ON DIGITAL CASH-LIKE PAYMENT SYSTEMS

Daniel A. Nagy
Queen's University, Dept. of Math. and Stats.
Jeffrey Hall, Kingston, ON, K7L 3N6, Canada
Email: nagydani@mast.queensu.ca

Keywords: online payment systems, digital cash, security, cryptography

Abstract: In present paper a novel approach to on-line payment is presented that tackles some issues of digital cash that have, in the author's opinion, contributed to the fact that despite the availability of the technology for more than a decade, it has not achieved even a fraction of the anticipated popularity. The basic assumptions and requirements for such a system are revisited, clear (economic) objectives are formulated and cryptographic techniques to achieve them are proposed.

1 INTRODUCTION

Chaum *et al.* begin their seminal paper (D. Chaum, 1988) with the observation that the use of credit cards is an act of faith on the part of all concerned, exposing all parties to fraud. Indeed, almost two decades later, the credit card business is still plagued by all these problems and credit card fraud has become a major obstacle to the normal development of electronic commerce, but digital cash-like payment systems similar to those proposed (and implemented) by D. Chaum have never become viable competitors, let alone replacements for credit cards or paper-based cash.

One of the reasons, in the author's opinion, is that payment systems based on similar schemes lack some key characteristics of paper-based cash, rendering them economically infeasible. Let us quickly enumerate the most important properties of cash:

1. "Money doesn't smell." Cash payments are – potentially – *anonymous* and untraceable by third parties (including the issuer).
2. Cash payments are final. After the fact, the paying party has no means to reverse the payment. We call this property of cash transactions *irreversibility*.
3. Cash payments are *peer-to-peer*. There is no distinction between merchants and customers; anyone can pay anyone. In particular, anybody can receive cash payments without contracts with third parties.
4. Cash allows for "acts of faith" or *naïve transactions*. Those who are not familiar with all the anti-forgery measures of a particular banknote or do not have the necessary equipment to verify them, can still transact with cash relying on the fact that what they do not verify is nonetheless verifiable in principle.
5. The amount of cash issued by the issuing authority is public information that can be verified through an auditing process.

The payment system proposed in (D. Chaum, 1988) focuses on the first characteristic while partially or totally lacking all the others. The same holds, to some extent, for all existing cash-like digital payment systems based on untraceable blind signatures (Brands, 1993a; Brands, 1993b; A. Lysyanskaya, 1998), rendering them unpractical.

In his invited paper to Scientific American (Chaum, 1992), D. Chaum eloquently argues the importance of untraceability, so there is no need to repeat it here. It is worth noting, however, that while coins are truly untraceable in practice, paper-cash with its unique serial numbers is not. Yet, it does not seem to hamper its wide acceptance, because the anonymity of the transactions provides for sufficient privacy. The importance of the other four characteristics lies in the economics behind cash:

Irreversibility removes an important transaction cost, namely that of potential reversal. An insurance against reversal has to be built into the price of

services offered in exchange for reversible payment. Anonymity is a necessary, but not sufficient component of irreversibility. The payment system proposed in (D. Chaum, 1988) sacrifices irreversibility in order to allow for off-line transactions, assuming that communication with the issuing authority is more expensive than communication between the transacting parties or complex computations. At the time of writing, this might have been the case, but today, when the infrastructure for low-bandwidth communication (such as short text messages, http queries, etc.) is ubiquitous, the benefits of off-line transactions are clearly inferior to those of irreversible transactions.

The *peer-to-peer* nature of a payment system also removes a significant cost; if a contract with a third party is necessary to receive payments, it is very likely that this third party will charge for its service. This raises the entry barrier for sellers and thus narrows the assortment of goods and services available in exchange for the payment that is not peer-to-peer, reducing its liquidity. In addition to this, merchant contracts unnecessarily expose sellers to the provider of the payment service; their income becomes known. It is important to emphasize that by peer-to-peer payment I do not imply that there are no servers or other centralized entities involved; it merely means that there is no distinction between sellers and buyers, merchants and customers. Anyone can pay anyone.

Naïve transactions help reducing the costs of distributing the tools (hardware and software) used for transactions. Contrarily to the assumptions of (D. Chaum, 1988), computation is far less ubiquitous than communication. While everyone with a cellular or a touch-tone telephone, a web-browser or email client in its readily available, out-of-box configuration is able to transmit short messages (up to a few hundred bits), performing complex calculations involving strong asymmetric cryptography requires additional tools which not everyone possesses or can afford to run. The fact that it is impossible to transact without performing complex calculations in real time is a far more serious obstacle than the need to contact the issuer for each transaction. It also undermines the trust in the system, as the the failure of the equipment used for storing and transacting with such "cash" (a very serious problem with (Brands, 1993b)) can cause unlimited damage, that cannot be mitigated. The fact that low-tech, naïve transactions are possible (and, in fact, quite common) with cash, greatly contributes to its acceptance and popularity. It is important to stress that no-one is *forced* to transact naïvely, and always has a choice of performing extra verification and discover attempts at cheating. Just as one always has the option of verifying one or more security features of a banknote before accepting it.

The *transparent governance* of the issuer is perhaps the most important reason to trust it. If the issuer is able to issue digital money without anybody noticing, its creditworthiness cannot be established and the incentive to hyper-inflate (overborrowing by irresponsible emission) is enormous. While the information about the distribution and the holders of cash is private, its total amount should be public and verifiable. The lack of transparency of emission, in the author's opinion, is among the primary reasons for the failure of digital cash-like payment systems in the market.

In the rest of the paper, we develop a set of protocols that provide for all of the above characteristics of a digital payment system under certain model assumptions. The proposed system resembles the one proposed by Jakobbson (Jakobsson, 1999) in that it can be regarded as one with disposable anonymous accounts. Such disposable anonymous account based systems have achieved greater acceptance in the market (most notably WebMoney at `http://wmtransfer.com`) than those based on untraceable transfers between accounts tied to identity, but the current implementations either do not provide sufficient security for high-value transactions or impose too high overhead costs on low-value ones. The system outlined in this paper permits the users to choose the appropriate security measures that they deem appropriate for the given transaction. This is our principal contribution.

2 PRELIMINARIES

In the proposed system, the issuer I maintains a public record of transactions, consisting of a chronologically ordered sequence of digitally signed statements S_i, where $i = 1, 2, 3, \ldots$ is called the *serial number* of the statement. The serial number can be unambiguously inferred from S_i. Digitally signed means that anybody can verify using only publicly available information in a computationally inexpensive way that S_i originates form I. Public-key signature schemes such as those described in (R. L. Rivest, 1978; Elgamal, 1985; NIST, 1991) can provide for such functionality in practice, together with some public key distribution protocol. These implementation details lie outside of the scope of this paper.

After some S_n has been published, it can be verified by anyone that for all $i \in \mathbb{N}^+$ such that $i < n$, S_i has also been (previously) published and that different statements do not share the same serial number. The structure of the statements is the following: $S_i = (i, I, V_i, C_i, N_i, \Sigma_i)$ where $\Sigma_i = \sigma_I(i, I, V_i, C_i, N_i)$ is the digital signature unique to the rest of S_i and I. Each statement implies the promise of issuer I to pay V_i units of value to anyone who first responds to cryptographic challenge C_i (which requires the possession of some secret D_i). N_i is the request message result-

ing in issuing S_i that may be the response to some earlier C_j (where $j < i$). Note that in a practical implementation the promise can be explicitly stated as an additional piece of information within S_i, signed together with the rest.

The request message N_i can be one of the following five kinds:

1. $\mathcal{E}$: emission request. In this case $N_i = (\mathcal{E}, C_i, V_i, \Omega_i)$ where C_i is a new challenge, V_i is the value of newly issued currency and $\Omega_i = \sigma_J(\mathcal{E}, C_i, V_i)$ is the digital signature unique to the rest of N_i and J – an authorized individual. After receiving N_i, the issuer verifies Ω_i and the fact that C_i has never been used before. If the request is accepted, a new statement $S_i = (i, I, V_i, C_i, N_i, \Sigma_i)$ is issued where i is just the next available serial number at the time of receiving the request.
2. $\mathcal{X}$: exchange request. In this case $N_i = (\mathcal{X}, j, C_i, R_j)$ where $j < i$, C_i is a new challenge and R_j is the additional information making N_i a valid response to C_j – an older challenge. After receiving N_i, the issuer verifies whether or not it constitutes the first valid response to C_j, and if it does and C_i has never been used before, the new statement $S_i = (i, I, V_j, C_i, N_i, \Sigma_i)$ is issued. The fact of issuing S_i "invalidates" S_j in that future responses to C_j can be rejected by pointing to N_i inside S_i; a previous response.
3. $\mathcal{M}$: merge request. In this case $N_i = (\mathcal{M}, j, k, R_j)$ where $j, k < i$,R_j is the additional information making N_i a valid response to C_j – an older challenge. After receiving N_i, the issuer verifies whether or not it constitutes the first valid response to C_j, and if it does and S_k is a pending promise in that it has not been fulfilled or superseded answering an earlier request, the new statement $S_i = (i, I, V_j + V_k, C_k, N_i, \Sigma_i)$ is issued. The fact of issuing S_i fulfills the promise of S_j and supersedes the promise of S_k thus "invalidating" both of those.
4. $\mathcal{S}$: split request. In this case $N_i = (\mathcal{S}, j, C_i, V_i, C_{i+1}, R_j)$ where $j < i$, C_i and C_{i+1} are new challenges, $V_i < V_j$ and $R_j$is the additional information making N_i a valid response to C_j – an older challenge. After receiving N_i, the issuer verifies whether or not it constitutes the first valid response to C_j, and if it does and C_i and C_{i+1} have never been used before and $V_i < V_j$ then two new statements are issued: $S_i = (i, I, V_i, C_i, N_i, \Sigma_i)$ and $S_{i+1}(i + 1, I, V_j - V_i, C_{i+1}, N_i, \Sigma_{i+1})$. The fact of issuing S_i or S_{i+1} "invalidates" S_j by fulfilling its promise. Note that S_i and S_{i+1} can be reconstructed from one another by I, thus the issuing of the two can be regarded as an atomic operation.
5. $\mathcal{I}$: invalidation request. In this case $N_i = (\mathcal{I}, j, \Omega'_i, R_j)$ where $j < i$ and $R_j$is the additional information making N_i a valid response to C_j – an older challenge and $\Omega'_i = \sigma_J(\mathcal{I}, j)$ is the digital signature of J – an authorized individual. After receiving N_i, the issuer verifies Ω'_i and whether or not N_i constitutes the first valid response to C_j, and if it does, the new statement $S_i(i, I, 0, C_j, N_i, \Sigma_i)$ is issued effectively invalidating S_j and removing the amount V_j from circulation. N_i constitutes a proof that the promise of S_j has been fulfilled (outside the payment system).

Since all S_i statements are public, anyone can verify that they follow the above specification. Most importantly that the ones with $\mathcal{X}$, $\mathcal{M}$ and $\mathcal{S}$ requests make and fulfill promises of equal values. The amount of issued currency can be calculated as follows:

$$V = \sum_{i:N_i \in \mathcal{E}} V_i - \sum_{i:N_i \in \mathcal{I}} V_{j(N_i)}$$

that is the summary value of emission requests minus the summary value of invalidation requests. The ability of the issuer to live up to its outstanding promises can be verified through traditional auditing.

3 TRANSACTION PROTOCOLS

A party in possession of D_i is said to be the holder of the (public) promise embodied in S_i, unless that promise has already been fulfilled or superseded. Thus, it is the set of D_i secrets that constitute the title to certain value. The physical means of storing these secrets does not really matter as long as it permits the owner to protect the secrecy and to retrieve when necessary. Because of this, anybody can hold such a currency who is able to store and retrieve small amounts of information, allowing for the third property of cash discussed in section 1. The "act of faith" mentioned in conjunction with the fourth property of cash is in this case believing that some D_i indeed corresponds to C_i from S_i which is indeed a statement issued by the issuer.

In order to transact securely, the following capabilities can be required:

1. Sending short messages to the issuer in a reliable fashion and access to the public records with the issuer's public statements.
2. Verifying the digital signature of the issuer.
3. Generating random pairs of challenges C and corresponding secrets D required for a valid response.
4. Generating R_i for a valid response to some C_i once in possession of the corresponding D_i.
5. An established digital identity with the capability of sending signed messages in a secure fashion.

3.1 FUND TRANSFER WITHOUT RECEIPT

In this scenario Alice (A) wants to transfer some V amount of funds to Bob (B), but does not need a proof for some third party that Bob has received the funds; all she needs is to make sure that nobody else but Bob receives the payment and she knows that it has happened. For doing so, Alice only needs to possess some $\mathbf{D}_k = \{D_{k1}, D_{k2}, \ldots\}$ set of secrets so that $\sum_{j \in k(\mathbf{D}_k)} V_j \geq V$ that is she needs to have enough funds.

1. A assembles a set $\mathbf{D}_m = \{D_{m1}, D_{m2} \ldots\}$ of secrets such that $\sum_{j \in \{m(\mathbf{D}_m)\}} V_j = V$ and a set $\mathbf{J}_m$ of corresponding serial numbers. If $\mathbf{D}_k$ has a suitable subset, then she can use that. If not, she selects a subset $\mathbf{D}_n \subset \mathbf{D}_k$ (with a corresponding serial number set $\mathbf{J}_n$) and an additional secret $D_x \in \mathbf{D}_k \setminus \mathbf{D}_n$ such that $\sum_{j \in n(\mathbf{D}_n)} V_j < V$ and $\sum_{j \in n(\mathbf{D}_n)} V_j + V_x > V$. Then she generates two new challenge-secret pairs (C_y, D_y) and (C_z, D_z), and sends the message $N = (\mathcal{S}, x, C_y, V - \sum_{j \in \{n(D_n)\}} V_j, C_z, R_x)$ to I. At this point, $\mathbf{D}_m := \mathbf{D}_n \cup \{D_x\}$ and $\mathbf{J}_m = \mathbf{J}_n \cup \{i\}$ where i is the serial number of the statement that I published in response to N.
2. A sends $\mathbf{D}_m = \{D_{m1}, D_{m2}, \ldots\}$ and $\mathbf{J}_m = \{j_{m1}, j_{m2}, \ldots\}$ to B.
3. B generates a set of new challenge-secret pairs $\{(C_{b1}, D_{b1}), (C_{b2}, D_{b2}), \ldots\}$ with the same cardinality as $\mathbf{D}_m$ and $\mathbf{J}_m$. Then, for each $D_k \in \mathbf{D}_m$ he sends the following message to I: $N_k = (\mathcal{X}, j_{mk}, C_{bk}, R_k)$ where R_k is calculated from D_k and the rest of N_k. His set $\mathbf{D}_b = \{D_{b1}, D_{b2}, \ldots\}$ becomes a value worth V at this point. Using further $\mathcal{S}$ and $\mathcal{M}$ messages, he can rearrange it in any way he wants.
4. A can verify that the transaction has been completed by verifying that all the promises embodied in $\{S_j : j \in \mathbf{J}_m\}$ have been fulfilled. If she is convinced that the message to B has not been intercepted, she can be also convinced that B took possession of the transfered funds. However, she has no means of proving this to a third party.

The above described transaction is in direct analogy with cash transfers: first A selected an appropriate amount of cash from her wallet (possibly splitting a large denomination at the bank to make sure she has exact change), then handed it over to B, who exchanged all the received cash with the bank (so that A doesn't know the serial numbers of his banknotes).

This protocol is perfectly suitable for low-value purchases (e.g. micropayment), as the computational and communicational requirements on A's part are minimal. For example, if A has to pay for viewing a webpage, and she has "exact change", that is she possesses some D_x, such that the corresponding S_x is a valid promise of the required value, all she has to do is to enter D_x (and x, if it cannot be retrieved) when the website asks for payment.

Note, furthermore, that the transaction is initiated by the sender, thus anybody can be paid in this fashion; the naïve recipient can make an "act of faith" (believing that the honest sender "forgot" the secrets that have been transfered) and use the received secrets as payment without exchanging them with the issuer.

3.2 FUND TRANSFER WITH RECEIPT

For high-value transactions, the protocol described in 3.1 is unsuitable, because the recipient can deny the receiving of funds without legal or reputational consequences, as the sender has no means to prove it to a third party. In the scenario described in this section, Alice (A) wants to buy something expensive (worth V) from Bob (B).

1. B generates a challenge-secret pair (C, D) and sends the signed invoice $Y = (V, C, X, \Theta)$ to A, where X is the identifier of the service that A wants to buy from B and $\Theta = \sigma_B(V, C, X)$ is the digital signature of the invoice by B. Y constitutes a promise by B that upon receiving V amount of funds in a way accessible to the holder of the secret corresponding to C he will perform X.
2. A verifies Θ and if it is correct, she assembles $\mathbf{D}_m$ as in 3.1 and sends a sequence of messages to I. The first message is $N_{m1} = (\mathcal{X}, j_{m1}, C, R_{m1})$ where R_{m1} is calculated from D_{m1} and the rest of N_{m1}. Let i_{mk} denote the serial number of the statement published by I in response to N_{mk}. N_{m1} is followed by a sequence of $N_{mk} = (\mathcal{M}, j_{mk}, i_{m(k-1)}, R_{mk})$ for $k = 2, 3, \ldots$.
3. At this point A is in possession of a conclusive proof that she has fulfilled her side of the contract: the pair $P_X = (Y, i_{mk})$. The private invoice Y is signed by B certifying his offer, while the public statement $S_{i_{mk}}$ signed by I certifies that the corresponding payment has been made. The two are linked by the equalities of $V_{i_{mk}} = V$ and $C_{i_{mk}} = C$. A sends P_X to B.
4. B extracts V, C and Θ from the received P_X, verifies Θ, downloads $S_{i_{mk}}$ from the public records, verifies $\Sigma_{i_{mk}}$ and checks whether $V_{i_{mk}} = V$ and $C_{i_{mk}} = C$. If everything matches, he performs X.

In case of dispute, A can show P_X to the judge at which point it is up to B to prove that X has been performed.

4 ATTACKS AND VULNERABILITIES

The security of the proposed payment system depends on the nature of the used cryptographic challenges; the actual objects behind C_i, D_i and R_i and the way the various messages are transfered between the various participants. Even without defining these, it is clear that the untraceability hinges on the fact that the users of the payment system are not identified when sending the messages (those denoted by N_i) to the issuer. This is weak anonymity and in this respect the proposed system is inferior to the ones based on blind signatures. However, it is not inferior to paper-based cash and prevents the issuer from knowing the turnover of the individual users, which the system described in (D. Chaum, 1988) does not.

It is also very important to emphasize that the costs of protecting oneself against fraud should not exceed the transaction value. Since on-line payment systems are often used for micropayment, it is important that it can be performed with minimum effort and tools, even at the cost of exposing oneself to fraud by a highly sophisticated attacker. As long as the attack costs significantly more than the transaction value, the payment system can be considered secure enough.

In this section, some attack and fraud scenarios are investigated.

4.1 THEFT

Successful theft is defined as follows: attacker (T) manages to make I issue a public statement S_t so that $V_t > 0$ and D_t corresponding to C_t in known to T, even though T has not previously owned any secret corresponding to the challenges on the already published statements.

By definition, I issues public statements with $V_i > 0$ only upon accepting $\mathcal{E}$, $\mathcal{X}$, $\mathcal{M}$ and $\mathcal{S}$ messages. Thus, T needs to forge one of these.

The acceptance of $\mathcal{E}$ messages depends on the validity of Ω_i. If the signature function σ is secure, then forging the signature for an $\mathcal{E}$ message with a newly generated C_i is infeasible. Previous valid $\mathcal{E}$ messages cannot be reused, because C_i has to be new. Intercepting and modifying a valid $\mathcal{E}$ message is similarly computationally infeasible, if σ is secure.

$\mathcal{X}$, $\mathcal{M}$ and $\mathcal{S}$ messages are accepted if they constitute a valid response to some earlier challenge C_j. One way of forging such a message is by guessing the secret that corresponds to one of the valid challenges. Let us assume that there is some maximal reasonable complexity for the challenge and the probability of finding a corresponding secret by random guess to such a challenge is p. Note that the secret is not assumed to be unique to the challenge. If there are n valid challenges, the probability of guessing one is $1-(1-p)^n$ which if $p \ll n^{-1}$ is approximately pn. If T has the resources for trying m secrets, the probability of one of them corresponding to a valid challenge equals $1-(1-p)^{nm}$ which if $p \ll m^{-1}n^{-1}$ is approximately pnm. If this number is comfortably low, the system is secure against brute-force attacks. However, since it is the users who generate the challenge-secret pairs, I cannot protect them against poorly chosen (low-entropy) secrets; having a weak random source leaves one vulnerable to theft.

Another way of forging $\mathcal{X}$, $\mathcal{M}$ or $\mathcal{S}$ messages is by extracting a suitable secret from public information or intercepted communication. In the public records, T can find a large number of challenge-response pairs and this number is growing as the system is being used. Thus, it is instrumental for the security of the system that challenges and responses are uncorrelated.

Secrets are being communicated directly in the protocol described in 3.1. Hence, if $T$is able to intercept and decode the messages that payers send to recipients in this protocol, he is able to use them before the recipient. Therefore, it is important that the secrecy of the communication between the users is well protected in this protocol. Otherwise the payer is vulnerable to theft during the period of time when A has already sent the message to B but B hasn't yet sent the messages to I. Naïve recipients who do not exchange the received secrets immediately are vulnerable not only to fraud by the payer but also to theft if the communication has been intercepted. Naïve payers using open channels of communication are vulnerable to theft for a short period of time, which for micropayments can be a manageable risk worth taking, if using a secure channel is overly expensive.

In both protocols described in 3.1 and 3.2, one of the parties sends messages to I. If such a message (N_i) can be intercepted and decoded by T before it reaches I, much depends on the nature of the cryptographic challenge. In general, R_i is the function of some D_j and the rest of N_i. If it is feasible to compute (or guess with a high probability) D_j from N_i or some R_i' so that substituting C_i with C_i' (generated by T) and R_i with R_i' results in a valid response to C_j before the message reaches I, then the parties are vulnerable to theft. If it is not feasible to forge R_i without knowing D_j or to alter the message so that N_i' remains a valid response to C_j then the parties are not vulnerable to theft in this way, even if the communication with I can be intercepted, decoded and tampered with.

4.2 USER FRAUD

User fraud is intentional deception of a user of the payment system by another user assuming that the

issuer issues public statements only as described in 2. There are two meaningful deceptions within a payment system: the paying party (A) fraudulently claims that a payment has been made or the receiving party (B) fraudulently claims that a payment has not been received.

There are two distinct issues with fraud: whether or not the other party can detect it and whether or not it can be proven to a third party. Naïve users can be defrauded in many ways. The present analysis is restricted to participants that perform all the necessary verifications in order to avoid being deceived.

In case of receiptless fund transfer as described in 3.1, fraudulent claims (of both kinds) cannot be proven to a third party, as none of the messages used in the transaction can be linked to the involved parties. However, A and B know exactly what has happened. Thus, this protocol is suitable only for transactions where one can afford the loss of one payment to establish the dishonesty of the other party.

The protocol described in 3.2 offers much better protection against fraud for both users. In order to claim a payment, A must produce P_X. A valid P_X, where $i(P_X)$ and $Y(P_X)$ are such that $C(S_i) = C(Y)$, cannot be produced without actually transferring the right amount of funds into B's exclusive possession, assuming that Y (which is signed by B) cannot be forged and some other Y' does not offer the same service X. It is instrumental that X is unique to each transaction. In order for A to be able to verify the uniqueness of X, X may incorporate a signed order of the service from A. P_X is a conclusive proof of the payment, disproving the fraudulent claim of B that the payment has not been made. The proof of rendering service X depends on the service and is not part of the payment system.

The vulnerability of users to fraud by one another does not depend on how the cryptographic challenges are implemented, as long as it is computationally infeasible to respond to the challenge without knowing the corresponding secret.

4.3 ISSUER FRAUD

By issuing digital currency, I is essentially borrowing from all holders. In this framework, fraud can be interpreted as misrepresenting one's creditworthiness. The public sequence $S_1, S_2, \ldots, S_i$ is essentially I's credit history. There are two kinds of meaningful deceptions when it comes to credit: borrowing more or defaulting on (parts of) existing debt without leaving a trace in the credit history. Since in the proposed system the act of borrowing (issuing) is the same as the act of publishing it (Grigg, 2004), the first kind of fraud is impossible by definition.

Thus, we can define successful issuer fraud as failure to respond to user messages as defined in 2. It is important to emphasize that the defrauded party is *not* the one who has sent the message that has not been appropriately processed but all the holders of I's "currency". Before going into more detailed analysis, it is worth noting that the messages do not contain information regarding their origin, thus if I attempts fraud, it can never be sure that it is not a spot-check by an auditor or the law enforcement. Thus, there are strong incentives not to commit fraud when dealing with anonymous customers.

The first important observation is that I can ignore the received messages; pretend as if they have not been received. It is the information carrier service that can provide various facilities to prevent this from happening without a trace, but most carriers do not provide them.

Secondly, the issuer can obviously do anything to a message that an attacker described in section 4.1 can. Thus, all the vulnerabilities mentioned there apply; theft can be perpetrated by I under the same conditions. The only difference is that I can operate with $\mathcal{I}$ requests as well.

If the cryptographic challenge is implemented in such a way that a valid response does not divulge the secret or allow for altered valid responses (see also 4.1), the customer (A) can accuse I by publishing the message (N) that has been supposedly ignored by I. Of course, it has no immediate consequences for I, as A could not have transfered the message previously, but after N has been published, I can disprove the accusation of fraud by processing N.

5 SUITABLE CRYPTOGRAPHIC CHALLENGES

In this section, some cryptographic challenge implementation are proposed and their advantages and disadvantages explained. Since it is the legitimate holder of the value who picks the corresponding challenge, it is possible to implement more challenges and let the users decide which ones they deem appropriate for protecting their wealth and privacy.

This list is by no means comprehensive. New kinds of challenge-response pairs are being developed, fulfilling various requirements resulting from different assumptions about the capabilities of the paying and the receiving party.

5.1 MESSAGE DIGEST (HASH) FUNCTION

Challenge: $C = h(D)$, an element from the range of the hash function

Secret: D, a random element from the domain of the hash function

Response: $R = D$ same as the secret. Valid if $h(R) = C$.

In this case, the challenge is the cryptographic hash (e. g. the SHA1[1](NIST, 1995)) of the secret, which is chosen randomly from a large enough pool, so that the probability of guessing it is sufficiently low. The valid response to the challenge is simply a message including the secret itself.

The advantages of this implementation are the following: It is very simple and computationally undemanding, offering good protection with relatively short secrets (e.g. 200 bits), that can be transfered using very narrow channels (e.g. speech, barcodes, typing, etc.). It is easy to compute the challenge corresponding to the secret, which in turn can be used as the key of the public statement database. Hence, it is not necessary to store the index of the corresponding statement together with the secret.

The disadvantage is that the response reveals the secret, thus leaving the payer vulnerable to theft, when communicating with the issuer over an insecure channel.

5.2 PUBLIC KEY SIGNATURE

Challenge: $C = K$, a public signature key

Secret: $D = K'$, the private pair of K

Response: $R = \sigma_K(N')$, the digital signature of the message.

In this case, the challenge is a public signature key (e.g. an RSA or a DSA public key (NIST, 1991; R. L. Rivest, 1978)) and the secret is its private pair. The two are selected randomly by the customer from a large enough pool, so that the probability of guessing is sufficiently low. The valid response is a message with a valid digital signature.

The advantages are the following: The secret is not revealed by the response, thus the ownership can be proved without disclosure. It is secure against theft even if the communication channel is insecure. It is secure against theft by the issuer.

[1]At the time of writing, the collision attack against SHA1 by Wang *et al.* was not known. However, even a successful collision attack against the hash function used in this implementation (and the ones below) does not allow, to the author's best knowledge, for attacks against the proposed payment system, as long as finding pre-images for a given hash value is infeasible, so even in the light of recent developments, it is safe to use SHA1 for this purpose. Nevertheless, it may be wise to consider alternatives. Collision attacks against the hash function used as part of the σ function (from Section 2) can be more worrisome.

Disadvantages: The secret is too long to be transfered though low-capacity channels or to be recorded quickly using low-tech means (e.g. scribbled down on a piece of paper). The transactions are computationally costly. In particular, generating a secure random key-pair takes minutes even on a modern computer. Elliptic Curve Cryptography (ECC) promises to alleviate these problems to some extent by providing equivalent security with shorter keys.

Note that it is possible to use a blinding scheme compatible with this type of challenge to break traceablity. The real difficulty in this case is preserving the accountability of the issuer. A scheme similar to the one proposed in (M. Stadler, 1995) could be utilized as a disincentive for the issuer to issue unbacked "coins".

5.3 PUBLIC-KEY SIGNATURE AND MESSAGE DIGEST

Challenge: $C = h(K)$, the one-way hash of the public signature key

Secret: $D = (K, K')$, a public/private key pair

Response: $R = (\sigma_K(N'), K)$, a digital signature and the corresponding public key

This modification of the previous scheme allows for seamless integration with the scheme described in 5.1, as the challenge has the same format. Thus, the same system can easily provide for both kinds of challenges. The advantages and the disadvantages are the same as those in 5.2.

An additional advantage is that the public key is not available for cryptanalysis by an attacker until too late. Since the key has to be used only once for generating exactly one signature, it can be substantially weaker than the one required for the previous case, allowing for a decrease in the required computational power on the user side even with traditional asymmetric cryptography.

5.4 PUBLIC-KEY SIGNATURE AND SYMMETRIC-KEY BLOCK CIPHER

Challenge: $C = (K, \rho_D(K'))$, a public signature key and the encrypted version of its private pair

Secret: D, a randomly chosen symmetric key for the block cipher ρ.

Response: $R = \sigma_K(N')$, the digital signature of the message

In this case, the challenge consists of a public signature key and its private counterpart encrypted using a symmetric-key block cipher ρ (e.g.). The secret D is

the symmetric key needed to decrypt the private key K'. The valid response is the same as in 5.2.

The advantage of this challenge over the one described in 5.2 is that the secret is short and thus can be transfered and stored easily using low-tech means, similarly to 5.1. However, the challenge cannot be deduced from the secret, thus one needs to record the index of the corresponding public statement as well.

5.5 MESSAGE DIGEST, PUBLIC-KEY SIGNATURE AND BLOCK CIPHER

Challenge: $C = (h(D), K, \rho_D(K'))$, a hash of the secret, a public key and the encrypted version of its private pair

Secret: D, a randomly chosen symmetric key for the block cipher ρ.

Response: $R = \sigma_K(N')$ or $R = D$, the digital signature of the message or the secret itself

In this case, the challenges described in sections 5.1 and 5.4 are used in conjunction, so that a valid response to either one is accepted. The corresponding secret is the same.

The advantages of this approach include all the advantages of the two methods, with the exception of computational simplicity offered by 5.1; generating a random challenge is still difficult.

It is up to the individual customers to chose which part of the challenge they use, depending on the available facilities and security requirements.

6 CONCLUSIONS

The proposed digital payment system is more similar to cash than the existing digital payment solutions. It offers reasonable measures to protect the privacy of the users and to guarantee the transparency of the issuer's operations. With an appropriate business model, where the provider of the technical part of the issuing service is independent of the financial providers and serves more than one of the latter, the issuer has sufficient incentives not to exploit the vulnerability described in 4.3, even if the implementation of the cryptographic challenge allowed for it. This parallels the case of the issuing bank and the printing service responsible for printing the banknotes.

The author believes that an implementation of such a system would stand a better chance on the market than the existing alternatives, none of which has lived up to the expectations, precisely because it matches paper-based cash more closely in its most important properties.

Open-source implementations of the necessary software are being actively developed as parts of the ePoint project. For details, please see
`http://sf.net/projects/epoint`

REFERENCES

A. Lysyanskaya, Z. R. (1998). Group blind digital signatures: A scalable solution to electronic cash. In *Proceedings of Financial Cryptography: Second International Conference (FC'98)*, page 184.

Brands, S. (1993a). An efficient off-line electronic cash system based on the representation problem. *Technical Report CS-R9323, CWI*.

Brands, S. (1993b). Untraceable on-line electronic cash in wallets with observers. In *Advances in Cryptology, CRYPTO '92*, pages 302–318. Springer-Verlag.

Chaum, D. (1992). Achieving electronic privacy. *Scientific American*, pages 96–101.

D. Chaum, A. Fiat, M. N. (1988). Untraceable electronic cash. In *Advances in Cryptology, CRYPTO '88*, pages 319–327. Springer-Verlag.

Elgamal, T. (1985). A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, IT-31:469–472.

Grigg, I. (2004). The ricardian contract. In *Proceedings of IEEE Workshop on Electronic Contracting July 6*, pages 25–31.

Jakobsson, M. (1999). Mini-cash: A minimalistic approach to e-commerce. In *Proceedings of the Second International Workshop on Practice and Theory in Public Key Cryptography*, pages 122–135.

M. Stadler, J.-M. Piveteau, J. C. (1995). Fair blind signatures. In *Advances in Cryptology, EUROCRYPT '95*, volume 921, pages 209–210. Springer-Verlag.

NIST (1991). Proposed federal information processing standard for digital signature standard (dss). *Federal Register*, 56:42980–42982.

NIST (1995). Secure hash standard. *FIPS 180-1*.

R. L. Rivest, A. Shamir, L. A. (1978). A method for obtaining digital signatures and public-key cryptosystem. *Communications of the ACM*, 21:120–126.