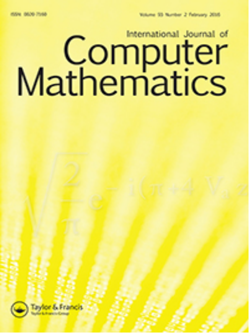

International Journal of Computer Mathematics

ISSN: 0020-7160 (Print) 1029-0265 (Online) Journal homepage: http://china.tandfonline.com/loi/gcom20

The electronic cash system based on non-interactive zero-knowledge proofs

Fucai Zhou, Yuxi Li, Qingshi Zhou, Jingwei Miao & Jian Xu

To cite this article: Fucai Zhou, Yuxi Li, Qingshi Zhou, Jingwei Miao & Jian Xu (2016) The electronic cash system based on non-interactive zero-knowledge proofs, International Journal of Computer Mathematics, 93:2, 239-257, DOI: 10.1080/00207160.2014.933816

To link to this article: http://dx.doi.org/10.1080/00207160.2014.933816

Accepted author version posted online: 12 Jun 2014.
Published online: 15 Jul 2014.

International Journal of Computer Mathematics, 2016
Vol. 93, No. 2, 239–257, http://dx.doi.org/10.1080/00207160.2014.933816

The electronic cash system based on non-interactive zero-knowledge proofs

Fucai Zhou[a*], Yuxi Li[b], Qingshi Zhou[c], Jingwei Miao[d] and Jian Xu[a,e]

[a] *Software College, Northeastern University, Shenyang, Liaoning, China;* [b] *College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning, China;* [c] *College of Science, Purdue University, West Lafayette, IN, USA;* [d] *University of Lyon, CNRS, INSA-Lyon, LIRIS, UMR5205, Lyon F-69621, France;* [e] *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*

(*Received 9 January 2014; revised version received 29 April 2014; accepted 9 June 2014*)

Electronic cash is an electronic form of currency, which allows the cash transactions over communication networks under privacy protections. However, it still has some aspects that have not been well studied. Known constructions suffer from at least one of the following limitations: (1) relying on a random oracle, (2) not supporting multiple bank setting and users dynamically joining or (3) prohibitively expensive. In this paper, we propose a new electronic cash system that avoids all these limitations. In other words, our system is anonymous against chosen-ciphertext attack (CCA) in the standard model, and supports multiple banks enrolling and users dynamically joining, which is achieved by the utilization of non-interactive zero-knowledge proof and dynamic group signature. Finally, in the standard model, a formal security proof is given to claim that our system has CCA anonymity, unforgeability, traceability and no double-spending. Compared with the existing systems, ours has advantages of both the efficiency and security.

Keywords: electronic cash; dynamic group signature; non-interactive zero-knowledge proofs; Groth–Sahai system; chosen-ciphertext attack

2010 AMS Subject Classifications: 94A60; 14G50; 11D09; 06F20; 00A69

1. Introduction

Nowadays, thanks to the progression of the Internet and cloud computing, information technology is impacting many aspects of human life. Among the most significant usage of this technology is the electronic commerce: performing financial transactions by message exchanged in telecommunication lines. A key requirement for electronic commerce is the development of secure and efficient electronic cash systems.

Electronic cash, as its name implies, is the physical cash in an electronic form which is usually stored in the so-called electronic wallet, such as software and smart card, and it allows for the cash transactions over communication networks under privacy protections. General electronic cash systems include three entities and four protocols: the entities are the *bank*, *merchant* and *user*, and the protocols involved in are *open*, *withdraw*, *spend* and *deposit*. An electronic cash system has a number of features: (1) unforgeability: bank's signatures on the user's e-cash should not be produced by anyone else. (2) Anonymity: the spender of the e-cash must remain

[*] Corresponding author. Email: fczhou@mail.neu.edu.cn

anonymous. If the e-cash is spent legitimately, neither the recipient nor the bank can identify the spender. (3) Unlinkability: given two deposited cash, the bank cannot tell if they come from the same withdrawal or are from the same user, and the bank cannot link a deposited cash with any withdrawal. Moreover, some advantages are unique to electronic cash scheme: (1) divisibility: an electronic cash can be broken up and spent in multiple payments. (2) Anonymity revocation: in some cases, to prevent illegal activities, the anonymity of user may be revoked. (3) Portability: there is no physical limitation for user to spend his e-cash because the e-cash can be transferred via the Internet.

The first electronic cash system was introduced by Chaum in 1982 [6], aiming at solving the limitations of physical cash in transaction and effectively protecting users' anonymity. However, while electronic cash has many advantages, it is still not widely used. Even after 30 years of research, and with 40+ candidate constructions in the literature, the state is not completely satisfactory. Essentially, known constructions in the literature suffer from at least one of the following limitations: (1) the systems' constructions mainly rely on a random oracle, (2) the systems do not support multiple bank setting and users dynamically joining or (3) the systems are prohibitively expensive, which is a major obstacle while making the electronic cash universally applicable.

To the best of our knowledge, the security problem of electronic cash systems has not been previously defined in a general form. And most systems are constructed in random oracle model, which cannot be instantiated in standard model or cannot be proved to be secure. For example, Goldwasser and Kalai [12] use the technology of Fiat–Shamir heuristic to achieve proved security under random oracle model, but they mentioned that it is still not secure in the standard model. Therefore, it is a huge challenge for researchers to construct an off-line dynamic electronic cash system that is anonymous in the standard model.

The prior electronic cash systems require at least three rounds of interaction in every protocol. It means that both the bank and user should be online simultaneously, which makes it prohibitively expensive. For example, the communication complexity of every protocol is a large polynomial with the security parameter and may have performance problems when the sets are large.

To address the limitations of prior art on electronic cash systems, we focus on the problem discussed above, and give a non-interactive solution that avoids all of the above limitations. In order to make it dynamic and available for multiple banks, we make use of the dynamic group signature, requiring that each local bank and user act as a group member joined dynamically in a register phase. Moreover, considering the problems of communication complexity, non-interactive zero-knowledge (NIZK) proofs can improve this situation to some extent. However, until the recent Groth–Sahai (GS) proof system, there were no efficient NIZK proof systems in cryptographic constructions. But due to the complexity of GS proof system, many researchers tend to treat it as a black box, but ignore its proof details and verification methods, and others tend to derive the non-interactive proofs from interactive proofs via the Fiat–Shamir heuristic which is not known to be provably secure. However, constructing an efficient provably secure electronic cash scheme is simply not a matter that replaces the Fiat–Shamir-based proofs with the GS system. If the GS proof method is applied to electronic cash systems, the security can be ensured by the properties of the proof method, and moreover, the system does not need to interact between the bank and user (or user and user) that can reduce the time of communication and storage cost.

Aiming at the above problems, and inspired by the work of Bellare *et al.* [1] (Bellare, Shi and Zhang, BSZ), GS [15] and Nishide *et al.* [19], we propose a practical electronic cash system based on the GS proof system and dynamic group signature scheme in BSZ model. Compared with the existing related works, we have four contributions as follows:

(1) Our first result is the efficient pairing-based instantiation of a new electronic cash security model in the standard model, and formally define the security properties that the electronic cash system should satisfy.

(2) Based on the BSZ dynamic group signature, the system preserved anonymity of group signature even if the adversary can see arbitrary key exposures or arbitrary openings of other group signatures. Moreover, it achieves multiple banks enrolling and users dynamically joining in the register phase.
(3) We apply the GS proof system for pairing product equations and construct a specific proof procedure of NIZK for the proposed system. That means it need not interact between the user and bank (or user and user), which can reduce the communication time and storage cost compared with the existing electronic cash systems. Moreover, we prove and sign the blocks of messages instead of limiting the proved message to only one bit (0 or 1), which improves the proof efficiency.
(4) In the standard model, we give the strict security proof for the proposed system that is with chosen-ciphertext attack (CCA) anonymity, unforgeability, traceability and no double-spending. Then, we compare this system with other existing systems on security and performance, it turns out that the proposed system is better than the existing ones.

The remainder of this paper is arranged as follows: in Section 2, we list a few related works in the literate that relates to our topic; we recall the basic notions and primitives in Section 3; the formal definitions are given in Section 4; the detailed construction is given in Section 5; the security and performance analysis are given in Section 6 and Section 7 concludes this paper.

2. Related works

In previous research, general electronic cash digital signature is the most widely used cryptographic tool for implementing secure electronic cash systems, due to its integrity and authentication. The integrity property ensures that the sent and received electronic cash are not modified, and the authentication property ensures that the spender is not in disguise. Because of its significance, many variations of the digital signature scheme were used to construct electronic cash systems, such as blind signature, group signature, etc.

The first electronic cash system, proposed by Chaum [7], is based on the blind signature which makes e-cash anonymous and unlinkable. Blind signature allows the content of a message to be disguised (blinded) before signing, and the blind-signed message can be publicly verified. Chaum's proposal allowed users to obtain electronic cash from a bank and spend it in a manner that it is untraceable by the bank or any other party. In 1990, he extended this idea (with Amos Fiat and Moni Naor) to prevent double-spending [8]. In 1992, Brands put forward an off-line and completely anonymous electronic cash system based on restrictive blind signature, and improved it in 1993. However, due to the increasingly serious information attack in network times, electronic cash systems with complete anonymity property of blind signature can, in addition to protect users' privacy, bring an opportunity to criminal offences such as corruption, money laundering, racketeering and so on [21]. In 1997, Davida *et al.* [9] presented a controllable anonymous fair electronic cash system, through a trusted party, users' anonymity can be removed in a particular situation (such as legal requirements).

Moreover, there is only one bank entity in charge of the distribution of all money in the most existing electronic cash systems, which makes the bank overload and may lead to signal peer invalid problem. And it is too difficult to match the multiple banks' situation in real life. For the above reasons, in 1998, Lysyanskaya and Ramzan [17] proposed the first multiple banks electronic cash system in the financial cryptography. They utilized blind signature and group signature in their system, and opened up a new direction for further research.

In group signatures, any member of a group can sign messages anonymously on behalf of the whole group, and a user can verify whether a signature is generated by the group member or not, with the group's public key that is usually constant and unique. However, he/she cannot learn the personal identity of the member who signs the message. Hence, compared with blind signatures, group signatures can better satisfy the security requirement of electronic cash in reality. With the continuous progress on group signatures, many cryptographic researchers use it to design electronic cash system. In 2001, a fair electronic cash system was proposed by Maitland and Boyd [18], which is based on a coalition-resistance group signatures. After two years, another efficient fair electronic cash system is proposed by Canard which is a breakthrough in the research history of e-cash [5]. Except other essential security properties in the standard model, it is proved traceable for double-spending.

It is always a hard problem for electronic cash system to prevent double-spending, and many researchers have paid a lot of attention to it. Chaum's [8] system used cut-and-choose mechanism, which encodes the owners identity into the e-cash such that he remains anonymous after one payment but will be identified if he double-spends. However, this mechanism is highly inefficient in terms of the data exchanged between the spenders and the merchants during each payment as each coin contains reasonably large size. Subsequent to the original proposal, several improvements and new constructions [5,17,18] have been proposed.

In recent years, there is a tendency that electronic cash systems are possessing specific features for specific requirements. In 2008, with the cryptography tools of zero-knowledge proofs and verifiable encryption, Blanton [2] proposed an efficient transferable electronic cash system based on Camenisch–Lysyanskaya (CL) group signatures. However, the anonymity in its deposit protocol is not strong enough. Recently, in 2011, a new off-line e-cash protocol, proposed by Eslami and Talebi [10], achieved not only anonymity and traceability, but perfectly fraud control. And what is worth mentioning is that it makes bank processing data more effective by attaching expiration date to every cash, so the bank can abolish the outdated e-cash directly.

3. Preliminaries

In this section, we recall and define basic notions and primitives used.

3.1 *Bilinear pairing*

Let G be a cyclic multiplicative group of prime order p, generated by element $g \in G$. Let G_T be a cyclic multiplicative group of the same order, and there exists a pairing $e : G \times G \to G_T$ which has the following properties:

(1) Bilinearity: $e(g^a, g^b) = e(g, g)^{ab}$ for all $a, b \in Z_n$.
(2) Non-degeneracy: $e(g, g) \neq 1_{G_T}$ and 1_{G_T} is a generator of G_T.
(3) Computability: for all $P, Q \in G$, $e(P, Q)$ is efficiently computable.

Typically, the map e will be derived from either the Weil or Tate pairing on an elliptic curve over a finite field.

We call (p, G, G_T, e, g) a tuple of bilinear pairing parameters, produced as the output of a probabilistic polynomial time algorithm that runs on input 1^k.

3.2 *GS proof system*

GS proof system is a non-interactive kind of zero-knowledge proof system. Zero-knowledge proof is introduced in 1985 by Goldwasser *et al.* [13]. Based on this, provers can reveal nothing other than the validity of assertion being proven. Zero-knowledge proof system has been used in many cryptographic protocols, such as anonymous credentials, anonymous signatures, online voting and so on. A general kind of it is interactive, where verifier needs to make random challenges with prover several times, which is not practical. The definition of NIZK was first introduced by Blum *et al.* [3]. It can make verifier to be sure of whether a statement is true or not without disclosing any information and multiply interactive between both sides. In addition, in 2012, an efficient pairing-based non-interactive proof system was constructed by GS [15] in EUROCRYPT. The common basic concepts are as follows:

Commitment values $C_{m,m=1\ldots M}$ hide $x_{m,m=1\ldots M} \in G_1$ by selecting randomly $a_{q,q=1\ldots Q} \in G_1$, $b_{q,q=1\ldots Q} \in G_2$, $\alpha_{q,m,q=1\ldots Q,m=1\ldots M} \in Z_p$, $\beta_{q,m,q=1\ldots Q,m=1\ldots M} \in Z_p$, to compute $C_m = a_q \prod_{m=1}^{M} {x_m}^{\alpha_{q,m}}{}_{m=1\ldots M}$. The statement s consists of all the commitments and bilinear pairing product equations

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

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

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

3.3 *Dynamic group signature scheme in BSZ model*

In this section, we mainly describe the first dynamic group signature scheme proposed by Bellare *et al.* [1]. They formally defined a strong security model (called BSZ model) for group signature, which is available for members to join dynamically.

The dynamic group signature scheme based on BSZ model consists of the following algorithms: Setup, Join, GSig, GVf, Open and Judge.

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 obtain a certificate from the issuer, in other words a signature under the issuer's key. Any group member can generate a group signature simply by signing the message using 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.

(1) Setup(1^k): It is the initialization algorithm, on input the security parameter, 1^k, produces a pair of signing and verifying keys, a pair of encryption and decryption keys.

(2) Join(U_i): It is an interactive protocol between a user U_i (using his secret key $usk[i]$) and the group manager (using his private key msk). At the end of the protocol, the user obtains a signing key sk (or group membership certificate), and the group manager adds the user to the registration list, storing some information in $Reg[i]$.

(3) GSig(pk, sk, m): It is the group signing algorithm. To sign a message m, the user uses his secret key sk, and output a signature σ under the group public key pk.

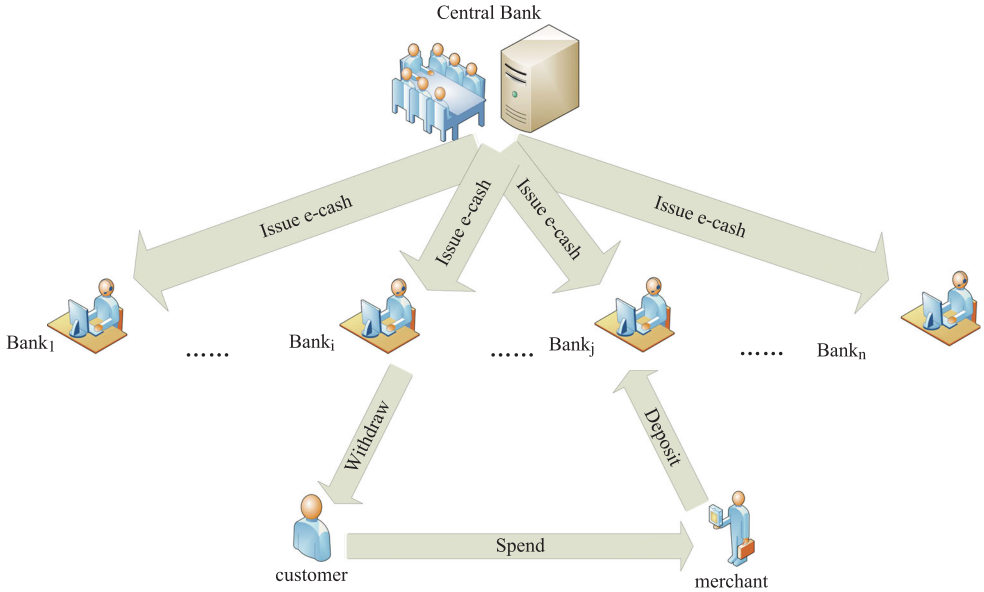

Figure 1. The e-cash circulation.

(4) GVf(gpk, σ, m): It is the group signature verification algorithm. Anybody should be able to verify the validity of the signature σ on the message m with the public key pk. This algorithm thus outputs 1 if the signature is valid, and 0 otherwise.
(5) Open(gpk, σ, ok, m): It is the open algorithm. The opener can provide the identity signer, after granting the opening key sk, for a valid signature with the public key pk. It thus outputs the user i, together with a proof π.
(6) Judge(gpk, i, m, σ): It is the judge algorithm. This algorithm is used to verify the correctness of the open algorithm. If valid, output 1; otherwise output 0.

4. Definitions

To better describe our electronic system, in this section, we give the following definitions that are used in our system.

4.1 *Entity constitution*

There are three types of entities included in this system, as shown in Figure 1: the central bank, the local banks and the 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 managing and to 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 entities concerned, either of them can apply to the central bank for extracting the identity of the other and executing the arbitration.

4.2 *Formal definition*

Our proposed electronic cash system includes a tuple of protocols and algorithms, which are defined as follows:

(1) Setup(1^k) $\rightarrow$ (pk, tk, PK, d): This is a probabilistic algorithm that takes a security parameter k as input, and the central bank's public key pk and extract key tk, the local bank group public key pk, and the local bank signature private key d (used for issuance of electronic cash for his signature).
(2) Register($M_i : PK.sk_i, pk_i$; central bank: pk)$\rightarrow$($M_i : C_{ID}$, central bank : $reg[ID]$): This is a probabilistic protocol that the user takes his sk and pk as input, and the central bank outputs the user's certificate C_{ID}, at the same time establish an entry for the member in the registration form, the content of which is the user's certificate.
(3) Open$(C_{ID}, pk, d) \rightarrow \Sigma$: This is a protocol that user uses his own certificate C_{ID} to generate statements Σ, outputs Σ to the bank.
(4) Withdraw$(C_{ID}, pk, d) \rightarrow \Gamma$: This is a protocol that bank uses its own certificate to generate statements, and use the central bank's public key pk and the user's private key to generate the signature σ of the e-cash m, eventually integrating it and outputs σ to the user.
(5) Spend$(M) \rightarrow 1/0$: This is a protocol that the customer takes the e-cash signature σ and the value T together as M to act as input. After receipt of the e-cash, the merchant verifies its correctness, if correct, then outputs 1, otherwise outputs 0.
(6) Deposit$(M) \rightarrow 1/0$: This is a protocol that the merchant deposits his e-cash into the bank, the bank verifies its correctness, if correct, then outputs 1, otherwise outputs 0.
(7) Trace$(C_{ID}, tk) \rightarrow ID$: This is a protocol that is used in the dispute when the local bank inputs customer's ID to the central bank, central bank uses the extract key tk to open the commitment value, and gets the user identity information.

4.3 *Security definition*

In this section, we mainly discuss the security properties of our proposed system. We consider that this scheme satisfies correctness and secure properties such 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 electronic cash systems is simulated by some encryption services. The adversary gain access to these services by simulator S. The properties are elaborated as follows.

4.3.1 *Correctness*

Correctness refers to the group signature produced by the legal group member (local bank or user), such as: 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 always be accepted by the local bank.

In the following experiments that formalize the security notions, the adversary can run the register protocol through the following two oracles:

(1) O_{Register}: It creates an honest user (HU) for those who do not know the secret keys: the index i is added to the HU list. The adversary gets back the public part of the certificate $pk[i]$.

(2) $O_{\text{Register}'}$: It interacts with the group manager to create a user it will control: the index i is added to the corrupted users (CU) list. The adversary gets back the whole certificate $pk[i]$ and $sk[i]$.

For users whose secret keys are known to the adversary, we let the adversary play on their behalf. For HU, the adversary can interact with them, granted some oracles:

(1) $O_{\text{corrupt}(i)}$: If $i \in$ HU, then it provides the secret key $sk[i]$ of this user. The adversary can now control it. The index i is then moved from HU to CU.
(2) $O_{\text{sign}(i;m)}$: If $i \in$ HU, then it plays as the HU i would do in the signature process. Then i is appended to the list $S[m]$.

4.3.2 *Anonymity*

Anonymity indicates that it is hard for the local banks and users 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 sk, performs two phases with simulation S: probing phase and challenge phase. In the probing phase, on input of the bank's secret key and public parameters *params*, A performs a bounded number of queries in the polynomial time to the simulation S in an adaptive manner to such an extent to obtain the identity of the user or to extract the secret key. Then in the challenge phase, A outputs two legal identities C_{ID_0} and C_{ID_1} and e-cash m to S, S runs the system to output a signature σ on m. In the consequence, A will find it hard to tell which signature is signed by which identity, so his advantage $\text{Adv}_A^{an}(k)$ is a negligible value ε if this system is anonymous. The formal definition is as follows:

```
Experiment $Exp_A^{anon}(k)$
$(pk, tk, PK, d) \leftarrow \text{Setup}(1^k)$
$(m, i_0, i_1) \leftarrow A_{Withdraw}^{O_corrupt, O_Register}(pk, tk, PK, d)$
$\sigma \leftarrow Withdraw(pk, i_b, m, sk[i])$
$b' \leftarrow A_{Withdraw}^{O_corrupt, O_Register}(\sigma)$
If $i_0 \notin HU\ OR\ i_1 \notin HU$
    Return 0
Else return 1
$Adv_A^{nfanon}(k) = \Pr[Exp_A^{anon0}(k) = 1] - \Pr[Exp_A^{anon1}(k) = 1] = \varepsilon$
```

4.3.3 *Unforgeability*

Unforgeability refers that it is hard for any of the local banks or users 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 performs a bounded number of hash queries and signature queries in the polynomial time to the simulation S in an adaptive manner. When performing hash-query per time, A chooses (m_i, R_{1i}, R_{2i}) randomly, then obtains $H(m_i, R_{1i}, R_{2i})$ from S; when performing signature query per time, A queries S for m_i, then S outputs $(m_i, U_{1i}, U_{2i}, V_{1i}, V_{2i})$ by simulating the signature process; in the output phase, if the signature outputted by A is accepted, then he wins the game. We denote his advantage by $\mathrm{Adv}_A^{nf}(k)$. If the system is with unforgeability, the adversary A will find it hard to output a signature which can be accepted by the verification, so his advantage $\mathrm{Adv}_A^{nf}(k)$ is a negligible value ε. We thus say that:

```
Experiment $Exp_A^{nf}(k)$
$(pk, tk, PK, d) \leftarrow \mathrm{Setup}(1^k)$
$(m, \sigma) \leftarrow A_{Withdraw}^{O_corrupt,O_Register}(pk, tk, PK, d)$
If Verify$(PK, m, \sigma) = 0$, return 0.
If $\exists i \in HU \setminus S[m]$
    Open$(pk, m, \sigma, tk) = (i, \pi)$
    Return 1
Else return 0
$Adv_A^{nf}(k) = \Pr[Exp_A^{nf}(k) = 1] = \varepsilon$
```

4.3.4 *Traceability*

Traceability is divided into two sides: on the 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. So the advantage $\mathrm{Adv}_A^{\mathrm{trac}}(k)$ is that

```
Experiment $Exp_A^{tr}(k)$
$(pk, tk, PK, d) \leftarrow \mathrm{Setup}(1^k)$
$(m, \sigma) \leftarrow A_{Withdraw,Open}^{O_Register,O_Register'}(pk, tk, PK, d)$
If Verify$(PK, m, \sigma) = 0$, return 0.
If $\exists j \notin CU \setminus S[m]$
    Open$(pk, m, \sigma, tk) = (i, \pi)$
    Return 1
Else return 0
$Adv_A^{tr}(k) = \Pr[Exp_A^{tr}(k) = 1] = \varepsilon$
```

4.3.5 *No double-spending*

No double-spending indicates that there is no provision for the 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.

5. Main construction

In this section, we describe the main construction of our system, which contains a tuple of protocols and algorithms defined in Section 4.2.

5.1 *Setup*

(1) Calls BilinearSetup (1^K) to generate system parameters (p, G, e, g, h), where $e : G \times G \rightarrow G_T$, $G = \langle g \rangle$ 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 $\vec{u} := (u_1, u_2, u_3, \ldots, u_n)$, $\vec{v} := (v_1, v_2, v_3, \ldots, v_n)$ and $\vec{w} := (w_1, w_2, w_3, \ldots w_n)$, and define a hash function as $H : \{0,1\}^* \rightarrow \{0,1\}^n$.

(2) The central bank chooses $\alpha \leftarrow Z_n$ as its secret key, then uses 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 $d = \{d_1, d_2\} = \{g^{\alpha r'}, g^{\alpha k} g^{r'}\}$.

5.2 *Register*

There are two phases included in the register protocol: 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 system, must perform this interactive protocol with the central bank. For instance, if it is bank registration, the phase is shown as follows:

(1) $Bank_i : (k, g^k) \leftrightarrow Central\,Bank : (g^k)$
By running the Groth [14] protocol, bank i obtains its public key $pk := g^k$ and secret key $pk := g^k$, while at the same time the central bank only obtains bank i's public key $pk := g^k$.

(2) $Central\,Bank \rightarrow Bank_i : (cert_{B_i})$
The central bank chooses $id \in_n^*$ randomly, then generates bank i's certificate $cert_{B_i} = (\sigma_1, \sigma_2)$, where $\sigma_1 = (h \cdot pk)^{1/(r+id)} \sigma_2 = g^{id}$; calculates $c_{ID} = H(\sigma_1, \sigma_2)$ and records it in the corresponding $reg[ID]$, finally sends $cert_{B_i} = (\sigma_1, \sigma_2)$ to bank i. Then the central bank records it in the corresponding $reg[ID]$.

(3) $Bank_i \rightarrow Central\,Bank : (cert_{B_i})$
After obtaining the certificate, bank i judges 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 will 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 no need to state more.

5.3 *Open*

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

(1) $User \rightarrow Bank_i : (\sum)$
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 it is n, and c_i is the ith 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 NIZK proof (π_1, π_2, π_3) on C, as follows:

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

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

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

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

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

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

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

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

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

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

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

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

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.

5.4 *Withdraw*

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

(1) $Bank_i \rightarrow User : (\Gamma)$
First, 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; second, 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.

(2) $User : accept/reject$
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 the spend protocol.

5.5 *Spend*

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 ith e-cash the customer has spent, the following stages will be experienced.

(1) $User_C \rightarrow User_M : (M)$
We assume that the identification of this transaction is R, and the e-cash of this transaction is the ith 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 first, and then add these two values to the e-cash which is to be sent, and at last send 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 the customer has been added, but it will not be leaked out in the process of transaction. If a customer uses the same e-cash as r and i are the same in two transactions, his identity will be confirmed by using the identification R and R' T and $T'$of these two transactions.

(2) $User_M : accept/reject$
After the merchant obtains an e-cash, he will judge whether to accept this e-cash by the following three 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 spent twice.

If the above three steps 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.

5.6 *Deposit*

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

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

(2) $User_{Merchant} \rightarrow Bank_j : (M)$
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 that bank j has obtained is real and valid, and will be deposited into the account of this merchant.

5.7 *Trace*

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:

$Bank_i \rightarrow Central\,Bank : (C)$

The central bank approaches 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 and give this user relevant arbitrate.

6. Security and performance

According to the security definition in Section 4.3, we give the strict security proof based on the new security model. Then we compare our system with the existing similar ones and give a detailed performance analysis.

6.1 *Security analysis*

The security of the electronic cash system is related to the hardness of the following assumptions.

Assumption 1 (Tate Diffie–Hellman assumptions (TDH)) *On input* $\{g, g^a, g^{a^2}, \ldots, g^{a^t}, g^{ak}, g^{a^2k}, \ldots, g^{a^tk}\}$ *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*

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

Assumption 2 (Discrete Logarithm assumptions (DLA)) *On input* $(u, v, g, u^r, v^s) \in G$, *it is computationally infeasible to distinguish* $w = g^{r+s}$ *and* $w = g^{r+s+z}$. *Formally, DLA holds for* *groups output by bilinear setup if there exists a negligible function v such that*

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

The correctness of the proposed system can be proved by verifying that 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.

6.1.1 *Anonymity*

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

Proof The process is based on the DLA.
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 $\text{Adv}_A^{\text{anon}}(k)$ as the advantage to breach the anonymity of the protocol by adversary A and take $\text{Adv}_S^{\text{DLA}}(k)$ as the advantage of simulator S winning 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 electronic cash system, and adversary A gets the public key $pk : (g^\alpha, g^{\alpha^2}) \in G$ of the central bank and extracts 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 sends the e-cash M to S, and gets relevant certificate from it.
Challenge phase. A chooses the values $M0$ and $M1$ of e-cash and sends them to S, and S chooses b equal to 0 or 1 randomly and produces the member's identity information to send to A.

Finally, adversary A outputs the judgement of b finally, and it has two cases:
Case 1: 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 being correct is $1/2 + \varepsilon$.
Case 2: 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 chooses the value of commitment about identity information which A get from the value of e-cash in the challenge phase. So every bit in $c_i \in \{0, 1\}$ has

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

in which r_i, s_i, t_i are any values, $i = 1, 2$. The probability of judging out every bit by adversary A is $(1/2)^n$, and we let it to 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 above two cases, if A's answer is right, which means $b' = b$, S outputs $s = 1$ to show its judgement: $w = g^{r+s+z}$; or S outputs $s = 0$ to indicate $w = g^{r+s}$. Given $\Pr[w = g^{r+s+z}] = \Pr[w = g^{r+s}] = 1/2$, we can obtain

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

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

6.1.2 *Unforgeability*

Theorem 2 *The proposed system is unforgeable, if there is an adversary A breached the unforgeability of the system with the advantage which cannot be ignored, and there exists a simulator S in the 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 electronic cash system 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.

(1) Whether $(m_i, R_{i1}, R_{i2}, h_i)$ exists, if true, S will send h_i to A.
(2) If $(m_i, R_{i1}, R_{i2}, h_i)$ does 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.

(1) Choose two random numbers $c_i, d_i \leftarrow Z_q^*$ to calculate $U_{i1} = g^{kd_i}$ and $U_{i2} = g^{kc_i d_i - k^2 d_i}$.
(2) 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 $V_{i1} = g^{\alpha^{-1}kc_i d_i} g^{\alpha c_i h_i}$ and $V_{i2} = g^{a^2 c_i h_i - a^2 k h_i}$, 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 $t = \alpha^{-2} k d_i$, the signature above

can be expressed as

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

Output phase: An output of 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 signatures [4,20] to make $m^* \neq m_i$

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

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

6.1.3 *Traceability*

THEOREM 3 *The proposed system is traceable, if there does not exist a simulator S in the probabilistic polynomial time can breach the bind of GS proof system and forge an untraceable e-cash with the advantage which cannot be ignored, then the advantage* $\mathrm{Adv}_A^{\mathrm{trac}}(k)$ *of any A in the polynomial time winning this traceable game can be ignored.*

Proof According to the definition in Section 2.3, $\mathrm{Adv}_A^{\mathrm{trac}}(k)$ has two parts:

(1) As we use the GS proof system whose commitment has the feather of binding, so the probability of two different C_{ID} producing the same values can be ignored. And in the condition of DLA, when $(u, v, w) = (f^{r_u}, h^{s_v}, g^{r_u+s_v+z})$, the intercessor uses the key $tk = (x, y, z)$ through $(g^z)^{C_{ID}} = C_3C_1^{-1/x}C_2^{-1/y}$ to get the unique information of 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 to the proof of unforgeability, so we just describe it simply.

Table 1. Comparisons of security.

	Features				Security			
References	NI	DJ	TF	RV	AN	UF	TR	NDS
[10]	N	N	Y	Y	CCA	N	N	N
[22]	Y	N	Y	N	CPA	N	Y	Y
[11]	N	Y	Y	Y	CPA	N	Y	Y
[16]	N	N	Y	N	CPA	N	Y	Y
Ours	Y	Y	Y	N	CCA	N	Y	Y

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

Initialization phase. The simulator S chooses a double linear group (n, g, G, G_T, e), in which $G = \langle g \rangle$, to simulate the initialization of this system, 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 the 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 fact that 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 is contradictory to the assumption, and we can figure out 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 $\mathrm{Adv}_A^{\mathrm{trac}}(k)$ of A in the polynomial time winning this traceable game can be ignored. ■

6.1.4 *No double-spending*

Proof If the adversary A can be repeated to spend the same e-cash and not 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 $T = g^{C_{ID}} \cdot F_r(i)$ and $T' = g^{C_{ID}'} \cdot F_r(i)$. The merchant does not distinguish that the e-cash in the two transactions is spent twice, so $T \neq T'$, which means the values of C_{ID} in these two transactions are different. But as the electronic cash system 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. ■

6.2 *Performance analysis*

A comparative analysis of the proposed system and several typical electronic cash systems in recent years [10,11,16,22] is presented in this section. The characteristics and security properties of these systems are compared in Table 1, and the communication spend and computational spend are analysed in Table 2.

From Table 1, we can reach a conclusion that the proposed system satisfies the basic security requirements, such as anonymity, unforgeability, traceability and no double-spending. And it is worth saying that the anonymity in our system has reached the level of CCA. What is more, our system meets non-interactivity, joining dynamically, transferability and so on.

Table 2. Comparisons of communication and computation costs.

	Communication cost		Computation cost	
References	Interactions	Signature size	Exponentiations	Point multiplications
[10]	1	$7Zn$	–	13
[22]	3	$7G + 1Zn$	3	3
[11]	3	$8Zn$	11	11
[16]	2	$4G + 2Zn$	6	2
Ours	1	$4G + 2Zn$	6	3

In cryptographic systems, what is widely used in the analysis of the computation cost method is to compare the running times of different types of operation in different systems. So, in Table 2, we calculated the exponentiation times and point multiplication times to represent the computation spend of the system. And furthermore, what is worth considering is that, due to the number of banks is far less than the number of customers and merchants, and the protocols of registration, open and trace, 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 system is in the average level. The protocol in the literature [16] obtained a low computation spend, but it paid a high price for 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 four in G, and the number of random elements selected is two in Zn sent during interaction. With the rapid development of science and technology, increasingly rapid computing equipment is constantly updated to an extent such that the computation spend of each entity is far less important than the communication spends in the network environment.

7. Conclusion

In this paper, aiming at the bottle-neck problem of the security and performance of the previous electronic cash system, we combined group signature and NIZK proofs efficiently, and proposed a new practical electronic cash system which can not only maintain anonymity but also detect double-spending by using the GS proof system and dynamic group signature in BSZ model. Our proposed system is with CCA anonymity, unforgeability, traceability without random oracle. According to a comparative analysis with other systems, the proposed one has advantage on both the efficiency and security. In the future work, on the one hand, we will continue to improve the efficiency of the system, especially the register phase, and simplify the proof steps in standard model; on the other hand, another possible improvement is to refining the system to achieve more practical properties, such as transferable and divisible electronic cash.

Acknowledgements

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 National Science and Technology Major Project under Grant No. 2013ZX03002006, the National Natural Science Foundation under Grant No. 61300196, Liaoning Province Science and Technology Projects under Grant No. 2013217004 and the Fundamental Research Funds for the Central Universities under Grant No. N130317002.

References

[1] M. Bellare, H. Shi, and C. Zhang, *Foundations of group signatures: The case of dynamic groups*, in *Topics in Cryptology–CT-RSA 2005, San Francisco*, A. Menezes, ed., Springer, Berlin, Heidelberg, 2005, pp. 136–153.

[2] M. Blanton, *Improved conditional e-payments*, in *Applied Cryptography and Network Security, New York*, S.M. Bellovin, R. Gennaro, A. Keromytis, and M. Yung, eds., Springer, Berlin, Heidelberg, 2008, pp. 188–206.

[3] M. Blum, P. Feldman, and S. Micali, *Non-interactive zero-knowledge and its applications*, in *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, Chicago, USA*, J. Simon, ed., ACM, New York, NY, 1988, pp. 103–112.

[4] E. Brickell, D. Pointcheval, S. Vaudenay, and M. Yung, *Design validations for discrete logarithm based signature schemes*, in *Public Key Cryptography, Melbourne, Australia*, H. Imai and Y. Zheng, eds., Springer, Berlin, Heidelberg, 2000, pp. 276–292.

[5] S. Canard and J. Traoré, *On fair e-cash systems based on group signature schemes*, in *Information Security and Privacy, Wollongong, Australia*, R. Safavi-Naini and J. Seberry, eds., Springer, Berlin, Heidelberg, 2003, pp. 237–248.

[6] D. Chaum, *Blind signatures for untraceable payments in advances in cryptology*, in *Proceedings of Crypto 82, Santa Barbara, USA*, D. Chaum, R.L. Rivest, and A.T. Sherman, eds., Springer, Berlin, Heidelberg, 1983, pp. 199–203.

[7] D. Chaum, *Blind signatures for untraceable electronic cash*, in *Advances in Cryptology-CRYPTO, Santa Barbara, USA*, Vol. 82, D. Chaum, ed., Plenum Press, New York, NY, 1983, pp. 199–203.

[8] D. Chaum, A. Fiat, and M. Naor, *Untraceable electronic cash*, in *Advances in Cryptology-CRYPTO, Santa Barbara, USA*, G. Brassard, ed., Springer, Berlin, Heidelberg, 1990, pp. 319–327.

[9] G. Davida, Y. Frankel, Y. Tsiounis, and M. Yung, *Anonymity control in e-cash systems*, in *Financial Cryptography, Anguilla, British West Indies*, K. Kurosawa, ed., Springer, Berlin, Heidelberg, 1997, pp. 1–16.

[10] Z. Eslami and M. Talebi, *A new untraceable off-line electronic cash system*, Electron. Commerce Res. Appl. 10(1) (2011), pp. 59–66.

[11] C.-I. Fan and V.S.-M. Huang, *Provably secure integrated on/off-line electronic cash for flexible and efficient payment*, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 40(5) (2010), pp. 567–579.

[12] S. Goldwasser and Y.T. Kalai, *On the (in) security of the Fiat–Shamir paradigm*, in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, Cambridge, USA*, J. Simon, ed., IEEE, Washington, DC, 2003, pp. 102–113.

[13] S. Goldwasser, S. Micali, and C. Rackoff, *The knowledge complexity of interactive proof systems*, SIAM J. Comput. 18(1) (1989), pp. 186–208.

[14] J. Groth, *Fully anonymous group signatures without random oracles*, in *Advances in Cryptology – ASIACRYPT 2007, Kuching, Malaysia*, K. Kurosawa, ed., Springer, Berlin, Heidelberg, 2007, pp. 164–180.

[15] J. Groth and A. Sahai, *Efficient noninteractive proof systems for bilinear groups*, SIAM J. Comput. 41 (2012), pp. 1193–1232.

[16] F. Li, M. Zhang, and T. Takagi, *Identity-based partially blind signature in the standard model for electronic cash*, Math. Comput. Model. 58 (2012), 196–203.

[17] A. Lysyanskaya and Z. Ramzan, *Group blind digital signatures: A scalable solution to electronic cash*, in *Financial Cryptography, Anguilla, British West Indies*, R. Hirchfeld, ed., Springer, Berlin, Heidelberg, 1998, pp. 184–197.

[18] G. Maitland and C. Boyd, *Fair electronic cash based on a group signature scheme*, in *Information and Communications Security, Xi'an, China*, S. Qing, T. Okamoto, and J. Zhou, eds., Springer, Berlin, Heidelberg, 2001, pp. 461–465.

[19] T. Nishide, S. Miyazaki, and K. Sakurai, *Security analysis of offline e-cash systems with malicious insider*, J. Wireless Mob. Netw. Ubiquitous Comput. Dependable Appl. 3(1/2) (2012), pp. 55–71.

[20] D. Pointcheval and J. Stern, *Security arguments for digital signatures and blind signatures*, J. Cryptol. 13(3) (2000), pp. 361–396.

[21] S. von Solms and D. Naccache, *On blind signatures and perfect crimes*, Comput. Secur. 11(6) (1992), pp. 581–583.

[22] S. Wang, Z. Chen, and X. Wang, *A new certificateless electronic cash scheme with multiple banks based on group signatures*, in *IEEE 2008 International Symposium on Electronic Commerce and Security, Guangzhou, China*, N. Jun, ed., IEEE Computer Society Press, Washington, DC, 2008, pp. 362–366.