Smart Card Based on Hash Function

Zulkharnain and Ali Sher

American University of Ras Al Khaimah,
P.O. Box 10021, RAK, United Arab Emirates
{znain,asher}@aurak.ae

Abstract. With the ongoing new trends adopted for hacking smart card, there is need of updating the technology used in smart card, and particularly the math techniques used in it. The potential for counterfeiting and fraud may pose significant financial risks to institutions issuing payment obligations in these systems and to other participants in these areas. The world-wide use of public key algorithms to maintain electronic cash security, in smart card should be modified. However, public key algorithms use many computation resources, and make electronic cash schemes inefficient for Smart cards to compute. Furthermore, the denominations of electronic cash are fixed, and the electronic cash spending date is hard to record. It will cause the lack of flexibility while applying. This paper proposes a new electronic cash scheme that utilizes trapdoor hash function technology. The proposed scheme not only allows users to apply for desired denominations of electronic cash and to record transaction dates, but also decreases calculation costs. This is also an efficient procedure capable to be used in today's smart card.

Keywords: Smart card, E-Cash, E-Commerce, mobile devices, signature, hash function.

1 Introduction

Smart cards make use of E-cash and since the first electronic cash scheme was presented by D. Chaum [1], multiple electronic cash schemes [2][3][4][5][6] and [7] have been published in the literature. All of these published schemes utilize blind signature to maintain user anonymity. Information on electronic cash must be stored in a database to check for double spending. Yet when there is no expiration date for such information, database storage capacity is depleted quickly which increases maintenance costs. The partially blind signature technique [8][9][10] and [11] has allowed consumers to exchange information with banks, and to store information, such as withdrawal and expiration dates, onto electronic cash, resolving the problems of storage capacity and maintenance costs.

There are three life stages for electronic cash:

1. Consumers pay cash to banks in exchange for electronic cash;
2. Consumers pay for goods using electronic cash;
3. Merchants exchange electronic cash at banks.

A. Abd Manaf et al. (Eds.): ICIEIS 2011, Part IV, CCIS 254, pp. 41–51, 2011.

Consumers might not purchase goods immediately after receiving their electronic cash. Therefore, they believe banks should pay interest during this period, and give rise to technology that stamps consumption dates on electronic cash [12][13] and [14].

The scheme in [13] encodes the consumption dates into hash operations. Consumption dates must be expressed in an encoded format, causing such calculations to be extremely inefficient for mobile devices with low computation abilities [15] and [16]. Banks create consumption dates in [12] which increasing the bank's needs for communication. Electronic cash in [14] is embedded with the public key of the consumption date, and it allows consumers to create a signature indicating the date of consumption. This method, however, is incompatible for devices with low computing abilities, especially as consumers purchase more.

This research proposes a new electronic cash scheme with a trapdoor hash function that provides a consumption date stamp technology without increasing communication. The proposed scheme simultaneously decreases the need for calculating transactions while consumers are shopping. It only requires one integer multiplication and two integer additions, and makes the technology compatible for mobile devices with low computing abilities.

Section 2 provides a summary of the trapdoor hash function. Section 3 describes the requirements that electronic cash systems must fulfill. Sections 4 and 5 discuss the security and efficacy of electronic cash.

2 Review of Trapdoor Hash Functions

Hash functions are commonly applied to digital signature techniques, and digital signature algorithms can be broken down into three phases: signing key generation, signing document (generating signature), and signature verification. Generally, the procedures for signing document are as follows: Hash functions extract the abstract of the document that is needed to be signed, after which a digital signature algorithm signs the abstract.

Collision-resistance is one of the main features of traditional hash functions. For Chameleon functions [17] or trapdoor hash functions [18][19] and [20], the feature of collision-resistance is optional; the owner of a trapdoor key can easily find other collided pre-images and produce the same hash value. For instance, assuming TH (□) represents trapdoor hash function and the hash value $v = TH(h_1)$, after knowing h_1, the owner of a trapdoor key can then calculate h_2; hence, $v = TH(h_2) = TH(h_1)$.

Computing the value of Chameleon functions online requires a multiplication and modulo operation. Online computation means the amount of computations are required once the target message is determined. In the literature [15], only one modulo operation is required for this computation. In the literature [18] and [19], the computational requirement is further reduced to only one integer multiplication and addition. It is suitable for mobile devices with limited computational resources. The techniques mentioned in literature [18] and [19] are as follows:

Let p, q, t, P and Q be prime numbers, the compound number n is the product of P and Q; that is, $n = P \cdot Q$, $P = 2 \cdot p \cdot t + 1$, $Q = 2 \cdot q + 1$. $|P|$, $|Q|$, and $|p|$ represent the encoded bit length of P, Q, and p. Their lengths can be chosen as follows: $|P| = |Q| = 512$, $|p| = l = 160$.

The order of $g \in Z_n^*$ is p. Randomly selecting $x \in_R \{0, 1\}^l$; Calculating $y = g^x \bmod n$. The trapdoor key is $TK = x$, and the public key is $HK = (g, n, y)$.

If a message $m_1 \in_R \{0, 1\}^l$, the *hash operation* is to compute the hash value of the message m_1. The processes are as follows.

1. Random Selection: $r_1 \in_R \{0, 1\}^{2 \cdot l + k}$,
2. Calculation of the hash value: $v = TH_{HK}(m_1, r_1)$, i.e., $v = g^{r_1} \cdot y^{m_1} \bmod n$.

After determining r_1, then the hash value of message m_1 is $v = TH_{HK}(m_1, r_1)$. The owner of the trapdoor key can begin a *trapdoor operation* to obtain m_2 and r_2 such that $v = TH_{HK}(m_1, r_1) = TH_{HK}(m_2, r_2)$. The detailed processes of *trapdoor operation* are shown as follows:

1. Determining message: $m_2 \in \{0, 1\}^l$,
2. Calculating r_2, i.e., $TH_{TK}(m_2) = r_2 = r_1 + (m_1 - m_2) \cdot x$.

Although integer arithmetic is used for calculating r_2, the confidential information $(m_1 - m_2) \cdot x$ is still properly hidden behind the random number r_1 because $|r_1| = 2 \cdot l + k$, often $k = 80$. Similar information hiding techniques can also be seen in the literature [21][22] and [23].

3 The Proposed Electronic Cash Scheme

This research proposes an electronic cash scheme built with an RSA blind signature technique and trapdoor hash function allowing banks to file relevant information, such as: electronic cash denominations, withdrawal dates, expiration dates, and consumption dates.

Electronic cash schemes are composed of three roles: consumers (apply for and spend electronic cash), merchants (provide services and goods to consumers), and banks (authorize, distribute, and manage electronic cash), as depicted in Figure 1.

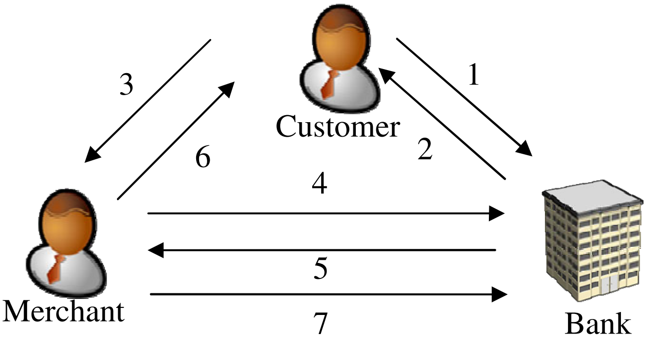


Fig. 1. Electronic cash scheme

Electronic cash scheme procedures:

1. Consumers apply for electronic cash from banks
2. Banks distribute electronic cash to consumers

3. Consumers purchase with electronic cash
4. Merchants check with bank for issues of double spending
5. Banks confirm validity of electronic cash
6. Merchants confirm that transaction was successful, pay consumers interest for the duration between application and spending, and provide goods and services
7. Merchants deposit electronic cash into banks, which pay merchants the amount of denomination and interest for the duration between application and deposit dates. (To incorporate the concept of fair electronic payment [24] and [25] into Figure 1, banks should confirm that the consumer have received goods and services before paying the money to merchants.)

3.1 Definition of Notations

The parameters and symbols of electronic cash scheme can be divided into: systems, banks, and consumers, described as follows:

The parameters and symbols for systems:

1. *EXD*: Expiration date of electronic cash
2. *APD*: Application date of electronic cash
3. *SPD*: Electronic cash purchase date
4. $h(\cdot)$: collision-free one-way hash function
5. $||$: bit concentration symbol
6. k, l: parameters of confidentiality in accordance with the development level of security, for example: $k = 80$, $l = 160$

The parameters and symbols for banks:

1. p, q, t, P and Q: both are prime numbers, with P and Q the same code length, and $P = 2 \cdot p \cdot t + 1$, $Q = 2 \cdot q + 1$. $|P| = |Q| = 512$, $|p| = l$
2. product of two large prime numbers, such as $n = P \cdot Q$
3. g : $g \in Z_n^*$ with an order of p
4. (d_{sig}, e_{ver}): bank utilizes the RSA cryptosystems to create keys of signing and verifying, $e_{ver} \in_R Z_n$, $\gcd(e_{ver}, \phi(n)) = 1$, $\phi(n) = (P - 1)(Q - 1)$, $e_{ver} \cdot d_{sig} = 1 \bmod \phi(n)$; e_{ver} is public key for verifying signatures, d_{sig} is secret key for signing documents.

The parameters and symbols for consumers:

1. x: consumer chooses a secret key for his trapdoor hash function, for example: $TK = x$, and $x \in_R \{0, 1\}^l$
2. *HK*: public key of consumer's trapdoor hash function, for example: $HK = (g, n, y = g^x \bmod n)$
3. m_1, r_1: consumers choose two messages randomly, for example: $m_1 \in_R \{0, 1\}^l$, $r_1 \in_R \{0, 1\}^{2 \cdot l + k}$

3.2 Stages of the Proposed Electronic Cash Scheme

There are four stages (phases) of the electronic cash schemes: 1. apply for electronic cash, 2. exterminate blind factors, i.e. receive electronic cash, 3. spend electronic cash, 4. deposit electronic cash into banks. Detailed descriptions of each step are as follows:

(1) Applying for electronic cash

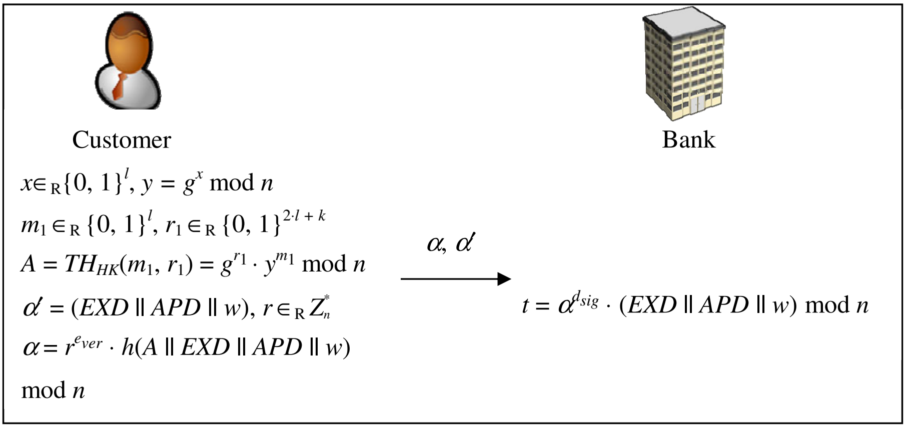
Consumers apply for w of electronic cash by first discussing relevant dates with bank and performing the following calculations once consensus has been reached:

1. create a random trapdoor key $x \in_R \{0, 1\}^l$, calculate $y = g^x \bmod n$
2. create a random message $m_1 \in_R \{0, 1\}^l$ and a random number $r_1 \in_R \{0, 1\}^{2 \cdot l + k}$
3. execute *hash operation*, $A = TH_{HK}(m_1, r_1) = g^{r_1} \cdot y^{m_1} \bmod n$
4. application date and denomination $\alpha' = (EXD \parallel APD \parallel w)$
5. select a blind factor $r \in_R Z_n^*$
6. use a one way hash function to generate messages $h(A \parallel EXD \parallel APD \parallel w\ )$
7. calculate $\alpha = r^{e_{ver}} \cdot h(A \parallel EXD \parallel APD \parallel w) \bmod n$

Consumer sends messages to bank after the calculation is complete, and store m_1, r_1, x for backup. The bank will verify the accuracy of the dates upon receiving the information and sign with private keys. The calculations are as follows:

1. verify date based on information $(EXD \parallel APD \parallel w)$
2. create blind signature $t = \alpha^{d_{sig}} \cdot (EXD \parallel APD \parallel w) \bmod n$

The bank resends the blind signature to consumers while deducting w amount of electronic cash from the consumers' accounts. The process of applying for electronic cash is shown in Figure 2, and is the first step in the electronic cash system procedures (shown in Figure 1).



$x \in_R \{0, 1\}^l$, $y = g^x \bmod n$

$m_1 \in_R \{0, 1\}^l$, $r_1 \in_R \{0, 1\}^{2 \cdot l + k}$

$A = TH_{HK}(m_1, r_1) = g^{r_1} \cdot y^{m_1} \bmod n$

$\alpha' = (EXD \parallel APD \parallel w)$, $r \in_R Z_n^*$

$\alpha = r^{e_{ver}} \cdot h(A \parallel EXD \parallel APD \parallel w)$ mod n

$t = \alpha^{d_{sig}} \cdot (EXD \parallel APD \parallel w) \bmod n$

Fig. 2. Stages of applying for electronic cash

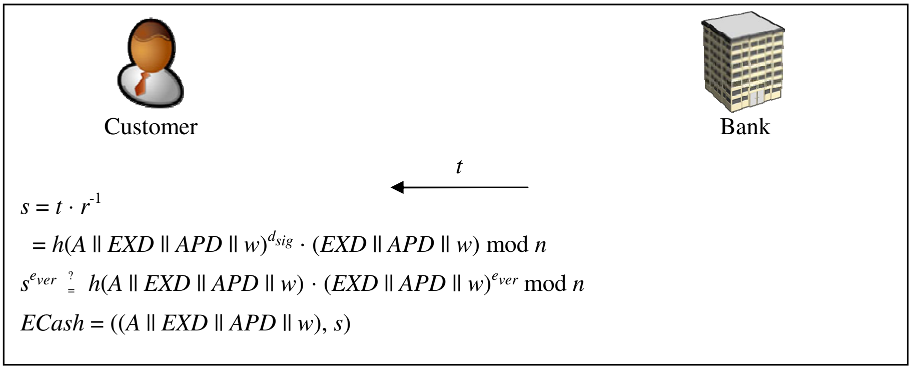
(2) Eliminate blind factor (receive electronic cash)

After users receive the blind signature t from banks, they can eliminate blind factors and receive electronic cash by performing the following calculations:

1. Eliminate blind factor, namely calculate $s = t \cdot r^{-1} = h(A \parallel EXD \parallel APD \parallel w)^{d_{sig}} \cdot (EXD \parallel APD \parallel w) \bmod n$.

2. Confirm signature, that is, verify $s^{e_{ver}} \overset{?}{=} h(A \,\|\, EXD \,\|\, APD \,\|\, w) \cdot (EXD \,\|\, APD \,\|\, w)^{e_{ver}} \bmod n$. Assuming the signature is correct, $((A \,\|\, EXD \,\|\, APD \,\|\, w), s)$ represents w denomination of electronic cash.

The process for eliminating the blind factor is depicted in Figure 3, and the step 2 of Figure 1 of electronic cash system is completed here.



$s = t \cdot r^{-1}$

$= h(A \,\|\, EXD \,\|\, APD \,\|\, w)^{d_{sig}} \cdot (EXD \,\|\, APD \,\|\, w) \bmod n$

$s^{e_{ver}} \overset{?}{=} h(A \,\|\, EXD \,\|\, APD \,\|\, w) \cdot (EXD \,\|\, APD \,\|\, w)^{e_{ver}} \bmod n$

$ECash = ((A \,\|\, EXD \,\|\, APD \,\|\, w), s)$

Fig. 3. Eliminate blind factor (receive electronic cash)

(3) Consumption (spend electronic cash)

When a customer wishes to purchase an item, he/she performs the following steps to complete transaction.

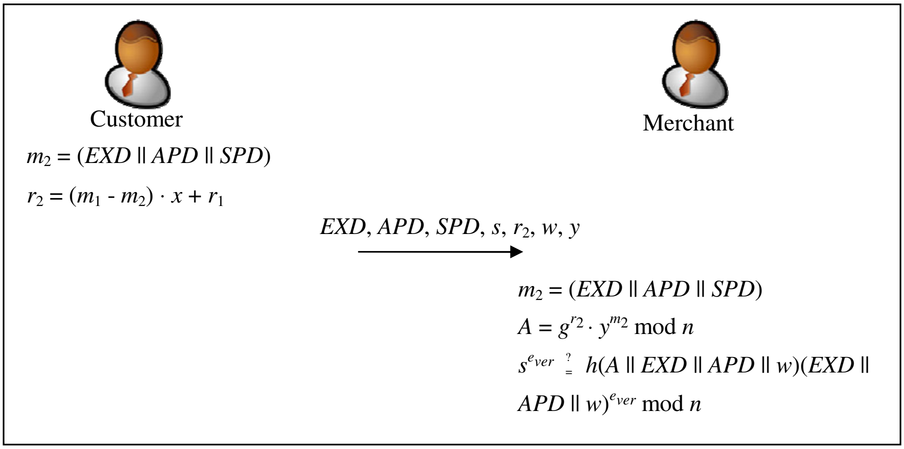
1. consumers sign on documents $m_2 = (EXD \,\|\, APD \,\|\, SPD)$; contents include expiration, application, and consumption dates of electronic cash
2. consumers withdraw information stored in database (m_1, r_1, x)
3. use trapdoor key x and carry out *trapdoor operation*, $TH_{TK}(m_2) = r_2 = (m_1 - m_2) \cdot x + r_1$

Additional information $(EXD, APD, SPD, s, r_2, w, y)$ is sent to merchants after calculation. Purchasing procedures are described in Figure 4, and the process of the third step in Figure 1 is completed here.

(4) Depositing

Merchants perform the following calculations upon receiving electronic cash from consumers:

1. decision document $m_2 = (EXD \,\|\, APD \,\|\, SPD)$
2. calculate $A = TH_{HK}(m_2, r_2) = g^{r_2} \cdot y^{m_2} \bmod n$
3. verify $s^{e_{ver}} \overset{?}{=} h(A \,\|\, EXD \,\|\, APD \,\|\, w)(EXD \,\|\, APD \,\|\, w)^{e_{ver}} \bmod n$



$m_2 = (EXD \| APD \| SPD)$

$r_2 = (m_1 - m_2) \cdot x + r_1$

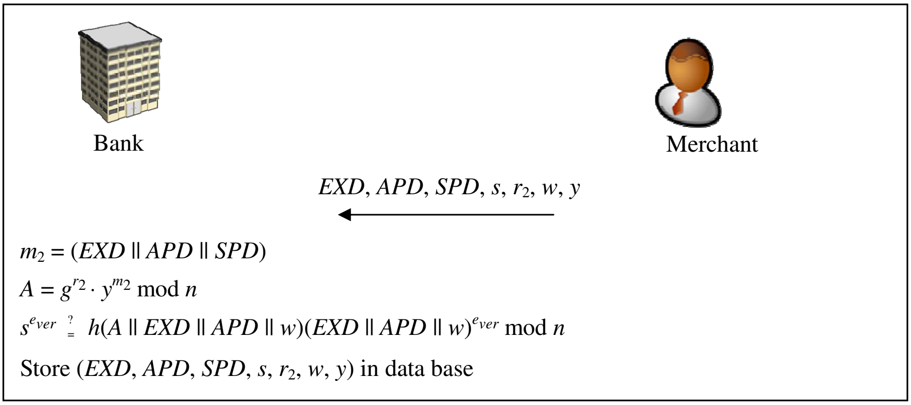
$$\xrightarrow{EXD,\ APD,\ SPD,\ s,\ r_2,\ w,\ y}$$

$m_2 = (EXD \| APD \| SPD)$

$A = g^{r_2} \cdot y^{m_2} \bmod n$

$s^{e_{ver}} \overset{?}{=} h(A \| EXD \| APD \| w)(EXD \| APD \| w)^{e_{ver}} \bmod n$

Fig. 4. Consumption (spend electronic cash)



$$\xleftarrow{EXD,\ APD,\ SPD,\ s,\ r_2,\ w,\ y}$$

$m_2 = (EXD \| APD \| SPD)$

$A = g^{r_2} \cdot y^{m_2} \bmod n$

$s^{e_{ver}} \overset{?}{=} h(A \| EXD \| APD \| w)(EXD \| APD \| w)^{e_{ver}} \bmod n$

Store (EXD, APD, SPD, s, r_2, w, y) in data base

Fig. 5. Depositing

Assuming that the above calculations are accurate, merchants may request banks to confirm that the electronic cash is not involved in double spending. If the electronic cash is verified, merchants may then accept the transaction and pay interest to the consumers while the bank records the information into their databases to prevent double spending. The bank then directs the *w* amount of cash that is spent by consumers and deposits interest into the merchants' accounts.

The depositing stage is depicted in Figure 5, steps 4-7 of the electronic cash system in Figure 1 are completed in this stage.

4 Security Analysis

This section discusses the nature of electronic cash regarding its accuracy, uniqueness, credibility, anonymity, double spending, and fraud interest.

Accuracy: Both the electronic cash that banks distributed to consumers and the electronic cash with consumption dates can be verified by any entity. The bank's public key is sufficient to verify the electronic cash.

Unforgeability: Due to the factorization is a mathematical assumption, blind signatures are secure. Forgers would have to know the bank's private key in order to forge successfully a signature.

Consumers will use their own private keys to calculate hash collision values. If anyone attempted to change the relevant dates, he or she would need to know the consumer's private keys to reach the same hash collision values.

Undeniability: Banks must give private signatures before electric signatures can be created and provided to consumers. Consumers then use the public keys provided by banks to verify the cash so banks cannot deny issuing electronic cash.

Consumers use private trapdoor key to calculate the pre-imaged of a pre-determined hash value, and transfer relevant information and electronic cash to merchants for the latter to verify. Upon receiving the necessary information, merchants can calculate the accurate pre-determined hash value so consumers cannot deny ever paying the electronic cash.

Anonymity and un-traceability: When consumers are applying for electronic cash, their information is mixed with blind factors before being submitted to banks. Once banks return the information with bank-authorized signatures, consumers then eliminate the blind factors to receive the actual message. No one is able to track the information without knowing which blind factor consumers randomly chose, making such process safe and anonymous.

For consumers who request special amounts of electronic cash, they are able to arrange for specific denominations so that their transactions remain anonymous and untraceable.

Double spending: In our proposal, the bank serves as a trustworthy third party. All information stored in the bank's database is legal and credible. The records of spending electronic cash are stored in the bank and will only be deleted when they have expired.

Upon receiving electronic cash, merchants first verify the cash, and then connect to the bank to confirm whether this electronic cash is stored in the bank database. It is easily preventing cases of double spending.

Fraud interest: when applying for electronic cash, relevant dates are encoded within the cash and are signed by banks. Due to consumers and merchants have conflicting interests, the two parties cannot conspire together to gain fraud interest.

5 Performance Analysis

Electronic cash systems [12] require banks to mark consumption dates; therefore, communication costs have increased. Table 1 can only analyze electronic cash systems [14]. Major roles involved in the electronic cash system include banks, merchants, and consumers. The life cycle of electronic cash includes receiving, spending, and depositing. Table 1 compares the amounts of electronic cash for each role during the cash's life cycle.

In Table 1, the difference between the proposed scheme and Juang's is in the calculation of the consumption stage. Juang's scheme requires one modular exponentiation operation (Exp), two modular multiplications (MM), one modular inverse operation (Inv), and one hash function operation (H); whereas this study only requires one integer multiplication (IM). According to an experiment recorded in [19], the calculation time for MM is roughly 1.4 times longer than IM. Assuming that the index is 160 bits, and a modular exponentiation operation requires an average of 240 modular multiplications, Juang's scheme's calculation time is roughly 336 (1.4 * 240) times longer than the proposed scheme. Although banks and merchants tend to have enough calculation resources, most consumers do not, especially those consumers with mobile devices or personal digital assistants (PDAs), make the proposed scheme more practical and user-friendly.

If the date and coding are not taken into account, the electronic cash in Juang's scheme (r, s, r', s') has a code length of 1504 (1024, 160, 160, 160) bits. The electronic cash in this study (s, r_2, y) has a code length of 2448 (1024, 400, 1024) bits, which is an additional 944 bits. If the transmission rate is originally 100 kbps, then the rate will be reduced by 10 ms (ms = 10^{-3} second). However, most mobile devices run on memory capacities of many hundreds of K bytes (1 byte = 8 bits); therefore, 944 bits is an insignificant increase.

Table 1. Comparison chart between proposed scheme and Juang's scheme [14]

	Consumer		Bank		Merchant	
	Juang's scheme [14]	Our scheme	Juang's scheme [14]	Our scheme	Juang's scheme [14]	Our scheme
Receiving electronic cash	6 Exp 6 MM 2 H 1 Inv	6 Exp 4 MM 1 H 1 Inv	1 Exp 1 MM	1 Exp 1 MM	0	0
Consuming	1 Exp 2 MM 1 H 1 Inv	**1 IM**			4 Exp 5 MM 1 H 1 Inv	4 Exp 2 MM 1 H
Depositing			4 Exp 5 MM 1 H 1 Inv	4 Exp 2 MM 1 H		

6 Conclusion

The smart card-electronic cash scheme proposed in this paper utilizes a blind signature to maintain consumer anonymity while spending electronic cash. The proposed scheme also allows consumers to request their own electronic cash denominations, which not only solves the problem of double spending, but also provides complete date and interest information, thus making the smart card system secure. Considering that smart cards are gaining popularity, the design also incorporates a hash function, which can help decreasing consumer's needs for calculation. Such a system is compatible for smart cards with low computing abilities and thus making electronic services increasingly popular.

References

1. Chaum, D.: Blind Signatures for Untraceable Payments. In: McCurley, K.S., Ziegler, C.D. (eds.) Advances in Cryptology 1981 - 1997. LNCS, vol. 1440, pp. 199–203. Springer, Heidelberg (1999)
2. Brands, S.: Untraceable Off-Line Cash in Wallets with Observers. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 302–318. Springer, Heidelberg (1994)
3. Camenisch, J., Piveteau, J.M., Stadler, M.: An Efficient Fair Payment System Protecting Privacy. In: Gollmann, D. (ed.) ESORICS 1994. LNCS, vol. 875, pp. 207–215. Springer, Heidelberg (1994)
4. Chaum, D., Fiat, A., Naor, M.: Untraceable Electronic Cash. In: Goldwasser, S. (ed.) CRYPTO 1988. LNCS, vol. 403, pp. 319–327. Springer, Heidelberg (1990)
5. Chaum, D., Pedersen, T.P.: Wallet Databases with Observers. In: Brickell, E.F. (ed.) CRYPTO 1992. LNCS, vol. 740, pp. 89–105. Springer, Heidelberg (1993)
6. Ferguson, N.: Single Term Off-Line Coins. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 318–328. Springer, Heidelberg (1994)
7. Okamoto, T., Ohta, K.: Universal Electronic Cash. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 324–337. Springer, Heidelberg (1992)
8. He, D., Chen, J., Hu, J.: An ID-based client authentication with key agreement protocol for mobile client-server environment on ECC with provable security. Information Fusion (2011)
9. Abe, M., Okamoto, T.: Provably Secure Partially Blind Signatures. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 271–286. Springer, Heidelberg (2000)
10. Fan, C.I., Lei, C.L.: Low-computation partially blind signatures for electronic cash, IEICE Transactions on Fundamentals of Electronics. Communications and Computer Sciences E81-A (5), 940–949 (1998)
11. Yang, F.Y., Jan, J.K.: A Secure Scheme for Restrictive Partially Blind Signatures. In: The Sixth International Conference on Information Integration and Web-based Applications & Services IIWAS 2004, Jakarta Indonesia, pp. 541–548 (2004)
12. Chang, C.C., Lai, Y.P.: A flexible Date-attachment Scheme on E-cash. Computers & Security 22(2), 160–166 (2003)
13. Fan, C.I., Chen, W.K., Yeh, Y.S.: Date Attachable Electronic Cash. Computer Communications 23(4), 425–428 (2000)
14. Juang, W.S.: D-Cash: A Flexible Pre-paid E-cash Scheme for Date-attachment. Electronic Commerce Research and Applications 6(1), 74–80 (2007)

15. Keating, W.S.: Performance analysis of AES candidates on the 6805 CPU core, AES Round 2 public comment, April 15 (1999)
16. Yang, C.H.: Performance Evaluation of AES/DES/Camellia on the 6805 and H8/300 CPUs. In: 2001 Symposium on Cryptography and Information Security (SCIS 2001), pp. 727–730 (2001)
17. Krawczyk, H., Rabin, T.: Chameleon signatures. In: Symposium on Network and Distributed Systems Security (NDSS 2000), pp. 143–154 (2000)
18. Shamir, A., Tauman, Y.: Improved online/Offline signature schemes. In: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 355–367. Springer, Heidelberg (2001)
19. Yang, F.Y.: Improvement on a trapdoor hash function. International Journal of Network Security 9(1), 17–21 (2009)
20. Yang, F.Y.: Efficient trapdoor hash function for digital signatures. Chaoyang Journal 12, 351–357 (2007)
21. Okamoto, T., Tada, M., Miyaji, A.: Efficient 'on the Fly' Signature Schemes Based on Integer Factoring. In: Pandu Rangan, C., Ding, C. (eds.) INDOCRYPT 2001. LNCS, vol. 2247, pp. 275–286. Springer, Heidelberg (2001)
22. Pointcheval, D.: The Composite Discrete Logarithm and Secure Authentication. In: Imai, H., Zheng, Y. (eds.) PKC 2000. LNCS, vol. 1751, pp. 113–128. Springer, Heidelberg (2000)
23. Poupard, D., Stern, J.: On the fly signatures based on factoring. In: Proceedings of the 6th ACM Conference on computer and communications security (CCS), pp. 48–57 (1999)
24. Asokan, N., Shoup, V., Waidner, M.: Optimistic fair exchange of digital signatures. IEEE Journal on Selected Areas in Communications 18(4), 593–610 (2000)
25. Yang, J.H., Chang, C.C.: An efficient fair electronic payment system based upon non-signature authenticated encryption scheme. International Journal of Innovative Computing, Information and Control 5(11A), 3861–3874 (2009)