An Efficient Anonymous Scheme for Secure Micropayments[*]

Magdalena Payeras-Capellà, Josep Lluís Ferrer-Gomila, and Ll. Huguet-Rotger

Universitat de les Illes Balears
Carretera de Valldemossa Km. 7.5, Palma de Mallorca, 07122, Spain
{mpayeras, dijjfg, dmilhr0}@uib.es

Abstract. Micropayment systems allow payments of low value. Low cost for transaction is the main requirement for micropayments. For this reason, anonymity, a desired feature, is difficult to implement in micropayment systems. This paper presents an anonymous and untraceable micropayment system that offers low transactional costs while the use of secure protocols avoids financial risks. The efficient anonymity is achieved thanks to the use of a double spending prevention technique. Finally, the bank behavior can be verified.

1 Anonymity in Micropayments

The ideal features of micropayments are: low transactional costs, financial risks control, atomic interchange, privacy and velocity. The achievement of these features requires adjustment in: number of interactions among parties, volume of information, use of asymmetric cryptography, storage requirements, use of tamper resistant devices, anonymity and use of specific coins. As a consequence, there is a relaxation in security and privacy aspects, and because anonymity and efficiency are in conflict, micropayment systems are rarely anonymous. Privacy can be achieved with the use of pseudonyms [3, 6], but when anonymity is implemented [7], the high computational cost associated is not suitable for low valued payments. In the existing proposals, we can distinguish systems that use universal coins and systems that use specific ones (can be spent only at a defined merchant [2, 4, 9]).

Neither on-line nor off-line systems are suitable for anonymous micropayments. On-line systems aren't efficient. In other hand, off-line systems with specific coins [8, 9] can't be anonymous (credit systems) and off-line systems with universal coins require double spending detection or tamper resistant devices. Another approach is the use of semi-offline systems [2, 3, 5]. The better combination of efficiency and anonymity in micropayments appears in systems that use semi-offline payments, and more exactly in those that achieve *double spending prevention* without the need of hardware devices. We present a micropayment system that allows anonymous payments while maintains the associated costs at a low level, using specific coins (formed by a chain of *coupons* that can be used in independent payments [4, 9]), and semi-offline payments. Customers only need to communicate with the bank when they have to create a new chain of coupons (coin). In order to maintain efficiency our scheme presents a new algorithm for *double spending prevention*. This way, anonymity doesn't need to be revocated and coins don't include identifying information.

[*] Partially supported by the Balearic Islands Government under project PRDIB-2002GC3-18.

J.M. Cueva Lovelle et al. (Eds.): ICWE 2003, LNCS 2722, pp. 80–83, 2003.

2 Efficient Anonymous Micropayment Scheme

In the description of the scheme, *A* realizes a payment to *B*, with a coin obtained from *F* (a financial entity that uses the key K_{sfQ} to create coins with value *Q*). The notation *H( )*, $Sign_i( )$, K_{pi} and K_{si} will be used to represent a hash function, a digital signature and the public and secret key of user *i,* respectively. The withdrawal coins are $M=K_{sfQ}(W_0)=Sign_{fQ}(W_1)$, in where W_1 and $W_0=H(W_1)$ are the secret proof and identifier of the coin, respectively. The withdrawn universal coins can be changed for specific coins: $K_{sfQ}(W_{0b}, W_{0a}, n)$.

2.1 Withdrawal Subprotocol

A requests a new coin to *F*, proving ownership of his account. A random number W_1 that must be kept secret will be the proof to validate the coin. W_0 is used to prevent double spending. In order to avoid traceability, a blind signature [1] is used. *A* includes the amount *Q* and the blinded identifier *W* in the request to *F*. *F* validates the identity of *A*, and encrypts the blinded identifier. *A* calculates de coin M_1 from the blinded coin M_1'. The identifier is the digest of the secret proof W_1, so M_1 is a signature on W_1. Now, the coin is universal. Only *A* can prove the knowledge of W_1.

This withdrawal provides anonymous universal coins, security against counterfeiting and authentication of the account's owner. Moreover, the subprotocol uses a secret proof of validity that will be required for the redemption of the coin.

A → F:	$\{Q, W, Sign_a(Q, W)\}$
F → A:	$M_1' = K_{sfQ}(W)$

2.2 Pre-payment Subprotocol

When *A* wants to realize a payment to *B, A* changes a universal anonymous coin for a coin specific for *B.* With this purpose, *B* generates a new pair of identifier and secret proof W_{1b} and W_{0b}, then sends to *A* the signed identifier and his certificate. *A* generates a chain of coupons from a random number, applying successively a hash function ($W_{na},\ldots,W_{0a}=H^n(W_{na})$) and sends to *F* the amount (Q_2), the identifier (supplied by *B*), the secret proof of the universal coin (W_1), the last item of the new chain ($W_{0a,}$ it is possible to calculate an element W_{ia} from another element W_{ja} only if *j* is greater than *i*), the number of coupons (*n*) to be generated and a new identifier (W_{0x}) to create a new coin with the remaining value (Q_3). The secret proof W_1 is encrypted to avoid counterfeiting. *F* checks the validity of the coin using W_1.

B → A:	$\{W_{0b}, Cert_b, Sign_b(W_{0b})\}$
A → F:	$\{M_1, K_{pf}(W_1), Q_2, W_{0a}, W_{0b}, n, W_{0x}\}$
F → A:	$\{M_2 = K_{sfQ2}(W_{0b}, W_{0a}, n), M_3 = K_{sfQ3}(W_{0x})\}$

2.3 Payment Subprotocol

Once executed the pre-payment subprotocol, *A* knows the coin M_2 and its identifier W_{0b}, but *A* doesn't know the proof of validity W_{1b}. Only *B* knows W_{1b} and for this reason, he can be sure that the coin has not been spent before the payment. There is no need to contact the bank on-line to check double spending. *A* sends a message formed by the coin (M_2), a coupon of the chain and the order number of this coupon. *A* can send an arbitrary pair of coupon and order number. The difference between this number and the last spent one multiplied by the value of each coupon represent the amount transferred. *B* checks the validity of the coin and saves M_2 and W_{0a}. These operations are done only once. *M* gets the order number *i* and applies *i* times the hash function over W_{ia}. If the result is the value W_{0a} the payment is valid. The elements of the customer's chain are revealed in upward order, so if an element with an order number minor than a used element is presented, the merchant detects a double spending attempt. In later payments the same operations are done with the last used order number (j, W_{ja}) instead of W_{0a}.

A → B: $\{M_2, W_{ia}, i\}$

2.4 Deposit Subprotocol

B can deposit the received coupons although he hasn't received all *n* coupons of the coin. *F* checks the secret proof (W_{1b}), the relation between W_{ia} and W_{0a} and the list of spent coupons of the coin comparing the value of the order number of the last deposited coupon (*j*) with the included in the deposit message (*i*). *F* will credit *B*'s account with the value of the group of deposited coupons (*i-j* coupons). If *i* is lower or equal to *j*, then double deposit is detected. Only *B*, with the knowledge of the coin's secret proof, is able to deposit.

B → F: $K_{pf}(M_2, W_{ia}, W_{1b}), ID_b$

2.5 Verificability of the Bank

It is possible verify the behaviour of *F* modifying the scheme. A specific proof must be generated for each coupon in the chain. *B* uses a hash function to generate a chain of proofs. When *A* requests the conversion of coin M_1 in the pre-payment, *F* checks if the coin has been converted or deposited before, preventing double spending. If the coin is valid, *F* sends an *Ack* (Ack = $Sign_f(W_0, W_{0b})$) to *A,* declaring that it will refresh a valid coin M_1. If *F* detects a double spending attempt, he sends a *Nack* (Nack=$Sign_f(W_1)$), showing the secret proof given by *A* when the coin was first deposited. The Payment doesn't suffer changes. Each element in the chain will be used to deposit payments with each coupon of the coin. For each deposit of coupons of the same coin, *B* shows a different secret proof. *F* needs to prove the knowledge of the right secret when detects double spending or double deposit and can't claim a non-existent reutilization: the bank is verifiable.

3 Conclusions

Although our payment subprotocol presents a minimised number of interactions, in the semi-offline system has to be considered, for each payment, the proportional part of transactional cost involved in the pre-payment (amortised in a high number of payments using coupons of the same coin). The use of asymmetric cryptography is minimised (withdrawal, transmission of the identifier between receiver and payer and transfer of secret proofs) and it's not used in the payment stage.

Counterfeiting, overspending and robbery are avoided. All parties can be sure the payments can be redeemed. In order to prevent double spending, the receiver maintains a list of identifiers together with their secret proofs until the reception of all coupons of the related coin or their expiration date. When payments are received, the receiver adds to the list the order number and the value of the last received coupon. The bank prevents double deposit listing the identifiers (secret proofs in the verifiable scheme) only for those unexpired coins received in deposit or pre-payment. If the receiver detects a payment with a double spent coupon (the order number of the coupon is lower than the stored one), the receiver prevents the double spending rejecting the payment. If a coin is reused, the related identifier is not found in the list of valid coins.

Anonymity in an off-line payment without the inclusion of identifying information in coins is possible due to our double spending and double deposit prevention technique. The traceament of payments is not possible even by a collusion between the receiver and the bank.

The presented micropayment scheme has a high degree of efficiency together with control of financial risks and anonymity (and untraceability) of the payer in front of the receiver and the bank. Finally, we have presented a solution in where the bank is verifiable.

References

[1] Chaum, D.: “Blind signatures for untraceable payments”, Crypto'82, pages 199–203, Springer Verlag, 1982.

[2] Glassman, S. et Al.: “The Millicent protocol for inexpensive electronic commerce”, 4th International World Wide Web Conference Proceedings, pages 603–618, O'Reilly, 1995.

[3] Gabber, E. and Silberschatz, A.: “Agora: A minimal distributed protocol for electronic commerce”, 2nd USENIX workshop on Electronic Commerce, pages 223–232, 1996.

[4] Hauser, R., Steiner and M. and Waidner, M.: “Micro-payments based on iKP”, 14th Worldwide Congress on Computer and Communication Security Protection, pages 67–82, 1996.

[5] Jarecki, S. and Odlyzko, A: “An efficient micropayment system based on probabilistic polling”, Financial Cryptography'97, LNCS 1318, pages 173–191, Springer Verlag, 1997.

[6] Lipton, R.J. and Ostrovsky, R.: “Micro-Payments via efficient coin flipping”, Financial Cryptography'98, LNCS 1465, pages 1–15, Springer Verlag, 1998.

[7] Mao, W: “A simple cash payment technique for the Internet”, ESORICS’96, LNCS 1146, pages 15–32, Springer Verlag, 1996.

[8] Rivest, R.: “Perspectives on financial cryptography”, Financial Cryptogaphy’97, LNCS 1318, pages 145–149, Springer Verlag, 1997.

[9] Rivest, R. and Shamir, A.: “Payword and Micromint: two simple micropayment schemes” 4th Workshop on Security Protocols, LNCS 1189, pages 69–87, Springer Verlag, 1996.