Weighted One-Way Hash Chain and Its Applications

Sung-Ming Yen[1] and Yuliang Zheng[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.
Email: `yensm@csie.ncu.edu.tw`

[2] LINKS – Laboratory for Information and Network Security
School of Network Computing
Monash University
McMahons Road, Frankston, Victoria 3199, Australia
Email: `yuliang.zheng@infotech.monash.edu.au`

Abstract. An one-way hash chain generated by the iterative use of a one-way hash function on a secret value has recently been widely employed to develop many practical cryptographic solutions, especially electronic micropayment schemes. In this paper, we propose a new concept called a weighted one-way hash chain. We then proceed to use the new concept to improve in a significant way the performance of micropayment schemes. We also show that the proposed technique is especially useful in implementing micropayment on a resource restrained computing device such as a hand-held computer.

Keywords: Cryptography, Electronic commerce, Micropayment, One-way hash chain, Portable computing device.

1 Introduction

Internet micropayment schemes have received growing attention recently, largely 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 [1], micropayment schemes allow a customer to transfer to a merchant a sequence of small amount payments over the computer network in exchange for services or electronic products from the merchant. With these services or products, often it is not quite appropriate 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. Possible practical applications of the above micropayment model include digital newspaper [2], on-line journal subscription, on-line database query, multimedia entertainment over the Internet, and Internet advertisement (say via lottery tickets [3]). More examples can be found in [3, 4]. In addition, accounting and pricing for Internet services and mobile telecommunication may represent yet another set of promising applications of micropayments [5–9].

The most notable representatives of micropayment schemes include those proposed in [3, 4, 10–14]. The fundamental cryptographic tool for most of these payment systems is a one-way hash chain which has been known widely by researchers ever since Lamport first proposed its use in one-time passwords [15, 16]. One-way hash chains have also been extensively employed in the development of a special class of high-speed signature schemes called the one-time signature schemes [17–21]. As this class of signature schemes use only one-way functions, they can be very fast by the use of efficient cryptographic hash functions, rather than less efficient trap-door one-way functions [22].

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*

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

When the function $h()$ in the iteration is instantiated with a one-way hash function, such as MD5 [23], SHA [24], and HAVAL [25], 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 Review of the PayWord Micropayment Scheme

The PayWord micropayment scheme [4] which is mainly based on the idea of using a one-way hash chain, will be particularly illustrative in explaining our ideas on weighted one-way hash chains. In this section we briefly review the scheme.

Prior to the first transaction taking place 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 [26], 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" which is 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}.

3 Related Work on Improving Micropayment Schemes

In our view, a good micropayment scheme should be designed in such a way to meet the requirements of minimizing the computational, storage, and administrative costs for interactions with the bank. Indeed, many micropayment schemes that have been developed by various researchers share a simple structure and acceptable performance with the one-way hash chain described above.

It should also be pointed out that ease of implementation is another fundamental requirement of a payment system, especially in such applications as portable micropayment systems and mobile telecommunication charging schemes as mentioned earlier. These environments generally involve the use of portable computing devices which often have limited computing resources, e.g., a small amount of memory space, a relatively slow CPU, and a short life span of batteries.

In [7], an experimental portable micropayment system based on PayWord [4] has been reported. From the discussions given in the paper, it becomes clear that for a general purpose portable device, while a small or moderately large value of n (call the *length* of the payment chain) would be acceptable, a larger n can cause unacceptably lengthy delay in computation. On the other hand, a larger value of n reduces the required amount of computation for public key based signature which is actually the essence of developing PayWord-like micropayment schemes.

To solve the above problem with contradicting requirements, a straightforward alternative of using a larger value of n was suggested in [4, 7]. The alternative method allows one to construct different payment chains for different denominations. This method, however, introduces a new problem in that it complicates the task of implementation and operations (also stated in [7]). The alternative method also requires much more memory space to store the x_n's (the random secret) for all the payment chains, and to remember the index of the last spent coin of each payment chain. Thus, the alternative method does not really provide satisfactory solutions from the viewpoint of portable devices. Furthermore, we also note that the method complicates the operation of the merchant and requires the merchant to store the last received coin of each chain from the customer. For all these difficulties in practice, proposals of this type will not be considered in the following discussions.

We argue that the development of efficient structures different from a simple one-way hash chain is necessary for good micropayment schemes, especially for those to be implemented on a portable computing device. Currently, only very limited research results on this topic can be found in the literature.

One of such attempts is a structure called PayTree proposed in [27]. While a PayTree does offer a solution for a multi-merchant environment where a PayTree can be spent among many different merchants, it has two drawbacks that discourage the use of PayTree in practice. The first drawback is that the customer needs to store all the leaf nodes (as independent random numbers) of a PayTree. These leaf nodes represent electronic coins bought by the customer from a bank. In a practical application, as the number of leaf nodes could be large, the customer may need to prepare a large amount of memory space to store all the node values. The second drawback is that double spending of a coin in the PayTree scheme cannot be avoided, although it can be detected afterwards. The reason is simple: the same coin can be paid to many different merchants by the customer. This drawback complicates greatly the system's operation.

In a more recent development, the authors of [28] proposed a new tree-based structure called an *unbalanced one-way binary tree* (UOBT). A major difference between UOBT and PayTree is that UOBT facilitates merchant specific micropayments in a simple way similar to the conventional one-way hash chain structure. In a scheme based on UOBT, a secret random value is chosen as the root (very much like the x_n in the one-way hash chain structure). This secret value is used to construct a tree from the root towards the lower levels in an unbalanced binary tree, such that given a child node, no parent node can be derived from the child node. In [28], it was demonstrated that the UOBT approach can improve the performance of micropayment schemes significantly. An independent analysis recently carried out by Peirce [29] further confirm that the UOBT approach is superior to the conventional one-way chain not only from a theoretical point of view, but also an implementor's point of view.

The main contribution of this paper is to propose a new technique for improving the performance of micropayments based on one-way hash chains. The core of the new technique is to assign, in a randomized manner, a different denomination to each coin. An interesting feature of the new technique is that it complements UOBT [28], and can be combined with the UOBT approach using two different methods. The first method is straightforward, and it assists the further enhancement of the performance of UOBT. This is based on the observation that although the UOBT constructs a tree instead of a chain, a value in the tree still stands for a coin in the same way as a value in an ordinary one-way hash chain. The second method is of special interest in that it enables the spending of a single UOBT structure among several different merchants, whereby further enhancing the overall performance of a micropayment system.

4 The Proposed Solution

In a real world application involving a customer and a merchant, it would be quite possible that the customer is asked to pay several coins a time in order to obtain services from the merchant. Think of the case of browsing an interesting web site. Instead of paying fees page by page, it would make more sense for the customer to pay the overall costs prior to or after viewing all the relevant pages for a particular topic.

Let us consider the case where the number of coins to be paid, say c, is assumed to be a random integer selected from $[1, t]$. Examining the micropayment schemes (and their implementation) published previously in the literature, one can see that all these proposals have implicitly assumed that c is a random integer. Realizing this, some researchers suggest to use different chains for different denominations [4, 7]. While this would certainly improve the performance, the major aim of this paper is to examine ways on how to further improve, in a significant way, the performance of a PayWord-like micropayment scheme in which the customer is asked to pay more than one coin a time to the merchant. The starting point of our solutions is the use of so-called weighted one-way hash chains. We show how these weighted chains can be used to provide a far better alternative to improve the overall performance and implementation of a micropayment system.

4.1 The Weighted One-way Hash Chain

With the conventional one-way hash chain approach, typically each coin in a payment chain is assumed to be worth d cents. When x_j is the last spent coin and c more coins need to be paid to the merchant for a new payment transaction, the customer computes $x_{j+c} = h^{n-(j+c)}(x_n)$ and passes it over to the merchant as a new coin. The last spent coin is then updated to x_{j+c}, and j to $j+c$. Clearly, the value of each payment will be worth a multiple of d cents. The actual value of d can be determined at the outset between the customer and the merchant.

In order to improve the performance of micropayment systems based on one-way hash chains, e.g., the PayWord scheme [4], the following new concept is introduced and its applications will be examined.

Definition 1 *A weighted one-way hash chain* $\{x_0, x_1, x_2, \cdots, x_{n-1}, x_n\}$*, where* $x_i = h^{n-i}(x_n)$*, consists of a generic one-way hash chain together with a specific weighting value* w_i *being assigned to each* x_i*.*

A possible weighting assignment mechanism is the self-encoding method, i.e., to let $w_i = f(x_i)$ where the function f can be any well defined mapping from a value x_i to a weight w_i. For a concrete example, when w_i needs to be a random variable over $[1, t]$, the function can be defined as

$$w_i = (x_i \bmod t) + 1. \tag{1}$$

When $x_i \gg t$, the function can be modified to

$$w_i = ((x_i \bmod 2^b) \bmod t) + 1 \tag{2}$$

where $b \geq \lceil \log_2 t \rceil$. In Eq. (2), w_i can be computed with little effort by a binary arithmetic processor, especially when $x_i \gg t$ and t is not a power of two. Note that x_i generated by a typical cryptographic one-way hash function will have a value close to 10^{40}. However, if t is not a power of two, w_i obtained from Eq. (1) will be distributed more uniformly over $[1, t]$. As a real example, for $t = 6$, setting $b = 5$ or $b = 6$ in Eq. (2) will result in an acceptable w_i that is more or less uniformly distributed over $[1, 6]$.

4.2 Micropayment with Varying Denomination

Based on a weighted one-way hash chain as discussed above, the original PayWord micropayment scheme can be modified as follows. The payment chain generation process is still the same but the signature part needs some minor modifications. More specifically, we define

$$Sign_C(\text{Merchant-ID}||x_0||\text{Other-Info}||\text{Cert})$$

where 'Other-Info' describes the parameters t and b and some other necessary information, e.g., d if it is different for each chain. During the payment process, the *denomination* of each coin is determined by its weighting value described in the previous sub-section.

Suppose that in the ith payment transaction, the customer is required to pay the merchant $c_i \cdot d$ cents (where c_i is an integer between $[1, t]$), i.e., paying c_i coins in the one-way hash chain. Fig. 2 describes the payment procedure performed by the customer when the weighted one-way hash chain is employed. For each new payment which costs $c_i \cdot d$ cents, the customer computes a pair of integers $\{x_j, e\}$. If $e = 1$, x_j is the coin value exactly next to the one received by the merchant in the previous transaction. If $e > 1$, more than one coin will be spent and x_j is the dominating one, i.e., all other coins are derived from x_j.

An additional variable B (it stands for balance) can be defined to further facilitate the use of a weighted payment chain. The variable B is originally set to zero and will be modified during a payment transaction. The operations between Step (2.7) and Step (2.17) are optional. Note that if both c_i and w_j are random integers over $[1, t]$, then the variable B will statistically remain in a limited range $[-\delta, +\delta]$ for most of the time. The parameter δ depends mainly on the value of t. From this point of view, operations between Step (2.7) and Step (2.17) are not really required. However, the customer may intentionally break the rule by spending c_i larger than the computed w_j for every payment. This will make B much larger than an expected parameter δ after the whole payment chain is spent, resulting in a situation that is not fair to the merchant. To avoid this possible cheating by the customer, operations between Step (2.7) and Step (2.17) compute one or more additional coins in order to keep B within the specified range. In Step (2.7), $\delta = 2t$ is selected, primarily as an example.

```
Initially:  $B \leftarrow 0$;  $j \leftarrow 0$
---------------------------------------------------------
For each new payment worth $c_i \cdot d$ cents:
2.1   if $j < n$ then
2.2     $j \leftarrow j + 1$
2.3     $x_j \leftarrow h^{n-j}(x_n)$
2.4     $w_j \leftarrow ((x_j \bmod 2^b) \bmod t) + 1$
2.5     $B \leftarrow B + (w_j - c_i)$
2.6     $e \leftarrow 1$
2.7     while $B < -2t$ do
2.8       if $j < n$ then
2.9         $j \leftarrow j + 1$
2.10        $x_j \leftarrow h^{n-j}(x_n)$
2.11        $w_j \leftarrow ((x_j \bmod 2^b) \bmod t) + 1$
2.12        $B \leftarrow B + w_j$
2.13        $e \leftarrow e + 1$
2.14      else
2.15        (stop because there are not enough coins)
2.16      endif
2.17    endwhile
2.18  else
2.19    (stop because there are not enough coins)
2.20  endif
2.21  send $x_j$ and $e$ to the merchant
```

Fig. 2. Customer's payment procedure based on a weighted one-way hash chain.

Fig. 3 describes the corresponding payment verification process performed by the merchant. The variable X_ℓ is initially set to x_0. It is used to store the last received coin. Naturally, the coin will have to pass the verification process. For each newly received payment $\{x_i, e\}$, the merchant checks whether or not $X_\ell = h^e(x_i)$. The merchant also adjusts the variable B according to one or more computed weighting values w and checks whether or not $B \geq -2t$ (let δ be $2t$).

If the two verifications are both correct, the merchant stores x_i into X_ℓ as the last received coin. After the successful completion of the present transaction, say the pth transaction, the merchant should have collected $(\sum_{j=1}^{p} w_j) \cdot d$ cents from the customer. As in the payment procedure by the customer, the checking of whether $B_{temp} \geq -2t$ in Step (3.8) is optional.

```
Initially:  $B \leftarrow 0$;  $X_\ell \leftarrow x_0$
------------------------------------------------------

For each received $\{x_i, e\}$ worth $c_i \cdot d$ cents:
3.1    $X \leftarrow x_i$;   $B_{temp} \leftarrow B$
3.2    for $j$ from 1 to $e$
3.3      $w \leftarrow ((X \bmod 2^b) \bmod t) + 1$
3.4      $B_{temp} \leftarrow B_{temp} + w$
3.5      $X \leftarrow h(X)$
3.6    endfor
3.7    $B_{temp} \leftarrow B_{temp} - c_i$
3.8    if $X = X_\ell$ and $B_{temp} \geq -2t$ then
3.9      (accept the payment)
3.10     $X_\ell \leftarrow x_i$
3.11     $B \leftarrow B_{temp}$
3.12   else
3.13     (reject the payment)
3.14   endif
```

Fig. 3. Merchant's payment verification based on weighted one-way hash chain.

4.3 Some Useful Special Weighting Assignment Algorithms

For some practical micropayment applications, the value of c_i may not be uniformly distributed over the range of $[1, t]$, e.g., small values or large values of c_i will occur more frequently. In the following, the weighting value w_i (over $[1, t]$) of each coin x_i is computed by using simple algorithms given in Fig. 4, where $(x_{i,t-2}, x_{i,t-3}, \ldots, x_{i,1}, x_{i,0})_2$ are the $t-1$ least significant bits in the binary representation of x_i. For the case of small value bound payments in which small values of c_i will occur more frequently, the special weighting assignment mechanism in Fig. 4(a) can be employed. As an example, in Fig. 4(a), if $t = 5$ then the probabilities for w_i to be 1, 2, 3, 4, and 5 are $\frac{8}{16}$, $\frac{4}{16}$, $\frac{2}{16}$, $\frac{1}{16}$, and $\frac{1}{16}$, respectively.

On the other hand, for the case of large value bound payments, the weighting assignment mechanism in Fig. 4(b) can be useful. Consider the case of $t = 5$. Then the probabilities for w_i to be 1, 2, 3, 4, and 5 are $\frac{1}{16}$, $\frac{1}{16}$, $\frac{2}{16}$, $\frac{4}{16}$, and $\frac{8}{16}$, respectively.

One more problem to be raised is whether or not a special weighting assignment mechanism is required for medium value bound payments (or more

```
Input:  $(x_{i,t-2}, x_{i,t-3}, \ldots, x_{i,1}, x_{i,0})_2$
Output: $w_i$
---------------------------------------------
4a.1  $w_i \leftarrow 1$; $j \leftarrow t-2$
4a.2  while $x_{i,j} = 1$ do
4a.3      $w_i \leftarrow w_i + 1$
4a.4      $j \leftarrow j - 1$
4a.5  endwhile
```

(a) Algorithm for small value bound payments.

```
Input:  $(x_{i,t-2}, x_{i,t-3}, \ldots, x_{i,1}, x_{i,0})_2$
Output: $w_i$
---------------------------------------------
4b.1  $w_i \leftarrow t$; $j \leftarrow t-2$
4b.2  while $x_{i,j} = 1$ do
4b.3      $w_i \leftarrow w_i - 1$
4b.4      $j \leftarrow j - 1$
4b.5  endwhile
```

(b) Algorithm for large value bound payments.

```
Input:  $(x_{i,t-2}, x_{i,t-3}, \ldots, x_{i,1}, x_{i,0})_2$
Output: $w_i$
---------------------------------------------
4c.1  $w_i \leftarrow \lceil \frac{t}{2} \rceil$; $j \leftarrow t-2$; $k \leftarrow 1$; $p \leftarrow 0$
4c.2  while $x_{i,j} = 1$ do
4c.3      $w_i \leftarrow w_i + (-1)^p \cdot k$
4c.4      $j \leftarrow j - 1$; $k \leftarrow k + 1$; $p \leftarrow p + 1$
4c.5  endwhile
```

(c) Algorithm for near normally distributed value bound payments.

Fig. 4. Proposed special weighting assignment algorithms.

precisely as the *normally* distributed with mean at $\lceil \frac{t}{2} \rceil$). Interestingly, the assignment mechanisms provided in Eq. (1) or Eq. (2) more or less meet the above requirement, since the weighting w_i computed via these two approaches will be a random integer, almost *uniformly* selected from the range of $[1, t]$. Therefore, statistically the variable B will still remain in a limited range $[-\delta', +\delta']$ during the payment process. However, in this case, δ' may be larger than the value δ, especially when either Eq. (1) or Eq. (2) is employed for uniformly random payments as shown in Fig. 2. As a better alternative, for near normally distributed value bound payments in which medium values of c_i are expected to occur more frequently, the special weighting assignment mechanism described in Fig. 4(c) can be employed. Consider the case of $t = 5$. Then the probabilities for w_i to be 1, 2, 3, 4, and 5 are $\frac{1}{16}$, $\frac{2}{16}$, $\frac{8}{16}$, $\frac{4}{16}$, and $\frac{1}{16}$, respectively.

Of course, the actual distribution of c_i over the range $[1, t]$ or equivalently the expected value of c_i will be determined by a number of factors, such as the customer's payment behavior, prices of products, or a combination of the above two. These, however, are out of the main concerns of this paper.

Nevertheless, let us point out that in typical micropayment applications, the parameter t should not be a very large integer, e.g., $10 \leq t \leq 20$ may represent a reasonable range of values. Therefore, generally speaking, there is perhaps little need to consider more complex weighting assignment mechanisms than the simple ones discussed above. In a rare situation where a large payment (i.e., $c_i > t$) needs to be executed, the customer can simply send more than one coin to the merchant, just like in the conventional one-way hash payment chain approach.

5 Analysis of Performance

In this section, a comparison of the performance of various possible micropayments based on the weighted and the conventional one-way hash chains will be carried out. To simplify the analysis, it will be assumed that the value of c_i is uniformly distributed over the range of $[1, t]$ and $e = 1$ for every transaction.

Lemma 1. *With a conventional one-way hash chain (with length n), if each payment transaction is worth c_i coins (i.e., $c_i \cdot d$ cents) where $c_i \in_R [1, t]$, then the expected computational cost for each d cent transaction is approximately $n/(t + 1)$ evaluations of a one-way hash function.*

Proof. The expected total number of hashing operations required to evaluate all the coins c_i within a payment chain is

$$T = \sum_{\substack{i = 1 \\ c_i \in_R [1, t]}}^{m} \left(n - \sum_{j=1}^{i} c_j \right) \approx \frac{n^2}{t + 1}$$

where c_m is the last spent coin of the chain. Therefore, the expected computational cost for every d cent transaction is roughly about $T/n = n/(t+1)$ hashing operations. □

Lemma 2. *With a weighted one-way hash chain (with length n), if each payment transaction is worth $c_i \cdot d$ cents, where $c_i \in_R [1, t]$, then the expected computational cost for each d cent transaction is approximately $n/(t + 1)$ evaluations of a one-way hash function.*

Proof. The expected number of all the cents embedded in the chain is

$$D = \sum_{i=1}^{n} (c_i \cdot d) = \frac{(t + 1) \cdot d \cdot n}{2}.$$

Whereas the total number of hashing operations required to evaluate all the coins x_i $(i = 1, 2, \cdots, n)$ is

$$T = \sum_{i=1}^{n} (n-i) = \frac{n \cdot (n-1)}{2}.$$

Therefore, the expected computational cost for each d cent transaction is $\frac{T}{D/d} = (n-1)/(t+1) \approx n/(t+1)$ evaluations of hashing. Note that we have taken into account the fact that $n \gg t$ in most applications. $\square$

Note that for transactions worth $c_i \cdot d$ cents ($c_i \in_R [1, t]$) in a payment chain of length n, Lemma 1 and Lemma 2 indicate that on average, the conventional one-way hash chain and the proposed weighted one-way hash chain perform equally well for paying d cents (i.e., the defined smallest amount of payment for the specific micropayment scheme).

However, this does not imply that both techniques perform equally efficiently for all micropayment applications. While a conventional one-way hash chain can be used to embed $n \cdot d$ cents, a weighted chain with length n can on average represent $(\frac{t+1}{2} \cdot n) \cdot d$ cents. We note that if the parameter t is equal to one, i.e., being an equally weighted chain with all $w_i = 1$, the weighted chain is reduced to a conventional chain.

Recall that with the conventional one-way hash chain approach, after spending all the n coins, it requires the customer to generate a new payment chain and to create a new public key based signature for the chain. When this happens too often, the overhead of the micropayment scheme will be increased significantly. At the first glance, the selection of a large value n may require the customer to sign chains less frequently. In fact, this does not completely solve the problem, simply because a large n will result in the requirement of spending more time in evaluating each single coin. As was mentioned earlier, the first good solution to the problem is to use an unbalanced one-way binary tree [28]. A major benefit of such an unbalanced tree is that it requires less frequent generations of public key based signatures and a smaller amount of computation time for each coin. In what follows, we analyze in detail how and why the use of weighting over each coin represents yet another important contribution to solving the problem.

Lemma 3. *Suppose that $(\frac{t+1}{2} \cdot n) \cdot d$ cents will be embedded into a single chain and each payment transaction will be worth $c_i \cdot d$ cents where $c_i \in_R [1, t]$. Then, the conventional chain approach will take $\frac{t+1}{2}$ times of computation when compared with the approach of employing a weighted one-way hash chain.*

Proof. In order to embed $(\frac{t+1}{2} \cdot n) \cdot d$ cents into a single conventional chain, it requires the chain length to be enlarged to $n' = \frac{t+1}{2} \cdot n$. However, from the result of Lemma 1, this will result in

$$\frac{n'}{t+1} = (\frac{t+1}{2}) \cdot \frac{n}{t+1} = \frac{n}{2} \tag{3}$$

expected number of hashing evaluations for each d cent transaction. This represents $\frac{t+1}{2}$ times of the computational cost required by a weighted one-way hash chain. $\square$

Clearly, by reducing the computational complexity from $\frac{n}{2}$ (see Eq. (3)) down to $\frac{n}{t+1}$, $t \geq 1$, the proposed weighted one-way hash chain technique will significantly improve the performance of micropayment systems, by an order of $O(t)$.

We have the following more general result that indicates in greater detail performance improvement achievable by the use of a weighted one-way hash chain.

Lemma 4. *Assume that there will be an equal amount of money to be embedded into two one-way hash chains, one conventional and the other weighted. Further assume that each payment transaction will be worth $c_i \cdot d$ cents, where c_i is a random (but not necessarily uniformly distributed) integer selected from $[1, t]$ and with an expected value of E. Then, the conventional one-way hash chain approach will take E times of computation when compared with the weighted chain approach.*

To summarize the above discussions, by using a weighted one-way hash chain, the computational cost of the customer can be reduced to about $\frac{1}{E}$ times of that required by a conventional one-way hash chain. It is equally important to note that the same level of improvement is achieved on the merchant's side.

As discusses earlier, weighted one-way hash chains and unbalanced one-way binary trees (UOBTs) introduced in [28] represent two complementary methods for improving the performance of micropayment systems. These two techniques can be combined to provide even a greater number of efficient solutions. To close this section, we note that

(1) The UOBT technique improves the customer's performance in the order of $O(\sqrt{n})$. However, it does not improve the merchant's performance and in fact it requires the merchant to store $O(\sqrt{n})$ more temporary values (for details see [28]).
(2) The proposed weighted one-way hash chain technique improves both the customer's and the merchant's performances in the order of $O(t)$.
(3) Both the weighted one-way hash chain and the UOBT are suitable for implementing on portable computing devices.

6 Conclusions and Future Works

We have proposed the novel concept of a weighted one-way hash chain, and proven that the chain is very useful in the design and implementation of electronic micropayments, especially for those systems to be used with portable devices. An important characteristic of the proposed weighted one-way hash chain is that it improves not only the performance of the customer but also that of the merchant.

It is interesting to consider the role of weight assignment on each computed hash value. If weighting is defined to encode the coin denomination, then a micropayment scheme with varying denomination is obtained. On the other hand, if weighting is defined to encode the identity of merchant, then a micropayment scheme for multiple merchants can be readily constructed. This is particularly true when each hash value within the UOBT is mapped to a predefined set of merchant identities.

More interestingly, if each hash value of the UOBT is given two weighting values via two separate weighting assignment functions, one being used to encode a merchant's identity and the other to encode the coin denomination, then a micropayment scheme for multiple merchants with varying denominations can be obtained.

Examining other possible weight assignment methods for various applications other than micropayments is an interesting open research topic.

7 Acknowledgments

The research of S.M. Yen was supported in part by the National Science Council of the Republic of China under contract NSC 89-2213-E-008-055.

The authors wish to thank Chung-Kai Hwang from the Department of Computer Science, National Central University for stimulating discussions on the special weighting assignment algorithms.

References

1. D. O'Mahony, M. Peirce, and H. Tewari, *Electronic Payment Systems*, Artech House, INC., 1997.
2. J.W. Palmer and L.B.Eriksen, "Digital newspapers explore marketing on the Internet," *Commun. of ACM*, Vol.42, No.9, pp.33–40, 1999.
3. R.L. Rivest, "Electronic lottery tickets as micropayments," *Proc. of Financial Cryptography Conference, FC '97*, Lecture Notes in Computer Science, Vol.1318, Springer Verlag, pp.307–314, 1998.
4. R.L. Rivest and A. Shamir, "PayWord and MicroMint: Two simple micropayment schemes," *Proc. 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.
5. G. Horn and B. Preneel, "Authentication and payment in future mobile systems," *Proc. 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," *Proc. 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. N. Daswani and D. Boneh, "Experimenting with electronic commerce on the PalmPilot," *Proc. of 3rd Financial Cryptography Conference, FC '99*, Lecture Notes in Computer Science, Vol.1648, Springer Verlag, February 1999.

8. D. O'Mahony, L. Doyle, H. Tewari, and M. Peirce, "NOMAD – An application to provide UMTS telephony services on fixed terminals in COBUCO," *Proc. of 3rd ACTS Mobile Communications Summit*, Vol.1, pp.72–76, Rhodes, Greece, June 1998.
9. M. Peirce and D. O'Mahony, "Micropayments for mobile networks," Technical Report of the Dept. of Computer Science, Trinity College Dublin, Ireland, 1999.
10. R. Anderson, C. Manifavas and C. Sutherland, "NetCard – A practical electronic cash system," *Proc. of Security Protocols Workshop*, Lecture Notes in Computer Science, Vol.1189, Springer Verlag, pp.49–57, 1997.
11. R. Hauser, M.Steiner, and M.Waidner, "Micro-payments based on iKP," *Proc. of SECURICOM '96, 14th Worldwide Congress on Computer and Communications Security and Protection*, pp.67–82, 1996.
12. T. Pedersen, "Electronic payments of small amounts," *Proc. of Security Protocols Workshop*, Lecture Notes in Computer Science, Vol.1189, Springer Verlag, pp.59–68, 1997.
13. S. Glassmann, M. Manasse, M. Abadi, P. Gauthier, and P. Sobalvarro, "The Millicent protocol for inexpensive electronic commerce," *Proc. of 4th International World Wide Web Conference*, Boston, MA, pp.603–618, Dec. 1995.
14. S.M. Yen, J.M. Lee, and J.G. Lee, "PayFair: A prepaid Internet micropayment scheme promising customer fairness," *Proc. of International Workshop on Cryptographic Techniques and E-Commerce, CrypTEC '99*, Hong Kong, pp.213–221, 5-8 July 1999.
15. L. Lamport, "Password authentication with insecure communication," *Commun. of ACM*, Vol.24, No.11, pp.770–772, 1981.
16. N.M. Haller, "The S/KEY one-time password system," *Proc. of the ISOC Symposium on Network and Distributed System Security*, San Diego, CA, Feb. 1994.
17. M.O. Rabin, "Digital signatures," *Foundations of Secure Computation*, Academic Press, pp.155–168, 1978.
18. L. Lamport, "Constructing digital signatures from a one-way function," Technical Report SRI Intl. CSL 98, 1979.
19. R.C. Merkle, "A digital signature based on a conventional encryption function," *Advances in Cryptology – CRYPTO '87*, Lecture Notes in Computer Science, Vol.293, Springer Verlag, pp.369–377, 1988.
20. R.C. Merkle, "A certified digital signature," *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science, Vol.435, Springer Verlag, pp.218–238, 1990.
21. S. Even, O. Goldreich, and S. Micali, "On-line/off-line digital signatures," *Advances in Cryptology – CRYPTO '89*, Lecture Notes in Computer Science, Vol.435, Springer Verlag, pp.263–275, 1990.
22. A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of applied cryptography*, CRC Press, 1997.
23. R. Rivest, "The MD5 message digest algorithm," *RFC 1321*, Apr. 1992.
24. FIPS 180-1, "Secure Hash Standard," NIST, US Department of Commerce, Washington D.C., April 1995.
25. Y. Zheng and J. Pieprzyk and J. Seberry, "HAVAL - a one-way hashing algorithm with variable length of output," *Advances in Cryptology – AUSCRYPT'92*, Lecture Notes in Computer Science, Vol.718, Springer-Verlag, pp.83–104, 1993.
26. R.L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystem," *Commun. of ACM*, Vol.21, No.2, pp.120–126, 1978.

27. C.S. Jutla and M. Yung, "PayTree: Amortized-signature for flexible micropayments," *Proc. of Second USENIX Association Workshop on Electronic Commerce*, pp.213–221, November 1996.
28. S.M. Yen, L.T. Ho and C.Y. Huang, "Internet micropayment based on unbalanced one-way binary tree," *Proc. of International Workshop on Cryptographic Techniques and E-Commerce, CrypTEC '99*, Hong Kong, pp.155–162, 5-8 July 1999.
29. M. Peirce, personal communication, February 2000.