TamKang University
Department of Electrical Engineering
Laboratory of Cryptography and Information Security

Technical Report
TR-98-2

Improved Micro-payment System

Sung-Ming Yen *Pao-Yu Ku*

April 27, 1998

Improved Micro-payment System *

Sung-Ming Yen[1)] Pao-Yu Ku[2)]

April 27, 1998

[1)] Dept of Electrical Engineering, Tamkang University
Tamsui, Taipei Hsien, Taiwan 25137, R.O.C.
E-mail: yensm@ee.tku.edu.tw

[2)] Technology Research Division, Institute for Information Industry
Taipei, Taiwan, R.O.C.
E-mail: ku@iii.org.tw

Abstract. An important type of electronic payment system is analyzed in this paper. It is widely considered to have a variety of practical applications. Such kind of scheme handles a sequence of very small payments which therefore should be quite efficient to be payed and verified. A micro-payment scheme and its improved version proposed by Rivest is the best candidate for such applications. In general, there are three parties in the payment process, the payer, the receiver, and the bank. In this paper, the Rivest's scheme is thoroughly studied and performance improvement for all the three parties are achieved.

Indexing terms: Cryptography, Electronic commerce, Micro-payment, One-way hash function

1 Introduction

An electronic cash system provides methods for a customer, or called the payer, to make a payment to a merchant, or called the payee, either by a face-to-face approach or over a computer network, especially the Internet, by sending messages called the *electronic money* or *electronic cash*.

A special kind of electronic cash system is called the micro-payment system in which a customer may wish to make a series of small payment to a same merchant over the computer network and it is not appropriate to pay the total amount of money afterwards. More precisely, each really

*This research was sponsored by MOEA and supported by Institute for Information Industry, R.O.C. Also, this work was supported by the National Science Council, R.O.C., under contract NSC87-2213-E-032-012.

TR-98-2

small amount of payment should be received and verified by the merchant before giving the service to the customer. Evidently, in such a scenario of payment, all the following works should be minimized:

(1) Computational cost: This cost should be comparable with the value to be paid. Therefore, the involution of public key computation should be prevented or at least be kept as few as possible.

(2) Storage cost: Since there will be a large amount of payment to be handled, although each of which be of a tiny value, it is not feasible to keep a record of each payment. This will possibly make the cost of processing the payment overloaded.

(3) Administrative cost: This includes the minimization of interactions with the trusted third party, usually the bank, the frequency of doing withdraw and deposit.

Some possible and practical applications of the above micro-payment model have been pointed out, especially those listed in the R.J. Anderson's work [1, 2]. Almost all of these applications are quite straightforward from the very beginning of the motivation of micro-payment. Other more practical applications are believed to be developed in the very near future. Typical applications nowadays are:

(1) Journal publication: A reader often wishes to read only one or a few specific articles of a journal issue, especially of a technical journal, or just wishes to take a look of the abstracts of each articles in that issue. The reader may be discouraged if he should purchase an entire volume of the whole year. It would be more convenient for him to purchase on-line the required article or even the required page. This *information on demand* scenario could possibly encourage sales. By the way, we believe that electronic journal article subscription with the help of automatic on-line keyword searching will be a good publication model in the future.

(2) Newspaper publication: In the similar way of subscribing journal article, when a newspaper is made available online, it would be much convenient for its readers to subscribe an issue, one special topic, one page, a popular column, or a specific article, each with its required small payment. Such flexibility may also encourage sales. Furthermore, we strongly believe that subscribing newspaper articles on-line (we mean to order the electronic version of the article) will be more and more important for the following two reasons: (a) through the help of advanced DSP, digital signal processing, technology, *listening* newspaper article while working will be convenient; (b) order different versions of an article from two or more newspaper publishers to

get a comparison without the need to buy too many will not to read garbage.

(3) Database query: At present, large database services, especially technological article citation databases, are expensive and typically sold by long-term subscription to the libraries of large universities and companies. Both the database providers and its readers will receive benefit, if such kind of database query can be get popular and paid electronically.

(4) Homepage reading: Since the Internet is in fact a web of knowledge, how to encourage people to share more knowledged information with other people requires a methodology to give feedback to the information providers. The most straightforward approach is to give a small amount of payment to the homepage developer. The reader pays when he read the page or each hypertext link to that page.

(5) Advertisement: In a completely reverse scenario, the merchant may wish to pay some small amount of money to his customers when the customers visit the merchant's commercial advertising homepage and answer some questions on the page. Since each payment will be very small, it is not appropriate to pay use the general electronic payment systems. Micro-payment systems, especially the probabilistic micro-payment scheme that will be reviewed later, will be very useful to this commercial advertisement requirement. We strongly believe that interactive advertising with payment to the customers/visitors will be more effective than sending many junk mails (from the view point of the customers) to the customers' mail box or e-mail box. Of course, this will require a well developed advertisement network for the customers to search for the required information.

2 Micro-payment System Based on Cryptographic Hash Chain

In the following, the micro-payment system developed by Rivest and Shamir [3] will be reviewed briefly. Performance analysis and enhancement will also be given. The fundamental cryptographic construct of this payment system is the one-way hash chain which is well known due to its previous application in the development of one-time password by Lamport [4]. Some security considerations about using this simple and general cryptographic construction can be found in [5].

In the Rivest-Shamir micro-payment system, a user/payer commits to a given merchant/payee a *payword chain* consisting of values

$$X_0, X_1, X_2, ..., X_n$$

where $X_i = h(X_{i+1})$ for $i = 0, 1, ..., n-1$, and $h()$ is the underlying *one-way hash function*, e.g., MD5 [6] or SHA [7] (or more precisely, $X_i = h^{n-i}(X_n)$) by signing, e.g., using RSA [8], a message containing the root value X_0. Each successive payment is made by releasing the next consecutive value in the payword sequence, which can be verified easily by checking that it hashes to the previous element. Each element X_i in the sequence is predefined to be of a fixed value for small payment.

For example, the buyer B will initially sign using public key cryptography $Sign_B(X_0)$ with his secret key. The buyer then sends both $Sign_B(X_0)$ and X_0 to the merchant. At the buyer's side, he will store the last coin X_n and the number of coins spent, say the variable j. In the merchant's machine, the message X_0 with its signature and the last received coin from the buyer X_k should be kept. Each time, when the buyer wishes to pay the merchant another new coin, he computes the next new coin $X_{j+1} = h^{n-(j+1)}(X_n)$ and updates j to be $j+1$. The merchant just checks whether $h(X_{j+1}) = X_k$ and updates the last received coin to be X_{j+1} if it is valid. This ensures that a sequence of small payments can be made to a specific merchant and the payer has to make only one computationally expensive public key based digital signature which is for the purpose of commitment.

2.1 Performance Improvement Based on Time-space Trade-off

In the following, some efficiency analysis is sketched from both the view points of space complexity and time complexity. At the buyer's side, since he only stores the last coin X_n, he has to compute each required new coin which consists of a sequence of hash function computations. It is clear from the description of the payment system, 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

$$((n-1) + (n-2) + ... + 1)/n = (n-1)/2$$

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.

Here, we consider the efficiency improvement for the buyer. Evidently, if we use much smaller value for the parameter n, the computational efficiency seems to be enhanced extensively. However, this will eventually slow down the overall system efficiency, because this will require the buyer to generate public key based signature much more frequently. The most important issue of the above micro-payment system is the minimization of using public key cryptography. To solve the computational problem, a feasible approach is the space-time trade-off methodology. An example of improvement is to

cache the middle value $X_{n/2}$ when the buyer computes the root X_0 here we assume n to be an even integer. Under this construction, only one signature has to be computed by the buyer, i.e., $Sign_B(X_0)$. However, the generation of X_1 costs only $n/2-1$ hash computation and X_2 costs $n/2-2$ hash computation, and so on. Computational cost for $X_{n/2+1}$ through X_n are the same as before. On average, each new coin generation will cost

$$2 \cdot ((n/2-1) + (n/2-2) + ... + 1)/n = n/4 - 1/2$$

hash computation. In general, if appropriate m values of the coins are cached during the computation of X_0, computational efficiency of the micro-payment system can be enhanced m times, so each new coin can be obtained using $n/(2m)$ hash computation.

2.2 Optimal Cache Points in the Chain for Practical Applications

For practical applications, the payer may not wish to spend all the money in the chain in a single round to the payee. The payer may only spend half of the chain and the other half will be used later. Under such condition, the optimal cache point (or optimal cache points if more memory is available) to store some value(s) within the chain to speed up coin generation should be selected depending on how many coins will be spent right after the chain being constructed. This subsection will exploit this topic and gives simple while efficient solutions.

As mentioned before, if the payer caches the coin in the middle of the chain when it is constructed, each coin to be spent costs on the average $n/4$ hash computation. This cache position is of course optimal if the constructed chain of coins will be used right away. Recall that in the micro-payment system we wish to reduce the computational cost of each payed coin as little as possible. This is especially important for the case of *on-line* computation. Therefore, in the following we consider how to improve the efficiency for on-line coin generation. Suppose ℓn $(0 \leq \ell \leq 1)$ coins will be spent right after the chain generation and the cnth coin to be spent is cached. It is evident that $0 \leq c \leq \ell \leq 1$ because it is always less efficient to cache the bnth $(b > \ell)$ coin than to cache the cnth $(c \leq \ell)$ coin if only the first ℓn coins will be spent right after the coin generation. Based on the above assumption, the average number of hash computation for each coin within the range of the first ℓn coins $T(\ell, c)$ is approximately

$$T(\ell, c) = \frac{c^2 n + (2-\ell-c)(\ell-c)n}{2\ell}.$$

Optimal selection of cache position of the cnth coin can be obtained by minimizing the above $T(\ell, c)$ under the related parameter ℓ. Some examples of cache position selection are listed in the following Table.

Table 1: Some examples of optimal cache position

the parameter ℓ:	the optimal cache position parameter c:	average cost of each coin within the range
1	1/2	$0.25n$
5/6	1/2	$0.28n$
4/6	1/2	$0.29n$
1/2	1/2	$0.25n$
2/6	2/6	$(2/12)n = 0.167n$
1/6	1/6	$(1/12)n = 0.08n$

Two interesting results in the above Table are:

(1) When $\ell \leq 1/2$ the optimal cache position is the last coin of the first continuous ℓn coins to be spent, i.e., $c = \ell$, and the average number of hash computation for each coin is evidently $(\ell n)/2$.

(2) When $1/2 \leq \ell \leq 1$ no matter how many coins will be spent, the optimal cache position is always the $n/2$th coin. The average number of hash computation under this situation can be computed from $T(\ell, c)$.

Both the above two results can be proved by contradiction. For the first case, it will take more hash computation if a bnth ($b < \ell$) coin is cached. This is because that the coins between bnth and ℓnth coins will be more expensive to be generated than their former ones even using a cache position at $b < \ell$. The same reason for the second case, using a cache coin at positions of either larger or smaller than $n/2$ will make the average cost more expensive.

3 Probabilistic Micro-payment Scheme – Electronic Lottery Tickets

Although the micro-payment system described previously is efficient enough, there is still some problems to be considered. In the conventional cash system, the very large amount of micro-payments transferred between the payers and the payees need not the involution of the trusted third party, usually the bank. However, in the electronic micro-payment system, each hash chain based payword 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, it would be much beneficial for the bank if the signed commitment $Sign_B(X_0)$ can be verified/processed in a batch manner such

that a large group of signatures can be verified together. The signature scheme suitable for batch verification proposed by Yen and Laih [9] is what required for this purpose. The batch verification signature is a modification of the famous Schnorr's signature scheme [10] with the motivation of probabilistic processing. However, the batch processing will induce a long delay before the real deposit of many payword chains owned by a specific receiver can be handled. This could be undesirable for some practical applications. Furthermore, this will add more storage cost for both the bank and the payment receiver. Recall that the storage is one of the costs which should be minimized for micro-payment systems.

In the following, a most recent improved micro-payment scheme proposed by Rivest will be reviewed [11]. The improved version is also a probabilistic scheme in order to minimize the work of the bank. The Rivest's improved micro-payment system is motivated from the idea of lottery ticket issuing and payment. This lottery ticket based scheme is efficient since the bank handles only the winning tickets, instead of handling each micro-payment.

The idea of the improved micro-payment can be sketched in the following scenario of issuing lottery tickets [11]. For example, the issuer or the bank can issue an electronic lottery ticket for a \$10.00 prize with a 1/1000 chance of winning, then each lottery ticket has an expected value of one cent. A customer can pay a merchant one cent by giving him a lottery ticket and one dollar by giving him 100 tickets, and so on. From the bank's point of view, such kind of payments would be much efficient because it has only to pay off winning lottery tickets issued by that payer from that payer's account. This probabilistic aggregation of many small payments into a few winning lottery tickets is the essential reason why electronic lottery tickets as payment system is efficient. Furthermore, the bank does not need to perform any actions corresponding to the withdrawal portion of standard electronic coin schemes.

In such an electronic lottery ticket construction, the ticket is itself a digital signature of the following message (list only the most important parts):

(1) The root of a cryptographic hash chain as before.

(2) The name of the issuer/payer who created the electronic lottery ticket.

(3) The name of the buyer who is using the electronic lottery ticket as a means of payment. In general, the buyer may be the same as the issuer.

(4) The name of the recipient/merchant.

(5) A winning ticket number indicator that indicates how the winning number will be determined for the specific payword chain or for all payword chains.

(6) A ticket face value that specifies the payment to be received if the lottery ticket turns out to be a winner.

Most importantly for the above construction, there are in general two types of winning ticket number indicators, one of the internal type and the other the external approach [11].

(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 three digits (just as an example) 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 only the hash value $h(w)$ to the issuer. The issuer then includes $h(w)$ into the commitment/signature when creating the lottery ticket chain. Note that because a one-way hash is employed, the issuer of the lottery ticket chain would not know w when he issues the ticket.

There are some trade-off between the above two winning indicator methods. If the recipient wishes to know whether each received ticket is a winner, the internal indicator approach is the choice, however it is obvious from the above example that interactions between the payer and the payee are required before the ticket will be constructed and used. On the other hand, a much simple and efficient probabilistic micro-payment system can be obtained when the external indicator is employed. However, now the recipient/payee must suffer the risk that the issuer may unable to pay for the winnings afterwards for some reasons.

Some constructions of probabilistic micro-payment systems [11] are demonstrated and all are directly derived from the hash chain based micro-payment scheme proposed by Rivest and Shamir [3].

(1) External indicator based approach: the protocol (1)

This can be a *non-interactive* version where the payer constructs a payword chain as usual and gives the commitment of this chain of coins (in fact the root of this chain) to the payee. In the commitment, usually a digital signature produced by the payer, an announcement of the winning policy WP, e.g., the source where winning number to get, will be included in the signature as $Sign_B(X_0||WP)$.

Two important considerations/drawbacks about the external indicator approach are [11]:

(a) Collaboration between the payer and the source of issuing winning number, e.g., the bank, should be prevented. The better way is to include more independent sources and some of which are publicly trusted authorities.

(b) The merchant/payee must store all electronic lottery tickets he received, until the related winning numbers are revealed and the tickets are checked.

(2) Internal indicator based approach-1: the protocol (2)

This will be an *interactive* version where the recipient/payee first randomly selects the winning number w (in fact only a portion of it will be used as the winning number, e.g., the last three decimal digits) but he will just send the hash value $h(w)$ to the issuer/payer. The payer constructs the chain of coins in the same way but now he gives the commitment

$$Sign_B(X_0||h(w))$$

to the recipient. Because now the recipient has the winning number, he can get immediate information about whether the received ticket is a winning one or not. Therefore, a real payment from the payer can be requested by showing the winning number w, so two previously mentioned drawbacks can be prevented; they are (1) the risk that the issuer may unable to pay for the winnings afterwards for some reasons; (2) The merchant/payee must store all electronic lottery tickets he received, until the related winning numbers are revealed and the tickets are checked.

However, showing the winning number to the payer will force him to stop using the remaining coins in this chain! Evidently, this will somewhat destroy the game rule of the lottery ticket as payment. Quantitative analysis of this result by experiment will proceed. If the immediate request for payment in case of winning is a must, for example the merchant must do its accounting work at the end of each day. The next internal indicator based protocol will be another choice, however the next protocol will be much less efficient for the merchant as will be described later. Fortunately, another improved version with internal indicator allowing immediate payment request will be proposed in the following.

(3) Internal indicator based approach-2: the protocol (3)

In this approach, the payer also constructs his payword chain. The more interesting part is that now the merchant also constructs his chain, we called it the *winning number chain*, as

$$W_0, W_1, W_2, ..., W_n$$

where $W_i = h(W_{i+1})$ for $i = 0, 1, ..., n-1$. The merchant then gives the root of this winning number chain, i.e., W_0, to the payer. The payer/issuer prepares the commitment as usual as

$$Sign_B(X_0||W0)$$

and sends it with the root X_0 to the merchant.

Then the i-th ticket in the buyer's chain, with value X_i, is a winning ticket if and only if $X_i = W_i \pmod{1000}$ (just as an example). In this way, because the merchant knows the number W_i he can immediately inform the buyer when X_i wins by revealing W_i while without giving the buyer the information about W_{i+1} and something further. This solves the problem described before that showing the winning number to the payer will force him to stop using the remaining coins in this chain.

3.1 Improved Probabilistic Micro-payment Scheme – Internal indicator based approach-3

It should be noted that what the merchant gave to the payer is the number W_0 which is equal to $h^n(W_n)$. This implies that the merchant now takes almost the same computational complexity to check each received ticket as the payer does to generate each ticket to be paid. In fact, the merchant takes one more hash computation for each ticket checking. On average, for the merchant's side, each received ticket will cost

$$(((n-1)+(n-2)+...+1)/n)+1=(n+1)/2$$

hash computation to be verified and checked for its validity and winning status, respectively. Recall that in all the previous protocols, the merchant always takes one hash function computation to verify the validity and to check the winning status of each received new coin. This new protocol allowing immediate payment requesting although solves one problem, however brings another more serious drawback. Generally speaking, the merchant doing on-line service may serve quite many users at the same time and the merchant still has other main tasks, i.e., the service itself, to be performed. Therefore, a computationally inefficient approach like the one reviewed in this item is not a practical solution. This is not the case for the payer who will usually communicate with only one or very few servers/merchants.

We think that the immediate payment request is a practical operation model for some commercial applications. Therefore, a much efficient approach than the previous one should be developed. This can be achieved by combining both the ideas given in the protocols (2) and (3) and it will be sketched in the following.

In this improved system, the payer also constructs the chain of coins as

$$X_0, X_1, X_2, ..., X_n$$

where $X_i = h(X_{i+1})$ for $i = 0, 1, ..., n-1$. Similarly, the merchant constructs his winning number chain as

$$W_0, W_1, W_2, ..., W_m$$

where $W_i = h(W_{i+1})$ for $i = 0, 1, ..., m-1$. However, this time the winning number chain can be much shorter than the payword chain, i.e., $m \ll n$. The merchant also gives the root of this winning number chain, i.e., W_0, to the payer. The payer/issuer prepares the commitment as usual as

$$Sign_B(X_0||W_0)$$

and sends it back with the root X_0 to the merchant.

There are some differences compared with the previous protocol. Now, the i-th ticket X_i in the buyer's chain is the first winning ticket (of this chain only) if and only if $X_i = W_1 \pmod{1000}$ (also just as an example). When this happened, the merchant can show the first winning number W_1 he selected (in fact, computed) to the payer to ask for immediate payment. The payment continues and when another ticket X_j satisfies the condition such that $X_j = W_2 \pmod{1000}$, then the second winning ticket has been received by the merchant. In this way, revealing the first winning number W_1 will not give the buyer the information about the second winning number W_2 and all the following winning numbers. It is important that since this is a probabilistic payment protocol and usually with very small possibility of double or more winnings in a single payword chain, e.g., considering the arrangement of

(a) winning number of 4 decimal digits;

(b) payword chain of length between 10 to 50.

Generally, the winning number chain of length (i.e., the parameter m) between 2 to 5 is enough. This resolves the problem described before that slows down the performance of the merchant.

In this new design, the merchant can compute the next required wining number by a sequence of hash evaluations from W_m or just to keep the list of selected/computed winning numbers in a table for later lookup purpose.

As far as we known, this proposed improvement with the properties of probabilistic processing and immediate payment request is the most efficient protocol in this topic.

4 Conclusions

Besides the purposes to hash a large message before signing and to construct the one-time password mechanism, Rivest demonstrated another novel application of one-way hash function to be as an efficient electronic micro-payment system. In the Rivest's micro-payment system, a sequence of coins to be spent are prepared in the form of one-way hash chain such that the coin-chain is generated in the computationally easy direction while it is

payed to the payee in the computationally hard direction. Rivest even proposed an improved version in the probabilistic approach trying to reduce the amount of works of the bank. As pointed out in the introduction, micro-payment would play a major role in the future electronic commerce. Development of secure and efficient micro-payment systems for general purpose applications is an important issue undoubtedly. For this reason, Rivest's payment system and its modification are thoroughly studied in this paper in order to make the already efficient schemes even more perfect. The final version of the scheme is efficient for all the three parties of the payment, i.e., the payer, the payee, and the bank.

References

[1] R. Anderson, H. Manifavas and C. Sutherland, 'NetCard – A practical Electronic cash system', Manuscript 1995, Available from the author or from his WWW homepage.

[2] R. Anderson, 'Electronic cash system', Manuscript, Available from the author or from his WWW homepage.

[3] R.L. Rivest and A. Shamir, 'PayWord and MicroMint: two simple micropayment schemes', presented at the *RSA '96 Conference*, 1996.

[4] L. Lamport, 'Password authentication with insecure communication', *Commun. of ACM*, **24** (11), 1981, pp.770–772.

[5] S.M. Yen, 'Security consideration of using cryptographic hash chain', Technical Report TR-97-10B, Dept. of Electrical Engineering, TamKang University, Republic of China, 1997.

[6] R. Rivest, 'The MD5 message digest algorithm', *RFC 1321*, (Apr. 1992).

[7] FIPS 180-1, 'Secure Hash Standard', NIST, US Department of Commerce, Washington D.C., April 1995.

[8] R.L. Rivest, A. Shamir, and L. Adleman, 'A method for obtaining digital signatures and public-key cryptosystem', *Commun. ACM*, Vol.21, pp.120–126, Feb. 1978.

[9] S.M. Yen and C.S. Laih, 'Improved digital signature suitable for batch verification', *IEEE Trans. on Computers*, Vol.44, No.7, pp.957–959, July 1995.

[10] C.P. Schnorr, 'Efficient identification and signatures for smart cards', *Crypto'89*, Vol.435 (Lecture Notes in Computer Science), New York: Springer-Verlag, pp.239–252, 1990.

[11] R.L. Rivest, 'Electronic lottery tickets as micropayments', Manuscript, Available from the author or from his WWW homepage, 1997.