{i=1}^{n} u_i^{-1}) \tag{7}$$

$$e_{22} = e(f, \pi_{33}) \text{and} e_{23} = e(C_3, C_3 \prod_{i=1}^{n} u_i^{-1}) \tag{8}$$

$$e_{31} = e(f, \pi_{12}) e(h, \pi_{21}) \text{and} e_{32} = e(C_1, C_2 \prod_{i=1}^{n} v_i^{-1}) e(C_2, C_1 \prod_{i=1}^{n} u_i^{-1}) \tag{9}$$

$$e_{33} = e(f, \pi_{13}) e(h, \pi_{31}) \text{and} e_{41} = e(C_1, C_3 \prod_{i=1}^{n} v_i^{-1}) e(C_3, C_1 \prod_{i=1}^{n} u_i^{-1}) \tag{10}$$

$$e_{42} = e(f, \pi_{23}) e(h, \pi_{32}) \text{and} e_{43} = e(C_2, C_3 \prod_{i=1}^{n} v_i^{-1}) e(C_3, C_2 \prod_{i=1}^{n} u_i^{-1}) \tag{11}$$

If the statement of verifying passes, the bank will return confirmation message to the user, which means this user can access e-cash here, and the bank will also save his statement, and establish an account for him. If not, the bank will return failure message to the user.

3.3.4 Withdraw Protocol

If a user wants to withdraw an amount of e-cash m from bank *i*, he will go through the following stages:

Step1: $Bank_i \rightarrow User : (\Gamma)$

Firstly, the bank *i* will commit and prove its certificate, and produce a statement of the certificate in the same way of section 3.3.2; Secondly, the bank *i* will sign at e-cash m, and choose a random number $t' \leftarrow Z_q^*$ to calculate the signature string $\delta = \{U_1, U_2, V_1, V_2\} = \{g^{\alpha^2 \cdot t'}, g^{\alpha \cdot r' \cdot t'}, d_1^h, d_2^{t'+h}\}$, in which $h = H(m, U_1, U_2)$; At last, the bank will send $\Gamma = \{\delta, \Sigma : e(V_1, g^\alpha) = e(PK_A, U_1) e(PK_A^h, g^{\alpha^2}) e(U_2, g)$ $e(V_2, g)\}$ as the final e-cash to the user.

Step2: $User : Accept/Reject$

The user will verify the correctness of the e-cash he has obtained: First, he should verify the identity of the bank; Then he will judge the signature of this e-cash, which means to verify if the equation in $\Gamma = \{\delta, \Sigma : e(V_1, g^{\alpha}) = e(PK_A, U_1)e(PK_A^h, g^{\alpha^2})e(U_2, g)e(V_2, g)\}$ is right. If passed, the e-cash user has obtained is valid, and it can be used for spend protocol.

3.3.5 Spend Protocol

The participators in this phase are customers and merchants in users' group. If the identification of this transaction is R, and the e-cash of this transaction is the i-th e-cash the customer has spent, the following stages will be experience.

Step1: $User_C \rightarrow User_M : (M)$

We assume that the identification of this transaction is R, and the e-cash of this transaction is the i-th e-cash the customer has spent. To avoid double-spending, the customer should calculate the serial number $S = F_s(i)$ and the value of non-double-spend $T = g^{C_{ID}} \cdot F_r(i)^R$ of this e-cash spend firstly, and then add these two values to the e-cash which is to be sent, and at last send $\mathrm{M} = \{\delta, \Sigma, S, T, comm : e(V_1, g^{\alpha}) = e(PK_A, U_1)e(PK_A^h, g^{\alpha^2})e(U_2, g)e(V_2, g)\}$ as the e-cash to merchant. In fact, in this condition the identity information of customer has been added, but it will not be leaked in the process of transaction. If a customer uses the same e-cash as r, i is same in two transactions, his identity will be confirmed by using the identification R and $R'T$ and $T'$of this two transactions.

Step2: $User_M : Accept/Reject$

After the merchant obtains an e-cash, he will judge whether to accept this e-cash by the following 3 steps:

(1) Judging whether the statement and its proof $\pi_{i,j}, i = 1,2,3, j = 1,2,3$ is legal, and if legal, this e-cash's issuing bank is affirmed by the central bank;

(2)Judging whether the equation $e(V_1, g^{\alpha}) = e(PK_A, U_1)e(PK_A^h, g^{\alpha^2})e(U_2, g)e(V_2, g)$ in M is true, and if true, this e-cash is issued by the legal bank.

(3) Comparing the values of non-double-paying and in two transactions to judge if the e-cash is spend twice.

If the 3 steps above have been verified successfully, the e-cash will be accepted, and the system will clear the identification of transaction , and execute this transaction; otherwise the e-cash will be refused.

3.3.6 Deposit Protocol

Suppose bank j is another bank in local bank group. Then the participators in this phase are merchants and bank j:

Step1: The merchant needs to deposit the e-cash he gets from customers into the bank j, and the e-cash is $\Gamma = \{\delta, \Sigma, comm : e(V_1, g^{\alpha}) = e(PK_A, U_1)e(PK_A^h, g^{\alpha^2})e(U_2, g)e(V_2, g)\}$.

Step2: $User_{Merchant} \rightarrow Bank_j : (M)$

The bank j will execute the verification in double times: First it will affirm that the statement as the identity of a merchant is real and legal; Next it will judge whether the equation in M as the signature of a merchant is true. If both are true, the e-cash bank j has obtained is real and valid, and will be deposit into the account of this merchant.

3.3.7 Trace Protocol

If a bank has contradiction with a user about an amount of e-cash, the bank can apply the central bank to find out the original user who has paid this e-cash. So the participators in this phase are local

banks and the central bank:

$Bank_i \rightarrow CentralBank : (C)$

The central bank applies a local bank for the information of a user, and this local bank sends the user's values of commitment $C = (C_1, C_2, C_3) = (f^r \prod_{i=1}^{n} u_i{}^{c_i}, h^s \prod_{i=1}^{n} v_i{}^{c_i}, g^{r+s} \prod_{i=1}^{n} w_i{}^{c_i})$to the central bank. After that the central bank will use the extract-key $tk = (x, y, z)$to get c_{ID}in exhaustion through $(g^z)^{c_{ID}} = C_3 {C_1}^{-1/x} {C_2}^{-1/y}$,and then seek in reg[ID] to trace out the identity of this user who will get relevant arbitrated.

4 Security and Performance

4.1 Security Analysis

The security of the e-cash protocol is related to the hardness of the following assumptions.

Assumption 1 (TDH assumptions) On input $\{g, g^a, g^{a^2}, ..., g^{a^t}, g^{ak}, g^{a^2k}, ..., g^{a^tk}\}$ in which $a, k \leftarrow Z_q^*$, it is computationally infeasible to distinguish $g^{a^{t+1}k} \cdot g^r$ and g^{ar}. Formally, TDH assumption holds for groups if there exists a negligible function v such that

$$\Pr[a, k \leftarrow Z_q^*, g, g^a, g^{a^2}, ..., g^{a^t}, g^{ak}, g^{a^2k}, ..., g^{a^tk} A(g, g^a, g^{a^2}, ..., g^{a^t}, g^{ak}, g^{a^2k}, ..., g^{a^tk}, r) = g^{a^{t+1}k} \cdot g^r \Lambda g^{ar}, r \leftarrow Z_q^*] < v(k) \quad (12)$$

Assumption 2 (DLA assumptions) On input $u, v, w, u^r, v^s \in G$, it is computationally infeasible to distinguish $z_0 \leftarrow w^{r+s}$ and $z_1 \leftarrow G$. Formally, DLA assumption holds for groups output by Bilinear Setup if there exists a negligible function v such that

$$\Pr[(p, G, e, g, h) \leftarrow BilinearSetup(1^k); r, s \leftarrow G_p; u, v, w \leftarrow G; b \leftarrow \{0, 1\}; z_0 \leftarrow w^{r+s}; z_1 \leftarrow G : A(p, G, G_T, e, g, h, u, v, w, u^r, v^s, z_b) = b] < 1/2 + v(k) \quad (13)$$

The correctness of the proposed protocol can be proved by verifying the equation is valid on the properties of the bilinear group, which is needless to give unnecessary details here for it is not complicated. This section emphasizes on the analysis of the security properties of the protocol: anonymity, unforgeability, traceability and no double-spending.

4.1.1 Anonymity

Theorem 1: Under the DLA assumption in section 2.1, the above protocol is anonymity. More specifically, if there is an adversary A that succeeds with a non-negligible probability to breach anonymity of the protocol, then there is a simulator S running in polynomial time that solves the DLA problem with a non-negligible probability.

Proof:

The process is based on the DLA assumption.

Initialization phase Using the DLA case to initialize, first, we need to send the initial parameters of the e-cash protocol to the simulator S, and give $(u, v, g, u^r, v^s) \in G$,at the same time we take $Adv_A^{anon}(k)$as the advantage to breach the anonymity of the protocol by adversary A and take $Adv_S^{DLA}(k)$as the advantage of simulator S wining the DLA game, and the value of both cannot be ignored. If A has a probabilistic polynomial time algorithm which can recover the users' identity from the values of their commitment to the certificate, simulator S can invoke the algorithm of adversary A to distinguish $w = g^{r+s}$ and $w = g^{r+s+z}$ .Simulator S chooses bilinear group (n, g, G, G_T, e), where $G = \langle g \rangle$, to simulate the initialization of the e-cash protocol, and adversary A gets the public key pk: $(g^\alpha, g^{\alpha^2}) \in G$of the central bank and extract key tk: (x, y, z) from S.

Query phase The adversary A queries users' identity from simulator S for many times, which means that A send the e-cash M to S, and gets relevant certificate from it.

Challenge phase A choose the values M0 and M1 of e-cash and sends them to S, and S chooses b equals 0 or 1 randomly and produce the member's identity information to send to A.Finally, adversary A output the judgment of b finally, and it has two cases:

Case1: In the S's five-tuples,$w = g^{r+s+z}$,and the reference list for the initialization of S is $(u,v,w) = (f^{r_u}, h^{s_v}, g^{r_u+s_v+z})$,so this game is a real anonymous game, and A can guess out $b = b'$ with the advantage which cannot be ignored in this case, which means the probability of correct is $1/2+\varepsilon$.

Case2: In the S's five-tuples,$w = g^{r+s+z}$,and the reference list for the Initialization of S is $(u,v,w) = (f^{r_u}, h^{s_v}, g^{r_u+s_v+z})$,so this game is a real anonymous game. in this case, S choose the value of commitment about identity information which A get from the value of e-cash in challenge phase .so in every bit in $c_i \in \{0,1\}$ has:

$$c = (f^{r_0}\prod_{i=1}^{n} u_i^0, h^{s_0}\prod_{i=1}^{n} v_i^0, g^{r_0+s_0}\prod_{i=1}^{n} w_i^0) = (f^{r_1}\prod_{i=1}^{n} u_i^1, h^{s_1}\prod_{i=1}^{n} v_i^1, g^{r_1+s_1}\prod_{i=1}^{n} w_i^1) \tag{14}$$

in which $r_i, s_i, t_i$is any value, $i = 1,2$.The probability of judging out every bit by adversary A is $(1/2)^n$,and we let it be ε' which can be ignored, so in this case, A can guess out $b = b'$ with probability which cannot be ignored.

Based on the two cases above, if A's answer is right, which means $b' = b$, S output $s = 1$to show its judgment :$w = g^{r+s+z}$; Or S output $s = 0$ to indicate : $w = g^{r+s}$ Given $\Pr[w = g^{r+s+z}] = \Pr[w = g^{r+s}] = 1/2$,we can get:

$$\begin{aligned}
&Adv_A^{[anon]}(k)]_{\Gamma_0} - Adv_A^{[anon]}(k)]_{\Gamma_1} \\
&= \Pr[s = 1 | w = g^{r+s+z}] - \Pr[s = 1 | w = g^{r+s}] \\
&= 2\Pr[s = 1, w = g^{r+s+z}] - 2\Pr[s = 1, w = g^{r+s}] \\
&= 2(1/2+\varepsilon) - 2\varepsilon' \\
&= 2Adv_S^{\mathrm{DLA}}
\end{aligned} \tag{15}$$

Because ε is the advantage which cannot be ignored, S can solve DLA problem in polynomial time. And DLA is a hard problem that cannot be solved in polynomial time, so A cannot break the anonymity of the protocol.

4.1.2 Unforgeability

Theorem 2: The e-cash protocol is unforgeable. More specifically, if there is an adversary A breached the unforgeability of the protocol with the advantage which cannot be ignored, and there exists a simulator S in probabilistic polynomial time can solve TDH problem with the advantage which cannot be ignored.

Proof:

The process is based on the TDH assumption.

Initialization phase We will use the TDH case to initialize, and send the four-tuples $g^{\alpha}, g^{\alpha^{-1}k}, g^{\alpha^2 k}$, $g^{\alpha^2} \in G$ and the other initial parameters of the e-cash protocol to the simulator S, then give $(u,v,g,u^r,v^s) \in G$.As an adversary A breached the unforgeability of the protocol with the advantage ε which cannot be ignored, which means that A has the algorithm to breach the process of signature for the value of e-cash in this protocol, then S can invoke the algorithm of A to calculate $g^{\alpha k} \cdot g^r$ and $g^r \in G$,and $r \leftarrow Z_q^*$, In the process of simulating to initialize protocol by S, adversary A gets the public key pk: $(g^{\alpha}, g^{\alpha^2}) \in G$of the central bank.

Query phase Hash-query: In the process of constructing the signature scheme, we used the Hash function for (m,U_1,U_2), so in the process of proving , an Adversary A can hash to query for at most q_0 times in the Hash phase. The simulator s holds an empty table, whenever A queries, S will check the table first. Whether (m_i,R_{i1},R_{i2},h_i) exists, if true, S will send h_i to A.

(1) If (m_i,R_{i1},R_{i2},h_i) not exists, which means (m_i,R_{i1},R_{i2})has never queried for hash prediction. Then S will save (m_i,R_{i1},R_{i2},h_i) into the table, and choose $h_i \leftarrow Z_q^*$ randomly to send to A.

Signature query: In this phase A is permitted to query for signature for at most q_{d_s}times. To every query on m_i, the simulator S will execute.The following operations to get the result.

(2) Choose two random number $c_i,d_i \leftarrow Z_q^*$ to calculate $U_{i1}=g^{kd_i}$ and $U_{i2}=g^{kc_id_i-k^2d_i}$;

Then choose a random number $h_i \leftarrow Z_q^*$, and save (m_i,U_{i1},U_{i2},h_i) in the table.

(3) Calculate $V_{i1}=g^{\alpha^{-1}kc_id_i}g^{\alpha c_ih_i}$ and $V_{i2}=g^{a^2c_ih_i-a^2kh_i}$, and the simulator S will send $(U_{i1},U_{i2},V_{i1},V_{i2})$ to A as the result. If S set $g^{r_i}=g^{\alpha c_i-\alpha k}$ and $t=\alpha^{-2}kd_i$, the signature above can be expressed as:

$$\begin{aligned} U_{i1} &= g^{\alpha^2 t}=g^{\alpha^2\alpha^{-2}kd_i}=g^{kd_i} \\ U_{i2} &= g^{\alpha r_i t}=g^{\alpha(\alpha c_i-\alpha k)\alpha^{-2}kd_i}=g^{kc_id_i-k^2d_i} \\ V_{i1} &= \left(g^{\alpha k}g^{r_i}\right)^{(t+h)}=g^{\alpha c_i(\alpha^{-2}kd_i+h)}=g^{\alpha^{-1}kc_id_i}g^{\alpha c_ih_i} \\ V_{i2} &= g^{\alpha r_ih_i}=g^{\alpha(\alpha c_i-\alpha k)h_i}=g^{(\alpha c_i-\alpha^2k)h_i}=g^{\alpha^{-1}kc_id_i}g^{\alpha c_ih_i} \end{aligned} \tag{16}$$

Output phase An output a signature list$\sigma_0=(m^*,U_{j1},U_{j2},V_{j1},V_{j2})$, in which$m^*$is the e-cash that has never queried for signature prediction. S can produce two legal signature [15] [16]to make $m^* \neq \mathrm{m_i}$.

$$\begin{aligned} \sigma_0 &= (m^*,U_{j1},U_{j2},V_{j1},V_{j2}) \\ \sigma_1 &= (m^*,U_{j1},U_{j2},V_{j1}^{'},V_{j2}^{'}) \end{aligned} \tag{17}$$

S can calculate d_{j1} and d_{j2} in the following way. As $V_{j1}^{'}/V_{j2}=\left(g^{\alpha k}g^{r_j}\right)^{(t+h_j^{'})}/\left(g^{\alpha k}g^{r_j}\right)^{(t+h_j)}=\left(g^{\alpha k}g^{r_j}\right)^{(h_j^{'}-h_j)}$, $V_{j2}^{'}/V_{j1}=g^{\alpha r_jh'}g^{\alpha r_ih}=g^{\alpha r_i(h'-h)}$. S can breach this TDH case in polynomial time. And as we all know that TDH is a problem that cannot be solved in polynomial time, so after an Adversary A hash-querying for at most q_0 times , the probability of breaching the unforgeability of the protocol is ε which can be ignored.

4.1.3 Traceability

Theorem 3: The e-cash protocol has traceability, if there doesn't exist a simulator S in probabilistic polynomial time can breach the bind of GS proving system and forge an untraceable e-cash with the advantage which cannot be ignored, then the advantage $Adv_A^{trac}(k)$ of any A in polynomial time wining this traceable game can be ignored.

Proof:

According the definition in section 2.3, $Adv_A^{trac}(k)$ has two parts:

(1) As this protocol uses the GS proving system whose commitment has the feather of binding, so the probability of two different $\mathrm{C_{ID}}$ producing the same value can be ignored. And in the condition of DLA assumption, when $(u,v,w)=(f^{r_u},h^{s_v},g^{r_u+s_v+z})$, the intercessor uses the key $\mathrm{t}k=(x,y,z)$ through $(g^z)^{C_{ID}}=C_3C_1^{-1/x}C_2^{-1/y}$ to get the unique information of $\mathrm{C_{ID}}$ from $C=(C_1,C_2,C_3)=(f^r\prod_{i=1}^n u_i{}^{c_i},h^s\prod_{i=1}^n$

$v_i^{c_i}, g^{r+s}\prod_{i=1}^{n} w_i^{c_i})$:

$$\begin{aligned} &C_3{C_1}^{-1/x}{C_2}^{-1/y} = w^{C_{ID}} g^{r+s} \cdot (u^{C_{ID}} f^r)^{-1/x} \cdot (v^{C_{ID}} h^s)^{-1/y} \\ &= g^{(r_u+s_v+z)C_{ID}} \cdot g^{r+s} \cdot (f^{r_u \cdot C_{ID}} \cdot f^r)^{-1/x} \cdot (h^{s_v \cdot C_{ID}} \cdot h^s)^{-1/y} \\ &= g^{(r_u+s_v+z)C_{ID}} \cdot g^{r+s} \cdot (g^{r_u \cdot C_{ID} \cdot x} \cdot g^{rx})^{-1/x} \cdot (g^{s_v \cdot C_{ID} \cdot y} \cdot g^{sy})^{-1/y} \\ &= g^{(r_u+s_v+z)C_{ID}} \cdot g^{r+s} \cdot (g^{r_u \cdot C_{ID}} \cdot g^r)^{-1} \cdot (g^{s_v \cdot C_{ID}} \cdot g^s)^{-1} \\ &= (g^z)^{C_{ID}} \end{aligned} \tag{18}$$

(2) The probability of tracing out the memberships' identities through the e-cash forged in the tracing algorithm can be ignored.

The proving process of this part is similar with the proof of unforgeability, so we just describe it simply.

Initialization phase The simulator S chooses double linear group (n, g, G, G_T, e) ,in which $G=\langle g \rangle$, to simulate the initialization of e-cash protocol, and the adversary A gets the public key pk: $(g^{\alpha}, g^{\alpha^2}) \in G$ of the central bank and the tracing key tk: $(x, y, z)$from S.

Query phase The adversary A queries for users' certificates and signature from S: The adversary A sends the value M of e-cash to S, and gets relevant certificate C_{ID} from it; A sends the e-cash without signature to S and gets signature $(U_{i1}, U_{i2}, V_{i1}, V_{i2})$ from it.

Output phase The adversary A can forge the signature $(U_{i1}{}^*, U_{i2}{}^*, V_{i1}{}^*, V_{i2}{}^*)$ on the e-cash m^* of the user who holds the certification C_{ID*} by its knowledge, and meet $e(V_1{}^*, g^{\alpha}) = e(PK_A, U_1{}^*)e(PK_A^h, g^{\alpha^2})$ $e(U_2{}^*, g)e(V_2{}^*, g)\}$, so the adversary can forge the user's signature on the e-cash can be proved. But the unforgeability of the signature has been proved, so the conclusion has contradiction with the assumption, and we can figure that the probability of tracing out the memberships' identities through the e-cash forged in the tracing algorithm can be ignored.

In the conclusion, the advantage $Adv_A^{trac}(k)$of A in polynomial time wining this traceable game can be ignored.

4.1.4 No double-spending

If the adversary A can repeat to spend the same e-cash and not be distinguished, we can formalize it as: The adversary A succeeds to pay the e-cash M

and M' with the same serial-number $F_r(i)$ to the merchant in the payment phase, in which $T = g^{C_{ID}} \cdot F_r(i)$,$T' = g^{C_{ID}{}'} \cdot F_r(i)$. The merchant doesn't distinguish that the e-cash in the two transactions is spend twice, so $T \neq T'$,which means the values of C_{ID}in this two transactions is different. But as the e-cash protocol has the unforgeability, every e-cash is relevant to a membership's identity information, so the assumption is false, which means this e-cash protocol has the character of no double-spending.

4.2 Performance Analysis

A comparative analysis of the above e-cash protocol and several typical e-cash protocols in recent years [10] [17] [18] [19]is present in this section. The characteristics and security properties of these protocols are compared in Table 1, and the communication spend and computational spend are analyzed in Table 2.

	Features				Security			
	NI	DJ	TF	RV	AN	UF	TR	NDS
[10]	N	N	Y	Y	CCA	N	N	N
[17]	Y	N	Y	N	CPA	N	Y	Y
[18]	N	Y	Y	Y	CPA	N	Y	Y
[19]	N	N	Y	N	CPA	N	Y	Y
Ours	Y	Y	Y	N	CCA	N	Y	Y

(NI: non-interactivity; DJ: dynamically to join; TF: transferability; RV: reversibility; AN: anonymity; UF: unforgeability; TR: traceability; NDS: no double-spending))

Table 1 Comparisons of Security

	Communication Cost		Computation Cost	
	Interactions	Signature Size	Exponentiations	Point Multiplications
[10]	1	$7Z_n$	-	13
[17]	3	$7G + 1Z_n$	3	3
[18]	3	$8Z_n$	11	11
[19]	2	$4G + 2Z_n$	6	2
Ours	1	4G+2Zn	6	3

Table 2 Comparisons of Communication and Computation Cost

We can reach a conclusion that the proposed e-cash protocol satisfies the basic security requirements, such as anonymity, unforgeability, traceability and no double-spending. And it is worth to say that the anonymity in our protocol has reached the level of CCA (chosen ciphertext attack).What is more, our protocol meet non-interactivity, joining dynamically, transferability and so on.

In cryptographic protocols, what is widely used to analysis the computation cost method is to compare the running times of different types of operation in different protocols. So, in table 2, we calculated the exponentiations times and point multiplication times to represent the computation spend of the protocols. And furthermore, whit is worth considering is that, due to the number of Banks is far less than the number of customers and merchants, and the protocols, such as registration protocol, open protocol and trace protocol, is executed far less frequently than the spend protocol, the spend protocol has been the most frequent and critical step. So the communication spends compared in table 2 are focused on the number of interactions in the spend protocol and the number of elements required in e-cash signature. Through the comparison of these protocols, we can see that the computation spend of our protocol is in the average level. The protocol in literature [19] obtained a low computation spend, but it paid a high price of large communication spends. In terms of communication spend, our protocol is more efficient than others, for the reasons that only one interaction is needed between both sides when spending the e-cash, and the totality of elements is 4 in G, and the number of elements selected is 2 in Z_n sent during interaction. With the rapid development of science and technology, increasingly rapid computing equipment is constantly updated to such an extent that the computation spend of each entity is far less important than the communication spends in network environment.

5 Conclusion

In this paper, we proposed a new practical e-cash protocol based on dynamic group signature with NIZK proofs system. Our protocol is proven CCA security without random oracles, since we adopted the approach of the cryptography tools of group signatures under BSZ security model and GS proof system. According to a comparative analysis with other protocols, the proposed one has advantage both on the

efficiency and security. In the future work, we should improve the efficiency of the register protocol, make more detailed analysis on preventing insider attacks and simplify the proof steps in standard model.

6 Acknowledgments

The authors would like to thank the reviewers for their detailed reviews and constructive comments, which have helped improve the quality of this paper. This work was supported in part by the Major National Scientific&Technological Projects under grant No.2013ZX03002006, the Liaoning BaiQian-Wan Talents Program No.2011921071, the Natural Science Foundation of Shenyang City of China under Grant No. F12-277-1-41

References

[1] D. Chaum, "Blind signatures for untraceable payments," in *Proc. of Advances in Cryptology-CRYPTO (Crypto'82), Santa Barbara, California, USA*. Springer-Verlag, August 1982, pp. 199–203.

[2] S. von Solms and D. Naccache, "On blind signatures and perfect crimes," *Computers & Security*, vol. 11, no. 6, pp. 581–583, 1992.

[3] A. Lysyanskaya and Z. Ramzan, "Group blind digital signatures: A scalable solution to electronic cash," in *Proc. of the 2nd International Conference on Financial Cryptography (FC'98), Anguilla, British West Indies, LNCS*, vol. 1465. Springer-Verlag, February 1998, pp. 184–197.

[4] N. T. Ferguson, "Single term off-line coins," in *Proc. of the 1993 Workshop on the Theory and Application of of Cryptographic Techniques (EUROCRYPT'93), Lofthus, Norway, LNCS*, vol. 765. Springer-Verlag, May 1993, pp. 318–328.

[5] T. Nishide, S. Miyazaki, and K. Sakurai, "Security Analysis of Offline E-cash Systems with Malicious Insider," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 3, no. 1/2, pp. 55–71, March 2012.

[6] G. Maitland and C. Boyd, "Fair electronic cash based on a group signature scheme," in *Proc. of the 3rd International Conference on Information and Communications Security (ICICS'01), Xian, China, LNCS*, vol. 2229. Springer-Verlag, November 2001, pp. 461–465.

[7] S. Canard and J. Traoré, "On fair e-cash systems based on group signature schemes," in *Proc. of the 8th Australasian Conference on Information Security and Privacy (ACISP'03), Wollongong, Australia, LNCS*, vol. 2727. Springer-Verlag, July 2003, pp. 237–248.

[8] S. Canard and A. Gouget, "Divisible e-cash systems can be truly anonymous," in *Proc. of the 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT'07), Barcelona, Spain, LNCS*, vol. 4515. Springer-Verlag, May 2007, pp. 482–497.

[9] M. Blanton, "Improved conditional e-payments," in *Proc. of the 6th International Conference on Applied Cryptography and Network Security (ACNS'08), New York City, New York, USA, LNCS*, vol. 5037. Springer-Verlag, June 2008, pp. 188–206.

[10] Z. Eslami and M. Talebi, "A new untraceable off-line electronic cash system," *Electronic Commerce Research and Applications*, vol. 10, no. 1, pp. 59–66, 2011.

[11] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Proc. of the 20th Annual ACM symposium on Theory of Computing (STOC'88), Chicago, Illinois, USA*. ACM, May 1988, pp. 103–112.

[12] M. Bellare, D. Micciancio, and B. Warinschi, "Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions," in *Proc. of the 22nd International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT'03), Warsaw, Poland, LNCS*, vol. 2656. Springer-Verlag, May 2003, pp. 614–629.

[13] M. Bellare, H. Shi, and C. Zhang, "Foundations of group signatures: The case of dynamic groups," in *Proc. of the Cryptographers' Track at the RSA Conference 2005 (CT-RSA'05), San Francisco, California, USA, LNCS*, vol. 3376. Springer-Verlag, February 2005, pp. 136–153.

[14] J. Groth, "Fully anonymous group signatures without random oracles," in *Proc. of the 13th International Conference on Theory and Application of Cryptology and Information Security (ASIACRYPT'07), Kuching, Malaysia, LNCS*, vol. 4833. Springer-Verlag, December 2007, pp. 164–180.

[15] E. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung, "Design validations for discrete logarithm based signature schemes," in *Proc. of the 3rd International Workshop on Practice and Theory in Public Key Cryptosystems (PKC'00), Melbourne, Victoria, Australia, , LNCS*, vol. 1751. Springer-Verlag, January 2000, pp. 276–292.

[16] D. Pointcheval and J. Stern, "Security arguments for digital signatures and blind signatures," *Journal of cryptology*, vol. 13, no. 3, pp. 361–396, 2000.

[17] S. Wang, Z. Chen, and X. Wang, "A new certificateless electronic cash scheme with multiple banks based on group signatures," in *Proc. of the 2008 International Symposium on Electronic Commerce and Security (ISECS'08), August, Guangzhou, China*. IEEE, August 2008, pp. 362–366.

[18] C.-I. Fan and V.-M. Huang, "Provably secure integrated on/off-line electronic cash for flexible and efficient payment," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 5, pp. 567–579, 2010.

[19] F. Li, M. Zhang, and T. Takagi, "Identity-based partially blind signature in the standard model for electronic cash," *Mathematical and Computer Modelling*, vol. 58, no. 1/2, pp. 196–203, July 2012.

Author Biography

Jian Xu received his doctor degree of computer application technology in 2013 at Northeastern University. Now he is a lecture of Software College at Northeastern University. His research interests include cryptography and network security.

Yuxi Li received her Bachelor degree of computer science and technology in 2012 at Sichuan University. Now she is a Master candidate. Her research interests include cryptography and network security.

Jingwei Miao received his doctor degree in computer science at INSA-Lyon, France. His research interests are in the areas of mobile computing, delay tolerant networks, and social networks. He received his M.S. degree in software engineering from Peking University, China in 2009. He received his B.S. degree in software engineering from Dalian University of Technology, China in 2006.

Fucai Zhou received his PhD degree of Computer Software and Theory at Northeast University. He is currently a Professor and Doctoral Supervisor of Software College in Northeastern University. His research interests include cryptography, network security, trusted computing, basic theory and critical technology in electronic commerce.