

AN INTERNET BASED ANONYMOUS ELECTRONIC CASH SYSTEM

Israt Jahan and Mohammad Zahidur Rahman

Department of Electronics and Computer Science
Jahangirnagar University, Savar, Dhaka, Bangladesh.
E-mail: isratju1@yahoo.com

ABSTRACT

There is an increase activity in research to improve the current electronic payment system is parallel with the progress of internet. Electronic cash system is a cryptographic payment system which offer anonymity during withdrawal and purchase. Electronic cash displays serial numbers which can be recorded to allow further tracing. Contrary to their physical counterparts, e-cash have an inherent limitation; they are easy to copy and reuse (double-spending). An observer is a tamper-resistant device, issued by the Internet bank, is incorporated with the Internet User's computer that prevents double-spending physically, i.e., The user has no access to her e-cash and therefore he cannot copy them. In this paper, we shall present an anonymous electronic cash scheme on the internet which incorporates tamper-resistant device with user-module.

1 INTRODUCTION

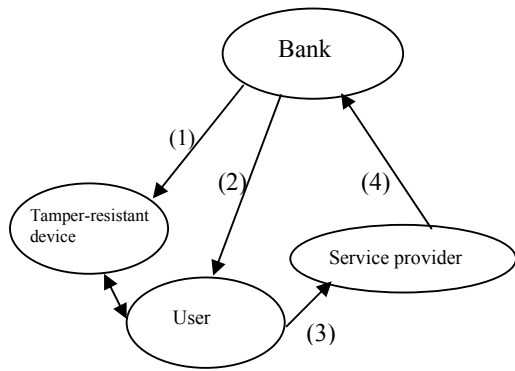
Electronic commerce is one of the most important applications for the internet. The prerequisite for establishing an electronic marketplace is a secure payment. Several electronic protocols have been proposed to implement different kinds of payment: credit card payments, micropayments, and digital e-cash. Cryptographically, the most challenging task is the design of digital e-cashes. for every payment system mentioned above we have the requirement that the payment token has to be unforgeable. In 1982, D. Chaum [4] presented the notion of blind signatures that offer the possibility to design electronic e-cashes. The bank signs a set of data chosen by the user which guarantees both the unforgeability of the e-cashes and their anonymity, since the bank does not get any information about data it signed. But blind signatures solve only half of the problem: since

digital data can be copied, a user can spend a valid e-cash several times (double-spending) if the deposit of e-cashes is not done on-line. To validate each e-cash on-line means that the vendor has to contact the bank in every purchase. From the efficiency's point of view this is undesirable. Therefore, we restrict our attention to off-line systems, i.e., the vendor has to check the validity of e-cashes without contacting the bank. An e-cash is constructed in a way that allows its owner to spend it anonymously once, but reveals his identification if he spent it twice. From a theoretic point of view this solution is quite elegant. But in practice it is unsatisfactory. A way to prevent the user physically from copying her coins is to store essential parts of a coin in a tamper-resistant device called the observer.

2. AN E-CASH MODEL WITH TAMPER-RESISTANT DEVICE

An internet based anonymous off-line electronic eash scheme[2,5,7] with tamper-resistant device consists of three collections of probabilistic, polynomially-bounded parties[3], a bank B, users U_i , and shops S_j , and four main procedures: withdrawal, blind signature issuing, payment and deposit (Figure 1). Users and shops maintain separate account with the Internet Bank.

- When User (U_i) needs e-cash, then Bank issues e-cash from user's account in his (user's) tamper-resistant device T_i over an authenticated channel.
- When user(U_i) wants to spend this e-cash, it is validated by bank(B) by blind signature issuing protocol.
- U_i spends an e-cash by participating in a payment protocol with a shop S_j over an anonymous channel, and
- S_j performs a deposit protocol with the bank B, to deposit the user's e-cash into his account.



- (1) Withdrawal protocol (3) Payment protocol
 (2) Blind signature issuing protocol (4) Deposit protocol

Figure 1: Model of E-cash with tamper-resistant device

3. AN INTERNET BASED ANONYMOUS E-CASH SYSTEM

We shall now represent an anonymous off-line e-cash transaction system on the Internet.

3.1 The Bank's setup protocol

- All arithmetic is performed in a group G_q of prime order q chosen by bank (B). The bank generates independently at random four numbers $g_0, g_1, g_2, h \in G_q$ and a number $x \in Z_q$. The bank also determines a collision-free hash function $H(\cdot)$ such as to make the Schnorr signature scheme secure[1]. A public key that is issued by the bank to the user is a pair $(h'_i, a'_i) \in G_q * G_q$.

- The number x is the secret key of the bank, and the corresponding public key is the tuple $(g_0, g_1, g_2, h, G_q, H(\cdot))$. A certificate of the bank on the public key

(h'_i, a'_i) of the user is a triple (z'_i, c', r') such that $c' = H(h'_i, a'_i, z'_i, g_0^{r'} h^{-c'}, (h'_i)^{r'} (z'_i)^{-c'})$.

-The secret key that corresponds to the public key (h'_i, a'_i) of the user is a pair $((\beta_1, \alpha_1), (\beta_2, \alpha_2))$, such that $h'_i = g_1^{\beta_1} g_2^{\alpha_1}$ and $\alpha'_i = g_1^{\beta_2} g_2^{\alpha_2}$.

3.2 The actions

The Internet bank will be denoted by B, the user by U_i , and the service provider by S_j . The computer of U_i is denoted by C_i , and his tamper-resistant device by T_i .

3.2.1 Account establishment protocol

U_i installs on his computer, a software program for performing the protocols.

When U_i opens an account with B, the following procedure takes place.

- C_i generates independently at random a secret key $x_{i2} \in Z_q$, and stores it. C_i sends $h_{i2} = g_1^{x_{i2}}$, to B, together with an appropriate verifiable description of the identity of U_i . It then generates independently at random a secret key $x_{i1} \in Z_q$ for U_i . B lists this number (h_{i2}) in its so-called account database, together with at least a balance variable that keeps track of the account of money that U_i has in its account with B, and the description of U_i 's identity.

- B then issues to U_i a tamper-resistant device T_i which has stored in non-volatile memory at least the following items: the numbers x_{i1} and g_1 , and a counter variable, from now on denoted by balce, that keeps track of the amount of money that is held by U_i .

- B makes $h_{i1} = g_1^{x_{i1}}$, known to U_i ; this is the public key of T_i . B then computes $h_i = h_{i1} h_{i2}$ (the joint public key of T_i and U_i and stores h_i in his account database along with its other information on U_i). The bank B does not

know the joint secret key, $(x_{i1} + x_{i2}) \text{ mod } q$, of T_i and U_i .

-Finally, B computes $(h_i g_2)^x$, which will henceforth be denoted by z_i known to U_i .

3.2.2 Withdrawal protocol

T_i is assumed to have in common with B a secret key k . This secret key, and a sequence number, seq , (which has been set to some initial value, such as zero), have been stored by B before issuing T_i to U_i . In addition, the description of a one-way function $f_1(\cdot)$ has been stored by B in T_i . B decreases the balance, balce', of U_i by amount. It then increases seq by one, and transfers $v \leftarrow f_1(k, seq, amount)$ to T_i by sending it to C_i . T_i receives v from C_i . It then computes $f_1(k, seq, amount)$, and compares it for equality with v . If equality holds, it increases seq by one, and balance by amount.

The withdrawal protocol appears as follows:

Tamper-resistant Device(T_i)	Bank(B)
Verify $v = f_1(k, seq, amount)$ then, $seq \leftarrow seq + 1$ $balce \leftarrow balce + amount$	$balce' \leftarrow balce' - amount$ $v \leftarrow f_1(k, seq, amount)$ $seq \leftarrow seq + 1$

Table 1: The withdrawal protocol

3.2.3 The Pre-processing of blind signature issuing protocol

Payment of an amount requires U_i to provide the service provider with a signature on the amount (and additional data). To prepare for the withdrawal of a blind signature on e-cash, T_i and C_i perform the following off-line processing.

- T_i generates independently at random a number $w_i \in_R Z_q$, and sends $a_i = g_1^{w_i}$ to C_i . T_i stores w_i for later use in the payment protocol.

- C_i generates independently at random a vector $(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5) \in Z_q^5$, such that $\alpha \neq 0 \pmod q$. It then computes $h_i' \leftarrow (h_1 g_2)^{\alpha_1}$, $a_i' = a_i^{\alpha_1} g_1^{\alpha_2} g_2^{\alpha_3}$, $z_i' \leftarrow z_i^{\alpha_1}$, $temp_1 \leftarrow h^{\alpha_4} g_0^{\alpha_5}$, $temp_2 \leftarrow (z_i')^{\alpha_4} (h_1 g_2)^{\alpha_1 \alpha_5}$.

3. C_i stores (h_i', a_i') and $(\alpha_1, \alpha_2, \alpha_3)$ and $temp_1, temp_2, \alpha_4$ and α_5 for the later use in the payment protocol.

3.2.3 The blind signature issuing protocol

The issuing of blind signature is done by means of the following on-line certificate issuing protocol between C_i and B .

The blind signature issuing appears as follows:

<i>Computer(Ci)</i>	<i>Bank(B)</i>
	$w \in Z_q$
	$a \leftarrow g_0^w$
	$b \leftarrow (h_1 g_2)^w$
$c' \leftarrow H(h_i', a_i', a, temp_1, b^{\alpha_1} temp_2)$	
$c \leftarrow c' + \alpha_4 \pmod q$	
	$\leftarrow (a, b) \quad b \leftarrow (h_1 g_2)^w$
	$\rightarrow (c)$
	$\leftarrow (r) \quad r \leftarrow cx + w \pmod q$

Table 2: The blind signature issuing protocol

3.2.4 The pre-processing of payment protocol

To pay to S_j an amount, T_i and C_i perform the following pre-processing.

- C_i determines the specification, denoted by *spec*, of the payment. This number is a concatenation, in a standardized format, of that is to be transferred, the time and date of transaction, and an identification number that is uniquely associated with S_j . Additional data fields may be included in variable *spec*. C_i then sends (h_i', a_i') and *spec* to T_i .

- T_i verifies that w_i is still in memory, and that balance exceeds amount (T_i can read this value from *spec*). If this is the case, it computes $d = H(h_i', a_i', \text{spec})$ and $r_1 = dx_{i1} + w_i \pmod q$. It then decreases

balance by amount, erases w_i from memory, and sends r_1 to C_i .

- C_i computes $d = H(h_i', a_i', \text{spec})$, and verifies that $g_1^{r_1} h_{i1}^{-d} = a_i$. If this is the case, C_i computes $r_1' = \alpha_1(r_1 + dx_{i2}) + \alpha_2 \pmod q$, $r_2 \leftarrow d\alpha_1 + \alpha_3 \pmod q$.

The pre-processing of payment protocol appears as follows:

<i>User computer(Ci)</i>	<i>Tamper-resistant device(Ti)</i>
(h_i', a_i')	\rightarrow
	$d = H(h_i', a_i', \text{spec})$
	balce \rightarrow balce-amount
	$r_1 = dx_{i1} + w_i$
	$\leftarrow (r_1)$ erases w_i
	$d = H(h_i', a_i', \text{spec})$
	verify
	$g_1^{r_1} h_{i1}^{-d} = a_i$
	$r_1' \leftarrow \alpha_1(r_1 + dx_{i2}) + \alpha_2 \pmod q$
	$r_2 \leftarrow d\alpha_1 + \alpha_3 \pmod q$

Table 3: The preprocessing of payment protocol

3.2.5 The payment protocol

The actual payment is done by means of the following on-line payment protocol between C_i and S_j .

- C_i sends $(h_i', a_i'), (z_i', c', r'), (r_1', r_2)$ to S_j .

- S_j computes d in the same way as did C_i and T_i and accepts the transferred information if and only if $h_i' \neq 1, c' = H(h_i', a_i', z_i', g_0^{r_1} h^{-c'}, (h_i')^{r_1} (z_i')^{-c'})$ and $g_1^{r_1'} g_2^{r_2'} (h_i')^{-d} = a_i'$

The payment protocol appears as follows:

<i>Computer(Ci)</i>	<i>Service Provider(Sj)</i>
$-- (h_i', a_i'), (z_i', c', r'), (r_1', r_2) \rightarrow$	Check
	$d = H(h_i', a_i', \text{spec})$
	$c' = H(h_i', a_i', z_i', g_0^{r_1} h^{-c'}, (h_i')^{r_1} (z_i')^{-c'})$
	$g_1^{r_1'} g_2^{r_2'} (h_i')^{-d} = a_i'$

Table 4: The payment protocol

3.2.6 The deposit Protocol

- At a suitable time, preferably when network traffic is low, S_j sends the payment transcript, consisting of $(h_i', a_i'), (z_i', c', r'), (r_1', r_2)$ and *spec*, to B .

B verifies that *spec* has been formed correctly by S_j . If this is the case, it searches its so-called deposit database to find out if it has stored (h_i', a_i') before.

There are two possible situations:

1. (h'_i, a'_i) is not in the deposit database. B then computes $d=H(h'_i, a'_i, spec)$, and verifies the payment transcript by verifying that $h_i \neq 1, c'=H(h'_i, a'_i, z'_i, g_0^{r'} h^{c'}, (h_i)^{r'} (z_i)^{c'})$ and $g_1^{r_1'} g_2^{r_2'} (h_i)^{c'} = a'_i$. If these verifications hold, B stores $(h'_i, a'_i), (z'_i, c', r')$ and (r_1', r_2') in the deposit database, and credits the account of S_j by amount.

2. (h'_i, a'_i) is already in the deposit database. In that case a fraud has occurred. If *spec* of the already stored information is identical to that of the new payment transcript, then S_j is trying to deposit the same transcript twice.

Otherwise, B verifies the transcript as described in situation 1. If the verification holds (the payment transcript is valid), then the certified public key (h'_i, a'_i) must have been double-spent with overwhelming probability. Since, B now has at its disposal a pair (r_1', r_2') from the new transcript and a pair, say (r_1'', r_2'') , from the already deposited information, it can compute $(r_1' - r_1'') / (r_2' - r_2'') \bmod q$. B then searches its account database for joint public key $g_1^{(r_1' - r_1'') / (r_2' - r_2'')}$. Since, the identity of the corresponding account holder is known to B, appropriate legal actions can be taken. The number $(r_1' - r_1'') / (r_2' - r_2'') \bmod q$ serves as the proof of B that the traced user has compromised his tamper-resistant device and has double-spent the certified public key (h'_i, a'_i) .

4. DISCUSSIONS

The e-cash scheme with tamper-resistant device, the user's secret is shared between the user and his observer. The combined secret is a modular sum of the two shares, so one share of the secret reveals no information about the combined secret. Co-operation of the user and the tamper-resistant device is

necessary in order to create a valid response to a challenge during a payment transaction. It prevents the tamper resistant device from leaking any information about the user.

5. CONCLUSIONS

We presented electronic cash system which provides a physical defense against double-spending detection. To guarantee the prevention of double-spending the bank has to be sure that the tamper-resistant device cannot be tampered with by the users. The use of a tamper-resistant device is a kind of first line of defense. If the user cannot manipulate the device the tamper-resistant device can prevent double-spending. If the user succeeds in tampering the observer, the double-spending detection identifies the user afterwards.

REFERENCES

- [1] C.P.Schnorr. Efficient signature generation by smart cards. Journal of Cryptology, 4(3):161-174, 1991.
- [2] Israt Jahan, Mohammad Zahidur Rahman and Md. Golam Moazzem, Review of anonymous electronic payment system. Journal of Electronics and Computer Science, 2:25-39, 2003.
- [3] S.Brands. Untraceable off-line cash in wallets with observers. CRYPTO, pages 303-318, 1993.
- [4] D. Chaum. 'Blind signatures for untraceable payments'. Advances in cryptology-Proceedings of Crypto'82, Lecture Notes in Computer Science, Springer-Verlag, pp. 199-203.
- [5] D.Chaum and T.Pedersen. Wallet databases with observers. In proceedings of CRYPTO, pages 89-105, 1993.
- [6] 'System planned for shopping in the internet', Wall Street Journal, September 13, 1994, pp.B1. {about CyberCash, Inc.}
- [7] Y.S.Tsiounis. 'Efficient Electronic Cash: New Notions and Techniques. Ph.D thesis. Northern University, Massachusetts, June 1997.