Improving An on-line Electronic Check System with Mutual Authentication

Chin-Ling Chen[#1], Cheng-Hsiung Wu[#2], Wei-Chech Lin[#3]

[#1,2,3]*Department of Computer Science and Information Engineering*

Chaoyang University of Technology

168 Jifong E.Rd., Wufong Township Taichung Country, 41349, Taiwan (*R.O.C*)

[1]clc@mail.cyut.edu.tw

[2]s9827602@cyut.edu.tw

[3]weichech@gmail.com

Abstract— **In recent years, electronic check (e-check) has been more and more popular on the electronic commerce application. For this reason, there were many scholars have proposed the security issues for related e-check. Chang et al. present their views on e-check to improve the past scheme. Chang et al.'s scheme achieves the security of system and provides mutual authentication between payer and payee. However, there still has non-anonymity, time synchronous and too large computing issues in their scheme. We therefore propose an improving scheme to avoid these defects.**

Keywords— **electronic check, electronic commerce, mutual authentication, non-anonymity, time synchronous**

1. INTRODUCTION

In 1988, Chaum [1] was first proposed an idea of electronic check, the idea has a strong influence on e-commerce. It is convenient for user to pay on electronic payment. Today, the electronic payment system can be categorized into two types: electronic check (e-check) and electronic cash (e-cash). The electronic payment system has been become general payment mechanism, but it exists non-anonymity, large computing and non-repudiation etc. issues have become the attacking goal, so related security of electronic payment has become an important issue.

In recent year, more scholars research for electronic check and proposed many views of security. In 1988, Chaum proposed an on-line electronic check system, but the system has to build large computations. In 2001, Hsien et al. [5] proposed an electronic traveler's check scheme and divided it into three phases: withdrawal protocol, payment protocol and deposit protocol. Hsien et al. used ElGamal digital signature to achieve non-repudiation of electronic traveler's check. Due to many exponentiations exist in their scheme; it is a heavy load problem. Another important issue is anonymity. In 2007, Liaw et al.'s scheme [6] said that the features of traditional electronic check are following: anonymity, transferability and convenience. It is important in electronic check issue. Because it can protect an identity of payer, if the data is stolen, it doesn't worry that attacker knows the information of data. In 1989, Chaum proposed an off-line electronic check system [2] to solve the problem of large computing. But Chen [3] pointed that Chaum's scheme is not really an efficient system, so in 2005, Chen proposed an efficient on-line electronic check protocol. In Chen's scheme, it used less hashing operation to improve the efficiency of the system. After transaction, the face value and payee's identity have to add on electronic check to verify the check whether correct. But in Chen's scheme, it cannot achieve the security requirement in the efficient electronic check system. For this reason, Chang et al. [4] proposed a secure electronic check system has to achieve the following requirements: uniqueness, robustness, mutual authentication, non-repudiation. Chang et al. improved Chen's scheme and proposed their views. The proposed scheme divided into two phases: registration phase and payment phase, they used blind signature and RSA digital signature to achieve above four requirements. But in their scheme, the electronic check system used the user real identity to conduct transaction with payee and used hash chain to make a large computation. It can't achieve anonymity and efficiency. We therefore improve Chang et al.'s scheme.

In Chang et al.'s protocol, it could achieve above four requirements, but does not achieve anonymity. In addition we find that the computation is very large in Chang et al.'s scheme. Chang et al. used a large face value to execute an on-line hash chain exponentiation. It is not suitable to real time system. In this paper, we will improve above problems and analyse to show that our scheme can achieve more secure.

The rest of this article is organized as follows. In section 2, we first review Chang et al.'s scheme and make a security analysis. Next we propose our improved scheme in section 3 and analyse the security of our scheme in 4 section. Finally, we make a conclusion in section 5.

2. REVIEW OF CHANG ET AL.'S SCHEME

In Chang et al.'s scheme, they used blind signature, one-way hash function, and RSA digital signature to propose an electronic check mechanism. Now, we review their scheme. The Chang et al.'s notation is shown in table 1.

TABLE 1
THE NOTATION OF CHANG ET AL.'S SCHEME

ID_X	The identifier of *X*.
pk_X	The public key of *X*.
sk_X	The private key of *X*.
$H(\cdot)$	A secure hash function.
$\|\|$	The string concatenation operator.
j	The times of check book can use.
i	The time of check book has been used
w	The maximum face value of e-check.
$E_X(\cdot)$	A symmetric encryption with the secret key *X*.
ck	The secret key shared between payer and bank.
T	The current timestamp.
a	The face value.

2.1. The Registration Phase

First, customers have to register account in the bank, and the bank has to choose randomly two different large prime numbers p and q, then compute $n = p \cdot q$. After that, bank uses RSA algorithm to generate the public key (pk_{bank}, n) and private key (sk_{bank}, p, q). Payer sets w into the maximum face value of e-check. The process is described as follows:

Step 1: The payer randomly chooses a secret integer r.

Step 2: The payer computes:
$m = H(ID_{payer} \| H^w(r))$ and $\alpha = H(m)$.

Step 3: Then the payer sends a registration request including (ID_{payer}, α) to the bank.

Step 4: After receiving the request, the bank verifies the payer's identity ID_{payer} and computes:

$$\alpha' = H^j(m)$$

$$s = (\alpha')^{sk_{bank}} \bmod n,$$

where j is the number of times that the payer can use the check book. Then, the bank sends (s, j) to the payer.

Step 5: Upon receiving the message from the bank, the payer verifies the integrity of the message by checking whether $s^{pk_{bank}} = H^j(m)$ or not. If it holds, the payer stores the e-check book (m, s, j).

2.2. The Paying Phase

When the payer decides to use the e-check buy some goods. Setting a is the face value for the transaction where $a \le w$.

Step 1: The payer randomly chooses two integers R and b, and computes:

$$k = R^{pk_{payee}} b \bmod n'$$

where (pk_{payee}, n') is the public key of the payee. The payer subsequently sends k to the payee.

Step 2: After receiving k, the payee computes:

$$k' = k^{sk_{payee}} = R \cdot b^{sk_{payee}} \bmod n'$$

where sk_{payee} is the private key of the payee. And then sends k' to payer.

Step 3: When the payer receives k', then computes:

$$M = k' \cdot R^{-1} = b^{sk_{payee}} \bmod n',$$

$$C_1 = H^{w-a}(r) \oplus M,$$

$$C_2 = E_{ck}(i \| T).$$

The usage of timestamp needs a synchronization mechanism between the payer and the bank.

Step 4: The payer then checks if $b = M^{pk_{payee}} \bmod n'$ holds or not. If it is valid, the payer terminates the transaction; otherwise, the payer sends the message $(ID_{payer}, ID_{bank}, a, b, j, s, C_1, C_2, T)$ to the payee.

Step 5: According to the received message, the payee can verify the integrity of the message by checking whether

$$s^{PK_{bank}} = H^j(H(ID_{bank} \| H^a(C_1 \oplus b^{sk_{payee}})))$$

holds or not.

Step 6: After receiving the message from the payee, the bank checks if the e-check in the database or not.

Step 7: The bank checks whether a is larger than the payer's deposit in the bank or not.

Step 8: The bank records the current timestamp T' when the message is received.

Step 9: The bank checks whether $(T' - T)$ is within the valid time interval ΔT .

Step 10: The bank verifies the e-check by computing

$$s^{pk_{bank}} = H^i(H^{j-i}(m))$$

If it holds, the bank refreshes the payer's e-check book and sends "Accept" message to the payee; otherwise, the bank sends "Reject" message to the payee.

Step 11: If the received message is accepted, the payee sends "Accept" message to the payer.

3. CHANG ET AL.'S SCHEME SECURITY ANALYSIS

In this section, we analyse the security of Chang et al.'s scheme, and indicate the faults of their scheme.

3.1. Non-anonymity issue

In Chang et al.'s protocol, in the paying phase, the payer has to use his or her real identity to buy something and send the message $(ID_{payer}, ID_{bank},$ $, a, b, j, s, C_1, C_2, T)$ to payee. In this message, the ID_{payer} is the payer's real identity, it is possible to be intercepted by attacker and the payer's identity can not be protected. So, Chang et al.'s scheme suffers from anonymous.

3.2. Large computing issue

In the registration phase of Chang et al.'s protocol

$$m = H(ID_{payer} \| H^w(r))$$

For above expression, the authors denote w is the maximum face value of electronic check, it must waste a lot of time to perform the exponentiation during the transactions, it's not acceptable.

3.3. Time synchronous issue

When the payer buys something in paying phase, it generates a timestamp T_{payer} . If payer's time is not the same as bank, it exists time synchronous issue. That is the payer's data is unable to update real-time with the bank then occurs time synchronous issue.

4. OUR IMPROVED SCHEME

In this section, we propose our views to improve the Chang et al.'s scheme and enhance security between payer and bank. The notation of our scheme is shown in Table 2.

TABLE 2
THE NOTATION OF OUR SCHEME

ID_X	The identity of *X*.
PK_X / SK_X	The public / private key of *X*.
R	The random number.
$h(\cdot)$	One-way hash function.
$\|\|$	The string concatenation operator.
$\oplus$	XOR operation.
j	The number of times that check book can use.
w	The maximum face value of e-check.
$E_X(\cdot)/D_X(\cdot)$	Symmetric encryption / decryption with the secret key *X*.
CK	The session key shared between payer and bank.
N	The nonce.
a	The face value.
$\dashrightarrow$	The secure channel.
$\longrightarrow$	The insecure channel.

The scenario is shown in Fig 1.

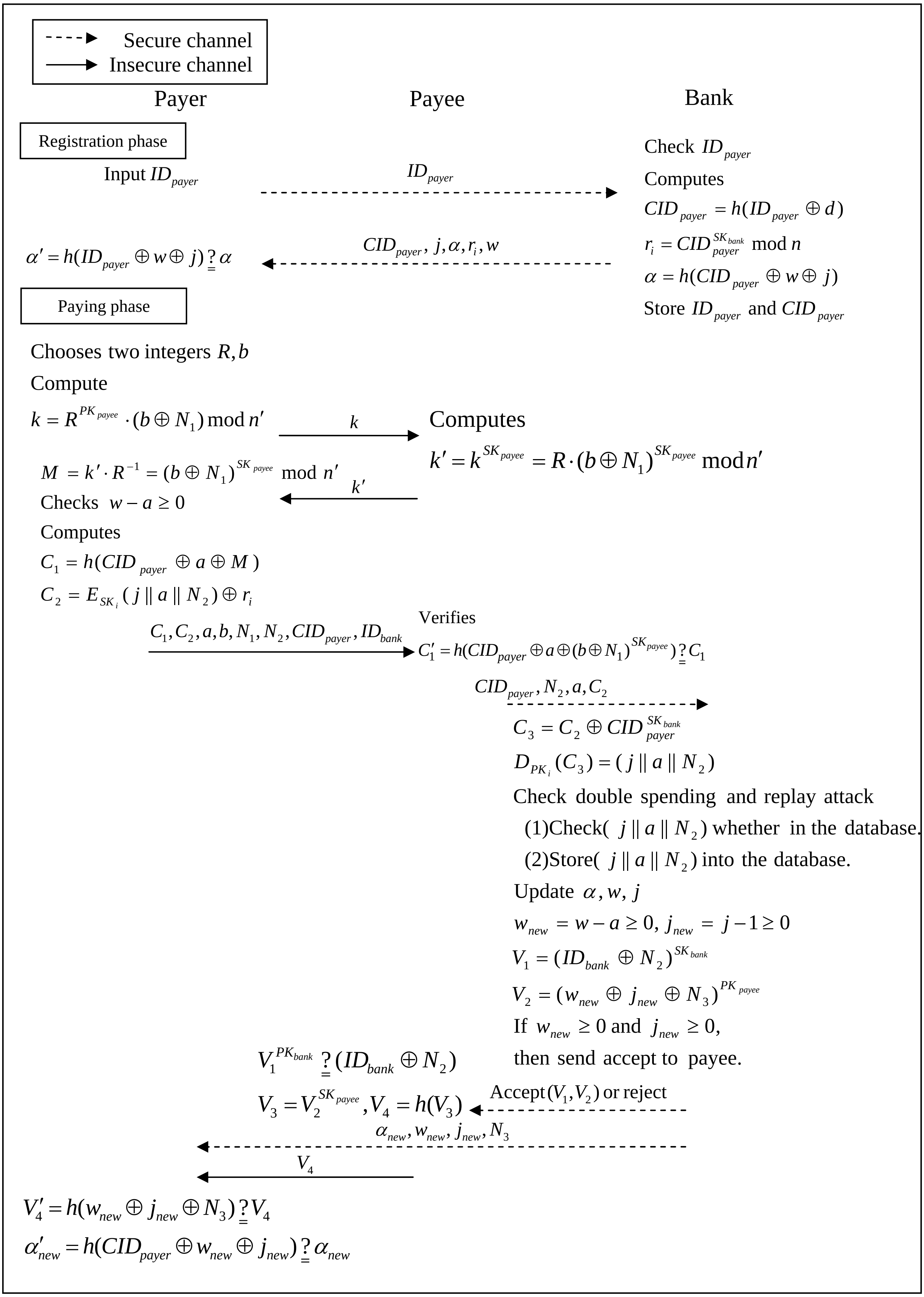
Secure channel
Insecure channel
Payer
Payee
Bank
Registration phase
Input ID_{payer}
ID_{payer}
Check ID_{payer}
Computes
$CID_{payer} = h(ID_{payer} \oplus d)$
$r_i = CID_{payer}^{SK_{bank}} \bmod n$
$\alpha = h(CID_{payer} \oplus w \oplus j)$
Store ID_{payer} and CID_{payer}
$CID_{payer}, j, \alpha, r_i, w$
$\alpha' = h(ID_{payer} \oplus w \oplus j) \overset{?}{=} \alpha$
Paying phase
Chooses two integers R, b
Compute
$k = R^{PK_{payee}} \cdot (b \oplus N_1) \bmod n'$
k
Computes
$k' = k^{SK_{payee}} = R \cdot (b \oplus N_1)^{SK_{payee}} \bmod n'$
$M = k' \cdot R^{-1} = (b \oplus N_1)^{SK_{payee}} \bmod n'$
k'
Checks $w - a \geq 0$
Computes
$C_1 = h(CID_{payer} \oplus a \oplus M)$
$C_2 = E_{SK_i}(j \| a \| N_2) \oplus r_i$
$C_1, C_2, a, b, N_1, N_2, CID_{payer}, ID_{bank}$
Verifies
$C_1' = h(CID_{payer} \oplus a \oplus (b \oplus N_1)^{SK_{payee}}) \overset{?}{=} C_1$
CID_{payer}, N_2, a, C_2
$C_3 = C_2 \oplus CID_{payer}^{SK_{bank}}$
$D_{PK_i}(C_3) = (j \| a \| N_2)$
Check double spending and replay attack
(1)Check($j \| a \| N_2$) whether in the database.
(2)Store($j \| a \| N_2$) into the database.
Update α, w, j
$w_{new} = w - a \geq 0,\ j_{new} = j - 1 \geq 0$
$V_1 = (ID_{bank} \oplus N_2)^{SK_{bank}}$
$V_2 = (w_{new} \oplus j_{new} \oplus N_3)^{PK_{payee}}$
If $w_{new} \geq 0$ and $j_{new} \geq 0$,
then send accept to payee.
$V_1^{PK_{bank}} \overset{?}{=} (ID_{bank} \oplus N_2)$
$V_3 = V_2^{SK_{payee}}, V_4 = h(V_3)$
Accept(V_1, V_2) or reject
$\alpha_{new}, w_{new}, j_{new}, N_3$
V_4
$V_4' = h(w_{new} \oplus j_{new} \oplus N_3) \overset{?}{=} V_4$
$\alpha_{new}' = h(CID_{payer} \oplus w_{new} \oplus j_{new}) \overset{?}{=} \alpha_{new}$

Fig. 1 the scenario of our scheme

4.1. Registration phase

First, a payer has to register an identity to bank, he/she sends the real identity to bank, then bank generates an anonymous identity for payer, the scenario is shown as follows.

Step 1: The payer takes his/her real identity ID_{payer}

Step 2: Payer sends ID_{payer} to bank.

Step 3: When bank receives the ID_{payer} , it will check the ID_{payer} , then computes as follows:

$CID_{payer} = h(ID_{payer} \oplus d)$

$r_i = CID_{payer}^{SK_{bank}} \bmod n$

$\alpha = h(CID_{payer} \oplus w \oplus j)$

Store ID_{payer} and CID_{payer}

where CID_{payer} is an anonymity identity of payer, then the bank sends $(CID_{payer}, j, \alpha, r_i, w)$ to the payer. After receiving the message, the payer checks α as follows:

$\alpha' = h(ID_{payer} \oplus w \oplus j) \overset{?}{=} \alpha$

4.2. Paying phase

After the registration phase, the payer can use the anonymous identity to buy some goods from payee. First, we have to denote that is secure channel between bank and anyone, the scenario is described as follows:

Step 1: The payer chooses two integers R and b and computes:

$k = R^{PK_{payee}} \cdot (b \oplus N_1) \bmod n'$

then sends k to payee.

Step 2: After the payee receive the message k , he / she computes:

$k' = k^{SK_{payee}} = R \cdot (b \oplus N_1)^{SK_{payee}} \bmod n'$

And then sends k' to payer.

Step 3: Upon receiving k' , the payer computes

$M = k' \cdot R^{-1} = (b \oplus N_1)^{SK_{payee}} \bmod n'$

And verifies $w \overset{?}{\geq} 0$

when payer buys some goods from payee, he / she compute C_1 and C_2 as follows:

$C_1 = h(CID_{payer} \oplus a \oplus M)$

$C_2 = E_{SK_i}(j \| a \| N_2) \oplus r_i$

and sends the message $(C_1, C_2, a, b, N_1,\ N_2, CID_{payer}, ID_{bank})$ to payee.

Step 4: When the payee receives the message $(C_1, C_2, a, b, N_1, N_2, CID_{payer},\ ID_{bank})$, he / she checks the trading face value and the payer's identity if holds or not

$C_1' = h(CID_{payer} \oplus a \oplus (b \oplus N_1)^{SK_{payee}}) \overset{?}{=} C_1$

then sends the message $(C_2, a, N_2,\ CID_{payer})$ to the bank.

Step 5: After the bank receives the message $(C_2, a, N_2, CID_{payer})$, it can check the payer's identity and decrypt the message C_3 as follows:

$C_3 = C_2 \oplus CID_{payer}^{SK_{bank}}$

$D_{PK_i}(C_3) = (j \| a \| N_2)$

the bank gets the messages j, a, N_2 .

Step 6: The bank checks the transaction if double spending in the bank database as follows:

(1) Check if $(j \| a \| N_2)$ in the database,

(2) Store $(j \| a \| N_2)$ into the database.

Assume the transaction occurs double spending, the bank will reject the transaction and notify the payee.

Step 7: If the transaction has no double spending, the bank updates w, j, α as follows:

Check $w_{new} = w - a \overset{?}{\geq} 0,$

$j_{new} = j - 1 \overset{?}{\geq} 0$

$\alpha_{new} = h(CID_{payer} \oplus w_{new} \oplus j_{new})$

Compute

$V_1 = (ID_{bank} \oplus N_2)^{SK_{bank}}$

$V_2 = (w_{new} \oplus j_{new} \oplus N_3)^{PK_{payee}}$

the bank sends (accept message, V_1, V_2) to payee and sends $(\alpha_{new}, w_{new}, j_{new}, N_3)$ to payer.

Step 8: After receiving the accept message, the payee verifies the bank's identity and decrypts V_2 and obtains V_4 as follows:

$V_1^{PK_{bank}} \overset{?}{=} (ID_{bank} \oplus N_2)$

$V_3 = V_2^{SK_{payee}} = w_{new} \oplus j_{new} \oplus N_3$

$V_4 = h(V_3)$

Then sends V_4 to payer.

Step 9: When receiving the messages $\alpha_{new}, w_{new}, j_{new}, N_3$ from bank and V_4 from payee, payer verifies the identity of payee and bank whether hold by:

$$V_4' = h(w_{new} \oplus j_{new} \oplus N_3) \overset{?}{=} V_4$$

$$\alpha_{new}' = h(CID_{payer} \oplus w_{new} \oplus j_{new}) \overset{?}{=} \alpha_{new}$$

If it holds, then the transaction finishes.

5. SECURITY ANALYSIS

In this section, we will analyse security of our scheme to prove that are better than Chang et al.'s scheme.

5.1 The replay attack issue

Assume an attacker wants to replay the message $(C_1, C_2, a, b, N_1, N_2, CID_{payer}, ID_{bank})$ to pass the check of payee. In step 4 of paying phase, payee will check the nonce whether duplicate. When attacker replays the message, payee checks:

$$C_1' = h(CID_{payer} \oplus a \oplus (b \oplus N_1)^{SK_{payee}})$$

$$C_1' \overset{?}{=} C_1$$

In the above expression, the attacker can't pass payee's verification since N_1 is already sent one time, if attacker sends two times then payee can find the N_1 is duplicate and reject the message $(C_1, C_2, a, b, N_1, N_2, CID_{payer}, ID_{bank})$. Therefore, our scheme can prevent the replay attack.

5.2 The forgery attack issue

If an attacker wants to forger a message to deceit the payee, the attacker will fail. After the attacker forges a message $(C_1, C_2, a, b^*$ $N_1^*, N2, CID_{payer}, ID_{bank})$ and sends the forgery message to payee, then the payee checks $C_1' = h(CID_{payer} \oplus a \oplus (b^* \oplus N_1^*)^{SK_{payee}}) \overset{?}{=} C_1$

But the attacker doesn't have SK_{payee}, so attacker can't forge C_1 to deceit the payee. Our scheme can prevent the forgery attack.

5.3 The impersonating attack issue

If an attacker is able to impersonate a payer to use the e-check, the attacker will fail. First, the attacker has to negotiate with payee for obtaining

$$k' = k^{SK_{payee}}$$
$$= R \cdot (b \oplus N_1)^{SK_{payee}} \bmod n'$$

and computes as follows:

$$C_1^* = h(CID_{payer} \oplus a \oplus M)$$

$$C_2^* = E_{SK_i}(j \| a \| N_2) \oplus r_i$$

then sends the impersonating message $C_1^*, C_2^*, a, b, N_1, N_2, CID_{payer}, ID_{bank}$ to the payee and checks $C_1' = h(CID_{payer} \oplus a \oplus (b \oplus N_1)^{SK_{payee}}) \overset{?}{=} C_1$

If the above equation holds, sends the message to the bank. After receiving the message, the bank known it is an impersonating message since the message unable to pass the verification without the true session key *CK* to decrypt the *C*$_3$ to obtain $(j \| a \| N_2)$.

5.4 Denial-of-Service (DoS) attack issue

When an attacker wants to perform the denial-of-service (DoS) attack to the payee or bank, it will be fail. In our scheme, as long as send the message no matter payer, payee or bank, it must check the correctness of the message:

$$C_1' = h(CID_{payer} \oplus a \oplus (b \oplus N_1)^{SK_{payee}}) \overset{?}{=} C_1$$

Through verification messages, our scheme can decide the message whether correct. It prevents the DoS attack.

5.5 Double spending issue

Assume the payee wants to double spending the e-check, it is impossible. The bank will check the message C_2, a, N_2, CID_{payer} from payee. In step 6 of the paying phase, the bank searches the database whether exists $j \| a \| N_2$. If it holds, that must be double spending and reject the transaction. Our scheme can prevent the double spending.

5.6 Anonymity issue

In our proposed scheme, we think the e-check should be anonymous throughout the whole transaction process. In the registration phase, the payer could obtain an anonymous identity CID_{payer} from the bank. If the payer's e-check the number of times $j = 0$, he/she can take the anonymous identity CID_{payer} to apply for the e-check again from the bank. In the whole transaction process, the payee can not know the payer’s real identity. The proposed scheme keeps anonymity.

5.6 The validity of e-check

Assume the payer uses the e-check to buy some goods; the payee will doubt the e-check whether correct. This time, payee can send C_2 to bank and use $CID_{payer}^{SK_{bank}}$ to compute $C_3 = C_2 \oplus CID_{payer}^{SK_{bank}}$. Next, the bank decrypts the message C_3 as follows: $D_{PK_i}(C_3) = (j \| a \| N_2)$, the bank can know whether payer’s identity and the check correct or not.

Finally, we make a comparison with Chang et al.’s scheme in table 2.

TABLE 3
THE COMPARISONS OF OUR SCHEME AND CHANG ET AL.’S SCHEME

	Chang et al.’s scheme	Our scheme
Prevent replay attack	Y	Y
Prevent forgery attack	Y	Y
Prevent impersonating attack	Y	Y
Prevent DoS attack	Y	Y
Solve double spending	Y	Y
Solve anonymity issue	N	Y
Solve large computation issue	N	Y
Solve time synchronous issue	N	Y

6. CONCLUSION

In this paper, we proposed an efficient and secure on-line electronic check system. The proposed scheme not only prevents some attacks (such as the replay attack, forgery attack, impersonating attack and DoS attack) but also solves the Chang et al.’s anonymity, large computation, time synchronous issues. Our scheme is more secure and practical to be applied to e-commerce.

REFERENCES

[1] D. Chaum, A. Fiat, M. Naor, Untraceable electronic cash, *Proceedings of advances in Crypto’88*, Califomia, USA, vol. 403, pp. 319–327, 1990.

[2] D. Chaum, B. Den Bore, E. Van Heyst, S. Mjolsnes, A. Steenbeek, “Efficient offline electronic check”, *Proceedings of advances in Eurocrypt’98*, Germany, vol. 434, pp. 294–301, 1989.

[3] W. K. Chen, “Efficient on-line electronic checks,” *Applied Mathematics and Computation.*, vol. 162, pp. 1259–1263, 2005.

[4] C. C. Chang, S. C. Chang, and J. S. Lee, “An on-line electronic check system with mutual authentication,” *Computers and Electrical Engineering*, vol. 35, pp. 757–763, 2009.

[5] J. E. Hsien, C. C. Hsuen, and C. Y. Chen, “An electronic traveler’s check system,” *2001 Conference on Theory and Practice for Electronic Commerce*, pp. 164–169, 2001.

[6] H. T. Liaw, J. F. Lin, and W. C. W, “A new electronic traveler’s check scheme based on one-way hash function,” *Electronic Commerce Research and Applications*, vol. 6, pp. 499–508, 2007.