Mobile Banking Payment System

Fuw-Yi Yang[*] Zhen-Wei Liu, Su-Hui Chiu
Chaoyang University of Technology
Wufong District, Taichung, Taiwan
{yangfy,s9727628,suhui}@cyut.edu.tw

Abstract

With the rapid development of the Internet and information technology, electronic commerce system is flourishing. In electronic commerce, customers use electronic money (e-money) instead of cash. E-money in wide use includes electronic cash, electronic checks, electronic coupons, and so on. E-money is fixed at face-value and is backed by banks. The fact that e-money is not issued in the actual amount to be used, it is somewhat inconvenient for customers to use. In addition, security requirements consume huge amounts of computing resources. Limited computing capability makes it is difficult to engage in electronic commerce with mobile devices. This study presents a new payment system, wherein customers are not required to purchase e-money of a fixed value in advance. The amount of every transaction is deducted directly from the customer's bank account, eliminating the inconvenience of fixed value currency, and reducing online computing requirements. Through the application of trapdoor hash functions to streamline computational processes, the system can be used with mobile devices. We have named the system, *mobile banking payment system*.

Keywords: Mobile banking payment system, E-money, E-cash, Trapdoor hash functions

1 Introduction

Following the development and immense popularity of the Internet, the use of e-commerce related online services has become increasingly common. In recent years, Internet developers have introduced many forms of e-money including e-cash, e-check, e-coupon, and others. For example, e-cash can be divided into e-cash (fixed amount) payment system and micropayment system. In 1982, D. Chaum introduced an e-cash system based on blind signatures [1]. In this system, customers had to purchase e-cash via the Internet or directly from the bank before they could buy anything, and banks issued e-cash only in fixed denominations. Due to these restrictions, e-cash of this sort lacked the flexibility demanded by most users. Subsequently, e-money developers worked to develop the *e-cash transaction protocol* [2, 3, 4, 5]. These transaction protocols (agreed upon standards) consume large computational resources through the use of modular exponential operations, imposing a heavy load on the system and driving up the costs of computations and communications. In 1999, Dai and Lo introduced the micropayment system [6], wherein customers could register with a bank via the Internet, and cash deposits could be transferred to e-cash with the approval of the bank. Using the systems of micropayment, the amounts of transaction is limited to a small sum of money. PayPal [7] defines those transactions of less than 12 USD as micropayments; Visa [8] defines transactions under 20 USD as micropayments; other micropayment systems have different limitation on the amount of transactions [9, 10, 11, 12]. However, the value of transactions in this system was rather low, making it unsuitable for high value purchasing behavior.

By promoting the convenience of credit cards, financial institutions introduced the concept of post-payment. Unfortunately, customers often overspent and ended up getting themselves in debt when using installment payments. This could lead to a vicious circle going beyond the isolated financial difficulties

[*] Fuw-Yi Yang is the corresponding author.

of individuals, to plant the seeds of a global financial crisis. For these reasons, the current e-cash transaction system needs to be discussed.

To eliminate the shortcomings of the existing e-cash paradigm, this study introduces a mobile banking payment system based on trapdoor hash functions. This system would freeze or deduct the amount of the transaction from customers' bank accounts. Customers would be unable to exceed the balance of his/her account, thus avoiding the possibility of overspending. To integrate trapdoor hash function into the system, the computation on the customer side would require only the use of integer multiplication and addition; hence, it would reduce computing costs, and make the system more suitable for application with mobile devices and enable mobile shopping. The rest of the study is organized as follows. Section 2 reviews the trapdoor hash functions, Section 3 presents the proposed scheme, Section 4 analyzes the security of the proposed scheme, Section 5 discusses the advantages of this study, and Section 6 concludes the study.

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 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 [13] or trapdoor hash functions [14, 15, 16, 17], 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 required once the target message is determined. In the literature [14], only one modulo operation is required for this computation. In the literature [15, 16], the computational requirement is further reduced to only one integer multiplication and addition, suitable for mobile devices with limited computational resources. The techniques mentioned in literature [15, 16] 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, $n = PQ, P = 2pt+1, Q = 2q+1$. $|P|, |Q|$, 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: $r_1 \in_R \{0,1\}^{2l+k}$,
2. Calculation of the hash value: $v = TH_{HK}(m_1,r_1)$, i.e., $v = g^{r_1}y^{m_1} \bmod n$.

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 $r_2, i.e., TH_{TK}(m_2) = r_2 = r_1 + (m_1 - m_2)x$.

Although integer arithmetic is used for calculating r_2, the confidential information $(m_1 - m_2)x$ is still properly hidden behind the random number r_1 because $|r_1| = 2l + k$, often $k = 80$. Similar information hiding techniques can also be seen in the literature [18, 19, 20].

3 The proposed scheme

This study introduces a mobile banking payment system based on trapdoor hash functions. If a customer wants to purchase products via the Internet, after receiving the purchase order, the merchant would sign the order and send the valid transaction information back to the customer. When payment is required, the customer would use an electronic voucher issued by the bank to request payment from the bank. After receiving the voucher from the customer, the bank would have to confirm whether the customer's balance, electronic payment voucher and order information were valid. If all checked out, the bank would temporarily freeze the amount of the transaction in the account and send the transaction information to the merchant, verifying that the customer has sufficient funds to purchase the item. The merchant would then accept the order and deliver the item to the customer. After receiving confirmation that the customer has received the product, the bank would then deduct the total amount of the purchase from the customer's account and deposit it into the merchant's account.

The mobile banking payment system consists of three entities: customer, merchant, and bank. This system can be illustrated as in Figure 1.

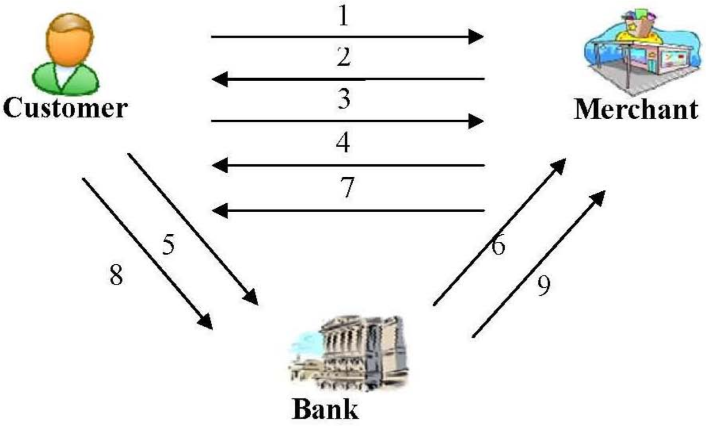


Figure 1: The Mobile Banking Payment System

The procedural steps in Figure 1 are as follows:

1. Customer requests merchant to provide the price of the item.
2. Merchant provides the price of the item.
3. Customer accepts the price and places the order.
4. Merchant confirms the details of the order and signs to verify the order.
5. Customer signs request for the bank to pay the merchant; the bank freezes the purchase amount in the customer's account.

6. Bank notifies the merchant that the customer is prepared to pay for the goods.
7. Merchant delivers item to the customer.
8. Customer confirms with the bank that they have received the item; bank may proceed with the payment.
9. The bank deposits the payment into the merchant's account.

3.1 Parameters and Symbols

The parameters and symbols for the mobile banking payment system can be divided into four sections: public system parameters, bank parameters, customer parameters, and merchant parameters. They are described as follows:

Public System Parameters

- $||$: It stands for string concatenation symbol.
- $H()$: This is a collision-resistant one-way hash function and is defined by $H() : \{0,1\}^* \to Z_n^*$.
- k, l: The confidential parameters classified in accordance with level of security. For instance, $k = 80, l = 160$.

Bank Parameters

- ID_{BK}: It signifies the identity of a bank.
- P, Q, p, q, t: All are large prime numbers and P, Q have the same encoded bit length. $P = 2pt + 1, Q = 2q + 1$.
- n: Product of two large prime numbers, for instance, $n = PQ$.
- g:$g \in_R Z_n^*$ is with order p, the encoded bit length of p is l, that is, $|p| = l$.
- d, e: Bank's signing key and public verification key, $e \in_R Z_n, de = 1 \bmod (P-1)(Q-1)$.
- $Enc_{BK}(), Dec_{BK}()$: They are the encryption and decryption functions of the bank, respectively.

Customer Parameters

- ID_i: It signifies the identity of a customer.
- x_1: Trapdoor key selected by the customer, for instance, $TK = x_1$, and $x_1 \in_R \{0,1\}^l$.
- HK: Public key corresponding to trapdoor key, for instance, $HK = (g, n, y_1 = g^{x_1} \bmod n)$.
- m_1, r_1: Random message and random number selected by the customer, for instance, $m_1 \in_R \{0,1\}^l, r_1 \in_R \{0,1\}^{2l+k}$.

Merchant Parameters

- ID_M: It signifies the identity of merchant.
- SPM: It stands for the price of the purchased item.

- SPN: It stands for the name of the purchased item.
- NOW: It stands for the transaction information includes transaction time, date, and serial number of the purchase order.
- $Sig_M()$: Merchant uses signing key to sign the message and generates signature.
- $Ver_M()$: Customer or bank verifies the signature with the merchant's public verification key.$Ver_M()$ returns either string "$true$" or "$false$".
- $Enc_M(), Dec_M()$: Encryption/decryption functions of the merchant.

3.2 Procedures of the mobile banking payment system

The parameters of the mobile banking payment system are illustrated in Section 3.1. The process of this system can be divided into four phases: customer registration phase, purchasing phase, payment phase, and bank deduction phase. Figures 2, 3, 4, and 5 show the four phases. The followings describe the details of each phase.

(1) Customer Registration phase In this phase, the customer has to open a bank account, select a random number $x_1 \in_R \{0,1\}^l$,and calculate $y_1 = g^{x_1} \bmod n$. The public key for the trapdoor hash function is $HK = (g,n,y_1)$, trapdoor key is x_1. To apply for an electronic payment voucher, the customer calculates the trapdoor hash value and sends his or her identity message ID_i and the hash value to the bank. The process is as follows.

1. Randomly generate message $m_1 \in_R \{0,1\}^l$ and number $r_1 \in_R \{0,1\}^{2l+k}$.
2. Calculate the trapdoor hash value $A = TH_{HK}(m_1,r_1) = g^{r_1} y_1^{m_1} \bmod n$.
3. Send identity message ID_i, and the trapdoor hash value A to the bank.

The customer stores the random pair (m_1,r_1). This pair will enable customer to calculate another pair (m_2,r_2) in the payment phase such that both pairs generate the same trapdoor hash value. After receiving the registration message, the bank uses its signing key d to sign the message and generate an electronic payment voucher σ. The bank then sends the e-payment voucher σ to the customer for calculation. The process is as follows.

1. Calculate $\sigma = H(ID_i,A)^d \bmod n$.
2. Send the electronic payment voucher σ to the customer.

After receiving σ, the customer verifies if σ is a valid signature on the registration message signed by the bank; that is, check $\sigma^e \overset{?}{=} H(ID_i,A) \bmod n$. Figure 2 shows the process of the customer registration phase.

(2) Customer Purchasing phase When a customer wishes to purchase an item, he/she first requests the price. After learning the price, the customer creates a purchase order with information including: identity message, e-payment voucher, trapdoor hash value, and the price of the item. The customer encrypts the order, that is, $\alpha = Enc_M(\sigma,ID_i,A,SPN,SPM)$, and sends the encrypted order α to the merchant. After receiving the encrypted order α, the merchant uses his decryption key to obtain the order information and verify the validity of the e-payment voucher. The merchant then uses his signing key to sign the customer's order. The process is as follows.

1. Use the decryption key to decrypt the message $(\sigma,ID_i,A,SPN,SPM) = Dec_M(\alpha)$.

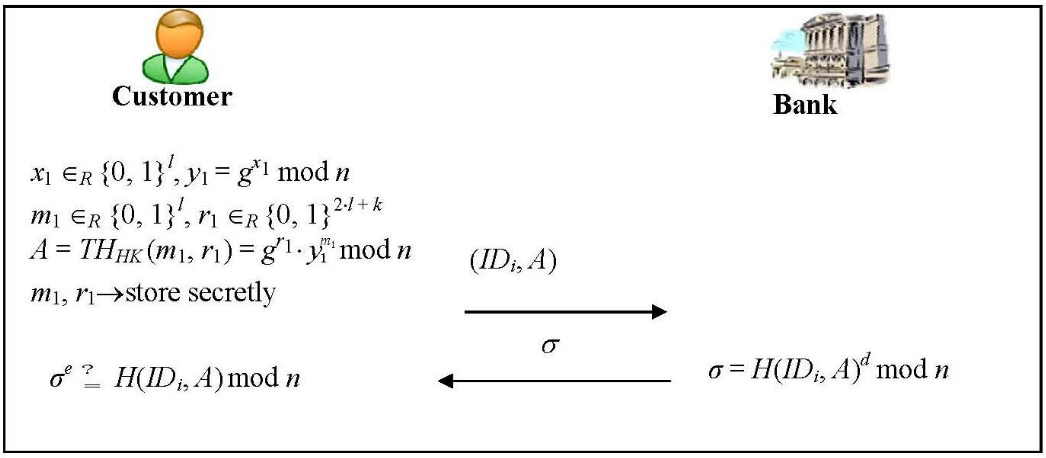

Customer
Bank
$x_1 \in_R \{0,1\}^l, y_1 = g^{x_1} \bmod n$
$m_1 \in_R \{0,1\}^l, r_1 \in_R \{0,1\}^{2 \cdot l+k}$
$A = TH_{HK}(m_1, r_1) = g^{r_1} \cdot y_1^{m_1} \bmod n$
$m_1, r_1 \rightarrow$ store secretly
(ID_i, A)
σ
$\sigma^e \overset{?}{=} H(ID_i, A) \bmod n$
$\sigma = H(ID_i, A)^d \bmod n$

Figure 2: The Customer Registration Phase

2. The merchant uses the bank's public key to verify the validity of the e-payment voucher, that is to detect $\sigma^e \overset{?}{=} H(ID_i, A) \bmod n$.
3. The merchant adds the transaction information NOW to the order and forms a quotation then signs with his signing key, that is, $s = Sig_M(\sigma, ID_i, A, SPN, SPM, NOW)$. Then the merchant send the transaction information NOW and signature s to the customer.

The customer verifies the validity of the quotation signed by the merchant; that is, check $Ver_M(s) \overset{?}{=}$ "$true$". Figure 3 shows the process of customer purchasing phase. Steps 1 to 4 of the mobile banking payment system (see Figure 1) are conducted in this phase.

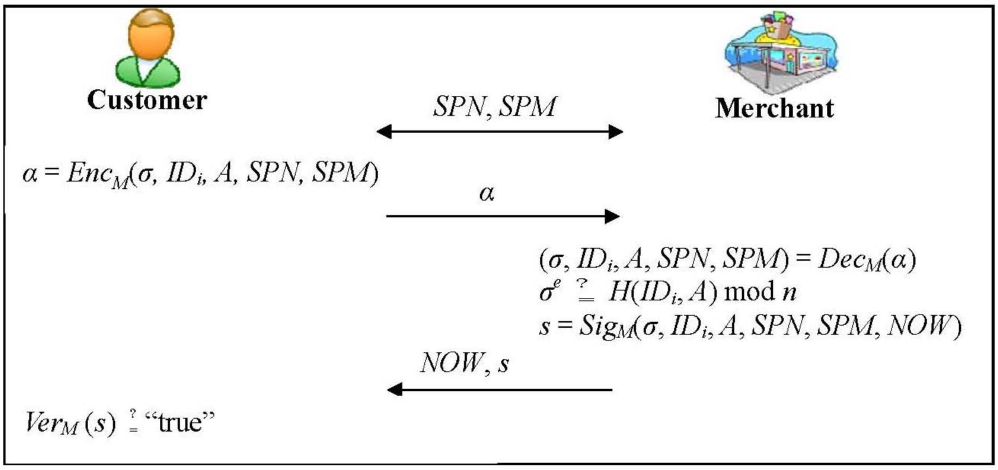

Customer
Merchant
SPN, SPM
$\alpha = Enc_M(\sigma, ID_i, A, SPN, SPM)$
α
$(\sigma, ID_i, A, SPN, SPM) = Dec_M(\alpha)$
$\sigma^e \overset{?}{=} H(ID_i, A) \bmod n$
$s = Sig_M(\sigma, ID_i, A, SPN, SPM, NOW)$
NOW, s
$Ver_M(s) \overset{?}{=}$ "true"

Figure 3: The Customer Purchasing Phase

(3) Customer Payment phase After accepting the quote for the purchase, the customer then uses the e-payment voucher permitting bank to proceed with the payment. The process is as follows:

1. Constructing the document $m_2 = (ID_i, ID_M, ID_{BK}, SPN, SPM, NOW)$. The contents of the document include: identity messages of the customer, merchant and bank, name and price of the purchased item, and transaction information.
2. The customer retrieves the stored information (m_1, r_1).
3. First, select a new trapdoor key $x_2 \in_R \{0,1\}^l$ and calculate $y_2 = g^{x_2} \bmod n$. Then execute the trapdoor operation $TH_{TK}(m_2) = r_2 = r_1 + x_1 m_1 - x_2 m_2$. For $r_1 + x_1 m_1 = r_2 + x_2 m_2$, it implies that $A = g^{r_1} y_1^{m_1} = g^{r_2} y_2^{m_2} \bmod n$.
4. $(\sigma, ID_i, ID_M, ID_{BK}, SPN, SPM, NOW, s, r_2, y_2)$ forms the payment information. The customer then encrypts the information and sends it to the bank to give the bank the go ahead to complete payment transaction.

The customer payment phase is shown in Figure 4. In this phase, the fifth step (shown in Figure 1) of the mobile banking payment system is fulfilled.

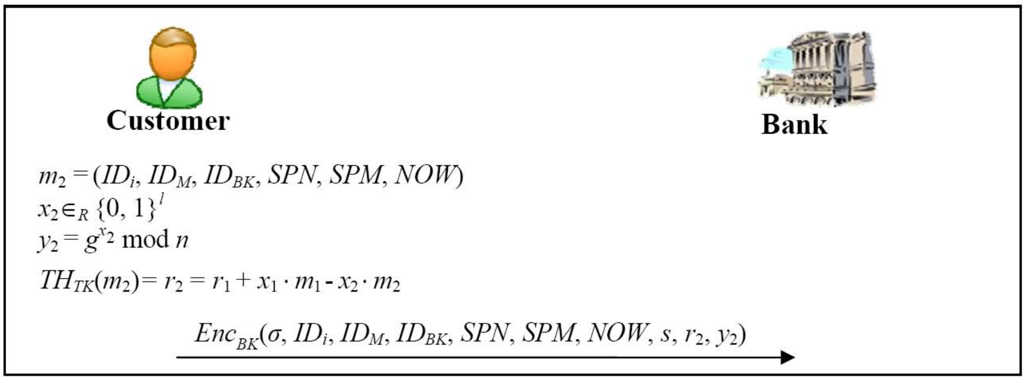

Customer
Bank
$m_2 = (ID_i, ID_M, ID_{BK}, SPN, SPM, NOW)$
$x_2 \in_R \{0, 1\}^l$
$y_2 = g^{x_2} \bmod n$
$TH_{TK}(m_2) = r_2 = r_1 + x_1 \cdot m_1 - x_2 \cdot m_2$
$Enc_{BK}(\sigma, ID_i, ID_M, ID_{BK}, SPN, SPM, NOW, s, r_2, y_2)$

Figure 4: The Customer Payment Phase

(4) The Bank Deduction phase After receiving the go ahead for payment, the bank uses the decryption key to decrypt the message and obtain payment information $(\sigma, ID_i, ID_M, ID_{BK}, SPN, SPM, NOW, s, r_2, y_2)$. The bank then verifies the customer's account balance for the purchase. The bank also verifies the legality of the customer's e-payment voucher and order information. The verification process is as follows:

1. Produce the document $m_2 = (ID_i, ID_M, ID_{BK}, SPN, SPM, NOW)$.
2. Calculate the hash value $A = TH_{HK}(m_2, r_2) = g^{r_2} y_2^{m_2} \bmod n$.
3. Verify the legality of e-payment voucher; that is $\sigma^e \overset{?}{=} H(ID_i, A) \bmod n$.
4. Verify the merchant's quote $Ver_M(s) \overset{?}{=}$ "$true$".

After verifying the above information, the bank accepts the payment information and temporarily freezes the *SPM* amount in the customer's account. The bank sends approved transaction information to the merchant; the merchant accepts the order and delivers the purchased item to the customer. After receiving the item, the customer sends the transaction completion message to the bank. After receiving the message, the bank saves the transaction details $(\sigma, ID_i, ID_M, ID_{BK}, SPN, SPM, NOW, s, r_2, y_2)$ in its database and transfers the amount to the merchant's account. Figure 5 shows the process of the bank deduction phase. In this phase, steps 6 to 9 of the mobile banking payment system are fulfilled.

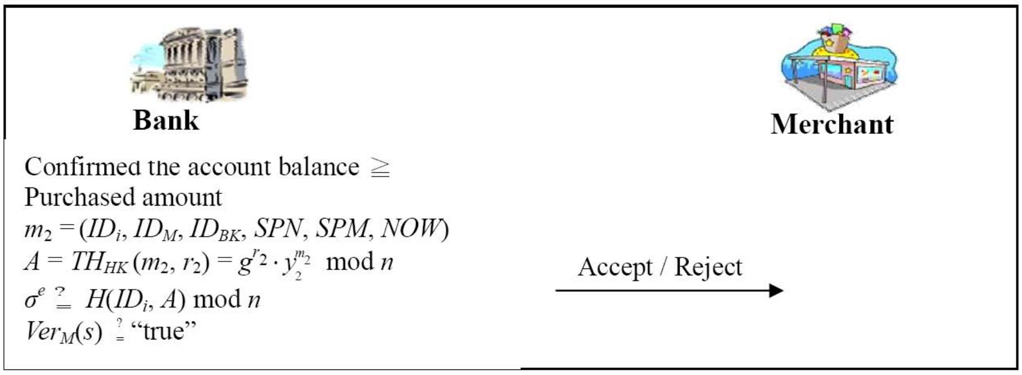

Bank
Merchant
Confirmed the account balance ≧
Purchased amount
$m_2 = (ID_i, ID_M, ID_{BK}, SPN, SPM, NOW)$
$A = TH_{HK}(m_2, r_2) = g^{r_2} \cdot y_2^{m_2} \bmod n$
$\sigma^e \overset{?}{=} H(ID_i, A) \bmod n$
$Ver_M(s) \overset{?}{=}$ “true”
Accept / Reject

Figure 5: The Bank Deduction Phase

4 Security analysis

The mobile banking payment system introduced here meets the following security requirements.

(1) Anti-forging protection
Every legal quotation has to be signed with the merchant’s signing key. If an attacker wanted to forge quotation information, he/she would have to obtain the merchant’s signing key, it would be difficult. In addition, only the actual customer possesses the trapdoor key. Without this trapdoor key, it would be impossible for an attacker to calculate (m_2, r_2) such that the hash value $A = TH_{HK}(m_2, r_2) = g^{r_2} y_2^{m_2} \bmod n$.

(2) Anti-denial protection
For every transaction made between the customer and merchant, the quotation has to be signed through the merchant’s signing key. After the signature is completed, the quotation is returned to the customer, who uses the public key published by the merchant to verify the quotation. This makes it impossible for the merchant to deny signing (the information on) the quotation. On the other hand, in the payment phase, the customer uses his/her own trapdoor key to calculate $TH_{TK}(m_2)$ and obtain (m_2, r_2) and generate the hash value A. Only a customer who knows the trapdoor key could perform the calculation of $TH_{TK}(m_2)$. This makes it impossible for the customer to deny that signing the transaction (payment) information.

(3) Accuracy
The quote generated by the customer and the merchant can be provided to any party for verification. If any party wants to check the accuracy of the quote, he/she would be able to use the public key published by the merchant to verify the signature as well as the transaction.

(4) Prevention of Double Deductions
Payment information $(\sigma, ID_i, ID_M, ID_{BK}, SPN, SPM, NOW, s, r_2, y_2)$ is generated after the transaction is made between the customer and the merchant. The bank saves the customer’s payment information to its database. Every time the bank receives a new payment request from the customer, the bank checks its database to ensure that no double spending occurs. If the same payment information is found, the request for the new payment is rejected. This makes it possible to efficiently eliminate any possibility of double deduction from the customer’s account.

5 Discussion

This section discusses the advantages of the mobile banking payment system in terms of flexibility of usage and efficiency of computation.

In the previous e-cash systems [2, 1, 3, 4], customers had to apply a purchase a fixed amount of e-cash from the bank for later transaction. The transaction amount would either be more or less than the applied amount, making it impossible to perform on-line transactions efficiently. Hence, in the proposed system, every transaction is paid directly to the merchant through the bank. This deduction comes directly from the customer's bank account, so the customer's consumption is based on his/her account balance. No unused e-money is left, and customers can avoid having to purchase e-cash from the bank repeatedly.

Chang and Juang [2, 4], developed a set of protocols that required a huge number of modulo and hash operations to complete the purchase and deposit phases. The application of such protocols with mobile devices had a number of shortcomings including limited battery power and low computational capabilities that could not be overcome. The protocols developed in this study include trapdoor hash functions to reduce the limitations of online computing. The trapdoor hash functions take the advantages of pre-computation (offline computation). Therefore only integer multiplication is required during the online operation, making the payment system possible for application with mobile devices.

In the proposed mobile banking payment system, the mostly heavy computations are modular exponentiations, i.e. computing $g^s \bmod p$. With respect to level 4 security, Lenstra and Verheul [21] and ECRYPT II Recommendations [22] proposed symmetric key lengths in 2010 of 80 bits and asymmetric key lengths of 1248 bits, respectively. Thus, assuming the bit lengths of $|s|$ and $|p|$ are 160 and 1248, respectively. Two experimental platforms have been selected to measure the efficiency of computation. In server side, the platform is: HP 6531s, CPU Intel Core2 Duo P8400 2.26 GHz 2.27 GHz, RAM 4 G Bytes, OS Windows 7(32 bit), program language Java 1.6, and IDE NetNeans 6.9, computing one modular exponentiation requires 4 ms. In the client side, the platform is: HTC Desire HD, CPU Qualcomm Snapdragon QSD8255 1GHz, RAM 768 M Bytes, OS Android 2.2, program language Java for Android SDK, and IDE NetNeans 6.9, computing one modular exponentiation requires 3.1 ms. Assume that the functions of encryption, decryption, signing and verifying signatures all require one modular exponentiation to complete. Then, either customer or merchant or bank takes no more than four modular exponentiations to compute in each phase. The computation of the proposed protocol is efficient.

6 Conclusion

With the rapid development of the Internet and information technology, electronic commerce system is flourishing. Unfortunately, current e-cash systems have proven inconvenient. In this study, we present an on-line banking payment system based on trapdoor hash functions, thus enabling customers to perform transactions directly through their bank. The computation is very efficient, using a smart phone as the client side, computation time is no more than 6.2 ms. By utilizing trapdoor hash functions and preparing the modular exponentiation during the idle time, this study can further reduce the online computational demands to only several micro-seconds, making the system applicable to most mobile devices, and meeting the security requirements of e-cash commerce systems.

6.1 Acknowledgments

This work was partially supported by the I-Services project funded by Ministry of Education and the National Science Council (under Grant NSC 98-2221-E-324-019), Taiwan, R.O.C.

References

[1] D. Chaum, “Blind signature for untraceable payments,” in *Proc. of the 2nd Annual International Cryptology Conference (CRYPTO’82) - Advances in Cryptology, Santa Barbara, California, USA*. Plenum Press, August 1982, pp. 199–203.

[2] C.-C. Chang and Y.-P. Lai, “A flexible date-attachment scheme on e-cash,” *Computers & Security*, vol. 22, no. 2, pp. 160–166, February 2003.

[3] Y.-Y. Chen, J.-K. Jan, and C.-L. Chen, “A novel proxy deposit protocol for e-cash systems,” *Applied Mathematics and Computation*, vol. 163, no. 2, pp. 869–877, April 2005.

[4] W.-S. Juang, “D-cash:a flexible pre-paid e-cash scheme for date-attachment,” *Electronic Commerce Research and Applications*, vol. 6, no. 1, pp. 74–80, March 2007.

[5] J.-S. Wang, F.-Y. Yang, and I. Paik, “A novel E-cash payment protocol using trapdoor hash function on smart mobile devices,” *International Journal of Computer Science and Network Security*, vol. 11, no. 6, pp. 12–19, June 2011.

[6] X. Dai and B. W. Lo, “NetPay-an efficient protocol for micro-payments on the WWW,” in *Proc. of the 5th Australian World Wide Web Conference (AusWeb99), Lismore NSW 2480, Australia*, April 1999. [Online]. Available: http://ausweb.scu.edu.au/aw99/papers/dai/paper.html

[7] Paypal team, “Micropayments,” payclick.com.au, 2010.

[8] Visa team, “Visa launches new way to pay online,” payclick.com.au, 2010.

[9] The Exception Magazine, “Exception magazine launches news industry’s first mobile micropayment system,” Exceptionmag.com, 2010.

[10] Nytimes, “In online world, pocket change is not easily spent,” http://www.nytimes.com, 2007.

[11] S. O’Hear, “Flattr opens to the public, now anybody can like a site with real money,” http://eu.techcrunch.com/2010/08/12/flattr-opens-to-the-public-now-anybody-can-like-a-site-with-real-money/, 2010.

[12] IBM team, “Ibm micro payments,” http://www-4.ibm.com/software/webservers/commerce/payment/mpay/, 1999-2000.

[13] H. Krawczyk and T. Rabin, “Chameleon signatures,” in *Proc. of the 2000 Network and Distributed System Security Symposium (NDSS’00), San Diego, California, USA*. Internet Society, February 2000, pp. 143–154.

[14] A. Shamir and Y. Tauman, “Improved online / offline signature schemes,” in *Proc. of the 21st Annual International Cryptology Conference (CRYPTO’01) - Advances in Cryptology, Santa Barbara, California, USA, LNCS*, vol. 2139. Springer-Verlag, August 2001, pp. 355–367.

[15] F.-Y. Yang, “Efficient trapdoor hash function for digital signatures,” *Chaoyang Journal*, vol. 12, pp. 351–357, 2007.

[16] ——, “Improvement on a trapdoor hash function,” *International Journal of Network Security*, vol. 9, no. 1, pp. 17–21, July 2009.

[17] F.-Y. Yang, S.-H. Chiu, and C.-M. Liao, “Trapdoor hash functions with efficient online computations,” in *Proc. of the 2006 Multimedia and Networking Systems Conference (MNSC’06), Kaohsiung City, Taiwan*, December 2006. [Online]. Available: http://ir.lib.cyut.edu.tw:8080/bitstream/310901800/5861/1/C18.pdf

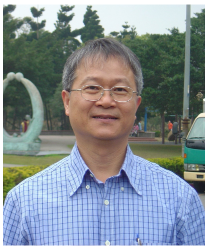
[18] D. Pointcheval, “The composite discrete logarithm and secure authentication,” in *Proc. of the 3rd International Workshop on Practice and Theory in Public Key Cryptography (PKC’00), Melbourne, Australia, LNCS*, vol. 1751. Springer-Verlag, January 2000, pp. 113–128.

[19] G. Poupard and J. Stern, “On the fly signatures based on factoring,” in *Proc. of the 6th ACM Conference on computer and communications security (CCS’99), Singapore, Singapore*. ACM, November 1999, pp. 48–57.

[20] T. Okamoto, M. Tada, and A. Miyaji, “Efficient ‘on the fly’ signature schemes based on integer factoring,” in *Proceedings of the 2nd International Conference on Cryptology in India, INDOCRYPT’01*, vol. 2247. Springer-Verlag, December 2001, pp. 275–286.

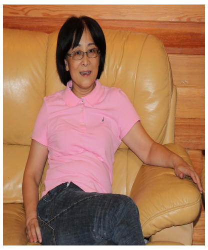
[21] A. Lenstra and E. Verheul, “Selecting cryptographic key sizes,” in *Proc. of the 3rd International Workshop on Practice and Theory in Public Key Cryptography (PKC’00), Melbourne, Australia, LNCS*, vol. 1751. Springer-Verlag, January 2000, pp. 446–465.

[22] ECRYPT II Recommendations, “Yearly report on algorithms and keysizes (2010),” D.SPA.13 Rev 1.0, ICT-2007-216676 ECRYPT II, 03/2010, 2010.



Fuw-Yi Yang received the BS and MS degree in the Department of Electronic Engineering from National Taiwan University of Science and Technology, Taipei, Taiwan, and the Ph.D. degree in the Department of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. He is currently an associate professor with the Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung, Taiwan. He is a member of the Chinese Cryptology and Information Security Association (CCISA) and Taiwanese Association for Consumer Electronics (TACE). His research interests include computer cryptography, network security, and information security.

Zhen-Wei Liu received the BS degree in the Department of Computer Science and Information Engineering, Nan Kai University of Technology. He is currently pursuing his MS degree in the Department of Computer Science and Information Engineering, Chaoyang University of Technology. His research interests include network security and information security.



Su-Hui Chiu received the BS degree in the Department of Accounting and Information, Overseas Chinese University, Taichung, Taiwan, and the MS degree in the Department of Industrial Engineering and Management, Chaoyang University of Technology. She is currently the director of the Office of Accounting, Chaoyang University of Technology. Her research interests include financial management and information security.