

MINX: Micropayments with Secure Network Exchange

Krzysztof Szczypiorski, Aneta Zwierko, Igor Margasiński

Warsaw University of Technology, Institute of Telecommunications,
ul. Nowowiejska 15/19, 00-665 Warsaw, Poland
e-mail: {K.Szczypiorski, A.Zwierko, I.Margasinski}@tele.pw.edu.pl

Abstract. This article presents two concepts of schemes – MINX: Micropayments with Secure Network Exchange – for implementation of pre-paid cards. The first scheme is based on secure one-way hash functions, the second one is based on cryptographically secure pseudorandom bit generators, which produce pseudorandom sequences. Both schemes enable the utilization of cards bought prior and make securing and hiding users' data and details of service possible. Proposed schemes are similar to existing micropayments schemes, but have distinctive features created for pre-paid cards. These features include: the possibility of using some of the impulses on a card at any time, the possibility of making a payment with the same card for services with different base-unit costs. The card usage is anonymous: a user does not have any public/private key. Transactions do not require a presence of a Trusted Third Party. The system contains another unique feature, because the schemes provide secure a communication between operator and user, without the need for any kind of key distribution scheme.

1 Introduction

Most e-commerce transactions are based on the following macropayments scheme: a user makes few but large (high cost) transactions. These kind of schemes involve public key cryptography and frequent on-line communication between a user (client), a vendor and a broker (bank). They require lots of computation and a strong cryptography, but their frequency is so low, that this does not seem to pose a problem. However, there are some kinds of services, where a user wants to make frequent payment of small amounts. Examples of such services include: buying web content on the Internet, using video-on-demand or other streaming services provided by telecommunications networks. A payment of small amounts occurring very frequently should require little computation and should be possible to be performed without a broker (off-line). In this paper we propose two new micropayments schemes (MINX - Micropayments with Secure Network Exchange) based on different cryptographic primitives (one-way functions and cryptographically secure pseudorandom bit generators). They provide both a user and a vendor with reasonably fast and secure protocol for micropayments.

2 Related Work

The most important and known micropayments schemes are PayWord and Micromint proposed by Ronald L. Rivest and Adi Shamir in 1996 [13]. Both schemes are based on Trusted Third Party (TTP) named broker and need Public Key Infrastructure (PKI) in order to work properly [9]. The first one, PayWord, is based on the idea of cryptographic one-way functions. The functions have the following properties: collision resistance (strong and weak) and one-way property (detailed definition is given in 3.1). A user produces sequence of coins using the one-way function. A payment is granted by a broker. Once a day vendors contact brokers and their money gets transferred. The mentioned scheme is easy to implement, but requires PKI and a broker as TTP. The second scheme is based on a different idea – it also uses one-way functions as a method of producing coins, but coins come from a broker and are then distributed to users. This special method makes forging coins much more difficult than producing real ones by a broker. This concept is based on the birthday-paradox for one-way functions [9].

Another micropayments scheme was proposed by Torben P. Pedersen and was named the CAFÉ project ([11], [1]). It is also based on one-way functions and it is very similar to schemes proposed by Shamir and Rivest, but was developed independently. The CAFÉ system is part of the ESPERIT project. Other schemes were proposed by Shamir (based on lottery tickets system [12], improved in [10]). A similar micropayments scheme, based on hash functions and called NetPay, was proposed by Xiaoling Dai and John Grundy [2]. Their paper also provides details of a possible architecture and an implementation of such system. The idea of combining some properties of macro- and micro-payments schemes was introduced by Stanisław Jarecki and Andrew Odlyzko in [8]. Their scheme combines the simplicity of an off-line micropayment scheme with an on-line security of a transaction, which means that a vendor is consulting with a broker from time to time during a communication with a client. This enables a vendor to check if his client is not cheating. Many other micropayment methods are discussed in [3] and [5]. One of the commercial systems was proposed by IBM: Internet Keyed Payment Systems (*iKP*), discussed in [6]. Another interesting system was proposed for multi-hop cellular networks [7].

3 Proposals of New Schemes

We propose two new schemes for micropayments. Both are pre-paid cards oriented, which means they have almost all of the advantages and disadvantages of real-life pre-paid cards. The main novel idea of MINX system is the ability to perform cryptographic key distribution with the micropayment process.

3.1 Properties of Pre-paid Cards

Pre-paid cards can be treated as kinds of micropayments. Contradictory to classic micropayment schemes, there is no trusted third party; when a user buys a pre-paid

card he/she has to trust an operator that it is valid and ready to be used by a client. In a traditional purchase (not pre-paid), a user knows exactly how it works. That is why a trusted operator is a major factor in the schemes discussed. Another advantage of a pre-paid card is the possibility of using only a part of it. A partially used card is ready to be utilized at any time. The process does not require a user to provide an operator with any information during card purchase or its usage.

Proposed schemes are based on two different cryptographic primitives:

- one-way hash functions,
- cryptographically secure pseudorandom bit generators (CSPRNG).

A hash function h maps an input x to output $h(x)$ of a fixed length. For a given x , $h(x)$ is easy to compute. A one-way hash function (h) has the following properties [9]:

- one-way (preimage resistance) – for $y = h(x)$, computing x from y is infeasible,
- weak collision resistance (2^{nd} preimage resistance) – for given x_1 and $h(x_1)$ it is computationally infeasible to find such x_2 that $h(x_1) = h(x_2)$,
- strong collision resistance – it is computationally infeasible to find such x_1 and x_2 that $h(x_1) = h(x_2)$.

Pseudorandom bit generator (PRNG) is a deterministic algorithm which for given input sequence (a truly random binary sequence) outputs with different binary sequence, much longer, which "appears" to be random. The PRNG passes the *next-bit* test if there is no polynomial-time algorithm which can differentiate between this PRNG output and a truly random sequence with probability significantly greater than $\frac{1}{2}$. The PRNG, which passes the *next-bit* test, even under some plausible, unproved mathematical assumptions, is called the cryptographically secure pseudorandom bit generator (CSPRNG).

3.2 MINX General Overview

Basic definitions:

- *key*:
in MINX system key means a secret key for symmetric cipher (like Advanced Encryption Algorithm – AES, RC6),
- *impulse*:
impulse means one unit of payment, which can be extracted from a pre-paid card,
- *ID*:
every user who wants to use his/her valid card has a unique identifier – named ID assigned by an operator; the ID enables an operator to find a proper secret key for decrypting received data from each user.

Both MINX schemes are based on the same **four steps** (Fig. 1):

- Step 1.** A user shows part of a card to an operator.

- Step 2.** The operator sends a confirmation and an assigned ID to the user; at the same time the operator computes a current key.
- Step 3.** The user computes the current key and an impulse, encrypts it with a requested data and sends it to the operator.
- Step 4.** The operator validates it and sends a response back to the user.

After the completion of step 4, it is possible to establish a secure communication between a user and an operator – a key (shared between a client and an operator) is **destined to be used as a session key in all secure exchanges between the parties until a new key gets established**. The last two steps are repeated until a user wants to use a service provided by an operator (with a set fee) or his/her virtual pre-paid card is not used up.

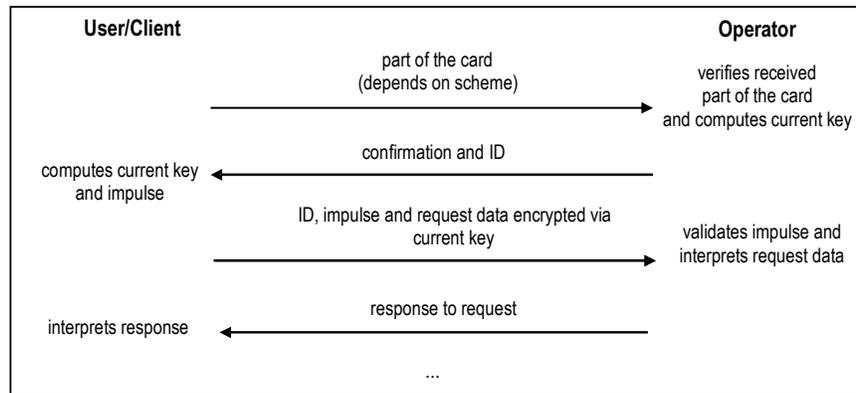


Fig. 1. MINX – four basic steps

3.3 Scheme Based on One-Way Hash Functions

A client buys a pre-paid card, which consists of 4 elements:

- secret initialization value (seed) – **x**,
- card's value,
- number of impulses – **z**,
- function for generating impulses – **h**.

A card with the above parameters has to be delivered secretly and should be authorized by an operator. We do not specify a way of buying a card. This topic is not included in this paper's scope and can be realized by a macropayment system or a physical purchase.

When a user wants to use a pre-paid card, he/she sends the following values to an operator:

- value of the card,

- number of impulses,
- $h^z(x)$ – the z^{th} hash of x value computed using h .

This initial step of communication can be kept secret. For example, a user can encrypt his/hers card with an operator's public key. The operator does not need to authorize this activity, because only when x is known to the user he/she can participate in the rest of the communication.

An operator, using $h^z(x)$ and other values send by a user (step 1), can identify a pre-paid card in its own database and validate it. While using a contemporary secure hash function, $h^z(x)$ is a unique identifier for each card. The length of $h^z(x)$ should be from 160 bits (SHA-1 – Secure Hash Algorithm) up to 512 bits (SHA-512).

If a card is valid, an operator computes the first secret key $h^{z-1}(x)$ and gives a user a unique identifier (ID), so the user's messages can be distinguished from other messages. At the same time the operator sends user confirmation and ID (step 2).

A user, after receiving a confirmation from his operator, also computes the first secret key: $h^{z-1}(x)$ and the first impulse: $h^{z-2}(x)$. Next he/she encrypts the impulse and the information about service that he/she requested with a secret key and sends it along with a unique identifier to the operator (step 3). After decryption an operator can verify the impulse value by hashing it twice and checking if it equals to what is stored in the database ($h^z(x)$). Then, a user is provided with the requested service and data for this card is changed in the database. An operator computes a new key and changes the value of card from: $h^z(x)$ to $h^{z-2}(x)$ (step 4).

When an operator receives an impulse equal to x from a user, a card gets used up. An operator should hold it in its database: $h^z(x)$, x and value of the card to be able to validate incoming cards. Changing $h^z(x)$ to $h^{z-2}(x)$ (new values) enables an operator to hold current value/number of impulses in his database. A user does not have to send every impulse to an operator. A user can show an operator that he/she wants to use more impulses at this time to pay for more expensive services or to use the service for a longer time.

The impulses themselves, in this scheme, are random-looking. Based on the properties of a hash function, it is not possible to compute x from $h^z(x)$. The reason for the implementation of additional secret keys, connected to impulses, is to provide a user with confidentiality of services, that he/she requests without the need for public key cryptography or secret-key sharing schemes.

The advantages of this scheme include:

- confidentiality of communication between a user and an operator,
- possibility of using services with different values/prices with one card,
- no need for TTP to compute impulses prior to card usage. A user does not have to request an authorization of a card.

The disadvantages include:

- computation of impulses and keys, their validation is slower then in classical micropayments schemes,
- an operator has to be trusted same as in the real world.

3.4 Scheme Based on Pseudorandom Bit Generator

This scheme is almost the same as the previous one. The only difference is that instead of the hash function, a client uses cryptographically secure pseudorandom number generator (CSPRNG). The CSPRNG is used for generating binary sequences in the manner described by Blum, Blum & Shub [9], which are treated as impulses or secret keys. The advantage of CSPRNG over hash function is that having x_n a user can compute x_{n-1} or x_{n+1} with the same amount of computation (if a user knows parameters of CSPRNG). If a user does not have these parameters the computation is very difficult (having x_n) x_{n-1} or x_{n+1} . This means that a generation and a verification of a key and an impulse take almost the same amount of time.

In this scheme a card is built of the following:

- secret seed – x ,
- card's value,
- number of impulses – z ,
- secret parameters of CSPRNG.

A user shows an operator x_z and hash of parameters of CSPRNG (step 1). The confidentiality of this operation can be based on an operator's public key.

The first key could be x_{z-1} and the first impulse x_{z-2} . An operator has only to compute x_z from x_{z-2} to verify the impulse (step 3). To check if a card is still valid and not used up an operator has to store x and z . The rest of the scheme is the same as in the previous one.

The advantages include:

- the same number of operations to generate key/impulse every time and to verify them,
- the same as in the previous scheme.

The disadvantages are:

- generating proper parameters of CSPRNG is quite complex,
- computation CSPRNG values is not very fast, and poses almost the same problems as public-key cryptosystems.

4 Sample Applications

There are at least two versions of potential MINX applications. The first one is based on an independent cryptosystem at the application layer where micropayments are provided. Keys placed on pre-paid cards are utilized to provide confidentiality for clients' requests or operators' responses including security of the content during the paying process.

It is also possible to use keys from pre-paid cards directly in existing, well known security protocols like SSL/TLS (Secure Sockets Layer/Transport Layer Security –

[4]). In this case (i.e. SSL/TLS), the adequate session key (SSL/TLS MasterKey) is extracted from a pre-paid card and is utilized to provide transaction security according to admitted context (for example duration or data volume).

5 Conclusions

The micropayments systems known from the e-commerce literature do not support confidentiality. Both original schemes presented in this article (the first one based on one-way hash functions, the second one based on cryptographically secure pseudorandom bit generators) are integrated with cryptographic key distribution. This approach creates very attractive telecommunication environment that provides the possibility of payment for access to resources without compromising users' privacy. The usage of keys placed in pre-paid cards reduces costs of key management system implementation and simplifies clients' software/hardware. Besides confidentiality, the main advantages of the proposed schemes are: a possibility of using services with different values/prices with one card and the absence of TTP.

References

1. Boly, J-P., Bosselaers, A., Cramer, R., Michelsen, R., Mjølsnes, S., Muller, F., Pedersen, T., Pfitzmann, B., de Rooij, P., Schoenmakers, B., Schunter, M., Halle, L., Waidner, M.: The ESPRIT Project CAFE. ESORICS 94, Springer-Verlag LNCS Vol. 875 (1994) 217-230
2. Dai, X., Grundy, J.: Architecture of a Micro-payment System for Thinclient Web Applications. Proceedings of the 2002 International Conference on Internet Computing (2002)
3. Dai, X., Grundy, J., Lo, B.: Comparing and Contrasting Micro-payment Models for E-commerce Systems. International Conferences of Info-tech and Info-net (ICII) (2001)
4. Dierks T., Allen C.: The TLS - Protocol Version 1.0. IETF RFC 2246 (1999)
5. Ellis, C.: Evaluation of Micropayment Schemes. Tech Report HPL-97-14 (1997)
6. Hauser, R., Steiner, M., Waidner, M.: Micro-Payments based on *i*KP. Research Report 2791 (# 89269), IBM Research (1996)
7. Jakobsson, M., Hubaux, J-P., Buttyan, L.: A Micro-Payment Scheme Encouraging Collaboration in Multi-Hop Cellular Networks. Financial Cryptography'03 (2003)
8. Jarecki, S., Odlyzko, A.: An Efficient Micropayment System Based on Probabilistic Polling. Financial Cryptography '97, Springer-Verlag LNCS Vol. 1318 (1998) 173-191
9. Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Inc. (1997)
10. Micali, S., Rivest, R.: Micropayments Revisited. CT-RSA 2002, Springer-Verlag LNCS Vol. 2271 (2002) 149-163
11. Pedersen, T.: Electronic Payments of Small Amounts. Technical Report IDAMI PB-495 (1995)
12. Rivest, R.: Electronic Lottery Tickets as Micropayments. Financial Cryptography '97, Springer-Verlag LNCS Vol. 1318 (1998) 307-314

8 **Krzysztof Szczypiorski, Aneta Zwierko, Igor Margasiński**

13. Rivest, R., Shamir, A.: PayWord and MicroMint: Two simple micropayment schemes. Proceedings of 1996 International Workshop on Security Protocols, Springer-Verlag LNCS Vol. 1189 (1997) 69-87