Flow Control: A New Approach for Anonymity Control in Electronic Cash Systems

Tomas Sander and Amnon Ta–Shma

International Computer Science Institute
1947 Center Street, Berkeley, CA 94704, USA
{sander,amnon}@icsi.berkeley.edu

Abstract. Anonymity features of electronic payment systems are important for protecting privacy in an electronic world. However, complete anonymity prevents monitoring financial transactions and following the money trail, which are important tools for fighting serious crimes. To solve these type of problems several "escrowed cash" systems, that allow a "Trustee" to trace electronic money, were suggested. In this paper we suggest a completely different approach to anonymity control based on the fact that law enforcement is mainly concerned with large anonymous electronic payments. We describe a payment system that effectively limits the amount of money a user can spend anonymously in a given time frame. To achieve this we describe a technique to make electronic money strongly non-transferable. Our payment system protects the privacy of the honest user who plays by the rules, while introducing significant hurdles for several criminal abuses of the system.

1 Introduction

Anonymous electronic payment systems are regarded as essential for the protection of the privacy of consumers participating in electronic transactions. On the other hand anonymity features have the potential of being abused for criminal activities. These include tax evasion, money laundering and blackmailing scenarios. To address these concerns several "escrowed cash" systems that allow to revoke the anonymity have been proposed, cf. [8,3,4,13,17]. Although these schemes differ in their individual features, they share the common characteristic of the existence of Trustees, i.e., escrow agents that are able to revoke the anonymity of each individual payment (cf. [8] for an overview and further references). The supported revocation features are usually coin tracing (the ability to associate a deposited coin with a withdrawal) and owner tracing which allows to identify the owner of a spent coin.

In this paper we describe a different approach to help to prevent criminals from abusing electronic cash systems which avoid the necessity to make each individual payment potentially traceable. It is based on the observation that honest users and abusive users of electronic cash usually have different interests in using anonymous money. Criminals are usually interested in huge anonymous transactions (e.g., laundering one million dollars that stem from drug deals)

M. Franklin (Ed.): FC'99, LNCS 1648, pp. 46–61, 1999.

while the common user usually does not even have these amounts of money, and is usually interested in small anonymous transactions (e.g. buying a political magazine or an adult video dealing with a certain fetish he does not want to be revealed - neither to his wife nor to the government). For high value transactions like buying a new car or a house, anonymity is not really an issue for most honest users.

This discrepancy is also reflected in U.S. policy: The Bank Secrecy Act, passed in 1970, instructs banks to report any cash transactions exceeding $10,000 to the IRS. These regulations practically eliminate the anonymity of huge cash transactions. Stanley E. Morris, at that time director of the U.S. Department of Treasury's Financial Crime Investigation Network stated that his organization is mainly interested in *large* payments [16]. Furthermore Morris suggests in [16] to solve some of the problems associated with electronic cash transactions by putting limits on the amounts people can spend with cyber payment and smart card based systems.

Thus in contrast to electronic cash systems with revokable anonymity that limit the overall privacy users of electronic payment systems enjoy, by making each individual transaction potentially traceable, we limit the privacy of users by allowing them to have unconditionally anonymous transactions *up to a certain amount*, i.e., we limit the money flow each user can create in a certain time frame. Via regulations one may enforce that to obtain anonymous electronic cash a user needs to open an electronic cash account, and when doing so he has to identify himself (with a photo-id, say). Furthermore, the bank checks that each person holds only one such account, and each month each person has the opportunity to buy at most, say, $ 10,000 in anonymous electronic cash from the bank.

However, there are at least two problems with this naive approach. First a user may accumulate large amounts of electronic cash throughout time. Second, and more serious, electronic money can be traded. Because the money is absolutely anonymous it can change hands without leaving any traces and anonymous money could be accumulated in large sums.

To avoid money–trading we introduce in Section 2 the requirement of *non–transferability* for a payment system and argue that it is an important feature to have for electronic cash system to limit criminal abuses. In Section 4 we describe an on–line system that is non–transferable, amount–limited and guarantees unconditional payer anonymity. It builds upon the provable secure system presented by Damgard [6] (and later corrected in [20]). This basic system has the disadvantage of forcing users to exchange their unused expired coins every month. This gives the bank a good estimate of the amount a user has spent per month and may be regarded as a privacy violation. In Section 5 we describe a coupon based variant of this protocol that achieves accumulation control and does not have this disadvantage. Both systems are described in the general computation model and make use of secure multi party computations, so they are polynomial–time, but not efficient in a practical sense. In Section 6 we sketch how an efficient, off-line, amount-limited payment system can be constructed from Brands' scheme [2,1].Finally in Section 7 we argue how this helps to defend against common attacks and abuses of electronic payment systems.

2 Non–transferability and Amount–Limitedness

2.1 On the importance of non–transferability.

A considerable amount of work has been done to design payment systems that are transferable (i.e. where a coin received during a payment can be further spent by the receiver without intermediation of a bank, cf e.g. [18,19,5,7]), so that electronic cash enjoys some of the conveniences of physical cash. However physical cash is only conveniently transferable in small amounts (e.g. also by mail) or in large amounts between users that are physically close to each other. It was stated in a recent report on research performed by RAND for FinCEN [15] that "the physical movement of large quantities of cash is the money launderer's biggest problem". This hurdle to criminal abuse is potentially removed by cyberpayment systems ([15]): "The international dimension of these systems, and the fact that value transfers may take place with rapidity and with a degree of anonymity that impedes oversight by governmental authorities, is clearly a serious concern." Thus transferable electronic money seems not to be a good idea from the perspective of crime prevention.

Developers of (anonymity controlled) electronic payment systems have not yet paid explicit attention to the problems transferability presents. To illustrate this problem consider a non-anonymous system that is also fully transferable (each coin carries a serial number that is recorded during withdrawal and deposit). At first sight the authorities can have full information about the system, but is it really so?

In a fully-transferable system, after a few hand changes of the cash token among different users and shops it is likely to be *practically* anonymous, and the information of who withdrew the coin in the first place is probably going to be irrelevant as the transaction chain of the token during its life span will be almost impossible to reconstruct. Money changing organizations may exploit transferability intentionally to create huge amounts of anonymous money that then can be used for anonymous (criminal) transactions. Thus, a revokable but also easily transferable escrowed cash system might turn out to be too weak. Recall that the off–line electronic cash systems that had been suggested so far are only non–transferable up to the degree that payers and receivers are not willing to take double spending related risks.

2.2 How to build a non-transferable system

A coin usually passes three stages in its life: a user (whose identity is known to the bank) withdraws a coin. We call him the *owner* of the coin. In the payment phase a merchant is *paid* with a coin. In the deposit phase a merchant deposits a coin into the bank. We say a payment scheme is *non-transferable* if only the person who withdrew a coin is able to use it for payment. Thus, dollar bills are clearly transferable, checks are usually transferable but can also be made non-transferable, and credit card payments are normally non–transferable.

Can a non–transferable system be built? Suppose Alice withdraws a coin c. If Alice gives the coin c to Bob, along with all the information needed for paying with it, then Bob can use the coin c himself, and the money is made transferable. In general, as coins are bit strings that can be easily copied and transmitted it seems impossible to achieve non-transferability by purely technical means, unless tamper–resistant hardware is used. Thus instead of making it technically impossible for a user to copy and transfer coins we want to design a system in which a (rational) user does *not want* to transfer his coins.

We suggest the following ways of achieving that: we assume each user holds *one* secret that he does not want to (or can not) give away. We call this a "non–transferability secret". When a user withdraws electronic money his non–transferability secret is imprinted in the coin, and the payment protocol assures that only a person knowing this secret is able to *pay* with the coin. Thus, giving away the knowledge how to spend the coin gives away the non–transferability secret which he does not want.

One way to provide users with secrets they do not want to give away is to have a public key infrastructure in place such that liabilities are assigned to the secret key of a digital signature scheme. For example if a receiver of the secret key could completely impersonate the owner in a digital world, e.g. for receiving loans, signing contracts, etc. This secret key seems to be well suited to serve as a non–transferability secret which its owner may not *want* to give away. Examples for physical protection measures are: the secret is generated and stored on a tamper–resistant device (like a smart card) in a way that the secret is kept hidden from its owner. Even more, the usage of the device could require a biometric authentication of the holder as comparing the fingerprint of the card holder to the registered fingerprint.

It is clear that no system is absolutely non-transferable. E.g., a criminal can kidnap his victim, "cut off his finger" and use the stolen smart card to make a transaction. However, as each user is limited in the amount of coins he can spend, to get a significant amount of money a criminal will have to do that procedure for many persons. While everything is theoretically possible, we consider it unlikely.

A similar idea using non-transferability secrets has been suggested by Dwork, Lotspiech and Naor in their construction of "digital signets" [9] in the context of copyright protection for digital goods. The concept of non–transferable secrets is probably useful in many other situations where one does not want the user "to give something away", and for anonymity controlled payment systems in particular. It may be desirable to add this feature to other existing payment systems to make their anonymity control features stronger. We expect that many existing payment schemes can be technically modified to achieve non–transferability. In this paper we show how to add these features to Damgard's on-line cash system, and to Brands' efficient off-line system.

Jakobsson and M'Raihi recently described an on-line payment system which is account based [12]: users initiate fund transfers from their bank account to another user's bank account. Anonymity is achieved by a mix–network involving banks and an Ombudsman by which fund transfers are processed. By its account based nature, no "value" that could be transferred ever leaves the bank. Thus, although the system was not designed to be non-transferable, the system has strong non–transferability features by its very nature. Amount–limitedness can be added easily to the system by restricting the amount of funds a user is allowed to transfer per time frame. However the anonymity of transactions is revokable by a quorum of banks (and Ombudsman), and the system is on-line.

2.3 Technical requirements

From now on we assume that the infrastructure provides each user with a non–transferability secret and formalize the technical requirements that a payment systems needs to have to make use of it:

Non–transferability :

- If a coin c was withdrawn by A and c is spend by C, then C knows the non–transferability secret S_A of A.
- Under no scenario of system events any coalition of polynomial time players can learn any information about a non–transferability secret S_A of a user not within that coalition.

Amount limitedness : During each time frame T_i each user U can spend at most b electronic coins that were withdrawn by him.

We first note that amount–limitedness is very easy to achieve by restricting the amount of coins a user can withdraw within a time frame. To avoid that users spend coins that have been withdrawn during earlier time frames coins may carry e.g. an expiration date and it could be enforced that coins are only used during the time frame in which they were withdrawn. Yet, as argued, this property is not of much use unless it is combined with the non–transferability property.

We now concentrate on the non–transferability definition. The first part of the definition requires that a person who spends a coin knows the non–transferability secret of the person who withdrew that coin. Thus, the non–transferability property is useful in the real-world only when users are unwilling (or unable or both) to surrender their non–transferability secrets.

The second part of the non–transferability definition requires that under no chain of system events a non–transferability key is revealed. This definition is more delicate than it first seems. Brands' off-line system, e.g., is a system where each user has a secret that normally remains statistically secure, and is only revealed when a user double spends. Brands' system does not achieve, as it is, the non–transferability requirement as double spending is a possible chain of events that results in the revelation of a non–transferability secret. This is clearly not appropriate as a disclosure of the non–transferable secret may cause serious damage for the user. Our off-line system achieves the above strict requirement. In particular, each user in our system has two secrets: one is the non–transferability secret (that never gets revealed) while the other is revealed whenever a user double spends.

3 System

The participants: users, merchants, a bank, a CA and the government.

Infrastructure: We assume there is a public-key infrastructure (PKI) in place s.t. each participant holds a public/private key pair (P_U, S_U), there is a reliable way to authenticate a user's public key P_U via a CA. We make the central:

NTA (Non-transferability Assumption) - We assume each user U has a *non-transferability* secret S_U s.t. most users U will not do any action that will reveal their non-transferability secret S_U to any other group of players.

Finally we assume that the bank's and CA's public keys are known to everybody.

Time: We assume that there are consecutive time frames denoted $T_1, T_2, \ldots$. (in our earlier examples the T_i's were consecutive months)

Amount–limit: We assume that there is a limit b, s.t. that each user is allowed to spend b electronic coins anonymously during a time frame T_i.

Computing Power: All participants are probabilistic polynomial time players.

Trust Model: Users and merchants trust the bank not to steal their money. The government (i.e. the party who is interested in controlling anonymity) trusts the CA to reliably identify persons, and the bank to perform the necessary checks (as described in the protocol) reliably during all transactions. The network is reliable and communication over it is anonymous.

System Events: We focus on the following system events: Opening an account, withdrawing money, paying money to a merchant and depositing money (or expired coins) at the bank.

We have the following requirements for our system:

Unforgeability: It is infeasible for any coalition of participants in the system excluding the bank to create an amount of payments accepted by the bank that exceeds the amount of withdrawn coins.

Non–transferability: is defined as in Subsection 2.3.

Amount limitedness: is defined as in Subsection 2.3.

Unconditional Payer Anonymity: A payer has unconditional anonymity, i.e. transcripts of withdrawals are statistically uncorrelated to transcripts of payments and deposits.

4 A Protocol for an On–Line Amount Limited Cash System

Our protocol is based on the system suggested by Damgard [6] with the correction suggested by [20]. At withdrawal time Alice receives a signature from the bank for a coin M by employing a secure computation protocol that encodes the non–transferable secret of Alice in M, and at spending time Alice presents M along with a proof that she knows a signature for it. We start with some necessary background:

Digital Signatures: A digital signature scheme for signing messages M by B consists of a (possibly randomized) polynomial time signature algorithm $\sigma(M, S_B)$ which produces a signature of M using the secret key S_B of B and a polynomial time verification algorithm $V(\sigma, M, P_B)$ which returns "true" iff σ is a valid signature for the message M w.r.t the public key P_B. For formal definitions see, e.g., [11]. A signature scheme is history independent if an honest signer can sign a message without knowing his previous signatures. Signature schemes that are existentially unforgeable (cf. [11]) and simultaneously history–independent exist under the random oracle hypothesis (cf. e.g. [21]), and under the general assumption that one–way permutations exist [14]. [1]

Secure computation: Two players, Alice and Bob, hold private inputs x and y respectively. If one way trapdoor permutations exist [26,10] then for any functionality (f_A, f_B) there is a multi-round two party protocol s.t. Alice learns $f_A(x, y)$ and Bob learns $f_B(x, y)$ with the following properties: Both players have computational confidence in the result , Alice has *perfect* privacy and Bob has *conditional* privacy . Furthermore the value $f_A(x, y)$ can be learned by Alice only as the last message of the protocol. There is an efficient simulator which can simulate the view of each of the players (cf. [10]), even in the presence of early abortions and malicious faults.

4.1 The protocol

Opening an account: During account opening a user Alice identifies herself to the bank (e.g. by a driver's license or with a certificate issued by a CA). The bank checks the authenticity of the user's public key. The bank checks that Alice does not have another electronic cash account by querying its database of registered users. The bank registers the user's identity together with the user's public key.

Withdrawal: Alice identifies herself to the bank. The bank checks that the amount of electronic coins withdrawn by Alice in the time frame T_i is smaller then the maximal amount b. Alice chooses a random string R. Alice and the bank engage themselves in a secure computation with perfect privacy for Alice. The public data are $P_A, P_B, Time$, Alice's private input is S_A, R and the Bank's private input is S_B. The outputs are obtained in the following way. First it is verified that S_A is a secret key matching P_A and if not the output FAILED is given to Alice and the bank, otherwise the bank gets COIN ISSUED and Alice's output is the bank's signature $\sigma\ (S_A \circ Time \circ R\ ,\ S_B)$.

After receiving the output Alice checks that this is indeed a valid signature for $M = S_A \circ Time \circ R$. The bank deducts the value of the electronic coin from her account and increases the number of coins withdrawn by Alice during the time frame T_i by 1.

[1] In [14] even a computationally *blind* secure signature scheme is described. However we do not need the blindness property for our protocol to work.

Spending: Alice wants to spend a coin $M = S_A \circ Time \circ R$ to a merchant C. Alice sends R to the merchant. Alice sends R and the merchant's identity P_C to the bank. Alice and the bank again engage themselves in a secure computation with perfect privacy for Alice. This time the public data are $P_C, R, Time$ and Alice's private input is $S_A, \sigma = \sigma(S_A \circ Time \circ R)$. If $V(\sigma, S_A \circ Time \circ R, P_B) = True$, i.e. if σ is a valid signature of $S_A \circ Time \circ R$ the Bank's output is VALID, otherwise NOT VALID. If the output is VALID, the bank checks that R has not been spent before, and records it in its database of spent coins. The bank credits C's account and sends C a notice that a payment for transaction R has occurred to his account.

Depositing expired coins: Alice wants to deposit her unused expired coins to her account. This is implemented in an analogous, obvious way. Alice identifies herself to the bank. It is checked, via a secure computation, that Alice knows the secret encrypted in the coin and that this secret matches the public key registered with her account.

Notice that the bank does not know what signatures were produced and therefore the t'th signature can not depend on the previous $t-1$ signatures, which forces us to use history-independent signature schemes.

Theorem 1. *The payment system described above achieves non-forgeability, unconditional payer anonymity, non-transferability and is amount limited.*

Proof. The proof for non–forgeability is standard and follows the arguments in [6,20]. We omit it here. Amount–limitedness is immediate.

Non-Transferability: Suppose Charlie is spending a coin. Then Charlie convinces the bank with a secure computation that he knows some M along with its signature σ. It is easy to see that the coin must have been withdrawn before. Let us assume it was obtained by Alice. At withdrawal time Alice had to posses $S_A, Time$ and R_A s.t. $M = S_A \circ Time \circ R_A$, and S_A matches Alice's public key P_A. Hence, in particular, Charlie who knows M also knows S_A the secret key of Alice.

Anonymity: We need to show that the distribution the bank sees at deposit time π_D is statistically independent of the distribution at withdrawal time π_W. What does the bank get at deposit time? By the privacy property of secure computation the bank only gets to know a VALID/NOT VALID answer (so all valid coins generate a VALID answer) and a value R which is statistically independent of π_W and all previous values seen by the bank. Thus π_D is statistically independent of π_W. The situation does not change even if some of the other players collaborate with the bank.

□

In the next section we describe a refinement of this basic protocol for non–transferable, amount–limited, electronic cash which allows for greater flexibility in the use of the system.

5 Anonymity Coupons

There are two disadvantages of the on–line amount limited cash system described in the last paragraph. First a user has to deposit and exchange his expired coins that he did not use during a time frame T_i. This gives the bank a good estimate of the amount a user has spent during T_i and may be regarded as a privacy violation. Second the amount limitedness property of this payment system limits its use as a universal payment system, as a user's total transactions using these electronic coins can not exceed the limit b during T_i.

Both disadvantages can be solved with the following variant of the described payment system that we sketch here briefly. To do this we introduce "anonymity coupons"[2]. Coupons look like coins: they are of the form $(S_A \circ Time \circ R)$, i.e., they encode Alice's non-transferability secret and the time frame $Time$, together with a signature T from the bank created with the bank's coupon signature scheme. However coupons do not carry any monetary value and a user Alice obtains $b = 10.000$ of them at the beginning of each month *for free*. Their sole function is to limit the amount a user can spend anonymously. Electronic coins (that do carry monetary value) can be withdrawn *without amount restrictions* for the user. Coins may carry an expiration date much longer then T_i. Coins, too, contain Alice's non–transferability secret. The spending protocol is modified as follows: During payment Alice has the choice to make an anonymous or a non–anonymous payment. In anonymous payments Alice pays both with a valid coin and a valid coupon (and for that she needs, as before, to know the common non–transferability secret imprinted in them). In non–anonymous payments Alice reveals her public identity, pays with a coin only and proves that the non–transferability secret imprinted in the coin matches her public key.

After having demonstrated the ideas and concepts of non–transferable, amount–limited electronic cash in the framework of general computation based payment systems we now turn to the description of a practical system.

6 An Efficient Off–Line System

We now modify Brands' system s.t. double-spending does not reveal the non-transferability secret of the user. In our system each user has one fixed identity (P_A, S_A) where S_A serves as the non-transferability secret and remains private even in the case of double spending, and one per transaction secret (u_1, u_2, s) that gets revealed in case of his double spending. This results in some changes to Brands' system which we now describe. We mention that our modification also makes the system overspending robust, i.e. even if a user double spent he can not be framed for double spendings he has not done.

Let m', g_1, d_1, d_2, d_3 be elements of a prime order subgroup of Z_p^*. We say that a value m' has a representation (a_1, a_2, a_3, a_4) with respect to a set of generators (g_1, d_1, d_2, d_3) if $m' = g_1^{a_1} d_1^{a_2} d_2^{a_3} d_4^{a_4}$. Brands suggested a *restrictive*

[2] see also [24], where unlinkable serial transactions were studied for some other possible applications of our techniques.

blind signature scheme such that after withdrawal Alice (and also any other participant) can know only one representation of m' and this representation has to take a specific form. In our case it takes the form (sS_A, su_1, su_2, s) where S_A is Alice's non–transferability secret, and u_1, u_2, s are random numbers of her choice. The blinded signature is received for $m' = m^s$ where m is the message sent to the bank. At spending time the payer has to reveal $a_3 + ca_4$ for a random challenge c, and to prove knowledge of $a_1 = sS_A$ and $a_4 = s$. ¿From this we can deduce non-transferability, anonymity and more.

In our system S_A always remains private, even in the case of double spending. The per transaction secret (u_1, u_2, s) gets revealed in case of double spending. Using s the bank can associate the double spent coin with the user who withdrew it. We use a variant of Brands' system that was suggested by Brands ([1] , pages 31-32, Method 1) and has the feature that double-spending reveals some parts of the information the user has about the representation of the coin, while keeping other parts secret. This results in a minor inefficiency by increasing the number of rounds at withdrawal and payment time.

6.1 The protocol

Bank's setup: p, q long enough primes. $p = cq + 1$ for some integer c. G_q is the subgroup of order q of Z_p^*. The bank picks random elements $g, g_1, d_1, d_2, d_3 \in G_q$, a secret $x \in G_q$ and computes $h = g^x$. The bank uniformly selects hash functions $\mathcal{H}, \mathcal{H}_0$ from a collection of collision intractable hash functions. The bank makes $p, q, g, g_1, d_1, d_2, d_3$ and its public key h public and keeps x secret. By convention $a \cdot b$ is computed modulo p, except for exponents where $g^l = g^{l \bmod q}$ that are computed modulo q as g will always be taken from G_q.

Account opening: Alice identifies herself along with a number $P_A (= g_1^{S_A})$, and proves to the bank that S_A is Alice's non–transferability secret. The bank records Alice's identity together with P_A.

Withdrawal: Alice identifies herself to the bank. Then she picks $u_1, u_2 \in_R Z_q$ and computes $T = d_1^{u_1} d_2^{u_2}$. She sends T to the Bank along with a proof of knowledge of a representation of T according to the generators (d_1, d_2) (Proofs of knowledge of a representation are performed as described in [1]). Then Alice obtains a "restrictively blind" signature [1] of $m = T \cdot P_A \cdot d_3$. Alice will end up with a Schnorr-type [23] signature on $m' = m^s$ where s is a secret random number chosen by Alice. We also require that $m' \neq 1$. The signature is $sig(m') = (z', a', b', r')$ s.t. $g^{r'} = h^{H(m', z', a', b')}$ and $(m')^{r'} = (z')^{H(m', z', a', b')} b'$. See Figure 1 . We note that Alice knows a representation of m' according to the generators g_1, d_1, d_2, d_3, namely $(sS_A, u_1 s, u_2 s, s)$. The bank records that user P_A obtained a blind signature on m and deducts the corresponding amount from her account.

Payment: During payment Alice supplies the merchant with the coin $(m', sig(m'))$. She receives a challenge $c \in Z_q$ and answers with $a_3 + ca_4$, where (a_1, a_2, a_3, a_4) is her representation of m' according to the generator–base (g_1, d_1, d_2, d_3). The protocol for this is identical to ([1] , pages 31-2, method 1). See Figure 2.

Fig. 1. Withdrawal

Alice		Bank
$u_1, u_2 \in_R Z_q$, $T \leftarrow d_1^{u_1} \cdot d_2^{u_2}$	$\xrightarrow{T}$	
	Alice proves knowledge of representation of T with respect to (d_1, d_2) $\leftrightarrow$	
		$m = T \cdot P_A \cdot d_3$
		$z \leftarrow m^x$, $w \in_R Z_Q$
$s, u, v \in_R Z_Q$	$\xleftarrow{z,a,b}$	$a \leftarrow g^w$, $b \leftarrow m^w$
$z' \leftarrow z^s$, $m' \leftarrow m^s$, $m' \neq 1$		
$a' \leftarrow a^u g^v$, $b' \leftarrow b^{us}(m')^v$		
$c' \leftarrow \mathcal{H}(m', z', a', b')$		
$c \leftarrow \frac{c'}{u} \ mod\ q$	$\xrightarrow{c}$	
$h^c a \overset{?}{=} g^r$, $z^c b \overset{?}{=} m^r$	$\xleftarrow{r}$	$r \leftarrow w + cx \ mod\ q$
$r' \leftarrow ur + v \ mod\ q$		

Deposit: The merchant sends the transcript of the payment protocol execution to the bank and the bank checks its correctness. The bank checks that m' has not been spent before and then credits the merchant's account. If the same payment transcript is deposited twice the bank knows that the merchant tries to deposit the same coin twice. Otherwise if there are two different transcripts for the same money, they reveal two different linear equations $r = a_3 + ca_4$ and $r' = a_3 + c'a_4$. ¿From these two linear equations the bank computes $a_4 = s$. Recall that $m' = m^s \neq 1$, thus $q \not| s$, and as q is prime it follows that $(s, q) = 1$. Therefore, using the extended gcd algorithm, the bank can compute a number l s.t. $s \cdot l = 1(\ mod\ q)$. Now, $(m')^l = m^{ls\ mod\ q} = m$. The bank then searches its database to find the transaction in which m was withdrawn, revealing the identity of the double spender.

6.2 Security of the off–line system

For the security proofs we make the following (reasonable) assumptions, that have been commonly used in security proofs for (variants of) Brands' system. We assume $DLOG$ is hard and make the

Random oracle assumption : $\mathcal{H}, \mathcal{H}_0$ behave like random functions.

Withdrawal protocol assumption : The withdrawal protocol is a restrictive blind signature protocol, i.e. for the message $m = g_1^{a_1} d_1^{a_2} d_2^{a_3} d_3$ the receiver obtains a signed message of the form $g_1^{sa_1} d_1^{sa_2} d_2^{sa_3} d_3^s$ for a number s.

Theorem 2. *Under the above assumptions, the system is non–transferable, unforgeable and allows to detect double–spenders. Single–spenders have uncondi-*

Fig. 2. Payment

Alice		Merchant
$m' = g_1^{a_1} d_1^{a_2} d_2^{a_3} d_3^{a_4}$	$\xrightarrow{m',sig(m')}$	Verify signature , $m' \neq 1$
		$TD \leftarrow$ Time/Date
		$ID \leftarrow$ Merchant-ID
	$\xleftarrow{c}$	$c = \mathcal{H}_0(TD, ID, m')$
	$\xrightarrow{r=a_3+ca_4}$	
	Alice proves knowledge of representation of $\frac{m'}{d_2^r}$ w.r.t. $(g_1, d_1, \frac{d_3}{d_2^c})$ $\leftrightarrow$	

tional anonymity. If a user double spends his identity is revealed, but no knowledge is gained about his non-transferability secret key. If, in addition, Alice is required to sign each interaction during withdrawal, then no polynomial time bank can falsely accuse her of double spending she has not done.

Proof.
Unforgeability: Schnorr-type signatures are unforgeable under the random oracle assumption [21]. Brands showed that this implies that it is infeasible to existentially forge a coin [1].

Anonymity: If a user spends each coin once, then the information that the bank gets to learn includes: T and c at withdrawal time, m' and $sig(m')$ along with r on spending time and proofs of knowledge of a representation. Proofs of knowledge of a representation do not reveal, in an information theoretical sense, any information about the representation the user has. We next observe that until r is revealed the bank gets to see only T and has no clue as to the actual representation $T = d_1^{u_1} d_2^{u_2}$ the user has for it. Thus, from the bank's point of view, u_2 and hence $r = su_2 + cs$ (for a known c) is uniform and independent of all other values. We are left with T, c, m' and $sig(m')$ that participated in the signature generation. However as the signature scheme is unconditionally blind we have that c and T are independent of m' and $sig(m')$ as required. Thus the withdrawal and spending transcripts are statistically independent.

The non-transferability secret is protected: The non-transferability secret is protected even when a user double spends. To see that notice that the only place a user Alice uses her knowledge of S_A is in the payment protocol where she gives a proof of knowledge of a representation. However, this proof of knowledge, provably does not reveal any information about the actual representation Alice knows. All the rest can be simulated with the knowledge of P_A alone, and the bank can simulate it itself. Thus the bank does not get any information that he could not have obtained from P_A itself.

Double spending: First, because of the unforgeability property, when a user Alice double spends she uses a coin m' that has been withdrawn before, let us say w.l.o.g. again by Alice. As all players (including the bank) are polynomial time players they can not find two different representations for any number in G_q (unless with negligible probability) and in particular, Alice knows at payment time at most one representation (a_1, a_2, a_3, a_4) of m' with respect to the generators (g_1, d_1, d_2, d_3). As the signature scheme is restrictive the representation Alice knows of m' has the form (sb_1, sb_2, sb_3, sb_4) where (b_1, b_2, b_3, b_4) is the representation of $m = TP_A d_3$. Now we note that Alice knows a representation of T with respect to (d_1, d_2), let's say this representation is (u_1, u_2). Hence Alice knows a representation $(S_A, u_1, u_2, 1)$ of m, and as we noted before, this is the only representation *Alice* knows for m. We conclude that $b_4 = 1$ and $a_4 = s$.

If Alice convinces the merchant at payment time, then (except for negligible probability) she has to reveal $a_3 + ca_4$ by the the soundness property of the proof of knowledge protocol. Now, if Alice double spends the same coin appears in two payment transcripts with two different linear equations, and a_3 and $a_4 = s$ are revealed. In particular s is revealed and the bank can find m from m' as described in the protocol. Finally, the bank has full knowledge as to who withdrew m.

Non–transferability: If a user Charlie spends a coin m' this coin must have been withdrawn before. Let us say Alice withdrew the coin. As before, Alice knows a representation (sS_A, su_1, su_2, s) of m' with respect to the generators (g_1, d_1, d_2, d_3). By the properties of the proof of knowledge of a representation r must be $a_3 + ca_4$. We further notice that Alice knows a representation (a_1, a_2, a_4) of $\frac{m'}{d_2^r}$ with respect to the generators $(g_1, d_1, \frac{d_3}{d_2^c})$. Charlie also proves he knows a representation of $\frac{m'}{d_2^r}$ with respect to $(g_1, d_1, \frac{d_3}{d_2^c})$. As all the players are polynomial Charlie (in coalition with other players including Alice) can only know one representation (a_1, a_2, a_4), as polynomial players can know at most one solution to the representation problem. Hence, he knows $a_4 = s$ and $a_1 = sS_A$, and Charlie knows S_A.

Framing–freeness: If the bank claims Alice double spent $m' = m^s$ it has to present the signature Alice gave for m. Therefore, if the bank claims Alice double spent $m' = m^s$, then indeed Alice withdrew m and Alice knows a representation $(b_1 = sS_A, su_1, su_2, b_4 = s)$ of m'. As we assume the bank is also polynomial time the bank can not know any other representation for m'.

Now to prove double spending the bank also has to show two different transcripts for the deposit of the same coin m'. However, to show a valid transcript the bank has to answer a random challenge it has no control of (because of the random oracle assumption). We already showed that for a polynomial time bounded machine this amounts to the knowledge of b_1 and b_4 and hence of S_A. Thus, a polynomial time bank can not falsely accuse a user of double spending.

Finally, the bank can implement time frames by using different generator tuples for different time frames.

7 Defenses against Attacks and System Abuses

In this section we discuss how amount–limited and non-transferable payment systems help to defend against several attacks and abuses of payment systems.

Anonymous blackmailing : The anonymous blackmailing attack was introduced by van Solms and Naccache in [25]. The major benefit of amount limited non–transferable electronic cash is that *private* person to person blackmailing involving large sums of electronic cash is now impossible as a blackmail victim can withdraw at most 10. 000 $ per month. The cooperation of the powerful player of the bank is needed. The bank has its own interests and is unlikely to surrender. We do not claim blackmailing for electronic money becomes impossible. We do claim, however, that it becomes extremely more complicated and that it can be handled and controlled by policy decisions of bank and government.

Money laundering : The non–transferability and the amount limitedness feature of the system assure that the overall amount of tradable electronic money in the payment system is small. A consequence of this is that in order to move large funds anonymously around (e.g. from one country to another), the cooperation of many users is necessary who withdraw the needed amounts of electronic cash from their account and is thus probably impractical. Important traditional money laundering detection techniques like the observation of bank accounts that have a transaction activity bigger then the business of the account holder would justify may help to detect these type of suspicious activities.

Bribery : Suppose Alice wants to bribe Bob. As our system is non–transferable coins that have been withdrawn from Alice's account can only be spent by a user who knows Alice's non–transferability secret. Unless Alice is willing to reveal her non–transferability secret to Bob coins withdrawn from her account can not be used by Bob. Our system can not easily be made payee anonymous.

Purchase of illegal goods : As the system is amount–limited consumers may buy their weekend dose of cocaine with the system anonymously, however it is not possible to buy a pound of cocaine with it. A society might tolerate these "minor" abuses of electronic payment systems as it already does today with physical cash.

Bank robbery attack : This very strong attack where the secret key used by the bank to sign coins is compromised was introduced and defended against in [13]. The system that we describe in this paper does not defend against this attack.

It has been pointed out before in [12] that the vulnerabilities of several electronic payment systems to the bank robbery and the blackmailing attack have been a consequence of their usage of (blind) digital signature techniques. In [22] the authors describe an amount–limited and non–transferable payment system that does not rely on blind digital signature techniques and strongly defends against the blackmailing and the bank robbery attack. This shows that the vulnerabilities to these strong attacks are not a consequence of the anonymity features of electronic cash systems but rather of the technologies that have been used to implement them.

8 Acknowledgments

We would like to thank Andres Albanese, Oded Goldreich and Omer Reingold for interesting conversations. We are very thankful to Birgit Pfitzmann and the anonymous referees for their most valuable comments on an earlier version of this paper. Special thanks go to Markus Jakobsson for his many helpful comments that greatly improved the presentation of this paper.

References

1. S. Brands. An efficient off-line electronic cash system based on the representation problem. In 246. Centrum voor Wiskunde en Informatica (CWI), ISSN 0169-118X, December 31 1993. AA (Department of Algorithmics and Architecture), CS-R9323, URL=ftp://ftp.cwi.nl/pub/CWIreports/AA/CS-R9323.ps.Z. 48, 55, 55, 55, 55, 57
2. S. Brands. Untraceable off-line cash in wallet with observers. In Douglas R. Stinson, editor, Crypto 93, volume 773 of LNCS, pages 302–318. SV, 1993. 48
3. Ernie Brickell, Peter Gemmell, and David Kravitz. Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'95), pages 457–466. Sandia National Labs, January 1995. 46
4. J. Camenisch, U. Maurer, and M. Stadler. Digital payment systems with passive anonymity-revoking trustees. Lecture Notes in Computer Science, 1146:33–, 1996. 46
5. D. Chaum and T. Pedersen. Transferred cash grows in size. In R. A. Rueppel, editor, Advances in Cryptology—EUROCRYPT 92, volume 658 of Lecture Notes in Computer Science, pages 390–407. Springer-Verlag, 24–28 May 1992. 48
6. I. Damgard. Payment systems and credential mechanisms with provable security against abuse by individuals. In S. Goldwasser, editor, Crypto 88, LNCS, pages 328–335, Santa Barbara, CA, USA, August 1990. SV. 47, 51, 53
7. S. D'Amiano and G. Di Crescenzo. Methodology for digital money based on general cryptographic tools. In Alfredo De Santis, editor, Advances in Cryptology—EUROCRYPT 94, volume 950 of Lecture Notes in Computer Science, pages 156–170. Springer-Verlag, 1995, 9–12 May 1994. 48
8. G. Davida, Y. Frankel, Y. Tsiounis, and Moti Yung. Anonymity control in E-cash systems. In Rafael Hirschfeld, editor, Financial Cryptography: First International Conference, FC '97, volume 1318 of Lecture Notes in Computer Science, pages 1–16, Anguilla, British West Indies, 24–28 February 1997. Springer-Verlag. 46, 46
9. C. Dwork, J. Lotspiech, and M. Naor. Digitalsignets: Self-enforcing protection of digital information. In Proceedings of The Twenty-Eighth Annual ACM Symposium On The Theory Of Computing (STOC '96), pages 489–498, New York, USA, May 1996. ACM Press. 49
10. O. Goldreich, S. Micali, and A. Wigderson. How to play ANY mental game. In Alfred Aho, editor, Proceedings of the 19th Annual ACM Symposium on Theory of Computing, pages 218–229, New York City, NY, May 1987. ACM Press. 52, 52
11. S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. SIAM Journal on Computing, 17(2):281–308, 1988. Special issue on cryptography. 52, 52

12. M. Jakobsson and D. M'Raihi. Mix-based electronic payments. In Fifth Annual Workshop on Selected Areas in Cryptography, 1998. 50, 59
13. M. Jakobsson and M. Yung. Revokable and versatile electronic mony. In Clifford Neuman, editor, 3rd ACM Conference on Computer and Communications Security, pages 76–87, New Delhi, India, March 1996. ACM Press. 46, 59
14. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures. In CRYPTO: Proceedings of Crypto, 1997. 52, 52
15. R.C. Molander, D.A. Mussington, and P. Wilson. Cyberpayments and money laundering. RAND, 1998. http://www.rand.org/publications/MR/MR965/MR965.pdf/. 48, 48
16. S. Morris. Contribution to the panel "a session on electronic money: Threat to law enforcement, privacy, freedom, or all three?" at the sixth conference on computers, freedom, and privacy (cfp96), cambridge, ma. available as RealAudio document at http://swissnet.ai.mit.edu/ switz/cfp96/plenary-money.html, 1996. 47, 47
17. D. M'Raihi. Cost-effective payment schemes with privacy regulation. In Kwangjo Kim and Tsutomu Matsumoto, editors, Advances in Cryptology— ASIACRYPT '96, volume 1163 of Lecture Notes in Computer Science, pages 266– 275, Kyongju, Korea, 3–7 November 1996. Springer-Verlag. 46
18. T. Okamoto and K. Ohta. Disposable zero-knowledge authentications and their applications to untraceable electronic cash. In Advances in Cryptology: CRYPTO '89, pages 481–497, Berlin, August 1990. Springer. 48
19. T. Okamoto and K. Ohta. Universal electronic cash. In Joan Feigenbaum, editor, Proceedings of Advances in Cryptology (CRYPTO '91), volume 576 of LNCS, pages 324–337, Berlin, Germany, August 1992. Springer. 48
20. B. Pfitzmann and M. Waidner. How to break and repair a "provably secure" untraceable payment system. In Joan Feigenbaum, editor, Proceedings of Advances in Cryptology (CRYPTO '91), volume 576 of LNCS, pages 338–350, Berlin, Germany, August 1992. Springer. 47, 51, 53
21. D. Pointcheval and J. Stern. Security proofs for signature schemes. In Ueli Maurer, editor, Advances in Cryptology—EUROCRYPT 96, volume 1070 of Lecture Notes in Computer Science, pages 387–398. Springer-Verlag, 12–16 May 1996. 52, 57
22. T. Sander and A. Ta–Shma. Auditable, counterfeiting resistant electronic cash. In preparation. 59
23. C. Schnorr. Efficient signature generation by smart cards. Journal of Cryptology, 4(3):161–174, 1991. 55
24. Paul F. Syverson, Stuart G. Stubblebine, and David M. Goldschlag. Unlinkable serial transactions. In Rafael Hirschfeld, editor, Financial Cryptography: First International Conference, FC '97, volume 1318 of Lecture Notes in Computer Science, pages 39–55, Anguilla, British West Indies, 24–28 February 1997. Springer-Verlag. 54
25. S. von Solms and D. Naccache. On bline signatures and perfect crimes. Computers and Security, 11(6):581–583, October 1992. 59
26. A. Yao. How to generate and exchange secrets. In 27th Annual Symposium on Foundations of Computer Science, pages 162–167, Los Angeles, Ca., USA, October 1986. IEEE Computer Society Press. 52