COMBINATION OF A SMARTCARD E-PURSE AND E-COIN TO MAKE ELECTRONIC PAYMENTS ON THE INTERNET

Antonio Ruiz-Martínez, Antonio F. Gómez-Skarmeta
Department of Information and Communications Engineering, University of Murcia, Murcia, Spain

Óscar Cánovas
Department of Computer Engineering, University of Murcia, Murcia, Spain

Keywords: Smart card e-purse, e-cash, Internet payments, e-commerce.

Abstract: Nowadays e-purses are not being offered as payment method on the Internet. This is mainly due to the fact that vendors have to integrate in their devices a security application module (SAM) to exchange security messages between the e-purse and that module during the payment phase. In this paper we introduce a new payment method that combines the main advantages of e-purses and the use of e-coins to make payments. This proposal does not need a SAM to make and verify payments on the Internet. Furthermore, it does not require the e-coin to be checked on-line. Thus, we introduce the possibility that this e-purse can be easily integrated in payment applications that vendors offer on the Internet.

1 INTRODUCTION

Although smart cards acting as e-purses are more secure than credit cards, and are very commonly used in Automatic Teller Machines (ATMs), they are not widely deployed as payment method on the Internet. Nowadays, if a vendor wants to sell his products on the Internet, by offering an e-purse as payment method, he has to integrate, in his e-commerce application, some points of sale (POS) devices with some Secure Application Modules (SAMs) to make the payment authentication between e-purse and SAM (CEPSCO, 1999), (EMV, 2000). This integration is not so straightforward and the transactions are slow due to the exchange of some messages, through a HTTP/TCP connection, between the e-purse and the SAM.

On the other hand, e-coins are more suitable to make payments on the Internet because they are based on sequences of bytes that can be easily conveyed as part of the purchase information. Besides, they can generally be checked via software and therefore there is no need for special hardware. In general, e-coins can be classified as generic or vendor-specific depending on whether they can be used with any vendor or only with a specific one. The main advantage of generic e-coins is that they can be used with any vendor. The main disadvantage of the previously proposed schemes is that it is necessary to check on-line, with the issuer, that the e-coin was not previously delivered to another vendor in order to avoid the double-spending (Chaum, 1998, 2000), (Peha, 2003).

On the other hand, vendor-specific e-coins allow a better control of double-spending because they are controlled by the vendor. The main problem is for the user, who has to deal with several issuers. Besides, he could end up with an important quantity of money which might not be used with any other issuer (Glassman, 1995), (Rivest, 1996), or that cannot be divided into smaller pieces, such as Payword (Rivest, 1996). However, from the vendor's point of view, it is very easy to check or to integrate e-coins in their system because all the verifications can be done via software or using on-line connections.

In this paper, we propose a new method that combines the advantages of both e-purses and e-coins. On the one hand, it is secure and portable as an e-purse because is a payment application stored in a smartcard. As any other e-purse system, e-money can be spent with any vendor. On the other hand, during the payment stage, the e-purse does not

Ruiz-Martínez A., F. Gómez-Skarmeta A. and Cánovas Ó. (2006).
COMBINATION OF A SMARTCARD E-PURSE AND E-COIN TO MAKE ELECTRONIC PAYMENTS ON THE INTERNET.
In *Proceedings of the International Conference on Security and Cryptography*, pages 203-206
DOI: 10.5220/0002104002030206

exchange messages with a SAM, but it generates a vendor-specific coin. This way the e-coin can be checked without an on-line connection with the bank. Furthermore, the vendor does not need any hardware to receive e-coins from the e-purse.

2 SYSTEM DESIGN

Our business model is based on prepayment and the participating entities are: client, vendor and e-purse issuer (usually, a financial service provider).

In this model, an essential requirement is that clients have a smart card with an e-purse which contains a private key and an e-purse certificate as the basis to make payments. It is worth noting that this key never leaves the smart card. The idea is that our e-purse, instead of exchanging APDUs with a SAM, generates vendor-specific e-coins. Thus, vendors only need to verify digital signatures and certificates in order to accept payments. The e-purse certificates are verified against a set of root certificates from trusted issuers. Optionally, depending on the issuer, vendor might need to manage certificate revocation lists which would be periodically distributed. Therefore, they do need neither a SAM device nor making an on-line connection for each payment. In this model, there are two entities that can generate e-coins: the e-purse issuer, to increase the card balance; and, the e-purse, to make payments to vendors.

When the user obtains the e-purse, if the private key and the certificate have not been previously pre-installed, he has to make a process of certification as it appears in phase I, in Figure 1 (*e-purse certification*). This process is similar to request a certificate to a PKI.

Before making payments with the e-purse, it is necessary to load the e-purse with e-money. This process is named *e-purse load* (Figure 1, phase II).

When a client wants to pay, the e-purse has to generate a vendor-specific e-coin of the appropriate amount. Next, the client sends the e-coin to the vendor by means of a previously agreed protocol. Then, he receives the product or the access to the service. These steps correspond with phase III (*payment*). Later, when the vendor estimates, he sends to the e-purse issuer the e-coins received as payment in order to be paid (phase IV-*deposit*). The e-purse issuer is simply called *issuer* from now on. His management tasks are vendor's deposit of e-coins, redemption of the e-coins received from vendors, transfer the total amount to their accounts, load of e-purses, operations related to certificates and detection of possible frauds.

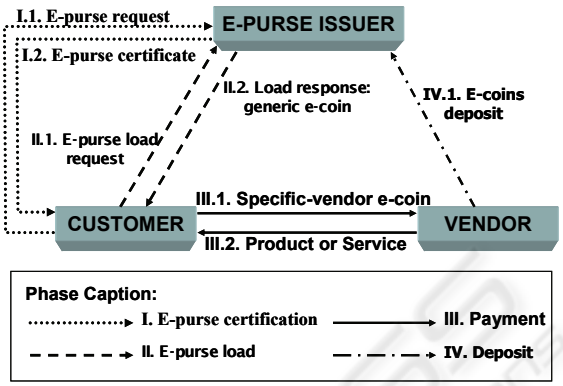

Figure 1: Business model.

2.1 Notation

In this section we describe the notation that we have used in the specification of our e-purse.

Table 1: Notation.

[Data]	It indicates that this piece of *Data* is optional, and it could not be in the message.
H(Data)	A message digest of *Data*, obtained using a hash algorithm as SHA2.
$\|Data\|_K$	*Data*, encrypted by a symmetric cipher using the key *K*.
$\{Data\}_X^{-1}$	*Data* is signed using a private key of *X*.
X => Y	It indicates that *X* sends one message to *Y*.
Ct, Vn	Customer and Vendor respectively
EP, EPI	E-purse and E-purse Issuer respectively.

2.2 E-purse Features

The e-purse stores the following information:

- *Serial number (SN_E)*. It is an array of bytes identifying the e-purse univocally.
- *Balance*. This counter stores the actual amount of money in the e-purse.
- *Transaction counter*. It stores the number of transactions made by the e-purse, that is, the *transaction number (TN_E)*. Each time an operation is made, its value is increased.
- *Private key*. It is any asymmetric key that supports digital signature operations.
- *Certificate*. It is generated by the issuer and it is according to the X.509v3 certificate format.
- *Issuer public key*. This key allows checking the information received from the issuer.

The e-purse should also provide the usual operations related to management of the private key and the certificate as well as operations (query, update,…) related to the fields mentioned in this section.

2.3 E-purse Load

The issuer can offer several payment methods to increase the e-purse balance such as: credit cards, bank transfers and so on. Besides the payment information, the user sends some fresh information to avoid replay attacks that could lead to increase several times the e-purse. Once the payment is made, the issuer mints an e-coin with this information. This e-coin is sent to the e-purse and, after being properly verified, the balance is increased with the indicated amount. This is the information exchanged during the load operation:

1. Ct => EP: (load initiation command)
2. EP => Ct: $\{SN_E, TN_E, RandNum\}_E^{-1}$
3. Ct => EPI: (Payment, $\{SN_E, TN_E, RandNum\}_E^{-1}$)
4. EPI=> Ct => EP: $\{SN_E, TN_E, RandNum, Amount\}_{EPI}^{-1}$

2.4 E-coin

In this proposal, an e-coin has the following format:

$$e\text{-}coin_E = \{SN_E, VnID, TID, RandNum, Amount\}_E^{-1}$$

where *VnID* is a vendor's identifier (the hash of his public key or an e-mail address). *TID* is a transaction identifier which is provided by the vendor in the payment process, and it is formed by the hash of the transaction information. Finally, *RandNum* is a random number.

2.5 Payment and Deposit

The payment process consists of generating a vendor-specific e-coin that will be sent by the user to the vendor using a previously agreed payment protocol. When the vendor receives and verifies the payment, he delivers or provides the request product or service. In this section, we explain how an e-coin is generated but we do not specify the protocol to send the e-coin and receive the product since it is out of the scope of this paper.

To mint an e-coin the e-purse needs the information shown in the Step 1. Next, if there is enough balance, the e-purse mints an e-coin and decreases its balance.

When the vendor receives the e-coin and the certificate, he checks that the certificate is still valid, that the e-coin was signed with the private key associated to that certificate, and the e-coin's amount. In that case, the vendor will provide the product or service requested. It is worth noting that there is no need for on-line verifications or SAM modules. Next, we show the information exchanged.

1. Vn => Ct: (VnID, TID, Amount)
2. Ct => EP: (Command to generate e-coin with VnID, TID, Amount)
3. EP => Ct: $\{SN_E, VnID, TID, RandNum, Amount\}_E^{-1}$, $Cert_E$
4. Ct => Vn: $\{SNE, VnID, TID, RandNum, Amount\}_E^{-1}$, $Cert_E$
5. Vn => Ct: *product or service*

The vendor can store the different e-coins received, which will be deposited at the end of each day (or other suitable period) by sending them to the issuer. In this way, the vendor will get paid.

3 SECURITY ANALYSIS AND BENEFITS

In this section we analyze both the security of this new e-purse and its main benefits. Regarding the security analysis:

1. *E-purse*. The e-purse is a tamper-resistant device that manages its private key and the operations mentioned in section 2. This key never cannot be exported in order to prevent the generation of fake money.
2. *Security in e-purse load process*. The security of the whole transaction depends on both the protocol used to pay and receive e-coins, the security of the e-coin itself and how is loaded in the e-purse. The protocol to pay and receive the e-coin is out of the scope of our proposal. This process should be made using a fair protocol. On the other hand, the security of the issuer e-coin depends on the length of the issuer's private key. If the length is long enough we could be sure that nobody, except the issuer, can generate a valid coin. If the issuer's private key was compromised, the certificate would be revoked. Thus, the e-purse increases its balance after receiving an e-coin signed by the issuer and containing the information indicated in step 2 of the load process.

3. *Security in e-coin generation process*. If we guarantee this process, we can be sure that nobody, except an e-purse, could mint an e-coin. To generate e-coins without the e-purse, we would have to sign some information with a private key which is certified by the issuer. So the user should obtain an e-purse private key, that is, he would have to hack the e-purse. Anyway, the cost of such type of attacks might be even higher than the benefit obtained. Besides, after detecting fake money, the related certificate will be revoked, and therefore the private key. Finally, since we mint a vendor-specific e-coin, only the true vendor can deposit the e-coin.
4. *Double-spending*. The vendor uses *TID* value provided during the payment phase, to check whether the e-coin was previously delivered.
5. *Security in a payment*. In this phase the security depends on both the security of the e-coins and the payment protocol involved. The protocol should guarantee fairness and provide enough information to resolve conflicts.
6. *Non repudiation*. It is impossible to mint e-coins unless e-purse private keys are compromised. Therefore, any minted e-coin should be accepted by a vendor except when it has been previously delivered.

Next, we underline our proposal's advantages:

1. *Prepayment*. Prepayment systems are well accepted by both end users, since it is comfortable and anonymous, and financial entities since they receive the money in advance
2. *Portability*. User can convey comfortably his money because is stored in his smart card.
3. *Generic e-coins*. "E-coins" contained in the user's e-purse are generic, and then, they can be used with any vendor.
4. *Divisibility*. We can specify the exact amount of e-coins.
5. *Reduction of the number of elements in the system*. The vendors do not need either a SAM or an on-line connection with the issuer to verify e-coins. Therefore, the exchange of messages to make a payment is reduced, the payment process is faster and the costs of transaction are lower.
6. *Pay-per-click*. This scheme could be easily introduced to make payments-per-click as well as in mobile phones or in Bluetooth devices.
7. *ATM*. Due to the fact that this e-purse has been designed to avoid the on-line connection with the issuer, it could be incorporated easily in any POS (Point of Sale).
8. *Anonymity*. Since the e-coin does not contain any personal information, the payment is anonymous.

4 CONCLUSIONS

We have proposed a payment scheme based on e-purses in which the payment can be checked by software without having special keys in a SAM. Our contribution solves this problem with a payment method based on smart cards that combines the advantages of an e-purse with the use of a vendor-specific e-coins. Unlike others proposals, we do not need the e-coin to be validated against a third party. Besides, the e-purse can generate e-coins for any vendor. In such way, we can conclude that the incorporation of e-purse payment to the Internet applications is facilitated against some previous proposals.

As future research directions we are considering the integration of this e-purse with a fair protocol.

ACKNOWLEDGEMENTS

This work has been partially supported by PROFIT SESTERCIO FIT-360000-2005-23 project.

REFERENCES

CEPSCO, 1999. CEPSCO LLC: Common Electronic Purse Specifications, March 1999.

Chaum, D., Fiat, A., Naor, M., 1988: Untraceable electronic cash. In Advances in Cryptology-CRYPTO'88, volume 403 of Lecture Notes in Computer Science, pages 319-327. Springer-Verlag.

EMV, 2000 Integrated Circuit Card Specification for Payment Systems, December 2000.

Glassman, S. et al, 1995: The Millicent protocol for inexpensive electronic commerce. World Wide Web Journal, 4th International WWW Conference Proceedings, pages 603-618, December 1995.

GlobalPlatform, 2000: Open Platform Card Specification v2.0.1. April 2000.

Peha, J. M., Khamitov, I.: Pay Cash, 2003: A secure Efficient Internet Payment System. Proceedings of 5th Intern. Conference on E-Commerce, October 2003.

Rivest, R. L., Shamir, A., 1996: Payword and Micromint: two simple micropayment schemes. Proc. of Intern. Workshop on Security Protocols, Lecture Notes in Computer Science n 1189, p. 69-87. Springer, 1997.