An offline electronic payment system based on an untraceable blind signature scheme

Article *in* KSII Transactions on Internet and Information Systems · January 2017
DOI: 10.3837/tiis.2017.05.018

6 authors, including:


Abdullah Al Rahat
BAUET
1 PUBLICATION **1** CITATION
SEE PROFILE


Kazi Md. Rokibul Alam
Khulna University of Engineering and Technology
33 PUBLICATIONS **73** CITATIONS
SEE PROFILE


R. Tahsin
Khulna University of Engineering and Technology
1 PUBLICATION **1** CITATION
SEE PROFILE


G. G. Md. Nawaz Ali
Nanyang Technological University
54 PUBLICATIONS **137** CITATIONS
SEE PROFILE

An Offline Electronic Payment System Based on an Untraceable Blind Signature Scheme

Md. Abdullah Al Rahat Kutubi,[1] Kazi Md. Rokibul Alam,[1] Rafaf Tahsin,[1]
G. G. Md. Nawaz Ali,[1,2*] Peter Han Joo Chong[3] and Yasuhiko Morimoto[4]
[1] Department of Computer Science and Engineering, Khulna University of Engineering and Technology, Bangladesh
[e-mail: rahatcse2k9@gmail.com, rokib@cse.kuet.ac.bd, rafaftahsin@yahoo.com]
[2] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[e-mail: nawaz.ali@ntu.edu.sg]
[3]Department of Electrical and Electronic Engineering, Auckland University of Technology, New Zealand
[e-mail: peter.chong@aut.ac.nz]
[4]Graduate School of Engineering, Hiroshima University, Higashi-Hiroshima 739-8521, Japan
[e-mail: morimoto@mis.hiroshima-u.ac.jp]
*Corresponding author: G. G. Md. Nawaz Ali

Received September 3, 2016; revised December 9, 2016; accepted January 11, 2017; published May 31, 2017

Abstract

This paper proposes a new offline electronic payment (e-payment) system that satisfies the major security requirements of e-payment, i.e. anonymity, unlinkability, unforgeability, double spending control, conditional traceability, and fraud prevention. The central idea is the use of Hwang et al.'s RSA-based untraceable blind signature (BS), which disables the link between the e-coin and its owner and ensures the anonymity of both the customer and the merchant. It attaches an expiration, a deposit and the transaction dates to each e-coin in order to manage the database of the bank effectively, to correctly calculate the interest on the e-coin and to aid arbitration if a dishonest customer attempts to double-spend the coin. It also ensures the anonymity of the customer as long as the coin is spent legitimately. Only when a fraudulent e-coin transaction is detected can the bank, with the help of the central authority (a trusted entity), determine the identity of the dishonest customer. The system is referred to as offline since the bank does not need to be concurrently involved in transactions between a customer and a merchant. Finally, analyses of the performance of the prototype and the primary security requirements of the proposed system are also presented.

Keywords: Electronic payment system, RSA, Untraceable blind signature, Digital signature

1. Introduction

Unlike conventional payment systems, an electronic payment (e-payment) system uses cryptographic protocols and computer networks to exchange an electronic coin (e-coin) and goods between a customer and a merchant with the assistance of a bank. Here, the customer and the merchant involved in this system do not need to interact with each other physically; this is therefore a convenient system with respect to time and conveyance. On the other hand, an e-payment system is more vulnerable than conventional payment systems, since it is easy to copy an e-coin for misuse or double spending. E-payment systems can be categorized as online and offline. In an online system [2-4], the bank is generally able to check each payment between the customer and the merchant. Thus, if double spending occurs, the bank can terminate the transaction instantly. However, since the bank remains online during each payment, this causes transaction congestion; this also sacrifices the anonymity of the customer. In contrast, an offline system [5-7, 10] allows a customer to pay an e-coin to a merchant without the involvement of a bank or any other central authority (CA). Finally, the merchant can deposit the e-coin obtained from the customer to a bank account at a convenient time. Here, double spending is checked only after connecting with the bank at the time of depositing the coin. Thus, an offline system gives rise to less congestion; it also provides anonymity and privacy to the customer from the bank, the merchant and any other third party during the payment. An offline system therefore has specific advantages.

For offline e-payment systems, a number of blind signature (BS)-based systems have been proposed [5, 7, 8, 10, 11, 15]. However, the majority of the existing BS-based systems are not completely untraceable. Unlinkability is an essential requirement for an e-payment system in order to maintain the anonymity of the customer, and relies on untraceable BS. To ensure anonymity in an offline e-payment system, the e-coin usually contains the identity of its owner in the blind form. When using this information, the merchant, the bank or any attackers are unable to identify the owner of the e-coin; only the CA knows how to link the e-coin and its owner to determine the owner's identity if she is dishonest. Thus, anonymity depends on unlinkability, and these are complementary requirements.

In any offline e-payment system, there is a controversial relationship between untraceability and the detection of double spending. Untraceability disables the link between the e-coin and its owner, and thereby ensures the privacy of the honest customer. On the other hand, the detection and prevention of double spending are also important, so that a dishonest customer cannot double-spend an e-coin and a merchant cannot deposit the same e-coin more than once. An ideal e-payment system must therefore satisfy the following requirements [8, 10].

1. *Anonymity:* The eligibility of the customer is proven without publicly disclosing her identity or information. Thus, no one but the CA can determine the identity of the customer from the contents of the e-coin.
2. *Unlinkability:* No one but the CA can determine the link between the e-coin and its owner. This helps to ensure the anonymity of the customer.
3. *Unforgeability:* Only the Bank can produce legal e-coins.
4. *Double Spending Control:* No e-coin is allowed to be spent more than once.
5. *Conditional Traceability:* No one but the CA should be able to determine the identity of a dishonest customer when double spending occurs.
6. *Theft prevention:* No one but the real owner of a valid e-coin can spend it successfully.
7. *Offline Payments:* When transactions between the customer and the merchant occur, the bank does not need to be concurrently involved.

8. *Date Attachability:* Expiration, transaction and deposit dates should be attached to each e-coin.

In order to satisfy the above requirements, this paper proposes a new offline e-payment system. To achieve anonymity and unlinkability, the proposed system exploits Hwang et al.'s [1] RSA-based untraceable BS, in which only the owner of the e-coin and the CA can determine the link between the e-coin and its owner. It should be mentioned that although a number of BS schemes are available in the domain of cryptography, the majority of these cannot meet the requirement for untraceability. However, Hwang et al.'s RSA-based BS fully satisfies all the requirements of an ideal BS scheme [1]. The bank's signature on each e-coin, which uses an RSA digital signature, ensures authenticated and confidential communication. The proposed system also embeds three types of dates with each e-coin: the expiration date, the transaction date and the deposit date [8, 10]. The expiration date allows the information of expired e-coins to be erased from the bank's database. This prevents the database from becoming too large through the constant increase in e-coin information. The transaction date and the deposit date aid in the detection of dishonest customers, who may try to duplicate the e-coin and perform double spending. These dates also help in the detection of dishonest merchants who may try to deposit the same e-coin more than once. In order to prevent the forgery of an e-coin when the bank issues it to a customer, the bank attaches the customer's public verification key to it.

The rest of this paper is organized as follows. Section 2 summarizes some related works. Section 3 briefly explains the cryptographic tools required to develop the proposed system. Section 4 explains the configuration of the system, and Section 5 describes the individual stages of the system. Sections 6 and 7 discuss the experimental analysis and the security analysis of the system respectively. Finally, Section 8 concludes this paper.

2. Related Works

Extensive research on e-payment systems or schemes has been carried out previously. The scheme proposed in [14] first introduced offline e-payment systems. In order to ensure the anonymity of the customer, this work introduced the concept of BS, although this system was unable to achieve untraceability. These authors used cut-and-choose technology to address the double spending problem. However, cut-and-choose technology is highly inefficient in terms of the data exchange between customers and merchants during the payment process. To overcome the limitations of this approach, several schemes were later proposed which did not rely on cut-and-choose technology to address the double spending problem [16, 17]. They also aimed to meet the requirements of an ideal offline e-payment system. Unfortunately, none of these were globally acceptable. Most of the recently proposed schemes have been shown to be susceptible to serious security flaws, and some are inefficient to implement in practice. Schemes which address the divisibility and transferability of e-coins are generally inefficient, since all of the transaction history of a payment is recorded within an e-coin for the purposes of detection of double spending. The size of the e-coin therefore gradually increases with every transaction, and this limits the maximum number of transactions.

The scheme proposed in [5] failed to achieve anonymity and unlinkability, which are essential requirements of an offline e-payment system [7]. The offline payment system proposed in [10] also has three security flaws: an attack on the detection of double spending, possible forging of the expiration date of a valid e-coin, and cheating on the exchange protocol, as shown by the cryptanalysis presented in [8]. Again, the cryptanalysis presented in [19]

showed that the offline payment scheme proposed in [8] has security flaws in its verifiability, double spending detection and unforgeability. In this system, a malicious spender can withdraw coins without injecting their actual identity, and can forge a valid coin using the homomorphic property of modular operation. Although the Mondex system developed by the National Westminster Bank in the UK successfully ensures the anonymity of the customer, it is vulnerable to illegal fund transfers which cannot be traced [20]. Bitcoin is a globally successful cryptocurrency following a distributed mode of transaction in which a bank has no role. Its primary advantage is that it reduces the high processing costs to the bank; however, its major drawback is that transactions are public, and sensitive transaction data can be traced [18].

3. Cryptographic Building Blocks

This section discusses the required cryptographic tools that are used in developing the proposed e-payment system. These are the RSA digital signature, the RSA cryptosystem, and Hwang et al.'s RSA-based untraceable BS [1]. RSA-based schemes are chosen here since are they are by far the easiest to understand and implement of all the public-key algorithms proposed over the years [12].

3.1 The RSA Digital Signature

1. Any entity X (e.g. the customer, the merchant, the bank or the central authority) chooses two large primes p_x and q_x.
2. X then computes $n_x = p_x.q_x$ and $S_xV_x \equiv 1 \bmod (p_x - 1)(q_x - 1)$.
3. X then retains (S_x, p_x, q_x) as a private signing key and publishes (V_x, n_x) as a public verification key.
4. X chooses a message m for signature verification.
5. X carries out the signing operation using X's private signing key S_x, i.e., $m_s = m^{Sx} \bmod n_x$.
6. Later on, a verification operation is carried out using X's public verification key V_x, i.e. $m = m_s^{Vx} \bmod n_x$.

3.2 The RSA Cryptosystem

1. Any entity X (e.g. the customer, the merchant, the bank or the central authority) chooses two large primes $p_{x'}$ and $q_{x'}$.
2. X the computes $n_{x'} = p_{x'}.q_{x'}$ and $E_{x'}.D_{x'} \equiv 1 \bmod (p_{x'} - 1)\,(q_{x'} - 1)$.
3. X then retains $(D_{x'}, p_{x'}, q_{x'})$ as a private decryption key and publishes $(E_{x'}, n_{x'})$ as a public encryption key.
4. X chooses a message m for encryption-decryption.
5. X carries out the encryption operation using X's public encryption key $E_{x'}$, i.e. $m_d = m^{Ex'} \bmod n_{x'}$.
6. Later on, a decryption operation is carried out by X using X's private decryption key $D_{x'}$, i.e. $m = m_d^{Dx'} \bmod n_{x'}$.

3.3 Hwang et al.'s Blind Signature

Hwang et al.'s blind signature [1] is based on the RSA cryptosystem and consists of five phases, which are described as follows.

1. *Initialization phase*: This phase is the same as the RSA digital signature. The signer keeps (S, p, q) as the secret signing key and publishes (V, n) as the public verification key.
2. *Blinding phase*: The customer or the merchant, Y, requires a message m (i.e. her identity) to be signed by the bank. First, she randomly chooses two distinct integers r_1 and r_2 as the blinding factors. Following this, she randomly chooses two distinct primes a_1 and a_2 such that $a_1 \neq a_2$ and the greatest common divisor (GCD) (a_1, a_2), is 1. Then, Y computes the blinded messages $\alpha_1 = r_1^V . m^{a1} \bmod n$ and $\alpha_2 = r_2^V . m^{a2} \bmod n$, and sends ($\alpha_1$, α_2) to the bank.
3. *Signing phase*: After receiving (α_1, α_2) from Y, the bank randomly chooses two distinct primes b_1 and b_2 such that $b_1 \neq b_2$ and GCD (b_1, b_2) = 1, and signs the blinded message by computing $t_1 = \alpha_1^{b1S} \bmod n$ and $t_2 = \alpha_2^{b2S} \bmod n$. Now ($t_1$, t_2) is a signed, blinded message. The bank sends (t_1, t_2, b_1, b_2) to Y.
4. *Unblinding phase*: After receiving (t_1, t_2, b_1, b_2) from the bank, Y computes a_1b_1 and a_2b_2. Due to the use of four distinct primes (a_1, a_2, b_1, b_2) where GCD (a_1, a_2) = 1 and GCD (b_1, b_2) = 1, GCD (a_1b_1, a_2b_2) is also equal to 1. Since GCD (a_1b_1, a_2b_2) = 1, there must be exactly two integers w and t that satisfy the equation $a_1b_1w + a_2b_2t = 1$. This is known as the extended Euclidean algorithm [12]. The four parameters (a_1, a_2, w, t) are kept secret by Y. Now Y computes $s_1 = t_1.r_1^{-b1} = m^{a1b1S} \bmod n$ and $s_2 = t_2.r_2^{-b2} = m^{a2b2S} \bmod n$. Following this, Y derives the signature S by computing $S = s_1^w.s_2^t \bmod n$ and then publishes (m, S).
5. *Verification phase*: As a result, S is the signature on the message m. Now anyone can verify the legitimacy of the signature by checking whether $S^V = m \bmod n$.

4. Configuration of the Payment System

The entities involved in the proposed system are a trusted central authority (CA), the bank, the customer and the merchant. The roles of these entities are described below.

The Central Authority (CA)

1. The CA is the trusted judge in the system, and conducts the registration of the other entities involved in the system, signing the bank account number of every customer and the merchant. When registering, the bank, the customer and the merchant interact with the CA using their identity; the CA therefore knows their identities. Moreover, the CA manages a database that stores the information of the customer and the merchant involved with the same Bank, and this is shown in **Table 1**.
2. The CA possesses a separate private key and public key (using the RSA cryptosystem/digital signature) for both encryption-decryption and signing-verification. The CA's private signing key is (S_{ca}, p_{cas}, q_{cas}), the public verification key is (V_{ca}, N_{cas}), the public encryption key is (E_{ca}, N_{cae}) and the private decryption key is (D_{ca}, p_{cae}, q_{cae}).
3. In addition, the CA has another secret encryption-decryption key pair, i.e. {(E_s, N_s), (D_s, N_s)}, for encryption and decryption respectively of the identity of the customer and the merchant; these are stored in the database as shown in the first column of **Table 1**.

The Bank

1. The bank manages bank accounts for both the customer and the merchant, and authenticates these with a bank account number signed by the CA. The bank also manages a database that contains the information of the customer and the merchant, as shown in **Table 2**.
2. Prior to signing the e-coin, the bank attaches an expiration date to it. The bank also stores the withdrawn e-coin in the withdrawal database, as shown in **Table 3**.
3. When a merchant asks to deposit an e-coin, the bank verifies the validity of this e-coin and deposits it in the deposit database, as shown in **Table 4**. The bank also detects and prevents double spending, and discloses the identity of any double spender with the help of the CA.
4. Again, the bank renews a customer's unused but outdated e-coin with a new expiration date, and modifies the withdrawal information of the e-coin, as shown in **Table 3**.
5. The bank possesses a separate private key and public key (using the RSA cryptosystem/digital signature) for both encryption-decryption and signing-verification. The bank's private signing key is (S_{bn}, p_{bns}, q_{bns}), the public verification key is (V_{bn}, N_{bns}), the private decryption key is (D_{bn}, p_{bne}, q_{bne}) and the public encryption key is (E_{bn}, N_{bne}). The bank also possesses an identity ID_{bn} and two primes (p_{bn1}, p_{bn2}) such that $p_{bn1} \neq p_{bn2}$ and GCD (p_{bn1}, p_{bn2}) = 1.

The Customer

1. The customer withdraws e-coins from her bank account AN_{cn}. Before withdrawal, she must be authenticated by the bank as a valid bank account holder. The customer also blinds her identity ID_{cn} (using Hwang et al.'s BS) to keep herself anonymous with respect to the other entities involved in the system and to all other third parties except the CA. She therefore uses her blinded identity (α_{cn1}, α_{cn2}) when withdrawing the e-coin from the bank and paying it to the merchant. She also renews her unused but outdated e-coin from the bank with a new expiration date.
2. She possesses a separate private key and public key (using the RSA cryptosystem/digital signature) for both encryption-decryption and for signing-verification. Her private signing key is (S_{cn}, p_{cns}, q_{cns}), her public verification key is (V_{cn}, N_{cns}), her private decryption key is (D_{cn}, p_{cne}, q_{cne}), her public encryption key is (e_{cn}, N_{cne}) and her identity number is ID_{cn}. She randomly chooses two distinct blinding factors (b_{cn1}, b_{cn2}) and two primes (p_{cn1}, p_{cn2}) such that $p_{cn1} \neq p_{cn2}$ and GCD (p_{cn1}, p_{cn2}) = 1, and another two secret integers w and t.

The Merchant

1. The merchant sells goods to the customer in exchange for the e-coin. He also blinds his identity ID_{mn} (using Hwang et al.'s BS) to keep himself anonymous from all other entities except the CA. He verifies the e-coin received from the customer. He also attaches a transaction date D_{tn} and a deposit date D_{dn} to the valid e-coin, and deposits it with the bank.
2. He possesses a separate private key and public key (using the RSA cryptosystem/digital signature) for both encryption-decryption and signing-verification. His private singing key is (S_{mn}, p_{mns}, q_{mns}), his public verification key is (V_{mn}, N_{mns}), his private decryption key is (D_{mn}, p_{mne}, q_{mne}), his public encryption

key is (E_{mn}, N_{mne}) and his identity number is ID_{mn}. He randomly chooses two distinct blinding factors (b_{mn1}, b_{mn2}) and two primes (p_{mn1}, p_{mn2}) such that $p_{mn1} \neq p_{mn2}$ and GCD (p_{mn1}, p_{mn2}) = 1.

Table 1. Database of the central authority

Encrypted identity of customer/merchant	Blinded identity of customer/merchant	The bank's identity
$E(ID_{c1})$	$(\alpha_{c11}, \alpha_{c12})$	ID_{b1}
$E(ID_{m1})$	$(\alpha_{m11}, \alpha_{m12})$	ID_{b1}
$E(ID_{c2})$	$(\alpha_{c21}, \alpha_{c22})$	ID_{b2}
$E(ID_{m2})$	$(\alpha_{m21}, \alpha_{m22})$	ID_{b2}
.	.	.
.	.	.
$E(ID_{cn})$	$(\alpha_{cn1}, \alpha_{cn2})$	ID_{bn}
$E(ID_{mn})$	$(\alpha_{mn1}, \alpha_{mn2})$	ID_{bn}

Table 2. Database of the bank, containing the information of the customer and merchant

Blinded identity of customer/merchant	Account number	Account balance ($)
$(\alpha_{c11}, \alpha_{c12})$	AN_{c1}	2000
$(\alpha_{m11}, \alpha_{m12})$	AN_{m1}	5000
$(\alpha_{c21}, \alpha_{c22})$	AN_{c2}	4000
$(\alpha_{m21}, \alpha_{m22})$	AN_{m2}	3000
.	.	.
.	.	.
$(\alpha_{cn1}, \alpha_{cn2})$	AN_{cn}	2000
$(\alpha_{mn1}, \alpha_{mn2})$	AN_{mn}	1000

Table 3. Withdrawal information database of the bank

E-coin	Customer's blinded identity
$(t_{c11}, t_{c12}, V_{c1}, A_{ec1}, D_{ec1}, A_{c1}, D_{c1}, e\text{-}coin_{c1})$	$(\alpha_{c11}, \alpha_{c12})$
$(t_{c21}, t_{c22}, V_{c2}, A_{ec2}, D_{ec2}, A_{c2}, D_{c2}, e\text{-}coin_{c2})$	$(\alpha_{c21}, \alpha_{c22})$
$(t_{c31}, t_{c32}, V_{c3}, A_{ec3}, D_{new3}, A_{c3}, D_{c3}, e\text{-}coin_{c3})$	$(\alpha_{c31}, \alpha_{c32})$
.	.
$(t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, e\text{-}coin_{cn})$	$(\alpha_{cn1}, \alpha_{cn2})$

Table 4. Deposit database of the bank

E-coin	Deposit Date	Transaction Date
$(D_{d1}, t_{c11}, t_{c12}, V_{c1}, A_{ec1}, D_{ec1}, A_{c1}, D_{c1}, e\text{-}coin_{c1})$	D_{d1}	D_{t1}
$(D_{d2}, t_{c21}, t_{c22}, V_{c2}, A_{ec2}, D_{ec2}, A_{c2}, D_{c2}, e\text{-}coin_{c2})$	D_{d2}	D_{t2}
.	.	.
$(D_{dn}, t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, e\text{-}coin_{cn})$	D_{dn}	D_{tn}

5. Individual Stages of the System

The proposed system consists of six stages: registration, withdrawal, payment, deposit, renewal, and tracing. Each of these stages is described below.

5.1 Registration Stage

The objective of this stage is to register the bank, the customer, and the merchant with the CA, so that the CA can later certify their identities. For this purpose, the bank, the customer and the merchant must provide the necessary information to the CA when they register. This stage is described in detail below and is illustrated in **Fig. 1**.

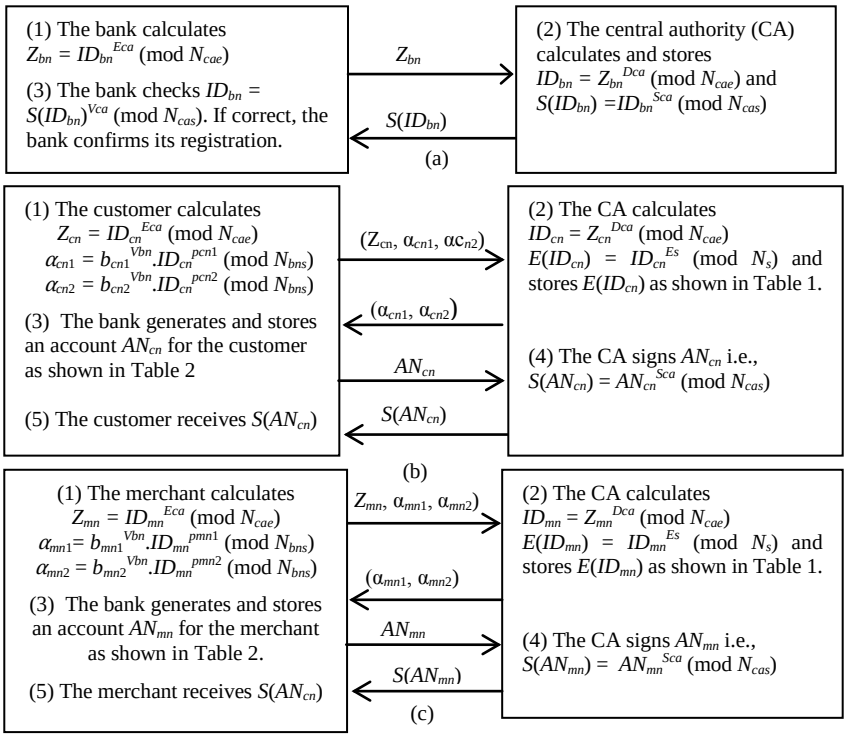

Fig. 1. Data flows at the registration stage for (a) the bank, (b) the customer, and (c) the merchant

Step1: Registration of the bank

1. The bank encrypts its identity ID_{bn}, i.e. the bank calculates $Z_{bn} = ID_{bn}^{Ec} \pmod{N_{cae}}$ where (E_{ca} and N_{cae}) form the CA's public encryption key. The bank then sends Z_{bn} to the CA.
2. The CA decrypts Z_{bn} with its private decryption key D_{ca} to obtain and store ID_{bn} in its database, as shown in **Table 1**.
3. Now the CA signs ID_{bn} using its private signing key (S_{ca}, N_{cas}), i.e. the CA calculates $S(ID_{bn}) = ID_{bn}^{Sac} \pmod{N_{cas}}$ and sends $S(ID_{bn})$ to the bank to confirm its registration.
4. The bank verifies $S(ID_{bn})$ using the CA's public verification key, (V_{ca} and N_{cas}), i.e. the bank checks $ID_{bn} = S(ID_{bn})^{Vca} \pmod{N_{cas}}$. If this is correct, the bank is convinced of its registration.

Step2: Registration of the customer

1. First the customer encrypts her identity ID_{cn}, i.e., the customer calculates $Z_{cn} = ID_{cn}^{Eca} \pmod{N_{cae}}$ and submits this to the CA, where (E_{ca}, N_{cae}) is the CA's public encryption key. Then, using Hwang et al.'s BS, the customer blinds her identity ID_{cn} as $(\alpha_{cn1}, \alpha_{cn2})$, i.e. the customer calculates $\alpha_{cn1} = b_{cn1}^{Vbn} . ID_{cn}^{pcn1} \pmod{N_{bns}}$ and $\alpha_{cn2} = b_{cn2}^{Vbn} . ID_{cn}^{pcn2} \pmod{N_{bns}}$ where (p_{cn1}, p_{cn2}) are two distinct primes and (b_{cn1}, b_{cn2}) are two distinct blinding factors chosen by the customer. Now the customer sends Z_{cn}, α_{cn1}, α_{cn2} to the CA.
2. After receiving this from the customer, the CA decrypts Z_{cn} with its private decryption key D_{ca} to retrieve the customer's identity ID_{cn}. To keep the customer's ID_{cn} secret, the CA again encrypts ID_{cn} with the CA's personal secret encryption key (E_s) and computes $E(ID_{cn}) = ID_{cn}^{Es} \pmod{N_s}$.
3. Following this, the CA stores $(\alpha_{cn1}, \alpha_{cn2})$ which corresponds to the customer's encrypted identity $E(ID_{cn})$, as shown in **Table 1**, and sends $(\alpha_{cn1}, \alpha_{cn2})$ to the bank.
4. The bank generates and stores a bank account number AN_{cn} for the customer, which corresponds to $(\alpha_{cn1}, \alpha_{cn2})$, as shown in **Table 2**, and sends AN_{cn} to the CA.
5. The CA signs AN_{cn}, i.e. the CA calculates $S(AN_{cn}) = AN_{cn}^{Sca} \pmod{N_{cas}}$ and sends it to the customer. Now the customer can withdraw the e-coin from her account while authenticating herself to the bank using $S(AN_{cn})$.

Step 3: Registration of the merchant

1. First the merchant encrypts his identity ID_{mn}, i.e., the merchant calculates $Z_{mn} = ID_{mn}^{Eca} \pmod{N_{cae}}$ and submits this to the CA, where (E_{ca}, N_{cae}) is the CA's public encryption key. Then, using Hwang et al.'s BS, the merchant blinds his identity ID_{mn} as $(\alpha_{mn1}, \alpha_{mn2})$, i.e. the merchant calculates $\alpha_{mn1} = b_{mn1}^{Vbn} . ID_{mn}^{pmn1} \pmod{N_{bns}}$ and $\alpha_{mn2} = b_{mn2}^{Vbn} . ID_{mn}^{pmn2} \pmod{N_{bns}}$, where (p_{mn1}, p_{mn2}) are two distinct primes and (b_{mn1}, b_{mn2}) are two distinct blinding factors chosen by the merchant. Now the merchant sends Z_{mn}, α_{mn1}, α_{mn2} to the CA.
2. After receiving this from the merchant, the CA decrypts Z_{mn} with its private decryption key D_{ca} to retrieve the merchant's identity ID_{mn}. To keep the merchant's ID_{mn} secret, the CA again encrypts ID_{mn} with the CA's secret encryption key (E_s), and computes $E(ID_{mn}) = ID_{mn}^{Es} \pmod{N_s}$.
3. Following this, the CA stores $(\alpha_{mn1}, \alpha_{mn2})$, which corresponds to the merchant's encrypted identity $E(ID_{mn})$, as shown in **Table 1**, and sends $(\alpha_{mn1}, \alpha_{mn2})$ to the bank.
4. The bank generates and stores a bank account number AN_{mn} for the merchant corresponding to $(\alpha_{mn1}, \alpha_{mn2})$, as shown in **Table 2**, and sends AN_{mn} to the CA.
5. The CA signs AN_{mn}, i.e. the CA calculates $S(AN_{mn}) = AN_{mn}^{Sca} \pmod{N_{cas}}$ and sends it to the merchant. Now the merchant can deposit the e-coin to his account while authenticating himself to the Bank using $S(AN_{mn})$.

5.2 Withdrawal Stage

The objective of this stage is to enable a legitimate customer to withdraw the e-coin deposited in her bank account. For this purpose, the customer first needs to be authenticated by the bank by her bank account number, i.e. $S(AN_{cn})$, which has already been signed by the CA. This stage is described in detail below and is illustrated in **Fig. 2**.

1. The customer sends $S(AN_{cn})$ to the bank. The bank then verifies $S(AN_{cn})$ with the CA's public verification key, i.e. the bank generates $AN_{cn} = S(AN_{cn})^{Vca} \pmod{N_{cas}}$ where (V_{ca}, N_{cas}) is the CA's public verification key. If AN_{cn} does not exist in the bank's database, as shown in **Table 2**, the bank terminates the process at this stage.
2. Otherwise, the bank asks the customer to send her identity in its blinded form, i.e. $(\alpha_{cn1}, \alpha_{cn2})$ and stores it in the bank's database, as shown in **Table 2**.
3. Following this, the customer requests an e-coin of amount A_{ecn} from the bank by submitting $(\alpha_{cn1}, \alpha_{cn2})$.
4. If $(\alpha_{cn1}, \alpha_{cn2})$ does not exist in the bank's database, it terminates the process at this stage.
5. Otherwise the bank signs $(\alpha_{cn1}, \alpha_{cn2})$ using Hwang et al.'s BS, i.e. the bank calculates $t_{cn1} = \alpha_{cn1}^{pbn1Sbn} \pmod{N_{bns}}$ and $t_{cn2} = \alpha_{cn2}^{pbn2Sbn} \pmod{N_{bns}}$, where p_{bn1} and p_{bn2} are the two distinct primes and S_{bn} is the bank's private signing key.
6. The bank then calculates the withdrawn e-coin as $sum_{temp} = (t_{cn1} + t_{cn2} + A_{ecn} + D_{ecn} + V_{cn})$, where A_{ecn} is the e-coin amount requested by the customer from the bank, D_{ecn} is the expiration date of the e-coin, and V_{cn} is the public verification key of the customer. Then the Bank signs sum_{temp} with its private signing key S_{bn}, i.e. the bank calculates $ecoin_{cn} = sum_{temp}^{Sbn} \pmod{N_{bns}}$, $A_{cn} = A_{ecn}^{Sbn} \pmod{N_{bns}}$, and $D_{cn} = D_{ecn}^{Sbn} \pmod{N_{bns}}$. A valid e-coin involves an eight-tuple, i.e. $(t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, ecoin_{cn})$.
7. The bank then stores this eight-tuple e-coin $(t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, ecoin_{cn})$ corresponding to the customer's blinded identity $(\alpha_{cn1}, \alpha_{cn2})$ in the bank's withdrawal database, as shown in **Table 3**. The bank also debits A_{ecn} from the customer's bank account and sends this e-coin to the customer along with its distinct primes (p_{bn1}, p_{bn2}).
8. After receiving $(p_{bn1}, p_{bn2}, t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, ecoin_{cn})$ from the bank, the customer unblinds (t_{cn1}, t_{cn2}), i.e. the customer calculates $\{s_{cn1} = t_{cn1}.b_{cn1}^{-pbn1} \pmod{N_{bns}}$, $s_{cn2} = t_{cn2}.b_{cn2}^{-pbn2} \pmod{N_{bns}}\}$ and $S_{cn} = S_{cn1}^{w}.S_{cn2}^{t} \pmod{N_{bns}}$, where w and t are the two secret integers held by the customer. Finally the customer verifies S_{cn} using the bank's public verification key (V_{cn}, N_{bns}), i.e. the customer calculates $ID_{cn} = S_{cn}^{Vbn} \pmod{N_{bns}}$.
9. If the correct ID_{cn} is calcuated, the customer accepts the e-coin.

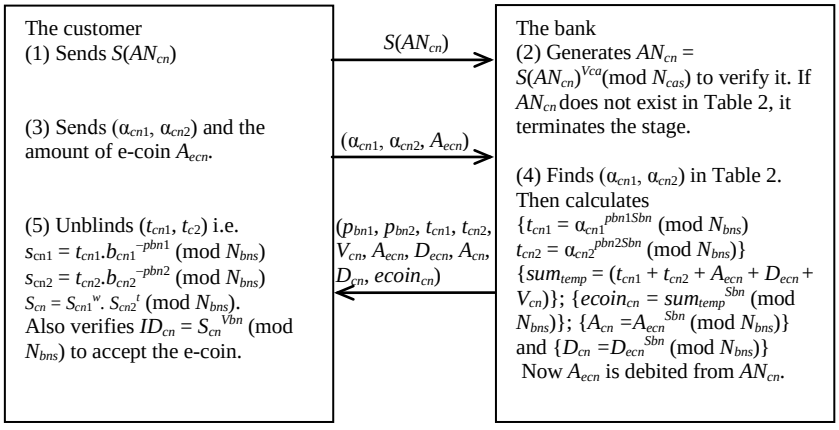

Fig. 2. Data flows at the withdrawal stage

5.3 Payment Stage

The objective of this stage is to pay the merchant the e-coin for the goods purchased by the customer. After withdrawing the e-coin from the bank, the customer can make this payment. Here, the merchant does not need to communicate with the bank while the customer pays him.

Furthermore, it is easy for the merchant to verify the e-coin received from the customer. This stage is described in detail below and is illustrated in **Fig. 3**.

1. The customer firstly sends the eight-tuple e-coin (t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, $ecoin_{cn}$) to the merchant.
2. After receiving this from the customer, the merchant checks whether ($t_{cn1} + t_{cn2} + A_{enc} + D_{ecn} + V_{cn}$) = $ecoin_{cn}^{Vbn}$ (mod N_{bns}), $A_{ecn} = A_{cn}^{Vbn}$ (mod N_{bns}) and $D_{ecn} = D_{cn}^{Vbn}$ (mod N_{bns}). If this is found to be incorrect, the merchant terminates the process at this stage.
3. Otherwise, the merchant decides that it is a valid e-coin. Then he checks the expiration date D_{ecn} to confirm whether the e-coin is outdated. If it is outdated, the merchant terminates the process at this stage.
4. Otherwise, the merchant calculates $temp = ecoin_{cn}^{Vbn}$ (mod N_{bns}), from which he again calculates $C_m = (temp + D_{tn})$, where D_{tn} is the transaction date. Now the merchant sends (D_{tn}, C_m) to the customer to sign both of these individually.
5. After receiving (D_{tn}, C_m) from the merchant, the customer checks D_{tn} to confirm the validity of the transaction's date and time with respect to the current time.
6. If it is valid, the customer individually signs both D_{tn} and C_m with her private signing key, i.e. the customer calculates $S(C_m) = C_m^{Scn}$ (mod N_{cns}) and $S(D_{tn}) = D_{tn}^{Scn}$ (mod N_{cns}), and sends these to the merchant.
7. After receiving $S(C_m)$ and $S(D_{tn})$ from the customer, the merchant decrypts both of them using the customer's public key. If the original C_m and D_{tn} sent to the customer are retrieved, he accepts the e-coin and sells the goods to the customer. The merchant stores $S(C_m)$ and $S(D_{tn})$ until he successfully deposits the e-coin. If double spending is detected, $S(C_m)$ and $S(D_{tn})$ can be used to prove that the transaction of the e-coin was valid. Since the e-payment system is offline, the merchant does not need to deposit the e-coin to the bank immediately; it can be deposited at any convenient time.

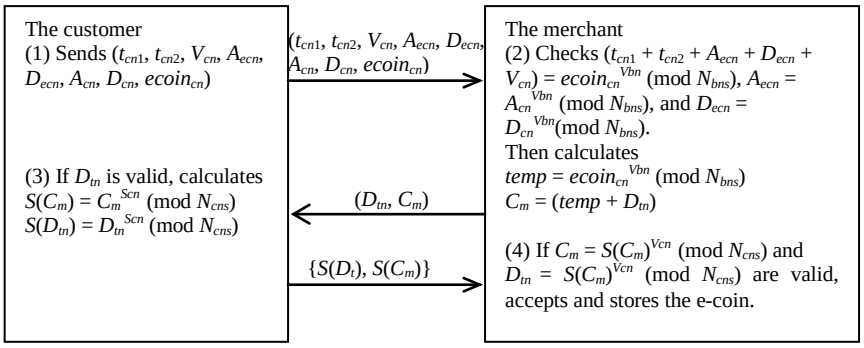

Fig. 3. Data flows at the payment stage

5.4 Deposit Stage

The objective of this stage is to enable the merchant to deposit the e-coin obtained from the customer to his bank account. For this purpose, the merchant attaches the deposit date to the e-coin received from the customer. When the merchant deposits the e-coin, the bank checks for double spending. If the e-coin has already been deposited, the bank simply ignores the deposit of the e-coin, alerting the merchant that he is trying to deposit the same e-coin more than once. Moreover, if the e-coin has already been spent, the bank discloses the identity of the e-coin owner with the help of the CA, as discussed below in Section 5.6 (the tracing stage). This stage is described in detail below and is illustrated in **Fig. 4**.

1. Firstly, for authentication, the merchant submits his bank account number $S(AN_{mn})$, which has already been signed by the CA, to the bank. If authenticated, the merchant attaches the signed transaction date $S(D_{tn})$ and deposit date D_{dn} with the e-coin received from the customer and sends ($S(D_{tn})$, D_{dn}, t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, $ecoin_{cn}$) along with his blinded identity, i.e. (α_{mn1}, α_{mn2}), to the bank.
2. The bank then checks the correctness of the expiration date D_{ecn}, the signed transaction date $S(D_{tn})$, and the deposit date D_{dn}. To verify $S(D_{tn})$, the bank calculates $D_{tn} = S(D_{tn})^{Vcn} \pmod{N_{cns}}$.
3. If the Bank finds that the e-coin exists in the withdrawal database, as shown in **Table 3**, and does not exist in the deposit database, as shown in **Table 4**, it proceeds to Step 4. Otherwise, the bank rejects the e-coin and sends a rejection message to the merchant.
4. Finally, the bank stores the e-coin with the deposit date D_{dn} and transaction date-time D_{tn}, as shown in **Table 4**, in the merchant's bank account.

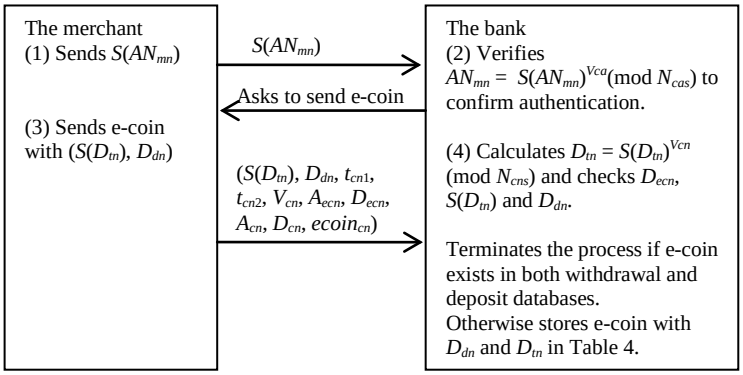

Fig. 4. Data flows at the deposit stage

5.5 Renewal Stage

The objective of this stage is to enable the customer to renew any unused but outdated e-coin. This stage protects the database of the bank from becoming prohibitively large due to the continuous operations of withdrawing and depositing e-coin. When an unused e-coin expires, the customer renews it at the bank with a new expiration date. This stage is described in detail below and is illustrated in **Fig. 5**.

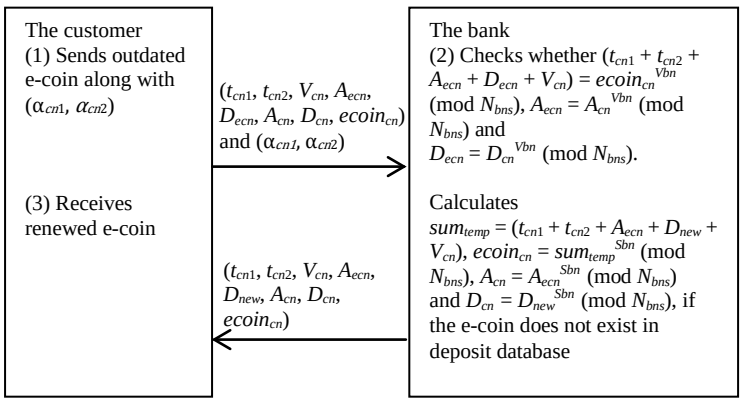

Fig. 5. Data flows at the renewal stage

1. The customer sends an unused but outdated e-coin (t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, $ecoin_{cn}$) along with her blinded identity (α_{cn1}, α_{cn2}).
2. The bank the checks whether $(t_{cn1} + t_{cn2} + A_{ecn} + D_{ecn} + V_{cn}) = ecoin_{cn}^{Vbn} \pmod{N_{bns}}$, $A_{ecn} = A_{cn}^{Vbn} \pmod{N_{bns}}$ and $D_{ecn} = D_{cn}^{Vbn} \pmod{N_{bns}}$. If this is not correct, the Bank terminates the process at this stage.
3. Otherwise, the bank checks whether the e-coin exists in the deposit database, as shown in **Table 4**.
4. If the e-coin does not exist in the deposit database, the bank attaches a new expiration date D_{new} to renew the e-coin and calculates $sum_{temp} = (t_{cn1} + t_{cn2} + A_{ecn} + D_{new} + V_{cn})$. The bank then signs on sum_{temp} with its private signing key S_{bn}, i.e. the bank calculates $ecoin_{cn} = sum_{temp}^{Sbn} \pmod{N_{bns}}$, and $D_{cn} = D_{new}^{Sbn} \pmod{N_{bns}}$. Now ($t_{cn1}$, t_{cn2}, V_{cn}, A_{ecn}, D_{new}, A_{cn}, D_{cn}, $ecoin_{cn}$) is a valid e-coin, and the bank sends it to the customer so that she can use it to pay to the merchant.
5. The bank then replaces the previous eight-tuple e-coin in the bank's withdrawal database (t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, $ecoin_{cn}$) with the renewed one, i.e. (t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{new}, A_{cn}, D_{cn}, $ecoin_{cn}$) corresponding to the customer's blinded identity (α_{cn1}, α_{cn2}), as shown in **Table 3**.

5.6 Tracing Stage

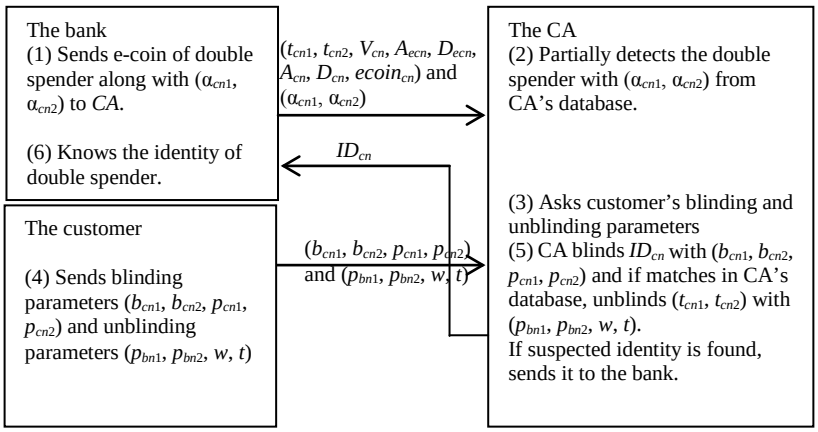

Fig. 6. Data flows at the tracing stage

The objective of this stage is to detect double spending by a customer. When an e-coin is double-spent, this implies the existence of two e-coins with the same record. The bank checks for double spending when it receives these e-coins. If double spending occurs, the bank discloses the identity of the e-coin's owner with the help of the CA. This stage is described in detail below and is illustrated in **Fig. 6**.

1. When the bank receives the e-coin from the merchant for depositing, the bank checks whether the e-coin already exists in the deposit database and/or the withdrawal database. If the e-coin exists in the withdrawal database but not in the deposit database, the bank terminates the process at this stage.
2. If the e-coin exists in both databases, the Bank sends the eight-tuple e-coin (t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, $ecoin_{cn}$) along with the customer's blinded identity, i.e. (α_{cn1}, α_{cn2}) to the CA.

3. The CA then searches (α_{cn1}, α_{cn2}) in the second column of the CA's database, as shown in **Table 1**, to determine the identity of the dishonest customer. The CA retrieves the identity ID_{cn} by decrypting the contents of the first column of **Table 1** using its secret decryption key (D_s, N_s). Thus, the CA partially detects the double spending.
4. In order to detect any dishonesty of the bank towards the customer, the CA asks the suspended customer to send her blinding parameters, i.e. (b_{cn1}, b_{cn2}, p_{cn1}, p_{cn2}), and unblinding parameters, i.e. (p_{bn1}, p_{bn2}, w, t), as described in Sections 4 and 5.2.
5. Using the blinding parameters of the suspended customer, the CA then blinds the customer's identity ID_{cn} and checks whether this matches with her previously stored blinded identity, i.e. (α_{cn1}, α_{cn2}), as shown in **Table 1**. If it matches, the CA confirms that the customer has sent the authentic blinding parameters to the CA.
6. The CA then unblinds (t_{cn1}, t_{cn2}) using (p_{bn1}, p_{bn2}, w, t) as described in Section 5.2. If the unblinded identity matches that of the suspended customer, the CA confirms double spending and sends the customer's identity to the bank.

6. Experimental Analysis

This section evaluates the performance of the proposed system.

6.1 Experimental Setup

To measure the computational time requirements for the withdrawal, payment and deposit stages, a prototype of the proposed system was developed using an Intel Core i3-2.20 GHz processor with 2 GB of RAM running on a Windows 7 operating system. For encryption, GMP [13] with a 1024-bit modulus was used. All computation times exclude communication times. Moreover, the various operations of this system that are not related to cryptography are not considered here.

6.2 Experimental Results

The operational stages of the proposed system consist of authentication, encryption, decryption, blinding, signing, unblinding and verification. The registration stage involves the highest number of operations; that is, 18 exponent operations and one summation. The withdrawal stage consists of 10 exponent operations and one summation; the payment stage consists of eight exponent operations and two summations; the deposit stage consists of five exponent operations and one summation; the renewal stage consists of six exponent operations and two summations; and the tracing stage consists of 12 exponent operations and two summations. **Table 5** shows the required exponent operations and the computation time for the withdrawal, payment and deposit stages.

Table 5. Major cryptographic operations required at each stage of the proposed system and computation time required for the basic three stages

Stage	Major operations	Time requirement (ms)
Registration	18 E	-
Withdrawal	10 E	29
Payment	8 E	24
Deposit	5 E	22
Renewal	6E	-
Tracing	12 E	-

6.3 Comparisons

This section compares the proposed system with the systems in [8-10], in terms of operations of modular multiplication (M) and modular exponentials (E). The operations considered for comparison are the computational cost of withdrawing and paying an e-coin by a customer (c_1); the computational cost to the bank at the withdrawal stage (c_2); the computational cost to the merchant of verifying the e-coin (c_3); the transaction mode (c_4); and the fundamental hard problem of the secure e-payment system (c_5). These are presented in **Table 6** below.

Table 6. Performance comparisons

Case	Eslami and Talebi [10]	Juang [9]	Baseri et al. [8]	Proposed system
c1	5E+9M	3E+6M	6E+8M	7E+11M
c2	1E+2M	1E+2M	2E+1M	6E+6M
c3	6E+3M	2E+2M	6E+2M	3E+3M
c4	Offline	Offline	Offline	Offline
c5	Factoring, DLP	DLP	Factoring, DLP	Factoring

6.4 Comparison of Security Requirements

This section compares the proposed system with those in [8-10] in terms of major security requirements. This comparison is presented in **Table 7** below.

Table 7. Comparison of requirements

Requirements	[10]	[9]	[8]	Proposed system
Anonymity	√	√	√	√
Unforgeability	√	√	×	√
Double spender detection	×	√	×	√
Date attachment	√	×	√	√

7. Security Analysis

The proposed e-payment system satisfies the requirements listed below.

Anonymity and unlinkability: In this system, even if an adversary knows all the information of a particular customer except her secret parameters, i.e. (p_{cn1}, p_{cn2}, w, t), the e-coin cannot be linked to its owner. Here, a valid e-coin is the eight-tuple (t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, $ecoin_{cn}$), where (t_{cn1}, t_{cn2}) is the blinded signed identity of the customer who owns this e-coin. An adversary who does not know p_{cn1}, p_{cn2}, w and t cannot unblind (t_{cn1}, t_{cn2}), and therefore cannot identify the owner of the e-coin. Again, if the customer wants to obtain a further e-coin by choosing secret parameters such as ($pcn1'$, $pcn2'$, w', t'), her second e-coin is then ($tcn1'$, $tcn2'$, Vcn, $Aecn'$, $Decn'$, Acn', Dcn', $ecoincn'$). Thus, no one can detect that these two e-coins originate from the same customer; only the CA can detect this when these e-coins are used dishonestly.

***Unforgeability*:** In order to forge a valid e-coin, an adversary needs to alter the content of (t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}) from the eight-tuple of a valid e-coin, since A_{cn} is the signature of the bank on the amount of e-coin A_{ecn}; D_{cn} is the bank's signature on the expiration date D_{ecn}; and $ecoin_{cn}$ is the signature of the bank on ($t_{cn1} + t_{cn2} + A_{ecn} + D_{ecn} + V_{cn}$). If an adversary alters the content of either A_{cn} or D_{cn} or $ecoin_{cn}$, it cannot re-generate the bank's signatures on these. Therefore an adversary cannot forge the e-coin.

Theft prevention: When a customer pays her e-coin (t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, $ecoin_{cn}$) to a merchant, the merchant verifies the customer as a valid e-coin owner or not through V_{cn}, the customer's public verification key. In order to do this, the merchant sends D_{tn} to the customer to get her signature, i.e. the customer calculates $S(D_{tn}) = D_{tn}^{Scn} \pmod{N_{cns}}$. The merchant then verifies $S(D_{tn})$ using V_{cn}. If the original D_{tn} is found, he accepts the e-coin. Since, the customer's private signing key S_{cn} is secret, no one except the customer can generate the signature on D_{tn}. Thus, the merchant cannot re-spend or double-spend the e-coin accepted from the customer, and only a valid e-coin owner can spend it successfully.

Detection of double spending: No one can alter the customer's public verification key V_{cn} within the eight-tuple of a valid e-coin (t_{cn1}, t_{cn2}, V_{cn}, A_{ecn}, D_{ecn}, A_{cn}, D_{cn}, $ecoin_{cn}$) because it is bundled in $ecoin_{cn}$ and is generated by the signature of the bank on ($t_{cn1} + t_{cn2} + A_{ecn} + D_{ecn} + V_{cn}$). Thus, only the valid e-coin owner can double-spend a valid e-coin, as she has the private signing key corresponding to her public verification key V_{cn}. However, in order to detect double spending, the bank separately maintains withdrawal and deposit databases as shown in **Tables 3** and **4** respectively. If an e-coin already exists in the deposit database, the bank rejects the deposit of that e-coin. If an already deposited e-coin comes from a different merchant, this shows that the customer has paid the same e-coin to two different merchants. The bank then discloses the identity of the e-coin owner, as discussed in Section 5.6 (the tracing stage).

8. Conclusions

In this paper, an offline e-payment system is proposed which fully meets the primary security requirements of e-payment systems. Here, the use of Hwang et al.'s RSA-based untraceable BS ensures the anonymity of any customer, the unforgeability of e-coins and the untraceability of an e-coin to its owner. In this system, the customer is anonymous to all entities (although partially known to the CA) and is prevented from double-spending since her blinded signed identity is attached to the e-coin. With the help of the bank and the attempted spender, only the CA is able to determine the identity of the owner of the e-coin from the contents of the e-coin. Thus the system maintains the anonymity of the customer more securely compared with existing systems. The system ensures the prevention of theft through the attachment of the public verification key to the e-coin. Moreover, because the dates of transactions are attached to the e-coin at various stages in the system, the bank can easily control the size of its database and contribute to arbitration between the double spender and the merchant. Since the system also offers the renewal of e-coins, the customer can renew any unused but outdated e-coins with the bank. A comparison of computational costs at the various stages of the system with several existing systems shows the proposed system is scalable.

References

[1] M. S. Hwang, C. C. Lee and Y. C. Lai, "An Untraceable Blinded Signature Scheme," *IEICE Trans. Fundamentals*, Vol. E86-A, No.7, pp. 1902–1906, 2003.

[2] R. Martínez-Peláez and F. J. Rico-Novella, "New electronic cash model: a script anonym," in *Proc. of the IADIS International Conference on E-Commerce, (e-commerce'06)*, pp. 392–396, 2006.

[3] R. Sai Anand and C. E. Veni Madhavan, "An online, transferable e-cash payment system," in *Proc. of International Conference on Cryptology in India*, pp. 93–103, 2000. Article (CrossRef Link)

[4] D. Chaum, A. Fiat and M. Naor, "Untraceable electronic cash," in *Proc. of Cryptology*, Springer-Verlag New York, pp. 319–327, 1990. Article (CrossRef Link)

[5] C. Fan, V. Huang and Y. Yu, "User efficient recoverable offline e-cash scheme with fast anonymity revoking," *Mathematical and Computer Modeling*, Vol. 58, pp. 227–237, 2013. Article (CrossRef Link)
[6] Y. Chen and J. Chou, "Cryptanalysis on secure untraceable offline electronic cash system," *IACR Cryptology ePrint Archive* 2014: 63, 2014.
[7] Y. Chen and J. Chou, "On the privacy of user efficient recoverable offline e-cash scheme with fast anonymity revoking," *International Journal of Network Security*, 17(6):708711, January 2015.
[8] Y. Baseri, B. Takhaei and J. Mohajeri, "Secure untraceable offline electronic cash system," *Scientia Iranica*, 20(3), pp. 637–646, 2013.
[9] W. Juang, "D-cash: a flexible pre-paid e-cash scheme for date-attachment," *Electronic Commerce Research and Applications*, 6(1), pp. 74–80, 2007. Article (CrossRef Link)
[10] Z. Eslami and M. Talebi, "A new untraceable offline electronic cash system," *Electronic Commerce Research and Applications*, 10(1), pp. 59–66, 2011. Article (CrossRef Link)
[11] H. Oros and C. Popescu "A Secure and Efficient Offline Electronic Payment System for Wireless Networks," *International Journal of Computers, Communications & Control*, Vol. V, No. 4, pp. 551–557, 2010. Article (CrossRef Link)
[12] B. Schneier, "Applied Cryptography," *2nd Edition. John Wiley*, 2008.
[13] T. Granlund, GNU Multiple Precision Arithmetic Library (GMP), Software available at http://gmplib.org/ July 2015.
[14] D. Chaum, "Blind signatures for untraceable payments," in *Proc. of Advances in Cryptology (CRYPTO'82)*, pp. 199–203, Santa Barbara, California, USA, 1982.
[15] Z. Tan, "An offline electronic cash scheme based on proxy blind signature," *The Computer Journal*, Vol. 54, No. 4, pp. 505–512, 2011. Article (CrossRef Link)
[16] S. Brands, "Untraceable offline cash in wallet with observers," in *Proc. of Cryptology-CRYPTO'93*, Springer, pp. 302–318, 1994. Article (CrossRef Link)
[17] F. Stalder, "Failures and successes: Notes on the development of electronic cash," *The Information Society: An International Journal*, Vol. 18, no. 3, pp. 209–219, 2002. Article (CrossRef Link)
[18] I. Miers, C. Garman, M. Green and A. D. Rubin, "Zerocoin: Anonymous Distributed E-Cash from Bitcoin," *IEEE Symposium on Security and Privacy*, pp. 397–411, 2013. Article (CrossRef Link)
[19] F. Wang, C. Chang, and C. Lin, "Security Analysis on Secure Untraceable Offline Electronic Cash System," *International Journal of Network Security*, Vol. 18, No. 3, pp. 454–458, 2016.
[20] F. Stalder, "Failures and successes: Notes on the development of electronic cash," *The Information Society: An International Journal*, Vol. 18, No. 3, pp. 209–219, 2002. Article (CrossRef Link)

Md. Abdullah Al Rahat Kutubi is currently a student of M. Sc. program in the Dept. of Applied Information Technology of Kookmin University, Korea. He has completed his B.Sc. degree in Computer Science and Engineering from Khulna University of Engineering & Technology, Bangladesh.

Kazi Md. Rokibul Alam is currently a professor in the Dept. of Computer Science and Engineering of Khulna University of Engineering & Technology, Bangladesh. He received Dr. (Eng.) degree in System Design Engineering from University of Fukui, Japan, and M.Sc. and B. Sc. degrees both in Computer Science and Engineering from Bangladesh University of Engineering & Technology and Khulna University, Bangladesh in 2010, 2004 and 1999 respectively. His research interests include applied cryptography, information security and machine learning.

Rafaf Tahsin has completed his B.Sc. degree in Computer Science and Engineering from Khulna University of Engineering & Technology, Bangladesh. Now he works as a software engineer at NextGen media lab.

G. G. Md. Nawaz Ali G. Md. Nawaz Ali is currently a Postdoctoral research fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore. He received his PhD in the Department of Computer Science, City University of Hong Kong in 2013. He received his B.Sc. degree in Computer Science and Engineering from Khulna University of Engineering & Technology, Bangladesh in 2006. He is a member of IEEE and IEEE VTS. His current research interests include wireless broadcasting, mobile computing, network coding, and ad hoc networking with a focus on vehicular ad hoc networking.

Peter Han Joo Chong is currently a Professor and Head of Department of Electrical and Electronic Engineering at Auckland University of Technology, Auckland, New Zealand. He received the B.Eng. (with distinction) in Electrical Engineering from the Technical University of Nova Scotia, Halifax, NS, Canada, in 1993, and the M.A.Sc. and Ph.D. degrees in Electrical and Computer Engineering from the University of British Columbia, Vancouver, BC, Canada, in 1996 and 2000, respectively. He has visited Tohoku University, Japan, as a Visiting Scientist in 2010 and Chinese University of Hong Kong (CUHK), Hong Kong, between 2011 and 2012. He is currently an Adjunct Faculty at the Department of Information Engineering, CUHK. He was previously an Associate Professor (tenured) from 2009 to 2016 and Assistant Professor from 2002 to 2009 in the School of Electrical and Electronic Engineering at Nanyang Technological University (NTU), Singapore. Between 2011 and 2013, he was an Assistant Head of Division of Communication Engineering. Between 2013 and 2016, he was a Director of Infinitus, Centre for Infocomm Technology. He was the recipient of 'EEE Teaching Excellence Award' and 'Nanyang Award Excellence in Teaching' in 2010, and 'Nanyang Education Award (College)' in 2015. In 2015, he became a Fellow of the Teaching Excellence Academy in NTU. From February 2001 to May 2002, he was a Research Engineer at Nokia Research Center, Helsinki, Finland. Between July 2000 and January 2001, he worked in the Advanced Networks Division at Agilent Technologies Canada Inc., Vancouver, BC, Canada. He is the Co-Founder of P2 Wireless Technology based in Hong Kong. He is an Editorial Board Member of Security and Communication Networks, Wireless Sensor Network, and an Editor of Far East Journal of Electronics and Communications, and KSII Transactions on Internet and Information Systems. His research interests are in the areas of mobile communications systems including radio resource management, multiple access, MANETs/VANETs, multihop cellular networks and Internet of Things/Vehicles.

Yasuhiko Morimoto is an associate professor at Hiroshima University. He received his B.E., M.E. and Ph.D degrees from Hiroshima University in 1989, 1991 and 2002 respectively. From 1991 to 2002, he had been with IBM Tokyo Research Laboratory where he worked for data mining project and multimedia database project. Since 2002, he has been with Hiroshima University. His current research interest includes data mining, machine learning, geographic information system and privacy preserving information retrieval.