Anonymous Communication and Anonymous Cash

Daniel R. Simon

Microsoft Corp.
One Microsoft Way
Redmond WA 98052 USA
dansimon@microsoft.com

Abstract. We propose considering the problem of electronic cash in the context of a network in which anonymous, untraceable communication is assumed to be possible. We present a formal model for such a network, and define security criteria for an electronic cash system in such a setting. Finally, we show that there exists a remarkably simple electronic cash system which meets the security criteria of the proposed model.
Key words: Electronic cash, anonymous communication

1 Introduction

Money is always there but the pockets change; it is not in the same pockets after a change, and that is all there is to say about money.

—Gertrude Stein

The ultimate intuitive goal of an electronic cash system is to combine the best features of physical cash (privacy, anonymity, unforgeability) with the best features of electronic commerce (speed, ease and potential security of transport and storage). A large number of solutions to the problem have been proposed, but all have lacked a full, formal model characterizing the necessary and sufficient conditions for such a scheme to be secure. As a result, it has been impossible to dispel doubts (not always unfounded; see [PW91]) about the security of previously proposed schemes. Moreover, the lack of formality in proposed solutions has been an obstacle to careful discussion and comparative evaluation of the plausibility and practicality of their (sometimes nebulously defined) underlying assumptions, and the sufficiency of their claimed properties.

In this paper, we propose a formal model of a network based on a fairly natural set of assumptions, and give an explicit characterization of the barest essential properties of untraceable electronic cash in that model. We then present an extremely simple scheme which satisfies this set of properties. Thus, while the model and assumptions are naturally still open to improvement, and the characterization of the necessary security properties can arguably be strengthened, we believe that the simplicity and security of the proposed scheme make it genuinely practical under the right circumstances, and, moreover, that the formal model presented here represents an important (and too-long-delayed) foundational step

N. Koblitz (Ed.): Advances in Cryptology - CRYPTO '96, LNCS 1109, pp. 61-73, 1996.

in the resolution of the problem of realizing a practical, secure, untraceable electronic cash system.

1.1 The problem

The fundamental difficulty of anonymous electronic cash is simple to state: if the spender of an electronic "coin" is not identified in two successive transactions, then how is he or she to be prevented from acting as if the first transaction never occurred, and spending the same coin again? One proposed solution to this problem was presented in [CFN88], and was based on the premise that it would be sufficient for such "double spending" to be detected, and the spender identified, upon presentation of the same "electronic coin" twice to the central bank. This premise has also been used in a number of other schemes (see, for example, [Bra93, FY93, OO91]), all with the advantage that the bank need not be involved in each transaction. Practically speaking, however, this premise has enormous drawbacks: fraudulent transactions are detected only long after they have taken place, and if the perpetrator can be confident of not being brought to justice (either by being inaccessible, or by managing to use someone else's identity and cash), then he or she can double-spend at will.

However, if such fraudulent use of electronic cash is to be prevented, then some authority must somehow be involved in each transaction as it occurs, so as to be able to recognize and alert targets of double-spending. How, then, is anonymity to be preserved? One approach (see, for example, [EGY83]) is to rely on tamper-resistant hardware to force spenders to behave honestly (i.e., not to double-spend) even when they remain anonymous. Schemes based on this premise are, however, extremely brittle: if anyone ever succeeds in tampering with the hardware, then not only is that person capable of double-spending, but anyone, anywhere who obtains (purchases, perhaps) the information hidden in the hardware can spend arbitrarily high amounts with impunity. Current tampering resistance technology is far from being dependable enough to be trusted to thwart such an enormous risk. (An exception may be the use of such technology for low-value transactions exclusively, much the way coins are used today to represent small amounts of cash despite their being relatively easy to counterfeit.)

Another approach is to use cryptographic assumptions to preserve anonymity even in carefully monitored transactions. For example, under a particular very strong cryptographic assumption (namely, that RSA signature keys are secure against adaptive chosen message attacks), it is possible to construct protocols that create "blinded" cash—information which can be recognized later as valid (and previously unspent) cash, but cannot be connected with any particular run of the "cash creation" protocol. (See [Cha89a],[Cha89b].) However, a much simpler solution, requiring weaker cryptographic primitives, can be obtained by assuming instead the availability of *anonymous communication*. If parties can communicate with each other (and with the bank) without being identified, then a measure of anonymity can be provided without depriving the bank of the

knowledge necessary to prevent double-spending. Moreover, even if the assumption turns out to be false, then only the anonymity of the system is compromised, not its overall security.

This assumption of anonymous communication has actually been studied fairly extensively, with regard both to its implementation (for example, in [Cha81, Cha88a, RS93]) and its applications, in particular for secure election protocols (such as [Cha88b]), and for electronic cash ([Yac94, MN93]). In fact, upon reflection, it is clear that the ability to communicate anonymously is in some sense necessary *a priori* if anonymous cash transactions are to occur, since information about a party's communications will obviously reveal information about that party's business dealings. In practice, the anonymity of communication may perhaps be based on nothing more than confidence that the telephone company safeguards the confidentiality of its system; alternatively, parties may trust in one or more anonymous remailers (analogous to the "mixes" described in [Cha81]) to obscure their identities, or rely on an implementation of one of the other, more elaborate techniques from the aforementioned literature.

1.2 A solution

Suppose, then, that communications between parties are anonymous not only with respect to third parties, but also that the communicating parties are anonymous to each other. (In typical implementations, the latter condition is a natural consequence of the former, barring self-identification.) A simple (somewhat) anonymous electronic cash protocol in such a setting is as follows: a party can withdraw a coin (non-anonymously) by requesting that the bank associate a monetary value with $f(x)$, where x is a random value chosen by the party (and kept secret, even from the bank) and f is a one-way function. The bank complies by digitally signing a statement to that effect, thus "certifying" $f(x)$ as a valid coin. (The generally accepted criteria for a digital signature scheme to be secure can be found in [GMR88]; a construction meeting the definition based on any one-way function appears in [Rom90].)

At any time, a party can "exchange" a coin (anonymously) by supplying the bank with x and $f(y)$ for some randomly chosen y (kept secret from the bank); again, the bank simply certifies $f(y)$ as a valid coin, and keeps x as proof that $f(x)$ has already been "spent". (x can also be made public at that time, so that everyone can recognize $f(x)$ as a spent coin.) The new coin, like the old one, has an associated public value, $f(y)$, and a secret, y, possessed only by the coin's "owner". Actual spending of coins is a similar process: the spending party passes x to the receiving party, who verifies first that it has not previously been spent, then immediately exchanges it for a "fresh" coin with randomly generated secret y and corresponding "public" value $f(y)$. (Alternatively, the receiving party can first pass $f(y)$ to the spending party, who can then perform the exchange using x and $f(y)$, delivering the newly-certified $f(y)$ to the receiving party.) Unspent coins can also be deposited (non-anonymously) with the bank at any time.

In addition to requiring anonymous communication and bank intervention in every transaction, the scheme described above does not provide complete

anonymity. For example, a withdrawn coin can be associated with a chain of exchanges eventually leading to a deposited coin; the bank can conclude that a sequence of transactions consistent with the origin and final destination of the coin, and not longer than the number of exchanges in the sequence, must have occurred. On the other hand, the paper money currently in use is itself subject in principle to exactly the same tracing, by virtue of its use of serial numbers to certify validity. Moreover, as the number of transactions in the chain increases, the information provided by the existence of such a chain is in practice likely to become minuscule. Possessors of coins can also lengthen this chain themselves simply by executing repeated anonymous exchanges of their coins for new ones, thus increasing the bank's perceived limit on the number of transactions in which a given coin can have participated.

Parties are also protected from the bank's reneging on a (non-anonymously deposited) coin $f(x)$, in that the bank can be required to honor $f(x)$ unless it can present the secret x as proof that $f(x)$ has already been spent. (This feature is an important distinction between the scheme described here and the one presented in [MN93], in which the secret representing each coin is shared with the bank. Another is that in the scheme described here, the bank's only secret is its private signature key; all of its information about spent and unspent coins may be made public.) Of course, the bank could always renege on a coin during an anonymous exchange, by claiming upon receiving x that the coin has already been spent. However, the bank cannot possibly know who is being cheated by such a "dine and dash" ploy, and is therefore vulnerable to monitoring and public exposure. Moreover, this problem seems to be inevitable in any exchange of secrets (see [Cle86]), which an anonymous on-line electronic cash transaction, it appears, must necessarily be.

1.3 Security

One of the difficulties in analyzing the security of electronic cash schemes is that they are usually defined in terms of protocols involving pairs (or perhaps triples) of participants, even though their security properties must still hold in a full network of many parties engaging in complex interactions. In contrast, protocols for such tasks as secure multiparty computation ([Bea91, MR91]), untraceable communication ([RS93]) and authentication ([BR93]) have been analyzed using models that encompass the entire operation of a network, even when a protocol or subprotocol only involves a fixed subset of the parties. We will adopt this approach here, defining our electronic cash scheme in a full network model in which anonymous communication is possible.

Perhaps a more fundamental difficulty is that the goals of an electronic cash scheme are somewhat less clearly defined than those of such standard cryptographic primitives as encryption and digital signature. The way that cash is used in our society is both complex and limited by its various physical, legal and social properties; there is no reason to believe that it cannot be improved upon, in some ways at least, in converting it to an electronic medium. Our formalization of the desirable properties of an electronic cash scheme in the context of

a complete network model should therefore be taken as a preliminary attempt, and we encourage the many who have already contributed characterizations of desirable properties in various other models to consider recasting them using the network model approach, the better to analyze the success of proposed solutions in conforming to them.

2 The Model

2.1 The Anonymous Communication Network

We present here a formal model of a communication network in which it is assumed that parties can communicate anonymously. In the simplest such model, parties can send individual messages to each other anonymously. (The mechanism by which anonymous communication is made possible is not specified; rather, it is simply assumed as an abstract property of the network.) A stronger assumption is that parties receiving anonymous messages can also reply to them; an intermediate one is that one or more parties can (non-anonymously) broadcast messages efficiently (and thus reply to anonymous ones without jeopardizing that anonymity). Note that we do not assume the actual contents of communications to be at all private—only that they arrive unaltered, with their originators hidden, before eavesdroppers can preempt them. Moreover, parties may explicitly make information "public"; such information is not immediately available to every party, but is assumed to be reliably stored, and ultimately accessible to everyone.

Definition 1. A *communication protocol* is a family of vectors $(A_1, ..., A_n)$ of circuits (*parties*) each of which accepts, in addition to its normal inputs and outputs, special "work", "sending", "reply", "public" and "broadcast" outputs, and special "random", "work" and "receiving" inputs. The random inputs of all parties contain independent, unbiased random bits. When all parties have finished computation, they resume computing again, with their work outputs having been transferred to their "work" inputs; their computation is thus naturally divided into *rounds* consisting of the computation between these "restarts".

- In an *anonymous message protocol*, pairs of the form (i, m) appearing in a party's sending output at the end of a round result in the value (m) appearing in the receiving input of party A_i when computation is resumed at the beginning of the next round. (Multiple values "sent" to the same party appear in any order.)
- In addition, in an *anonymous exchange protocol*, pairs of the form (i, m) appearing in a party's reply output at the end of a round result in the value (k, m) appearing at the beginning of the next round in the receiving input of the party whose kth sending/reply output caused the ith value to appear in the party's receiving input.
- Finally, in an $\{A_{i_1}, ..., A_{i_k}\}$-*broadcast anonymous message (resp., exchange) protocol*, a value m appearing in the broadcast output of a member $A_j \in$

$\{A_{i_1}, ..., A_{i_k}\}$ at the end of a round results in the pair (j, m) appearing in the receiving output of every party at the beginning of the next round (in addition to all those values appearing in inputs in a simple anonymous message (resp., exchange) protocol).

Note that this definition assumes that reliable, synchronous communication is possible. While this simplifying assumption may be unrealistic, it is not actually exploited in the proposed protocol (nor is it clear that it is even possible to do so, given the definitions presented below). Rather, the assumption of synchrony serves to "discretize" time, abstracting out the issue of communications delays without preventing adversaries from taking advantage of them (since messages arriving during the same time period are queued in arbitrary order, as if any of them might have arrived first). Parties are also assumed, for simplicity's sake, to have arbitrarily large queues; the issue of the realistic message-processing constraints on parties is addressed instead by the efficiency requirement defined on electronic cash protocols below.

Definition 2. The *view* of a party A_i is the concatenation of the contents of all its inputs and outputs in every round, along with the (unlabeled) contents of the receiving inputs and public outputs of every other party in the network.

2.2 Electronic Cash Protocols

We model the exchange of electronic cash in a very simple way: parties in the network simply withdraw coins from the bank, or "pay" them to other parties, based on an (arbitrary) input, adjusting their output "wallet balances" upward or downward as they make or receive payments. Parties may also "deposit" coins; this action actually stands more generally for any "redemption" of a coin for some other form of value, much as a payment represents any reciprocal exchange of value. However, the "fairness" of these exchanges or payments is not addressed in the model; in practice, many different social, legal and economic means, as well as technical ones, may be involved in mediating fair exchanges between untrusting parties.

For simplicity, we consider all coins to have equal (unit) value, and assume that parties behave independently but synchronously; an input is given to some arbitrary party once per "cycle" (each cycle consisting of a fixed number of computation rounds sufficient for all parties to process a single input), commanding that party to spend (or withdraw, or deposit) a coin. ("Honest" parties' transactions are thus assumed to be strictly ordered in time.) The bank is also considered to be a party, accepting withdrawals and deposits and keeping track of every other party's current "balance". Other parties, in addition to their current "wallet balances" of stored coins, keep track of their "ledger balances" of coins withdrawn from and deposited with (i.e., redeemed by) the bank.

The crucial properties that should hold, both during the normal protocol and after this deposit process, whenever it is performed, are, informally:

1. **Correctness:** Parties following the protocol should be able to spend and receive electronic coins, as well as to withdraw coins from the bank and deposit them, and correctly to update their balances accordingly.
2. **Integrity:** Given an honest bank, parties who always follow the protocol should always know their correct ledger and wallet balances, and hence have no disputes with other parties that follow the protocol regarding their final balances. For example, they should be immune to other parties' attempts to lay claim to "their" coins, or to pass "forged" coins which the bank will not accept. Moreover, the bank should itself be immune to attempts to deposit forged coins which do not correspond to any originally withdrawn coin.
3. **Recoverability:** If the bank is permanently dishonest, there is little any party can do; the bank can simply refuse to honor any of its coins. However, it is assumed that parties are able to recognize such dishonest behavior in a bank, at least from the first failed deposit/redemption, and, moreover, that the bank can be forced to behave honestly once accused (by subjecting subsequent transactions to the scrutiny of the appropriate authorities). In this case, however, honest parties should be able to continue using the coins in their possession regardless of the state of the bank when its dishonesty was discovered. In other words, if a dishonest bank is replaced by an honest bank that knows only public information, then it can still guarantee the protocol's correctness and integrity.
4. **Anonymity:** Parties' spending should be untraceable, in the following (weak) sense: no party or coalition of parties should be able to obtain any more information about the spending of other parties than is given by their own spending/receiving information plus the beginning and end points of the path of each (distinguished) electronic coin.
5. **Efficiency:** We will require that each transaction require an amount of work per transaction (both computation and communication) for each party involved (spender, receiver and bank) which is at most polynomial in the security parameter and the logarithm of the number of parties in the network. (At least that much work is required simply to identify other parties uniquely.) We will not limit the amount of space required for the entire protocol (beyond the limits implied by the work constraints); in fact, almost every electronic cash scheme proposed so far in the literature suffers from the problem of requiring a large, publicly accessible database, replacing the "secure distributed database" implicit in physically circulated currencies such as coins or banknotes. (Protocols following the model of [EGY83] rely instead on "tamper-proof" hardware to allow the database to be distributed over all users.)

Other desirable properties have been mentioned in the literature, but we will concentrate on those listed here.

Definition 3. An anonymous message (resp. exchange, $\{B\}$-broadcast) *electronic cash protocol* is an anonymous message (resp. exchange, $\{B\}$-broadcast) protocol $(A_1, ..., A_n, B)$ which operates on the following inputs to parties:

1. Each party begins execution with input containing:
   - The number $n+1$ of parties in the network, and a label i (or B, representing the bank), distinct for each party, which represents the party's identity;
   - A security parameter m, input in unary;
   - A randomly chosen public-key/secret-key pair (p_i, s_i) for a secure digital signature system (see [GMR88],[Rom90]), with security parameter m (for each A_i as well as for B); and
   - The public key p_B of B (B itself receives the public keys p_i of every A_i).
2. In addition, at the beginning of every $k(m, \log n)$-round "cycle" starting with the second (for some fixed polynomial $k(m, \log n)$), some party other than B receives as "transaction" input the label of some party, to which it is to "pay" a coin (receiving the label B indicates that a coin is to be "deposited" at the bank; receiving the special label W indicates that a coin is to be "withdrawn" from the bank).
3. At the end of the round prior to the one where the $(r+1)$th input is to be received by some party (for $r \geq 0$), each party A_i outputs (as its "normal" output) a "wallet balance" value wal_i^r and a "ledger balance" value led_i^r, and B outputs a list of "bank balance" values $(bal_{j_1}^r, ..., bal_{j_k}^r)$ for all the values bal_j^r for which $bal_j^r \neq bal_j^{r-1}$.
4. The entire protocol executes for a number of rounds which is at most a fixed polynomial in m and $\log n$.

We now give definitions of correctness, security, anonymity and efficiency for electronic cash protocols in our model, following the criteria described informally above. The formal description is meant to be minimally restrictive; for example, honest parties' wallet balances are only required to be at least as great as they would be if all parties were honest, since "dishonest" parties may simply pass out their cash to others arbitrarily. Two of the key security requirements, listed below, are that the bank never accepts more coins than it issued, and that an honest party be able to deposit successfully all the coins it has been convinced are valid.

Definition 4. An electronic cash protocol $(A_1, ..., A_n, B)$ is *correct* if any variant $(A'_1, ..., A'_n, B)$ in which $A'_i = A_i$ and $A'_j = A_j$ has the following properties with probability $1 - m^{-\omega(1)}$ (over the choices of random bits used by all the parties):

1. if the rth transaction input is to A'_i, and is j (representing "payment to j"), and $wal_i^{r-1} > 0$, then $wal_i^r \geq wal_i^{r-1} - 1$, and $wal_j^r \geq wal_j^{r-1} + 1$.
2. if the rth transaction input is to A'_i, and is B (representing "deposit"), and $wal_i^{r-1} > 0$, then $wal_i^r \geq wal_i^{r-1} - 1$, $bal_i^r = bal_i^{r-1} + 1$, and $led_i^r = led_i^{r-1} + 1$.
3. if the rth transaction input is to A'_i, and is W (representing "withdrawal"), then $wal_i^r \geq wal_i^{r-1} + 1$, $bal_i^r = bal_i^{r-1} - 1$, and $led_i^r = led_i^{r-1} - 1$
4. Otherwise, $wal_i^r \geq wal_i^{r-1}$, $bal_i^r = bal_i^{r-1}$ and $led_i^r = led_i^{r-1}$.

Definition 5. An electronic cash protocol $(A_1, ..., A_n, B)$ is *secure* if any variant $(A'_1, ..., A'_n, B)$, in which each A_i is replaced with A'_i, has the following properties with probability $1 - m^{-\omega(1)}$ (over the choices of random bits used by all the parties):

1. $bal_i^0 = 0$, for $i = 1, ..., n$.
2. if $A'_i = A_i$, then $wal_i^0 = led_i^0 = 0$ and $bal_i^r = led_i^r$ for each r.
3. $\sum_i bal_i^r \leq 0$ for each r.

Note that under the above two definitions, an honest party can successfully deposit every coin it receives, without causing more coins to be deposited than were issued. Hence a correct, secure anonymous cash protocol also prevents counterfeiting.

Definition 6. An electronic cash protocol $(A_1, ..., A_n, B)$ is *recoverable* if for any round r of any run of any variant $(A'_1, ..., A'_n, B')$ there exists a set $\{bal_i^r\}$ of finite values (dependent only on r and i) such that the protocol is still correct and secure even if, following round r, the following occurs:

- B' is replaced with B;
- the contents of B''s work tape are erased and replaced with the string, "recover", followed by $\{bal_i^r\}$, then the contents of all parties' public outputs, and a new randomly chosen public-key/secret-key pair (p_B^*, s_B^*) for a secure digital signature system with security parameter m; and
- ("recover", p_B^*) is placed in the input of all other parties.

Definition 7. An electronic cash protocol $(A_1, ..., A_n, B)$ is *anonymous* if for any variant $(A'_1, ..., A'_n, B')$, in which each A_i is replaced with A'_i, and B with B', and for any set $S = \{i_1, ..., i_j\}$ for which $A'_{i_t} = A_{i_t}$ in S, there exists an assignment of labels to the transaction inputs (corresponding to "chains" consistent with a coin being passed from one party to the next) with the following properties:

1. There are exactly as many labels as there are transaction inputs with value B.
2. If a transaction input with label ℓ is j, then the next transaction input with label ℓ is input to A_j.
3. The distribution on the concatenation of the views of all the parties outside S remains unchanged when the inputs to the parties are changed as follows: for any portion of one of the chains described above in which all the parties are in S, the parties in the chain are replaced with any arbitrary sequence of members of S.

Definition 8. An electronic cash protocol $(A_1, ..., A_n, B)$ is *efficient* if the following properties hold:

1. All parties are uniform circuits of size polynomial in m and n.
2. The total number of computations executed by A_i or B in each $k(m, \log n)$-round cycle is polynomial in m, $\log n$, and the total length of the inputs it received in that cycle.

3 The Protocol

3.1 Description

We present here the proposed protocol, which follows the outline given in the introduction, and is expressed below as the description of an arbitrary party's behavior in response to its possible inputs. At first, all parties are assumed to be identical; they are eventually distinguished only by their initial inputs. The protocol assumes that the bank can either reply to anonymous messages, or broadcast the response. The protocol's security depends on the assumption that the digital signature key pairs used in it are generated by a secure digital signature scheme $\mathcal{S}$, and that the globally known polynomial-time-computable function f is one-way. Parties also use f to extend the reply assumption into arbitrarily long conversations as follows: a message is accompanied by a value $f(h)$, and the value h is supplied in the subsequent message in the conversation (along with an $f(h')$, if necessary, for a following message), to convince the recipient that the two messages have the same origin.

Definition 9. The electronic cash protocol $\prod = (A_1, ..., A_n, B)$, where all parties are identical, is an exchange/$\{B\}$-broadcast protocol described by the following behaviors:

- **Given input** $(n, i, m, (p_i, s_i), p_B)$: A party initializes itself by setting its own identity (to A_i), security parameter, network size, and public and secret signature keys, as well as the bank's public signature key, according to the input. It also sets variables wal_i^0 and led_i^0 to 0, and initializes an empty "coin list".
- **Given input** $(n, i, m, (p_B, s_B), (p_1, ..., p_n))$: A party initializes itself by setting its own identity (to B), security parameter, network size, and public and secret signature keys according to the input, as well as organizing all the other parties' public signature keys p_i and balances bal_i into efficient data structures, and initializing bal_i^0 to 0 (for $i = 1, ..., n$). B also initializes another (empty) efficient data structure to act as "spent coin list". Finally, the party outputs $p_1, ..., p_n, p_B$ as a public output.
- **Given input** W (signifying "withdrawal"), a party A_i (other than B) chooses a random m-bit string x, computes $f(x)$, constructs the message $\mu = f(x)$, uses s_i to generate signature σ_μ of μ, and outputs $(B, (i, \mu, \sigma_\mu))$ as sending output. If the next round's receiving input includes a message of the form, $\mu' = (\mu, \sigma)$, where σ is a signature of $\mu = f(x)$ verifiable using p_B, then A_i stores (x, μ') in its "coin list", increases wal_i by 1, decreases led_i by 1, and outputs public output μ'. Otherwise, A_i outputs "fraud".
- **Given input** $j \in \{1., , , .n\}$ (signifying "payment to j"), a party A_i (other than B) chooses the oldest coin (x, μ') in its coin list, removes it, and outputs $(j, (x, \mu'))$ as sending output, reducing wal_i by 1.
- **Given input** B (signifying "deposit"), a party A_i (other than B) chooses the oldest coin (x, μ') in its coin list, removes it, constructs a message of the form, $(B, (i, x, \mu'))$, and outputs this message as sending output, subtracting

1 from wal_i and adding 1 to led_i. If the next round's receiving input does not include a message of the form $((i, x, \mu), \sigma)$, where σ is a signature of (i, x, μ) verifiable using p_B, then A_i outputs "fraud". Otherwise, it outputs (x, μ) as public output.

- **Given receiving input** (i, μ, σ_μ) (signifying "withdrawal by i"), party B verifies σ_μ using p_i, and if valid, decreases bal_i by 1, and generates signature σ of μ using s_B, producing reply/broadcast and public outputs (μ, σ).
- **Given receiving input** (i, x, μ'), where $\mu' = (\mu, \sigma)$ (signifying "deposit by i"), party B verifies that σ is its own valid signature of μ and that $f(x) = \mu$, and if the verification is successful, increases bal_i by 1, adding (x, μ) to the "spent coin list" (and to its public output). The next round, B signs (i, x, μ) and sends a message of the form $(i, ((i, x, \mu), \sigma))$.
- **Given receiving input** $(x, (f(x), \sigma))$ (signifying "payment from another party"), party A_i (other than B) outputs $(B, (f(h), f(x), \sigma))$ (for a randomly chosen m-bit h) as sending output. If the next round's receiving input includes a message of the form, $(f(h), \text{"yes"})$, then A_i chooses a random m-bit string y and outputs $(B, (h, x, f(x), f(y))$ as sending output. If the next round's receiving input includes a message of the form, $\mu' = (\mu, \sigma)$, where σ is a signature of $\mu = f(y)$ verifiable using p_B, then A_i stores (y, μ') in its "coin list", outputs μ' as a public output, and increases wal_i by 1. Otherwise, A_i outputs "fraud". If instead of $(f(h), \text{"yes"})$, a response of $(f(h), \text{"collision"})$ is received as receiving input, then A_i recomputes $(B, (f(h), f(x), \sigma))$ (the same way as before, but with a new, randomly chosen h) and outputs it again as sending output, treating subsequent receiving inputs exactly as though it had been sent for the first time.
- **Given receiving input** (τ, μ, σ), where σ is a signature of μ verifiable using p_B (signifying "coin exchange"), B checks whether μ is absent from its "spent coin list", and outputs the answer ("yes" or "no"), preceded by τ, as reply/broadcast output. (If more than one such input is received in the same round containing the same unspent μ value, then an answer of "collision" replaces "yes" in the response outputs to all but a randomly chosen one of the receiving inputs with a unique τ value for that round.) If the next round's receiving input includes a message of the form, (t, x, μ, υ), where $f(t) = \tau$ and $f(x) = \mu$, then B outputs (υ, σ) (where σ is a signature of υ generated using s_B) as reply/broadcast and public output, and adds (x, μ) to the "spent coin list" (as well as to its public output).
- **Given input** ("recover", p_B^*), a party A_i thereafter verifies all signatures against both p_B and p_B^*, and checks all messages that can be verified using p_B to see if their contents were in some party's public output before the "recover" input was received.
- **Given input** ("recover", $\{bal_i\}$, $\{pub_i^\tau\}$, (p_B^*, s_B^*)), party B sets each A_i's balance to bal_i, recovers its spent coin list and all parties' public signature keys from its old public output (including its own old one, p_B), and thereafter checks all messages that can be verified using p_B to see if they were in some party's public output before the "recover" input was received.

Theorem 10. *If f is a one-way function, and $\mathcal{S}$ is a digital signature scheme secure against adaptive chosen message attack, then Π is a correct, secure, recoverable, anonymous, efficient electronic cash protocol.*

Proof: The proof, while involved, follows fairly straightforwardly from the definitions. Details will appear in the final paper.

4 Conclusions and Open Problems

The basic protocol described here can in practice be enhanced in a number of ways. The addition of expiry dates to electronic coins, for example, has the benefit of reducing the size of the list of spent coins that must be maintained at any one time, since spent coins that have also expired can be removed from the list. Non-anonymously withdrawn coins with expiry dates can also be replaced if lost, once they have expired without being used. Another enhancement is to allow payers to use a one-way function to embed information (such as transaction information or even an identity) into the preimages used to construct their electronic coins, to demonstrate the origin, and to provide evidence of the intended purpose, of a particular coin at some later time, if necessary. The introduction of public key cryptography (beyond digital signature) to the protocol allows a number of further variations, such as the passing of electronic coins to payees in a manner that forces the payee to relay some additional information to the bank (such as an intended transaction description) along with the coin's associated preimage.

This protocol also raises a number of as-yet-unanswered questions about the practical implementation of electronic cash. Where does the trade-off lie between the security of on-line communication and its cost? How are transactions that are too small to be worth this trade-off to be performed off-line in a sufficiently secure manner? How is the "dine-and-dash" problem—or, more generally, the "fair exchange mediation" problem—to be addressed in an electronic world in which the physical and cultural constraints on in-person business are absent? (The "ripping coins" solution proposed by Jakobsson in [Jak95], for example, fails in the case where one party—the bank—profits from the pre-spending destruction of coins.) What other practical considerations lie in the way of the implementation of secure electronic cash? And finally, what, exactly, do we mean when we say that we would like electronic cash to be "secure"?

5 Acknowledgements

Many thanks to Josh Benaloh, Terence Spies and Yacov Yacobi for their valuable comments and suggestions.

References

[Bea91] D. Beaver, *Foundations of Secure Interactive Computing*, Proc. CRYPTO '91, pp. 377-391.

[Bra93] S. Brands, *Untraceable Off-Line Cash in Wallet with Observers*, Proc. CRYPTO 93, pp. 302-318.

[BR93] M. Bellare and P. Rogaway, *Entity Authentication and Key Distribution*, Proc. CRYPTO '93, pp. 232-249.

[Cha81] D. Chaum, *Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms*, CACM vol. 24, no. 2 (1981), p. 84-88.

[Cha88a] D. Chaum, *The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability*, Journal of Cryptology vol. 1, no. 1 (1988), pp. 65-75.

[Cha88b] D. Chaum, *Elections with Unconditionally Secret Ballots and Disruption Equivalent to Breaking RSA*, Proc. EUROCRYPT '88, pp. 177-182.

[Cha89a] D. Chaum, *Privacy Protected Payments–Unconditional Payer and/or Payee Untraceability*, SMART CARD 2000: The Future of IC Cards–Proc. IFIP WG 11.6 Intl Conf., North-Holland (1989), pp. 69-93.

[Cha89b] D. Chaum, On-line Cash Checks, Proc. EUROCRYPT 89, pp. 288-293.

[Cle86] R. Cleve, *Limits on the Security of Coin Flips When Half the Processors are Faulty*, Proc. 18th ACM Symposium on Theory of Computing (1986), pp. 364-369.

[CFN88] D. Chaum, A. Fiat and M. Naor, *Untraceable Electronic Cash*, Proc. CRYPTO '88, pp. 319-327.

[EGY83] S. Even, O. Goldreich and Y. Yacobi, Electronic Wallet, Proc. CRYPTO 83, pp. 383-386.

[FY93] M. Franklin and M. Yung, *Secure and Efficient Off-Line Digital Money*, Proc. 20th Intl Colloquium on Automata, Languages and Programming (1993), pp. 265-276.

[GMR88] S. Goldwasser, S. Micali and R. Rivest, *A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks*, SIAM J. Computing 17(2) (1988), pp. 281-301.

[Jak95] M. Jakobsson, *Ripping Coins for a Fair Exchange*, Proc. EUROCRYPT '95, pp. 220-230.

[MN93] G. Medvinsky and B.C. Neuman, *Netcash: a Design for Practical Electronic Currency on the Internet*, Proc. 1st ACM Conference on Computer and Communications Security (1993).

[MR91] S. Micali and P. Rogaway, *Secure Computation*, Proc. CRYPTO '91, pp. 392-404.

[OO91] T. Okamoto and K. Ohta, *Universal Electronic Cash*, Proc. CRYPTO 91, pp. 324-337.

[PW91] B. Pfitzmann and M. Waidner, *How to Break and Repair a "Provably Secure" Untraceable Payment System*, Proc. CRYPTO '91, pp. 338-350.

[Rom90] J. Rompel, *One-Way Functions Are Necessary and Sufficient for Secure Signatures*, Proc. 31st IEEE Symp. on Foundations of Computer Science (1990), pp. 387-394.

[RS93] C. Rackoff and D. Simon, *Cryptographic Defense Against Traffic Analysis*, Proc. 25th ACM Symp. on the Theory of Computation (1993), pp. 672-681.

[Yac94] Y. Yacobi, *Efficient Electronic Money*, Proc. ASIACRYPT 94.