# Towards Provably Secure Efficient Electronic Cash
(extended abstract)

Matthew Franklin*
Columbia University
Computer Science Department,
New York, NY 10027

Moti Yung
IBM Research Division,
T.J. Watson Center,
Yorktown, NY 10598

April 24, 1992

**Abstract**

An "electronic coin scheme" as defined by Chaum, Fiat, and Naor [5] is a collection of protocols to achieve untraceable, unforgeable coins with offline purchasing; this is the minimum set of properties to make electronic money useful. We give a new electronic coin scheme that is simple and practical. Withdrawal requires only two rounds of interaction, while purchase and deposit are non-interactive; all previous efficient cash schemes require interaction (cut-and-choose) for purchases. Moreover, messages during purchase and deposit contain only a few encrypted values, independent of the tolerable probability of cheating. We present a security model for electronic coins, and prove the security of our scheme relative to certain specific cryptographic assumptions (hardness of Discrete Log and possibility of secure blind signature).

**TR CUCS-018-92**

*Partially supported by an AT&T Bell Laboratories Scholarship

# 1 Introduction

Six desirable properties of electronic money are stated by Okamoto and Ohta [15]: (a) independent of physical requirements, (b) unforgeable and uncopyable cash, (c) untraceable purchases, (d) off-line payment between vendor and consumer, (e) cash transferable from one consumer to another, and (f) cash subdividable by consumer. An "electronic coin scheme" is, informally stated, a collection of protocols that achieve the first four of these properties; this is the minimum set of properties to make electronic money useful. Chaum, Fiat, and Naor [5] define and present an electronic coin scheme whose security was left as an open problem. In this paper, we present an alternative scheme to achieve these four properties. Our scheme is simple, and its security, in a sense to be formally defined, can be proven relative to certain specific cryptographic assumptions.

We emphasize that our scheme is practical. The purchase protocol is non-interactive (a single message sent from customer to vendor), as is the deposit protocols (a single message sent from vendor to bank). We know of no other practical coin scheme for which purchase is non-interactive[1]. Moreover, message sizes during purchase and deposit are independent of the size of the "confidence parameter" $k$ to achieve a probability $\frac{1}{k}$ of undetected cheating (amplifiable to inverse exponential in $k$ by expanding the scheme $k$-wise). We know of no other coin scheme with this property either. The withdrawal protocol consists of only two rounds of messages between a customer and the bank. Message sizes are reasonable; if discrete log modulo a 150 digit prime is hard, and if forging 150 digit hashed RSA signatures is hard, then coin size is only 750 decimal digits during purchase and deposit (and a factor of $k$ larger during withdrawal).

We suggest a precise model for electronic coins, together with a notion of security against an attack by $t$ parties ("$t$-security"). Relative to specific cryptographic assumptions, we prove that our coin scheme is 1-secure, i.e., that no single consumer, vendor, or bank can cheat to either their advantage or others' disadvantage. The assumptions we require are that the problem of extracting discrete logarithms is intractable, and that secure blind signature schemes exist (e.g., that RSA signature is strongly unforgeable if a good hash function is applied to the message before signing). Despite weaknesses in the power of the scheme, the level of security, and the generality of the underlying intractability assumptions, we feel that, besides practicality, the presentation of our formal model and proof of security are important contributions to the overall development and maturity of the area of electronic money, and will help to steer its future progress.

Electronic money schemes are first considered by Chaum [2] (although Even, Goldreich, and Yacobi [10] attack similar problems but using physical assumptions). In particular, Chaum [2] introduced the important idea of blinded signatures, whereby one party can have a message signed without revealing its contents. Chaum, Fiat, and Naor [5] define offline electronic coin schemes, and propose an implementation; efficiency improvements are

[1] A theoretical, and impractical, offline electronic coin scheme – with non-interactive purchase – is given by De Santis and Persiano [9] as an application of non-interactive zero-knowledge proofs of knowledge in the shared random string model.

given by Chaum *et al* [3]. Okamoto and Ohta [14] [15] present complex offline electronic money schemes with more features than our basic coin scheme (i.e., transferability [14] and divisibility [15]). Pfitzmann and Waidner [16], building on work of Damgård [7], present an offline payment scheme that adapts the basic mechanism of Chaum, Fiat, and Naor to a setting that blends practical protocols with general purpose secure multiparty computation protocols.

In Section 2, a formal model for electronic coin schemes is given. Intractability assumptions are defined in Section 3. In Section 4, we present our electronic coin scheme, a proof of its security, and some practical implications. In Section 5, we consider the consequences of a stronger model of security, and discuss open problems.

# 2 Electronic Coin Model

There is a bank $B$, a collection of vendors $\{V_i\}$, and a collection of customers $\{C_i\}$, all of whom are assumed to be communicating Turing Machines. Each machine has a private input tape, a private output tape, and a private work tape. Each pair of machines share a pair of public communication tapes, one for communication in each direction. The bank has a public encryption key $[e_B, N_B]$ known to all parties and private decryption key $d_B$ known only to itself[2].

An electronic coin scheme consists of three protocols: a withdrawal protocol between $B$ and any $C_i$; a purchase protocol between any $C_i$ and any $V_j$; and a deposit protocol between any $V_j$ and $B$. The withdrawal protocol begins with $C_i$ having $[e_B, N_B]$ on its input tape, and $B$ having $[e_B, d_B, N_B]$ on its input tape, and ends with $B$ having the identity of $C_i$ on its private output tape, and $C_i$ having a "coin" $c$ on its private output tape. The purchase protocol begins with $C_i$ having a "coin" $c$ and $[e_B, N_B]$ on its input tape and with $V_j$ having $[e_B, N_B]$ on its input tape, and ends with $V_j$ having a "spent coin" $c'$ on its private output tape. The deposit protocol begins with $V_j$ having a spent coin $c'$ and $[e_B, N_B]$ on its input tape and with $B$ having $[e_B, d_B, N_B]$ on its input tape, and ends with $B$ having the "deposited coin" $c''$ and the identity of $V_j$ on its private output tape.

We will say that a collection of protocols is "$t$-secure" if the scheme is safe against cheating by any $t$ parties (where each party is either the bank or a vendor or a customer). Stated informally, cheating can occur in several basic ways. The identity of an honest consumer can be extracted from a spent or deposited coin that has not been reused. More coins can be deposited into the bank than were withdrawn (either through forgery or undetected reuse). An honestly withdrawn coin can be improperly rejected during a purchase by a cheating vendor. An honestly spent coin can be improperly rejected during a deposit by a cheating bank. An honest consumer can withdraw a bad coin from a cheating bank. An honest consumer can be improperly framed as a coin reuser.

[2] It is possible to define the model more generally without assuming public-key encryption; for simplicity, we include it in the basic model.

We formalize the definition of a "1-secure" electronic coin scheme in the following requirements. Note that these requirements eliminate the types of cheating informally described in the preceding paragraph, when only a single party can behave dishonestly. They formalize the first four desirable properties noted by Okamoto and Ohta, i.e., (a) independence from physical requirements; (b) unforgeable and uncopyable cash; (c) untraceable purchases; and (d) off-line payment between vendor and customer.

1. Cheating by a Customer

   (a) (Forgery) Suppose that $C_i$ participates in polynomially many withdrawal protocols with $B$, and then $C_i$ can easily compute a $c$ which differs from the output coins from these withdrawals. Then if $C_i$ participates in a purchase protocol with any $V_j$ using $c$ on its input tape, $V_j$ will reject the purchase with high probability.

   (b) (Reuse) Suppose that $c$ is the output of a withdrawal protocol between $C_i$ and $B$. Further suppose that $C_i$ participates in two purchase protocols with the same coin $c$ on its input tape, resulting in spent coins $c_1'$ and $c_2'$ on the private output tapes of the vendors (possibly the same vendor), and that these spent coins are used in deposit protocols resulting in deposited coins $c_1''$ and $c_2''$. Then the fact that $c$ was reused should be easily computable with high probability from $c_1''$ and $c_2''$, and also the identity of $C_i$ should be easily computable with high probability from $c_1''$ and $c_2''$.

2. Cheating by the Bank

   (a) (Framed Reuse) Suppose that a cheating $B$ claims that $C_i$ reused a coin, basing the claim on deposited coins $c_1''$ and $c_2''$ (either real or constructed). Then this cheating will be apparent to a polynomial-time external judge who receives $c_1''$, $c_2''$, $[e_B, N_B]$, and the history of all purchase protocols involving the relevant (one or two) vendors.

   (b) (Rejectable Withdrawal) Suppose that $C_i$ and $B$ participate in a withdrawal protocol, and a cheating $B$ causes the protocol to end with $c^*$ on the private output tape of $C_i$, where an honest $V_j$ will reject a purchase protocol that began with $c^*$ on the input tape of $C_i$. Then $C_i$ will detect this cheating with high probability during the withdrawal protocol. Moreover, this cheating will be apparent to a polynomial-time external judge who receives $[e_B, N_B]$ and a transcript of the withdrawal protocol as input.

   (c) (Revealed Identity) Suppose that $C_i$ and $B$ participate in a withdrawal protocol resulting in output coin $c$, and then $C_i$ and $V_j$ participate in a purchase protocol resulting in spent coin $c'$, and then $V_j$ and $B$ participate in a deposit protocol resulting in deposited coin $c''$. Then the identity of $C_i$ should be hard to compute from the view of $B$ of the deposit protocol together with the view of $B$ of all withdrawal protocols.

   (d) (Rejected Deposit) Suppose that $C_i$ and $B$ participate in a withdrawal protocol resulting in output coin $c$, and then $C_i$ and $V_j$ participate in a purchase protocol using coin $c$ and resulting in spent coin $c'$, and then $V_j$ and $B$ participate in a deposit protocol using spent coin $c'$. Further suppose that a cheating $B$ improperly rejects the deposit. Then this cheating will be apparent to a polynomial-time external judge who receives $[e_B, N_B]$ and a transcript of the deposit protocol as input.

3. Cheating by a Vendor

   (a) (Forgery) Suppose that $V_j$ participates in polynomially many purchase protocols, and then can easily compute a $c'$ which differs from the spent coins from these purchases. Then if $V_j$ participates in a deposit protocol using $c'$ on its input tape, $B$ will reject the purchase with high probability.

   (b) (Reuse) Suppose that $V_j$ participates in more than one deposit protocol with the same spent coin $c'$ on its input tape, resulting in deposited coins $c''_1$ and $c''_2$. Then the fact that $c'$ was reused should be easily computable with high probability from $c''_1$ and $c''_2$.

   (c) (Revealed Identity) Suppose that $C_i$ participates in a withdrawal protocol with $B$ resulting in coin $c$ on the output tape of $C_i$, and then $C_i$ participates in a purchase protocol with $V_j$ resulting in spent coin $c'$ on the output tape of $V_j$, and then $V_j$ and $B$ participate in a deposit protocol using spent coin $c'$. Then the identity of $C_i$ should be hard to compute from the view of $V_j$ of the purchase and deposit protocols.

   (d) (Rejected Purchase) Suppose that $C_i$ and $B$ participate in a withdrawal protocol resulting in output coin $c$, and then $C_i$ and $V_j$ participate in a purchase protocol using coin $c$. Further suppose that a cheating $V$ improperly rejects the purchase. Then this cheating will be apparent to a polynomial-time external judge who receives $[e_B, N_B]$ and a transcript of the purchase protocol as input.

We note that requirements 2c and 3c are not as strong as one might like. They do not exclude the possibility that some information about the identity of the consumer can be extracted from a spent or deposited coin (e.g., excluding half the possible candidates from consideration). They also do not exclude the possibility that two or more purchases can be linked to the same anonymous consumer. When 2c and 3c exclude any information about the purchaser or purchase linkage being revealed, we call the resulting coin scheme "strongly 1-secure" (or "strongly $t$-secure"). We will only prove that our coin scheme is 1-secure, although we will later discuss strong 1-security as well.

# 3 Intractability Assumptions

In this section, we give the intractability assumptions that underlie the proof of 1-security for our electronic coin scheme. The first one is that the discrete logarithm problem is hard. This is a standard cryptographic assumption first used by Diffie and Hellman [8]. Anonymity of purchases in our scheme depends on this assumption.

**Discrete Log Assumption**: Let $p$ be a prime, let $g$ be a generator of $Z_p^*$, and let $a$ be an integer between 0 and $p-1$. Define $DLP_{p,g}(a)$ to be $i$ such that $g^i = a \bmod p$, $0 \le i < p$. Then no probabilistic polynomial-time Turing Machine can, on input $p$, $g$, and $a$, output $DLP_{p,g}(a)$ non-negligibly[3] better than random guessing.

[3] A quantity is "negligible" if it is smaller than the reciprocal of any polynomial of (sufficiently large) relevant parameters

The second intractability assumption, on which rests the proof of unforgeability for our coin scheme, relates to the difficulty of forging RSA signatures [19]. Our assumption is in terms of a very general type of attack on any signature scheme: existential forgery under an adaptive chosen plaintext attack (defined by Goldwasser, Micali, and Rivest [11]). A signature scheme fails this attack if a polynomial-time adversary, after being allowed to see the signature of polynomially many messages of its choosing, can construct the signature of any new message.

**RSA Signature Assumption**: Let $N$ be an RSA modulus, with public encryption key $e$ and private decryption key $d$. Then, for every $D$, $|D| \geq N$, there exists a function $h : D \rightarrow Z_N^*$ such that the signature scheme $\sigma(x) = h(x)^d \bmod N$ is existentially unforgeable under an adaptive chosen plaintext attack.

Note that RSA signatures by themselves do not withstand this type of attack. First, since the encryption key is public, it is always possible to forge RSA signatures of "random" messages (i.e., $x$ is the signature of its encryption $x^e \bmod N$). Second, due to the multiplicative homomorphic property of RSA encryption and decryption, the signature of any combination of multiplications and inversions of messages can be forged whenever the signatures of the individual messages are known.

However, if a good one-way hash function is used to scramble each message before raising to the RSA decryption key, then these types of forgeries are foiled; this idea is used by Chaum and Evertse [4] and discussed in detail by Damgård [6]. In particular, the function $h$ given by the Assumption affects the message space to prevent forgery based on the multiplicative homomorphism, while preserving the RSA homomorphism to allow blinded signatures[4]. Several secure hash functions have been designed specifically to thwart these signature attacks (as well as to reduce the size of the signed message). In a later subsection, we mention some candidate hash functions for implementing our coin scheme.

# 4 An Efficient 1-Secure Electronic Coin Scheme

In this section, we present protocols for an electronic coin scheme, prove its 1-security relative to cryptographic assumptions, and then discuss some aspects of the proof and practical implications of the scheme. Informally, every coin hides a line whose coefficients encode the identity of its withdrawer. When a coin is spent, the vendor is given a (guaranteed) point on the line. If a coin is spent twice, then two points on the line are revealed, from which the identity of the reuser can be extracted. We note that our scheme allows offline spending, and in fact the purchase protocol is non-interactive (a single message from the consumer to the vendor). The deposit protocol is also non-interactive (a single message from the vendor to the bank). Only the withdrawal protocol is interactive, and it is short (two rounds).

[4]The RSA Signature Assumption could be replaced by the more general assumption that there exists *some* blind signature scheme existentially unforgeable under an adaptive chosen plaintext attack; we forgo this generalization to simplify the presentation of our scheme.

Withdrawal

1. $C \rightarrow B$: $[r_1^{e_B}(h(g^{a_{11}} \bmod p || g^{a_{01}} \bmod p)) \bmod N_B, \cdots, r_k^{e_B}(h(g^{a_{1k}} \bmod p || g^{a_{0k}} \bmod p)) \bmod N_B]$, where each $r_i \in_R Z^*_{N_B}$.
2. $B \rightarrow C$: Challenge $j$.
3. $C \rightarrow B$: $[r_i, a_{1i}, a_{0i}]$ for all $i \neq j$, $1 \leq i \leq k$. $B$ verifies that the received information is consistent, that $a_{1i} < q$, and that $a_{1i}a_{0i} = pad_{|p|}(id_C) \bmod p-1$ for all $i \neq j$, $1 \leq i \leq k$.
4. $B \rightarrow C$: $r_j(h(g^{a_{1j}} \bmod p || g^{a_{0j}} \bmod p))^{d_B} \bmod N_B$. $C$ extracts the "coin" $c = [g^{a_{0j}} \bmod p, g^{a_{1j}} \bmod p, (h(g^{a_{1j}} \bmod p || g^{a_{0j}} \bmod p))^{d_B} \bmod N_B]$.

Purchase

1. $C \rightarrow V$: $c' = [g^{a_{1j}} \bmod p, g^{a_{0j}} \bmod p, (h(g^{a_{1j}} \bmod p || g^{a_{0j}} \bmod p))^{d_B} \bmod N_B, x = (id_V || time), y = a_{1j}x + a_{0j} \bmod p-1]$. $V$ accepts the coin if signature is correct, $x$ is of the correct form and not a repeat, and $g^y = (g^{a_{1j}} \bmod p)^x (g^{a_{0j}} \bmod p) \bmod p)$.

Deposit

1. $V \rightarrow B$: $c'' = [id_V, g^{a_{0j}} \bmod p, g^{a_{1j}} \bmod p, (h(g^{a_{1j}} \bmod p || g^{a_{0j}} \bmod p))^{d_B} \bmod N_B, x, y]$. $B$ accepts the deposit if the same conditions hold as for $V$ in the purchase protocol.

Figure 1: 1-Secure Electronic Coin Scheme

## 4.1 Description of the Protocols

The electronic coin scheme is detailed in Figure 1, and summarized below.

We let $|x|$ denote the length, in binary, of $x$. Let $N_B$ be the bank's RSA modulus, with public encryption key $e_B$ and private decryption key $d_B$. Let $p$ be a prime such that $p-1$ has one large factor $q$. Let $h : \{0,1\}^{2|p|} \rightarrow \{0,1\}^{|N_B|}$ satisfy the requirements of the RSA Signature Assumption. Let $id_C$ be the identity of consumer $C$, where $|id_C| = m \leq |p|$, and assume that there are exactly $2^m$ consumers (so that each id is possible). Let $id_V$ be the identity of vendor $V$, $|id_V| + |time| = |p|$, where $time$ is a standard encoding of a moment in time. Let $k$ be a security parameter. The concatenation of strings $a$ and $b$ is represented $a||b$, and $pad_l(x)$ represents an otherwise random string of length $l$ that encodes the shorter string $x$ in some fashion (e.g., high order bits).

The withdrawal begins with the customer sending $k$ encrypted and blinded lines to the bank. The product of the two coefficients of each line should give the customer's identity. The bank challenges all but one of the lines, and the customer responds by decrypting and unblinding the challenged lines to prove that they actually encode its identity. The bank signs the remaining encrypted and blinded line, and the customer unblinds it to produce his coin.

To make a purchase, the customer presents the coin (a signed encrypted line), together with one point on the corresponding line. The $x$-coordinate of the point is a simple predetermined function of the vendor's identity and the time of purchase. The line has been encrypted in such a way that the vendor can verify that the point is indeed on the line.

To make a deposit, the vendor passes on to the bank what was received from the customer during the purchase. The bank can also verify that the point is on the encrypted line.

## 4.2 Proof of 1-Security

**Theorem 1** *There is a 1-secure electronic coin scheme, under the Discrete Log Assumption and the RSA Signature Assumption.*

**Proof :** (sketch) We show that the protocols from the last subsection suffice.

1. Cheating by a Customer

    (a) Suppose that $C$ participates in polynomially many withdrawal protocols with $B$, resulting in coins $c_1, \cdots, c_{poly(n)}$, where $c_i = [z_i, z_i', \sigma_i]$. Suppose $C$ can then compute a coin $c = [z, z', \sigma]$ which differs from these withdrawn coins. To convince $V$, $[h(z||z'), \sigma]$ must be a valid ciphertext-plaintext pair. This contradicts the RSA Signature Assumption (i.e., the view of $C$ after the withdrawal protocols can be simulated by a probabilistic polynomial-time machine able to see $h(x)^{d_b} \bmod N_B$ for polynomially many values of $x$ of its choosing).

    (b) Suppose that $C$ reuses the coin $c = [g^{a_{0j}} \bmod p, g^{a_{1j}} \bmod p, (h(g^{a_{1j}} \bmod p||g^{a_{0j}} \bmod p))^{d_B} \bmod N_B]$. Then the deposited coins are of the form $c_1'' = [g^{a_{0j}} \bmod p, g^{a_{1j}} \bmod p, (h(g^{a_{1j}} \bmod p||g^{a_{0j}} \bmod p))^{d_B} \bmod N_B, x_1, y_1]$ and $c_2'' = [g^{a_{0j}} \bmod p, g^{a_{1j}} \bmod p, (h(g^{a_{1j}} \bmod p||g^{a_{0j}} \bmod p))^{d_B} \bmod N_B, x_2, y_2]$. The fact that $c$ was reused is obvious, since the deposited coins agree on all but their last two components. Since the vendors accepted the purchases, we have that $g^{y_1} = (g^{a_{1j}} \bmod p)^{x_1}(g_{a_{0j}} \bmod p$ and $g^{y_2} = (g^{a_{1j}} \bmod p)^{x_2}(g_{a_{0j}} \bmod p$. Thus $a_{1j}x_1 + a_{0j} = y_1 \bmod p-1$ and $a_{1j}x_2 + a_{0j} = y_2 \bmod p-1$. This gives two points on the line $y = a_{1j}x + a_{0j} \bmod p-1$; these points are distinct (else a vendor would have complained for one of the purchases). Thus $B$ can recognize the double usage by the first part of the coin and then interpolate to recover $a_{1j}$ and $a_{0j}$ such that $a_{1j} < q$; with high probability these values are uniquely determined (unless $x_1 - x_2$ is a multiple of $q$, which is unlikely to be possible if $q$ is large). From these $a_{1j}a_{0j}$ can be found. With probability $1 - \frac{1}{k}$, this product is equal to $id_C$ (since it survived the cut-and-choose procedure from the withdrawal protocol). See the note in the next subsection for reducing the cheating probability to be inverse exponential in $k$, and the note on the likelihood that interpolation must be unique.

2. Cheating by the Bank

(a) The two deposited coins $c_1''$ and $c_2''$ that prove reuse by some consumer $C$ must end with some values $x_1, y_1$ and $x_2, y_2$, where $x_1$ and $x_2$ are constructed from the time of the purchase protocol and the vendors $V_1$ and $V_2$ (possibly the same) that were involved. If the bank is cheating, then it is easy for an external judge to see that either $c_1' = c_1''$ does not appear in the history of purchase protocols involving $V_1$ or that $c_2' = c_2''$ does not appear in the history of purchase protocols involving $V_2$.

(b) Checking an RSA signature is easy for $C$ (and for the judge).

(c) The view of $B$ of the withdrawal protocol is simulatable (since the unopened coin is a uniformly random element of $Z_{N_B}$), so we need only consider its view of the deposit protocol. Suppose that $B$ could determine $a_{1j}a_{0j} \bmod p-1$ from $g, p, x, g^{a_{1j}} \bmod p, g^{a_{0j}} \bmod p$, and $a_{1j}x + a_{0j} = y \bmod p-1$ (ignoring the hashed and signed term which $B$ can easily generate on its own). Then $B$ could determine $ab' \bmod p-1$ from $g, p, x^*, g^a \bmod p, g^{y^*}(g^a \bmod p)^{-x^*} \bmod p$, and $y^*$, where $b' = y^* - ax^* \bmod p-1$. Knowing $x^*, y^*, ab' = ay^* - a^2x^* \bmod p-1$, $B$ can solve for $a$ modulo each prime divisor of $p-1$; in fact, the quadratic equation has two solutions $\alpha_1$ and $\alpha_2$ modulo each prime divisor $q$, but the correct one can be found by verifying the identity $(g^a \bmod p)^{(p-1)/q} = g^{\alpha_i(p-1)/q} \bmod p$. From the value of $a$ modulo each prime divisor of $p-1$, the value of $a$ modulo $p-1$ can be found using the Chinese Remainder Theorem. Thus $B$ could extract $a$ from $g, p, g^a \bmod p$, which violates the Discrete Log Assumption.

(d) The judge checks the same things that the vendor checked during the purchase protocol.

3. Cheating by a Vendor

(a) This case is similar to 1a.

(b) Depositied coins are identical to spent coins, so this condition is immediate.

(c) This case is similar to 2c. However, the vendor cannot construct a message with the bank's signature, so the reduction is slightly different: If the vendor could learn the identity of the consumer, then the *bank* could violate the Discrete Log Assumption.

(d) This case is similar to 2d.

□

## 4.3 Comments on the Proof

We note that the probability of a consumer reusing a coin anonymously (case 1b) can be reduced from $\frac{1}{k}$ to inverse exponential in $k$ at a cost of a $k$-fold increase in the size of the coin. The idea is for $B$ to challenge only half of the $k$ components in step 2 of the withdrawal protocol, and then blindly sign the other half. The consumer can extract $\frac{k}{2}$ "sub-coins" and combine these to be the actual coin. The purchase and deposit protocols proceed on all $\frac{k}{2}$ sub-coins in parallel. Reuse is determined by taking the majority identity revealed by the $\frac{k}{2}$ interpolated lines. To cheat during withdrawal, the consumer must cheat on at least $\frac{k}{4}$ of the sub-coins, and avoid detection when $\frac{k}{2}$ of them are challenged; the chance of avoiding detection is at most $(\frac{3}{4})^{k/2}$. The consumer cannot assemble a reusable coin by sneaking some sub-coins through on several withdrawals; no matter what strategy is adopted, the chances of avoiding detection is always inverse exponential in $k$.

Consider the probability that the interpolated line is uniquely defined (from the proof of case 1b). Interpolation is unique (subject to $a_{1j} < q$) unless $x_1 - x_2$ is a multiple of $q$. If vendor id's are chosen at random, then there exist valid $x_1, x_2$ with this property with probability less than $\frac{v^2t^2}{q}$, where $v$ is the number of vendors, and $t$ is the number of moments of time between deposits (alternatively, certainty of uniqueness can be guaranteed with a careful choice of vendor id's and encoding of time whenever $q > vt$). In practice, since $q$ needs to be large enough to insure that discrete logs modulo $p$ are difficult to compute, this probability will always be quite small (see next subsection).

### 4.4 Practical Considerations

How large should coins be in practice? We take 150 decimal digits as a reasonable size for $p$ to insure that the discrete logarithm problem is hard (see, e.g., [12]), and as a reasonable size for an RSA modulus. Using these sizes for $|p|$ and $|N_B|$ in our scheme, the size of the coin at purchase and deposit time would be only $5|p| = 750$ decimal digits (while the size of message during withdrawal is less than $3k|p| = 450k$ decimal digits for a cheating probability of at most $\frac{1}{k}$).

How large a probability $\frac{1}{k}$ of undetected cheating can be tolerated? In practice, virtually any probability of undetected cheating is manageable, since detected cheating exposes the identity of the perpetrator. The penalty for cheating can be set to be sufficiently severe relative to the value $val$ of the coin (e.g., any penalty in excess of $\frac{val}{k-1}$ would make cheating unprofitable in the long run). If the probability of undetected cheating needs to be very small, then the modification described in the preceding subsection can be used.

Consider the issue of the uniqueness of the interpolated line when a coin is reused. In the preceding subsection, we showed that the interpolated line will be ambiguous with probability less than $\frac{v^2t^2}{q}$ when vendor id's and time id's are randomly chosen (or with certainty if $q > vt$ when carefully chosen). In practice, this is not a problem. Since $q$ is a large factor of $p-1$, it will be nearly the same size, e.g., at least 140 decimal digits. Assuming $v = 10^9$ vendors and $t = 3 \times 10^9$ milliseconds per month implies a probability of ambiguous interpolation less than $10^{-100}$.

Suppose that strong 1-security is desirable, i.e., that no vendor or bank can learn partial information about an honest consumer's identity, or link two purchases to the same anonymous consumer. The scheme that we present in this paper may be strongly 1-secure, but we can supply no proof of that fact. Further protection against leakage and linkage might be gained heuristically by storing the consumer's identity more indirectly than as the product $a_{1j}a_{0j} \bmod p-1$. For example, all of the $a_{1i}$ and $a_{0i}$ could be chosen *randomly*, and the coin could contain the extra component $id_C \oplus f(a_{1j}a_{0j} \bmod p-1)$; here $f$ is some strongly hard to invert function which behaves "like a random oracle," and $\oplus$ is bitwise xor. Now learning any bits of $id_C$ is infeasible without learning all bits of $a_{1j}a_{0j} \bmod p-1$, which is as hard as finding discrete logs. Finding linkage requires showing that $g^{a_1}, g^{a_0}, g^{a'_1}, g^{a'_0}$ is consistent with $f(a_1 a_0 \bmod p-1) \oplus f(a'_1 a'_0 \bmod p-1)$, which is infeasible without learning all of the bits of both products.

What practical hash functions $h$ are available for use in an implementation of our scheme? Two possible candidates for $h$ are the MD4 [17] and MD5[18] message digest algorithms. These were designed specifically for securely preprocessing a document before being signed with a public-key cryptosystem. They are also fast (11.6 Mbit/sec on a SUN Sparc station for MD4, and about 30% slower for MD5). Other candidates include the DES-based hash function of Brachtl *et al* [1], and the Secure Hash Standard (SHS) [13].

# 5 2-Secure Electronic Coin Schemes

We can generalize the definition of security to allow attacks requiring the coordinated cheating of more than one party. For the scheme we presented in the previous section, most desirable properties can still be shown versus such conspiracies (e.g., forgery and reuse of coins, revealed identity of honest consumer). One in particular, however, seems problematic. The protocol as given leaves a consumer vulnerable to a charge of reuse by a corrupt bank working together with one or more corrupt vendors (who change their purchasing histories to help implicate the consumer).

Protecting the consumer against this stronger attack could proceed along the lines suggested by Chaum, Fiat, and Naor [5], e.g., incorporating an encrypted function of the unblinded coin into the withdrawal protocol. For example, each consumer $C$ could have its own RSA encryption key $< N_C, e_C, d_C >$. In step 1 of the withdrawal protocol, each blinded coin could be accompanied by the signed value $(f(a_0||a_1||time))^{d_C} \bmod N_C$, where $f$ is some hard to invert function. During withdrawal, the inverse of $k-1$ of these signed function values would be learned by the bank; to prove reuse of a coin, the bank would have to exhibit the inverse of all $k$ signed function values. This would seem to disallow cheating by the bank to frame a consumer as a reuser, while still protecting the anonymity of honest consumers, but security seems more difficult to prove. In general, then, we leave as an open problem the construction of a provably 2-secure electronic coin scheme.

There are many other open problems that arise from this work, including reducing intractability assumptions, adding other useful money properties (e.g., transferability, divisibility), achieving provably strong 1-security, and achieving provable $t$-security ($t > 1$) for efficient electronic coin schemes. We hope this paper will encourage more work on these challenging problems.

# References

[1] B. O. Brachtl, D. Coppersmith, M. M. Hyden, S. M. Matyas, Jr., C. H. W. Meyer, J. Oseas, Sh. Pilpel, and M. Shilling, "Data authentication using modification detection codes based on a public

one way encryption function," U.S. Patent No. 4,908,861, issued March 13, 1990.

[2] D. Chaum, "Security without identification: transaction systems to make big brother obsolete," CACM 28, 10 (October 1985).

[3] D. Chaum, B. den Boer, E. van Heyst, S. Mjolsnes, and A. Steenbeek, "Efficient offline electronic checks," Eurocrypt '89, pp. 294-301.

[4] D. Chaum and J. Evertse, "A secure and privacy-protecting protocol for transmitting personal information between organizations," Crypto 86, pp. 118-167.

[5] D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash," Crypto 88, pp. 319-327.

[6] I. Damgård, "Collision free hash functions and public key signature schemes," Eurocrypt 87, pp. 205-216.

[7] I. Damgård, "Payment systems and credential mechanisms with provable security against abuse by individuals," Crypto 88, pp. 328-335.

[8] W. Diffie and M. Hellman, "New directions in cryptography," IEEE Transaction on Information Theory, vol. IT-22, 1976, pp. 644-654.

[9] A. De Santis and G. Persiano, "Communication efficient zero-knowledge proofs of knowledge (with applications to electronic cash)," STACS 1992, pp. 449-460.

[10] S. Even, O. Goldreich, and Y. Yacobi, "Electronic Wallet," Crypto 83, pp. 383-386.

[11] S. Goldwasser, S. Micali, and R. Rivest, "A secure digital signature scheme," SIAM Journal on Computing, Vol. 17, 2 (1988), pp. 281-308.

[12] B. A. LaMacchia and A. M. Odlyzko, "Computation of discrete logarithms in prime fields," Crypto 90, pp. 616-618.

[13] National Institute of Science and Technology, Secure Hash Standard, draft, February 1992.

[14] T. Okamoto and K. Ohta, "Disposable zero-knowledge authentications and their applications to untraceable electronic cash," Crypto '89, pp. 481-496.

[15] T. Okamoto and K. Ohta, "Universal electronic cash," Crypto 91, pp. 324-337.

[16] B. Pfitzmann and M. Waidner, "How to break and repair a 'provably secure' untraceable payment system," Crypto 91, pp. 338-350.

[17] R. Rivest, "The MD4 message digest algorithm," Crypto 90, pp. 303-311.

[18] R. Rivest, "The MD5 message digest algorithm," Crypto 91, rump session.

[19] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," CACM, vol. 21, 1978, pp. 120-126.