Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 4 / ISSUE 1 / APRIL 2009 / 91-100

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000100008

An Improved and Efficient Micro-payment Scheme

Praneetha R. Bayyapu[1] and Manik Lal Das[2]

Dhirubhai Ambani Institute of Information and Communication Technology
Gandhinagar-382007, Gujarat, India
[1] praneetha_reddy@daiict.ac.in, [2] maniklal_das@daiict.ac.in

Received 28 March 2008; received in revised form 28 August 2008; accepted 20 January 2009

Abstract

The rapid growth of data communications networks in recent years has led to enormous development in electronic commerce. Internet banking and trading are two important applications that execute financial transaction from anywhere in the world. This enables banks and merchants to simplify their financial transaction process as well as to provide customer friendly service 24-hours a day. On one hand the cost of manpower and infrastructure comes down drastically and on the other hand the cost of transportation, third party royalty and securing customer information is increased. Electronic micro-payment is one of the most important research topics in electronic commerce, particularly, low-cost online payment scenarios and offline payments in rural areas. In this paper, we discuss some of the important micro-payment schemes, observe their merits and limitations, and then propose an improved micro-payment scheme. We discuss two basic micro-payment schemes, which use the public/private key concept and then we review another scheme that uses the concept of hash chain. We observe certain limitations in these and related schemes, which motivate us to extend one of the attractive schemes, namely, the Hwang and Sung' scheme [8], towards more efficient, retaining all other features intact without compromising the security strength of the scheme. We compare the improved scheme with others and show that the improved scheme provides better security and efficiency, which enables the scheme viable for real-world applications, in particular, in resource-constrained environments such as mobile payment through handheld devices or customer's chip card for debit/credit transaction through point of sale terminal.

Key words: Micro-payment, Hash chain, Blind signature, Double spending, Electronic commerce

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 4 / ISSUE 1 / APRIL 2009 / 91-100

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000100008

1 Introduction

With the rising importance of intangible goods in global economies and their instantaneous delivery at negligible cost, conventional payment methods tend to be more expensive than the actual product. Digital contents or online service could be worth for the value smaller than the smallest denomination currency in the physical world. There are two ways of managing electronic payments: *on-line* payments - in which seller verifies the payment sent by buyer with a broker before serving the buyer, and *off-line* payments - in which double spending, overspending should be prevented, therefore, no on-line connection to the broker is required.

Electronic payment models [2], [13], [16] can further be divided into two groups in respect of their on-line requirements, consumer base stability and basic paying methods. The first one is *Payments through transactions* model, where individual payments does not require prior arrangements between buyer and seller. Another is *payments through accounts* model, where buyer and seller must have set up accounts with broker and some kind of agreement between them before execution of actual payment transactions. The payments through transactions can be divided into two subgroups: *The credit card payment transactions* are typically designed for large-fee payments of, say, several tenths, hundreds or even thousands of dollars. On the contrary, *net money transactions* are typically low-value payments with hard transaction cost and on-line requirements, thus resembling the idea of the micro-payments very closely. The disadvantage with the credit card payment transaction is the cost of transactions, especially from the viewpoint of the sellers that have to pay certain bills to the clearinghouse they have made contract with. This of course, will have direct influence on the pricing policy and the interest among the potential customers. According to the transaction amount, a special kind of electronic payment system called micro-payment (Site 2) dealing with small value transactions on digital content and service is gaining increasing interests from the research community. Some important applications of micro-payment are electronic publishing, multimedia service, gamming, etc. In these applications, because of the small transaction amount, the merchant gains relatively mall profit from each transaction. Consequently, expensive computations such as digital signature and exponentiation must be restricted in order to decrease the investments in hardware, software, networks, etc. In the recent past, micro-payments [3], [4], [6]-[10], [12], [14], [15], [17], [18], [21]-[26] present a relatively key innovation in the online revenue stream. The basis of micro-payments is to take advantage of the high volume of viewers by offering content for a low price. Other variations on this idea are to charging fractions of cents for equally fractional amounts of contents, for example, a tenth of a cent per single web page of an online magazine. The important factors in micro-payment system are small amounts of payment value and high frequency of transactions on the electronic commerce network.

1.1 Requirements for Micro-payment System

A micro-payment system comprises the following entities [16]:

- *Customer*: The Customer buys e-cash/e-coin from the Brokers, using real money, via a macro-payment system. The Customer can then use e-coin/e-cash to perform micro-payment purchases.
- *Merchant*: The Merchant is the “data bank”. S/he supplies customers with data, services or both.
- *Broker*: The Broker mediates between merchants and customers in order to simplify the tasks they perform. Typically, s/he acts like a bank and provides the electronic currency for the micro-payments. A generic working flow-diagram of micro-payment system is depicted in Figure 1.

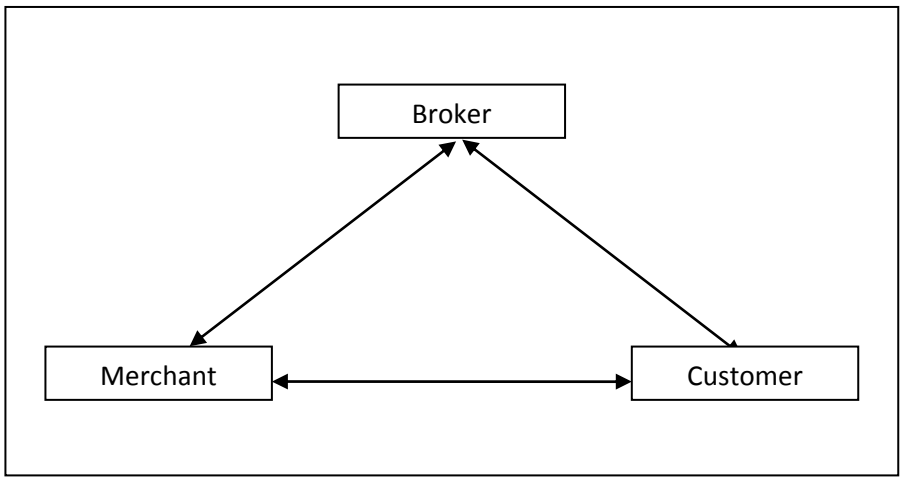

Figure 1: Micro-payment System

While using electronic currency (e-cash/e-coin), a common set of requirements for a payment system is:

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 4 / ISSUE 1 / APRIL 2009 / 91-100

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000100008

- *Anonymity*: e-currency should not provide any users information, in other words, it should be anonymous e-cash transaction.
- *Divisibility*: e-currency can be divided or its changes can be offered because the currency has a basic unit.
- *Transference*: e-currency can be transferred to a third party through offering the appropriate amount of money.
- *Prevention of double spending*: e-currency cannot be used for a second time unless it is faked.

The electronic payments are stored and transported in digital form, which poses new problems for the development of secure electronic payment systems: the payments are easy to be duplicated in contrary to the traditional physical paying instruments. Since the digital payments are represented as simple byte arrays or sequences of bits, nothing in them prevents the copying of them as such. If the security of the payment system is dependent on the way the payments are hidden from outsiders, anyone that can have access to payments could use them, possibly many times. We observe that achieving anonymous cash transaction is an important requirement, and at the same time providing efficiency is another concern. In this paper, we discuss a merchant specific micro-payment system [18]; a multiple merchant supported micro-payment system [10] and followed by a scheme [8] that provides anonymity property using the concept of blinding signature and hash chain. We then present an improvement of the blinding signature phase used in the scheme for [8] achieving more efficiency without compromising its security.

The remainder of the paper is organized as follows: Section 2 reviews three micro-payment schemes [18], [10], [8], which influence significantly to come up with a new micro-payment scheme. The first two schemes [18], [10] are being treated as basic micro-payment systems. The third one [8] is an efficient one, which supports multiple merchants and anonymity property as well. Section 3 presents an improvement of the scheme [8]. Section 4 analyzes the improved scheme. We conclude the paper with Section 5.

2 Related Work

The DigiCash system (Site 1), [3], [4] is based on what is called a single use token system. The user generates blinded electronic bank notes and sends them to his bank to be signed with his bank's public key. The bank signs the notes, deducts the amount from the user's account, and sends the signed notes back to the user. The user removes the blinding factor and uses them to purchase at the shop. The shop verifies the authenticity of the bank notes using the bank's corresponding public key and sends them to the bank where they are checked against a list of notes already spent. The amount is deposited into the shop's account, the deposit confirmed, and the shop in turn sends out the goods. The Millicent [7] is a decentralized micro-payment scheme, which is designed to allow payments as low as 1/10 of a cent. It uses a form of electronic currency, which is called “scrip”. It is designed to make the cost of committing a fraud, more than the value of the actual transaction. It uses symmetric encryption for all data transactions. Millicent is a lightweight and secure protocol for electronic commerce over the Internet. It is designed to support purchases costing less than a cent. It is based on decentralized validation of electronic cash at the vendor's server without any additional communication, expensive encryption, or off-line processing. Later, voucher schemes, lottery ticket and coin-flipping protocols [6], [12], [17] have been proposed, which have the potential to minimize the number of messages involved in each transaction. The electronic vouchers can be transferable but the direct exchange between buyers and sellers is not possible. Consequently, a financial intermediary is required and this will increase the transactions costs of exchange. In addition, the coin-flipping and lottery ticket protocols are based upon the assumption that economic agents are risk-neutral and will be satisfied with fair bets. Over the years, several new micro-payment schemes and improvements [1], [8]-[10], [14], [15], [18], [21]-[26] have been proposed. In this section, we review two basic schemes [10], [18] for a basic understanding of micro-payment system and then we discuss a more efficient and flexible scheme [8]. Interested readers can refer [16] for several other schemes and a wide variety of features and design criteria of micro-payment system. While discussing the subsequent sections in the paper, we use the notation shown in Table 1.

2.1 The PayWord Scheme

In 1996, Rivest and Shamir [18] proposed the PayWord system, which is a credit-based system. The system uses RSA public key cryptography [19] and the concept of hash chain [20]. In the PayWord system, when a registered customer asks the merchant for a service, s/he needs to create a PayWord chain that acts as the currency made payable to the merchant. The merchant then needs to inspect whether the customer is legitimate and the paywords (coins of the PayWord system) chain is generated by the customer.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 4 / ISSUE 1 / APRIL 2009 / 91-100

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000100008

Table 1: Notation used in the Schemes

U	Customer
M	Merchant
B	Broker
ID_X	Identity of X, where X ∈ {U, M, B}
A_X	Address of X, where X ∈ {U, M, B}
C_U	U's certificate
E_U	U's certificate expiry information
E_M	Expiry information for redemption
I_U	U's certificate serial number credit unit info.
OI	Order information (category, amount, etc)
PK_X	Public key of X, where X ∈ {U, M, B}
SK_X	Private key of X, where X ∈ {U, M, B}
r_X	Random number chosen by X, where X ∈ {U, M, B}
P	A generator point on elliptic curve
h( )	Cryptographically secure hash function
K	Secret key of B
{m}K	M is encrypted under key K
⊕	Exclusive-OR operator
||	Concatenation operator

Later, the merchant collects customer's paywords and redeems the payment from the broker. The PayWord system reduces the number of on-line communications between broker and merchant because the merchant does not need to settle for every purchase. The paywords chain generation with length n can be expressed as $w_i = h(w_{i+1})$, where $i = n-1, n-2, \ldots, 0$. When generating the paywords, the customer selects a random number w_n, called a seed. Then w_n is hashed iteratively in reverse order until the root of the chain w_0 is generated. During shopping, the customer in order from w_1 to w_n releases paywords, and the merchant verifies it easily by hash operation.

PayWord is a merchant-specific payment system, that is, the paywords chain is spent only to a particular merchant. When a customer contacts a new merchant M for requesting service, besides generating a new chain, the customer must send a commitment to M. The commitment contains the identity of the merchant, the certificate issued by the broker, the root of an unused chain, the current date and other information.

To execute continuous transactions, the customer pays paywords within the chain which belongs to the merchant in order. After an appropriate period, the merchant will contact with the broker to request redemption. For each chain, the merchant sends the latest paywords s/he received and the customer's commitment to the broker; therefore, hashing from latest paywords to the root of the chain can validate the correctness of transactions. If the validating is correct, the broker debits customer's account with the used length of the chain and credits merchant's account with the same amount. We depict the transaction process of the PayWord system in Figure 2.

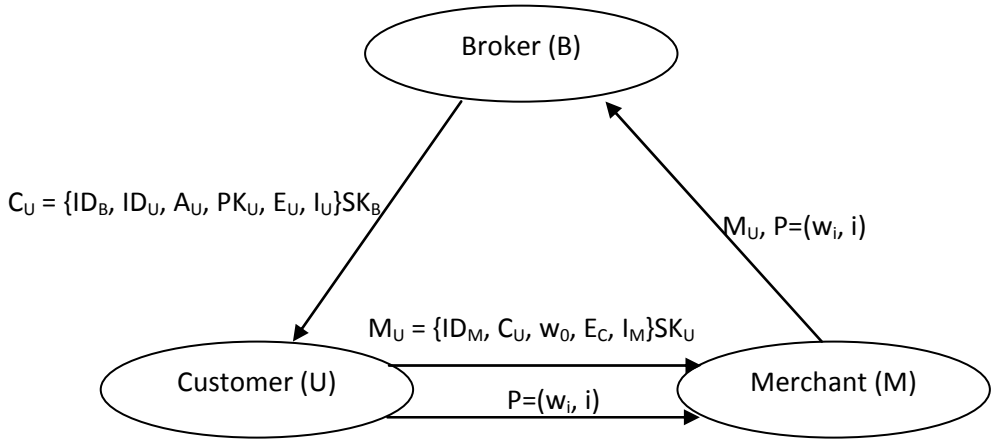

Figure 2: The PayWord scheme

Our Observations: PayWord is designed as a credit-based system. It takes advantage of hash chain to ensure computational efficiency, and achieves non-repudiation for each payment belonging to the same chain by only one signature. After receiving a certificate, a customer is authorized to transact with a merchant in a specified amount without the online communication to broker, which reduces the communication and the risk of the broker becoming a bottleneck, provides the customer more flexibility. However, the system suffers from the following limitation:

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 4 / ISSUE 1 / APRIL 2009 / 91-100

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000100008

- PayWord is a merchant-specific payment system; therefore, customers have to maintain tuples of particular data of chains corresponding to distinct merchants;
- Customer has to perform hash chain operations as many as the number of merchants each time s/he wants to transact business with;
- Customer has to store all the different paywords of each merchant and the last index used for the transactions; and
- Customer could make payments exceeding her/his authorized credit limit.

2.2 Kim and Lee's scheme

In 2003, Kim and Lee [10] proposed a micro-payment scheme that supports multiple merchants. The scheme is divided into three phases: certificate issuing phase, payment phase, and redemption phase.

Certificate Issuing Phase: Customer requests a certificate to a broker by transmitting her/his private information via a pre-established secure channel. The broker sends C_U, which guarantees to be justified and S_U which will be used for the root value in payment phase later. Each customer generates her/his public and private key pair (PK_U, SK_U) and sends PK_U with I_U that includes the maximum number of merchants N, the length of hash chain n as well as her/his credit card data to the broker. Since a customer's certificate signed by a broker, those who intend to use this key must trust him.
The broker creates special data T_U, which acts as a key factor of the root value. It is used to make clear that the new hash values generated by the broker are issued to whom, since no one except the broker can create it.

$T_U = h(U, r_B, K)$, where K is the secret key of the broker
$S_U = \{s_i \mid s_i = h(s_{i+1}, T_U), i = N\text{-}1, \ldots, 0\}$, where s_i is generated by a shared user-broker secret key.

The certificate C_U, in which all the elements including the expiry date of the certificate E_U are signed by the broker and sent to the customer with S_U and a nonce r_U.

$$C_U = \{ID_B, ID_U, PK_U, T_U, I_U, E_U\}SK_B$$

We depict the transaction process of Kim and Lee's protocol in Figure 3.

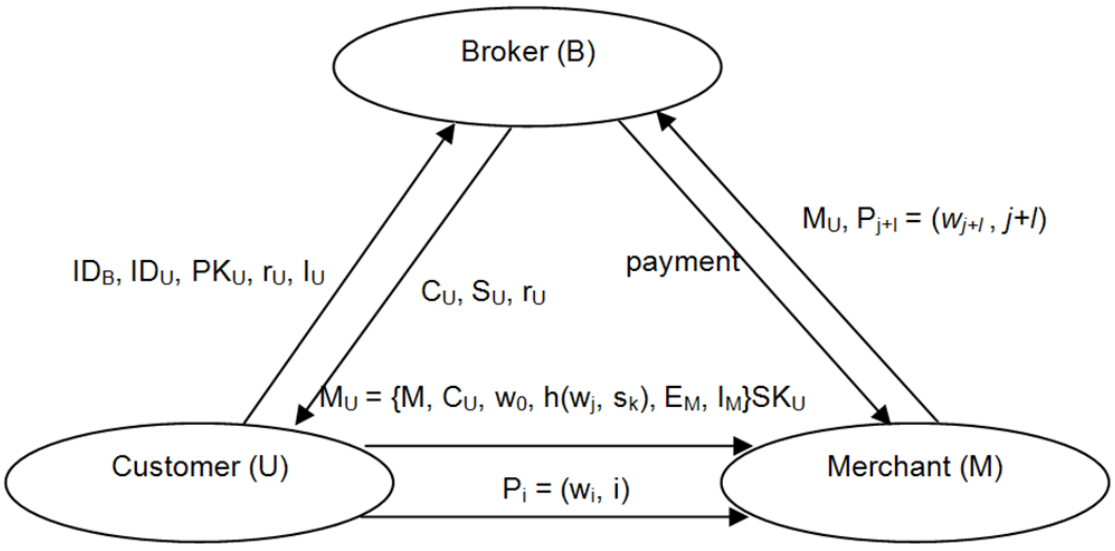

Figure 3: Kim and Lee's scheme

Payment Phase: The root value of paywords is combined with *si* received from a broker, which enables a customer to use the rest of the unspent paywords in succession for multiple payments to different merchants. The customer who gets the certificate in previous phase can now create paywords and commitment. The commitment consists of the identity of the merchant with whom a customer intends to do business, the certificate, the root components which are modified into w_j, $h(w_j, s_k)$, the expiry date of the commitment (E_M), and other information (I_M), where $0 \le j \le n$ used to setup root value for different merchants. Then the customer signs the components

$$M_U = \{V, C_U, w_0, h(w_j, s_k), E_M, I_M\}SK_U$$

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 4 / ISSUE 1 / APRIL 2009 / 91-100

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000100008

In order to spend the rest of the paywords in succession, a customer should set the root value of paywords to be spent in next payment phase into the combination of the hash chain values respectively generated by a customer and a broker. For example, if it is assumed that a customer used paywords as many as w_{j-1} in previous transactions and spent l paywords at the current transaction with k^{th} merchant, the root value of paywords should be identical with $h(w_j, s_k)$ to be proper for the payments. The customer can apply his paywords to different merchants up to the maximum transaction limit of N unless the last payword exceeds w_n. The merchant stores the last received payment information of $P_{j+l} = (w_{j+l}, j+l)$ and the commitment, and terminates the payment phase.

Redemption Phase: Merchant should carry out the redemption process with a broker within a pre-agreed period of time. The broker confirms whether the payment request of the merchant is correct or not by verifying the certificate. The merchant first requests for redemption to a broker by sending a customer's commitment and payment information. From these data, the broker verifies his signature marked at the certificate and redeems P_{j+l} to a corresponding amount of money. We note that a broker can verify paywords only from w_j to w_{j+l} for that request. In other words, since the corresponding root value is w_j, the only thing imposed to the broker is that the last received payword w_{j+l} is identical with w_j by applying hash function l times. The broker processes redemption requests from merchants less than N before being overdue. Finally, the broker completes the redemption process if the last received value w_l is less than the maximum value of the hash chains.

Our Observations: The scheme supports multiple merchant payments and prevents overspending payment. Moreover, in PayWord system, whenever a customer wants to establish transactions with each vendor, s/he has to obtain a certificate from a broker and create a series of paywords, while a customer is able to make transactions with different merchants by performing only one hash chain operation in Kim and Lee's scheme. Nevertheless, we observe the following limitation on this scheme:

- the system performance is reduced by necessarily frequent signing in each transaction;
- the customer has to keep different hash chains and corresponding indices, however the overhead of merchants is relatively high. To securely deposit, the bank has to collect all paywords belonging to the same chain. It needs an additional storage space and wastes undetermined waiting time; and
- a dispute arises if the merchant forges transaction records or the customer double spends.

2.3 Hwang and Sung's scheme

Hwang and Sung [8] proposed a micro-payment scheme using hash chain [20] that provides customer's anonymity by applying a blinding signature using elliptic curve cryptography [11]. As we propose an improved version of this scheme in next section, which provides more efficiency than its current performance, we explain a bit more on this scheme so that a straightforward comparison can be established while measuring performance of the schemes in section 4.2. The scheme is divided into four phases: registration phase, blinding phase, transaction phase, and redemption phase.

Registration Phase: Both customer and merchant have to register with a broker. The customer shares a secret key K_{UB} with the broker, and the merchant shares a secret key K_{MB} with the broker. The customer selects a pseudonymous identity ID_U, which is unique to every customer. The broker selects a master secret key k and keeps it secret from others.

Blinding Phase: The customer sends a withdrawal request to the broker before s/he requests any service from merchant. The phase works as follows:

Step 1: U sends $\{ID_U, I_U\}$ to B. After checking ID_U, B computes $R' = k \cdot P$ and sends R' to U.

Step 2: Upon receiving R', U selects a random number w_n and creates a hash chain
$w_n, w_{n-1}, \ldots, w_1, w_0$, where $w_i = h(w_{i+1})$, $i = n-1, n-2, \ldots, 1, 0$, and n is the limited amount that B allows U to spend.
Then U computes $R = uR' + vP$, where P is a generator point on elliptic curve
$m = h(R \,||\, w_0)$
$m' = m/u$,
where u and v are two secret random numbers and sends $\{m', n\}K_{UB}$ to broker.

Step 3: If n is smaller than the limited amount then broker calculates $S' = SK_B \cdot m' + k$ and sends it to the customer. Otherwise, B rejects U's request.

Step 4: Upon receiving S', U calculates $S = S'u + v$ and checks whether $S \cdot P = m \cdot PK_B + R$
If it does hold, s/he obtains a valid signature (R, S) on message m.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 4 / ISSUE 1 / APRIL 2009 / 91-100

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000100008

Step 5: Broker now creates two factors: T_U and S_U , where
$T_U = h(ID_U, r_B)$
$S_U = \{s_i \mid s_i = h(s_i + 1, T_U), i = N\text{-}1,\ldots, 0\}$, where N is the maximum number of merchants that a customer can do her/his business.

Transaction Phase: U asks service from M by the following steps of operation:

Step 1: U sends transaction request $\{A_M, ID_U, ID_B\}K_{UB}$ to B.

Step 2: B maintains a table of each U's ID_U and knows the secret key K_{UB}. Now B decrypts the request and checks U's authenticity. If U is authenticated, B creates a one-time session key K_{UM} for U and M, and sends $\{K_{UM}\}K_{UB}$ to U.

Step 3: Upon receiving K_{UM}, U calculates $R_{UM} = h(w_j \oplus (s_k \,||\, K_{UM}))$ and sends $\{R_{UM}, (R, S, m), w_0, (w_j , t), s_k, OI, Exp\}K_{UM}$ to M, where $t = j - i + 1$, and Exp denotes the date after which the hash chain is invalid. The Exp can limit the length of time and both merchant and broker need to store information about the state of a hash chain.

Step 4: M verifies the signature and R_{UM} as follows:
$S \cdot P = m \cdot PK_B + R$
$w_{n-1} = h(w_n)$, where $n = j\text{-}1, j\text{-}2, t, 1$
$h^{t-1}(w_j) = h^{t-2}(w_{j-1}) = \ldots = h(w_{j-t+2}) = w_{j-t+1}$
$R'_{UM} = h(w_{j-t+1} \oplus (s_k \,||\, K_{UM}))$
If these checks hold, M will sell electronic items or services to U.

Redemption Phase: M would carry out the redemption process with B by the following steps.
Step 1: M sends $\{R_{UM}, (R, S, m), w_0, (w_j , t), s_k, OI, Exp\}K_{MB}$ for redemption to B.

Step 2: B checks the validity of date, and verifies the blind signature. Then B verifies each paywords (w_j , t). This process is the same as in Step 4 of the transaction phase. Finally, B extracts the payment from U's account and transfers the amount to M's account.

Our observations: The scheme supports multiple merchant payments and prevents overspending payment. Further, the scheme provides an important property, namely, anonymous cash transaction, which is strongly related to the practical applications in electronic business. However, to achieving anonymity property, the scheme uses a blinding phase using elliptic curve cryptography, which increases overall cost of the scheme. This motivates us to work further for an efficient blinding phase which provides both efficiency and security retaining all other features of Hwang and Sung's scheme intact.

3 An Improvement of the Hwang and Sung's scheme

We present a blinding phase using the RSA-based blinding signature [3], [5]. We show that this refinement makes

Hwang and Sung's scheme more efficient, retaining all other features intact.

Blinding Phase: The customer sends a withdrawal request to the broker before s/he requests any service from merchant. The phase works as follows:

Step 1: Broker
1.1. Choose two large primes p and q, which are secret
1.2. Compute $\lambda = pq$
$\varphi(\lambda) = (p\text{-}1)(q\text{-}1)$
1.3. Select public key e such that $1<e<\varphi(\lambda)$ and $\gcd(e, \varphi(\lambda)) = 1$
1.4. Compute private key d satisfying
$ed \equiv 1 \bmod \varphi(\lambda)$

Step 2: Customer
2.1. Choose random numbers r and u
2.2. Compute $\alpha = r^e h(w_0)(u^2+1) \bmod \lambda$
2.3. Send (a, α) to the broker
Here the information 'a' can represent the expiry date, the amount of cash (upper limit) that the customer can use, the value of each hash coin.

Step 3: Broker
3.1. Choose a random factor $x < \lambda$
3.2. Send x to the customer

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 4 / ISSUE 1 / APRIL 2009 / 91-100

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000100008

Step 4: Customer
4.1. Select a random number r'
4.2. Compute $b = r \cdot r'$
4.3. Send $\beta = b^e (u\text{-}x) \bmod \lambda$ to the broker

Step 5: Broker
5.1. Compute $\beta^{-1} \bmod \lambda$
$t = h(a)^d (\alpha(x^2+1)\beta^{-2})^{2d} \bmod \lambda$
5.2. Send (β^{-1}, t) to the customer

Step 6: Customer
6.1. Compute $c = (ux + 1)\cdot\beta^{-1}\cdot b^e = (ux+1)(u\text{-}x)^{-1} \bmod \lambda$
6.2. Compute $s = t \cdot r^2 \cdot (r')^4 \bmod \lambda$
The tuple (a, c, s) is the signature on message w_0. Any one can verify this signature by checking whether $s^e \equiv h(a)h(w_0)^2(c^2+1)^2 \bmod \lambda$.

Correctness: $s^e \equiv (t \cdot r^2 \cdot (r')^4)^e \bmod \lambda$
$\equiv (h(a)^d(\alpha(x^2+1)\beta^{-2})^{2d} r^2 (r')^4)^e \bmod \lambda$
$\equiv (h(a)^d(r^e h(w_0)(u^2+1)(x^2+1) r^{-2e} (r')^{-2e}(u\text{-}x)^{-2})^{2d} r^2 (r')^4)^e \bmod \lambda$
$\equiv (h(a)^d(r^e h(w_0)(u^2x^2+x^2+u^2+1) r^{-2e}(r')^{-2e}(u\text{-}x)^{-2})^{2d} r^2 (r')^4)^e \bmod \lambda$
$\equiv (h(a)^d(r^e h(w_0)((ux+1)^2+(u\text{-}x)^2) r^{-2e} r^{r2e}(u\text{-}x)^{-2})^{2d} r^2 (r')^4)^e \bmod \lambda$
$\equiv (h(a)^d(r^e h(w_0)(c^2+1) r^{-2e}(r')^{-2e})^{2d} r^2 (r')^4)^e \bmod \lambda$
$\equiv h(a)h(w_0)^2(c^2+1)^2 \bmod \lambda$

4 Analysis of the Improved Scheme

This section has two subsections, Security and Efficiency analysis, as explained below.

4.1 Security Analysis

The improved phase resists the following possible threats:

Unforgeability: An adversary cannot derive forged signatures. To successfully pass the verification equation $s^e \equiv h(a)h(w_0)^2(c^2+1)^2 \bmod \lambda$, the adversary has to compute s such that $s \equiv h(a)^d h(w_0)^{2d}(c^2+1)^{2d} \bmod \lambda$, given the values $h(a)$, $h(w_0)$ and c. However, it is computationally infeasible to acquire the value of d without the factorization of λ, which is an intractable problem. On the other hand, given s, $h(a)$ and $h(w_0)$, it is infeasible to compute c such that $c^2 \equiv (s^e \cdot h(a)^{-1} \cdot h(w_0)^{-2})^{1/2}\text{-}1 \bmod \lambda$ without factoring of λ. Given a and c, the adversary is unable to acquire s' such that $s' \equiv s \cdot h(w_0)^{-2d} h(w_0')^{2d} \bmod \lambda$ without knowing d. Without factoring λ, it is infeasible to get c' such that $c'^2 \equiv (s^e \cdot h(a)^{-1} \cdot h(w_0')^{-2})^{1/2}\text{-}1 \bmod \lambda$. It is also difficult to derive message w_0' with $w_0' \equiv w_0 \bmod \lambda$ such that $h(w_0) \equiv h(w_0') \bmod \lambda$, as $h(\cdot)$ is a cryptographically secure hash function. Therefore, the adversary cannot succeed to forge the signature.

Unlinkability: For any given valid signature (a, c, s), no one except the requester is able to link the signature to its previous signing instance. This implies that the signer is unable to find the link between the signature and its corresponding signing process instance.

Randomization: The signer randomizes the blinded data using a random factor x before signing it in the signing phase. This randomization property keeps the blind signature scheme away from some chosen text attacks.

Double Spending Detection: Our scheme adopts the same transaction phase as in Section 2.3. U sends $\{R_{UM}$, (a, c, s), w_0, (w_j , t), s_k, OI, Expire$\}K_{UM}$ to M before taking service from M. The payment root R_{UM} is equal to $h(w_j \oplus (s_k \,||\, K_{UM}))$. We note that the s_k, K_{UM} will be different in every purchase. As a consequence, B would be able to detect double spent paywords if U expends double the paywords.

Forgery Prevention: U obtains B's signature on w_0 before any transaction. The blind signature is based on RSA algorithm, which is widely used a secure signature algorithm. Besides, in order to process a correct redemption, M must have knowledge of the payment information. It is practically impossible for someone to forge U's paywords without knowing the secret key K_{UM} and K_{MB}.

Multiple Payment: In the transaction phase, U sends a request to B to get K_{UM} and creates the payment root $R_{UM} = h(w_j \oplus (s_k \,||\, K_{UM}))$ where w_j is the first unused payword in the paywords sequence. Consequently, every time when U makes a purchase, the R_{UM} is not the same, which enables U to make payments with multiple merchants.

Journal of Theoretical and Applied Electronic Commerce Research
ISSN 0718–1876 Electronic Version
VOL 4 / ISSUE 1 / APRIL 2009 / 91-100

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000100008

4.2 Efficiency

In a micro-payment system, the profit gained by a merchant is small in each transaction. It is unwise to verify the transaction using a complicated method that leads the average cost of the system exceeding the profit. In other words, large computation in micro-payment is not advisable. In order to measure efficiency of our improvement, we compare the improved blinding phase with the Hwang and Sung's scheme [8]. The computational complexity of the remaining phases remains same in both schemes. We use the following notation to measure the efficiency of the schemes.

t_h : computation time for hash operation
t_a : computation time for modular multiplication (or point addition in elliptic curve)
t_e : computation time for modular exponentiation (or scalar multiplication of a point on elliptic curve)
t_s : computation time for symmetric key encryption

Table 2: Computational complexity in blinding phase

	Blinding Phase
Hwang and Sung scheme [8]	$4t_h + 7t_e + 3t_a + 1t_s$
Improved scheme	$2t_h + 6t_e$

As modular exponentiation is an expensive operation in comparisons with addition or hash operation, it is easy to see from Table-2 that our improvement is efficient than Hwang and Sung's scheme. Moreover, if one selects low public exponent *e*, say 3, then our scheme becomes more efficient. This makes public key operations faster while leaving private key operations unchanged. In that case, if one employs the low public exponent attack, s/he cannot succeed to this attempt because each signature is being randomized by some random factors. Therefore, the improved scheme reduces expensive exponential operation and achieves computational efficiency.

5 Conclusions

We discussed the requirements of micro-payment and review two micro-payment systems based on PayWord [18] and a scheme [8] based on hash chain. The hash-chain based scheme provides anonymity security property in addition to other security requirements of micro-payment system. The use of one-way hash function and the blind signature in [8] makes the scheme efficient and ensures the paywords untraceable. However, we observe that the blinding phase of the scheme in [8] takes significantly more computation effort and we present an alternate blinding phase using the RSA signature algorithm that achieves more efficiency than the existing ones. Although the improved scheme requires long key size, at least 1024-bit, in comparison with 160-bit key of its elliptic curve cryptography version, but we believe that computation and speed are two important requirements than storage cost, and in this context, the improved scheme would provide significant advantage for small value payments. The research work carried out in this paper has enormous future prospects and can be extended towards a light-weight scheme with only hash function and message authentication codes so that the costly operation, modular exponentiation, can be avoided and at the same time similar security strength can be achieved.

Websites List

Site 1: The Digicash Wallet. DigiCash Inc.
http://www.digicash.com

Site 2: Micropayments Overview. The W3C Ecommerce/Micropayment Working Group.
http://www.w3.org/ECommerce/Micropayments/

References

[1] Baddeley, M. Using e-cash in the new economy: An economic analysis of micro-payment systems. Journal of Electronic Commerce Research, vol. 5, no. 4, 2004
[2] Brands, S. Untraceable off-line cash in wallet with observers. In Proc. of the International Conference of Advances in Cryptology–CRYPTO'93, LNCS 773, Springer -Verlag, 302–138, 1993.
[3] Chaum, D. Blind signatures for untraceable payments. In Proc. of the International Conference of Advances in Cryptology, Plenum Press, pp. 199-203, 1982.
[4] Chaum, D. A Fiat and M. Naor. Untraceable electronic cash. In Proc. of Advances in Cryptology, LNCS 403, Springer-Verlag, pp. 319-327, 1988.
[5] Chien, H. Y., Jan, J. K. and Tseng, Y. M. RSA-based partially blind signature with low computation. In Proc. of the International Conference in Parallel and Distributed Systems, pp. 385–389, 2001.
[6] Foo, E and Boyd, C. A payment scheme using vouchers. In Proc. of the International Conference of Financial Cryptography, LNCS 1465, Springer-Verlag, pp. 103-121, 1998.

This paper is available online at
www.jtaer.com
DOI: 10.4067/S0718-18762009000100008

[7] Glassman, S., Manasse, M. S., Abadi, M., Gauthier, P. and Sobalvarro, P. The Millicent protocol for inexpensive electronic commerce. In Proc. of the International World Wide Web Conference, pp. 603–618, O'Reilly, 1995.
[8] Hwang, M. S. and Sung, P. C. A study of micro-payment based on one-way hash chain. International Journal of Network Security, vol. 2, no. 2, pp 81-90, 2006.
[9] Jakobsson, M., Hubaux, J. P. and Buttyan, L. A micro-payment scheme encouraging collaboration in multi-hop cellular networks. In Proc. of Financial Cryptography, LNCS 2742, Springer-Verlag, pp. 15–33, 2003.
[10] Kim, S. and Lee, W. A PayWord-based micro-payment protocol supporting multiple payments. In Proc. of the International Conference on Computer Communications and Networks, pp. 609-612, 2003.
[11] N. Koblitz. Elliptic Curve Cryptosystems. Mathematics of Computation, vol. 48, pp. 203-209, 1987.
[12] Lipton, R. J. and Ostrovsky, R. Micro-payments via efficient coin-flipping. In Proc. of the International Conference of Financial Cryptography, LNCS 1465, Springer Verlag, pp. 1-15, 1998.
[13] Mu, Y., Nguyen, K. Q. and Varadharajan, V. A fair electronic cash scheme. In Proc. of the International Symposium in Electronic Commerce, LNCS 2040, Springer- Verlag, pp. 20–32, 2001.
[14] Mu, Y., Varadharajan, V. and Lin, Y. New micropayment schemes based on payWords. In Proc. of the Australasian Conference on Information Security and Privacy, LNCS 1270, Springer-Verlag, pp. 283–293, 1997.
[15] Odlyzko, A. The practical problems of implementing Micromint. In Proc. of the International Conference of Financial Cryptography, LNCS 2742, SpringerVerlag, pp. 77-83, 2003.
[16] O'Mahony, D., Tewari, H. and Peirce, M. Electronic Payment Systems. Artech House, Inc, 1997.
[17] Rivest, R. Electronic lottery tickets as micropayments. In Proc. of the International Conference of Financial Cryptography, LNCS 1318, Springer-Verlag, pp. 307–314, 1997.
[18] Rivest, R. and Shamir, A. PayWord and MicroMint - two simple micro-payment schemes. In Proc. of the Security Protocols International Workshop, LNCS 1189, Springer-Verlag, pp. 69-87, 1996.
[19] Rivest, R., Shamir, A. and Adleman, L. A Method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, vol. 21, no. 2, pp. 120-126, 1978.
[20] Schneier, B. Applied Cryptography. John Wiley & Sons, 1996.
[21] Shirky, C. The case against micro-payments. OpenP2P O'Reilly Network, 2000.
[22] Stern, J. and Vaudenay, S. SVP: A flexible micro-payment scheme. In Proc. of Financial Cryptography, LNCS 1318, Springer-Verlag, pp. 161-172, 1997.
[23] van Someren, N. The practical problems of implementing Micromint. In Proc. of the International Conference of Financial Cryptography, LNCS 2339, Springer-Verlag, pp. 41-50, 2001.
[24] van Someren, N., Odlyzko, A., Rivest, R., Jones, T. and Scot, D. G. Does anyone really need micropayments? In Proc. of the International Conference of Financial Cryptography, LNCS 2742, Springer-Verlag, pp. 69-76, 2003.
[25] Wang, C., Chang, C. and Lin, C. A new micro-payment system using general payword chain. Electronic Commerce Research Journal, vol. 2, no. 1-2, pp. 159-168, 2002.
[26] Yen, S., Ho, L. and Huang, C. Internet micro-payment based on unbalanced one-way binary tree. In Proc. of the International Conference of Cryptec'99, 155-162, 1999.