Easy Come - Easy Go Divisible Cash

Agnes Chan* Yair Frankel** Yiannis Tsiounis***

Abstract. Recently, there has been an interest in creating practical anonymous electronic cash with the ability to conduct payments of exact amounts, as is typically the practice in physical payment systems. The most general solution for such payments is to allow electronic coins to be divisible (e.g., each coin can be spent incrementally but total purchases are limited to the monetary value of the coin). In Crypto'95, T. Okamoto presented the first efficient divisible, anonymous (but linkable) off-line e-cash scheme requiring only $O(\log \mathcal{N})$ computations for each of the withdrawal, payment and deposit procedures, where $\mathcal{N}$ = (total coin value)/ (smallest divisible unit) is the *divisibility precision*. However, the zero-knowledge protocol used for the creation of a blinded unlinkable coin by Okamoto is quite inefficient and is used only at set-up to make the system efficient. Incorporating "unlinkable" blinding only in the set-up, however, limits the level of anonymity offered by allowing the linking of all coins withdrawn—rather than a more desirable anonymity which allows only linking of subcoins of a withdrawn coin.
In this paper we make a further step towards practicality of complete (i.e., divisible) anonymous e-cash by presenting a solution where all procedures (set-up, withdrawal, payment and deposit) are bounded by tens of exponentiations; in particular we improve on Okamoto's result by 3 orders of magnitude, while the size of the coin remains about 300 Bytes, based on a 512 bit modulus. Moreover, the protocols are compatible with tracing methods used for "fair" or "revokable" anonymous cash.

1 Introduction

Off-line untraceable electronic cash has sparked wide interest among cryptographers ([CFN90, FY93, Oka95, CP93a, PW92, Bra93b, BGK95, OO92, DC94, EO94, FTY96, CMS96, CFMT96, DFTY97], etc). In its simplest form, an e-cash system consists of three parties (a bank $\mathcal{B}$, a user $\mathcal{U}$ and a receiver $\mathcal{R}$) and four main procedures (account establishment, withdrawal, payment and deposit). In a coin's life cycle, the user $\mathcal{U}$ first performs an *account establishment protocol* to open an account with bank $\mathcal{B}$. To obtain a coin $\mathcal{U}$ performs a *withdrawal protocol* with $\mathcal{B}$ and during a purchase $\mathcal{U}$ spends a coin by participating in a *payment*

* College of Computer Science, Northeastern University, Boston, Massachusetts.e-mail: ahchan@ccs.neu.edu
** CertCo, NY, NY. e-mail: frankely@certco.com
*** GTE Laboratories, Waltham MA. Work partially performed when affiliated with the College of Computer Science, Northeastern University, Boston, MA. e-mail: ytsiounis@gte.com

protocol with the receiver *(shop)* $\mathcal{R}$. To deposit a coin, $\mathcal{R}$ performs a *deposit protocol* with the bank $\mathcal{B}$. An e-cash system is *anonymous* if the bank $\mathcal{B}$, in collaboration with the receiver $\mathcal{R}$, cannot trace the coin to the user, except with negligible probability. The system is *off-line* if during payment $\mathcal{R}$ and $\mathcal{U}$ do not communicate with the bank $\mathcal{B}$. For the bank's protection, if an off-line coin is double spent, the user's identity is revealed with overwhelming probability.

An e-cash system as presented above, however, is not a complete solution. In practice it is desirable (similar to other payment systems, electronic or physical) that payments of arbitrary amounts (up to the withdrawn amount) can be made. Intermediate solutions are possible (see remarks below) but they either diminish the provided anonymity and/or result in communication overhead. Therefore divisibility is the missing link for constructing practical e-cash systems. This paper presents the first truly efficient divisible e-cash system.

Previous work: There has recently been a strong effort in developing secure divisible untraceable off-line electronic cash protocols [OO92, DC94, EO94, Oka95]. With *divisible e-cash* a coin of value $\$x$ can be spent in several increments but the total amount cannot exceed $\$x$, unless the user is willing to be identified with high probability. In Crypto '95, Okamoto [Oka95] presented the first divisible e-cash scheme in which all procedures can be performed efficiently (i.e., in $O(\log \mathcal{N})$, where $\mathcal{N}$ = (total coin value)/(smallest divisible unit)); this result has been recently proven to be asymptotically optimal [OY98]. Furthermore, all protocols are of comparable efficiency with the most efficient *non-divisible* off-line e-cash systems available, *except for the account establishment protocol which takes more than 4000 multi-exponentiations modulo a 1030 bit prime.*[4] Hence, [Oka95] can be practical only if account establishment is performed infrequently (typically once) for each user. Account establishment is used to create a "license" with which coins are withdrawn; hence the cost of not performing it at each withdrawal is that withdrawals of coins using the same license can be linked. As noted by [PW92], the more the user uses the same license the more likely he can be traced by other means (i.e., correlating various payments' locality, date, type, frequency, etc.). In fact, there have been independent results [Oka96] which reduce the computation of the account establishment of [Oka95] by two orders of magnitude. But our "account establishment" protocol is three orders of magnitude more efficient than [Oka95] (see Section 6), hence its functionality can be included in every withdrawal and, unlike [Oka95], there is no trade-off between the degree of unlinkability among coins and efficiency attained.

Alternatives to divisibility: In contrast to a divisible coin, an exact payment protocol using multiple single term coins was analyzed in [FPST97]. That result complements the protocol presented here, in the sense that [FPST97] is more efficient for small divisibility precisions ($\mathcal{N}$) but it becomes impractical when higher precision is required. As analysed in detail in [Tsi97], keeping multiple coins is in general (except for very large $\mathcal{N}$) faster at payment, but it becomes a bottleneck in storage and computation at withdrawal. The exact threshold de-

[4] Each multi-exponentiation is equivalent to approximately 1.2 modular exponentiations.

pends on an additional parameter K, defined as "the number of exact payments that can be performed after one withdrawal"; for a 512 bit modulus the computation at withdrawal for the divisible system is smaller when $K \cdot \ln \frac{N}{K} \geq 48$, while the storage requirements are smaller when $K \cdot \ln \frac{N}{K} > 61$.

Other approaches to exact payments involve having the user simply state the amount s/he is spending, by including the amount in the hash-computed challenge used for the payment (challenge semantics/electronic checks). Then, either the payment is checked on-line with the bank [PW92, Cha85] to prevent over-spending, or a trustee is allowed to trace the user upon over-spending [JY96]. But in several settings on-line payments are clearly undesirable, while providing divisibility "directly" allows the bank to call upon the trustee only when a judge orders such tracing (i.e., presumably much less frequently). Another on-line solution is making "change" with the bank just before a purchase [BGK95]. Methods for off-line "electronic checks" also exist [dCv+89, Bra93a, dST98]. However the model of electronic checks is quite different in the following respect: users can conduct only a single payment (unless of course there is on-line communication) regardless of the amount. Thus, "having \$1,000 in one's wallet" is a relative term, since after e.g., a \$1 purchase the "wallet" is empty. In this sense the model is a subset[5] of the "K-payment model" presented in [FPST97], for $K = 1$. Yet another approach would be to ask the shop to return change, but this transfers the problem of "exact payments" to the shop, while it creates anonymity-related problems and it may require on-line communication with the bank for "refreshing" the anonymity of the returned change.

Our contributions: The major advantage of our system is that the construction of the electronic license (the bulk of the computation in [Oka95]'s "user account establishment" protocol) can be performed with a few tens of exponentiations, while [Oka95] requires several orders of magnitude more (see Section 6 for details). Furthermore, in contrast to our scheme, the number of exponentiations in [Oka95] depends on the length of the RSA modulus (which is a potentially insufficient 512 bits in their efficiency calculations), impairing scalability.

Our system remains efficient during payment, while the size of our coin is around 300 bytes for a 512 bit modulus (see Section 6 for a detailed analysis).

An additional advantage to our system is that it is compatible with tracing methods for e-cash [CMS96, DFTY97, FTY96], thus a full solution (e-cash with exact payments and anonymity revokation) can be employed. Transferability can also be added in a modular fashion, as described in Section 8.

We also present a tool for "range-bounded commitment" which has applications outside electronic cash (e.g., in group signatures [CS97]). Lastly, the security of the divisible electronic cash protocol is provided under a formal model [FY93], based on new cryptographic assumptions (similar to those appearing in [Bra93b] and [Oka95]—see Appendix A).

Organization: We first give an overview of Okamoto's [Oka95] scheme in Sec-

[5] However, electronic checks allow unlinkability between 2 payments (the purchase and the refund); unlinkability between K payments is much more costly—see [Tsi97] for an analysis.

tion 2, focusing on the account establishment and withdrawal protocols. In Section 3 we sketch our idea and illustrate the necessity of a multiplicative commitment, which is achieved using the range-bounded commitment presented in Section 4. Our scheme is then described in Section 5. Next we discuss the scheme's efficiency (Section 6) and its security (Section 7). We conclude with a discussion and open problems in Section 8.

2 The Okamoto scheme

In Okamoto's divisible off-line e-cash scheme [Oka95] each user $\mathcal{U}$ generates a composite number N of the form $N = p^i q^j$ where p, q are primes with $p \equiv 3 \bmod 8$, $q \equiv 7 \bmod 8$ and i, j are odd integers[6]. $\mathcal{U}$ is associated with $g^{p^i} \pmod P$, where $P = 2Q + 1$ with P, Q primes and g a generator of the subgroup G_Q of Z_P^*.

In **the account establishment protocol** the user obtains a license $(N, L_1 = (N + a_1)^{1/K} \bmod n_1, L_2 = (N + a_2)^{1/K} \bmod n_2)$ where (n_i, K) is an RSA public key and $a_i \in_R Z_{n_i}^*$ is also public. To provide for anonymity of the user and security for the bank, this protocol takes approximately 4000 "multi-exponentiations" modulo a 1030 bit prime, assuming 256 bit primes p and q. Furthermore the number of exponentiations depends on the length of the RSA modulus. Due to an attack in [CFMT96], Okamoto suggests a fix to this protocol, requiring approximately 150 additional multi-exponentiations (see appendix of [CFMT96]).

Withdrawal of the coin is nothing more than an RSA blind signature [Cha83] on $H(N||b)$, where H is a one-way function and b is a random value. The bank's public key is chosen based on the value of the coin.

The payment protocol consists of two parts:

- **(Coin Authentication)** $\mathcal{U}$ convinces $\mathcal{R}$ that the coin is a legitimate coin (i.e., (1) it is signed by $\mathcal{B}$, and (2) N is of the correct form).
- **(Denomination Revelation)** Nodes of a tree defined by N are "opened" such that if the spent amount exceeds the monetary value of the coin then N can be factored.

The coin authentication protocol in combination with the denomination protocol guarantee that N is a composite of the form $N = p^i q^j$ where p, q are primes with $p \equiv 3 \bmod 8$, $q \equiv 7 \bmod 8$ and i, j are odd integers. If a node is double spent (*same node rule*), or if an ancestor or descendant of an already spent node is opened (*root route rule*), then using [Oka95] in conjunction with the observation of [CFMT96] N will be factored and $g^{p^i} \pmod P$ will be determined.

The reader should note that the same N is used for each coin with the same license. Hence, coins can be linked. Our system does not have this property.

[6] In [Oka95] N was a Williams Integer, i.e., $i = j = 1$, however, as observed in [CFMT96] the [Oka95] protocols can only guarantee that N is of the form presented above.

3 The basic idea

Okamoto's scheme [Oka95] is quite efficient. In fact it is only inefficient during the account establishment protocol. To emulate the functionality of this protocol, all that is needed is a method for providing a receiver $\mathcal{R}$ with an N, such that (1) N is a composite of two numbers, (2) N is signed by the bank, and (3) $\mathcal{R}$ (and subsequently the bank, at deposit time) is guaranteed that if N is factored, the owner of the coin will be identified.

The denomination revelation protocol of [Oka95] guarantees condition (1). It determines that N is of the form $p^i q^j$ and allows the generation of a binary tree such that each tree node represents a portion of the coin's value, and over-spending results in factoring N.

We suggest a new approach for withdrawal (i.e., signing N) and coin authentication (i.e., proving the correctness of N to $\mathcal{R}$). Our idea is to modify the Brands [Bra93b] protocol for withdrawal and coin authentication. At withdrawal, $\mathcal{U}$ randomly generates $N = pq$ and identifies the particular withdrawal (hence himself) with $I \equiv g_1^p \pmod P$ where g_1, g_2 are generators of the subgroup G_Q of Z_P^*. During withdrawal, $\mathcal{U}$ ends up with a message ($A = g_1^{pq} g_2^q \pmod P, B = [N, g_2^q]$) and a signature on A, B: $sign_{\mathcal{B}}(A, B)$. Hence (2) above is guaranteed. The correctness of A and the unforgeability of the signature are guaranteed by the protocol in [Bra93b].

To guarantee condition (3), we observe that during payment N is revealed and if the coin is over-spent N can be factored, based on the result in [Oka95] and as corrected in [CFMT96]. At coin authentication $\mathcal{U}$ proves that $A \equiv g_1^N X \pmod P$ for some $X \equiv g_2^\beta \pmod P$. Since the withdrawal guarantees that $A \equiv g_1^{pq} g_2^q \pmod P$, this indirectly guarantees that $N = pq$, i.e., the factorization of N reveals the user's identity I. Notice though that this only holds if we guarantee that for a given ${g_1}^p, {g_1}^q, {g_1}^N$ with ${g_1}^{pq} \equiv {g_1}^N \pmod P$, we have that $N = pq \in Z$ (instead of simply $N \equiv pq \pmod Q$). We call this property *"multiplicative commitment"*. There are various promising ways to satisfy it, such as working over a composite modulus (instead of a prime P), or limiting the size of p, q so that no wrap-around occurs in the index operations (i.e., in $g_1^{pq} \pmod P$, $pq < Q$, where Q is the order of g_1 in Z_P^*). Here we present a concrete way of satisfying this multiplicative property using a range-bounded commitment (i.e., by limiting the size of p, q).

4 Range-bounded commitment

The idea of checking whether a committed integer is in a specific range was developed in [Oka95] for license generation. We shall call such protocols *range-bounded commitments (RBC)*. Here we formalize the notion of RBC and present an efficient instantiation based on the Discrete Logarithm Assumption.

This protocol is of independent value and can be used in limiting the range of numbers in other protocols; e.g., in the group signature scheme of [CS97, pp.

422, 423] it is proposed to limit the size of Alice's secret key (x) using a cut-and-choose approach; our range-bounded commitment is a much more efficient way to implement this check.

A **range-bounded commitment** *(informal def.)* is a protocol between a prover, $\mathcal{P}$, and a verifier, $\mathcal{V}$, with which $\mathcal{P}$ commits to a string, x, and proves to $\mathcal{V}$ that x is within a predetermined range, H, with accuracy δ. Hence the protocol uses a secure bit commitment as a building block; then, with overwhelming probability in the security parameter k, $\mathcal{P}$ can conduct an efficient proof as long as $0 \leq x < 2^H$ and $\mathcal{V}$ is convinced that[7] $|x| \leq (1+\delta)H$. In the process a negligible in k amount of information about x is leaked to $\mathcal{V}$.

We now present an efficient range-bounded commitment protocol based on the DLA in which $\mathcal{P}$ sends commitment $X \equiv g^x \bmod P$ and then proves to verifier $\mathcal{V}$ that $|x| \leq (1+\delta)H \bmod Q$.

Setup: Input a range H and a security parameter k.
Choose the accuracy δ such that $\delta > (2k+2)/H$; this ensures that for a legitimate prover the probability of failure or of information leak is negligible. Select primes Q, P with $P = 2Q+1$, $|Q| = 2(1+\delta)H + 6$. (For simplicity we assume that δ is calculated so that δH is an integer.)
Secret input to $\mathcal{P}$ is x, with $0 \leq x < 2^H$. Common input is a commitment $X \equiv g^x \bmod P$ on x.

The protocol: (For iteration i)

(1) $\mathcal{P}$ picks $u_i \in_R \{0, \ldots, 2^{(1+\delta)H} - 1\}$, and sends $U_i \equiv g^{u_i} \bmod P$ to $\mathcal{V}$.
(2) $\mathcal{V}$ sends $e_i \in_R \{0, \ldots, 2^k - 1\}$.
(3) $\mathcal{P}$ responds with $u_i' = e_i x + u_i$.
(4) $\mathcal{V}$ verifies $g^{u_i'} \equiv X^{e_i} U_i \bmod P$ and $0 \leq u_i' < 2^{(1+\delta)H}$.

The protocol can be made **non-interactive**, as discussed in [BR93], if $e = e_1 \ldots e_j = \mathcal{H}_0(X, U_1, \ldots, U_j)$, where $\mathcal{H}_0 : \{0,1\}^* \mapsto \{0,1\}^{2k}$ is a random oracle. In this case $j = 2$ iterations are needed, so that $|e| = 2k$. If the protocol is **interactive** one iteration suffices, as shown in *soundness* below.

Notice that in this particular instantiation of RBC $\mathcal{P}$ reveals more information than in a Schnorr proof of knowledge [Sch91] of the representation of X; hence $\mathcal{P}$ *also proves that he knows the representation of* X *w.r.t.* g.

For **soundness,** (recall that we can only guarantee that $|x|$ is within the extended range $[0, (1+\delta)H]$) observe that if $|x| > (1+\delta)H \pmod Q$, i.e., $x \geq 2^{(1+\delta)H}$ or $x \leq -2^{(1+\delta)H}$, then for a fixed u_i, there exists only one e_i for which $0 \leq u_i' < 2^{(1+\delta)H}$. For all $e_i' \neq e_i$ the verification fails, hence the prover can only cheat by guessing e_i, with probability of success $1/2^k$ in *each* iteration.

For **completeness,** notice that a legitimate $\mathcal{P}$ (i.e., for which $0 \leq x < 2^H$) fails to convince $\mathcal{V}$ when, in some iteration i, $u_i \geq 2^{(1+\delta)H} - e_i x$. The probability of this happening is at most $2^{k+H-(1+\delta)H} = 2^{k-\delta H}$ and, for j rounds, $1-(1-2^{k-\delta H})^j$. From the choice of δ and $j = 2$, this becomes $1-(1-2^{-k-2})^j = 1-(1-2^{-k-2})^2 \approx 2^{-k-1}$.

[7] Throughout this paper, $|\alpha|$ denotes the length of α in bits.

For **secrecy,** the probability that $\mathcal{V}$ can extract some information regarding x is negligible in k, as analyzed in [FGY96] where operations with exponents are also performed on the integers—but for a different reason and without checking the range.

Formally, we can prove the following theorem:

Theorem 1. *Assuming* $g^x \pmod P$ *is a secure string commitment on* x*, the above protocol is a range-bounded commitment.*

Remarks: For applicability in our e-cash scheme, the leak of information about x from $X \equiv g^x \bmod P$ does not impact security since X is already known to $\mathcal{V}$.

We also want to note that it is possible to use similar ways as [CS97] to "blind" the commitment on $X = g^x$, i.e., by commiting to $X = g^x G^y$ for a second generator G and random y, and conducting a proof of knowledge of x, y in parallel.

5 The scheme

Bank Initialization (setup) procedure:
The bank $\mathcal{B}$ chooses the security parameters k and H, and computes $\delta > (2k + 2)/H$ and primes Q, P, with $P = 2Q + 1$, $|Q| = 2(1 + \delta)H + 6$. Intuitively, k controls security for the bank and H security for the users (anonymity). For the remaining of the paper the notation for modulo operations is simplified; e.g., $h \equiv g^x \pmod P$ is written $h = g^x$. All arithmetic is performed in a subgroup G_Q of Z_P^* of order Q, except for the operations involving exponents, which are performed in Z_Q. $\mathcal{B}$ chooses:

- Generators g, g_1, g_2 of G_Q,
- $\mathcal{H}, \mathcal{H}_0, \mathcal{H}_1, \ldots$, from a family of collision intractable hash functions [Dam88],
- A private key $x \in_R Z_Q$ (a different key can be used for every denomination).

$\mathcal{B}$ publicizes the description of G_Q (i.e., P and Q), the generator-tuple (g, g_1, g_2), the description of $\mathcal{H}, \mathcal{H}_0, \mathcal{H}_1, \ldots$, and its public keys $h = g^x, h_i = g_i^x,\ i = (1, 2)$.
User Initialization (account establishment) procedure:
The user $\mathcal{U}$ shows (by physical or other means) his identity to the bank $\mathcal{B}$ and then $\mathcal{U}$ and $\mathcal{B}$ establish a means with which an authenticated channel can be used during withdrawal.

5.1 Withdrawal

The signature that is used by the bank to sign a coin is a variation of the Schnorr signature [Sch91] and is also used in [Bra93b]. The signature $sign_{\mathcal{B}}(A, B)$ on the pair $(A, B) \in G_Q \times G_Q$, consists of a tuple $(z, a, b, r) \in G_Q \times G_Q \times G_Q \times Z_Q$, such that:

$$g^r = h^{\mathcal{H}(A,B,z,a,b)} a \quad \text{and} \quad A^r = z^{\mathcal{H}(A,B,z,a,b)} b \tag{1}$$

The withdrawal protocol

At the beginning of the withdrawal protocol, the user creates an authenticated channel with the bank. This is needed in all e-cash (and physical cash!) protocols to guarantee that only the owner of an account withdraws money from it and that the user is communicating with the real bank.

- $\mathcal{U}$: (This step can be pre-computed.)
  Select primes $p \equiv 3 \bmod 8, q \equiv 7 \bmod 8$ at random such that $|p| = |q| \leq H = (|Q| - 6)/[2(1+\delta)]$ and calculate $N = pq$.
  Send $I = g_1^p$.
- $\mathcal{U}, \mathcal{B}$: Using *the range-bounded commitment* with security parameter k and range H, $\mathcal{U}$ proves to $\mathcal{B}$—in an interactive way and with just one iteration—that he knows the representation of I w.r.t. g_1 (i.e., the number p) and that $|p| \leq (1+\delta)H$.
- $\mathcal{B}$: Verify $I \neq \{1, g_1\}$, $Ig_2 \neq 1$.
  Pick $w \in_R Z_Q$, and send $a' = g^w$, $b' = (Ig_2)^w$ to $\mathcal{U}$.
- $\mathcal{U}$: Compute $z' = h_1^p h_2$ $(= (Ig_2)^x$ since $h_1 = g_1^x$, $h_2 = g_2^x)$. This step can be pre-computed (or z' can be supplied by $\mathcal{B}$).
  Let $A = (Ig_2)^q = g_1^N g_2^q$, $B = [N, Y = g_2^q]$, and $z = z'^q$.
  Pick $v_1, v_2 \in_R Z_Q$ and compute $a = a'^{v_1} g^{v_2}$ and $b = b'^{qv_1} A^{v_2}$.
  Compute the challenge $c = \mathcal{H}(A, B, z, a, b)$, and send the blinded challenge $c' \equiv c/v_1 \pmod{Q}$ to $\mathcal{B}$.
- $\mathcal{B}$: Send the response $r' \equiv c'x + w \pmod{Q}$ to $\mathcal{U}$, and debit $\mathcal{U}$'s account.
- $\mathcal{U}$: Accept iff $g^{r'} = h^{c'} a'$ and $(Ig_2)^{r'} = z'^{c'} b'$. Compute $r \equiv r'v_1 + v_2 \pmod{Q}$, to get the signature (z, a, b, r) on (A, B).

5.2 Payment & Deposit

We remind the reader that [Oka95], given a composite N signed by the bank, shows in the denomination revelation how to define a coin as a binary tree, and identify the user upon over-spending. Since we have a different signature scheme for N, we describe a new verification in the coin authentication below; otherwise payment and deposit are the same as in [Oka95].

Coin Authentication:

- **$\mathcal{U}$ sends the coin to $\mathcal{R}$:** Send $A, B = [N, Y], sign_{\mathcal{B}}(A, B)$, where $N = pq, Y = g_2^q$ as defined at withdrawal.
- **$\mathcal{R}$ verifies that the coin is legitimate:**
  1. **Verify the signature** $sign_{\mathcal{B}}(A, B)$, and that $Y \neq g_2$, $Y \neq g_2^N$, $A \neq 1$, $(-1/N) = 1$, $(2/N) = -1$. Here, (a/N) denotes the Jacobi symbol of a modulo N.
  2. **Prove that q is chosen correctly:** Use *the range-bounded commitment* (base g_2) with Y, to prove that $\mathcal{U}$ knows the representation of Y w.r.t. g_2 (i.e., he knows q) and that $|q| \leq (1+\delta)H$. The challenge e is computed based on a hash function so that even a collaboration of $\mathcal{U}$ and $\mathcal{R}$ cannot

forge the proof; the input of the hash function includes the shop's identity, the payment date and time, a description of the purchase, and a description of the nodes that are to be spent. 2 iterations are performed as discussed in Section 4 for the non-interactive case.

3. **Verify that A is correctly constructed:** $g_1^N Y \stackrel{?}{=} A$.
4. **Limit the way $\mathcal{U}$ can misbehave:** Check whether N is divided by the first $|N|$ primes congruent to 3 mod 4. This simplifies identification of double-spenders as pointed out by [CFMT96]. [CFMT96] also describe the tracing protocol used by $\mathcal{B}$ in this case. We adopt this protocol.

Denomination Revelation: We use [Oka95]'s protocol with the only modification being the substitution of the coin (C, N) in the hash functions of Okamoto with our coin, (A, B). This protocol guarantees that if one of the node rules (see Section 2) is violated, then $\mathcal{U}$ has released enough information to allow $\mathcal{B}$ to factor N. Note that if $k' < k/4$ nodes are spent, then $2 \cdot (k/4 - k')$ additional square roots of randomly chosen numbers must be shown by the user; these are described in [Oka95]'s coin authentication and are also performed here.

Deposit: $\mathcal{R}$ sends the payment transcript to $\mathcal{B}$.

6 Efficiency

We examine the efficiency when $H = |p| = |q| = 256$, $k = 40$, $N = 512$ bits,[8] $|Q| = 688$ (i.e., $\delta \approx 0.33$), $|P| = 689$, and the binary tree has 18 levels, i.e., the divisibility precision is 2^{17}, hence sufficient to divide a \$1,000 coin down to 1 cent. We assume the existence of fast, random hash functions. No pre-processing is assumed (unless explicitly stated). In practice several of the steps can be pre-computed.

Below we compare our system with [Oka95] when [Oka95] is run with the setup at each withdrawal. Also, as a baseline, we use exponentiations over a 512 prime modulus with 512 bit exponents, since the two systems use a different modulus at different steps in the protocols.

Storage requirements: The information $\mathcal{U}$ needs to store for one coin $(p, q, (a, b, r))$ is 323 Bytes (up to 495 Bytes if $\mathcal{U}$ stores, rather than recalculating before each payment, A and/or z). In comparison, the coins in [Oka95] are 264 Bytes and in [Bra93b] 384 Bytes, when the same parameters are used.

Computation and communication: Our exponentiations are approximately 3 times less costly than [Oka95], since we use a modulus of 689 bits instead of 1030 and exponentiations require $O(\log Q \log^2 P)$ bit operations [Kob87]; here we assume that a modular multiplication is performed in $O(\log^2 P)$ steps, i.e., in time quadratic[9] to the modulus size [Kob87]. Due to the reduced exponent size

[8] Although we believe that 512 bits are not sufficient for an RSA modulus, we use this value for comparison with [Oka95]. However our coin remains small if, e.g., $|N| = 1024$ (+300 Bytes).

[9] Algorithms performing multiplication in time $O(\log^{1.585} P)$ do exist [Knu81] but are applicable for numbers much larger than the ones we use here.

some of our exponentiations, including the ones in our range-bounded commitment, are approximately 6 times less expensive than [Oka95]. We should note that a multi-exponentiation (of the form $g_1^{x_1} g_2^{x_2}$) costs the equivalent of 1.2 exponentiations [Oka95, Knu81].

At withdrawal $\mathcal{U}$ and $\mathcal{B}$ perform the equivalent of 12 full exponentiations modulo N (512 bits). $\mathcal{U}$ also needs to calculate one Williams integer, but he can pre-compute one any time before withdrawal. In contrast, in [Oka95] $\mathcal{U}$ needs to perform more than 4000 exponentiations[10] when setup is performed at each withdrawal to obtain the same functionality (i.e., to obtain unlinkability between coins in both systems). This equates to approximately 32,000 multi-exponentiations for the 512 bit baseline.

In the coin authentication phase, $\mathcal{U}$ transmits 774 Bytes. $\mathcal{U}$ needs to perform the equivalent of around 5 exponentiations modulo N (if he re-computes A, Y and z) and $\mathcal{R}$ around 7.

In the denomination revelation phase,[11] 9 nodes (on the average) are paid. For each node, two 512 bit values are sent to $\mathcal{R}$, for a total of 1,152 Bytes. In addition, about 320 Bytes (on the average) are sent for verifying that N is a Williams integer. As in [Oka95] the user computes approximately 20 square roots $\pmod N$ which the shop verifies.

7 Security

We now present our security model and give an overview of the proofs.

As with [Oka95], our security model has been based on [FY93] and is modified to work for divisible, unlinkable coins. We model the security of our scheme by requiring that it satisfies four requirements, which are slightly stronger than the corresponding [Oka95] properties (included in brackets) which do not include unforgeability: *unreusability* [No overspending], *untraceability* [No tracing], *unexpandability* [No forging and No swindling[12]], and *unforgeability*. The use of a probabilistic polynomial time machine *(p.p.t. TM)* in our model simulates user collaboration as views to the TM. Thus, in establishing the security we prove that even a collaboration of users (and/or shops) cannot break the scheme.
Our proofs of security are based on the following **assumptions:**

- **(Factoring and decision Diffie-Hellman)** Let $P = 2Q+1, Q, p_0, q_0, p_1, q_1$ be primes, $N_0 = p_0 q_0$, $N_1 = p_1 q_1$, and $H = |p_0| = |q_0| = |p_1| = |q_1| \leq (|Q| - 6)/(2(1+\delta))$ for some $\delta > 0$. Let the order of g in the multiplicative group Z_P^* be Q. Then, no p.p.t. TM $\mathcal{M}$ can, given $P, Q, \delta, g, [Y_0(\equiv g^{q_0} \bmod P), N_0(= p_0 q_0)], [Y_1(\equiv g^{q_1} \bmod P), N_1(= p_1 q_1)]$ and $[I_r(\equiv g^{p_r} \bmod$

[10] Due to the attack in [CFMT96] the computation is actually higher (Okamoto suggested a method requiring 4% more exponentiations); we omit these computations.

[11] We adopt the calculations of [Oka95].

[12] "No swindling" is defined as the (computational) impossibility of the shop to double-spend a paid coin. This is guaranteed here from unreusability and unexpandability: even a collaboration of users/shops cannot over-deposit the withdrawn/paid coins.

$P), I_{1-r}(\equiv g^{p_1-r} \bmod P)]$ ($r \in_R \{0,1\}$), compute r with probability better than $1/2 + 1/H^c$, for all constants c and sufficiently large H (i.e., $\mathcal{M}$ cannot compute r non-negligibly better (in H) than random guessing).
- **(Withdrawal protocol)** If random hash functions exist, then our withdrawal protocol is a *restrictive blind signature protocol*: the message $m = Ig_2 = g_1^{u_1} g_2$ is signed by the string $A = g_1^{\alpha} g_2^{\beta}$, in such a way that $\alpha/\beta = u_1$.
- **(Hash functions)** Hash functions $(\mathcal{H}, \mathcal{H}_0, \mathcal{H}_1, \ldots)$ act like random oracles.

Remarks on these assumptions are provided in appendix A.

Our *Withdrawal protocol* assumption is based on the representation problem in groups of prime order, which in turn is equivalent to the discrete logarithm assumption *(DLA)* [Bra93a]:

Definition 2. (The representation problem in groups of prime order)
Instance: A group G_Q, a generator-tuple $(g_1, \ldots, g_k), h \in G_Q$.
Problem: Find a representation of h with respect to $(g_1, \ldots, g_k)$.

For our proofs we use lemma 3 below, which has been proven by [PS96] based on the DLA and our *Hash functions* assumption.

Lemma 3. (Schnorr signatures) *Schnorr signatures [Sch91] are existentially unforgeable, even when the prover in the Schnorr identification protocol is queried polynomially many times.*

Theorem 4. Unreusability:
Let k be the security parameter. If the successfully deposited nodes of a coin violate the root route rule *or the* same node rule, *then the identity of the coin's owner can be efficiently (i.e., by a p.p.t. TM) computed (and subsequently proven) from the transcripts of the withdrawal and the deposit protocols with overwhelming probability (in k).*

Theorem 5. *Let k be the security parameter. WLOG we treat the collection of the portions of a coin as being a single, indivisible, coin.*
Unforgeability: *No p.p.t. TM can, from the views of users of arbitrarily many withdrawal and payment protocols, compute a single coin that does not embed the identity of at least one of these users and that will lead to two successful purchase (or deposit) protocols, except with negligible probability (in k).*

Unexpandability: *The probability that from the views of users and shops of N withdrawal and of N payment protocols, a p.p.t. TM can compute an additional coin that will lead to a successful purchase (or deposit), is negligible (in k).*

Theorem 6. Untraceability:
Let k be the security parameter. We treat the collection of the portions of a coin as being a single, indivisible, coin.
Let a p.p.t. TM $\mathcal{M}$ have access to all $\mathcal{B}$'s views of withdrawal, payment and deposit protocols. Then for any two coins C_i, C_j and withdrawals W_0, W_1, such

that C_i, C_j are the coins originating from W_0, W_1, $\mathcal{M}$ cannot distinguish non-negligibly better in random guessing (in k) whether C_j came from W_0 or W_1.
Unlinkability: *this theorem guarantees unlinkability in the average case.*[13]

8 Extensions and open problems

Transferability: There is a general method with which a coin can be transferred from the shop (who now acts as a payer) to another payee, proposed in [vA90]. The method preserves the anonymity of the shop and is applicable to all anonymous off-line e-cash schemes. The coins grow upon each transfer, but [CP93a] showed that this is inevitable, and the approach is asymptotically optimal. Intuitively, the shop obtains a "blank" (zero-valued) blind coin from the bank, and includes it in the hash of the random challenge to the user (for exact payments divisible "blank" coins can be obtained). Then the shop can transfer the payment by "spending" the blank coin with a payee. Note that the blank coin is "bound" to the original payment (since it is included in the random challenges used for that payment), while the shop cannot over-spend, or it is identified. The shop only needs to contact the bank (in an off-line manner) in order to obtain "blank" coins; finding algorithms to withdraw multiple (unlinkable) "blank" coins faster than performing multiple withdrawals in parallel is a problem pending further research.

Tracing methods: The tracing techniques of [CMS96, DFTY97, FTY96] build on withdrawal protocols that are similar to [Bra93b]. Thus, adapting their functionality to our protocols is straightforward.

Finding double-spenders: We remark that when a user over-spends the bank establishes a link between her/his deposit and withdrawal protocols for identification; ideally a link to the user's account should be possible, to minimize database accesses (since accounts are much fewer than withdrawals). Fortunately, the additional cost necessary for this case is minimal, and is omitted here for clarity; efficiency is only marginally affected (one Schnorr proof at payment) and security remains intact, with the exception of our first assumption which needs to be modified as shown in Appendix A.

It is apparent that the storage, computation and communication requirements of our scheme are minimal, resulting in the first anonymous, unlinkable divisible off-line electronic cash scheme that can be implemented in practice.

Two interesting open problems are to reduce the security of the system to more standard assumptions, and to find a way to break the linkability between portions of the same coin.

[13] In particular if the bank can link coins then it can trace users, except for some special cases, e.g., when all users have made the exact same number of withdrawals and payments.

Acknowledgements

We would like to thank Tatsuaki Okamoto for pointing out the need for checking the range of exponents (for which we designed the range-bounded commitment); Moti Yung for providing several comments and suggestions; Philip MacKenzie for joint work on [CFMT96] and discussions; and Matt Franklin for valuable discussions.

References

[BGK95] E. F. Brickell, P. Gemmell, and D. Kravitz. Trustee-based tracing extensions to anonymous cash and the making of anonymous change. In *Symposium on Distributed Algorithms (SODA)*, Albuquerque, NM, 1995.

[BR93] M. Bellare and P. Rogaway. Random oracles are practical: a paradigm for designing efficient protocols. *First ACM journal on Computer and Communications security*, 1993. Available at http://www-cse.ucsd.edu/users/mihir/crypto-papers.html.

[Bra93a] S. Brands. An efficient off-line electronic cash system based on the representation problem. Technical Report CS–R9323, CWI (Centre for Mathematics and Computer Science), Amsterdam, 1993. anonymous ftp: ftp.cwi.nl:/pub/CWIreports/AA/CS-R9323.ps.zip.

[Bra93b] S. Brands. Untraceable off-line cash in wallets with observers. In *Advances in Cryptology — Crypto '93, Proceedings (Lecture Notes in Computer Science 773)*, pages 302–318. Springer-Verlag, 1993. Available at http://www.cwi.nl/ftp/brands/crypto93.ps.Z.

[CFMT96] A. Chan, Y. Frankel, P. MacKenzie, and Y. Tsiounis. Mis-representation of identities in e-cash schemes and how to prevent it. In *Advances in Cryptology — Proceedings of Asiacrypt '96 (Lecture Notes in Computer Science 1163)*, pages 276–285, Kyongju, South Korea, November 3–7 1996. Springer-Verlag. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html.

[CFN90] D. Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In *Advances in Cryptology —Crypto '88 (Lecture Notes in Computer Science)*, pages 319–327. Springer-Verlag, 1990.

[Cha83] D. Chaum. Blind signatures for untraceable payments. In D. Chaum, R.L. Rivest, and A. T. Sherman, editors, *Advances in Cryptology. Proc. Crypto'82*, pages 199–203, Santa Barbara, 1983. Plenum Press N. Y.

[Cha85] D. Chaum. Security without identification: transaction systems to make Big Brother obsolete. *Commun. ACM*, 28(10):1030–1044, October 1985.

[CMS96] J. Camenisch, U. Maurer, and M. Stadler. Digital payment systems with passive anonymity-revoking trustees. In *Esorics '96*, Italy, 1996. To appear. Available at http://www.inf.ethz.ch/personal/camenisc/publications.html.

[CP93a] D. Chaum and T.P. Pedersen. Transferred cash grows in size. In *Advances in Cryptology — Eurocrypt '92, Proceedings (Lecture Notes in Computer Science 658)*, pages 390–407. Springer-Verlag, 1993.

[CS97] J. Camenisch and M. Stadler. Efficient group signature schemes for large groups. In B. Kaliski, editor, *Advances in Cryptology — CRYPTO '97 Proceedings, LLNCS 1294*, pages 410–424, Santa Barbara, CA, August 17–21 1997. Springer-Verlag. Available at http://www.inf.ethz.ch/personal/camenisc/.

[Dam88] I. B. Damgård. Collision free hash functions and public key signature schemes. In D. Chaum and W. L. Price, editors, *Advances in Cryptology — Eurocrypt '87 (Lecture Notes in Computer Science 304)*. Springer-Verlag, Berlin, 1988. Amsterdam, The Netherlands, April 13–15, 1987.

[DC94] S. D'Amiano and G. Di Crescenzo. Methodology for digital money based on general cryptographic tools. In *Advances in Cryptology, Proc. of Eurocrypt '94*, pages 157–170. Springer-Verlag, 1994. Italy, 1994.

[dCv+89] B. den Boer, D. Chaum, E. van Heyst, S. Mjolsnes, and A. Steenbeek. Efficient off-line electronic checks. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology, Proc. of Eurocrypt '89 (Lecture Notes in Computer Science 434)*, pages 294–301. Springer-Verlag, 1989. Houthalen, Belgium, April 10–13.

[DFTY97] G. Davida, Y. Frankel, Y. Tsiounis, and M. Yung. Anonymity control in e-cash. In *Proceedings of the 1st Financial Cryptography conference (Lecture Notes in Computer Science 1318)*, Anguilla, BWI, February 24-28 1997. Springer-Verlag. To appear. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html.

[dST98] A. de Solages and J. Traore. An efficient fair off-line electronic cash system with extensions to checks and wallets with observers. In *Proceedings of the 2nd Financial Cryptography conference*, Anguilla, BWI, February 1998. Springer-Verlag. To appear.

[EO94] T. Eng and T. Okamoto. Single-term divisible electronic coins. In *Advances in Cryptology — Eurocrypt '94, Proceedings*, pages 306 – 319, New York, 1994. Springer-Verlag.

[FGY96] Y. Frankel, P. Gemmell, and M. Yung. Witness-based cryptographic program checking and robust function sharing. In *Proceedings of the twenty eighth annual ACM Symp. in Theory of Computing, STOC*, 1996. To appear. Available at http://www.cs.sandia.gov/~psgemme/.

[FPST97] Y. Frankel, B. Patt-Shamir, and Y. Tsiounis. Exact analysis of exact change. In *Proceedings of the 5th Israeli Symposium on the Theory of Computing Systems (ISTCS)*, Ran-Gatan, Israel, June 17–19 1997. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html

[FTY96] Y. Frankel, Y. Tsiounis, and M. Yung. Indirect discourse proofs: achieving fair off-line e-cash. In *Advances in Cryptology, Proc. of Asiacrypt '96 (Lecture Notes in Computer Science 1163)*, pages 286–300, Kyongju, South Korea, November 3–7 1996. Springer-Verlag. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html.

[FY93] M. Franklin and M. Yung. Secure and efficient off-line digital money. In *Proceedings of the twentieth International Colloquium on Automata, Languages and Programming (ICALP 1993), (Lecture Notes in Computer Science 700)*, pages 265–276. Springer-Verlag, 1993. Lund, Sweden, July 1993.

[JY96] M. Jakobsson and M. Yung. Revokable and versatile e-money. In *Proceedings of the third annual ACM Symp. on Computer and Communication Security*, March 1996.

[Knu81] D. E. Knuth. *The Art of Computer Programming, Vol. 2, Seminumerical Algorithms*. Addison-Wesley, Reading, MA, 1981.

[Kob87] N. Koblitz. *A course in number theory and cryptography*, volume 114 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1987.

[Oka95] T. Okamoto. An efficient divisible electronic cash scheme. In Don Coppersmith, editor, *Advances in Cryptology, Proc. of Crypto '95 (Lecture Notes*

in Computer Science 963), pages 438–451. Springer-Verlag, 1995. Santa Barbara, California, U.S.A., August 27–31.

[Oka96] E. Fujisaki and T. Okamoto, 1996. Unpublished manuscript. Personal communication with T. Okamoto.

[OO92] T. Okamoto and K. Ohta. Universal electronic cash. In *Advances in Cryptology — Crypto '91 (Lecture Notes in Computer Science)*, pages 324–337. Springer-Verlag, 1992.

[OY98] T. Okamoto and M. Yung. Lower bounds on term-based divisible cash systems. In *International Workshop on Public Key Cryptography*, Yokohama, Japan, February 5-6 1998. Springer-Verlag. To appear.

[PS96] D. Pointcheval and J. Stern. Security proofs for signature schemes. In U. Maurer, editor, *Advances in Cryptology, Proc. of Eurocrypt '96*, pages 387–398, Zaragoza, Spain, May 11–16, 1996. Springer-Verlag. Available at http://www.ens.fr/dmi/equipes_dmi/grecc/pointche/pub.html.

[PW92] B. Pfitzmann and M. Waidner. How to break and repair a 'provably secure' untraceable payment system. In J. Feigenbaum, editor, *Advances in Cryptology, Proc. of Crypto '91 (Lecture Notes in Computer Science 576)*, pages 338–350. Springer-Verlag, 1992.

[Sch91] C. P. Schnorr. Efficient signature generation by smart cards. *Journal of Cryptology*, 4(3):161–174, 1991.

[Tsi97] Y. Tsiounis. *Efficient Electronic Cash: New Notions and Techniques.* PhD thesis, College of Computer Science, Northeastern University, Boston, MA, 1997. Available at http://www.ccs.neu.edu/home/yiannis/pubs.html.

[vA90] H. van Antwerpen. Electronic cash. Master's thesis, CWI, 1990.

A Remarks on assumptions

1. An assumption similar to *Factoring and decision Diffie-Hellman* appears in [Oka95]. It implies that the decision Diffie-Hellman and factoring problems are difficult, since if either can be solved the assumption does not hold. If for tracing of over-spenders a pointer to the account database is desired, as described in Section 8, this assumption needs to be modified as follows:
   (Multiple Factoring and decision Diffie-Hellman) (Let $Q, P, H, \delta, p_0, q_0, p_1, q_1, N_0, N_1, Y_0, Y_1, I_r, I_{1-r}$ be defined as previously, and $u_0, u_1 \in_R Z_Q$). No p.p.t. TM can, given $Q, \delta, g, [Y_0, N_0, Y_0'(\equiv g^{u_0 q_0} \bmod P)], [Y_1, N_1, Y_1'(\equiv g^{u_1 q_1} \bmod P)]$ and $[I_r, I_r'(\equiv g^{u_r} \bmod P)], [I_{1-r}, I_{1-r}'(\equiv g^{u_{1-r}} \bmod P)]$ ($r \in_R \{0,1\}$), compute r with probability better than $1/2 + 1/H^c$, for all constants c and sufficiently large H.
   We believe that this assumption represents an interesting number theoretic problem to be studied.
2. The *Withdrawal protocol* assumption appears in [Bra93b]. It is stronger than the DLA, but there are arguments [Bra93b] suggesting that breaking it requires breaking either the Schnorr signature scheme or the DLA.
3. The *Hash functions* assumption is difficult to guarantee. [Oka95] suggests using tamper-free devices. [BR93] suggest an implementation using MD5 in a special manner.