Micropayments with Privacy – a New Proposal for E-commerce

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 paper presents an original concept of micropayment schemes which combine both the simplicity of an electronic pre-paid card without trusted third party and user's privacy. Two protocols are proposed - the first one is based on a secure one-way hash function, the second one is based on a cryptographically secure pseudorandom bit generator. The main advantage of the proposed schemes is the ability to perform a cryptographic key distribution within the micropayment process.

Keywords: micropayments, e-commerce, privacy

1 Introduction

Micropayments (defined as electronic transactions transferring very small sums of money) are very attractive from the privacy's perspective. In this paper we propose two new micropayment schemes (MINX - **Mi**cropayments with Secure **N**etwork E**x**change) based on different cryptographic primitives (an one-way function and a cryptographically secure pseudorandom bit generator). They provide both the user and the vendor with reasonably fast and secure protocol for micropayments. The main idea of the proposed schemes is the ability to perform cryptographic key distribution with the micropayment process. Other advantages of the presented system include an immediate double spending detection, effective forgery prevention and confidentiality of transactions. Some of those unique properties were made possible by merging the role of a service provider (a vendor) and a broker (a guarantor of a payment) into one: an operator. This situation is typical in the telecommunications environment.

2 Related Work

The most important and known micropayment schemes are PayWord and Micromint proposed by Ronald L. Rivest and Adi Shamir in 1996 [14]. Both schemes are based on Trusted Third Party (TTP) named broker and need Public Key Infrastructure (PKI) in order to work properly [10]. In the first one, PayWord, an user utilizes a cryptographic one-way function to produce a sequence of coins. A payment is granted by the broker. Once a day vendors contact brokers and vendors' money gets transferred. The second scheme is based on a different idea – it also uses the one-way function as a method of producing coins, but the coins come from a broker and then are 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 [10]. Another micropayment scheme was proposed by Torben P. Pedersen and was named the CAFÉ project ([12], [1]). It is also based on the one-way function and it is very similar to the schemes proposed by Shamir and Rivest, but it was developed independently. The CAFÉ system is a part of the ESPIRIT project. Other schemes were proposed by Shamir (based on lottery tickets system [13], improved in [11]). A similar micropayment 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 micropayments schemes was introduced by Stanisław Jarecki and Andrew Odłyżko 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 a broker from time to time during communication with a client. This enables the vendor to check if the client is not cheating. Many other micropayment methods are discussed in [7] and [4]. One of the commercial systems was proposed by IBM: Internet Keyed Payment Systems (*i*KP), discussed in [6].

3 Proposals of New Schemes

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

3.1 Properties of Pre-paid Cards

Pre-paid cards can be treated as types of micropayments. Contradictory to classic micropayment schemes, there is no TTP. When a user buys a pre-paid card the user has to trust an operator that the card is valid and ready to use. In a traditional purchase (not pre-paid), the user knows exactly how it works. That is why a trusted operator is the major factor in the schemes discussed. Another advantage of the pre-paid card is the possibility to use only a fraction 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.

The proposed schemes are based on two different cryptographic primitives: the one-way hash function and the cryptographically secure pseudorandom bit generator.

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 [10]:

- 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 $h(x_1) = h(x_2)$.

A pseudorandom bit generator (PRBG) is a deterministic algorithm which for a given input sequence (a truly random binary sequence) outputs a different binary sequence, much longer, which "appears" to be random. The PRBG passes the *next-bit* test if there is no polynomial-time algorithm which can differentiate between this PRBG output and a truly random sequence with probability significantly greater than ½. The PRBG, which passes the *next-bit* test, even under some plausible, unproved mathematical assumptions, is called the cryptographically secure pseudorandom bit generator (CSPRBG).

3.2 MINX General Overview

Basic definitions:

- *key:* in MINX system a key means a secret key for a symmetric cipher (like Advanced Encryption Algorithm – AES, RC6),
- *impulse:* an impulse means one unit of payment which can be extracted from a pre-paid card,
- *ID:* every user who wants to use a valid card has a unique identifier – named ID assigned by an operator; the ID enables the operator to find a proper secret key for decrypting data received 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 a current key and an impulse, encrypts it with 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 the user and the operator – a key (shared between the user and the 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 the user wants to use a service provided by the operator (with a set fee) or the user's virtual pre-paid card is used up.

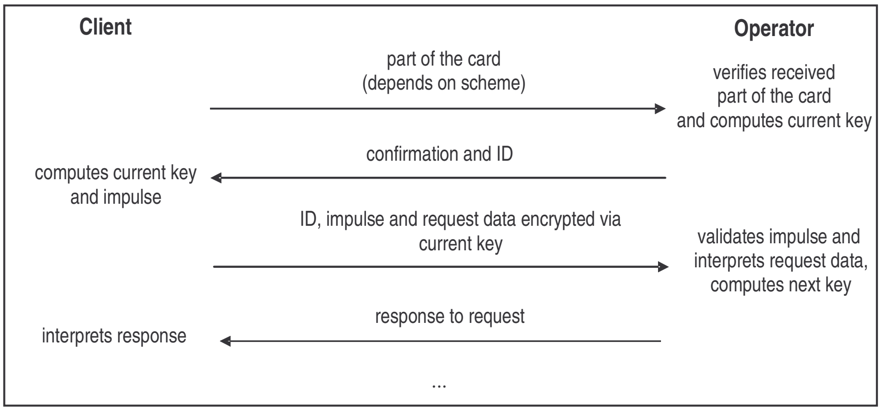

Fig. 1. MINX – four basic steps

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, the number of hashes – z (number of impulses on card is z/2) and function for generating impulses – **h**. The 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 and can be realized by a macropayment system or a physical purchase.

When a user wants to use a pre-paid card, the user sends the following values to an operator: value of the card, number of impulses and $h^z(x)$ – the z^{th} hash of x value computed using h. This initial step of communication can be kept secret. For example, the user can encrypt the card with the operator's public key. The operator does not need to authorize this activity, because only when **x** is known to the user, the user can participate in the rest of the communication.

The operator, using $h^z(x)$ and other values sent by the 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 the card is valid, the operator computes the first secret key $h^{z-1}(x)$ and gives the 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).

The user, after having received a confirmation from the operator, also computes the first secret key: $h^{z-1}(x)$ and the first impulse: $h^{z-2}(x)$. Next the user encrypts the impulse and the information about service that the user requested with a secret key and sends it to the operator along with a unique identifier (step 3). After decryption the 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, the user is provided with the requested service, and the data for this card is changed in the database. The operator computes a new key and changes the value of the card from: $h^z(x)$ to $h^{z-2}(x)$ (step 4).

When the operator receives the impulse equal to x from the user, the card gets used up. The 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 the operator to hold current value/number of impulses in its database. The user does not have to send every impulse to the operator. The user can show the operator that the user 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 the user with confidentiality of services that the user requests without the need for public key cryptography or secret-key sharing schemes.

The advantages of this scheme include:

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

The disadvantages include:

- computation of impulses and keys, their validation is slower than in classical micropayment schemes,
- the operator has to be trusted just like 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 a cryptographically secure pseudorandom number generator (CSPRBG). The CSPRBG is used for generating binary sequences in the manner described by Blum, Blum & Shub [10], which are treated as impulses or secret keys. The advantage of CSPRBG over hash function is that having x_n the user can compute x_{n-1} or x_{n+1} with the same amount of computation (if the user knows parameters of CSPRBG). If the user does not have these parameters the computation any of the values x_{n-1} or x_{n+1} is very difficult (even having x_n). This means that the generation and the verification of a key and an impulse take almost the same amount of time.

In this scheme the card consists of the following: secret seed – **x**, card's value, the number of products of CSPRBG – z (number of impulses on card is z/2) and the secret parameters of CSPRBG.

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

The first key could be x_{z-1} and the first impulse x_{z-2}. The operator only has to compute x_z from x_{z-2} to verify the impulse (step 3). To check if the 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 CSPRBG is quite complex,
- the computation of the CSPRBG values is not very fast, and poses almost the same problems as public-key cryptosystems.

3.5 Additional feature of schemes

Another feature of MINX, not commonly found in other micropayment systems, is the possibility to utilize more than one impulse at a time. The price of one unit of some services may be a multiple of the impulse, or a user may want to buy service for longer time / in larger amount. This kind of situation is typical in telecommunications: the user wants to be able to make both local and international calls using one phone card. Another typical situation is when the user wants to buy video-on-demand services for longer period of time (e.g. whole movie) without having to contact the operator every time the service bought for one impulse expires.

Let’s assume that a user has n impulses on a card ready to utilize. Now the user wants to pay for a service that would cost him m impulses. We assume that m < n, because otherwise buying a service would be impossible. The user sends to the operator $h^{(n-m-1)*2}(x)$ ((n-m-1)*2'th hash), including in the data sent the information that the user wants to utilize m impulses. The encryption of information is done with hash $h^{(2*n-1)}(x)$, using the session key that was previously computed by both the user and the operator. To check the validity of the impulse the operator hashes 2m times the delivered impulse and compares it with the information stored in the operator’s database. If the impulse is valid, the operator computes new session key: $h^{((n-m-1)*2-1)}(x)$ and sends requested service to the user.

A card based on the CSPRNG can be used in the same way (appropriate products of the generator instead of hashes should be computed).

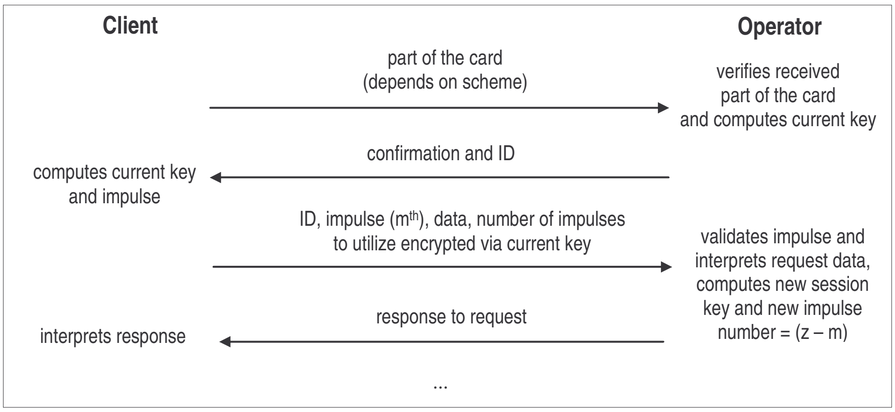

Fig. 2. Basic scenario for paying bigger amounts with MINX

This feature of the MINX gives a user a greater flexibility when using a card. Not only can the card be used at any moment, but also different amounts of impulses can be utilized at the same time. This feature makes the card suitable for a wider range of services.

4 Security

The security of both schemes is based on the same assumptions, therefore the security is analyzed for the overall scheme.

Main security assumptions are:

- the operator is trusted,
- the utilized hash function is hard to invert,
- the CSPRBG has properties as in point 3.1.

The security evaluation criteria, discussed below, are based on the proposals included in [4].

4.1 Forgery Prevention

There are two types of forgery in the proposed schemes: forgery of a card and forgery of an impulse.

In the first case, a malicious user can send a regular request for a validation of a card which is really invalid. If an operator cannot find such a card in its database, the situation is clear: the card is invalid and no validation can be made. No fraud is possible. But there can be a card with the same ID in the database. In this case, the operator computes the session key and sends validation along with ID to the malicious user. Still, the user is not able to use the card: the user is not able to invert hash function or compute following number of the CSPRBG to use it as the key. Therefore, the user cannot decrypt the data sent by the operator. The user is still also not able to compute the proper impulse value. The only situation in which the user can gain anything from guessing hash values from the card is when the user would produce all hashes or CSPRNG products (so all impulses and keys for this card), in proper amount (so their amount is the same or smaller than the operator has in its database). But this is very computationally ineffective and the probability of success is very small (when the operator uses large enough seed). It is even more ineffective for the CSPRBG than for the hash function scheme, because apart from computing impulses and keys, the user has to also find parameters of the pseudorandom generator and then try with all possible seed values. Therefore, the probability of such an attack is very small.

Another similar situation occurs when the user wants to use more impulses than the user's card poses. This is not possible since the operator stores the number of impulses in its database, and marks the card as used when this number reaches zero. At that moment the user has the same chance of deceiving the operator as in the case described earlier.

4.2 Double Spending Detection

The double spending of an impulse is almost impossible in the proposed schemes if a proper hash function or a CSPRBG is chosen (as described in section 3.1). During the transaction an operator is able to check validity of every impulse sent by a user. The operator also stores the number of impulses that the user is still able to spend. So, there is no sense in trying to spend the same impulse twice, if the user has a proper card, because the user is restricted in the number of impulses. Moreover, the possibility of having an impulse that is the same as the subsequent one is negligibly small for the proper hash functions and the CSPRBGs. As a result, in the proposed schemes, the double spending detection is immediate and double spending is not only nearly impossible but also non-profitable for the user.

4.3 Confidentiality

A third party can observe only a part of communication between a user and an operator that is not encrypted. As a result, it is only possible to misuse the information contained in the request for validation that the user sends to the operator in open-text. Only important data sent by the user is an initial impulse. If an eavesdropper could invert an impulse and generate a session key or the next impulse, this information would be very useful. However, since the hash function in the proposed scheme should be hard to invert and the generator used should be cryptographically secure, it is not possible. The rest of the communication between the operator and the user is encrypted, so it is secure and no one is able to misuse it.

4.4 Anonymity

The presented schemes also provide users with anonymity. No one observing communication between the user and the operator is able to acquire any information about a service that the user is requesting. It is also not possible to see how much money/impulses the user spent in a current transaction. Only when observing every transaction, especially every request for validity sent by the user to the operator, it is possible to identify how many impulses the user spent in the last transaction. But this problem can be eliminated by not including the number of valid impulses left on the card, in the request for validation. This data is needed only for additional validation of a card, so it is not necessary for the operator. This small change in the scheme equips a user with a complete anonymity against all observers. Unfortunately, the user is not that anonymous to the operator, who knows what services were sent and where. But this obvious disadvantage is reduced by the operator being a trusted party.

5 Applications

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

It is also possible to use the keys from pre-paid cards directly in the existing, well known security protocols like SSL/TLS (Secure Sockets Layer/Transport Layer Security – [3]). 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 the admitted context (for example duration time or data volume).

The presented micropayment schemes can be useful in case where users wish to protect their privacy during small, frequent payments. Considering repeated payments via Internet, there is a serious possibility of spying, tracing and profiling users. MINX is the solution for customers who prefer to protect information about their favourite products' preferences and e-commerce habits. Therefore, MINX, as a means of a payment for anonymity, is the next important application field. Implementations of anonymity providing systems equipped with the payments for the services occur very rarely – known systems from state of the art (e.g., Freedom [5]) include payments which can compromise users' privacy. The anonymity service providers will need effective methods for generating revenue, as they would not look for profits from advertisements or user profiling attempts. The MINX micropayment schemes, which protect consumers' anonymity, can be an effective method of payment for Web anonymity service on the Internet. Association of MINX and VAST [9] (*Versatile Anonymous System for Web Users*) – which provide anonymity for individuals browsing WWW pages – seems to be a solid and practical solution for Web privacy. Combination of VAST anonymous Web browsing with MINX anonymous payments for the service is a good proposal for widespread, commercial implementations.

6 Conclusions

The proposed schemes, which are similar to the existing micropayment systems, have distinctive features including: 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 the presence of a

TTP. Another unique feature of the proposed schemes is preservation of privacy: the micropayment systems known from the e-commerce literature do not support confidentiality. MINX provides a secure communication between an operator and a user without the need for any kind of key distribution scheme. This approach creates very attractive telecommunications environment that provides the possibility of a payment for an access to resources without compromising users' privacy.

7 References

[1] Boly, J-P., Bosselaers, A., Pedersen, T. et al.: 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] Dierks T., Allen C.: The TLS - Protocol Version 1.0. IETF RFC 2246 (1999)

[4] Ellis, C.: Evaluation of Micropayment Schemes. Tech Report HPL-97-14 (1997)

[5] Goldberg, I., Shostack, A. Freedom Network 1.0 Architecture and Protocols. Zero-Knowledge Systems. White Paper, 1999.

[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., Odłyżko, A.: An Efficient Micropayment System Based on Probabilistic Polling. Financial Cryptography '97, Springer-Verlag LNCS Vol. 1318 (1998) 173-191

[9] Margasiński, I., Szczypiorski, K.: VAST: Versatile Anonymous System for Web Users. Enhanced Methods in Computer Security, Biometric and Artificial Intelligence Systems, Springer-Verlag (2004)

[10] Menezes, A., van Oorschot, P., Vanstone, S.: Handbook of Applied Cryptography. CRC Press, Inc. (1997)

[11] Micali, S., Rivest, R.: Micropayments Revisited. CT-RSA 2002, Springer-Verlag LNCS Vol. 2271 (2002) 149-163

[12] Pedersen, T.: Electronic Payments of Small Amounts. Technical Report IDAMI PB-495 (1995)

[13] Rivest, R.: Electronic Lottery Tickets as Micropayments. Financial Cryptography '97, Springer-Verlag LNCS Vol. 1318 (1998) 307-314

[14] 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