A Loss-Free Micropayment Protocol for Multimedia Services

Jing-Jang Hwang[1], Jung-Bin Li[2]

[1]Department and Graduate Institute
of Information Management
Chang Gung University, Tao-Yuan, Taiwan
jjhwang@mail.cgu.edu.tw

[2]Instutue of Information Management
National Chiao Tung University
Hsinchu, Taiwan
jbli@iim.nctu.edu.tw

Abstract

A micropayment scheme is proposed to enable transactions of different denominations and to eliminate possibilities that either the vendor or the user in a payment system misbehaves to get illegal profit. The proposed offers a solution while keeping transactions efficient.

1. Introduction

The Internet is, nowadays, a source for information seekers to have a variety of services such as stock quotes, news reports, video streams, etc. These multimedia services in essence may include text, audio, and/or video information that have different values for information providers. As the cost of each kind of service differs from one anther, every service may need different numbers of tokens if single-denomination tokens, such as payword, the payment token of PayWord [1] protocol, is chosen as the information seeker's payment tool. PayWord uses a one-way hash function to produce successive tokens, and it is frequently referred to among other schemes in literature, such as Millicent [2], iKP [3], and MicroMint [1]. With the irreversible characteristic of one-way hash function, this scheme prohibits receivers of a payword token from guessing subsequent tokens while keeping the process of transactions efficient. If a service costs more than a single token, the nature of payword token leaves the potential risk to the token holder that he or she may lose some or all tokens given to the service provider without receiving anything. When the span of Internet services consistently grows, the flexibility of payment tools should be noticed.

In addition to flexibility, the issue of fairness is generally important for most on-line payments. In the environment of micropayment transactions, this issue is usually left intact. When a payword token is passed to the vendor, it is the user's potential loss that the vendor might redeem that token without sending the target information good to the user. Namely, the vendor may gain improper advantage over the user in PayWord protocol if he or she accumulates and redeems tokens without offering services.

We hereby propose a solution to prevent misconducts of both the user and the vendor while retaining its payment flexibility in transaction value. It is particularly suitable for web sites providing multimedia services that have payments of different values.

2. Literature Review

In this paper, discussion and analysis are focused on PayWord and other protocols based on the successive release of elements in a chain of cryptographic hash values. There are three participants in PayWord: the user *U*, the vendor *V* and the broker *B*. Each user has to register with information such as the broker's name, the user's name and the user's public key, with at least one broker, which offers a PayWord certificate. This relationship is represented by a PayWord certificate signed and issued by the broker, which binds the broker's name, the user's name and the user's public key together.

Before *U* requests a service from *V*, a fresh chain of hash values w_n, w_{n-1}, …, w_1, w_0 is generated by randomly picking w_n in formula (1).

$$w_{i-1} = h(w_i), \text{ for } i = n, n\text{-}1, \ldots, 1. \quad (1)$$

Notably, *h* is a one-way hash function, which is publicly known and cryptographically strong. Furthermore, w_0 represents a root of the hash chain, and is delivered to *V* at the beginning of a service session. Root w_0 authorizes *B* to pay *V* for any of the tokens in this chain that *V* redeems thereafter.

U's *i*th micropayment to *V* consists of the pair (w_i, *i*). The validity of this token can be verified by *V* using w_{i-1} which is known from the previous payment or from the commitment in case of $i = 1$.

If no dispute occurs, *V* presents w_0 and the final token received to *B*. *B* verifies their legitimacy and if successful, pays *V* the amount that corresponds to the tokens and charges the same amount to *U*'s account. When *V* misbehaves, *U* will lose at least the last token already sent to *V*. Even though the value of a payword token is not high enough to cause serious loss for the user, a persistently cheating vendor can collect a substantial amount of money by sending an unexpected service or nothing at all.

3. Revised scheme

The proposed micropayment scheme is a variation of PayWord, and it is also inspired by Asokan *et al.*'s proposal, an approach that achieves a complete and fair exchange [6]. The main idea of our scheme is to make a valid token a combination of two hash values from two independent hash chains. These two hash values can be regarded as two half-tokens. The first is forwarded to the vendor before service provision and the second is not sent until the service is provided. That is, unless the

vendor offers the requested service, a complete token cannot be collected and redeemed. This feature secures the user from the vendor's malicious redemption without offering the requested service.

Before transactions begin, *U* must register with *B* and obtain a signed certificate. Information recorded on this certificate includes *U*'s id, *B*'s id, and *U*'s public key. Thus, *U* generates fresh hash values; p_n, and q_n are randomly chosen, and *n* is decided based on his or her own need. Via the same secure one-way hash function $h(\ )$, *U* computes two hash chains $p_n, p_{n-1}, \ldots, p_0$ and $q_n, q_{n-1}, \ldots, q_0$ as shown in formulae (2) and (3).

$$p_{i-1} = h(p_i),\ i = n, n\text{-}1, \ldots, 1\ . \quad (2)$$

$$q_{i-1} = h(q_i),\ i = n, n\text{-}1, \ldots, 1\ . \quad (3)$$

The roots of the chains are p_0 and q_0, and *U* sends these values to *V* at the beginning of the service session. Thus, every legitimate token is composed of two hash values; i.e., the token used in the *i*th transaction is (p_i, q_i, i).

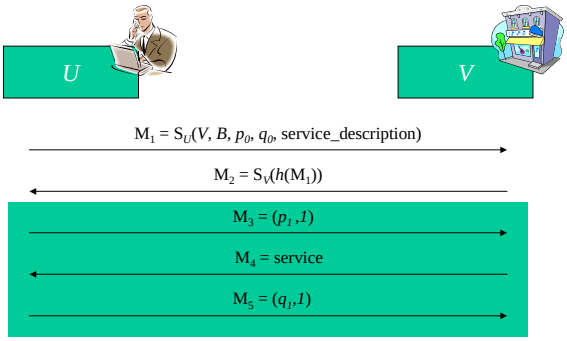

Fig. 1. The initialization steps and the first transaction of the proposed protocol

U begins a new transaction session by passing p_0, q_0, *n*, user certificate, as well as a general description of the requested service to *V* as a session commitment. A transaction session may have one or more transactions that *U* pays by the same token chain. *V* uses this commitment for token verification when tokens from *U* have been received. Then *V* signs and returns this message as his service provision commitment to offer the requested service. The initialization procedure of a transaction session is completed when the service provision commitment (M_2 in Figure 1) is transmitted to *U*. This procedure is shown as the first two messages transmitted in Figure 1.

When a transaction session begins, *U* sends (p_i, i) to *V* as the first half-token of the *i*th micropayment. As a half-token, p_i also represents *U*'s payment commitment for that transaction. *V* validates p_i by comparing p_0 and the result of hashing p_i *i* times. The root p_0 is retrieved from *U*'s session commitment. If the validity of p_i is confirmed, *V* may transmit the requested service to *U*. *U* must provide the second half-token (q_i, i) after receiving the service. The vendor examines this half-token in the same manner exactly as the first half was.

In an ideal case, *V* sends *B* the session commitment of *U* and the last token collected, requesting payment redemption. *B* pays *V* and charges *U* the value of *i* tokens.

Because the proposed enables payments that cost multiple tokens, the issue of dispute handling should be more carefully examined. If disputes occur between *U* and *V*, they can be either one of the following two cases. If *V* refuses to offer the service *U* requested after receiving the first half-token (p_i, i), *U* has no loss as long as *U* does not send (q_i, i). Even *V* receives other first half-tokens such as $(p_{i+1}, i+1)$ for successive transactions, he or she can only verify the validity of first half-tokens. Without receiving requested services, *U* will not send the second half-tokens. In this case, *V* cannot get any illegal profit by redeeming only half-tokens.

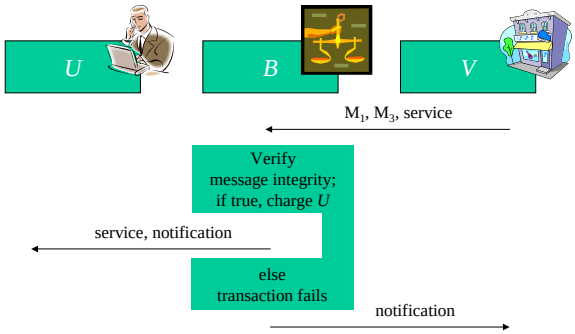

Fig. 2. Dispute handling

The other case is shown in Figure 2, which occurs when *U* refuses to pay the second half-token after receiving the requested service. *V* in this case may show *B* the information collected from *U*'s session and payment commitments previously received. The target service of this transaction also has to be sent to *B*. Being a trusted third party, *B* justifies the dispute based on the information offered by *V*. If it is *U*'s malicious behavior to refuse to pay, *B* will pay *V* from *U*'s registered account directly, and pass the target service to *U*. Hence neither *U* nor *V* benefits from misbehaving in this case.

If true fairness is not absolutely necessary or the dispute handing efficiency is a serious concern, the idea proposed by Buttyán can also be applied in this proposed scheme. *U* is charged once the first half-token is sent; regardless of whether the second half-token is paid for or not. When *V* attempts to redeem with only a half-token, he or she will not be given the value of the incomplete token. Revenues from such disputes are donated to charity.

4. Analysis

The proposed scheme makes the user free from possible losses caused by payments that cost more than

one token. And it is applicable to most practical Internet micropayment environment. When a user *U* intends to receive some information services from an on-line news web site *V*, for example, objects including text files, voice streams or even video clips satisfying the keyword are available. If these types of information objects are charged at different rates, the user of PayWord or other related extensions has to prepare token chains representing different denominations, otherwise he or she will have possible loss if paying multiple tokens of the same hash chain to a misbehaved web site. The user therefore has to prepare tokens of all denominations he or she needs before the transaction session begins. In case the user requests for a video clip that costs six tokens, either this user generates a new hash chain and tells the vendor that each token of this new chain represents exactly six units, or he has possible loss when giving all six tokens from the same hash chain. In other words, if malicious *V* is given w_6 in PayWord, all six tokens from w_1 to w_6 can be verified and redeemed directly. Similarly, if *V* is given w_6 in Buttyán's scheme, three tokens (w_1, w_2, *1*), (w_3, w_4, *2*) and (w_5, w_6, *3*) will be regarded as valid and redeemed. Another way to avoid such loss is to make every payment a combination of tokens from chains that represent different denominations. Both situations would not be convenient enough for the web surfers and would cause restriction to micropayment applications.

In addition, the PayWord user might lose all *i* tokens if he or she pays w_i for some service that costs *i* tokens. Similar situations occur in other PayWord-like micropayment schemes. In this proposed scheme, the two half-tokens, (p_i, *i*) and (q_i, *i*), are chosen from two independent hash chains. Hence even a service costs *i* tokens, *V* can only verify the received p_i. There is no means to compute a complete token unless *U* reveals the second half-token q_i. Taking the situation stated in the previous paragraph as an example, when *U* has paid p_i and the requested news report has yet to be received, *U* is free from any loss as long as he or she does not send q_i to *V*. It is easy for the on-line news web site to verify p_i via p_0, but mathematically difficult to compute or guess q_i. Hence, the web site *V* collects a full token (p_i, q_i) only if *U* receives the report and sends q_i.

In terms of efficiency, our scheme is roughly the same as Buttyán's. PayWord, Buttyán and the proposed all have the same number of public key cryptographic operations, including digital signature generation and verification. Although the proposed scheme requires twice the number of hash computation as PayWord does, it is still affordable due to the computation efficiency of hash function. Regarding transmission payload, this protocol sends 2 hash values for every consecutive transaction step, which is 1 hash value more than Buttyán's scheme. It would not cause serious network loading considering the size of a single hash value.

In comparison to Buttyán's and other PayWord-like schemes, the proposed offers better protection for both the user and the vendor. PayWord cannot justify the malicious denial of service when *V* redeems the received paywords. *U* can only minimize his or her loss by refusing to make further transactions with *V* passively, after losing some tokens for incomplete services. In other words, by accumulating tokens from numerous users, the malicious vendor still gets a fortune of illegal profit. As to Buttyán's protocol, although neither *U* nor *V* takes improper advantage over the other party, *V* still bears potential losses when *U* refuses to pay after receiving the requested service. *U* in this scenario will be charged, but the revenue will be donated to charity instead of distributing to *V*. Our scheme does not cause such losses. As long as the second half-tokens are kept by *U*, *V* cannot make any illegal profit by redeeming the incomplete first half-tokens. If *U* refuses to pay the second half-token after receiving the target service, *B* will be the trusted third party to justify whether *V* should be paid. Hence the proposed protocol causes no losses in the scenarios mentioned above.

5. Conclusion

A variation of the PayWord protocol was proposed to enable payments of different values without much loss in efficiency while the rights of both the user and the vendor are well protected. It solves the problem to pay varieties of multimedia services of different costs. Comparing to PayWord, it although requires twice as many on-line hash computations by the vendor, the complexity of hash computation should not be a real burden. Thus the advantages of this scheme lead micropayment to greater flexibility in real-world applications.

References

1. R. Rivest and A. Shamir, *PayWord and MicroMint: Two simple micro-payment schemes*, Technical report, MIT Laboratory for Computer Science, 1996.
2. S. Glassman, M. Manasse, M. Abadi, P. Gauthier, and P. Sobalvarro, 'The Millicent protocol for inexpensive electronic commerce', the Proceedings of the 4th International World Wide Web Conference, December 1995, pp. 603~618.
3. R. Hauser, M. Steiner, and M. Waidner, Micro-payments based on iKP, Technical report RZ 2791, IBM research, February 1996.
4. L. Buttyán, 'Removing the Financial Incentive to Cheat in Micropayment Schemes', Electronic Letters, January 2000, Vol. 36, No. 2, pp. 132~133.
5. L. Buttyán and N.B. Salem, 'A Payment Scheme for Broadcast Multimedia Streams', the Proceedings of the 6th IEEE Symposium on Computers and Communications, 2001, pp. 668~673.
6. N. Asokan, M. Schunter, and M. Waidner, 'Optimistic Protocols for Fair Exchange', the Proceedings of the 4th ACM Conference on Computer and Communications Security, 1997.