A Transferrable E-cash Payment System

Fuw-Yi Yang\textsuperscript{1}, Su-Hui Chiu\textsuperscript{2} and Chih-Wei Hsu\textsuperscript{3}

Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taiwan\textsuperscript{1,3}
Office of Accounting, Chaoyang University of Technology, Taiwan\textsuperscript{2}
yangfy@cyut.edu.tw\textsuperscript{1}
suhui@cyut.edu.tw\textsuperscript{2}
s10027608@cyut.edu.tw\textsuperscript{3}
Corresponding Author: yangfy@cyut.edu.tw

Abstract

In recent years, as the E-commerce technology develops rapidly, people’s daily life becomes more convenient. The E-cash payment system has been widely used for payment in E-commerce. It allows transactions to be completed on the Internet, and with full history recorded for future review. The traditional E-cash only allows one time usage. Namely, consumer withdraws E-cash from bank and sends it to shop for payment; shop sends E-cash to bank for exchanging money and terminates the life cycle of E-cash. Thus, consumer must withdraw E-cash from bank for each transaction. Therefore, some studies propose transferable E-cash. Transferrable E-cash permits consumers, shops and banks reuse the E-cash before converting into real money. In a sense, the transferrable E-cash is much like real cash. This paper proposes a transferable E-cash payment system. Consequently, the consumer needs not to withdraw E-cash from the bank each time. In addition, the proposed payment system saves energy because the computational cost for transferring E-cash is smaller than for withdrawing E-cash.

Keywords: E-cash; E-commerce; transferable E-cash
1. Introduction

As the network technique develops rapidly, many E-commerce services are extensively used in the Internet, such as E-cash payment system, E-auction system, E-voting and so on. Therefore, people use E-cash (electronic money) for transactions more and more frequently. At present, there are many applications of E-cash payment, such as Easy Card for public transport, or i-cash for buying commodities in convenience supermarkets and so on. These applications have remedied the inconvenience in traditional payment by cash. A transaction can be completed as long as there is a smart card. With the rapid development of smart phones, the E-cash has been applied to mobile devices gradually. However, the security consideration of E-cash influences people's willingness to use it. Therefore, how to improve the security of E-cash is a very important subject. The life cycle of general E-cash payment systems is very short, from withdrawal, payment to deposit. Therefore, the consumer must withdraw money before payment each time that increases the bank's computing cost. Therefore, some literatures about E-cash payment system [1-3] proposed transferable E-cash. The transferable means when the consumer pays E-cash to the shop, the shop can exchange the E-cash for cash, or keep the E-cash for subsequent circulation, so as to reduce the bank's computing cost.

The traditional E-cash payment system consists of three roles, consumer, bank and shop. The agreement is divided into three stages, including withdrawal stage, payment stage and deposit stage. Chaum (1982) first proposed a blind signature-based E-cash payment system. Due to the characteristics of blind signature, the E-cash was characterized by anonymity, verifiability and unforgeability, and these characteristics established the foundation of security feature for subsequent E-cash payment schemes (Chang & Lai, 2003; Fan et al., 2000; ) [5-11]. In 2011, Eslami and Talebi (2011) proposed an off-line E-cash payment system. This system used RSA encryption technique, blind signature technology and ElGamal signature technology to design an E-cash scheme difficult to be traced. Afterwards, Chen et al. [13] found that the Eslami and Talebi scheme could not solve the double-spending problem. Therefore, they proposed a bilinear pairing-based E-cash scheme. This scheme uses trusted third party and bilinear pairing characteristics to make the system meet the due security requirement. However, the bilinear pairing increases the computational cost greatly, so it is inapplicable to the environment of mobile equipments.

At present, there are few literatures about E-cash discussing the transferability, but some literatures have reached this characteristic [1-3]. These schemes use different techniques. Suitable design such that the computational cost for transferring E-cash is smaller than for withdrawing E-cash, the transferable E-cash payment system can reduce the computational cost of overall system. Because when the E-cash is transferable, the consumers don't need to withdraw E-cash from the bank for each payment. The E-cash transferred can be reused; it
will not disappear, unless the consumers deposit this E-cash in the bank, so that the overall computing cost is reduced.

The traditional E-cash is transferred only on one way, the consumers exchange cash for E-cash with the issuing bank, and the E-cash is transferred to the shops during shopping, the shops transfer the E-cash to the issuing bank for cash, and then the E-cash is cancelled. Differing from traditional E-cash, the transferable E-cash allows two-way transfer. Before exchanging for cash, the consumers, shops, or banks can reuse E-cash. As the amount of calculation of transfer is much smaller than the amount of calculation of withdrawing E-cash, the computing cost of the issuing bank is reduced.

Our proposal not only meets the aforesaid advantages, but also has several basic security features required of an ideal electronic cash system [12, 14-17]

1. Anonymity: the E-cash must have anonymity. At the payment stage, the consumer transfers the E-cash to the shop for payment, the shop cannot know the consumer's identity from the E-cash. At the deposit stage, the shop transfers E-cash to the bank to exchange cash. When the bank receives the E-cash, the bank cannot know the consumer's identity from the E-cash.
2. Verifiability: anybody can verify the legality of E-cash.
3. Unforgeability: the E-cash is unique; nobody can forge the same E-cash.
4. Untraceability: in the transaction process of E-cash payment system, nobody knows which E-cash payments of multiple E-cash payments are made by the same consumer.
5. Double-spending prevention: one E-cash can be paid only once, not reused.

2. Discussion about Related Knowledge and Technique

2.1 E-cash trading system

The e-cash payment system is a digital signature-based electronic payment mechanism. First, the consumer exchanges real cash for E-cash, afterwards, the consumer uses E-cash to perform commercial transactions via the Internet or E-cash payment system. Generally, the E-cash is deposited in hardware facilities; it is paid via real equipments during transactions. The E-cash payment system has three major roles, consumer, bank and shop. The system execution flow is divided into three stages, withdrawal stage, payment stage and deposit stage. The complete process is described below:

1. Withdrawal stage: The consumer registers at the bank, and then withdraws money. Finally, the e-cash is deposited in his hardware facility.
2. Payment stage: The consumer uses hardware facility to transfer the e-cash to the shop for payment, and then the shop confirms the legality and double-spending of the e-cash with the bank.
3. Deposit stage: The shop transfers the collected e-cash to the bank for deposit, and then the bank checks whether the e-cash is legal without double-spending. Finally, the cash is deposited in the bank account of the shop.

2.2 Bilinear pairing

In the bilinear pairing cryptosystems [15-20], a mapping function \( e: G_1 \times G_1 \rightarrow G_2 \) plays main role. Function \( e \) maps the elements of cyclic additive group \( G_1 \) into the other cyclic multiplicative group \( G_2 \). Both of the two groups have the same order \( q \). The quantity of \( q \) is a large prime number such that solving the elliptic curve discrete logarithm in \( G_1 \) and \( G_2 \) is infeasible. The bilinear pairing must meet the following characteristics.

- Bilinearity: Let \( P, P_1 \) and \( P_2 \) be arbitrary elements of the group \( G_1 \), \( a \) and \( b \) be arbitrary elements of \( \mathbb{Z}_q \). Then \( e(P_1 + P_2, P) = e(P_1, P)e(P_2, P) \) and \( e(aP, bP) = e(P, P)^{ab} \).
- Non-degeneracy: There are elements \( P \) and \( Q \) in additive group \( G_1 \) meet \( e(P, Q) \neq 1 \), where 1 is the identity element of \( G_2 \).
- Calculability: \( P \) and \( P_1 \) are arbitrary elements of additive group \( G_1 \), there is an effective algorithm to calculate \( e(P, P_1) \).

3. The Transferrable E-cash Payment Schemes

3.1 A conceptual transferrable E-cash payment scheme

A conceptual transferrable E-cash payment scheme has five roles, consumer \( C \), shop \( S \), issuing bank \( B \), consumer bank \( CB \), and shop bank \( SB \). The consumer \( C \) and shop \( S \) can carry out withdrawal, payment and deposit as consumer. The issuing bank \( B \) assumes the circulative E-cash issuing bank. The consumer bank \( CB \) and shop bank \( SB \) assume the verification bank. The process is divided into three stages, withdrawal stage, circulative E-cash transfer stage and deposit stage. The circulative E-cash consists of partial E-cash and certificates of circulative E-cash, which are the signed message of issuing bank \( B \), i.e. \( (R, S, M_{\text{cash}}) \), and the signed message of verification bank \( CB \), i.e. \( (R_1, S_1, M_{\text{cert}}) \), respectively; where \( (R, S) \) is the signature generated by the issuing bank \( B \), the message \( M_{\text{cash}} \) contains random number \( m \), circulative E-cash validity duration \( (Date, Time) \) and so on; \( (R_1, S_1) \) is the signature generated by the verification bank \( CB \), the message \( M_{\text{cert}} \) contains \( (R, S, M_{\text{cash}}) \), verification bank \( CB \), verification timestamp \( (Date_1, Time_1) \) and so on. The process architecture is shown in Figure 1, the information flow i, ii, and iii are described below.

3.1.1 Circulative E-cash withdrawal stage, information flow i

The consumer asks the issuing bank \( B \) for withdrawing "circulative e-cash", and designates the verification bank, the verification bank herein is the consumer bank \( CB \), responding to the consumer demand. The issuing bank \( B \) generates two signed messages, "partial E-cash" and "certificates of partial E-cash". The message of partial e-cash contains issuing bank \( B \), denomination, date of validity and so on. The message of certificates of partial e-cash contains verification bank \( CB \), denomination, date of validity and so on.
order to realize anonymity and untraceability, this scheme uses blind signature technology to implement the two signatures. The consumer is only allowed to use partial e-cash and certificates of partial e-cash to withdraw "certificates of circulative e-cash" from the designated verification bank CB. When the certificates of circulative e-cash are withdrawn, the certificates of partial e-cash are registered in the verification form managed by the verification bank CB, they cannot be reused, equivalent to disappearance.

![Figure 1. Transferable E-cash payment system process architecture](image)

The consumer transfers the partial e-cash and certificates of partial e-cash to the verification bank CB to withdraw certificates of circulative e-cash. The verification bank CB has two verification forms, one is circulative e-cash verification form, as shown in Table 1. The other one is verification form for certificates of partial e-cash, as shown in Table 2. The verification bank CB verifies the partial e-cash and certificates of partial e-cash. The verification algorithm is described below.

<table>
<thead>
<tr>
<th>Partial e-cash</th>
<th>Certificates of circulative e-cash</th>
<th>Timestamp of certificates of circulative e-cash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial e-cash₁</td>
<td>Certificates of circulative e-cash₁</td>
<td>Timestamp of certificates of circulative e-cash₁</td>
</tr>
<tr>
<td>Partial e-cash₂</td>
<td>Certificates of circulative e-cash₂</td>
<td>Timestamp of certificates of circulative e-cash₂</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Partial e-cash</th>
<th>Certificates of partial e-cash</th>
<th>Timestamp of certificates of partial e-cash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partial e-cash₁</td>
<td>Certificates of partial e-cash₁</td>
<td>Timestamp of certificates of partial e-cash₁</td>
</tr>
<tr>
<td>Partial e-cash₂</td>
<td>Certificates of partial e-cash₂</td>
<td>Timestamp of certificates of partial e-cash₂</td>
</tr>
<tr>
<td>:</td>
<td>:</td>
<td>:</td>
</tr>
</tbody>
</table>

1. The verification form for certificates of partial e-cash is checked, if the partial e-cash and certificates of partial e-cash are not stored in the verification form, the verification is successful, on the contrary, it fails.
2. The validity and legality of partial e-cash and certificates of partial e-cash are verified, if they are valid and legal, the verification is successful, otherwise it fails.

3. The certificates of circulative e-cash are generated, the certificate message contains verification bank $CB$, partial e-cash, verification timestamp and so on.

4. The partial e-cash, certificates of partial e-cash and verification timestamp are stored in the circulative e-cash verification form, as shown in Table 1. The partial e-cash, certificates of partial e-cash and verification timestamp are stored in the certificates of partial e-cash verification form, as shown in Table 2.

After the aforesaid algorithmic procedure, the consumer can obtain circulative e-cash, i.e. circulative e-cash = {partial e-cash, certificates of circulative e-cash}.

3.1.2 E-cash transfer stage, information flow ii

Before payment, the consumer transfers the circulative e-cash and verification bank $CB$ to the shop, and then the shop transfers the circulative e-cash and the verification bank $SB$ designated by the shop to the verification bank $CB$. The verification bank $CB$ verifies the signature of issuing bank $B$ and the signature of consumer bank $CB$. The verification algorithm is described below:

1. Confirm the consumer bank $CB$ is the verification bank recorded in the certificates, verify whether the circulative e-cash is valid and legal. If it is valid and legal, the verification is successful, otherwise it is rejected.

2. Check circulative e-cash verification form,
   a. If the circulative e-cash is not stored in the verification form, meaning this circulative e-cash has not yet been transferred, so there is no double-spending.
   b. If the circulative e-cash has been stored in the verification form, it may because the circulative e-cash was already transferred, and now it is transferred back. Therefore, the verification timestamp of circulative e-cash must be checked. If the verification timestamp of circulative e-cash is greater than the verification timestamp stored in the database, there is no double-spending, otherwise there is double-spending.

3. If there is no double-spending, the verification bank $CB$ stores the partial e-cash, certificates of circulative e-cash and verification timestamp in the circulative e-cash verification form.

4. Update the certificate message, including verification bank $SB$ and verification timestamp, calculate the certificates of circulative e-cash.

When the aforesaid algorithm is completed, the certificates of circulative e-cash are transferred to the shop. When the shop receives the certificates of circulative e-cash, the new circulative e-cash = {partial e-cash, certificates of circulative e-cash} is obtained by combination. The shop can choose to exchange the circulative e-cash for real cash or to
deposit it. When the shop exchanges the circulative e-cash for real cash, the circulative e-cash is equivalent to disappearance after deposit algorithm. When the shop deposits the circulative e-cash, it is unnecessary to withdraw money from the bank for subsequent payment, it only needs to transfer the circulative e-cash.

3.1.3 E-cash deposit stage, information flow iii

When the consumer wants to exchange the circulative e-cash for real cash, he must transfer the circulative e-cash to the verification bank CB recorded in the certificates to ask for real cash. When the verification bank CB receives the circulative e-cash, it verifies the legality and double-spending of the circulative e-cash, and then transfers the partial e-cash to the issuing bank B. The issuing bank B checks whether the partial e-cash is in the database, if not, the partial e-cash is stored in the database. Finally, the consumer obtains the real cash. The verification algorithm is described below:

1. Verify whether partial e-cash and certificates of circulative e-cash are valid.
2. Check circulative e-cash verification form
   a. If the circulative e-cash is not stored in the verification form, meaning this circulative e-cash has not yet been transferred, so there is no double-spending.
   b. If the circulative e-cash has been stored in the verification form, it may because the circulative e-cash was already transferred, and now it is transferred back. Therefore, the verification timestamp of circulative e-cash must be checked. If the verification timestamp of circulative e-cash is greater than the verification timestamp stored in the database, there is no double-spending, otherwise there is double-spending.
3. If there is no double-spending, the verification bank CB stores the partial e-cash, certificates of circulative e-cash and verification timestamp in the circulative e-cash verification form, as shown in Table 1.
4. The partial e-cash is transferred to the issuing bank B for verification and stored in the database.

If the aforesaid verification is successful, the consumer can obtain real cash successfully.

3.2 The proposed transferrable E-cash payment system

Our scheme has six roles, consumer C, shop S, issuing bank B, consumer bank CB, shop bank SB and key generation center KGC. There are four major stages, key application stage, withdrawal stage, circulative e-cash transfer stage and deposit stage. The circulative e-cash in this agreement consists of the signed message of issuing bank B \((R, S, M_{\text{cash}})\) and the signed message of consumer bank \(CB\) \((R_1, S_1, M_{\text{cer}1})\). The circulative e-cash is expressed as \(\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cer}1})\}\), where message \(M_{\text{cash}}\) contains random number \(m\), circulative e-cash validity duration \((Date, Time)\) and so on; \((R, S)\) is the signature generated by the issuing bank \(B\); the message \(M_{\text{cer}1}\) contains \((R, S, M_{\text{cash}})\), verification bank \(CB\), verification timestamp \((Date_1, Time_1)\) and so on. \((R_1, S_1)\) is the signature generated by the verification bank \(CB\). First,
the definitions of parameter symbols are introduced, as shown in Table 3. The complete procedure is described below.

**Table 3. Definitions of parameter symbols**

<table>
<thead>
<tr>
<th>Symbol (Subscript)</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_1$</td>
<td>An additive group, its order is $q$</td>
</tr>
<tr>
<td>$G_2$</td>
<td>A multiplicative group, its order is $q$</td>
</tr>
<tr>
<td>$P$</td>
<td>Element of group $G_1$, its order is $q$</td>
</tr>
<tr>
<td>$e$</td>
<td>Bilinear pairing mapping, $e: G_1 \times G_1 \rightarrow G_2$</td>
</tr>
<tr>
<td>$H_1$, $H_2$, $H_3$, $H_4$</td>
<td>One way hash function, $H_1: {0, 1}^* \rightarrow G_1$, $H_2: G_2 \rightarrow {0, 1}^l$, $H_3: {0, 1}^* \rightarrow \mathbb{Z}_q$, $H_4: {0, 1}^* \rightarrow {0, 1}^l$</td>
</tr>
<tr>
<td>$E_k()$, $D_k()$</td>
<td>Symmetric key $K$ is used for encryption and decryption computation</td>
</tr>
<tr>
<td>$s$</td>
<td>Private key of key generation center</td>
</tr>
<tr>
<td>$P_{pub}$</td>
<td>Public key of key generation center, $P_{pub} = sP$</td>
</tr>
<tr>
<td>$ID_U$</td>
<td>The identity of entity $U$, the entity $U$ can be consumer C, shop S, issuing bank B, consumer bank CB and shop bank SB</td>
</tr>
<tr>
<td>$Q_U$, $S_U$</td>
<td>Public and private keys of entity $U$, $Q_U = H_1(ID_U)$, $S_U = sQ_U$</td>
</tr>
</tbody>
</table>

### 3.2.1 Key application stage

The shop bank, consumer bank, issuing bank and consumer must apply to the key generation center for individual public and private keys, as shown in Figure 2. The procedure is described below.

**Figure 2. Key application stage**

1. **Step 1:** The consumer transfers his identity $ID_C$ via secure channel to the key generation center to apply for public and private keys.
2. **Step 2:** The key generation center uses the consumer’s identity $ID_C$ and its private key $s$ to calculate the consumer’s public key $Q_C$ and private key $S_C$. Finally, the public key $Q_C$ and private key $S_C$ are transferred via secure channel to personal cloud.

$$Q_C = H_1(ID_C)$$
$$S_C = sQ_C$$
3.2.2 Withdrawal stage

At the withdrawal stage, we use the identity-based blind signature technology [21]. The consumer must withdraw partial e-cash and certificates of partial e-cash from the issuing bank. The certificate contains the identity of designated verification bank $CB$, the consumer has to withdraw the certificates of circulative e-cash from the verification bank $CB$ in the certificate. Therefore, when the consumer transfers partial e-cash and certificates of partial e-cash to the verification bank $CB$, the bank checks whether the certificates of partial e-cash are in the verification form, and then generates certificates of circulative e-cash and verification timestamp for the consumer. Finally, the consumer can obtain the circulative e-cash. As shown in Procedure i of Chapter 3.1, first, the issuing bank and consumer must create a secret communication channel and negotiate a communication key $K_{B-C} = K_{C-B}$, as shown in Figure 3. The procedure is described below.

![Figure 3. Key agreement between Consumer and Issuing bank](image_url)

**Figure 3. Key agreement between Consumer and Issuing bank**

Step 1: The issuing bank selects random number $k$ from $Z_q^*$, calculates the consumer’s public key $Q_C$, and then calculates the symmetric key $K_{B-C}$. Finally, the message $M_1$ is calculated and transferred to the consumer.

\[
Q_C = H_1(ID_C) \\
K_{B-C} = H_2(e(S_B, kQ_C)) \\
M_1 = kQ_B
\]

Step 2: The consumer uses message $M_1$ and his private key $S_C$ to calculate symmetric key $K_{C-B} = H_2(e(S_C, M_1))$.

When both parties create agreed key $K_{B-C} = K_{C-B}$, the withdrawal stage begins, as shown in Figure 4, detailed as follows.

Step 1: The issuing bank selects random number $r$, $r_{cer}$ from $Z_q$, calculates the message $R'$, $R'_{cer}$, and uses symmetric key $K_{B-C}$ to encrypt the identity of issuing bank $ID_B$ and messages $R'$ and $R'_{cer}$ symmetrically. Finally, the encrypted message $C_1$ is transferred to the consumer.

\[
R' = rP \\
R'_{cer} = r_{cer}P \\
C_1 = E_{K_{B-C}}(ID_B, R', R'_{cer})
\]

Step 2: The consumer uses symmetric key $K_{C-B}$ to decrypt the encrypted message $C_1$ to obtain...
(ID_B, R', R'_cer), and checks whether the identity of issuing bank ID_B is legal, and then selects random numbers a, b, a_1, b_1, m and m' from \( Z'_q \), calculates the messages R, R_cer, h and h_1 and blind messages h' and h_1', and uses symmetric key K_{C:B} to encrypt the consumer’s identity ID_C, blind messages h' and h_1' symmetrically. Finally, the encrypted message C_2 is transferred to the issuing bank for signature.

\[(ID_B, R', R'_cer) = D_{K_{C:B}}(C_1)\]

\[R = aR' + bP\]

\[R_cer = a_1R'_cer + b_1P\]

\[h = H_3(m, R)\]

\[h_1 = H_3(m', ID_C, R_cer)\]

\[h' = h / a \mod q\]

\[h_1' = h_1 / a_1 \mod q\]

\[C_2 = E_{K_{C:B}}(ID_C, h', h_1')\]

\[\text{Figure 4. Stage of withdrawing partial e-cash and certificates of partial e-cash}\]

Step 3: The issuing bank uses symmetric key K_{B:C} to decrypt the encrypted message C_2 to obtain (ID_C, h', h_1'), checks whether the consumer’s identity ID_C is legal, calculates date of validity timestamp t, uses its private key S_B to sign blind messages h' and h_1' and date of validity timestamp t, and then uses symmetric key K_{B:C} to encrypt its
identity $ID_B$, signatures $S'$ and $S'_{cer}$ and validity duration $(Date, Time)$ symmetrically. Finally, the encrypted message $C_3$ is transferred to the consumer.

$$(ID_C, h', h'_1) = D_{K_{B,C}}(C_2)$$
$$t = H_3(Date, Time)$$
$$S' = rQ_B + h'_1tS_B$$
$$S'_{cer} = r_{cer}Q_B + h'_1tS_B$$
$$C_3 = E_{K_{B,C}}(ID_B, S', S'_{cer}, Date, Time)$$

Step 4: The consumer uses symmetric key $K_{C-B}$ to decrypt the encrypted message $C_3$ to obtain $(ID_B, S', S'_{cer}, Date, Time)$, checks whether the identity of issuing bank $ID_B$ is legal, calculates the signatures $S'$ and $S'_{cer}$, and calculates timestamp $t$ to verify whether the signature $(R, S)$ and signature $(S_{cer}, R_{cer})$ are legal. When the verification succeeds, the partial e-cash is $(R, S, M_{cash})$, and the certificates of partial e-cash are $(R_{cer}, S_{cer}, M_{cer})$, where $M_{cash} = (m, Date, Time), M_{cer} = (m', ID_{CB}, Date, Time)$.

$$(ID_B, S', S'_{cer}, Date, Time) = D_{K_{B,C}}(C_3)$$
$$S = aS' + bQ_B$$
$$S_{cer} = a_1S_{cer} + b_1Q_B$$
$$t = H_3(Date, Time)$$
$$e(S, P) \cdot e(Q_B, h_tP_{pub} + R)$$
$$e(S_{cer}, P) \cdot e(Q_B, h_1tP_{pub} + R_{cer})$$

When the consumer receives partial e-cash $(R, S, M_{cash})$ and certificates of partial e-cash $(R_{cer}, S_{cer}, M_{cer})$, he transfers them to the verification bank $CB$ to withdraw certificates of circulative e-cash. First, the consumer must create an anonymous secret communication channel with the verification bank $CB$, and they shall negotiate a communication key $K_{C-CB} = K_{CB-C}$, as shown in Figure 5. The procedure is described below.

![Figure 5. Key agreement between consumer and consumer bank](image)

Step 1: The consumer selects random number $z$ from $Z_q^*$, calculates the public key of consumer bank $Q_{CB}$, and then calculates the symmetric key $K_{C-CB}$. Finally, the message $M_2$ is calculated and transferred to the consumer.

$$Q_{CB} = H_1(ID_{CB})$$
$$K_{C-CB} = H_2(e(S_C, zQ_{CB}))$$
$$M_2 = zQ_C$$
Step 2: The consumer bank uses message $M_2$ and its private key $S_{CB}$ to calculate symmetric key $K_{CB-C} = H_2(e(S_{CB}, M_2))$.

When both parties have created agreed key $K_{CB-C} = K_{CB-C}$, the consumer withdraws certificates of circulative e-cash from the verification bank $CB$, as shown in Figure 6. The procedure is described below.

![Figure 6. Stage of withdrawing certificates of circulative e-cash](image)

Step 1: The consumer uses symmetric key $K_{CB}$ to encrypt partial e-cash $(R, S, M_{cash})$ and certificates of partial e-cash $(R_{cer}, S_{cer}, M_{cer})$ symmetrically. Finally, the encrypted message $C_4$ is transferred to the consumer bank.

$$C_4 = E_{K_{CB-C}}(R, S, M_{cash}, R_{cer}, S_{cer}, M_{cer})$$

Step 2: The consumer bank uses symmetric key $K_{CB-C}$ to decrypt the encrypted message $C_4$ to obtain $(R, S, M_{cash}, R_{cer}, S_{cer}, M_{cer})$, checks the identity of consumer bank $ID_{CB}$, the $ID_{CB}$ is contained in $M_{cer}$, and then calculates the messages $h$ and $h_1$ and date of validity timestamp $t$ to verify whether the partial e-cash $(R, S, M_{cash})$ and certificates
of partial e-cash \((R_{cer}, S_{cer}, M_{cer})\) are legal, and checks whether the certificates of partial e-cash are in the verification form. The verification procedure is shown as verification algorithm in the subsection 3.1.1.

\((R, S, M_{cash}, R_{cer}, S_{cer}, M_{cer}) = D_{K_{CB-C}}(C_4)\)

\(h = H_3(m, R)\)

\(t = H_3(Date, Time)\)

\(h_1 = H_3(m', ID_{CB}, R_{cer})\)

\(e(S, P) \overset{?}{=} e(Q_B, h_1 P_{pub} + R)\)

\(e(S_{cer}, P) \overset{?}{=} e(Q_B, h_1 t P_{pub} + R_{cer})\)

When the verification succeeds, the certificates of partial e-cash shall be registered as invalid, so the partial e-cash \((R, S, M_{cash})\), certificates of partial e-cash \((R_{cer}, S_{cer}, M_{cer})\) and current verification timestamp \((Date_1, Time_1)\) are stored in Table 2. The success of verification means the consumer has paid, and the certificates of circulative e-cash shall be issued. Therefore, the consumer bank selects random number \(y\) from \(Z'_q\), and calculates the messages \(M_{cer1}, R_1\) and \(h_2\), verification timestamp \(t_1\) and signature \(S_1\), and then uses symmetric key \(K_{CB-C}\) to encrypt messages \(M_{cer1}\) and \(R_1\) and signature \(S_1\) symmetrically. The encrypted message \(C_5\) is transferred to the personal cloud. The message \(M_{cer1}\) contains \((R, S, M_{cash})\), verification bank \(CB\)'s identity \(ID_{CB}\), verification timestamp \(Date_1, Time_1\) and so on. Finally, the partial e-cash \((R, S, M_{cash})\), certificates of circulative e-cash \((R_1, S_1, M_{cer1})\) and current verification timestamp \((Date_1, Time_1)\) are stored in Table 1.

\(M_{cer1} = ((R, S, M_{cash}), ID_{CB}, Date_1, Time_1)\)

\(R_1 = yP\)

\(h_2 = H_3(M_{cer1}, R)\)

\(S_1 = yQ_B + h_2 S_{CB}\)

\(C_5 = E_{K_{CB-C}}(R_1, S_1, M_{cer1})\)

Step 3: The consumer uses symmetric key \(K_{C-CB}\) to decrypt the encrypted message \(C_5\) to obtain \((R_1, S_1, M_{cer1})\), and then checks the consumer bank's identity \(ID_{CB}\), calculates message \(h_2\) to verify whether the signature \((R_1, S_1)\) is legal. When the verification succeeds, the circulative e-cash \\{\((R, S, M_{cash}), (R_1, S_1, M_{cer1})\)\} is obtained.

\((R_1, S_1, M_{cer1}) = D_{K_{C-CB}}(C_5)\)

\(h_2 = H_3(M_{cer1}, R)\)

\(e(S_1, P) \overset{?}{=} e(Q_{CB}, h_2 P_{pub} + R_1)\)

### 3.2.3 Circulative e-cash transfer stage

Before payment, the consumer encrypts and transfers the circulative e-cash \\{\((R, S, M_{cash}), (R_1, S_1, M_{cer1})\)\} to the shop, and then the shop transfers the circulative e-cash and the verification bank \(SB\) designated by the shop to the verification bank \(CB\) for verification. As \(M_{cer1}\) contains verification bank \(CB\), the shop can know the verification bank for this circulative e-cash. When the verification succeeds, the verification bank \(CB\) generates the
certificates of circulative e-cash. Finally, the shop obtains the circulative e-cash \( \{(R, S, M_{\text{cash}}), (R_2, S_2, M_{\text{cer2}})\} \). The message \( M_{\text{cer2}} \) contains \( (R, S, M_{\text{cash}}) \), verification bank \( SB \), verification timestamp \( (Date_2, Time_2) \) and so on. \( (R_2, S_2) \) is the signature generated by verification bank \( CB \), the procedure of verification is shown as algorithm ii in Chapter 3.1. First, the consumer and shop must create an anonymous secret communication channel, and negotiate a communication key \( K_{C:S} = K_{S:C} \), as shown in Figure 5. When both parties create agreed key \( K_{C:S} = K_{S:C} \), the circulative e-cash transfer stage begins, as shown in Figure 7, detailed as follows.

\[
C_6 = E_{K_{C:S}}((R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cer1}}))
\]

\[= D_{K_{S:C}}(C_6)
\]

\[
C_7 = E_{K_{C:SB}}((R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cer1}}), ID_{SB})
\]

\[= D_{K_{CB:1}}(C_7)
\]

**Check circulative e-cash**

- \( h = H_3(m, R) \)
- \( t = H_3(Date, Time) \)
- \( h_2 = H_3(M_{\text{cer1}}, R) \)
- \( Q_B = H_3(ID_B) \)
- \( e(S, P) \cdot e(Q_B, hTP_{pub} + R) \)
- \( e(S_1, P) \cdot e(Q_{CB}, hTP_{pub} + R_1) \)
- **Store** \( ((R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cer1}}), Date_2, Time_2) \)
- **Select** \( y_1 \in Z \)
- \( M_{\text{cer2}} = ((R, S, M_{\text{cash}}), ID_{SB}, Date_2, Time_2) \)
- \( R_2 = y_1 P \)
- \( h_3 = H_3(M_{\text{cer2}}, R) \)
- \( S_2 = y_1 Q_{CB} + h_3 S_{CB} \)
- \( C_8 = E_{K_{CB:2}}(R_2, S_2, M_{\text{cer2}}) \)

\[
(R_2, S_2, M_{\text{cer2}}) = D_{K_{S:C:B}}(C_8)
\]

**Check** \( ID_{SB} \)

- \( h = H_3(m, R) \)
- \( t = H_3(Date, Time) \)
- \( h_3 = H_3(M_{\text{cer2}}, R) \)
- \( Q_B = H_3(ID_B) \)
- \( Q_{CB} = H_3(ID_{CB}) \)
- \( e(S, P) \cdot e(Q_B, hTP_{pub} + R) \)
- \( e(S_2, P) \cdot e(Q_{CB}, hTP_{pub} + R_2) \)
- \( M_{\text{cash}} = (m, Date, Time) \)
- Circulative e-cash = \( \{(R, S, M_{\text{cash}}), (R_2, S_2, M_{\text{cer2}})\} \)

Figure 7. Circulative e-cash transfer stage
Step 1: The consumer uses symmetric key $K_{C-S}$ to encrypt circulative e-cash $\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}})\}$ symmetrically. Finally, the encrypted message $C_6$ is transferred to the shop.

$$C_6 = E_{K_{C-S}}((R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}}))$$

Step 2: The shop uses symmetric key $K_{S-C}$ to decrypt the encrypted message $C_6$ to obtain $\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}})\}$. When the shop decrypts $C_6$ and obtains circulative e-cash, the circulative e-cash $\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}})\}$ must be transferred to the designated verification bank $CB$ for verification. The message $M_{\text{cert}}$ contains the designated verification bank $CB$. First, the shop and verification bank $CB$ must create an anonymous secret communication channel, and negotiate a communication key $K_{S-CB} = K_{CB-S}$, as shown in Figure 6. When both parties create agreed key $K_{S-CB} = K_{CB-S}$, the shop uses symmetric key $K_{S-CB}$ to encrypt the circulative e-cash $\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}})\}$ and the verification bank $SB$ designated by the shop. Finally, the encrypted message $C_7$ is transferred to the verification bank $CB$.

$$\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}})\} = D_{K_{S-C}}(C_6)$$

$$C_7 = E_{K_{S-CB}}((R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}}), ID_{SB})$$

Step 3: The consumer bank uses symmetric key $K_{CB-S}$ to decrypt the message $C_7$ to obtain circulative e-cash $\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}})\}$ and verification bank $SB$. First, whether the identity of verification bank $SB$ is legal is checked, and whether the circulative e-cash is in the verification form is checked, as shown in Table 1. Afterwards, the messages $h$ and $h_2$, date of validity $t$ and the issuing bank’s public key $Q_B$ are calculated to verify whether the circulative e-cash is legal. When the aforesaid verification succeeds, the circulative e-cash $\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}})\}$ and current verification timestamp ($Date_2, Time_2$) are stored in Table 1. The message $M_{\text{cash}}$ contains random number $m$, circulative e-cash validity duration ($Date, Time$) and so on. $(R, S)$ is the signature generated by the issuing bank $B$. The message $M_{\text{cert}}$ contains $(R, S, M_{\text{cash}})$, verification bank $CB$, verification timestamp ($Date_1, Time_1$) and so on. $(R_1, S_1)$ is the signature generated by the verification bank $CB$.

$$\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}}), ID_{SB}\} = D_{K_{CB-Q}}(C_7)$$

$$h = H_3(m, R)$$

$$t = H_3(Date, Time)$$

$$h_2 = H_3(M_{\text{cert}}, R)$$

$$Q_B = H_1(ID_B)$$

$$e(S, P) \cdot e(Q_B, hTP_{\text{pub}} + R)$$

$$e(S_1, P) \cdot e(Q_{CB}, h_2P_{\text{pub}} + R_1)$$

When the aforesaid procedure is completed, meaning the consumer’s certificates of circulative e-cash are legal. Therefore, the new certificates of circulative e-cash are generated for the shop. The consumer bank selects random number $y_1$ from $Z_q$, and calculates the
messages $R_2$, $h_3$ and $M_{cer2}$ and signature $S_2$. The message $M_{cer2}$ contains $(R, S, M_{cash})$, verification bank $SB$ and verification timestamp $(Date_2, Time_2)$. Afterwards, the symmetric key $K_{CB}$ is used to encrypt the messages $R_2$ and $M_{cer2}$ and signature $S_2$ symmetrically. Finally, the encrypted message $C_8$ is transferred to the shop.

$$M_{cer2} = ((R, S, M_{cash}), ID_{SB}, Date_2, Time_2)$$

$$R_2 = y_1P$$

$$h_3 = H_3(M_{cer2}, R)$$

$$S_2 = y_1Q_{CB} + h_3S_{CB}$$

$$C_8 = E_{K_{CB}}(R_2, S_2, M_{cer2})$$

Step 4: The shop uses symmetric key $K_{S-CB}$ to decrypt the encrypted message $C_8$ to obtain $(R_2, S_2, M_{cer2})$, and then checks the shop bank’s identity $ID_{SB}$, calculates the messages $h$ and $h_3$, date of validity $t$, issuing bank’s public key $Q_B$ and consumer bank’s public key $Q_{CB}$ to verify whether the shop bank signatures $(R, S)$ and $(R_2, S_2)$ are legal. When the aforesaid verification succeeds, the circulative e-cash $((R, S, M_{cash}), (R_2, S_2, M_{cer2}))$ is obtained. The message $M_{cash}$ contains random number $m$, circulative e-cash validity duration $(Date, Time)$ and so on. $(R, S)$ is the signature generated by the issuing bank. The message $M_{cer2}$ contains $(R, S, M_{cash})$, verification bank $SB$, verification timestamp $(Date_2, Time_2)$ and so on. $(R_2, S_2)$ is the signature generated by the verification bank $CB$.

$$(R_2, S_2, M_{cer2}) = D_{K_{S-CB}}(C_8)$$

$$h = H_3(m, R)$$

$$t = H_3(Date, Time)$$

$$h_3 = H_3(M_{cer2}, R)$$

$$Q_B = H_1(ID_B)$$

$$Q_{CB} = H_1(ID_{CB})$$

$$e(S, P) = e(Q_B, h_3p_{pub} + R_2)$$

$$e(S_2, P) = e(Q_{CB}, h_3p_{pub} + R_2)$$

### 3.2.4 Deposit stage

When the consumer or shop wants to exchange the circulative e-cash for real cash, first, the circulative e-cash is transferred to the designated verification bank for this circulative e-cash to ask for real cash, and then the verification bank verifies whether the circulative e-cash is legal. Afterwards, partial e-cash is transferred to the issuing bank, and the issuing bank checks whether the partial e-cash exists in the database, if not, it is stored in the database. As the procedure iii in Chapter 3.1, first, the consumer and verification bank $CB$ must create a secret communication channel, and negotiate a communication key $K_{C-CB} = K_{CB-C}$, as shown in Figure 3. When both parties create agreed key $K_{C-CB} = K_{CB-C}$, the deposit stage begins, as shown in Figure 8, detailed as follows.
Step 1: The consumer uses symmetric key $K_{C,CB}$ to encrypt the consumer's identity $ID_C$ and the circulative e-cash $\{(R, S, M_{cash}), (R_1, S_1, M_{cert})\}$ symmetrically, and then the encrypted message $C_9$ is transferred to the consumer bank.

$$C_9 = E_{K_{C,CB}}(ID_C, (R, S, M_{cash}), (R_1, S_1, M_{cert}))$$

Step 2: The consumer bank uses symmetric key $K_{CB,C}$ to decrypt the encrypted message $C_9$ to obtain $(ID_C, (R, S, M_{cash}), (R_1, S_1, M_{cert}))$, checks whether the consumer's identity $ID_C$ is legal, and then verifies the circulative e-cash. The procedure of verification is shown as algorithm iii in Chapter 3.1. The messages $h$ and $h_2$, validity duration $t$ and issuing bank's public key $Q_B$ are calculated to verify whether the circulative e-cash is legal,

$$(ID_C, (R, S, M_{cash}), (R_1, S_1, M_{cert})) = D_{K_{CB,C}}(C_9)$$

$$h = H_3(m, R)$$

$$t = H_3(\text{Date}, \text{Time})$$

$$h_2 = H_3(M_{cert}, R)$$

$$Q_B = H_1(ID_B)$$

$$e(S, P) \cdot e(Q_B, h_2P_{pub} + R)$$

$$e(S_1, P) \cdot e(Q_{cb}, h_2P_{pub} + R_1)$$

When the verification succeeds, the circulative e-cash verification form is updated, and the circulative e-cash $\{(R, S, M_{cash}), (R_1, S_1, M_{cert})\}$ and current verification timestamp $(\text{Date}_3, \text{Time}_3)$ are stored. When the aforesaid procedure is completed, the verification bank $CB$ and

Figure 8. Deposit stage
issuing bank must create a secret communication channel, and negotiate a communication key \( K_{CB-B} = K_{B-CB} \), as shown in Figure 5. When both parties create agreed key \( K_{CB-B} = K_{B-CB} \), the consumer bank uses symmetric key \( K_{CB-B} \) to encrypt the consumer's identity \( ID_c \), consumer bank's identity \( ID_{CB} \) and partial e-cash symmetrically, and then transfers the encrypted message \( C_{10} \) to the issuing bank. As the partial e-cash contains the denomination of circulative e-cash, only partial e-cash needs to be transferred to the issuing bank.

\[
C_{10} = E_{K_{CB-B}}(ID_c, ID_{CB}, (R, S, M_{cash}))
\]

Step 3: The issuing bank uses symmetric key \( K_{B-CB} \) to decrypt the encrypted message \( C_7 \) to obtain \( (ID_c, ID_{CB}, (R, S, M_{cash})) \), checks whether the consumer’s identity \( ID_c \) and the consumer bank's identity \( ID_{CB} \) are legal, and then checks whether the partial e-cash exists in the database, if not, the partial e-cash is deposited in the consumer's account.

\[
(ID_c, ID_{CB}, (R, S, M_{cash})) = D_{K_{CB-B}}(C_{10})
\]

4. Analysis of Security and Effectiveness

This section will analyze the security and effectiveness of our scheme.

4.1 Anonymity

At the circulative e-cash transfer stage, the consumer pays to the shop. The shop receives the circulative e-cash, and transfers the circulative e-cash and the verification bank \( SB \) designated by the shop to the consumer bank to verify whether there is double-spending and to calculate certificates of circulative e-cash. Finally, the certificates of circulative e-cash are fed back to the shop. In the aforesaid transaction process, the shop cannot obtain the consumer's identity from the transaction message, and the verification bank \( CB \) and shop cannot know the consumer's identity from the circulative e-cash \( \{(R, S, M_{cash}), (R_1, S_1, M_{cer1})\} \). The message \( M_{cash} \) contains random number \( m \), circulative e-cash validity duration \( (Date, Time) \) and so on. \( (R, S) \) is the signature generated by the issuing bank. The message \( M_{cer1} \) contains \( (R, S, M_{cash}) \), verification bank \( CB \), verification timestamp \( Date_1, Time_1 \) and so on. \( (R_1, S_1) \) is the signature generated by the verification bank \( CB \). The circulative e-cash does not contain the message of consumer's identity, only the legality or double-spending of e-cash can be verified, so our system provides anonymity.

4.2 Unforgeability

Our transferable e-cash payment system uses the blind signature technology of Zhang and Kim [17], based on the assumption of difficulty level of solving elliptic curve discrete logarithm problem, the signature is secure. If somebody wants to forge a circulative e-cash \( \{(R, S, M_{cash}), (R_1, S_1, M_{cer1})\} \), first, the partial e-cash \( (R, S, M_{cash}) \) must be generated with the issuing bank’s private key \( S_B \) and random number \( r \), and the certificates of circulative e-cash \( (R_1, S_1, M_{cer1}) \) must be generated with the verification bank \( CB \)'s private key \( S_{CB} \) and random number \( y \). Therefore, our system provides unforgeability.
4.3 Non-connectivity

In this scheme, nobody can know which payments of multiple circulative e-cash payments are made by the same consumer. It is infeasible to trace which payments are made by the same consumer from multiple circulative e-cash payments. Because although the \((R, S, M_{\text{cash}})\) of our designed circulative e-cash \(\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}})\}\) is changeless, this signature is derived from blind signature technology, the non-connectivity is obtained. The \((R_1, S_1, M_{\text{cert}})\) is generated by general signature technology, but this part of signature changes after each transfer, so it is impossible to find out which payments are made by the same consumer by matching the same value, and the untraceability is obtained.

4.4 Double-spending

At the payment stage, we use on-line payment mechanism. When the consumer transfers payment message \(C_6\) to the shop for verification, the shop transfers \(C_7\) to the verification bank \(CB\). The verification bank \(CB\) receives the message, and decrypts \(C_7\) to obtain circulative e-cash, and verifies the legality and double-spending of the circulative e-cash, and then generates the certificates of circulative e-cash. Finally, the certificates of circulative e-cash are transferred to the shop. Therefore, our scheme can solve the double-spending problem.

4.5 Integrity

In the execution flow of this system, the messages transferred are encrypted by the symmetric key generated by bilinear pairing crypto-technique. For example, at the withdrawal stage, the issuing bank application services and personal cloud must create a secret communication channel, and negotiate a communication key \(K_{B:C} = K_{C:B}\), as shown in Figure 5. The communication keys \(K_{B:C}\) and \(K_{C:B}\) are used to encrypt and decrypt the transmitted messages. Therefore, nobody can modify or delete the encrypted messages in transmission, so our system provides integrity.

4.6 Verifiability

In our scheme, anybody can verify the circulative e-cash \(\{(R, S, M_{\text{cash}}), (R_1, S_1, M_{\text{cert}})\}\), as long as using the key generation center’s public key \(P_{\text{pub}}\), issuing bank’s public key \(Q_B\) and verification bank \(CB\)’s public key \(Q_{CB}\), \(e(S, P) \cdot e(Q_B, hTP_{\text{pub}} + R), e(S_1, P) \cdot e(Q_{CB}, h_2P_{\text{pub}} + R_1)\).

5. Effectiveness Analysis

This section analyzes the computing cost and communication cost of our scheme, as shown in Table 4. The cost is based on using AES encryption technique for symmetric encryption and decryption. A cipher block is 128 bits, the bit length of consumer’s identity \(ID_C\) is assumed to be 112 bits, the bit length of large prime number \(q\) is assumed to be 160 bits, the bit length of one way hash functions \(H_1, H_2, H_3\) and \(H_4\) is assumed to be 256 bits, the bit length of \(G_1\) group is 160 bits, and the bit length of timestamp \((Date, Time)\) is assumed to be 48 bits. At the withdrawal stage, the communication cost is \(512 + 512 + 512 + 1280 + \ldots\)
1024 = 3840 bits. As the consumer and issuing bank as well as the consumer and consumer bank negotiate the symmetric key, the communication cost 160 bit of key agreement must be added. Finally, the communication cost at withdrawal stage is 3840 + 160 + 160 = 4160 bits.

The communication cost of key agreement must be added to the circulative e-cash transfer stage and deposit stage. The communication cost at the circulative e-cash transfer stage is 1536 + 1664 + 1024 + 160 + 160 = 4544 bits. The communication cost at the deposit stage is 1664 + 896 + 160 + 160 = 2880 bits.

Table 4. Cost of computations and communications

<table>
<thead>
<tr>
<th></th>
<th>Computations</th>
<th>Communications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Withdrawal stage</td>
<td>$17 \text{SM} + 14 \text{P} + 2 \text{H}_1 + 4 \text{H}_2 + 9 \text{H}_3 + 1 \text{E}$</td>
<td>4160 bits</td>
</tr>
<tr>
<td>Circulative e-cash transfer stage</td>
<td>$3 \text{SM} + 12 \text{P} + 5 \text{H}_1 + 4 \text{H}_2 + 7 \text{H}_3 + 6 \text{E}$</td>
<td>4544 bits</td>
</tr>
<tr>
<td>Deposit stage</td>
<td>$8 \text{P} + 2 \text{H}_1 + 4 \text{H}_2 + 3 \text{H}_3 + 6 \text{E}$</td>
<td>2880 bits</td>
</tr>
</tbody>
</table>

Note: SM: Scalar multiplication; P: Bilinear pairing operation; H$_1$: Map to points operation; H$_2$, H$_3$: Maps to bit string operation; E: Symmetric encryption and decryption

Afterwards, we compare the computing cost of transferable e-cash with that of nontransferable e-cash. As shown in Table 5, SM represents scalar multiplication, P represents bilinear pairing operation. According to Table 5, at the withdrawal stage, the computing cost of transferable e-cash is higher than that of nontransferable by $12 \text{SM} + 4 \text{P}$. In order to reach transferability, two blind signatures and one normal signature are used at the withdrawal stage, and the verification bank shall verify the signature legality. When the circulative e-cash has been transferred once, the computing cost of transferable e-cash is higher than that of nontransferable e-cash by $11 \text{SM} + 2 \text{P}$. We assume when the circulative e-cash has been transferred $x$ times, the transferable e-cash spends $(19 + 4x) \text{SM} + (4 + 2x) \text{P}$, and the nontransferable e-cash spends $8x\text{SM} + 2x \text{P}$. According to references [22-23], a bilinear pairing operation P is approximately nine times of scalar multiplication SM. Therefore, the total cost of transferable e-cash is $(55 + 22x) \text{SM}$, and the total cost of nontransferable e-cash is $26x \text{SM}$. When $x$ is greater than 13, the computing cost of transferable e-cash is lower than that of nontransferable e-cash. It is not difficult to transfer the transferable e-cash more than 13 times. Therefore, the computing cost of transferable e-cash is not always higher than that of nontransferable e-cash.

Table 5. Comparison of computation between transferrable and nontransferable e-cash

<table>
<thead>
<tr>
<th></th>
<th>Transferable e-cash</th>
<th>Nontransferable e-cash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Withdrawal stage</td>
<td>IB 14 SM</td>
<td>7 SM</td>
</tr>
<tr>
<td></td>
<td>VB 4 P + 5 SM</td>
<td>0</td>
</tr>
<tr>
<td>Circulative e-cash transferred once</td>
<td>IB 14 SM</td>
<td>7 SM</td>
</tr>
<tr>
<td></td>
<td>VB 4 P + 5 SM</td>
<td>2 P + SM</td>
</tr>
<tr>
<td>Circulative e-cash transferred x times</td>
<td>IB 14 SM</td>
<td>7x SM</td>
</tr>
<tr>
<td></td>
<td>VB 4 P + 5 SM + 2x P + 4x SM</td>
<td>2x P + x SM</td>
</tr>
</tbody>
</table>

Note: SM: Scalar multiplication; P: Bilinear pairing operation
IB: denote Issuing bank; VB: denote Verification bank
Table 6 compares the rounds and e-cash bits of transferable e-cash with that of nontransferable e-cash. The transferable e-cash needs five rounds at the withdrawal stage. One circulative e-cash transfer needs five rounds at withdrawal stage and three rounds at circulative e-cash transfer stage, so there are eight rounds. If the circulative e-cash is transferred $x$ times, there are $5 + 3x$ rounds required. The nontransferable e-cash needs three rounds at the withdrawal stage. One circulative e-cash transfer needs three rounds at withdrawal stage and two rounds at circulative e-cash transfer stage, so there are five rounds. If the circulative e-cash is transferred $x$ times, there are $5x$ rounds required. When the circulative e-cash is transferred more than two times, the number of rounds of transferable e-cash is smaller than the number of rounds of nontransferable e-cash. Therefore, when the transferable e-cash meets the aforesaid two requirements, it will be more efficient than nontransferable e-cash. As the circulative e-cash of transferable e-cash has 1536 bits, and the circulative e-cash of nontransferable e-cash has 528 bits, we have to pay more circulative e-cash bits.

<table>
<thead>
<tr>
<th>Number of rounds at withdrawal stage</th>
<th>Transferable e-cash</th>
<th>Nontransferable e-cash</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of rounds of circulative e-cash transferred once</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>Number of rounds of circulative e-cash transferred $x$ times</td>
<td>$5 + 3x$</td>
<td>$5x$</td>
</tr>
<tr>
<td>Circulative e-cash bits</td>
<td>1536</td>
<td>528</td>
</tr>
</tbody>
</table>

6. Conclusion

We have proposed a transferable e-cash payment system. In this system, we use bilinear pairing characteristic to create symmetric key to avoid the communication data being falsified, and use blind signature technology to make our e-cash payment system meet the security requirement. This scheme is transferable, the consumer does not need to withdraw e-cash from the bank for each payment. When the shop receives the circulative e-cash paid by the consumer, the shop can decide whether or not to exchange the circulative e-cash for real cash, if not to, the shop can use this circulative e-cash for next consumption. According to the effectiveness analysis, although the computing cost of this scheme is higher than that of nontransferable e-cash, when the number of transfers of e-cash is larger than our estimated number, the computing cost of transferable e-cash is lower than that of nontransferable e-cash. In addition, more transfers represent lower computing cost. Therefore, our scheme is not only secure, but also spends an appropriate computing cost.

Acknowledgements

This work was partially supported by the Ministry of Science and Technology, Taiwan, MOST 103-2221-E-324-026.
References


