Improved Probabilistic Micropayment Scheme

Sung-Ming Yen[1]*, Chien-Ning Chen[1], Hsi-Chung Lin[1], Jui-Ming Wu[2], and Chih-Ta Lin[2]

[1] Laboratory of Cryptography and Information Security (LCIS)
Department of Computer Science and Information Engineering
National Central University
Chung-Li, Taiwan 320, R.O.C.
{yensm, ning, hclin}@csie.ncu.edu.tw

[2] Networks and Multimedia Institute
Institute for Information Industry
Taipei, Taiwan, R.O.C.
{raymond, cheetah}@nmi.iii.org.tw

Received 16 October 2007; Revised 30 November 2007; Accepted 9 December 2007

Abstract. Micropayment schemes are recently considered to have a variety of practical applications. Such kind of schemes handle a sequence of very small payments which therefore should be quite efficient to be payed and verified. An electronic lottery ticket scheme was proposed by Rivest as a means of probabilistic micropayment scheme in order to reduce the administrative cost of the bank. In this paper, the above probabilistic micropayment scheme is extensively studied and get improved. Analysis shows that the scheme by Rivest may reduce the administrative cost of the bank, however it brings extensive computational overhead to the merchant. A proposed simple modification improves the performance of Rivest scheme extensively. Furthermore, a novel development of self-randomized winning number generation is proposed in order to improve the fairness of the first proposed probabilistic micropayment scheme.

Keywords: Electronic lottery ticket, Electronic payment, Fairness, Micropayment, One-way hash function, Probabilistic micropayment.

1 Introduction

Internet *micropayment* schemes have received growing attention recently, mostly due to the fact that these schemes exhibit the potential of being embedded in numerous Internet based applications. As a special type of electronic payments, micropayment schemes allow a customer to pay a merchant a sequence of small payments over the computer network in exchange for services or electronic products from the merchant. With these services or products, it is often inappropriate to pay the total amount of money either in advance or afterwards. This is particularly true in certain cases where real time bargaining results in the requirement of a small payment being received and verified by the merchant. Evidently, in such a scenario of payment, all the following costs should be minimized:

(1) Computational cost: This cost should be comparable with the value to be paid. Therefore, the invocation of public key computation should be prevented or at least be kept as limited to large amount of payment as possible.
(2) Storage cost: Since there will be a large amount of payment to be handled and each of which is of a tiny value, it is not feasible to keep a record of each payment. This will probably make the cost of processing the payment exceeding the value of the transaction.
(3) Administrative cost: This includes the minimization of interactions with the trusted third party, usually the bank, the frequency of doing withdrawal and deposit.

* Correspondence author

Possible practical applications of the above micropayment model include digital newspaper [1], on-line journal subscription, on-line database query, multimedia entertainment over the Internet, and Internet advertisement (say via lottery tickets [2] and it is the primary consideration of this paper). More examples can be found in [2, 3]. In addition, accounting and pricing for Internet services and mobile telecommunication may represent yet another set of promising applications of micropayments [4–8].

Some notable representatives of micropayment schemes can be found in the literature [2, 3, 9–14]. The fundamental cryptographic tool for most of these payment systems is a one-way hash chain which has widely been known by researchers ever since Lamport first proposed its use in one-time passwords [15, 16].

Electronic lottery ticket was proposed in [2] as a means of probabilistic micropayment scheme in order to reduce the administrative cost of the bank. This new technique also enables the Internet advertisement. In an Internet advertisement, in a completely reverse scenario the merchant may wish to pay every time some small amount of money to his customers who visit the commercial advertising home page and fill some questionaries. Since each payment will be very small, it is not appropriate to pay by using a general purpose electronic payment systems. In this situation, an electronic lottery ticket as a payment will be very useful and will even promote to the commercial advertisement.

Due to the importance and potential applications of electronic lottery ticket as probabilistic micropayment, in this paper we analyze the scheme proposed in [2]. The result is that the scheme in [2] may reduce the administrative cost of the bank, however the scheme brings extensive computational overhead to the merchant. The main contribution of this paper is that two improved versions of electronic lottery tickets are proposed which solve the mentioned disadvantage in [2].

2 Micropayment System Based on Cryptographic Hash Chain

Some notations and symbols about a one-way hash chain are reviewed in the following.

Notation 1 *When a function h is iteratively applied r times to an argument X_n, the result will be denoted as $h^r(X_n)$, that is*

$$h^r(X_n) = \underbrace{h(h(\cdots(h}_{r \text{ times}}(X_n))\cdots)).$$

When the function $h()$ in the iteration is instantiated with a one-way hash function, such as MD5 [17] and SHA [18], the result is a *one-way hash chain* as shown in Fig. 1. Note that within the chain, each element X_i is computed as $h^{n-i}(X_n)$.

$$X_0 = h^n(X_n) \leftarrow X_1 = h^{n-1}(X_n) \leftarrow \cdots \leftarrow X_{n-1} = h^1(X_n) \leftarrow X_n$$

Fig. 1. One-way hash chain.

2.1 Review of PayWord Micropayment Scheme

The PayWord micropayment scheme [3] which is mainly based on the idea of using a one-way hash chain, will be briefly reviewed in the following.

Prior to the first transaction between a customer and a merchant, the following preparatory steps need to be carried out.

(1) The customer generates a *payment chain* as follows:

$$X_0 \leftarrow X_1 \leftarrow X_2 \leftarrow \cdots \leftarrow X_{n-1} \leftarrow X_n$$

where $X_i = h(X_{i+1})$ for $i = n-1, n-2, \cdots, 1, 0$, and $h()$ is a cryptographic one-way hash function. The value X_n is a secret value selected at random by the customer.

(2) The customer signs, e.g., using RSA [19], on the root X_0, together with the merchant's identity and other pieces of information (if required):

$$Sign_C(\text{Merchant-ID}||X_0||\text{Cert})$$

where "Cert" used as a proof of credentials, is a digital certificate issued to the customer by a bank. Note that the signature on X_0 acts as a commitment.

(3) The customer then sends

$$Sign_C(\text{Merchant-ID}||X_0||\text{Cert}), \text{Merchant-ID}, \text{Cert}, X_0$$

to the merchant.

After completing successfully the above steps between the customer and the particular merchant, the number X_i ($i = 1, 2, \ldots, n$) can now be used as the ith coin to be paid. When receiving a new coin X_i form the customer, the merchant verifies whether $X_{i-1} \stackrel{?}{=} h(X_i)$. The merchant accepts X_i as a valid payment only if the verification is successful. Note that the merchant can store a valid X_i in place of X_{i-1}.

2.2 Performance Analysis of PayWord

In the following, efficiency analysis is given. Suppose that the customer only stores the last coin X_n for space saving reason (a reasonable assumption for many practical environments, e.g., a smart IC card with few memory space), then he has to compute each necessary new coin when required. This consists of a sequence of hash function computation. It is clear that the computation of X_1 costs $n-1$ hash computation and X_2 costs $n-2$ hash computation, and so on. On average, each new coin generation will cost

$$((n-1) + (n-2) + \ldots + 1)/n = (n-1)/2$$

hash computation. If we consider also the computation of root X_0, the average cost will be $(n+1)/2$. At the merchant's side, it always takes one hash function computation to verify the validity of each received new coin. Note that at the user's side, the computational complexity of each new coin generation is $\mathrm{O}(n)$, where n is the length of the underlying hash chain. With some hash chain traversal techniques, the complexity can be reduced to $\mathrm{O}(\log^2 n)$ [20]. Simple idea to slightly improve the performance by adding one or more midway points has been considered in [21].

Evidently, the computational efficiency can be enhanced extensively if a much smaller value of n is chosen. Unfortunately, in a practical application, this setting will eventually slow down the overall system performance because that it requires the customer to generate public key based signature much more frequently. Recall that the most important issue of PayWord is the minimization of using public key cryptography.

3 Probabilistic Micropayment Scheme

Although the above micropayment system is efficient, there are still some problems to be considered and resolved. In the conventional paper cash system, the very large amount of micropayment transferred between the customers and the merchants does not need the invocation of the trusted third party, usually the bank. However, in the electronic micropayment system, each payment chain should be processed by the bank and as mentioned before the parameter n should not be too large for the sake of performance. So, a moderately large amount of signatures will be transfered from a customer to a merchant during a period of payment process, then will be transfered from a merchant to the bank during the clearing process. Therefore, it will be much beneficial to the bank if the signed commitments $Sign_C(X_0)$ can be verified/processed in a batch approach in such a way that a large group of signatures can be verified together. Some signature schemes suitable for batch verification [22, 23] are what required for this purpose. However, the batch processing will induce a long delay before the deposit of many payment chains issued by a specific customer can be handled. This could be impractical for many applications and will somewhat increase storage cost for the merchants. Recall that storage space is one of the costs to be minimized for a micropayment system.

Although the third party (the bank) should handle every transaction in clearing process, should the third party attend every transaction on-line (during the purchase)? In systems that a transaction succeeds only if the bank approves it, there is no way for a customer to exceed his solvency, but the communication cost is large. As mentioned before, communication cost is part of the administrative cost which should be minimized. In contrast, systems that do transaction off-line (the third party does not attend in the purchase process), the communication cost is reduced while the bank suffers the risk of user's overdrawing or other malicious behavior (e.g., abuse of certificate). In fact, it is the bank's trade-off between risk and efficiency. In [24], Jarecki and Odlyzko proposed a micropayment scheme based on probabilistic operation which tries to find a point with acceptable risk and feasible efficiency in that trade-off. Furthermore, in [2] a better solution with extensively small computational cost was provided and will be discussed later.

3.1 Electronic Lottery Tickets

A probabilistic micropayment scheme was proposed in [2] which was motivated from the idea of issuing lottery ticket. The scheme tries to simplify the work of the bank and was claimed efficient since the bank handles only the winning tickets, instead of all tickets (as micropayment).

In this probabilistic micropayment scheme, the lottery tickets issuer (it is often a customer with a ticket issuing certificate received from the bank) can issue electronic lottery tickets with winning prize of $10.00 and each ticket with a winning probability of $1/1000$. Then, each lottery ticket has an expected value of one cent and can be employed as a micropayment of one cent. From the bank's view-point, such kind of payments will be much efficient because the bank only has to manage the winning lottery tickets issued by customers.

In the scheme [2], the ticket construction is itself a digital signature of the following message (list only the most important parts).

(1) The root of a cryptographic hash chain as in PayWord.
(2) The name of the issuer/customer who creates the electronic lottery tickets. Also, the name of the recipient/merchant.
(3) A winning number indicator that indicates how the winning number will be determined for a specific payment chain or for all payment chains.
(4) A ticket face value that specifies the payment to be received if the lottery ticket comes up to be a winner.

Generally, there are two types of winning number indicators, one of the internal type and the other the external approach [2].

(a) External winning indicator: This is much like the conventional lottery ticket operation system. An external indicator refers to some source or authority who will announce a winning number in a specified date.
(b) Internal winning indicator: A straightforward example of an internal indicator is the last few digits, say 3 digits, of a 30-digit decimal number w whose MD5 or SHA hash value is $h(w)$. The winning number w is randomly selected by the recipient but he will send the hash value $h(w)$ to the issuer. So, w is kept secret from the issuer. The issuer then includes $h(w)$ into the commitment/signature when creating a lottery ticket chain.

There are some trade-off between the above two winning indicator methods. If the merchant wishes to know whether each received ticket is a winner, the internal indicator approach is the choice. However, it is obvious that interaction between the customer and the merchant is required in order to setup valid lottery tickets. On the other hand, a simpler system can be obtained with external indicator. However, in this design, the merchant suffers from the risk that the issuer may unable to pay afterwards. Also, all the received tickets should be stored before the winning number will be published.

Primary constructions of probabilistic micropayment in [2] are illustrated which are all directly related with PayWord [3]. Another probabilistic micropayment scheme proposed in [25] is a variant of [2]. It eliminates the hash chain structure and the two-way interaction between the user and the merchant, but each transaction requires two digital signatures and related verifications.

3.2 External Indicator Based Approach: The Protocol EI-1

In this protocol, the customer constructs a payment chain as usual and gives the commitment of this chain to the merchant. In the commitment, an announcement of the winning policy WP, e.g., indicating the source of winning number, will be included as $Sign_C(X_0||WP)$.

Two primary drawbacks of the external indicator approach are [2]:

(a) Collaboration between the customer and the source of issuing winning number, e.g., the bank, should be prevented. The better way is to include more independent sources.
(b) The merchant must store all received electronic lottery tickets until the related winning numbers are revealed and the tickets are checked.

3.3 Internal Indicator Based Approach-1: The protocol II-1

This protocol needs an initial interaction between the customer and the merchant in order to construct valid lottery tickets. The merchant selects the winning number w in random (in the protocol, only a portion of w will be considered as the winning number, e.g., the last three decimal digits), but he will send $h(w)$ to the customer. The customer constructs a payment chain and sends the following commitment

$$Sign_C(X_0||h(w))$$

to the merchant. Each micropayment X_i is defined to be a winning lottery ticket if $X_i \equiv w \pmod{1000}$. Here, we ignore the details of the bit string to integer conversion function, BS2IP [26], which converts hash function's outputs, X_i and w, into integers. Since the merchant knows the winning number w, he can get *immediate* information about whether the received micropayment X_i is a winning ticket. An immediate clearing process for a winning ticket can be conducted between the merchant and the bank by showing $Sign_C(X_0||h(w))$, X_0, X_i, and w. On the other hand, all unnecessary commitments and tickets can be totally removed from the merchant's machine. Therefore, both the two primary drawbacks of protocol EI-1 can be prevented.

However, a disadvantage still remains in this protocol II-1. Showing or releasing w ($w \bmod 1000$ is sufficient) enables the customer to decide whether each of unspent coins is a winning ticket. This of course somewhat destroys the rule of lottery tickets using as payment.

3.4 Internal Indicator Based Approach-2: The protocol II-2

In this protocol, the customer constructs a payment chain and the most interesting design is that the merchant also constructs a *winning number chain* as

$$W_0, W_1, W_2, \ldots, W_n$$

where $W_i = h(W_{i+1})$ for $i = 0, 1, \ldots, n-1$. The merchant gives the root W_0 of this winning number chain to the customer and the customer computes and sends the commitment

$$Sign_C(X_0||W_0)$$

with the root X_0 to the merchant.

The i-th ticket X_i in the payment chain is a winning ticket if and only if $X_i = W_i \pmod{1000}$. With the knowledge of W_i, the merchant can immediately inform the bank and receives money if X_i wins. Note that delivering W_i will not release the knowledge of W_{i+1}, therefore this protocol overcomes the disadvantage of protocol II-1.

4 Improved Probabilistic Micropayment Scheme Using Internal Indicator – The Protocol II-3

Although the protocol II-2 seems to solve all disadvantages of probabilistic micropayment protocol, especially the protocol II-1, however the protocol II-2 is evidently much less efficient for the merchant as well as less efficient for both the customer and the bank (when compared with the protocol II-1). An improved version with internal indicator allowing *immediate clearing* request from the merchant will be proposed in this section.

4.1 Analysis of the Protocol II-2

It should be noted that X_i is defined to be a winning lottery ticket if $X_i = W_i \pmod{1000}$. This implies that the merchant now needs almost the same computational complexity to check each received ticket as what the customer does to generate each ticket to be paid. In fact, the merchant takes one more hash computation for each ticket checking, i.e., to check whether $X_{i-1} = h(X_i)$. On average, for the merchant's side, each received ticket will cost

$$(((n-1) + (n-2) + \ldots + 1)/n) + 1 = (n+1)/2$$

hash computation to be verified for its validity and winning status. Recall that in all other previous protocols (except protocol II-2), the merchant needs one hash function computation to verify the validity and winning status of each received ticket. The extensive overhead for merchants becomes impractical for many applications, especially for those merchants that provide real-time service to extremely many customers at the same time, e.g., accounting and pricing servers for Internet services and mobile telecommunication.

In all other previous protocols (except protocol II-2), the bank needs to verify the onewayness relationship of a winning payment chain. However, in protocol II-2, the bank also needs to verify the onewayness relationship of the related winning number chain. The customer also suffers from this computational overhead.

Because of the potential importance of probabilistic micropayment and especially under the model of *immediate clearing* requirement, development of efficient such protocols are necessary. A possible improved protocol will be proposed in the following.

4.2 The Improved Protocol II-3

In this improved protocol, the customer constructs a payment chain (with length n) as

$$X_0, X_1, X_2, \ldots, X_n$$

where $X_i = h(X_{i+1})$ for $i = 0, 1, \ldots, n-1$, and the merchant constructs a winning number chain (with length m) as

$$W_0, W_1, W_2, \ldots, W_m$$

where $W_i = h(W_{i+1})$ for $i = 0, 1, \ldots, m-1$. Note that the length of the winning number chain can be much shorter than the length of the payment chain, i.e., $m \ll n$. This is the first primary difference between the protocol II-2 and the protocol II-3. The merchant first sends the root W_0 of the winning number chain to the customer and the customer prepares the commitment

$$Sign_C(X_0||W_0)$$

and sends it with the root X_0 to the merchant.

The secondary difference between the protocol II-2 and the protocol II-3 is the definition of winning ticket.

Definition 1 *The kth ($k = 1, 2, \ldots$) winning ticket of the protocol II-3 is defined as X_i (accordingly $X_i \bmod 1000$ as the winning number) with a smallest integer i such that $i > I$ (where X_I is the $(k-1)$th winning ticket) and $X_i \equiv W_k \pmod{1000}$. The integer I is set to zero for the first winning ticket.*

As in the protocol II-1 and the protocol II-2, the merchant can ask for immediate clearing process on a winning ticket by showing X_i, W_k, and other related information. Notice that revealing the ith winning number W_i will not release any W_j ($j > i$), so the disadvantage of the protocol II-1 is resolved.

In this improved protocol, the selection of much shorter winning number chain with $m \ll n$ leads to an extensive performance improvement. This is a probabilistic payment protocol and it is usually with very small possibility of winning (in order to enhance the performance). Furthermore, double or more winnings within a single payment chain is extremely less frequent. The following illustrative parameters setting is reasonable for most of the cases:

(a) winning number of 3 decimal digits (e.g., defined as $W_k \bmod 1000$);
(b) payment chain of length between 100 to 500 (i.e., the parameter n);
(c) winning number chain of length between 3 to 5 (i.e., the parameter m).

In the above protocol, the merchant can compute the most fresh winning number W_k and stores it for comparing with each received new ticket. The next winning number W_{k+1} will be computed only if required. The merchant can also compute all the winning numbers in advance and stores them in a small table for later use. Both the above two approaches are efficient since m is a very small integer.

5 Enhanced Protocol with Self-Randomized Winning Number Generation – The Protocol II-4

As mentioned in Section 1, a micropayment scheme tries to minimize both cost (computation or space costs) and administrative work (e.g., work load of clearing process). The probabilistic micropayment protocol in [2] reduces the work load of the bank by avoiding to deal with every payment chain. Unfortunately, that protocol increases the computational load of the merchant extensively.

The proposed protocol II-3 in Section 4 tries to reduce the load of both the bank and the merchant at the same time, and thus solves the disadvantage of a protocol in [2], i.e., the protocol II-2. However, there is so a problem remained in the protocol II-3 that "tricks" can be employed by the customer. Note that this disadvantage also exists in the protocol II-1. In this section, the protocol II-3 will be analyzed and a further enhanced protocol will be proposed.

5.1 Security Analysis of the Protocol II-3

A probabilistic micropayment protocol takes advantage of the nature of being probabilistic. In the proposed protocol II-3 (and also the protocol II-1 [2]), a greedy customer can obtain some advantages from a simple statistical analysis and makes the payment protocol be somewhat unfair.

Suppose that all the first t lottery tickets X_i ($i = 1, 2, \ldots, t$) are all distinct in their last 3 digits and all these tickets are not winning tickets, the customer could store all the $X_i \bmod 1000$ ($i = 1, 2, \ldots, t$) in a table for further analysis. We also assume that the hash function is a random permutation, so each value in that payment chain is an outcome of fair coin toss. Then, the following two cases happen on the next ticket value X_{t+1}:

(1) If $X_{t+1} \bmod 1000$ is not in the table, then the customer knows that the expected probability for X_{t+1} to be a winning ticket is $\frac{1}{1000-t}$ ($> \frac{1}{1000}$). Instead of suffering the larger probability to pay the merchant, the customer may discard the following tickets within this payment chain for some reasons (for example to claim the loss of the secret X_n) and pays with a new payment chain. This is especially noticeable for large value of t.
(2) If $X_{t+1} \bmod 1000$ is found within the table, then the customer can pay it without any cost since X_{t+1} is definitely not a winning ticket.

Therefore, in the protocol II-3 (and also protocol II-1), during the process of winning status checking, although the merchant does not reveal the winning number W_k directly, the merchant however releases implicitly the complementary information that the winning number is not X_i.

5.2 The Enhanced Protocol II-4 and Its Cryptanalysis

In order to avoid any possible *side-channel* of releasing the information of the current winning condition generation, one possible solution is to create different winning number for each related ticket X_i. This strategy was also employed in the development of protocol II-2 [2], but that straightforward design is not efficient at all. A novel design of *self-randomized winning number generation* is proposed in the following in order to develop a probabilistic micropayment protocol using lottery ticket without side-channel that is efficient for the customer, the merchant, and the bank.

In this enhanced protocol, the customer constructs a payment chain with length n as $X_0, X_1, X_2, \ldots, X_n$ and the merchant constructs an *implicit* winning number chain with length m as $W_0, W_1, W_2, \ldots, W_m$ where $m \ll n$. The merchant sends W_0 to the customer and the customer prepares the commitment as $Sign_C(X_0||W_0)$, then he sends the commitment and the root X_0 to the merchant.

The difference between the proposed protocol II-3 and this enhanced version II-4 is the adoption of the self-randomized winning number generation.

Definition 2 *The kth ($k = 1, 2, \ldots$) winning ticket of the protocol II-4 is defined as X_i (accordingly $X_i \bmod 1000$ as the winning number) with a smallest integer i such that $i > I$ (where X_I is the $(k-1)$th winning ticket) and $X_i \equiv h(i||X_i||W_k) \pmod{1000}$. The integer I is set to zero for the first winning ticket.*

Instead of using a fixed value $W_k \bmod 1000$ directly derived from W_k as the kth winning number, a varying random value

$$h(i||X_i||W_k) \bmod 1000$$

is computed as the kth winning number which is dependent on W_k (a fixed value), the payment ticket value X_i, and its index value i. Therefore, even if a same W_k is considered for determining the winning status of many payment tickets X_i's, all the related winning numbers are randomly computed by including the X_i and the index i. Note especially that distinct winning numbers are computed even if $X_i = X_j$ ($i \neq j$) since the index values i and j are also included during the winning number generation.

Considering the properties of cryptographic hash functions, preimage resistance, 2nd-preimage resistance, and collision resistance [27], there is no direct relationship between $h(i||X_i||W_k) \bmod 1000$ and $h(j||X_j||W_k) \bmod 1000$ for distinct i and j. With this design, the customer cannot conduct any useful statistical analysis. Precisely, the following two properties hold on X_i and X_j for distinct i and j:

Property (1) If $X_i \neq X_j$ and $X_i \not\equiv h(i||X_i||W_k) \pmod{1000}$ (which means X_i is not a winning ticket), then it does not guarantee directly that $X_j \equiv h(j||X_j||W_k) \pmod{1000}$ will happen with probability larger than $\frac{1}{1000}$ since both $h(i||X_i||W_k) \bmod 1000$ and $h(j||X_j||W_k) \bmod 1000$ are statistically independent. Therefore, no table maintenance of non-winning tickets (or its related last three digits) is useful for the customer to predict that a coming payment ticket will become a winning one with larger probability than $\frac{1}{1000}$.

Property (2) If $X_i = X_j$, then it does not imply directly that $h(i||X_i||W_k) \equiv h(j||X_j||W_k) \pmod{1000}$. Contrarily, both $h(i||X_i||W_k) \bmod 1000$ and $h(j||X_j||W_k) \bmod 1000$ are statistically independent if $h()$ is a perfect cryptographic hash function. So, if X_i is not a winning ticket, then X_j ($= X_i$) will be a winning ticket with the probability of $\frac{1}{1000}$. Therefore, no table maintenance of non-winning tickets (or its related last three digits) is useful for the customer to rule out the possibility of a coming payment ticket to be a winning one.

Based on the above two properties, the winning status decision process in the protocol II-4 for every payment ticket is unpredictable and is perfectly randomized to the customer no matter if X_i and X_j ($i \neq j$) are the same.

5.3 Generalization of the Winning Status Decision Process

In the following discussions, the winning status decision process is generalized as

$$X_i \equiv f(i, X_i, W_k) \pmod N$$

where $f()$ is the winning number producing function and N is an integer, say 1000 as in all previous illustrative examples.

A straightforward selection of $f()$ is a cryptographic hash function as in all previous examples. Due to the intrinsic assumption of infeasibility of computing the inverse hash function, given the information of $h(i, X_i, W_k)$, it is computationally infeasible for the customer to find an exact value of W_k. Note that even if a possible W_k is found, it may not be the desired one since cryptographic hash function may still collide. The derived possible W_k should satisfy the property of $W_{k-1} = h(W_k)$. Without the knowledge of current W_k, it is impossible for the customer to predict the coming winning numbers $h(i+1, X_{i+1}, W_k)$. In the above statements, it is supposed that in the protocol II-4 the merchant should send $h(i, X_i, W_k)$ to the customer. This is a very strong assumption since in a practical implementation the merchant needs only to indicate whether the received payment ticket X_i is a winning ticket (or the customer can collect this binary information by some other means). More precisely, only the information of whether $X_i \equiv h(i, X_i, W_k) \pmod N$ will be available to the customer.

Based on the above discussions, in a practical application where the merchant does not send $f(i, X_i, W_k)$ directly to the customer, the winning number producing function $f()$ can be a much simpler function with weaker cryptographic assumptions than a one-way function should provide. Possible examples of such function $f()$ are simplified versions of cryptographic hash function or encryption cipher with fewer rounds or simplified operations. It can also be any dedicated design. The only requirements for this simplified computation of $f(i, X_i, W_k) \bmod N$ are both the Property (1) and the Property (2) described in the subsection 5.2.

Therefore, in a design where $f()$ is a cryptographic hash the merchant needs two cryptographic hash computations for every received coin. The first hash computation verifies whether $X_{i-1} = h(X_i)$ and the other hash computation generates the new winning number. However, in most cases where $f()$ can be much efficient than a hash function $h()$, so the merchant needs slightly more than one hash computation for every received coin. Hence, both the merchant and the customer can perform in almost the same performance as in the original PayWord scheme [3]. Furthermore, the administrative work of the bank can be reduced extensively. Totally, this achieves the goal of a probabilistic micropayment that described in [2].

6 Conclusions

In the PayWord micropayment scheme [3], a sequence of coins to be spent are prepared in the form of one-way hash chain such that the payment chain is generated in the computationally easy direction while it is payed to the merchant in the computationally hard direction. Rivest also proposed a modified version in a probabilistic approach trying to reduce the administrative and computational work of the bank. In this probabilistic PayWord scheme [2], the merchant also has to prepare a winning number chain.

As pointed out in the introduction, micropayment would play a major role in the future electronic commerce and some other possible applications. Development of secure and efficient micropayment systems for general purpose applications is a growing important issue undoubtedly. Because of the potential importance of probabilistic micropayment and especially under the model of *immediate clearing* requirement, the probabilistic PayWord scheme is thoroughly studied in this paper. Analysis shows that the straightforward extension from PayWord [3] into probabilistic PayWord [2] does not provide a satisfactory result.

Therefore, we propose an enhanced probabilistic micropayment scheme (the protocol II-4) which is secure (in terms of fairness and intractability of deriving unspent coin) as in the original probabilistic PayWord scheme and most importantly the proposed scheme is efficient for all the three parties of the payment, i.e., the customer, the merchant, and the bank. This goal has not been achieved in the original probabilistic PayWord scheme.

The question of how to develop an efficient winning number producing function $f()$ as described in subsection 5.3 is still an open research problem.

Acknowledgment

This work was supported in part by the Heterogeneous Network Security Project of Institute for Information Industry and sponsored by MOEA, R.O.C.

References

[1] J.W. Palmer and L.B. Eriksen, “Digital newspapers explore marketing on the Internet,” *Communications of ACM*, Vol. 42, No. 9, pp. 33-40, 1999.

[2] R.L. Rivest, “Electronic lottery tickets as micropayments,” *Proceedings of Financial Cryptography Conference, FC ’97*, Lecture Notes in Computer Science, Vol. 1318, Springer Verlag, pp. 307-314, 1998.

[3] R.L. Rivest and A. Shamir, “PayWord and MicroMint: Two simple micropayment schemes,” *Proceedings of Security Protocols Workshop*, Lecture Notes in Computer Science, Vol. 1189, Springer Verlag, pp. 69-87, 1997. Also in *CryptoBytes*, Pressed by RSA Laboratories, Vol. 2, No. 1, pp. 7-11, 1996.

[4] N. Daswani and D. Boneh, “Experimenting with electronic commerce on the PalmPilot,” *Proceedings of Financial Cryptography Conference, FC ’99*, Lecture Notes in Computer Science, Vol. 1648, Springer Verlag, pp. 1-16, Feb. 1999.

[5] G. Horn and B. Preneel, “Authentication and payment in future mobile systems,” *Proceedings of ESORICS ’98*, Lecture Notes in Computer Science, Vol. 1485, Springer Verlag, pp. 277-293, 1998.

[6] K.M. Martin, B. Preneel, C.J. Mitchell, H.J. Hitz, G. Horn, A. Poliakova, and P. Howard, “Secure billing for mobile information services in UMTS,” *Proceedings of 5th International Conference in Services and Networks, IS&N ’98*, Lecture Notes in Computer Science, Vol. 1430, Springer Verlag, pp. 535-548, 1998.

[7] D. O’Mahony, L. Doyle, H. Tewari, and M. Peirce, “NOMAD – An application to provide UMTS telephony services on fixed terminals in COBUCO,” *Proceedings of 3rd ACTS Mobile Communications Summit*, Vol. 1, pp. 72-76, Rhodes, Greece, June 1998.

[8] M. Peirce and D. O’Mahony, “Micropayments for mobile networks,” Technical Report of the Dept. of Computer Science, Trinity College Dublin, Ireland, 1999.

[9] R. Anderson, C. Manifavas and C. Sutherland, “NetCard – A practical electronic cash system,” *Proceedings of Security Protocols Workshop*, Lecture Notes in Computer Science, Vol. 1189, Springer Verlag, pp. 49-57, 1997.

[10] S. Glassmann, M. Manasse, M. Abadi, P. Gauthier, and P. Sobalvarro, “The Millicent protocol for inexpensive electronic commerce,” *Proceedings of 4th International World Wide Web Conference*, pp. 603-618, Boston, MA, Dec. 1995.

[11] R. Hauser, M. Steiner, and M. Waidner, “Micropayments based on iKP,” *Proceedings of SECURICOM ’96, 14th Worldwide Congress on Computer and Communications Security and Protection*, pp. 67-82, 1996.

[12] C. Jutla and M. Yung, “Paytree: amortized signature for flexible micropayments,” *Proceedings of 2nd USENIX Workshop on Electronic Commerce*, pp. 213-221, 1996.

[13] T. Pedersen, “Electronic payments of small amounts,” *Proceedings of Security Protocols Workshop*, Lecture Notes in Computer Science, Vol. 1189, Springer Verlag, pp. 59-68, 1997.

[14] S.M. Yen, “PayFair: A prepaid Internet micropayment scheme ensuring customer fairness,” *IEE Proceedings: Computers and Digital Techniques*, Vol. 148, No. 6, pp. 207-213, Nov. 2001.

[15] N.M. Haller, “The S/KEY one-time password system,” *Proceedings of the ISOC Symposium on Network and Distributed System Security*, San Diego, CA, Feb. 1994.

[16] L. Lamport, “Password authentication with insecure communication,” *Communication of ACM*, Vol.24, No.11, pp.770-772, 1981.

[17] R.L. Rivest, “The MD5 message digest algorithm,” *RFC 1321*, Apr. 1992.

[18] FIPS 180-1, “Secure Hash Standard,” NIST, US Department of Commerce, Washington D.C.

[19] R.L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystem," *Communication of ACM*, Vol. 21, No. 2, pp. 120-126, 1978.

[20] D. Coppersmith and M. Jakobsson, "Almost optimal hash sequence traversal," *Proceedings of Financial Cryptography Conference, FC '02*, Lecture Notes in Computer Science, Vol. 2357, Springer Verlag, pp. 102-119, 2003.

[21] S.M. Yen and P.Y. Kuo, "Improved micro-payment system," *Proceedings of 8th National Conference on Information Security*, Taiwan R.O.C., May 1998.

[22] D. Naccache, D. M'Raihi, S. Vaudenay, and D. Raphaeli, "Can D.S.A. be improved? Complexity trade-offs with the digital signature standard," *Advances in Cryptology – EUROCRYPT '94*, Lecture Notes in Computer Science, Vol. 950, Springer Verlag, pp. 77-85, 1995.

[23] S.M. Yen and C.S. Laih, "Improved digital signature suitable for batch verification," *IEEE Transaction on Computers*, Vol. 44, No. 7, pp. 957-959, 1995.

[24] S. Jarecki and A. Odlyzko, "An efficient micropayment system based on probabilistic polling," *Proceedings of Financial Cryptography Conference, FC '97*, Lecture Notes in Computer Science, Vol. 1318, Springer Verlag, pp. 173-191, 1997.

[25] S. Micali and R.L. Rivist, "Micropayments revisited," *Proceedings of RSA Cryptographer's Track, CT-RSA '02*, Lecture Notes in Computer Science, Vol. 2271, Springer Verlag, pp. 149-163, 2002.

[26] IEEE P1363/D13 (Draft Version 13), *Standard Specifications for Public Key Cryptography*, IEEE, pp. 21-22, 1999.

[27] A.J. Menezes, P.C. van Oorschot, and Scott A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, pp. 323-325, 1996.