International Journal of Innovative Computing, Information and Control
Volume **8**, Number **7(A)**, July **2012**

ICIC International ©2012 ISSN 1349-4198
pp. 4469–4482

AN EFFICIENT ELECTRONIC CASH SCHEME WITH MULTIPLE BANKS USING GROUP SIGNATURE

Ming-Te Chen[1], Chun-I Fan[1,*], Wen-Shenq Juang[2] and Yi-Chun Yeh[2]

[1]Department of Computer Science and Engineering
National Sun Yat-sen University
No. 70, Lienhai Road, Kaohsiung 80424, Taiwan
ecsemtchen@gmail.com; *Corresponding author: cifan@faculty.nsysu.edu.tw

[2]Department of Information Management
National Kaohsiung First University of Science and Technology
No. 2, Jhuoyue Road, Nanzih, Kaohsiung 811, Taiwan
wsjuang@ccms.nkfust.edu.tw; u9724830@nkfust.edu.tw

Received March 2011; revised July 2011

Abstract. *In 2008, an electronic cash scheme with multiple banks based on group signatures was proposed by Wang et al. They adopted a group blind signature scheme based on bilinear pairings to generate the electronic cash and it can be verified by the bilinear pairings operation. However, we find some security problems in their approach. By the way, the cost of communication and computation in their scheme can be improved further. Hence, we propose an efficient and secure e-cash scheme from bilinear pairings with multiple banks. Not only can our approach solve all the security problems in Wang et al.'s scheme but also offer lower computation and communication cost.*

Keywords: Blind group signature, Bilinear pairing, E-cash, Multiple banks

1. Introduction.

With the flourishing development of the Internet technology, the phenomenon of people performing financial transactions via the Internet is gradually popular in the e-commerce environment. This situation is called electronic payment service [9, 13, 18]. Because of the insecure Internet environment, customers will face any kinds of security threats when performing electronic payment service with banks. A malicious attacker can carry out eavesdropping, tampering, stealing or performing other illegal acts on the customers' transaction data when they are doing this service with banks. Then it will result in that consumers' sensitive privacy information (such as customers' identity and password of financial cards) is stolen and she/he can impersonate one of customers to withdraw e-cash from banks. In order to prevent these threats happening, the electronic payment services must consider the security requirements including the authentication of customers, confidentiality of e-cash, and non-repudiation of e-cash.

When a customer pays her/his e-cash to a merchant, it should make that the merchant and the bank do not know who pays the e-cash anonymously. By the way, the merchant should be able to check the e-cash fast by using efficient e-cash verification method. In 1983, the first electronic cash (e-cash) was proposed by Chuam [4] and it adopted the blind signature as the building primitive.

In the meanwhile, there were some signature schemes [2, 21, 23] and some electronic cash payment mechanisms [14, 19, 22] also proposed. For the growing emphasis on the privacy protection of customers in electronic payment systems, the blind signature seems to be a perfect solution. Nevertheless, the blind signature cannot offer the fully anonymity protection to the customers. So the problem of unconditional anonymity was also indicated by Solms and Naccache [16]. On the other hand, in order to provide the traceability on the e-cash, a fair blind signature scheme was proposed by Stadler et al. [12] in 1996. This scheme allows the judge to trace and derive the real identity of the customer if needed.

To our best knowledge, some papers focus on making the key management processes simple in traditional PKI environment. However, there has the certificate management and revoking problems in these papers. In order to solve these problems, the idea of identity-based signature and encryption was proposed by Shamir [1] in 1985. The main concept shows that the system embeds the customer's public information (ex: name, e-mail address, or other identity information) into her/his public key in the key generation stage. Each customer can easily verify their public key through the public information without interacting with CA (Certificate Authority). Hence, the CA can also reduce the management loading of each customer's public key and certificate. In 1997, Park et al. [15] proposed an ID-based group signature scheme which used the public keys and identities of group members to verify the group signature produced by one of members. Moreover, there must be an assumption that the group signing key must be produced by a trusted third party called the group manager. Under this assumption, it also causes the key escrow problem due to the trusted level of the group manager. In 2003, in order to solve the key escrow problem [5] in the ID-based system, Chen et al. [20] introduced a new ID-based system based on pairings.

In general, most of the above proposed e-cash schemes assume that the customers and the merchants open their accounts in the same bank and only this dedicated bank can withdraw and deposit the e-cash to their accounts, respectively. However, the customers and the merchants may belong to the different banks in real life. This causes the inconvenient situation by using one of the above schemes directly. Thus, in order to solve this problem, there were many papers proposed [8, 17] in literature. In 2001, an electronic cash system with multiple banks was proposed by Zhang et al. [8]. In 2008, Wang et al. [17] proposed an electronic cash scheme based on an ID-based group signature. Their scheme is to remove the assumption of a trusted third party in Zhang et al.'s system. Wang et al. also claimed that their scheme is secure. Nevertheless, we found that it does not satisfy the unforgeable requirement since an attacker can impersonate to be a customer withdrawing a valid e-cash from the bank in their scheme. In order to solve this problem, we propose an efficient e-cash scheme with multiple banks that not only can preserve all the nice properties of Wang et al.'s scheme but also can solve the security problems in their scheme.

The remainder of this paper is organized as follows. We briefly introduce some preliminaries used in our scheme in Section 2. Our e-cash construction is presented in Section 3. In Section 4, we describe the correctness and security considerations of our scheme and performance comparison is also shown. In Section 5, a concluding remark is given.

2. **Preliminary.**

2.1. **Bilinear pairings and the underlying assumptions.**

The pairings refer to the corresponding linear map relationship between the two cyclic groups, so it is also called an admissible bilinear map. The set consists of all points on the elliptic curve which must establish the relationship of "group" in the abstract algebraic geometry, so the operation of the bilinear pairings can be applied to elliptic curve exactly. The bilinear pairings can be derived from the Weil or Tate pairings. The related parameters and symbols of the bilinear pairings are as follows. Let $(G_1, +)$ denote a cyclic additive group of a prime order q and $(G_2, \times)$ denote a cyclic multiplicative group of a prime order q. Let $e : G_1 \times G_1 \rightarrow G_2$ be a bilinear pairing. The bilinear pairing must satisfy the following three properties.

1. Bilinear: for $\forall P, Q \in G_1$ and $a, b \in Z_q^*$,
$$e(aP, bQ) = e(abP, Q) = e(P, abQ) = e(P, Q)^{ab}.$$
2. Non-degeneration: for all $P \in G_1$ and $Q \in G_1$, such that $e(P, Q) \neq 1$.
3. Computability: for any $P, Q \in G_1$, there is an efficient algorithm to compute $e(P, Q)$ in polynomial time.

We continue to make a description of some hard problems which are related to bilinear pairings.

1. Discrete Logarithm Problem (DLP):
Given two elements $P, Q \in G_1$, DLP is to calculate an integer n, such that $Q = nP$.
2. Computational Diffie-Hellman problem (CDHP):
Given (P, aP, bP), which $P \in G_1$ and $a, b \in Z_p^*$, CDHP is to compute abP.
3. Inverse Computational Diffie-Hellman Problem (Inv-CDHP):
Given P, aP, $a \in Z_p^*$, Inv-CDHP is to compute $a^{-1}P$. It is a variation of CDHP.

We assume that there does not exist any polynomial time algorithm to solve DLP, CDHP, and Inv-CDHP with a non-negligible probability.

2.2. **Tan et al.'s ID-based group signature scheme from bilinear maps.** In 2003, an ID-based group signature scheme from bilinear maps was proposed by Tan et al. [25]. Their scheme is based on the key-escrow method in Chen et al.'s scheme [20]. Let G_1 and G_2 be two cyclic groups of a prime order q and $e : G_1 \times G_1 \rightarrow G_2$ be a bilinear pairing function. Let $h_1 : \{0, 1\}^* \times G_1 \rightarrow Z_q^*$ and $h_2 : \{0, 1\}^* \times G_1 \rightarrow G_1$ be two hash functions. We briefly describe it in the following.

1. The setup phase: The key control center (KCC for short) first announces the public information $(e, q, P, P_{pub}, G_1, G_2, h_1, h_2)$. Let $P_{pub} = sP$ be the public key of KCC, where s be a master private key and h_1, h_2 be two hash functions.
2. The extracting phase: A user A computes her/his public key rP, where r is a random number. A transmits rP and her/his identity ID_A to KCC. Then KCC computes $S_{ID_A} = sh_2(ID_A \| rP)$ and sends it back to A.
3. The joining phase: A computes (rP, xP, ID_A, rxP) and forwards it to KCC, where x is a random number. KCC uses the equation $S_{ID_A} = sh_2(ID_A \| rP)$ to prove that A knows S_{ID_A} and verifies whether $e(rxP, P) = e(xP, rP)$ or not. If these two equations are satisfied, KCC generates the partial certificate $S_A = sh_2(ID_A \| rxP)$ for A. Thus, (S_A, rxP) is the member certificate of ID_A.
4. The signing phase: Let m be the message to be signed. A generates $U = k_1 rxP$ and computes $W = (q - k_1)xP$, $R = k_2 h_2(ID_A \| U + W)$, $H = h_1(U + W + R)$ and $V = Hk_2 S_A + k_1 rxh_2(m \| U + W + R)$, where k_1, k_2 are two random values of Z_q^*. Finally, (U, W, R, V) is the group signature of the message m.
5. The verification phase: The verifier has to compute $H = h_1(U + W + R)$ to check the validity by using the equation $e(V, P) = e(R, P_{pub})^H \cdot e(h_2(m \| U + W + R), U)$.
6. The opening phase: When KCC gets a signature of m, KCC can use the following three equations $e(U, P)e(W, rP) = e(rxP, P)$, $e(S_A, P) = e(h_2(ID_A \| rxP), P_{pub})$, and $e(S_{ID_A}, P) = e(h_2(ID_A \| rP), P_{pub})$ to find out the identity of the user A.

3. **Our Proposed Scheme.**

Now we propose our e-cash scheme based on group signatures. There are four kinds of participants: the central bank, the banks, the customer and the merchant which are involved in our scheme. The central bank needs to do the enrollment of any bank and any customer in a group and also records all the related information about the legitimate members of the group which includes the members' public keys. The central bank has privilege to manage and revoke the bank or customer's permissions in the group. If the double spending happens, it can trace and reveal the real identity of the customer. On the other hand, each bank can issue the e-cash to the customer. Before issuing the e-cash, the bank must verify whether the customer is the group member or not. If yes, the signing process will continue. Otherwise, this transaction will be aborted. Any customer can withdraw the e-cash from her/his registered bank. In the payment phase, the merchant needs to verify the signature of e-cash and transaction information provided from the customer.

3.1. **Notations.**

The notations used in our proposed scheme are defined in the following.

1. $\widehat{e}$: a bilinear pairing function, where $\widehat{e} : G_1 \times G_1 \to G_2$.
2. $(G_1, +)$, $(G_2, \cdot)$: two cyclic groups which are generated from the original point P in an order q.
3. $H(\cdot)$, $H_1(\cdot)$: two hash functions, where $H(\cdot)$: $\{0,1\}^* \times G_1 \longrightarrow Z_q^*$ and $H_1(\cdot)$: $\{0,1\}^* \times G_1 \longrightarrow G_1$.
4. $(r_c, S_{ID_{C_i}})$: the private key of the customer with the identity ID_{C_i}.
5. $(r_b, S_{ID_{B_i}})$: the private key of the bank with the identity ID_{B_i}.
6. $\left(r_cP, x_{c_i}P, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$: three public keys of the customer ID_{C_i}.
7. $\left(\frac{1}{H(x_{c_i}P)+H(r_c)}P, S_{C_i}\right)$: the member certificate of the customer ID_{C_i}.

3.2. **The setup protocol.**

The central bank generates her/his master secret key $s \in Z_q^*$ and her/his public key $P_{pub} = sP$. The information $< G_1, G_2, \widehat{e}, q, P, P_{pub}, H, H_1 >$ are announced by the central bank. Then, the customer and each bank must set up their public/private keys in the central bank, respectively.

A. The customer:

Step 1: The customer randomly chooses a secret value $r_c \in Z_q^*$ and generates the corresponding public key r_cP. Then she/he sends the public key to the central bank.

Step 2: After the central bank received r_cP, she/he will compute $S_{ID_{C_i}} = sQ_{ID_{C_i}} = sH_1(ID_{C_i} \| t_i, r_cP)$ with the master key s. Then the central bank will forward $S_{ID_{C_i}}$ back to the customer. Let $Q_{ID_{C_i}}$ be the public key of ID_{C_i} and t_i denote the valid time period for r_c. After preparing these information, she/he forwards them to ID_{B_i} securely.

Step 3: Finally, the customer can obtain her/his private key $(r_c, S_{ID_{C_i}})$.

B. The bank:

Step 1: The bank computes the public key r_bP with a random value $r_b \in Z_q^*$ and forwards it to the central bank.

Step 2: After receiving it from the bank B_i, the central bank produces $S_{ID_{B_i}} = sQ_{ID_{B_i}} = sH_1(ID_{B_i} \| T_i, r_bP)$, where $Q_{ID_{B_i}}$ is the public key of ID_{B_i} and T_i denotes the valid time period for r_b. Then she/he forwards this information to ID_{B_i} securely.

Step 3: The bank B_i also can obtain the private key $(r_b, S_{ID_{B_i}})$ for her/him.

3.3. **The registration protocol.**

Any new group member (B_i or C_i) who attempts to participate in the e-cash payment scheme must perform a request to the central bank for allowing her/him joining the group.

If a customer C_i attempts to join a group, she/he then performs in the following.

Step 1: C_i chooses a secret value $x_{c_i} \in Z_q^*$ and generates the relative public key $x_{c_i}P$. C_i then submits the information $\left(r_cP, x_{c_i}P, \frac{1}{H(x_{c_i}P)+H(r_c)}P, ID_{C_i}, S_{ID_{C_i}}\right)$ to the central bank.

Step 2: When the central bank obtains the information, she/he uses the following two verification Equations (1) and (2) to verify whether the public keys ID_{C_i} and $S_{ID_{C_i}}$ are correct or not. If yes, the central bank will send $S_{c_i} = sH_1\left(t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$ to ID_{C_i} secretly. Otherwise, this phase will be stopped. Then the central bank records $\left(r_cP, x_{c_i}P, \frac{1}{H(x_{c_i}P)+H(r_c)}P, ID_{C_i}\right)$ in the customer member table in private.

$$S_{ID_{C_i}} \overset{?}{=} sH_1(ID_{C_i}\|t_i, r_cP), \tag{1}$$

$$\hat{e}\left(H(x_{c_i}P)P + H(r_c)P, \frac{1}{(H(x_{c_i}P) + H(r_c))}P\right) \overset{?}{=} \hat{e}(P, P). \tag{2}$$

Step 3: On receiving the partial member certificate S_{c_i} in Step 2, C_i can verify S_{c_i} by computing $\hat{e}(S_{c_i}, P) = \hat{e}\left(H_1(t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P), P_{pub}\right)$. If yes, the customer obtains her/his complete member certificate $\left(\frac{1}{H(x_{c_i}P)+H(r_c)}P, S_{c_i}\right)$.

Step 4: Then the central bank will forward the information $v_i = H(ID_{B_i}, S_{c_i})$ to each eligible bank B_i $(i = 1, 2, 3, \ldots, k)$ via a secure channel.

If a bank B_i attempts to join a group, then she/he performs in the following. As the above three steps done by the customer, B_i must run the same process to get her/his own member certificate. Then B_i can use this certificate to generate the signature of electronic cash later. The registration process is shown in the following.

Step 1: B_i randomly chooses a secret number $\alpha_{b_i} \in Z_q^*$ and computes the public key $\alpha_{b_i}P$. She/he forwards the information $\left(r_bP, \alpha_{b_i}P, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P, ID_{B_i}, S_{ID_{B_i}}\right)$ to the central bank.

Step 2: When the central bank receives the request from B_i, she/he can check the correctness of the request by verifying if $S_{ID_{B_i}} \overset{?}{=} sH_1(ID_{B_i}\|T_i, r_bP)$ and $\hat{e}\Big(H(\alpha_{b_i}P)P$ $+H(r_b)P, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\Big) \overset{?}{=} \hat{e}(P, P)$. If yes, the central bank records $\Big(r_bP,\ \alpha_{b_i}P,$ $\frac{1}{H(\alpha_{b_i}P)+H(r_b)}P,\ ID_{B_i}\Big)$ about B_i in the bank member table. Then she/he generates $S_{B_i} = sH_1\left(T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$ and sends it back to B_i secretly.

Step 3: Finally, B_i can verify S_{B_i} by using $\hat{e}(S_{B_i}, P) = \hat{e}\left(H_1\left(T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right), P_{pub}\right)$. If it is valid, B_i will get her/his member certificate $\left(S_{B_i}, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$.

3.4. **The withdrawal protocol.** The participants in this phase are the customer and the bank. Before the customer requests for the signature of electronic cash from B_i, they should perform the authentication process. The customer and the bank have to prove that they know the secret S_{c_i} between them. The authentication phase is shown in the following.

Step 1: The customer chooses a random number $rc_j \in Z_q^*$ and computes the hash value $v_i = H(ID_{B_i}\|S_i)$, then she/he sends a request with a nonce γ_1 and the encrypted message $E_{v_i}(rc_j, H(ID_{B_i}\|\gamma_1))$ to the bank B_i.

Step 2: When B_i receives the information from C_i, she/he will use the information $v_i = H(ID_{B_i}\|S_i)$ received from the central bank to decrypt the encrypted message $E_{v_i}(rc_j, H(ID_{B_i}\|\gamma_1))$ and check whether γ_1 is fresh or not. If yes, she/he chooses $rb_j \in Z_q^*$ and sends back the encrypted message $E_{v_i}(rb_j, \gamma_1+1, \gamma_2)$ to C_i.

Step 3: C_i decrypts the received information by computing $D_{v_i}(E_{v_i}(rb_j, \gamma_1+1, \gamma_2))$ and checking if the nonce γ_2 is fresh. If yes, she/he computes $k_j = H(rb_j, rc_j, v_i)$. Then she/he forwards the encrypted message $E_{k_j}(\gamma_2+1)$ to B_i.

Now, we describe our proposed withdrawing process. We assume that the customer has passed the identity authentication with the bank B_i, so she/he can run the following steps with B_i to withdraw her/his e-cash.

Step 1: B_i randomly chooses a secret value $\kappa \in Z_q^*$ and then forwards the value $R = \kappa H_1\left(T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$ to C_i.

Step 2: C_i randomly chooses a blinding factor $b \in Z_q^*$ to transform R to $\tilde{R}$ and computes h, where $\tilde{R} = R + bP$ and $h = H(m, \tilde{R})$. Then C_i sends h back to B_i.

Step 3: B_i computes the information $\tilde{V}$ and W, where $\tilde{V} = \frac{1}{H(\alpha_{b_i}P)+H(r_b)}h$ and $W = (\kappa + \tilde{V})S_{B_i}$, then sends them to C_i.

Step 4: C_i computes $\tilde{W} = W + bP_{pub}$.

If all steps are executed successfully, C_i can get the blind group signature of m signed by B_i. C_i can acquire $\left(m, \tilde{R}, \tilde{V}, \tilde{W}, T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$ from B_i. Then C_i can use the following steps to verify the validity of $\left(m, \tilde{R}, \tilde{V}, \tilde{W}, T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$. First, C_i should compute $h = H(m, \tilde{R})$ and $Q = H_1\left(T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$. She/he uses $e(\tilde{W}, P) = e(\tilde{R} + \tilde{V}Q, P_{pub})$ to check if the e-cash is valid. If yes, the customer can accept the e-cash issued by B_i.

3.5. **The payment protocol.** When the customer C_i pays the e-cash to the merchant ID_s, the merchant needs to check if the e-cash is correct by using the group public key. If the signature $\left(m, \tilde{R}, \tilde{V}, \tilde{W}, T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$ is valid, the customer signs the information on the transaction θ with her/his secure key $\frac{1}{H(x_{c_i}P)+H(r_c)}$ and member certificate $\left(\frac{1}{H(x_{c_i}P)+H(r_c)}P, S_{c_i}\right)$, where $\theta = (ID_s\|time)$ including the identity ID_s of the merchant and the current trading time $time$.

Step 1: C_i chooses a random number $\varpi \in Z_q^*$ and computes the random value $R'' = \varpi H_1\left(t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$, $h'' = H(\theta, R'')$, $V'' = \frac{1}{H(x_{c_i}P)+H(r_c)}h''$ and $W'' = (\varpi + V'')S_{c_i}$.

Step 2: C_i sends $\left(\theta, R'', V'', W'', t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$ to the merchant.

Step 3: When the merchant receiving it from the customer, she/he checks whether t_i expire or not. If no, she/he then computes $\hat{Q} = H_1\left(t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$ and checks the correctness of $\left(\theta, R'', V'', W'', t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$ by verifying if $\hat{e}(W'', P) \overset{?}{=} \hat{e}\left(R'' + V''\hat{Q}, P_{pub}\right)$. If yes, the customer can get the goods.

3.6. **The deposit protocol.** We also assume that the central bank has a check list which records the used e-cash information. We assume that the merchant has an account in another bank B_j. When the merchant ID_s has to deposit the e-cash into her/his account, she/he must send the information $\left(m, \tilde{R}, \tilde{V}, \tilde{W}, T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$ and $\left(\theta, R'', V'', W'', t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$ to the bank B_j. The bank can perform if $\left(\theta, R'', V'', W'', t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$ is valid by checking $\hat{e}(W'', P) \overset{?}{=} \hat{e}(R'' + V''\hat{Q}, P_{pub})$ and verifying if $\left(m, \tilde{R}, \tilde{V}, \tilde{W}, T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$ has not been used before. If both of the verification are valid, B_j deposits the money into the merchant's account.

3.7. **The customer tracing protocol.** If the bank B_j discovers the double spending of some e-cash, she/he can ask the central bank to find out the owner of the e-cash. The central bank requests the information $\left(m, \tilde{R}, \tilde{V}, \tilde{W}, T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$ and $\left(\theta, R'', V'', W'', t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$ from B_j. She/he uses $\frac{1}{H(\alpha_{b_i}P)+H(r_b)}P$ to check the bank member list and finds out who the real signer of the e-cash is. Then she/he checks if the following two equations hold.

$$\hat{e}\left(H(x_{c_i}P)P + H(r_c)P, \frac{1}{(H(x_{c_i}P) + H(r_c))}P\right) \overset{?}{=} \hat{e}(P, P), \tag{3}$$

$$\hat{e}(S_{c_i}, P) \overset{?}{=} \hat{e}\left(H_1(t_i, \frac{1}{H(x_{c_i}P) + H(r_c)}P), P_{pub}\right). \tag{4}$$

If these two equations are valid, the double spending customer will be found.

3.8. **The revoking protocol.** Assume that the central bank has a certificate revocation list which records the information of the revoked group members. We assume that the format of each item in the certificate revocation list is $(Q_{ID_{B_i}}, T_{revoke})$ and $(Q_{ID_{C_i}}, t_{revoke})$ for the bank and the customer, respectively, where $(Q_{ID_{B_i}}, T_{revoke})$ is a group member with the public key $Q_{ID_{B_i}}$ (or $Q_{ID_{C_i}}$) at the time T_{revoke} (or t_{revoke}). Anyone can request the certificate revocation list to check whether the communicating party is included in a group or not.

4. **Security Analysis.**

4.1. **Correctness considerations.**

Proposition 4.1. *In the registration phase, the central bank can check the validity of the public key of the registrant (the customer or the bank) by verifying the equation* $\hat{e}\left(H(x_{c_i}P)P + H(r_c)P, \frac{1}{(H(x_{c_i}P)+H(r_c))}P\right) = \hat{e}(P, P)$.

Proof:

$$\begin{aligned}&\hat{e}\left(H(x_{c_i}P)P + H(r_c)P, \frac{1}{(H(x_{c_i}P) + H(r_c))}P\right)\\ =&\,\hat{e}((H(x_{c_i}P) + H(r_c))P, (H(x_{c_i}P) + H(r_c))^{-1}P)\\ =&\,\hat{e}(P, P)^{(H(x_{c_i}P)+H(r_c))*(H(x_{c_i}P)+H(r_c))^{-1}}\\ =&\,\hat{e}(P, P).\end{aligned}$$

Proposition 4.2. *In the withdrawal phase of our scheme, the equation* $\hat{e}(\tilde{W}, P) = \hat{e}(\tilde{R} + \tilde{V}Q, P_{pub})$ *is used to check the correctness of the signature* $\left(m, \tilde{R}, \tilde{V}, \tilde{W}, T_i, \frac{1}{H(r_bP)+H(\alpha_{b_i})}P\right)$ *of the e-cash by the customer. In the payment phase, the merchant also can use the equation* $\hat{e}(\tilde{W}, P) = \hat{e}(\tilde{R} + \tilde{V}Q, P_{pub})$ *to verify the e-cash of the customer.*

Proof:

$$\begin{aligned}
&\hat{e}(\tilde{W}, P)\\
&= \hat{e}(W + bP_{pub}, P)\\
&= \hat{e}(W, P)\hat{e}(bP_{pub}, P)\\
&= \hat{e}((\kappa + \tilde{V})S_{B_i}, P)\hat{e}(bP, P_{pub})\\
&= \hat{e}\left(\kappa H_1\left(T_i, \frac{1}{H(\alpha_{b_i}P) + H(r_b)}P, sP\right)\hat{e}\left(\tilde{V}H_1\left(T_i, \frac{1}{H(\alpha_{b_i}P) + H(r_b)}P\right), sP\right)\right)\hat{e}(bP, P_{pub})\\
&= \hat{e}(R + bP, P_{pub})\hat{e}\left(\tilde{V}H_1\left(T_i, \frac{1}{H(\alpha_{b_i}P) + H(r_b)}P\right), P_{pub}\right)\\
&= \hat{e}(\tilde{R} + \tilde{V}Q, P_{pub}).
\end{aligned}$$

Proposition 4.3. *In the payment phase of our proposed e-cash scheme, the merchant can verify the correctness of the signature* $\left(\theta, R'', V'', W'', t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$ *by utilizing the equation* $\hat{e}(W'', P) = \hat{e}(R'' + V''\hat{Q}, P_{pub})$.

Proof:

$$\begin{aligned}
&\hat{e}(W'', P)\\
&= \hat{e}((\varpi + V'')S_{c_i}, P)\\
&= \hat{e}(\varpi S_{c_i} + V''S_{c_i}, P)\\
&= \hat{e}\left(\varpi H_1\left(t_i, \frac{1}{H(x_{c_i}P) + H(r_c)}P\right) + V''H_1\left(t_i, \frac{1}{H(x_{c_i}P) + H(r_c)}P\right), P_{pub}\right)\\
&= \hat{e}(R'' + V''\hat{Q}, P_{pub}).
\end{aligned}$$

4.2. **Unforgeability.** The withdrawal phase of our proposed scheme can withstand the forgeability of e-cash. Assume that someone attempts to disguise as the bank ID_{B_i} to issue an e-cash. Since an e-cash is generated by the private key $\frac{1}{H(\alpha_{b_i}P)+H(r_b)}$ and the bank's member certificate $\left(S_{B_i}, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$, where r_b is generated randomly and hashed by the hash function H. No one can know the secret r_b except the bank herself/himself. In addition, before the bank signs an e-cash, she/he has to prove that she/he knows the shared-secret message v_i by running the authentication protocol. If the customer C_i performs the mutual authentication and sends the encrypted information $E_{v_i}(rc_j, H(ID_{B_i}\|\gamma_1))$ to B_i, we assume that B_i must know v_i. Then she/he can use v_i to decrypt it and get rc_j and $H(ID_{B_i}\|\gamma_1)$. Otherwise, she/he cannot do the next step in the authentication protocol.

If some malicious user $\mathcal{M}$ attempts to impersonate to be a legitimate customer C_1 to withdraw an e-cash from the bank B_i, $\mathcal{M}$ has to prove that she/he knows the secret certificate S_{C_1} which is shared between C_1 and B_i. But the shared-secret certificate is transmitted through a secret channel and the identity of the bank and S_{C_1} is hashed by the hash function H. $\mathcal{M}$ cannot learn the partial certificate of C_1 from v_i and H.

4.3. **Withstanding double spending.** The deposit phase of our proposed scheme can prevent the situation of double spending. In order to avoid the e-cash to be used twice, after the merchant has sent the e-cash to her/his corresponding bank B_j, the bank B_j can connect to the online database to check whether the e-cash exists or not. If yes, the bank B_j can ask for the central bank to find out the customer by performing the customer tracing protocol. If the e-cash has never been used, it will be deposited in the account of the merchant by the bank B_j and the central bank will record this used e-cash information.

4.4. **Anonymity.** When the merchant obtains the e-cash $\left(m, R', V', W', T_i, \frac{1}{H(\alpha_{b_i}P)+H(r_b)}P\right)$ and the signature of the transaction information $\left(\theta, R'', V'', W'', t_i, \frac{1}{H(x_{c_i}P)+H(r_c)}P\right)$ from ID_{C_i} in the payment stage, she/he cannot learn the identity of ID_{C_i}. Since x_{c_i} and r_c are randomly chosen and they are protected by the hash function H, no one can link the customer's identity with $\frac{1}{H(x_{c_i}P)+H(r_c)}P$ except the central bank (as described in the customer tracing protocol).

4.5. **Properties comparison.** In this section, we will describe the properties comparison among our e-cash scheme and related schemes. We summarize the comparison in Table 1. In Table 1, we can see that our e-cash scheme not only can satisfy all the security properties such as customer's anonymity protection, mutual authentication, e-cash verification, and customer tracing protocol but also can prevent double spending and the e-cash forging.

4.6. **Efficiency analysis and comparison.** Our scheme is based on the bilinear mapping in elliptic curves. Comparing with RSA, a small length key will be used to achieve the same security of RSA. For example, in ECC, the key length of 192-bit has the same security level with the key length of 1024-bit in RSA.

We assume that EC_p is the pairing operation on elliptic curve, EC_m is the point scalar multiplication operation on elliptic curve, EC_A is two points addition operation on elliptic curve, T_H is the computation cost of one-way hash function, T_s is the search time, $T(S)$ is the time of a symmetric encrypting/decrypting operation, $T(D)$ is the time of a Diffie-Hellman operation, I is the computation cost of the inverse operation, $\oplus$ is the computation cost of the exclusive-or operation, and M is the multiplication operation in a modulo as referenced in [6, 7]. Let *C1* be the communication cost, *C2* be the computation cost of the central bank, *C3* be the the computation cost of the customer, *C4* be the computation cost of the bank, *C5* be the computation cost of the merchant, *C6* be the computation cost of TTP, and *C7* be the computation cost of verification. We assume that a random number in Z_q^* is 160 bits, a point over elliptic curve is 160 bits, the output size of SHA-1 is 160 bits, and the block size of AES is 128 bits. From [24], we assume that E is the computation cost of a modulo exponentiation in a 1024-bit modulo and we also find out the relations such as $E \cong 8.24EC_m$, $E \cong 600T_H$, $E \cong 3.2EC_p$, $EC_A \cong 5M$, $I \cong 0.9M$ and $E \cong 240M$. The performance comparisons are shown in Tables 2-7.

From Table 3, we can see that the total computation cost in Popescu et al.'s scheme is about $342M$, that in Chou et al.'s scheme [10] is about $885M + 2T(D)$, that in Wang et al.'s scheme is about $372M$, and that in our scheme is about $571.9M$.

From Table 4, we can see that the communication cost of the withdrawing protocol in our scheme is $160 * 4 = 640$ bits and the communication cost for the authentication protocol in our scheme is 640 bits. Thus, *C1* in Table 4 is only 1280 bits. In Wang et al.'s scheme [17], the communication cost of the withdrawing protocol is $160 * 7 = 1120$ bits and the communication cost for the authentication protocol is 640 bits. Thus, *C1* of [17] in Table 4 is 1760 bits. The communication cost of the withdrawing protocol in Chou et al.'s scheme [10] is $384 + 1952 = 2336$ bits and Popescu et al.'s scheme is $160 * 7 = 1120$ bits. In [3], they do not provide the authentication protocol before performing the issuing phase. We only need three messages to complete the issue of e-cash. On the other hand, there are five messages in Wang et al. and Popescu et al.'s scheme.

In Table 4, we include the cost of communications and computation of authentication protocol in the withdrawal protocol. The computation cost of withdrawing an e-cash for

Table 1. Properties comparisons among our e-cash scheme and related schemes

	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	P11
Our	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Wang [17]	Yes	No	Yes	Yes	No	Yes	No	Yes	No	No	Yes
Chou [10]	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	Yes
Popescu [3]	Yes	No	No	Yes	No	Yes	No	No	No	No	No

P1: Anonymity of customer;
P2: Providing the tracing function;
P3: Supporting authentication in the withdrawal phase;
P4: Non-repudiability;
P5: Preventing forging the e-cash;
P6: Without the help of a TTP in the tracing protocol;
P7: Revokability;
P8: Supporting multiple banks;
P9: The verification of the member certificate for the customer;
P10: Verifiability;
P11: The verification of the e-cash for the customer in the withdrawal protocol.

Table 2. Computation cost comparisons of the setup protocol among our e-cash scheme and related schemes

	Our	Wang [17]	Chou [10]	Popescu [3]
C1	656 bits	656 bits	1152 bits	0 bits
C2	$6EC_m$ $\cong 180M$	$6EC_m$ $\cong 180M$	$5EC_m + 1T_H$ $\cong 151M$	$1EC_m$ $\cong 30M$

the customer in Wang et al.'s scheme [17] is five scalar multiplication of point operation in elliptic curve, three addition of point operation in elliptic curve, three hash function operation, and one multiplication operation in module. The total cost is about $168M$. In Chou et al.'s scheme [10], there are one pairings operation in elliptic curve, three scalar multiplication of point operation in elliptic curve, two hash function operation, five multiplication operation in module, one inverse operation, and two Diffie-Hellman operation. The total cost is about $176.9M + 2T(D)$. In Popescu et al.'s scheme [3], there are three scalar multiplication of point operation in elliptic curve, four addition of point operation in elliptic curve, and two hash function operation. The total cost is about $111M$. In our scheme, there are two scalar multiplication of point operation in elliptic curve, two addition of point operation in elliptic curve, three hash function operation, and three symmetric encrypting/decrypting operation. The total cost is about $71M + 3T(S)$.

The total computation cost of the withdrawing protocol in Wang et al.'s scheme [17] is about $615M$. That in Chou et al.'s scheme is about $1235.6M + 4T(D)$. That in Popescu et al.'s scheme [3] is about $232M$. That in our scheme is about $457M + 6T(S)$. In Popescu et al.'s scheme [3], they do not provide the verification function of the e-cash for the customer.

In Table 5, the computation cost of the customer in our scheme is about $61M$ for the payment phase. The computation cost of the customer in Wang et al.'s scheme is about $96M$, that in [10] is about $111M + 1T(D)$, and that in [3] is about $127M$. In the verification process, we only need four pairing operations, eight point scalar multiplication operations, and four addition of point operations to complete the verification precess.

TABLE 3. Cost comparisons of the registration protocol among our e-cash scheme and related schemes

	Our	Wang [17]	Chou [10]	Popescu [3]
C1	1088 bits	832 bit	1184 bits	1152 bits
C2	$1EC_m + 1T_H$ $\cong 31M$	$1EC_m + 1T_H$ $\cong 31M$	N/A	$1EC_m + 1T_H$ $\cong 31M$
C3	$2EC_m + 2T_H$ $\cong 62M$	$2EC_m$ $\cong 60M$	$1EC_p + 1EC_m +$ $1T_H$ $\cong 221M$	$3EC_m$ $\cong 90M$
C4	$2EC_m + 2T_H$ $\cong 62M$	$2EC_m$ $\cong 60M$	N/A	N/A
C6	N/A	N/A	$1EC_p + 5EC_m +$ $4T_H + 1EC_A +$ $1M$ $\cong 236.9M$	N/A
C7	$4EC_p + 3EC_m +$ $4T_H + 1EC_A +$ $1M$ $\cong 416.9M$	$2EC_p + 2EC_m +$ $1T_H$ $\cong 221M$	$2T(D) + 3EC_p +$ $6EC_m + 1T_H +$ $1EC_A + 1M$ $\cong 427.9M + 2T(D)$	$2EC_p + 2EC_m +$ $1T_H$ $\cong 221M$
Total	$571.9M$	$372M$	$885M$+$2T(D)$	$342M$

N/A: Not Available

TABLE 4. Cost comparisons of the withdrawal protocol among our e-cash scheme and related schemes

	Our	Wang [17]	Chou [10]	Popescu [3]
C1	1280 bits	1760 bits	2336 bits	1120 bits
C3	$2EC_m + 2EC_A +$ $3T_H + 3T(S)$ $\cong 71M + 3T(S)$	$5EC_m + 3EC_A +$ $3T_H + 1M$ $\cong 168M$	$1EC_p + 3EC_m +$ $2T_H + 5M +$ $1I + 2T(D)$ $\cong 176.9M + 2T(D)$	$3EC_m + 4EC_A +$ $2T_H$ $\cong 111M$
C4	$3EC_m + 3T_H +$ $3T(S)$ $\cong 91M + 3T(S)$	$5EC_m + 2T_H +$ $1M$ $\cong 152M$	$1EC_p + 4EC_m +$ $3T_H + 4M +$ $1I + 2T(D)$ $\cong 208.9M + 2T(D)$	$4EC_m + 2T_H$ $\cong 121M$
C7	$2EC_p + 4EC_m +$ $3EC_A$ $\cong 295M$	$2EC_p + 4EC_m +$ $3EC_A$ $\cong 295M$	$6EC_p + 12EC_m +$ $2T_H + 7M +$ $2I + 1EC_A$ $\cong 849.8M$	N/A
Total	$457M$+$6T(S)$	$615M$	$1235.6M$+$4T(D)$	$232M$

N/A: Not Available

In the deposit phase of Table 6, *C7* of our scheme only needs two pairings operations, four point scalar multiplication operations, and one addition of point operation. Through Table 6, we see that *C7* of our scheme is the lowest than the other schemes.

5. **Conclusions.** In this paper, we have proposed a new e-cash scheme with multiple banks. Our scheme can satisfy all the security properties such as the customer's anonymity protection, non-repudiability of e-cash, and unforgeability of e-cash. Our scheme is also

Table 5. Cost comparisons of the payment protocol among our e-cash scheme and related schemes

	Our	Wang [17]	Chou [10]	Popescu [3]
C1	1472 bits	1472 bits	800 bits	672 bits
C3	$2EC_m + 2T_H$ $\cong 61M$	$3EC_m + 1EC_A + 3T_H$ $\cong 96M$	$1EC_p + 1EC_m + 1T_H + 1T(D)$ $\cong 111M + 1T(D)$	$4EC_m + 1EC_A + 3T_H + 1M$ $\cong 127M$
C5	N/A	$1EC_A + 3T_H$ $\cong 6M$	$1EC_p + 1EC_m + 1T_H + 1T(D)$ $\cong 111M + 1T(D)$	$1EC_A + 3T_H$ $\cong 6M$
C7	$4EC_p + 8EC_m + 4EC_A$ $\cong 580M$	$6EC_p + 10EC_m + 4EC_A + 1T_H$ $\cong 1101M$	$3EC_p + 6EC_m + 1T_H + 6M$ $\cong 427.9M$	$2EC_p + 5EC_m + 3EC_A$ $\cong 325M$
Total	$641M$	$1203M$	$649.9M$+$2T(D)$	$458M$

N/A: Not Available

Table 6. Cost comparisons of the deposit protocol among our e-cash scheme and related schemes

	Our	Wang [17]	Chou [10]	Popescu [3]
C1	1472 bits	1472 bits	928 bits	672 bits
C4	N/A	N/A	$1EC_p + 1EC_m + 1T_H$ $\cong 111M + 1T(D)$	$1EC_A + 3T_H$ $\cong 6M$
C5	N/A	N/A	$1EC_p + 1EC_m + 1T_H$ $\cong 111M + 1T(D)$	N/A
C7	$2EC_p + 4EC_m + 1EC_A$ $\cong 285M$	$4EC_p + 6EC_m + 1EC_A + 1T_H$ $\cong 506M$	$3EC_p + 6EC_m + 1T_H + 6M$ $\cong 427.9M$	$2EC_p + 5EC_m + 3EC_A$ $\cong 325M$
Total	$285M$	$506M$	$649.9M$+$2T(D)$	$331M$

N/A: Not Available

Table 7. Cost comparisons of the tracing protocol among our e-cash scheme and related schemes

	Our	Wang [17]	Chou [10]	Popescu [3]
C1	1472 bits	1472 bits	160 bits	672 bits
C7	$4EC_p + 3EC_m + 1EC_A + 2T_H$ $\cong 416M$	$4EC_p + 2EC_m + 1T_H$ $\cong 381M$	$1T_H + 1T(D) + 1\oplus$ $\cong 1M + 1T(D) + 1\oplus$	$4EC_p + 2EC_m + 1T_H$ $\cong 381M$

more efficient than the other schemes [3, 10, 17]. In future works, we also will investigate additional functions of electronic cash such as the functions of negotiability or divisibility of e-cash.

Acknowledgment. This work was partially supported by the National Science Council of Taiwan under grants NSC 100-2221-E-327-019-MY2, NSC 100-2219-E-110-003, NSC

100-2219-E-110-005, and "Aim for the Top University Plan" of the National Sun Yat-sen University and Ministry of Education, Taiwan.

REFERENCES

[1] A. Shamir, Identity-based cryptosystems and signature schemes, *Proc. of CRYPTO'84, LNCS*, vol.196, pp.47-53, 1985.

[2] C.-I Fan, C.-I Wang and W. Z. Sun, Fast randomization schemes for Chaum blind signatures, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3887-3900, 2009.

[3] C. Popescu and H. Oros, An off-line electronic cash system based on bilinear pairings, *Proc. of the 14th International Workshop on Systems, Signals and Image Processing*, pp.438-440, 2007.

[4] D. Chaum, Blind signature for untraceable payments, *Proc. of Crypto on Advances in Cryptology*, pp.199-203, 1983.

[5] F. Hess, Efficient identity based signature schemes based on pairings, *Proc. of the 9th Annual International Workshop on Selected Areas in Cryptography, LNCS*, vol.2595, pp.310-324, 2003.

[6] F. Zhang and K. Kim, Efficient id-based blind signature and proxy signature from bilinear pairings, *Proc. of the 8th Australasian Conference on Information Security and Privacy, LNCS*, vol.2727, pp.218-219, 2003.

[7] F. Zhang, R. Safavi-Naini and W. Susilo, An efficient signature scheme from bilinear pairings and its applications, *Proc. of the 7th International Workshop on Theory and Practice in Public Key Cryptography, LNCS*, vol.2947, pp.277-290, 2004.

[8] F. Zhang, F. Zhang and Y. Wang, Electronic cash system with multiple banks, *Chinese Journal of Computers*, vol.24, no.5, pp.455-462, 2001.

[9] J.-H. Yang and C.-C. Chang, An efficient fair electronic payment system based upon non-signature authenticated encryption scheme, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3861-3873, 2009.

[10] J. S. Chou, Y. Chen, M. H. Cho and H. M. Sun, A novel id-based electronic cash system from pairings, *Cryptology ePrint Archive, Report 2009/339*, 2009.

[11] J. Zhong and D. He, A new type of group blind signature scheme based on bilinear pairings, *IACR Eprint Archive*, http://eprint.iacr.org/2006/439.pdf, 2006.

[12] M. Stadler, M. M. Piveteau and J. Camenisch, Fair blind signatures, *Proc. of International Conference on the Theory and Application of Cryptographic Techniques, EUROCRYPT'95, LNCS*, vol.921, pp.209-219, 1995.

[13] N. Asokan, P. A. Janson, M. Steiner and M. Waidner, The state of the art in electronic payment systems, *IEEE Computer*, vol.30, pp.28-35, 1997.

[14] S. Canard and A. Gouget, Divisible e-cash systems can be truly anonymous, *Proc. of the 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, EUROCRYPT 2007, LNCS*, vol.4515, pp.482-497, 2007.

[15] S. Park, S. Kim and D. Won, ID-based group signature, *Electronics Letters*, vol.33, no.19, pp.1616-1617, 1997.

[16] S. V. Solms and D. Naccache, On blind signatures and perfect crimes, *Computers and Security*, vol.11, pp.581-583, 1992.

[17] S. Wang, Z. Chen and X. Wang, A new certificateless electronic cash scheme with multiple banks based on group signatures, *Proc. of 2008 International Symposium on Electronic Commerce and Security*, pp.362-366, 2008.

[18] W. Qiu, A fair off-line electronic payment system, *Contributions to Ubiquitous Computing, Studies in Computational Intelligence*, vol.42, pp.177-195, 2007.

[19] W. Qiu, K. Chen and D. Gu, A new offline privacy protecting e-cash system with revokable anonymity, *Proc. of the 5th International Conference on Information Security, ISC 2002, LNCS*, vol.2433, pp.177-190, 2002.

[20] X. Chen, F. Zhang and K. Kim, New id-based group signature from pairings, *Journal of Electronics (China)*, vol.23, pp.892-900, 2006.

[21] Y.-F. Chung and K.-H. Huang, Chameleon signature with conditional open verification, *International Journal of Innovative Computing, Information and Control*, vol.5, no.9, pp.2829-2836, 2009.

[22] Y. Frankel, Y. Tsiounis and M. Yung, Indirect discourse proofs: Achieving efficient fair off-line e-cash, *Proc. of International Conference on the Theory and Applications of Cryptology and Information Security, ASIACRYPT'96, LNCS*, vol.1163, pp.286-300, 1996.

[23] Y.-M. Tseng, T.-Y. Wu and J.-D. Wu, An efficient and provably secure id-based signature scheme with batch verifications, *International Journal of Innovative Computing, Information and Control*, vol.5, no.11(A), pp.3911-3922, 2009.

[24] Z. Li, J. Higgins and M. Clement, Performance of finite field arithmetic in an elliptic curve cryptosystem, *Proc. of the 9th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunications Systems*, pp.249-256, 2001.

[25] Z. W. Tan and Z. J. Liu, A novel identity-based group signature scheme from bilinear maps, http://www.mmrc.iss.ac.cn/pub/mm22.pdf/17.pdf, 2003.