

Cyphermint™ Pay Cash System

Ildar M. Khamitov

Alkorsoft & Bank Tavrichesky
khamitov@paycash.ru

A preliminary description of the Cyphermint™ Pay Cash System is presented. The motivation of the specific choice of a cryptographic scheme underlying the system is given. The advantages and drawbacks of the system and its analogs are discussed.

Payment systems

At the present time there are no Internet payment methods that could fully meet all requirements of the increasing tide of electronic commerce. Nevertheless, the majority of payments on the Internet are carried out by means of credit cards. The credit card payments have a number of disadvantages, including, in particular, low level of protection and security, complete absence of privacy, narrow range of possible payments, strict division of customers into the buyers and sellers, relative slowness of payments. Besides, credit cards are not in widespread use in some countries, e. g. in Russia. In the near future these disadvantages will hardly be overcome because of the inertia caused by the existing infrastructure and the huge established base of conservative customers (moreover, modernization of the existing card infrastructure requires considerable investments and leads to the loss of the investments made earlier). All these indicate that there is a need for alternative Internet payment systems. The Tavrichesky Bank in co-operation with the Alkorsoft Company have developed their own Internet payment system called the Cyphermint™ Pay Cash System. It is convenient to give a preliminary description of this system, having premised it with the motives that have led us to the model embodied in the Cyphermint™ Pay Cash System.

A perspective Internet payment system that could occupy a niche in the Internet payment business not occupied by the card systems and that, under certain circumstances, could compete with credit card systems, should provide:

- high level of security, including protection against bank,
- high level of anonymity of the parties,
- wide range of payments, including payments of several cents,
- low cost of transaction,
- easy integration into various trading systems,
- equality of buyers and sellers which encourages new sellers and facilitates repayments,
- fast operation which stimulates spontaneous buyers (any operation in a few seconds),
- scalability.

The concept of payment system security implies that the system incorporates some dispute resolution procedures. These procedures should enable an impartial third party to fairly resolve

the dispute, basing its decision on non-secret data that have been submitted by each party involved in the dispute.

First of all, we rejected those models of payment systems in which protection of a bank essentially depends on the “honesty” of customer’s software and hardware. It is obvious that in such systems the customer’s software must operate on a specialized tamperproof equipment protected from any external intrusions be it physical or logical (program), e.g. on a special chip card computer. It seems to us that protection of bank secrets by trusted devices accessible for malicious examination outside the bank is hardly reliable, especially in comparison with strength and reliability of modern cryptographic methods. In particular, the modern chip cards are not able to effectively protect the bank secrets, as depressing history of successful attacks on such cards performed during the last few years in the academic environment shows. If one such trusted device leaks its secrets to criminals, the consequences for the whole system may be disastrous. An extreme example of payment system relying on the “honesty” of customer’s software and hardware is the Mondex card system.

The payment systems not relying on the “honesty” of customer’s software and hardware can be classified into two groups. The first group includes the hardware systems in which customer’s software operates on a specialized computer intended for use only within the framework of the payment system. At present the majority of hardware systems have simplified customer’s software with scanty functionality since it is oriented to low-efficient computers of chip cards. The second group includes the software only payment systems in which customer’s software is just one of the applications that the user runs on a general-purpose computer. In this case customer’s software can be sufficiently complex and multifunctional because the software developers are not too restricted by hardware constraints. Evidently, there is no fundamental difference between these two groups of payment systems though in practical applications some systems of these groups may have advantages over others. For example, the hardware systems are free from risks relating to computer viruses and “Trojan horses”.

We have decided to develop a software payment system because such a system can be easily modified by adding new services to it or by adapting it to ever-changing demands of the market. Furthermore, any Internet surfer always has at hand a “decent” computer on which he can run the relevant user’s software (possibly, this is not quite true for the WebTV users). A high rate of computer miniaturization and as well as increase in their productivity give promise that in the near future the difference between hardware and software systems will disappear since it will become possible to run any sophisticated software on a specialized computer of match box size. It should be noted that hybrid systems in which one part of operations are carried out by a specialized computer and the other part by a general-purpose computer, are also possible. The following discussion is restricted to the software only systems.

One method for providing customer’s anonymity consists in opening an anonymous account associated with some public key. In general, there is nothing the bank needs to know about the customer except his public key, which is used to verify digital signatures on the customer’s orders. Even though the bank cannot provide some services for an anonymous customer, e. g. lend money without security, an anonymous customer is very convenient for the bank since there is no need to spend resources on checking the customer’s background.

Therefore, overhead expenses on an anonymous customer are much lower than that on a non-anonymous one (at least, until digital identities come into wide use). We believe that in order to be cheap a mass payment system should admit presence of anonymous customers. In the Cyphermint™ Pay Cash System anonymous accounts are admitted.

The anonymity of a payer can be ensured if the payer effects payment by transferring bearer digital money certificates to the payee. The payer receives these certificates from a bank beforehand, using a blind signature technology. The blind signature technology allows to break the link between the certificate and the account from which money was withdrawn so that the bank is unable to recognize in which withdrawal session the given certificate was signed. Thus, the payments turns out to be untraceable. However there is a specific double-spending problem in this payment method since digital certificates are easily duplicated. There are two approaches to the solution of this problem, and the payment systems are classified as on-line and off-line ones according to the adopted approach.

An on-line payment system is the one in which the moment of receipt of payment by the payee is the moment of successful authorization of payment by the bank, i.e., the moment of recognizing the presented certificates by the bank. Thus, in the on-line payment system the payee has to contact the bank to get approval for each payment.

An off-line payment system is the one in which the moment of receipt of payment by the payee is the moment of successful verification of the certificates provided by the payer as payment, this verification being performed by the payee locally. This means that the payee, e. g. a seller, need not contact the bank at the moment of payment — he can just present all the received certificates to the bank, e. g. at the end of the working day. In this case seller's operation is less dependent on the bank's Internet accessibility and the bank saves on the time of establishing numerous repetitive Internet connections. The bank is obliged to credit the seller's account with the amount corresponding to the valid digital certificates. The bank must identify the reused certificates and prosecute double-spenders without participation of law-abiding customers. There is a number of clever techniques of blind issuance of digital money certificates and their subsequent transfer in the payment transaction, such that the identity of the payer is revealed in case the payer reuses his money certificates (see, e. g., [1–3]).

In the systems of both types the bank should maintain the list of used certificates against which all the certificates presented by the payees (sellers) should be checked in order to detect double spending. In comparison with on-line systems, off-line systems are noticeably more complicated, require more resources, and their protocols are more interactive. It is quite possible that when the off-line software systems will at last ripen for practical application, there will be no free place in the market of Internet payment systems. This means that the time of mass off-line software payment systems may never come. However the main reason for which we have chosen to develop the on-line system is that in the off-line system the customer cannot be anonymous when receiving money certificates from the bank (no anonymous accounts).

The Chaum coin system

As a starting point in the Cyphermint™ Pay Cash System construction we used the on-line payment system suggested by D. Chaum [4]. In this system the money certificate (the coin of denomination i) is defined by the following data:

$$\text{Coin}(i, X) = \{i, X, g_i^{-1}(f(X))\},$$

where X is the random serial number of a coin, chosen by a customer, X belongs to a large set $M' \subset M = (\mathbf{Z}/m\mathbf{Z})^*$, m is a composite number whose factorization is known only to the bank, $f: M' \rightarrow M$ is an easy-to-calculate mapping publicly known and hard-to-invert for all parties of the payment system except, possibly, for the bank (another variant: the image of f has a very special form), $g_i(x) = x^{E_i}: M \rightarrow M$ are publicly known mappings with appropriate exponents. The mapping g_i^{-1} is the RSA-signature of the bank corresponding to the coin denomination i . If several currencies are used in the system, then its own set of functions g_i (we drop the currency label) should correspond to each currency. The set M (the number m) and mapping f can also depend on the denomination and currency.

The bank is able to mint such coins for the customer blindly. The buyer effects a payment by transferring a set of coins to a seller, the sum of denominations of these coins coinciding with the amount of the payment. The seller sends the coins to the bank for authorization. The bank makes sure that the submitted coins are not on the list of used coins, then puts them into this list, credits the seller's account with the amount of payment and informs the seller of the success. The payments in this system are unconditionally untraceable. Now we list some drawbacks of the Chaum system.

From the theoretical point of view a substantial disadvantage of the Chaum system is that the payer and the bank have to trust each other. The bank can misappropriate the coin presented by the payer, claiming that it has already been spent earlier. On the other hand, a double-spender can accuse the bank of attempting to steal the coin that was allegedly never used before. The payee also has to be trusted if the coins are transferred to him openly. It should be noted that this disadvantage is a fundamental property of the bearer certificates rather than a specific property of the Chaum coins. The bearer certificates do not carry any secret of the bearer by means of which he could prove his rights to the certificate. Thus, in the Chaum system the conflicts that cannot be resolved by means of the system itself, are possible. So there must be some external dispute handling that can hardly be automatic. This increases the cost of running the system because special organizational measures (insurance funds, black lists, etc.) to handle such conflicts should be taken.

The main application of a payment system is electronic commerce. To resolve disputes within the framework of a trade system, any payment transaction must be linked to the corresponding merchandise transaction so that the payer would be able to prove the fact of payment as well as its purpose. Insofar as the Chaum system provides no internal means of integration with a trading system, in addition to the Wallet (payment system client) the payer

must also have a Buyer (trading system client) specific for this trading system that will tie the payment transaction to the merchandise transaction.

Sooner or later the list of used coins in the Chaum coin system will become too large to go in the storage allocated for it. Furthermore, the time of searching coins in this list increases as the list increases, even though logarithmically. Therefore, to keep the list within acceptable limits, the bank must limit the validity period of coins. In this case the used coins whose validity period has elapsed, may be safely removed from the list. The expired and not used coins may be redeemed in some off-line procedure. A too short validity period of coins does not make the payment system more attractive for consumers. It should be noted that the wider is the range and the finer is the granularity of possible payments the faster is the growth of the used coins list because to ensure a wide range and fine granularity of payments it is necessary to have many denominations of coins. As a consequence, the average number of coins in one payment increases. The increase in average number of coins in one payment proportionally increases the search time in the list of used coins. The continuous progress of computer technology gradually reduces the gravity of the large used coins list problem. Besides, Chaum [5] has proposed an ingenious method for blind return of change by the bank that makes it possible to use only one coin per payment.

Modified Chaum coin system

Let us modify the Chaum coin system as follows. For each planned coin, the customer now generates a random pair $\{S, P\}$ of private and public keys within the framework of a certain signature system, for example RSA. Let $\text{Sign}_P(\cdot)$ and $\text{Verify}_P(\cdot, \cdot)$ be the signing and the verifying function, respectively, so that $\text{Verify}_P(X, Y) = \text{True}$ if and only if $Y = \text{Sign}_P(X)$. The coin that the customer generates in cooperation with the bank consists of the following data:

$$\text{Coin}(i, P) = \{i, P, g_i^{-1}(f(P))\},$$

i.e., in this modification a random public key is used as the coin's serial number. Now, however, the customer effects payment by transferring the following extended coin:

$$\{\text{Order}, Y, \text{Coin}(i, P)\}, \quad (1)$$

where $Y = \text{Sign}_P(\text{Order})$, and Order is a unique description of the payment, which can, in particular, contain the number of the account in which it is intended to deposit the money. Particularities of the payment description are not essential for general consideration. The bank will authorize the payment only if the coin with a given P is not on the list of used coins and the following relation is fulfilled:

$$\text{Verify}_P(\text{Order}, Y) = \text{True}. \quad (2)$$

If the authorization is successful, the bank enters coin (1) on the list of used coins, deposits the amount represented by the coin in the seller's account and sends the signed receipts referencing Order to the seller and payer.

With the modification described above there is no need for the bank and the payer to trust each other. Indeed, the bank cannot misappropriate the coin because it is unable to produce the coin extension data satisfying relation (2). On the other hand, the bank defends itself from being accused of misappropriation of the coin by demonstrating the coin extension data satisfying relation (2).

This modification also allows easy integration of the payment system with practically any trade system: it is sufficient to include the hash of the contract describing the terms of the deal into the Order. In this case the receipt bearing the bank signature will link the payment with the contract. The contract itself remains unknown to the bank.

Note also that from the theoretical point of view the use of a public key as the coin's serial number strengthens security of the coin. For example, if a counterfeiter manages to invert the function f , in order to spend the brand-new coin he still needs to resolve a hard problem of finding the private key corresponding to the coin's public key.

In the process of payment a payer has to sign the Order as many times as many coins are included in the payment. Since the payment can include tens of coins and the signing process is rather a time-consuming operation, the scheme described above can be unacceptably slow for the existing personal computers. The situation can be noticeably improved by means of Chaum's method of blind change return. In this case a small number of coins, for example only one coin, may participate in each payment.

Cyphermint™ Pay Cash System

In the Cyphermint™ Pay Cash System a customer pays using data that are called the payment book and have the following structure:

$$\text{PayBook}(N, P) = \{N, P, g^{-N}(f(P))\},$$

where P , f , and g have the same meaning as above, and $g^{-N}(X) = g^{-1}(g^{-1}(\dots g^{-1}(X)\dots))$. A nonnegative integer N (the disposition of the book) determines paying capacity of the book. A necessary condition for the triple $\{n, P, A\}$ to be a payment book is the following equality:

$$f(P) = g^n(A). \tag{3}$$

This condition can be readily tested by any participant of the system, and in particular by the owner of the book. Thus, the payment book differs from the Chaum coin in that a random public key is used as a random serial number, and the value is encoded not with the help of the "denomination" but with the help of the powers of the signing mapping.

Obviously, any customer can create an empty payment book, that is a book with zero disposition $\text{PayBook}(0, P) = \{0, P, f(P)\}$. Moreover, having at his disposal the payment book $\text{PayBook}(N, P)$, the customer can construct all books with the same P but with lower dispositions:

$$\text{PayBook}(0, P), \text{PayBook}(1, P), \dots, \text{PayBook}(N - 1, P).$$

On the other hand, this is the only way for the customer to create non-empty payment books without the bank cooperation (reducing the disposition of the book known to him). To create a non-empty payment book from scratch without the bank cooperation is not easier than to create a coin in the initial Chaum system. Indeed, if a customer has managed to make the payment book $\text{PayBook}(N, P) = \{N, P, A\}$ with the disposition $N > 0$, then in the corresponding Chaum system he has also managed to make the coin $\text{Coin}(i, P) = \{i, P, g^{N-1}(A)\}$, where i is the denomination corresponding to the signature g^{-1} . Increasing the disposition of a nonempty payment book $\{N, P, A\}$ that is not obtained by reducing the disposition of the book known to customer, is equivalent to even harder problem of obtaining the bank signature $g^{-1}(A)$ on the data $A = g^{-N}(f(P))$ which are fixed and have random character.

In cooperation with the bank the customer can replenish the payment book with value, i.e., can increase its disposition; he can increase the disposition of the newly created payment book with a zero disposition as well as the disposition of the payment book replenished previously. To replenish the payment book $\text{PayBook}(N_1, P)$ with the sum N_2 the customer transfers the blinded data B to the bank. One variant of blinding is a simple modification of Chaum's method and has the following form:

$$B = g^{N_3}(r) g^{-N_1}(f(P)),$$

where r is a random element of the set M and $N_3 \geq N_2$. In exchange for the sum N_2 the bank returns the signature $C = g^{-N_2}(B)$ to the customer. The customer extracts the required part of the book $\text{PayBook}(N_1 + N_2, P)$ from the signature C as follows:

$$C/g^{N_3-N_2}(r) = g^{-N_1-N_2}(f(P)).$$

Other variants of the blinded RSA-signature can be used in this scheme, e. g., another Chaum's method [6]. Note that the bank does not obtain any information about the disposition of the book being replenished by the customer.

In the unblinded form the payment books are presented to the bank during payment transaction. The bank maintains the list of all public keys P corresponding to valid payment books ever presented to the bank, i.e., the books satisfying relation (3). Together with each key P the bank also keeps some other data, which we refer to as a virtual account. In particular, the virtual account includes the exposition (of the virtual account) that is equal to the largest disposition of the corresponding payment book that has been exposed to the bank in the payment transactions. Furthermore, the virtual account stores the sum of all expenses made by the corresponding book. The list of virtual accounts is an analog of the list of used coins in the Chaum coin system.

Now let us consider the payment transaction. Assume that the payer has at his disposal the payment book $\text{PayBook}(N, P)$. In general, the payer sends the following data to the payee as a payment:

$$\{\text{Order}, Y, \text{PayBook}(n, P)\},$$

where $n \leq N$ and Y and Order are described above. The payee forwards these data to the bank, which, in turn, authorizes the payment as follows.

1. The bank verifies the necessary validity condition (3) for the payment book presented. If it is not satisfied or if $n = 0$, then the bank refuses to authorize the payment.
2. The bank searches for the virtual account corresponding to P . If the account is not found, i.e., if the payment book with the public key P has never been presented, then the bank creates an account and sets the exposition of the new virtual account equal to n , and the sum of the expenses equal to zero.
3. If the exposition of the virtual account is less than n , then the bank sets the exposition equal to n .
4. If equation (2) is not satisfied, then the bank refuses to authorize the payment.
5. If the expenses of virtual account together with the payment amount do not exceed n , then the bank increases the expenses of the virtual account by the payment amount and authorizes the payment; otherwise, the bank refuses to authorize the payment.

If the authorization happens to be successful, the bank credits payee's account with an appropriate sum and sends the signed receipts referencing Order to the payer and payee.

If the payer is sure that the exposition of the virtual account corresponding to the payment book $\text{PayBook}(N, P)$ is sufficient to effect the projected payment, e. g., if in the previous receipt the bank indicated the exposition of the virtual account, then the payer can use the following reduced data as the payment:

$\{\text{Order}, Y, P\}$.

In this case the authorization goes with obvious modifications.

Note that the payment sum does not need to be a multiple of the withdrawal unit $g^{-1}(f(P))$, but may actually take arbitrary values. For instance, the customer may be constrained to withdraw from his account to the payment book only a whole number of cents and nevertheless be able to pay 2.718 cents.

The scheme described above can be very roughly characterized as a system of anonymous accounts with the possibility of untraceable transfer of money (obligations) from one account to another. Since the virtual account, as a rule, services many payments, the list of virtual accounts grows much slower than the corresponding list of used coins in the Chaum system.

It is obvious that all the payments made by means of the same payment book can be easily linked to each other by the bank via the corresponding virtual account. This puts customer's anonymity at some risk: if one of his payments will be attributed by means that are exterior to the payment system, then all his payment history associated with the same payment book can be revealed. To diminish this risk the customer can discontinue his payment history by discontinuing the use of old payment books and switching to the new ones. Besides, when the book expires, its payment history ends. It depends on the suspiciousness of a customer and on the cost of different operations charged by the bank how frequently the customer should discontinue his payment history and how many payment histories he should have simultaneously. The higher cost of having a new payment book (virtual account) and/or money

withdrawal to a payment book stimulates an average customer to use the same payment book for a greater number of payments, not preventing, however, a suspicious customer from having a possibility to restrict the number of payments made by means of one book. Furthermore, the fact that the bank charges a fee for a new virtual account protects the list of virtual accounts from overflow and gives the bank an additional source of revenue.

Now let us discuss the possibility to link a payment book (virtual account) with the account from which money has been transferred to this book. The blind signature technology used to replenish the payment book ensures only that the account cannot be linked with the payment book in the withdrawal session. However the bank can link the book with the account indirectly by other means, e. g., by the Internet address of their owner. The bank can also try to link the book with the account by analyzing the total amount of money transferred to, and spent by, the payment book. However this indirect method of linking is hampered to a great extent because the same payment book can be replenished with money from different accounts and also because in a process of payment the payer does not expose the whole sum N of the payment book $\text{PayBook}(N, P)$ to the bank at once. Furthermore, a slightly modified version of Chaum's method of blind change return can be easily built into the Cyphermint™ Pay Cash System. This method can be used for restructuring the amounts kept on payment books and, in particular, for transferring the money that remains on the book intended for destruction, to another payment book. This will still further reduce the possibility of linking the book with the account.

Thus, the approach adopted in the Cyphermint™ Pay Cash System ensures reasonable level of payment untraceability.

Implementation

When developing a real system, an overwhelming portion of efforts are spent not on the incarnation of the theoretical scheme itself but on proper data structuring and processing and on the handling of various errors and contingencies. In particular, of special importance is the problem of system recovery from hardware malfunctions and human errors, because one error can affect several parties acting in their own interests. The difficulties arising when solving such problems can make an excellent theoretical scheme hardly practicable. In the Cyphermint™ Pay Cash System such difficulties have been overcome, which is demonstrated by the working model that is accessible to public testing since January 15, 1998. In February, 1999 the pilot project in which real money circulates was started on the basis of this model. The commercial version of the system is planned to be released in 2000.

- [1] David Chaum, Amos Fiat, and Moni Naor, Untraceable Electronic Cash, *Advances in Cryptology CRYPTO ' 88*, Springer-Verlag, pp. 319-327.
- [2] Stefan Brands, Untraceable Off-Line Cash in Wallets with Observers, *Advances in Cryptology CRYPTO ' 93*, Springer-Verlag, pp. 302-318.
- [3] Niels Ferguson, Single Term Off-Line Coins, *Advances in Cryptology - EUROCRYPT ' 93*, Springer-Verlag, pp. 318-328.

- [4] David Chaum, Security without Identification: Transaction Systems to Make Big Brother Obsolete, ACM 28, no. 10, pp. 1030-1044 (Oct 1985).
- [5] David Chaum, Returned Value Blind Signature Systems, U.S. Patent 4 949 380, 14 Aug 1990.
- [6] David Chaum, Blind Unanticipated Signature Systems, U.S. Patent 4 759 064, 19 Jul 1988.