Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 4, pp. 551-557

A Secure and Efficient Off-line Electronic Payment System for Wireless Networks

H. Oros, C. Popescu

Horea Oros, Constantin Popescu
Department of Mathematics and Computer Science, University of Oradea
Str. Universitatii 1, Oradea, Romania
E-mail: {horos,cpopescu}@uoradea.ro

Abstract: An electronic cash system allows the exchange of digital coins with value assured by the bank's signature and with concealed user identity. In an electronic cash system, a user can withdraw coins from the bank and then spends each coin anonymously and unlinkably. In this paper we propose a secure and efficient off-line electronic payment system based on bilinear pairings and group signature schemes. The anonymity of the customer is revocable by a trustee in case of a dispute. Because the amount of communication in the payment protocol is about 480 bits, the proposed off-line electronic payment system can be used in wireless networks with limited bandwidth.
Keywords: Electronic payment system, bilinear pairings, group signatures, membership certificate.

1 Introduction

Chaum suggested the first electronic cash system [5] in 1982. In this system the technique of blind signatures was used to guarantee the privacy of users. Various extended systems have been proposed, which provide functionalities such as anonymity, double spending prevention, unforgeability, untraceability and efficiency [1], [4], [8]. Off-line electronic cash systems were first introduced in [6] and then developed further in [9], [10], [11], [12]. In off-line systems the bank's involvement in the payment transaction between a customer and a merchant was eliminated. Customers withdraw electronic coins from the bank and use them to pay a merchant (a shop). The merchant subsequently deposits the coins back to the bank.

In this paper we propose a secure off-line electronic payment system based on bilinear pairings and group signature schemes. In order to construct our electronic cash system, we use the group signature scheme of D. Yao and R. Tamassia [16] and the blind signature of Schnorr [13]. Due to the low amount of communication in the payment protocol that is about 480 bits, our off-line electronic payment system can be used in wireless networks with limited bandwidth.

The rest of this paper is organized as follows. In the next section we present our off-line electronic cash system. Furthermore, we discuss some aspects of security and efficiency in section 3. Finally, section 4 concludes the work of this paper.

2 The Proposed Off-Line Electronic Payment System

An e-cash system is a set of parties with their interactions, exchanging money and goods. A typical e-cash system has three parties:

- Customer: purchases goods or services from the merchant using the e-cash.
- Merchant: sells goods or services to the customer, and deposits the e-cash to the bank.
- Bank: issues the e-cash and maintains the bank account for customers and merchants.

And there are also three protocols: withdrawal, payment and deposit. A customer withdraws electronic coins from the bank and pays the coins to a merchant. Finally, the merchant deposits the paid coins to the bank.

Our electronic payment system consists of four types of participants: customers, merchants, banks and trusted parties. The customers honestly withdraw money from the bank and pay money to the merchant. The merchants get money from customers and deposit it in the bank. The banks manage customer accounts, issue and redeem money. The bank can legally trace a dishonest customer with the help of the trusted parties. An e-cash system is

anonymous if the bank in collaboration with the merchant cannot trace the coin to the customer. The system is off-line if during payment the merchant does not communicate with the bank.

In our off-line electronic cash system, all customers who open a bank account form a group and a trusted party is the group manager. When a customer wants to withdraw an electronic coin from his account, the bank applies a blind signature protocol [13] to this coin and decreases appropriate amount from the customer's account. Everyone including the merchant can verify the validity of the blind signature. The withdrawals are made by the bank by applying the blind signature of Schnorr [13] to a coin randomly selected by a customer and the payments are made by the customer by applying the group signature scheme of D. Yao and R. Tamassia [16] to the random coin.

2.1 System Parameters

This operation outputs the system parameters and public/private keys of users that will be used in the system.

- The group manager chooses a set of public parameters $Y = (G_1, G_2, e, P, H, H', H'')$, where G_1 and G_2 are groups of a large prime order q, G_1 is a gap group, $e : G_1 \times G_1 \to G_2$ is a bilinear map, P is a generator of G_1 and $H : \{0,1\}^* \to G_1$, $H' : \{0,1\}^* \to Z_q$ and $H'' : \{0,1\}^* \times G_1 \to Z_q$ are three collision-resistant hash functions. The group manager chooses his secret key $s_A \in Z_q^*$ and computes the public key $P_A = s_A P$.
- The customer chooses a secret $s_u \in Z_q^*$ as his private key and computes the product $P_u = s_u P$ as its public key.
- The bank selects a random secret x_b from the interval $[1, q-1]$ and calculates the point $P_b = x_b P$. The public key of the bank is P_b and the corresponding secret key is x_b.

The process for selecting the parameters and generating G_1, G_2, q, e, P is given in [2].

2.2 The Registration Protocol

We assume that communication between the customer and the group manager is secure, i.e., private and authentic.

Any customer who wants to withdraw a coin from the bank has to interact with the group manager and obtains two type of certificates from the group manager. One is long-term group membership certificate, which certifies the customer's public key information. The other is one-time signing permit, which certifies the customer's one-time signing key information. The latter is used for issuing signatures in the payment protocol.

The registration protocol involves the customer and the group manager as follows:

1. A customer obtains a long-term group membership certificate *Cert* from the group manager. The group manager computes $Cert = s_A H(info \| s_u P)$, where s_A is the private key of the group manager, $s_u P$ is the customer's public key and $info$ contains information such as group name and membership expiration date. *Cert* is given to the customer.
2. A customer also obtains one-time signing permits from the group manager. The customer randomly chooses a number of secrets $x_1, ..., x_l$ and computes one-time signing secret keys $x_1 P, ..., x_l P$ and one-time signing public keys $s_u x_1 P, ..., s_u x_l P$. The keys $s_u P$ and $s_u x_i P$ are sent to the group manager, for all $i = 1, ..., l$. The customer also sends *Cert* to the group manager.
3. The group manager first checks if the customer with public key $s_u P$ is a valid group member. This is done by verifying the following equality:

$$e(Cert, P) = e(P_A, H(info \| s_u P))$$

where P_A is the group manager's public key and $s_u P$ is the customer's public key. The protocol terminates if $e(Cert, P) \neq e(P_A, H(info \| s_u P))$. Then the group manager tests if $e(s_u x_i P, P) = e(s_u P, x_i P)$ for all $i = 1, ..., l$. If the test fails, the protocol terminates. Otherwise, the group manager computes:

$$S_i = s_A H(info \| s_u x_i P)$$

for all $i = 1, ..., l$. S_i is an one-time signing permit and is given to the customer. The group manager adds the tuple $(s_u P, x_i P, s_u x_i P)$ to its record for all $i = 1, ..., l$.

2.3 The Withdrawal Protocol

We assume that communication between the customer and the bank is secure, i.e., private and authentic. The withdrawal protocol allows a customer to withdraw e-coins from the bank. After having open a bank account, the customer withdraws an e-coin from his account by using the blind signature. Therefore, the bank cannot link the e-coin to the identity of the customer but can debit to the account correctly. The withdrawal protocol involves the customer and the bank in which the customer withdraws an electronic coin from the bank. First, the customer proves his identity to the bank using the elliptic curve version of the signature scheme of Shao [14]. Then, the bank uses the elliptic curve version of the blind Schnorr signature scheme [13] to sign the e-coin.

The customer must perform the following protocol with the bank:

1. The customer sets his electronic cash requirement:

$$m = H'(\textit{withdrawal require}||ID)$$

where ID is the identity of the customer. Then, the customer chooses a random value $k_u \in [1, q-1]$ and signs the message m using the elliptic curve version of the signature scheme of Shao [14]:

$$f = H'(m) \quad (1)$$
$$R = k_u f P \quad (2)$$
$$h = H''(m,R) \quad (3)$$
$$s = k_u - h s_u. \quad (4)$$

The customer sends m and its signature (h,s) to the bank.

2. The bank verifies the signature (h,s) of the message m:
   (a) The bank first computes $f = H'(m)$, $R' = f(hP_u + sP)$ and $h' = H''(m,R')$.
   (b) Then, the bank checks that the following equality holds:

$$h = h'.$$

   (c) If $h \neq h'$ the protocol terminates.
3. Then, the bank uses the elliptic curve version of the blind Schnorr signature [13] to sign the e-coin: selects $k' \in [1, q-1]$, computes the point $R'' = k'P$ and sends R'' to the customer.
4. The customer establishes a random coin c, randomly selects $\alpha, \beta \in [1, q-1]$, computes $R_b = R'' + \alpha P + \beta P_b, c_b = H''(c||\alpha, R_b)$ and blinds the e-coin by computing $c' = c_b - \beta \bmod q$. The customer sends the value c' to the bank.
5. The bank computes: $s' = k' - c' x_b \bmod q$ and forwards s' to the customer.
6. The customer computes $s_b = s' + \alpha \bmod q$. The pair (c_b, s_b) is a valid e-coin signature issued by the bank.
7. The customer verifies the blind signature (c_b, s_b) of the coin c, issued by the bank, by checking that the following equation holds:

$$s_b P + c_b P_b = R_b \quad (5)$$

8. The blind signature of the coin c is the pair (c_b, s_b).

The customer gets the coin c from his account.

2.4 The Payment Protocol

The payment protocol involves the customer and the merchant and should be done through a secure channel (i.e., data privacy and integrity). In the proposed system, during payment the merchant does not communicate with the bank. After withdrawing e-coins, the customer can pay for what the merchant provided. Then the merchant verifies the validity of the received e-coins.

In order to sign the coin c, the customer uses the protocol of Yao and Tamassia [16]. The merchant first sends a challenge c_m to the customer. Then, the customer produces a signature S_u of the coin c and merges the signature S_u with his one-time signing permit S_i associated with the secret $s_u x_i$. The details are as follows:

1. The merchant sends challenge $c_m = H'(ID_m||T)$ to the customer, where ID_m is the merchant's identity and T is the recorded time of the transaction.
2. The customer computes:
$$c_u = H'(c||c_m||c_b||s_b) \tag{6}$$
3. The customer computes $S_u = s_u x_i H(c_u)$.
4. The customer computes the signature $S = S_u + S_i$, where $S_i = s_A H(info||s_u x_i P)$.
5. The customer sends c, c_u and the signature $S = S_u + S_i$ of the coin c to the merchant.
6. The merchant verifies the signature S of the coin c as follows:
   (a) Computes the hash digest $H(c_u)$ and the hash digest $h' = H(info||s_u x_i P)$ of one-time signing permit.
   (b) The signature S is accepted if
$$e(S,P) = e(P_A, h')e(s_u x_i P, H(c_u)). \tag{7}$$
   If the test fails, the protocol terminates.

2.5 The Deposit Protocol

The deposit protocol permits the merchant to deposit the received e-coins to the bank. When receiving the deposited requirement from the merchant, the bank first verifies the validity of received e-coins and then credits the account of the merchant.

In on-line e-cash systems this protocol is part of the payment protocol as executed by the merchant. In our e-cash system, the deposit protocol is executed at a later moment, preferably in batch mode. The bank holds a record of spent cash to prevent double spending of e-cash. The bank cannot link deposited coins to a customer without collaboration from the group manager.

The deposit protocol involves the merchant and the bank as follows:

1. The merchant sends c, c_m, c_u, c_b, s_b to the bank.
2. The bank verifies the signature as given in the equation (6).
3. After verification succeeds, the bank checks if c obtained from the merchant exists in its database. If the coin c is in the database of the bank, then the bank finds the signature S' for the deposited coin in its database and sends it to the merchant (detection of double spending).
4. If the merchant receives S' from the bank, he/she checks whether $S' = S$. If $S' = S$, then the merchant rejects performing protocol (double spending). Otherwise, the merchant sends c_u and T to the bank.
5. The bank verifies the validity of the signature S using the equation (7).
6. If the signature S of the coin c is valid, then the bank accepts the coin c. Then, the bank will deposit the cash to the merchant's account and the merchant sends the goods to the customer. The bank stores c and $(c_u, s_u x_u P)$ in its database.
7. If the bank finds out that c and $(c_u, s_u x_u P)$ has been stored before but different T and c_m, then the coin c has been double spending. The bank performs the tracing protocol and detects the identity of the double spender with the help of the group manager.

2.6 The Tracing Protocol

The bank can legally trace the customer of a paid coin with the help of the group manager. The tracing protocol involves the bank and the group manager. Given a signature S and its associated public information P_A and $s_u x_i P$, the group manager verifies the signature S. If the signature S is valid, the group manager can identify a customer's public key $s_u P$ from $s_u x_i P$ value, by consulting the customer group record. The details are as follows:

1. The bank sends c_u and the signature S of the coin c to the group manager.
2. The group manager verifies the signature S using the equation (7).

3. The group manager can easily identify the customer from $s_u x_i P$. The group manager can provide a proof that it is indeed the customer's signature from the following equations:

$$e(s_u x_i P, P) = e(s_u P, x_i P) \tag{8}$$

4. The group manager searches through the group customer list to get the identity of the customer and sends it to the bank.

Similar to what is shown in the group signature scheme of Chen et al. [7], the group manager cannot misattribute a signature to frame the customer unless he can compute bP given q, P, aP and dP which satisfies:

$$a \equiv db \bmod q \tag{9}$$

The authors in [7] define this problem the Reversion of Computation Diffie-Hellman Problem. They prove that the Reversion of Computation Diffie-Hellman Problem is equivalent to Computational Diffie-Hellman Problem in G_1.

3 Security and Efficiency Analysis

In this section we discuss some aspects of security and efficiency of our off-line electronic payment system. We prove that our off-line electronic payment system is secure against tracing a honest customer by the bank and the proposed system is secure against forgery of the coin.

Theorem 1. *Our off-line electronic payment system is secure against existential forgery of the coin c.*

Proof: Long-term membership certificates, one-time signing permits and customer's signatures using one-time secret signing keys are generated by the sign protocol of the signature scheme of Boneh, Gentry, Lynn and Shacham [3]. The authors in [3] shown that their signature scheme is secure against existential forgery attacks. Therefore, if an adversary can forge any of these signatures, she can also forge signatures in the signature scheme of Boneh et al. [3]. Note that a signature computed with one-time secret signing key is in the form of $s_u x_i H(c_u)$, rather than $s_u H(c_u)$ as in the signature scheme [3]. It can be easily shown that if an adversary can forge a signature in a form of $s_u x_i H(c_u)$, then she can forge a signature in the form of $s_u H(c_u)$. Also, since the blind signature of Schnorr is secure against existential forgery, this allows only the legal bank to generate the signature for coin. As the hash function H' has the feature of collision free, the customer cannot find a value $c' \neq c$ with $H'(c'||c_m) = H'(c||c_m)$. Thus, our payment system satisfies unforgeability of the coin. □

Theorem 2. *The both valid signatures S and (c_b, s_b) in our payment system contain a proof of the group membership without revealing the identity of the customer.*

Proof: A valid signature S is obtained from an one-time signing permit of a customer and the customer's signature using the corresponding one-time signing key. That is $S = S_i + S_u$, where $S_i = s_A H(info||s_u x_i P)$ and $S_u = s_u x_i H(c_u)$. Because of the definition of signatures [3], a valid signature S implies that S_i is valid. This proves that the holder of key $s_u x_i P$ is a certified customer. A valid S also means that S_u is valid, therefore S_u is generated with the secret key $s_u x_i$. Thus, S contains a proof of the customer membership. Because the signing key $s_u x_i$ is one-time signing key and x_i is chosen randomly by the customer, the identity of the customer is not revealed. Also, since $c_u = H'(c||c_m||c_b||s_b)$ and the blind signature (c_b, s_b) of the coin c can not give any information for the coin c, the bank can not link the blind coin with the identity of the customer. □

Table 1: Storage space of the payment systems

	Our system	Wang	Lee	Au	Canard
Withdrawal	1120 bits	1824 bits	800 bits	8160 bits	6420 bits
Payment	480 bits	1282 bits	1304 bits	5188 bits	30740 bits
Deposit	960 bits	3232 bits	1656 bits	5164 bits	27648 bits

Table 2: Computation cost of the payment systems

	Our system	Wang	Lee	Au	Canard
Withdrawal Protocol					
multi-EXP	8	9	15	2156	5
Pairing	0	0	0	22	0
Payment Protocol					
multi-EXP	2	11	9	34	1673
Pairing	3	0	0	14	0
Deposit Protocol					
multi-EXP	0	5	7	10	14
Pairing	3	0	0	0	0

Next, we evaluate the storage space and computational time of the costly operations. Table 1 and Table 2 summarize the storage space and computation cost respectively, of different protocols of our e-cash system and the schemes in [1], [4], [9] and [15]. The overall efficiency is improved in our electronic cash system compared to Au et al.'s system [1], Canard et al.'s system [4], Lee et al.'s system [9] and Wang et al.'s e-cash system [15] in terms of the storage space and the computation cost. Our system has a point P of 160 bits and q of 160 bits. The off-line e-cash system proposed by Lee et al. has a point P of 160 bits and 160 bits prime q and the system of Wang et al. has 160 bits prime q and 321 bits prime p. Spending a coin in [15] requires 11 multi-based exponentiations and a total bandwidth of 1282 bits. The payment protocol in [9] requires 9 multi-based exponentiations and a total bandwidth of 1304 bits. For a moderate value $L = 10$ and $t = 40$, the payment protocol in [4] requires 1673 multi-based exponentiations and a total bandwidth of 30740 bits. The payment protocol in [1] requires 34 multi-based exponentiations, 14 pairings and a total bandwidth of 5188 bits. In contrast, the payment protocol in our e-cash system requires 2 multi-based exponentiation, 3 pairings and a total bandwidth of 480 bits.

4 Conclusions

In this paper we presented a secure and efficient off-line electronic payment system based on bilinear pairing and group signature schemes. We used the group signature scheme of Yao and Tamassia and the blind signature of Schnorr. Because the amount of communication between customer and merchant is about 480 bits, the proposed off-line payment system can be used in the wireless networks with the limited bandwidth.

Bibliography

[1] M. Au, W. Susilo, Y. Mu, Practical anonymous divisible e-cash from bounded accumulators, *Proceedings of Financial Cryptography and Data Security*, Lecture Notes in Computer Science 5143 Springer-Verlag, pp. 287-301, 2008.

[2] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairings. *Advances in Cryptology-Crypto 2001*, Lecture Notes in Computer Science 2139, Springer-Verlag, pp.213-229, 2001.

[3] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, Aggregate and verifiably encrypted signatures from bilinear maps. *In Advances in Cryptology - Eurocrypt'03*, Lecture Notes in Computer Science 2656, Springer-Verlag, pp. 416-432, 2003.

[4] S. Canard, A. Gouget, Divisible e-cash systems can be truly anonymous, *Proceedings of EUROCRYPT 2007*, Lecture Notes in Computer Science 4515, Springer-Verlag, pp. 482-497, 2007.

[5] D. Chaum, Blind signature for untraceable payments. *Proceddings of Eurocrypt'82*, Plenum Press. pp.199-203, 1983.

[6] D. Chaum, A. Fiat, M. Naor, Untraceable electronic cash, *Proceedings of the Crypto'88*, pp. 319-327, 1990.

[7] X. Chen,F. Zhang, K. Kim, A New ID-based Group Signature Scheme from Bilinear Pairings. *Journal of Electronics*, 23, pp. 892-900, 2006.

[8] C. Fun, Ownership-attached unblinding of blind signatures for untraceable electronic cash, *Information Science*, 176(3), pp. 263-284, 2006.

[9] M. Lee, G. Ahn, J. Kim, J. Park, B. Lee, K. Kim, H. Lee, Design and implementation of an efficient fair off-line e-cash system based on elliptic curve discrete logarithm problem, *Journal of Communications and Networks*, 4(2), pp. 81-89, 2002.

[10] T. Okamoto, K. Ohta, Universal electronic cash, *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pp. 324-337, 1992.

[11] T. Okamoto, An efficient divisible electronic cash scheme, *Proceedings of Crypto'95*, Lecture Notes in Computer Science 963, Springer-Verlag, pp. 438-451, 1995.

[12] C. Popescu, An Electronic Cash System Based on Group Blind Signatures. *Informatica*, 17(4), pp. 551-564, 2006.

[13] C.P. Schnorr, Efficient signature generation for smart cards, *Journal of Cryptology*, 4(1991), pp. 239-252, 1991.

[14] Zuhua Shao, A provably secure short signature scheme based on discrete logarithms, *Information Sciences: an International Journal*, vol.177(23), pp. 5432-5440, 2007.

[15] H. Wang, J. Cao, Y. Zhang, A flexible payment scheme and its role-based access control. *IEEE Transactions Knowledge Data Engeneering*, 17, pp. 425-436, 2005.

[16] D. Yao, R. Tamassia, Cascaded Authorization with Anonymous-Signer Aggregate Signatures. *Proceedings of the Seventh Annual IEEE Systems, Man and Cybernetics Information Asssurance Workshop*, USA, pp.84-91, 2006.

Horea Oros (b. August 22, 1977) received his PhD in Computer Science (2009) from "Babeş Bolyai" University of Cluj-Napoca, Romania. Since 2001 he is working within the Department of Mathematics and Computer Science, Faculty of Sciences, University of Oradea, Romania, where currently he is a lecturer. He also is lecturer at Agora University of Oradea. He co-authored three books in the filed of computer science and published 19 articles in several journals and proceedings of prestigious international conferences. His main research interest is in the field of cryptology and computer security.

Constantin Popescu (b. October 21, 1967) received his PhD in Computer Science (2001) from "Babeş Bolyai" University of Cluj-Napoca, Romania. Since 2005 he is a professor at the Department of Mathematics and Computer Science, University of Oradea, Romania. His research interests include cryptography, network security, group signatures, security protocols and electronic payment systems. He co-authored 7 books in the field of computer science and published 49 articles in several journals and proceedings of prestigious international conferences. He is reviewer for 10 journals and several prestigious international conferences.