Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **3**, May **2015**

A Secure and Efficient Scheme for Digital Gift Certificates

Yanjun Liu

Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education
Department of Electrical Engineering
Anhui University
111 Jiu-Long road, Hefei, 230039, China
Department of Computer Science and Information Engineering
Asia University
500 Liou-feng Road, Taichung, 413, Taiwan
yjliu104@gmail.com

Chin-Chen Chang

Department of Computer Science and Information Engineering
Asia University
500 Liou-feng Road, Taichung, 413, Taiwan
Department of Information Engineering and Computer Science
Feng Chia University
100 Wen-hwa Road, Taichung, 407, Taiwan
alan3c@gmail.com

Chin-Yu Sun

Department of Computer Science
National Tsing-Hua University
101 Kuang-Fu Road, Hsinchu, 300, Taiwan
sun.chin.yu@gmail.com

Received April, 2014; revised August, 2014

Abstract. *With the increasingly rapid developments in e-commerce, schemes for digital gift certificates have become prevalent electronic payment systems due to their practicality and simplicity. In 2002, Chan and Chang introduced the concept of digital gift certificates and proposed the first scheme associated with them. Recently, Chang and Chang (2005) proposed a more practical scheme, called DGC-US (digital gift certificates with unconstrained spending), which can reduce the computational cost of all entities involved in Chan and Chang's scheme, especially the computational cost of the electronic department store. However, the asymmetric encryption/decryption operations required in the DGC-US scheme still waste a significant amount of time. In this paper, we propose a novel authentication scheme for digital gift certificates based on Chebyshev chaotic maps and the blind signature. Our proposed scheme can satisfy the essential security requirements and withstand various well-known attacks. Our analyses have shown that our proposed scheme maintains all of the merits of the DGC-US scheme but is more efficient in terms of computational cost.*

Keywords: Digital gift certificate, Chebyshev chaotic map, Blind signature, E-commerce, Authentication

1. **Introduction.** With the tremendous breakthroughs in network technologies and information security, electronic payment systems have become one of the most significant modes of payment in on-line e-commerce. Typical electronic payment systems [1-9], such as on-line credit card payments, electronic check (e-check), electronic cash (e-cash), and digital gift certificates are all important in the transactions between customers and various electronic department stores due to their convenience and high efficiency.

In an on-line credit card payment system [10, 11], customers can pay for almost everything they purchase with their credit cards through the Internet. However, such payments are not feasible if a transaction involves a large amount of money that may exceed the limit approved by the issuer of the credit card. More serious security problems occur when credit cards are lost or stolen by an attacker for illegal use. An e-cash payment system [12, 13, 14, 15, 16, 17, 18] can maintain anonymity and reduce computational cost. However, since the ownership of the e-cash is not established, any person who has access to the e-cash can complete the transaction successfully, just as is done in on-line credit card payment systems. The e-check payment system [19, 20, 21] can overcome this weakness by incorporating the identity of the authorized customer into the e-check. Unfortunately, a trust bank must issue the e-check to the customer in advance and authenticate its validity later, which may complicate the payment system.

The payment system of digital gift certificates [22, 23] is a newly-emerging technique for e-commerce due to its practicality and simplicity. This kind of payment system, in which a digital gift certificate is used as currency between customers and electronic department stores, was introduced by Chan and Chang [22] in 2002. In their scheme, the electronic department store can sell a digital gift certificate to a specified customer for buying goods. The certificate is only generated by both of electronic department stores and customers and it contains a defined amount of money and some secret information about both entities. When a customer wants to buy some goods, the amount of the spent digital certificate is usually less than the price of the goods. If the amount of the spent digital certificate is more than the value of the goods, no change will be given.

In order to increase the demand for the use of such certificates, the electronic department store usually offers a discount when a customer purchases several digital gift certificates at one time. Unlike the aforementioned payment systems, digital gift certificates can be used by anyone, so, if the purchaser so desires, he or she can give or sell the certificates to anyone for subsequent use in buying goods. Therefore, the digital gift certificate can provide the functionality of transferring ownership from the holder to another person. In addition, the payment system used with digital gift certificates is much simpler since there is no need for a bank to be involved in the scheme.

Recently, Chang and Chang [23] pointed out that the computational cost of the electronic department store becomes large in Chan and Chang‘s scheme [22] when many customers purchase digital gift certificates or commodities at the same time, and they proposed a secure and practical scheme, i.e., DGC-US (digital gift certificates with unconstrained spending), to solve this problem. According to the analysis of its performance, the DGC-US scheme can reduce the computational cost of all entities involved in Chan and Chang‘s scheme, especially the computational cost of the electronic department store. Unfortunately, we determined that the DGC-US requires a large number of asymmetric encryption/decryption operations because it is based on the public key cryptosystem, and these operations take a lot of time.

In this paper, we propose a novel scheme for digital gift certificates. The contributions of our proposed scheme are listed below:

(1) To the best of our knowledge, we are the first to use chaotic maps as one of the

main building blocks in the architecture of a scheme for digital gift certificates.
(2) Our proposed scheme is secure against various well-known attacks, such as double spending, replay attacks, man-in-the-middle attacks, insider attacks, and impersonation attacks.
(3) The blind signature mechanism was used to ensure the mutual authentication of the customer and the electronic department store. Furthermore, the use of a blind signature provides two other functionalities, i.e., non-repudiation and confidentiality of price, which can enhance the security of our proposed scheme.
(4) Our proposed scheme maintains the merits of the DGC-US scheme, but it is more efficient in terms of computational cost.

The rest of this paper is organized as follows. Section 2 addresses some background information concerning the basic building blocks of the proposed scheme. Section 3 describes the details of our proposed scheme. Security and performance analyses of our proposed scheme are given in Sections 4 and 5, respectively. Our conclusions are presented in Section 6.

2. **Related Works.**

In this section, we briefly introduce some basic knowledge that is necessary in the design of our proposed scheme. More specifically, two fundamental building elements, Chebyshev chaotic maps [24, 25] and blind signature [26, 27] will be described in the following.

2.1. **Chebyshev chaotic maps.**

In the subsection, we first discuss the characteristics of Chebyshev chaotic maps and then introduce how to use chaotic maps to design a key agreement protocol [28, 29].

Let us select an integer n and a variable x in the interval [-1, 1]. Then, the Chebyshev polynomial $T_n(X)$ of degree n is defined as $T_n(X) = cos(n * arccos\ x)$. Additionally, $T_n(X)$ can be calculated in an iterative algorithm as $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$, where $T_0(x) = 1$, $T_1(x) = x$ and $n \geq 1$. One of the most important characteristics of the Chebyshev polynomial is that there exists a nice relation such that $T_u(T_v(x)) = T_v(T_u(x))$ which is called semi-group characteristic.
Recently, Xiao et al. [25] proposed a key agreement protocol based on Chebyshev chaotic maps that is very similar to the concept of Diffie-Hellman scheme [30]. Details of the key agreement protocol are described as follows:

Step 1. Two entities A and B share a common random number x, where $x \in [-1, 1]$ and x is made publicly known.
Step 2. A chooses one random large integer u and computes $U = T_u(x)$.
Step 3. A sends U to B.
Step 4. B chooses one random large integer v and computes $V = T_v(x)$.
Step 5. B sends V to A.
Step 6. A computes the key as $T_u(V) = T_u(T_v(x))$ and B computes the key as $T_v(U) = T_v(T_u(x))$.

Now we can get $T_u(V) = T_v(U) = T_{uv}(x)$ because of the semi-group characteristic. This implies that A and B can use the common session key to encrypt the transmitted message in future communications. Although this key agreement protocol is vulnerable to man-in-the middle attacks, the concept of Encrypted Key Exchange (EXE) [33] can be used in it to withstand this kind of attack. Interest reader can refer to the original paper [25] for detailed discussion. First, the signer selects two large random primes p and q and computes $n = p \cdot q$.

2.2. **The blind signature.** In 1983, Chaum [26] proposed the notion of blind signature which is a signature scheme based on the RSA cryptosystem [30, 31, 32]. Two entities, the signer and the user are involved in the blind signature scheme and they perform the initialization as follows. First, the signer selects two large random primes p and q and computes $n = p \cdot q$. Then, the signer selects a random integer e satisfying $1 < e < \phi(n)$ and $GCD(e, \phi(n)) = 1$, where $\phi(n) = (p-1) \cdot (q-1)$. (e, n) is the public key of the signer. After that, the signer computes his private key d satisfying $1 < d < \phi(n)$ and $d \cdot e \equiv 1(mod\phi(n))$. The signer publishes the public key (e, n) and preserves the private key d secretly. Assuming that the message to be signed is denoted as m, the process of the blind signature is executed below:

Step 1. The user selects a random integer R and computes $C = R^e m \ mod \ n$, where $GCD(R, n) = 1$.
Step 2. The user sends C to the signer.
Step 3. Upon receiving C, the signer computes the blind signature of m as $r = C^d \ mod \ n = (R^e m)^d \ mod \ n = Rm^d \ mod \ n$.
Step 4. The signer sends r to the user.
Step 5. The user computes the signature of m as $s = R^{-1} r \ mod \ n = m^d \ mod \ n$.

Therefore, the user can verify the signature s by checking whether m is equal to $s^e \ mod \ n$.

3. **Our Proposed Scheme.** In this section, we propose a novel authentication scheme for digital gift certificates based on chaotic maps and the blind signature. Two types of participants, i.e., the customer and the electronic department store (also called the server), are involved in our proposed scheme. Customers can purchase digital gift certificates from the electronic department store and use them later to purchase the commodities they want after being authenticated by the electronic department store. Of course, holders of digital gift certificates can give or sell them to someone if they so desire. Therefore, our proposed scheme enables the digital gift certificate to maintain a unique ownership that can be transferred from the holder to someone else. In addition, the customer can authenticate the validity of the electronic department store based on the blind signature mechanism.

Our proposed scheme consists of three phases, i.e., 1) the application phase; 2) the ownership-transference phase; and 3) the payment phase. In the following, we describe the detailed steps of each phase.

3.1. **Application phase.** The application phase allows a customer to purchase a digital gift certificate via a secure channel after communicating with the electronic department store. The digital gift certificate is generated by the cooperation of the customer and the electronic department store, so it includes secret information from both of them. Suppose that customer A wants to purchase a digital gift certificate with a face value m from an electronic department store S. Initially, S selects the public key pk_{server} and the private key sk_{server} according to the RSA cryptosystem [30, 31, 32]. The application phase is performed as follows and is shown in Figure 1.

Step 1. Customer A chooses m and a random number r. Then, A computes $a_0 = h(ID_A \parallel h^m(r))$ and $b_0 = h(a_0)$, where ID_A is the identity of A, $\parallel$ is the string concatenation operator, and $h(\ )$ is a secure, one-way hash function. In addition, $h^m(r) = h(h^{(m-1)}(r))$ for

$m \geq 1$.
Step 2. A sends ID_A, b_0, and m to S through a secure channel.
Step 3. S computes $b_1 = h^m(a_0)$ and generates a signature $s_0 = (b_1)^{sk_{server}}$. Then, S stores (ID_A, s_0, m) in the database. S also selects a private key n and publishes a public key $(x, T_n(x))$ based on Chebyshev chaotic maps [24].
Step 4. S sends s_0 to A securely.
Step 5. After receiving s_0 from S, A computes $b_2 = h^m(a_0)$ and then verifies whether $(s_0)^{pk_{server}}$ is equal to b_2 to check the integrity of the transmitted message. If they are not equal, A terminates the procedure; otherwise, A stores the digital gift certificate (ID_A, s_0, m, r).

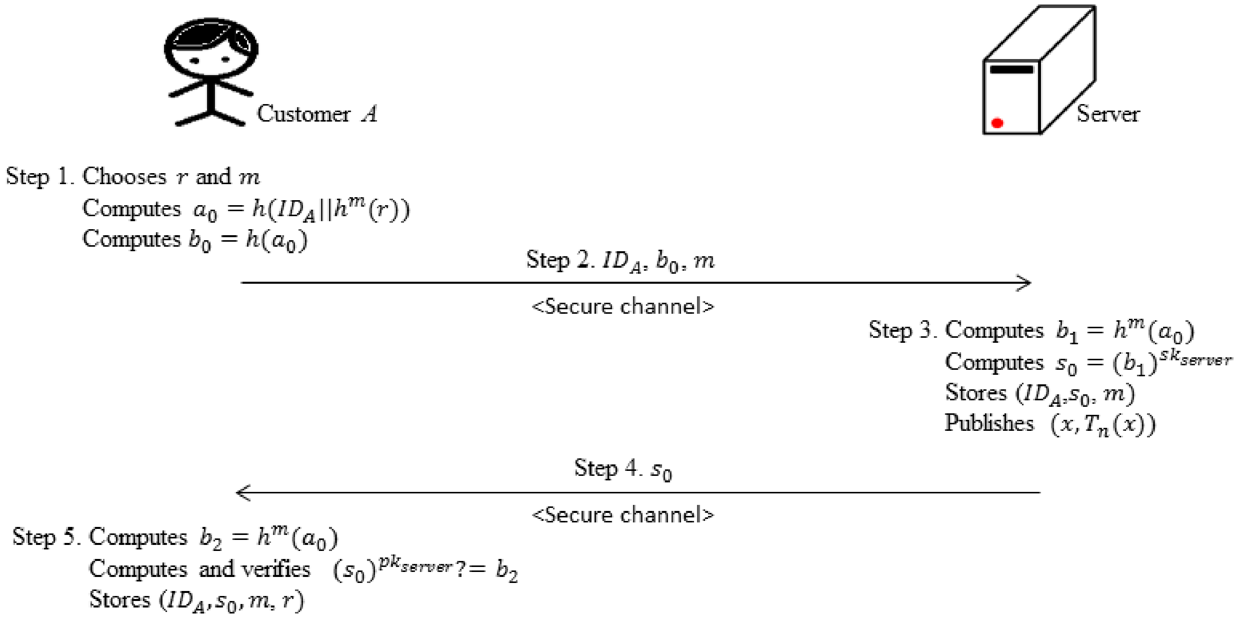
Customer A
Server
Step 1. Chooses r and m
Computes $a_0 = h(ID_A||h^m(r))$
Computes $b_0 = h(a_0)$
Step 2. ID_A, b_0, m
<Secure channel>
Step 3. Computes $b_1 = h^m(a_0)$
Computes $s_0 = (b_1)^{sk_{server}}$
Stores (ID_A, s_0, m)
Publishes $(x, T_n(x))$
Step 4. s_0
<Secure channel>
Step 5. Computes $b_2 = h^m(a_0)$
Computes and verifies $(s_0)^{pk_{server}} ?= b_2$
Stores (ID_A, s_0, m, r)

FIGURE 1. Application phase in our proposed scheme

3.2. **Ownership-transference phase.** If customer A does not want to purchase some commodities with the digital gift certificate he or she obtained in the application phase, he or she can give or sell his or her digital gift certificate to someone. This indicates that the ownership of the digital gift certificate can be transferred from A to another customer, namely C. Assuming that A and C know each others identity (ID), the details of the ownership-transference phase are described as follows and is shown in Figure 2.

Step 1. C chooses a random number w and computes $T_w(x)$. C also generates a timestamp T_c.
Step 2. C sends $T_w(x)$ and T_c to A.
Step 3. A chooses a random number v and computes $T_v(x)$. Then, A generates the key shared between A and C as $K_{AC} = T_vT_w(x)$ and the key shared between A and S as $K_{AS} = T_vT_n(x)$, respectively. After that, A encrypts ID_A, ID_C, s_0, m and T_c with the key K_{AS} as $i = E_{K_{AS}}(ID_A \parallel ID_C \parallel s_0 \parallel m \parallel T_C)$. After obtaining i, A encrypts ID_A, ID_C, s_0, m, T_c, and i with the key K_{AC} as $g = E_{K_{AC}}(ID_A \parallel ID_C \parallel s_0 \parallel m \parallel T_C \parallel i)$.
Step 4. A sends g and $T_v(x)$ to C.

Step 5. C computes the key $K_{CA} = T_w T_v(x)$ and then uses K_{CA} to decrypt g to obtain ID_A, ID_C, s_0, m, T_c, and i.
Step 6. C checks the timestamp T_C. If T_C is valid, C chooses a random number $r^{'}$. Then, C computes $a_1 = h(ID_C \parallel h^m(r^{'}))$ and $b_3 = h(a_1)$.
Step 7. C generates the key shared between C and S as $K_{CS} = T_w T_n(x)$ and a timestamp $T_C^{'}$. Afterwards, C uses K_{CS} to encrypt ID_A, ID_C, s_0, m, b_3, i and $T_C^{'}$ as $j = E_{K_{CS}}(ID_A \parallel ID_C \parallel s_0 \parallel m \parallel b_3 \parallel i \parallel T_C^{'})$.
Step 8. C sends j, $T_w(x)$, and $T_v(x)$ to S.
Step 9. S computes the key $K_{SC} = T_n T_w(x)$ and $K_{SA} = T_n T_v(x)$, respectively. Then, S decrypts j by K_{SC} and i by K_{SA} to retrieve ID_A, ID_C, s_0, m, b_3, T_C, and $T_C^{'}$. Then, S checks the validity of ID_A, ID_C, s_0, m and T_C. If they hold, the phase continues; otherwise, S terminates the phase.
Step 10. S searches for the record (ID_A, s_0, m) in the database. If the record exists, S can confirm that A indeed wants to give or sell the digital gift certificate to C and continue the procedure; otherwise, the procedure is terminated.
Step 11. S checks the timestamp $T_C^{'}$. If $T_C^{'}$ is valid, S computes $b_4 = h^m(a_1)$ and $s_1 = (b_1)^{SK_{server}}$. Then, S records (ID_C, s_1, m) in the database and deletes (ID_A, S_0, m).
Step 12. S encrypts s_1 and $T_C^{'}$ with the key K_{SC} as $E_{K_{SC}}(s_1 \parallel T_C^{'})$, and then sends it to C.
Step 13. C decrypts the transmitted message by the key K_{CS} to extract s_1 and $T_C^{'}$. Then, C checks the timestamp $T_C^{'}$. If $T_C^{'}$ is valid, C computes $b_5 = h^m(a_1)$ and verifies whether $(s_1)^{pk_{server}}$ is equal to b_5 or not. If they are not equal, C terminates the procedure; otherwise, C stores the digital gift certificate $(ID_C, s_1, m, r^{'})$.
Step 14. C sends $E_{K_{CA}}(Success)$ to A.
Step 15. A uses K_{AC} to decrypt the received message and deletes (ID_A, s_0, m, r).

3.3. **Payment phase.** After customer A purchases the digital gift certificate by performing the application phase, he or she can purchase some commodities with the digital gift certificate from the electronic department store S when the payment phase is completed. In the payment phase, A and S first make some pre-computations based on the blind signature mechanism. Here, we assume that c denotes the amount of money being spent in this transaction. Therefore, the pre-computation contains the following steps:
A chooses a random number R and computes $k = R^{pk_{server}} \cdot c$. Then, A sends k to S. Upon receiving k, S computes $k^{'} = (k)^{sk_{server}} = R \cdot c^{sk_{server}}$ and sends $k^{'}$ to A. After the pre-computation, the payment phase can be executed as follows and is shown in Figure 3.

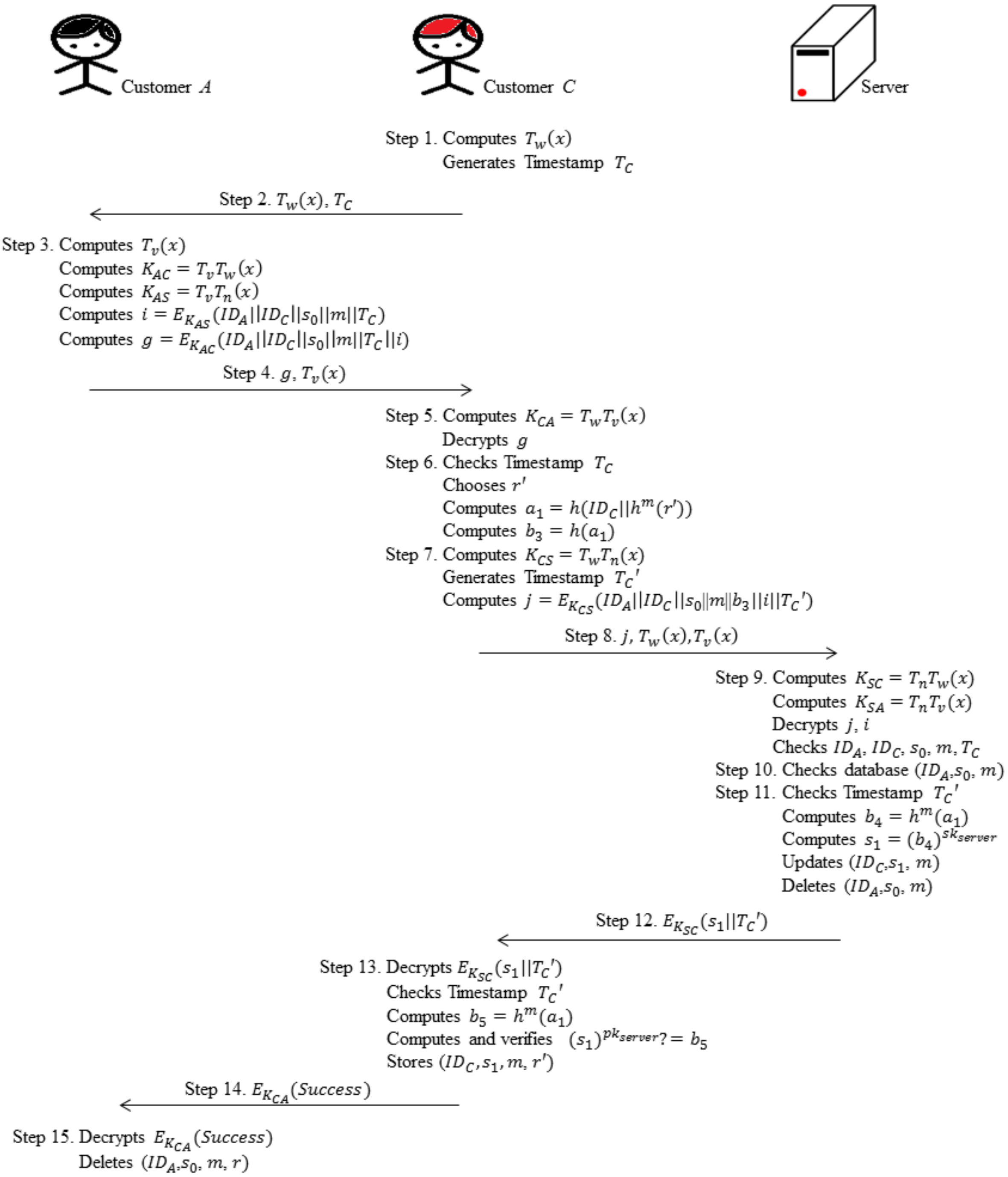
Customer A
Customer C
Server
Step 1. Computes $T_w(x)$
Generates Timestamp T_C
Step 2. $T_w(x)$, T_C
Step 3. Computes $T_v(x)$
Computes $K_{AC} = T_vT_w(x)$
Computes $K_{AS} = T_vT_n(x)$
Computes $i = E_{K_{AS}}(ID_A||ID_C||s_0||m||T_C)$
Computes $g = E_{K_{AC}}(ID_A||ID_C||s_0||m||T_C||i)$
Step 4. g, $T_v(x)$
Step 5. Computes $K_{CA} = T_wT_v(x)$
Decrypts g
Step 6. Checks Timestamp T_C
Chooses r'
Computes $a_1 = h(ID_C||h^m(r'))$
Computes $b_3 = h(a_1)$
Step 7. Computes $K_{CS} = T_wT_n(x)$
Generates Timestamp $T_C{}'$
Computes $j = E_{K_{CS}}(ID_A||ID_C||s_0||m||b_3||i||T_C{}')$
Step 8. j, $T_w(x)$,$T_v(x)$
Step 9. Computes $K_{SC} = T_nT_w(x)$
Computes $K_{SA} = T_nT_v(x)$
Decrypts j, i
Checks ID_A, ID_C, s_0, m, T_C
Step 10. Checks database $(ID_A$,s_0, $m)$
Step 11. Checks Timestamp $T_C{}'$
Computes $b_4 = h^m(a_1)$
Computes $s_1 = (b_4)^{sk_{server}}$
Updates $(ID_C$,s_1, $m)$
Deletes $(ID_A$,s_0, $m)$
Step 12. $E_{K_{SC}}(s_1||T_C{}')$
Step 13. Decrypts $E_{K_{SC}}(s_1||T_C{}')$
Checks Timestamp $T_C{}'$
Computes $b_5 = h^m(a_1)$
Computes and verifies $(s_1)^{pk_{server}} ?= b_5$
Stores $(ID_C$,s_1, m, $r')$
Step 14. $E_{K_{CA}}(Success)$
Step 15. Decrypts $E_{K_{CA}}(Success)$
Deletes $(ID_A$,s_0, m, $r)$

FIGURE 2. Ownership-transference phase in our proposed scheme

Step 1. A computes $d = R^{-1} \cdot k' = c^{sk_{server}}$ and then verifies whether $(d)^{pk_{server}}$ equals c or not. If they are equal, A computes $e = h^{m-c}(r) \oplus d$; otherwise, the transaction is terminated.
Step 2. A computes $T_v(x)$ and generates $K_{AS} = T_vT_n(x)$. Then, A generates a timestamp T_A.
Step 3. A sends $E_{K_{AS}}(ID_A \parallel s_0 \parallel m \parallel e \parallel c \parallel T_A)$ and $T_v(x)$ to S.
Step 4. S computes $K_{SA} = T_nT_v(x)$. After that, S uses K_{SA} to decrypt $E_{K_{AS}}(ID_A \parallel s_0 \parallel m \parallel e \parallel c \parallel T_A)$ to retrieve ID_A, s_0, m, e, c and T_A.
Step 5. S checks the validity of the timestamp T_A. Then, S searches for the record

(ID_A, s_0, m) in the database to check on double-spending. If the record exists, S continues the transaction; otherwise, the transaction is terminated.
Step 6. S checks whether $m \geq c$. If it holds, S computes $c^{sk_{server}}$ and $f = h^c(e \oplus c^{sk_{server}})$. Then, S checks the integrity of the transmitted message in Step 3 by verifying whether $h^m(h(ID_A \parallel f))$ equals $(s_0)^{pk_{server}}$. If they are equal, the transaction continues; otherwise, the transaction is terminated.
Step 7. S computes $b_6 = h^{m-c}(a_0)$ and $s_2 = (b_6)^{sk_{server}}$. After that, S updates (ID_A, s_0, m) to $(ID_A, s_2, m - c)$ in the database.
Step 8. S sends $E_{K_{SA}}(s_2 \parallel T_A)$ to A.
Step 9. A decrypts $E_{K_{SA}}(s_2 \parallel T_A)$ by the key K_{AS} to extract s_2 and T_A. Then, A checks the timestamp T_A. If T_A is valid, A computes $b_7 = h^{m-c}(a_0)$ and verifies whether $(s_2)^{pk_{server}}$ equals b_7. If the verification is passed, A updates (ID_A, s_0, m, r) to $(ID_A, s_2, m - c, r)$.

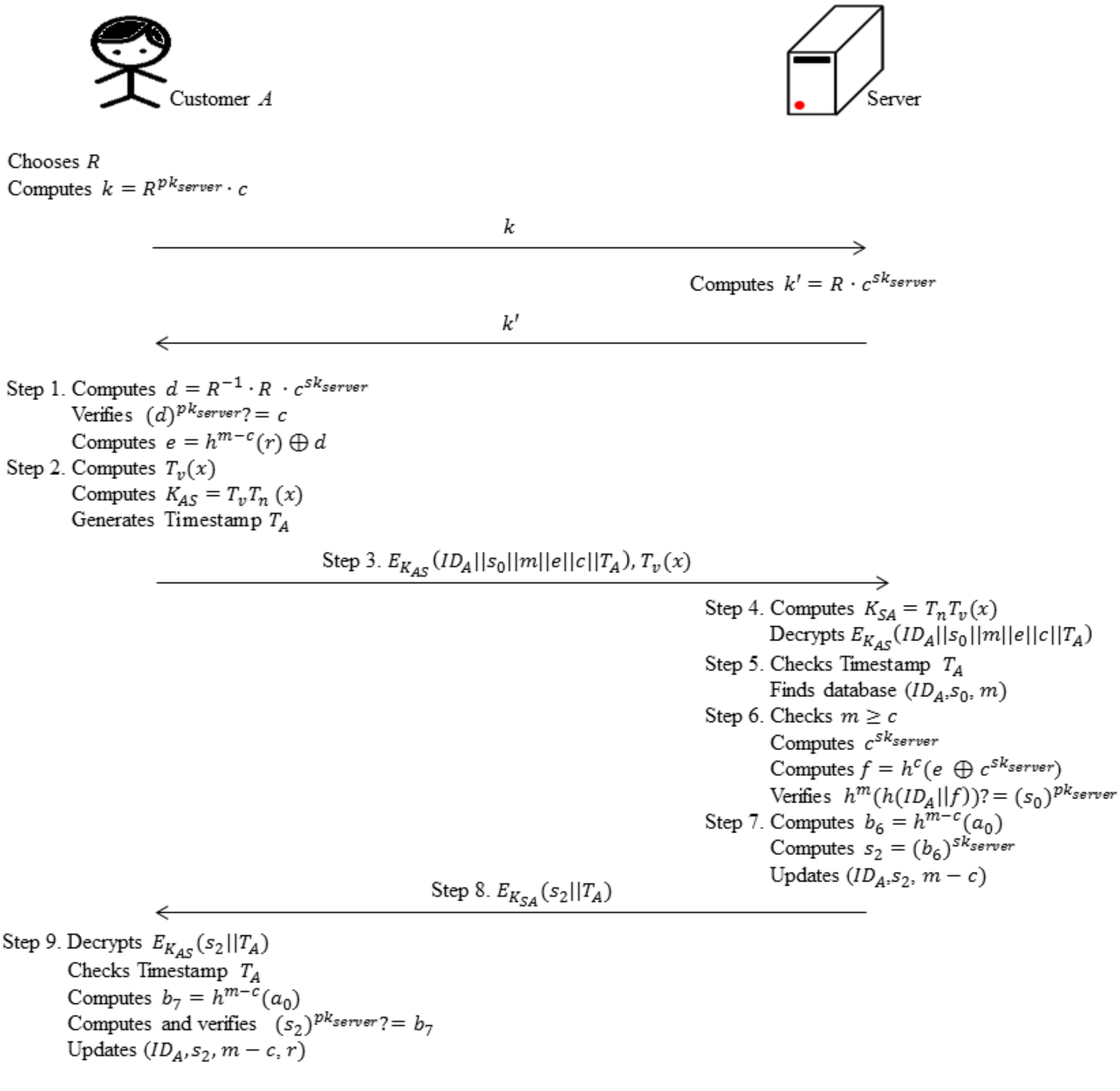
Customer A
Server
Chooses R
Computes $k = R^{pk_{server}} \cdot c$
k
Computes $k' = R \cdot c^{sk_{server}}$
k'
Step 1. Computes $d = R^{-1} \cdot R \cdot c^{sk_{server}}$
Verifies $(d)^{pk_{server}} ?= c$
Computes $e = h^{m-c}(r) \oplus d$
Step 2. Computes $T_v(x)$
Computes $K_{AS} = T_v T_n(x)$
Generates Timestamp T_A
Step 3. $E_{K_{AS}}(ID_A||s_0||m||e||c||T_A), T_v(x)$
Step 4. Computes $K_{SA} = T_n T_v(x)$
Decrypts $E_{K_{AS}}(ID_A||s_0||m||e||c||T_A)$
Step 5. Checks Timestamp T_A
Finds database (ID_A, s_0, m)
Step 6. Checks $m \geq c$
Computes $c^{sk_{server}}$
Computes $f = h^c(e \oplus c^{sk_{server}})$
Verifies $h^m(h(ID_A||f)) ?= (s_0)^{pk_{server}}$
Step 7. Computes $b_6 = h^{m-c}(a_0)$
Computes $s_2 = (b_6)^{sk_{server}}$
Updates $(ID_A, s_2, m - c)$
Step 8. $E_{K_{SA}}(s_2||T_A)$
Step 9. Decrypts $E_{K_{AS}}(s_2||T_A)$
Checks Timestamp T_A
Computes $b_7 = h^{m-c}(a_0)$
Computes and verifies $(s_2)^{pk_{server}} ?= b_7$
Updates $(ID_A, s_2, m - c, r)$

FIGURE 3. Payment phase in our proposed scheme

4. **Security Analysis.** In this section, we show that our analysis of the proposed scheme indicated that the scheme is secure under various attack scenarios. More specifically, our proposed scheme can resist double spending, replay attacks, man-in-the-middle attacks, insider attacks, and impersonation attacks. Throughout the security analysis, we assume that Bob is an attacker who is able to eavesdrop and intercept the message transmitted in the public channel between participants involved in our proposed scheme.

4.1. **Mutual authentication.** Our proposed scheme can achieve mutual authentication in the payment phase. Recall that the payment phase contains a pre-computation process in which the electronic department store S generates a blind signature $k^{'}$ of c with its private key sk_{server} as $k^{'} = (k)^{sk_{server}} = R \cdot c^{sk_{server}}$. Then, S sends $k^{'}$ to customer A. Subsequently, in Step 1 of the payment phase, A uses the received $k^{'}$ to compute $d = R^{-1} \cdot k^{'} = c^{sk_{server}}$ and verifies whether $(d)^{pk_{server}}$ equals c or not. If they are equal, A can confirm that S is valid since only a legal S has the correct private key sk_{server}. In addition, after A receives $E_{K_{AS}}(s_2 \parallel T_A)$ from S in Step 8, he can check the timestamp T_A and verify whether $(s_2)^pk_{server}$ equals b_7 by decrypting $E_{K_{AS}}(s_2 \parallel T_A)$. This indicates that S has the correct key $E_{K_{AS}}$ to encrypt s_2 and T_A. Therefore, A can verify the identity of S. On the other hand, if S can decrypt $E_{K_{AS}}(ID_A \parallel s_0 \parallel m \parallel e \parallel c \parallel T_A)$ sent from A in Step 3 to pass the following verification, S can authenticate that A is legal for the same reason.

4.2. **Functionalities of the blind signature.** In the payment phase, a pre-computation process is made based on the blind signature mechanism. First, A computes $k = R^{pk_{server}} \cdot c$ and sends it to S. Then, S generates a blind signature $k^{'}$ of c as $k^{'} = (k)^{sk_{server}} = R \cdot c^{sk_{server}}$ and sends $k^{'}$ to A. After that, A computes $d = R^{-1} \cdot k^{'} = c^{sk_{server}}$ and verifies whether $(d)^{pk_{server}}$ equals c or not.

Besides providing the authentication of S that is mentioned in Subsection 4.1, the blind signature mechanism used in our proposed scheme has two functionalities, i.e., 1) non-repudiation and 2) confidentiality of price. The first functionality means that S publishes c as the price of the goods that A wants to purchase, but later on, S may claim that the price is not c but another value $c^{'}$ and require A to pay an additional amount. However, in our proposed scheme, S cannot cheat A because he cannot deny that he has published the price c according to the blind signature. After receiving $k^{'}$ from S, A computes $d = R^{-1} \cdot k^{'}$ and verifies whether $(d)^{pk_{server}}$ equals c. If the verification is passed, A will be convinced that S has made a signature on c. For second functionality, we consider the scenario that the attacker Bob intercepts k and $k^{'}$. Since Bob does not know the random number R selected by A, he cannot obtain any useful information about the price c from k and $k^{'}$. If Bob is a competitor of S, this mechanism can prevent the price of goods provided by S from being revealed, thereby protecting the privacy of S. Therefore, the use of the blind signature can enhance the security of our proposed scheme.

4.3. **Resisting double spending.** If A somehow copies the original digital gift certificate and attempts to spend this digital gift certificate (ID_A, s_0, m, r) twice in our system, he will fail based on the following explanation. When A uses his or her digital gift certificate (ID_A, s_0, m, r) to spend for the first time, he or she sends $E_{K_{AS}}(ID_A \parallel s_0 \parallel m \parallel e \parallel c \parallel T_A)$ to S. S then decrypts the transmitted message, finds the record (ID_A, s_0, m) in the database, and updates (ID_A, s_0, m) to $(ID_A, s_2, m - c)$. After that, A sends $E_{K_{AS}}(ID_A \parallel s_0 \parallel m \parallel e \parallel c \parallel T_A^{'})$ with a new timestamp to S again. However, after decrypting, S cannot find (ID_A, s_0, m) at this time because S has replaced it with $(ID_A, s_2, m - c)$. Consequently, S will reject the double-spending request from A and terminate the procedure.

4.4. **Resisting the impersonation attack.** In this subsection, we demonstrate that our proposed scheme can withstand the impersonation attack in the ownership-transference phase and in the payment phase. In addition, we analyze two scenarios for the impersonation attack in each phase, i.e., (1) Bob's impersonating customer A and (2) Bob's impersonating the electronic department store S.

4.4.1. *The ownership-transference phase.* **Scenario 1.** The attacker Bob is impersonating A

If Bob attempts to impersonate A, he first intercepts $T_w(x)$ and T_c sent from C. Without A's correct parameters, v, s_0 and m, Bob must choose a random number v^* and generate fake ${s_0}^*$ and m^* by himself. Afterwards, Bob computes ${K_{AC}}^* = T_{v^*}T_w(x)$ and ${K_{AS}}^* = T_{v^*}T_n(x)$ and then uses ${K_{AC}}^*$ and ${K_{AS}}^*$ to generate the encrypted messages $i^* = E_{{K_{AS}}^*}(ID_A \parallel ID_C \parallel {s_0}^* \parallel m^* \parallel T_C)$ and $g^* = E_{{K_{AC}}^*}(ID_A \parallel ID_C \parallel {s_0}^* \parallel m^* \parallel T_C \parallel i^*)$. Next, Bob sends g^* and $T_{v^*}(x)$ to C. C computes ${K_{CA}}^* = T_wT_{v^*}(x)$ and then uses ${K_{CA}}^*$ to decrypt g^* to obtain ID_A, ID_C, ${s_0}^*$, m^*, T_c, and i^*. After checking the timestamp T_c, C chooses a random number r' and then computes ${a_1}^* = h(ID_C \parallel h^{m^*}(r'))$ and ${b_3}^* = h({a_1}^*)$. Then, C generates $K_{CS} = T_wT_n(x)$ and a timestamp $T_C{}'$, and computes $j^* = E_{K_{CS}}(ID_A \parallel ID_C \parallel {s_0}^* \parallel m^* \parallel {b_3}^* \parallel i^* \parallel T_C{}')$. After that, C sends j^*, $T_w(x)$, and $T_{v^*}(x)$ to S. Upon receiving the transmitted message, S computes $K_{SC} = T_nT_w(x)$ and ${K_{SA}}^* = T_nT_{v^*}(x)$. Then, S can decrypt j by K_{SC} and i by ${K_{SA}}^*$. Fortunately, S fails to find the record $(ID_A, {s_0}^*, m^*)$ in the database since ${s_0}^*$ and m^* cannot match the correct corresponding message in S's database. Therefore, S terminates the procedure immediately.

Scenario 2. The attacker Bob is impersonating S

If Bob wants to impersonate S, he first intercepts j, $T_w(x)$, and $T_v(x)$ sent from C in Step 8. However, Bob cannot compute the correct K_{SC} and K_{SA} because he cannot obtain the private key n selected by S for the Chebyshev chaotic maps scheme. As a result, Bob is unable to decrypt j and i to extract any useful parameters inside them. At this point, Bob jumps directly to Step 12 by skipping Steps 10 and 11. Here, Bob forges ${b_4}^*$, ${s_1}^* = ({b_4}^*)^{{sk^*}_{server}}$, $(T_C{}')^*$, and ${K_{SC}}^*$, and then impersonates S by sending ${K_{SC}}^*({s_1}^* \parallel (T_C{}')^*)$ to C. Fortunately, C cannot decrypt ${K_{SC}}^*({s_1}^* \parallel (T_C{}')^*)$ since the real key K_{SC} that C holds is different from the forged ${K_{SC}}^*$. Thus, C detects the illegitimacy of the received message and terminates the procedure.

4.4.2. *The payment phase.* **Scenario 1.** The attacker Bob is impersonating A

In this scenario, Bob performs the following steps to impersonate A. First, Bob chooses a random number R^* and computes $k^* = R^{*pk_{server}} \cdot c$. Then, Bob sends k^* to S. Upon receiving k^*, S computes $(k')^* = (k^*)^{sk_{server}} = R^* \cdot c^{sk_{server}}$ and sends $(k')^*$ to Bob. After that, Bob computes $d^* = (R^*)^{-1} \cdot (k')^* = c^{sk_{server}}$ and passes the verification of $(d^*)^{pk_{server}} = c$. Bob forges m^* and r^* to generate $e^* = h^{m^*-c}(r^*) \oplus d^*$. Then, he chooses a random number v^* and computes ${K_{AS}}^* = T_{v^*}T_n(x)$. After generating a timestamp ${T_A}^*$ and a fake ${s_0}^*$, Bob sends $E_{{K_{AS}}^*}(ID_A \parallel {s_0}^* \parallel m^* \parallel e^* \parallel c \parallel {T_A}^*)$ and $T_{v^*}(x)$ to S. S computes ${K_{SA}}^* = T_nT_{v^*}(x)$ and uses it to decrypt $E_{{K_{AS}}^*}(ID_A \parallel {s_0}^* \parallel m^* \parallel e^* \parallel c \parallel {T_A}^*)$. However, S is unable to find the record $(ID_A, {s_0}^*, m^*)$ in the database due to the incorrect ${s_0}^*$ and m^*. Consequently, S terminates the procedure.

Scenario 2. The attacker Bob is impersonating S

If Bob wants to impersonate S, he forges the private key ${sk^*}_{server}$ of S and computes $(k')^* = k^{{sk^*}_{server}} = R \cdot c^{{sk^*}_{server}}$ after intercepting the message k from A. Then, Bob sends $(k')^*$ to A and A computes $d^* = R^{-1} \cdot (k')^* = c^{{sk^*}_{server}}$. Fortunately, A finds out that $(d^*)^{pk_{server}}$ does not equal c since ${sk^*}_{server}$ is not equal to sk_{server}.

Based on the above analysis, we can conclude that the attacker Bob cannot successfully carry out an impersonation attack in our proposed scheme.

4.5. **Resisting the man-in-the-middle attack.** Although key exchange schemes based on Chebyshev chaotic maps are vulnerable to man-in-the middle attacks just as the Diffie-Hellman scheme is [30], our proposed scheme, which uses Chebyshev chaotic maps as its building blocks, can withstand this kind of attack. In the following, we discuss the attack scenarios in the ownership-transference phase and the payment phase.

4.5.1. *The ownership-transference phase.* **Scenario 1.** The attacker Bob exists between A and C

Assume that Bob intercepts the message $\{T_w(x), T_c\}$ and modifies it as $\{T_{w^*}(x), {T_c}^*\}$ in Step 2. Then, Bob impersonates C to cheat A by sending $\{T_{w^*}(x), T_c\}$ to A. After receiving the message, A computes ${K_{AC}}^* = T_v T_{w^*}(x)$ and $K_{AS} = T_v T_n(x)$. Then, A computes $i^* = E_{K_{AS}}(ID_A \parallel ID_C \parallel s_0 \parallel m \parallel {T_C}^*)$ and $g^* = E_{{K_{AC}}^*}(ID_A \parallel ID_C \parallel s_0 \parallel m \parallel {T_C}^* \parallel i^*)$ and sends g^* and $T_v(x)$ to C. At this time, Bob intercepts $\{g^*, T_v(x)\}$ and acts as A to send it to C. When C receives this message, he computes ${K_{CA}}^* = T_w T_v(x)$ and tries to decrypt g^* by the key ${K_{CA}}^*$. However, he will fail because ${K_{CA}}^*$ is different from ${K_{AC}}^*$, which encrypted g^*. As a result, C will perceive that the received message is invalid and terminate the procedure.

Scenario 2. The attacker Bob exists between C and S

Assume that Bob modifies the intercepted message $\{j, T_w(x), T_v(x)\}$ as $\{j, T_{w^*}(x), T_{v^*}(x)\}$ in Step 8 and then impersonates C by sending it to S. S computes ${K_{SC}}^* = T_n T_{w^*}(x)$ and ${K_{SA}}^* = T_n T_{v^*}(x)$. Fortunately, S cannot decrypt j and i correctly since he or she has no way to obtain the real decryption keys $K_{SC} = T_n T_w(x)$ and $K_{SA} = T_n T_v(x)$. Therefore, the procedure is terminated. Similarly, if Bob modifies the transmitted message in Step 12 and sends it to C, C is not capable of decrypting it and will terminate the procedure.

4.5.2. *The payment phase.* If Bob modifies $T_v(x)$ as $T_{v^*}(x)$ in the message in Step 3 and sends it to S, S cannot decrypt $E_{K_{AS}}(ID_A \parallel s_0 \parallel m \parallel e \parallel c \parallel T_A)$ without knowing the correct decryption key K_{SA}, and the procedure is terminated. In addition, if Bob modifies the message in Step 8 and sends it to A, A is unable to decrypt the received message by the decryption key K_{AS} that he or she has. So, the procedure cannot continue.

According to the analyses of these different scenarios, our proposed scheme can withstand man-in-the middle attacks.

4.6. **Resisting the replay attack.** Both the ownership-transference phase and the payment phase in our proposed scheme can resist the replay attack. We first analyze the scenarios when the attacker Bob replays the message in Steps 2, 4, 8, and 12 in the ownership-transference phase.

If the attacker Bob replays the message $\{T_w(x), T_c\}$ in Step 2, C can immediately detect the invalidity of this message by noting that the timestamp T_C is old. Even if Bob generates a fresh timestamp T_b to replace T_C and sends $\{T_w(x), T_b\}$ to A and then A sends g, which encrypts T_b, and other parameters to C, C can easily perceive that the decrypted T_b is a forged timestamp since T_b is not equal to T_C generated by C in Step 1. Therefore, the procedure is terminated. In addition, if Bob replays the message $\{g, T_v(x)\}$ in Step 4 to deceive C, he will fail because C can decrypt g to retrieve T_C, thereby finding out that T_C is not fresh. Thus, the replay attack cannot be mounted under this scenario. For the same reason, the attempts of replaying messages in Step 8 and 12 cannot succeed.

Due to the fact that the replay attack on messages in Steps 3 and 8 in the payment phase are similar to that in the ownership-transference phase, we only take the ownership-transference phase for example to analyze that our proposed scheme can resist the replay attack.

4.7. **Resisting the insider attack.** The insider attacker of our proposed scheme means that an attacker intrudes on the electronic department store S and inserts invalid records in the database. In our scheme, S stores the record (ID_A, s_0, m) in the database if the customer A buys a digital gift certificate from S. Then, S updates this record when A transfers the ownership of the digital gift certificate or spends the certificate. If the attacker Bob wants to mount the insider attacker, he will generate a record $(ID_B, {s_0}^*, m^*)$ and try to insert it into the database. Fortunately, he will fail because $(ID_B, {s_0}^*, m^*)$ is not a valid record due to the illegality of ${s_0}^*$ and m^*. This ensures that our proposed scheme can resist the insider attack.

5. **Performance Analysis.** In this section, we evaluate the performance of our proposed scheme and compare it with that of the DGC-US scheme [23]. First, we explain the symbols used in the performance analysis. S stands for symmetric encryption/decryption operation, As stands for the asymmetric encryption/decryption operation, Ch represents the Chebyshev chaotic maps operation, H represents the one-way hash function operation, and F is the number of specified hash function operations in the DGC-US scheme. Table 1-3 lists performance comparisons in terms of computational cost between our proposed scheme and the DGC-US scheme in the application phase, the ownership-transference phase, and the payment phase.

To construct the evaluation of performance, we cite two references [33] and [?]. In 1996, Schneier [33] mentioned that a hash function (MD5/SHA) was about 1000 times faster than a asymmetric cryptosystem (RSA-1024) and that one symmetric cryptosystem (DES) was about 100 times faster than one asymmetric cryptosystem. And, in 2013, Lee [?] mentioned that the hash function operation and the Chebyshev chaotic maps operation have the same computational cost. From the performance analysis, it obviously can be inferred that our proposed scheme is more efficient than the DGC-US scheme.

We used C++ language to implement the code of hash function on a Windows 7 workstation with an Intel(R) Core(TM) i7-3770 processor running at 3.40 GHz, 8,192 MB of RAM, and a 7,200 RPM Western Digital WD5000AAKX-08U6AA0 ATA drive. After inputting 64 letters for a 512-bit random string and running 10,000 times, the average time of one hash value was 6.4 ms. According to this result, the execution time for all of the phases are summarized in Table 4.

Table 1. Performance comparison in the application phase

	Participants			
Operations	Our scheme		DGC-US scheme [5]	
	A	S	A	S
S	0	0	0	0
As	1	1	4	4
Ch	0	1	0	0
H	$2(m+1)$	m	$6+F$	4

Table 2. Performance comparison in the ownership-transference phase

Operations	Participants					
	Our scheme			DGC-US scheme [5]		
	A	C	S	A	C	S
S	3	3	4	0	0	0
As	0	1	1	2	5	4
Ch	3	3	2	0	0	0
H	0	$2(m+1)$	m	1	$7+F$	5

Table 3. Performance comparison in the payment phase

Operations	Participants			
	Our scheme		DGC-US scheme [5]	
	A	S	A	S
S	2	2	0	0
As	3	4	0	0
Ch	2	1	0	0
H	$2(m-c)$	$2m+1$	$14+2F$	$8+F$

Table 4. Performance comparison in the payment phase

Phase	A	C	S
Application phase	$\approx 7.69s$	0	$\approx 7.04s$
Ownership-transference phase	$\approx 0.21s$	$\approx 7.90s$	$\approx 7.30s$
Payment phase	$\approx 19.98s$	0	$\approx 27.02s$

6. **Conclusions.** In this paper, we proposed a secure and efficient authentication scheme for digital gift certificates. Our proposed scheme combines the concepts of the Chebyshev chaotic maps and the blind signature, which can enhance efficiency while achieving mutual authentication. In addition, our proposed scheme can offer fundamental functionalities and resist various well-known attacks. Thus, our proposed scheme provides a practical approach to implement convenient and effective electronic payment systems.

Acknowledgment. This research was supported in part by the National Nature Science Foundation of China (grant number: 61202228).

REFERENCES

[1] X. Chen, J. Li, J. Ma, W. Lou, D. S. Wong, New and efficient conditional e-payment systems with transferability,*Future Generation Computer Systems*, article in press, 2013.

[2] A. Ruiz-Martnez, . C. Reverte, A. F. Gmez-Skarmeta, Payment frameworks for the purchase of electronic products and services,*Computer Standards & Interfaces*, vol. 34, no. 1, pp. 80-92, 2012.

[3] S. J. Lin, D. C. Liu, An incentive-based electronic payment scheme for digital content transactions over the Internet,*Journal of Network and Computer Applications*, vol. 32, no. 3, pp. 589-598, 2009.

[4] W. Li, Q. Wen, Q. Su, Z. Jin, An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network,*Computer Communications*, vol. 35, no. 2, pp. 188-195, 2012.

[5] Y. C. Yen, T. C. Wu, N. W. Lo, K. Y. Tsai, A fair-exchange e-payment protocol for digital products with customer unlinkability,*KSII Transactions on Internet and Information Systems*, vol. 6, no. 11, pp. 2956-2979, 2012.

[6] C. Kim, W. Tao, N. Shin, K.S. Kim, An empirical study of customers perceptions of security and trust in e-payment systems,*Electronic Commerce Research and Applications*, vol. 9, no. 1, pp. 84-95, 2010.

[7] S. M. Yen, H. C. Lin, Y. C. Chen, J. J. Hung, J.M. Wu, Paystar: a denomination flexible micropayment scheme,*Information Sciences*, article in press, 2013.

[8] X. Wen, Y. Chen, J. Fang, An inter-bank E-payment protocol based on quantum proxy blind signature,*Quantum Information Processing*, vol. 12, no. 1, pp. 549-558, 2013.

[9] W. H. Tsai, B. Y. Huang, J. Y. Liu, T. S. Tsaur, S. J. Lin, The application of web ATMs in e-payment industry: a case study,*Expert Systems with Applications*, vol. 37, no. 1, pp. 587-597, 2010.

[10] S. N. Vesal, M. Fathian, Efficient and secure credit card payment protocol for mobile devices,*International Journal of Information and Computer Security*, vol. 5, no. 2, pp. 105-114, 2012.

[11] J. J. Hwang, T. C. Yeh, J. B. Li, Securing on-line credit card payments without disclosing privacy information,*Computer Standards & Interfaces*, vol. 25, no. 2, pp. 119-129, 2003.

[12] S. Brands, Untraceable off-line cash in wallet with observers, Advances in Cryptology-CRYPTO93,*Lecture Notes in Computer Science*, vol. 773, pp. 302-318, 1993.

[13] T. Okamoto, K. Ohta, Universal electronic cash,*Advances in CryptologyCRYPTO*, pp. 324-337, 1992.

[14] J. Camenisch, A. Lysyanskaya, An efficient system for non-transferable anonymous credentials with optional anonymity revocation,*Advances in CryptologyEUROCRYPT*, pp. 93-118, 2001.

[15] C. I. Fan, V. S. M. Huang, Y. C. Yu, User efficient recoverable off-line e-cash scheme with fast anonymity revoking,*Mathematical and Computer Modelling*, vol. 58, no. 1-2, pp. 227-237, 2013.

[16] W. S. Juang, RO-cash: an efficient and practical recoverable pre-paid offline e-cash scheme using bilinear pairings,*Journal of Systems and Software*, 83 (2010) 638-645.

[17] S. Das, S. Mohanty, B. Majhi, A secure electronic cash based on a certificateless group signcryption scheme,*Mathematical and Computer Modelling*, vol. 58, no. 1-2, pp. 186-195, 2013.

[18] C. C. Chang, S. C. Chang, J. S. Lee, An on-line electronic check system with mutual authentication,*Computers & Electrical Engineering*, vol. 35, no. 5, pp. 757-763, 2009.

[19] W. K. Chen, Efficient on-line electronic checks,*Applied Mathematics and Computation*, vol. 162, no. 3, pp. 1259-1263, 2005.

[20] W. J. Hsin, L. Harn, Simple certified e-check with a partial PKI solution, *Proceedings of the 43rd Annual Southeast Regional Conference*, Kennesaw, Georgia, USA, vol.2, pp. 185-190, Mar. 2005.

[21] C. W. Chan, C. C. Chang, A scheme for digital gift certificates,*Proceedings of the 2nd International Workshop for Asian Public Key Infrastructures (IWAP 2002)*, Taipei, Taiwan, pp. 136-141, Oct. 2002.

[22] Y. F. Chang, C. C.Chang, Schemes for digital gift certificates with low computation complexity,*Informatica*, vol. 16, no. 4, pp. 503-518, 2005.

[23] D. Xiao, X. Liao, S. Deng, Using time-stamp to improve the security of a chaotic maps-based key agreement protocol,*Information Sciences*, vol. 178, no. 6, pp. 1598-1602, 2008.

[24] D. Xiao, X. Liao, S. Deng, A novel key agreement protocol based on chaotic maps,*Information Sciences*, vol. 177, no. 4, pp. 1136-1142, 2007.

[25] D. Chaum, Blind signature systems,*Proceedings of Advances in Crypto83*, New York, USA, pp. 153, 1983.

[26] D. Chaum, Blinding for unanticipated signatures,*Proceedings of Advances in Eurocrypt87*, Berlin, Germany, pp. 227-233, 1987.

[27] T. Y. Wu, Y. M. Tseng, An efficient user authentication and key exchange protocol for mobile client-server environments,*Computer Networks*, vol. 54, no. 9, pp. 1520-1530, 2010.

[28] T. Y. Wu, Y. M. Tseng, An ID-based mutual authentication and key exchange protocol for low-power mobile devices,*The Computer Journal*, vol. 53 no. 7), pp. 1062-1070, 2010.

[29] W. Diffie, M. Hellman, New directions in cryptography,*IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.

[30] R. L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems.*Communications of the ACM*, vol. 21, no. 2, pp. 120-126, 1978.

[31] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithms,*IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469-472, 1985.

[32] B. Schneier, Applied Cryptography, Protocols, Algorithms, and Source Code in C, 2nd ed., John Wiley and Sons Inc., New York, 1996.

[33] C. C. Lee, A simple key agreement scheme based on chaotic maps for VSAT satellite communications,*International Journal of Satellite Communications and Networking*, vol. 31, no. 4, pp. 177-186, 2013.