SMART CARD 2000
D. Chaum and I. Schaumüller-Bichl (Editors)
Elsevier Science Publishers B.V. (North-Holland)

Privacy Protected Payments
Unconditional Payer and/or Payee Untraceability

David Chaum

Centre for Mathematics and Computer Science
Kruislaan 413
1098 SJ Amsterdam, the Netherlands

Introduction

A payment system is in essence a means for allowing controlled transfers of value between accounts. The systems presented here keep confidential which pair of accounts is involved in each transfer—unless parties to the transfer wish this information to be revealed. The system provider maintains the balance of each account, and consequently knows each change in balance. But this knowledge need not enable the system provider to discover which withdrawal corresponds with which deposit, because funds are withdrawn and held in a bearer form, like "unmarked bills," before being deposited to other accounts. Also, because these bearer instruments are issued in standard denominations, much like coins or banknotes, the amount of a deposit need not reveal the account from which the funds were withdrawn.

It is initially assumed for simplicity that all accounts are managed by a single system provider, called the *bank*. Funds are said to be withdrawn from the account of a *payer* and deposited to the account of a *payee*. The bank might be able to trace a particular transaction; but, as mentioned above, this should require consent of the payer and/or the payee. Several consent conditions necessary to allow the bank to trace are considered: (a) payer only, (b) payee only, (c) payer and payee, and (d) payer or payee.

The first consent condition will be called *payer untraceability*. In a system providing this condition, even a collusion between the payee and the bank cannot determine the payer's account, unless the payer wishes to come forward. Payer untraceability thus can protect privacy by preventing payments for such things as goods, travel, entertainment, subscriptions, and professional services from being traced back to a consumer's account. The second consent condition, called *payee untraceability,* allows consumers to receive such things as change or refunds without permitting these payments to be traced to their accounts by the payer or the bank. Systems enforcing the final two consent conditions, which seem to have less widespread utility, can be constructed by combining and modifying the payer and payee untraceability systems.

The first section introduces the basic payer untraceability protocol by means of an analogy, and then presents the actual cryptographic mechanism using a formalism. The security of this protocol is considered in the second section, which includes a proof of unconditional security against tracing by the bank and payees. Protocols for payee untraceability and the other two remaining necessary tracing consent conditions are presented in section three. Section four discusses extensions that make the tracing conditions sufficient to let untraceable parties establish, indisputably, their role in transactions. Ways to assure that untraceable parties receive proper interest earnings and charges are presented in section five. Finally, section six considers efficiency and performance aspects for large-scale consumer payment systems and multiple financial institutions.

1. Payer Untraceability

The payer untraceability protocol presented here is based on *blind signatures* <Chaum 1983>, an extension of the digital signature concept <Diffie & Hellman>. Blind signatures are first introduced by an analogy with paper and envelopes in the context of payments. Next, the actual cryptographic protocol is explained discursively with reference to the analogy. Then a notation for such protocols is defined and used to formalize the payer untraceability system.

1.1. Concept with Envelopes

The blind signature concept is readily illustrated by an analogy with carbon-lined envelopes. If you make a mark on the outside of such an envelope, the carbon image of the mark will be transferred onto a piece of paper inside the envelope.

Suppose the bank has a special signature mark that it guarantees to be worth one dollar, in the sense that the bank will pay one dollar for any piece of paper with the mark on it. Several people each take to the bank a different carbon-lined envelope. The bank withdraws one dollar from each of their accounts, makes the signature mark on the outside of the envelopes, and returns each envelope to the same person who supplied it. Upon getting the envelope back, each person removes the slip of paper and makes sure that it is the original slip bearing the carbon image of the bank's signature. Each person then goes to any shop and buys something for one dollar, paying with the slip. Shops verify the signature on a slip before accepting it as payment. Later, when the bank receives the slips for deposit from the shops, it knows that these slips were in envelopes it signed—but because it does not know which envelope contained which slip, the bank cannot know which person went to which shop.

Beyond introducing this basic concept, the analogy can be extended to illustrate some further features and vulnerabilities of the protocol. To begin

with, the system would of course be broken if the bank's signature mark could be forged. Also, the bank must be able to protect itself against envelopes containing more than one slip, possibly by enforcing limits on the weight or thickness of envelopes and slips. To protect their own interests, payers should at least be able to recognize each slip they withdraw and spend; for this purpose, they might write a different random "note number" on each slip before they get it signed. Actually, each slip inherently bears the equivalent of a note number: the fiber pattern on the slip's surface. Note numbers (or their equivalents) do not compromise a payer's untraceability, and might serve, in case of problems, much as check numbers do in today's traceable systems.

1.2. Concept with Numbers

The RSA public-key system <Rivest, Shamir, & Adleman> is used in the description of the protocols below. While the bank's modulus m is publicly known, p and q, the prime factors of m, are assumed to be known only to the bank. The bank's public exponent is shown as b and, as usual, satisfies $(b, (p-1)(q-1)) = 1$; the corresponding private exponent, which is the multiplicative inverse of b modulo $(p-1)(q-1)$, is shown as $\bar{b}$. For simplicity and convenience, only residues modulo m are shown. Thus, the fundamental identity of the RSA public-key system is denoted $(x^{\bar{b}})^b = (x^b)^{\bar{b}} = x$.

The solution using digital signatures follows the envelope analogy quite closely. Suppose you are the payer. First you form a blinded note by choosing a one-hundred digit number at random and expanding it, using a standard public redundancy scheme detailed later, into a two-hundred digit "special number" n (corresponding to the note number written on the slip of paper); you choose a two-hundred digit random number r uniformly from the integers between 1 and $m-1$ (corresponding to choosing an envelope); you raise the random number r to the bank's public exponent b and multiply the result by the note number n, which yields nr^b (corresponding to putting the note-numbered slip in the envelope). Then you supply this blinded note to the bank, which deducts one dollar from your account and sends you a signed copy of the blinded note in the form $(nr^b)^{\bar{b}}$ (corresponding to the bank issuing the signed envelope to you).

When you receive the signed blinded note from the bank, you can consider it to be expressed as $n^{\bar{b}}r$. Thus you can remove the r by multiplying by its multiplicative inverse to obtain the signed note $n^{\bar{b}}$ (corresponding to removing the signed slip from the envelope). By raising this signed note $n^{\bar{b}}$ to the bank's public key b and verifying that the result is your original n, you make sure that the bank has returned exactly the right thing. Some time later, when you give the signed note $n^{\bar{b}}$ to a shop, they verify that it is a signed "special number"; they do this by raising it to the bank's public key and testing whether the result is of the form specified by the redundancy scheme. When the shop then forwards the signed note to the bank, the bank verifies it in the same way as the shop did and records the deposit to the shop's account.

In addition to the problems of signature forgery and multiple slips in an envelope (considered in §2.2 below), a further concern results when numbers are used instead of paper: What is to prevent a person from buying things at different shops with copies of the same note? This problem stems from the ease with which numbers, unlike slips of paper, can be copied exactly. For its own protection against this, the bank would in any case maintain a list of already "cleared" notes n and would accept for deposit only notes not yet listed. For a shop, a straightforward approach is to send the bank the signed note when it is received, finalizing the sale only after the bank accepts the deposit. Several possibly more attractive approaches for shops are considered later (see §§6.3 & 6.4).

1.3. Protocol Notation

In the formalism used to present the protocols here, the actions performed by a participant are grouped together into a *part* consisting of a series of *operations,* possibly followed by a *message.* (The operations of a part are performed as an indivisible unit, excluding other concurrent operations.) Each operation begins with a letter naming the participant (p for payer, e for payee, and b for bank), followed by a ':', and concluding with a description of the operation. Some of these operation descriptions show a message number in square brackets on the left of an equals sign, and, on the right, the way in which the message is formed; others denote Boolean expressions tested by the participant, who terminates the protocol if the value yielded is false; and still others informally state special operations or conditions as text enclosed in double quotes. How messages are formed is shown in the following sequence: a message number enclosed in square brackets; the name of the sender, an '→', the name of the recipient, and a ':'; and an expression showing the value that is transmitted.

A few primitives are used: $x \leftarrow random$ denotes the operation of choosing a value x at random independently and uniformly from the interval 1 to $m-1$; $x \leftarrow special$ denotes the choice of a value x independently and at random such that *check* x is true (e.g., *check* might be defined to be true if and only if the digits of a base 2 representation of x form a palindrome after some possibly cryptographic mixing transformation is applied, as described further in §§2.2 & 4.2); '$\doteqdot$' denotes a Boolean conditional that is true if and only if the expressions on both sides of it have the same value; $f(x_1, x_2, \ldots)$ denotes the result of applying a public one-way function (see, e.g., <Diffie & Hellman>) on the x_i, whose value depends on the type, number, order, and values of its arguments. Literal text strings are enclosed in single quotes.

1.4. Payer Untraceability Protocol

The following is the payer untraceability protocol introduced in §§1.1 & 1.2 expressed in the notation of §1.3:

p: $n \leftarrow special$
p: $r \leftarrow random$
$[1]p \rightarrow b$: nr^b

b: "withdraw from account of p"
b: $[2] = [1]^{\overline{b}}$
$[2]b \rightarrow p$: $n^{\overline{b}}r$

p: $[3] = [2]r^{-1}$
p: $[3]^b \doteqdot n$
p: "wait"
$[3]p \rightarrow e$: $n^{\overline{b}}$

e: $check[3]^b$
e: $[4] = [3]$
$[4]e \rightarrow b$: $n^{\overline{b}}$

b: $check[4]^b$
b: "$[4]^b$ not used before?"
b: "mark $[4]^b$ used"
b: "deposit to account of e"

The five parts of the protocol above may be described as follows: (1) the payer forms the special n and random r, forms the blinded note number from them, and transmits it to the bank; (2) the bank withdraws from the payer's account and returns the signed blinded note; (3) the payer unblinds the note, tests that the bank has returned the proper signature, possibly waits some time, and supplies the note to the payee; (4) the payee verifies the note and forwards it to the bank for deposit; (5) the bank verifies the signature on the note, verifies that the note number has not been deposited before, records the note as deposited, and deposits it to the account of the payee.

Secrecy and authentication of messages are not explicitly shown in the protocol; they could of course be readily provided using well-known techniques. Both are not required for every message, however. For example, if authentication of the participant making the withdrawal is provided for message [1], then message [2] needs no authentication or secrecy, since the payer tests it and an eavesdropper would not know r.

The true value of disputed balances can be determined by any third-party arbiter if each participant maintains proper receipts. To this end, a withdrawal request should contain [1] and be signed by the payer; it might, for example, be of the form f('withdraw', [1])$^{\overline{p}}$, where $\overline{p}$ indicates a signature made by the payer. A receipt for deposit should include [4] and a designation of the payee account, all signed by the bank; this might, for instance, take the form f(e, 'deposited', [4])$^{\overline{b}}$. Thus, an arbiter must count in the bank's favor all withdrawal requests signed by the withdrawer, and must count in the depositor's favor all deposit receipts signed by the bank. Two potential problems are that the bank might improperly refuse to issue a signed note for a

withdrawal request it has shown to the arbiter, or that it might improperly refuse to give a deposit receipt. In the first case, the bank cannot justify refusing to provide the note's signature, even if it claims to be doing so for the second time. A solution for the second case is presented in §4.2.2, under the title "Bankproof Payer Untraceability."

2. Fundamental Security Considerations

One concern is that the bank may be able to somehow trace notes. In an appropriate setting, such tracing can be shown to be impossible—without relying on any assumptions about computational infeasibility. The protocol is thus "unconditionally secure" against this threat. A second concern is that someone may be able to somehow produce more notes than are issued them, by outright counterfeiting or possibly by somehow combining valid notes. This threat would be precluded by a sufficiently secure underlying digital signature technique, and is consequently treated here only by assumption. A third consideration is what might be revealed to the bank by changes in balances. A fourth point relates to the possibility that note numbers may not be unique. Each of the four are considered in turn.

2.1. Unconditional Unlinkability

Can the bank learn anything, in the protocol above, about the correspondence between elements of the set of note numbers signed $\{[2]_i\}$ and elements of the set of notes later received for deposit $\{[4]_j\}$? The answer is no.

An intuitive explanation is quite simple. If the bank chooses any particular signed note [2] and any particular deposit [4], then the bank can always determine exactly one r $(=[2][4]^{-1})$ which would have been used if the two were to correspond. But this r reveals nothing to the bank about whether the two actually do correspond, since the r's are chosen independently and uniformly at random and are unknown to the bank.

A more rigorous argument is presented below:

Let G be a finite multiplicative abelian group and $b,\bar{b}$ be integers such that $b\bar{b}\equiv 1(\mathrm{mod}\,|G|)$. Denote by stochastic variables N_i, R_i, V_i, and W_i the special numbers (n), blinding factors (r^b), blinded notes ([2]), and signed notes ([3]), respectively, all over G, where $1\leqslant i\leqslant k$. The indices of the first three variables might, for instance, correspond to the time order of withdrawal, and those of the fourth to that of deposit. The stochastic variable S takes its values from the permutations of $1,\ldots,k$, and determines which withdrawals correspond to which deposits.

Payers form each blinded note as the product of a special number and corresponding blinding factor, and they check that applying the public key to the signed note yields the special number. More formally:

(1) $V_i=N_iR_i^b$, and

(2) $N_i=W_{S(i)}^b$.

Payers also choose the special numbers, blinding factors, and the permutation S independently and from the appropriate distributions:

(a) $N_1,\ldots,N_k$, $R_1,\ldots,R_k$ and S are mutually independent;

(b) the N_i are identically distributed; and

(c) each R_i is uniformly distributed over G.

Theorem:
The joint probability distribution of $V_1,\ldots,V_k,W_1,\ldots,W_k$ is independent of S.

Proof: From (1), (2), and the fact that raising to the b or $\bar{b}$ power is one-to-one on G, it follows that for each permutation s of $1,\ldots,k$, and for each i, there is a one-to-one correspondence between $(V_i,W_{s(i)})$ and (N_i,R_i), given by $N_i = W_{s(i)}^b$ and $R_i = V_i^{\bar{b}}N_i^{-\bar{b}} = V_i^{\bar{b}}W_{s(i)}^{-1}$. Let $v_i,w_i\in G$, for $1\leqslant i\leqslant k$. Then,

$$P(V_1=v_1,\ldots,V_k=v_k,W_1=w_1,\ldots,W_k=w_k,S=s)$$

$$= P(N_1=w_{s(1)}^b,\ldots,N_k=w_{s(k)}^b,R_1=v_1^{\bar{b}}w_{s(1)}^{-1},\ldots,R_k=v_k^{\bar{b}}w_{s(k)}^{-1},S=s),$$

and by (a) this is equal to $E\times P(S=s)$, where

$$E = \prod_{i=1}^{k} P(N_i=w_{s(i)}^b)\times\prod_{i=1}^{k} P(R_i=v_i^{\bar{b}}w_{s(i)}^{-1}).$$

It suffices to show that E is independent of s. By (b),

$$\prod_{i=1}^{k} P(N_i=w_{s(i)}^b) = \prod_{i=1}^{k} P(N_i=w_i^b).$$

And by (c),

$$P(R_i=v_i^{\bar{b}}w_{s(i)}^{-1}) = |G|^{-1}$$

for each i. Thus,

$$E = \prod_{i=1}^{k} P(N_i=w_i^b)\times|G|^{-k},$$

which is indeed independent of s.□

Now consider how this formal result can be applied in the setting of an actual payment system. Its assumptions are conditions under which payers following the protocol can keep the bank from gaining tracing information from the messages. The assumption that payers can enforce conditions (1) and (2) is easily satisfied fully, but conditions (a), (b), and (c) can be achieved only within the limitations of physical random number sources. Such sources currently available, however, give unlinkability that is, for all intents and purposes, perfect. It is conceivable that a payer could create an n or r not in the multiplicative group modulo m. This possibility, though, can safely be ignored

since if there was a non-negligible chance that such a value would be created, then m could be factored far too easily. If the bank chooses b so that $(b, (p-1)(q-1)) > 1$, then there is no $\bar{b}$ such that $b\bar{b} \equiv 1 \pmod{(p-1)(q-1)}$. But payers would detect this improper b with probability at least $(\frac{2}{3})^{-k}$, since less than a third of blinded notes would possess bth roots.

2.2. Conservation of Signatures

The present work is based on the following assumption: an adversary's ability to obtain signatures on k numbers of the adversary's choice gives no practical advantage in producing signatures on any $k+1$ special numbers. When some very simple special-number schemes are used, however, attacks exploiting the multiplicative properties of RSA signatures can violate this assumption. For example, if special numbers are defined as all numbers below some bound (l), then the product of signatures on two special numbers would be a signature on a third special number, assuming the product of the numbers is below the bound (i.e. if $l \geqslant s, t$, then $s^{\bar{b}}t^{\bar{b}} = (st)^{\bar{b}}$, and $l \geqslant st$ may hold). Thus, two such properly obtained and spendable notes could be used to create a third note that was not withdrawn—but that could nevertheless be spent. Redundancy schemes that require the left- or right-most digits of special numbers to be zero are not very effective in preventing such attacks <Chaum & de Jonge>. But some other simple schemes (described in §4.2), such as those in which the base is a suitable one-way function or just a cryptographic mixing function of the message to be signed, do appear to effectively prevent them. Furthermore, the system can be set up, as mentioned in §4.1.3, so that if it were detected that conservation of signatures had been violated, a fallback security strategy based only on a one-way function could be implemented.

2.3. Linking from Balance Changes

The way the protocol is used can completely reveal or completely hide the correspondence between withdrawals and deposits. If withdrawals and matching deposits alternate, for example, then the correspondence is completely revealed. Or, if the amounts are distinct and every withdrawal is for a single deposit, then a unique correspondence is again easily determined. If, on the other hand, all withdrawals have the same denomination and amount, and they precede all deposits, then nothing is revealed about which deposit corresponds to which withdrawal.

A large-scale consumer payment system implementing this protocol directly would likely reveal very little by account balance changes. The degree of concealment will be influenced, however, by the set of denominations chosen, as discussed in §6.1. Untraceability is improved in this setting if payers hold some randomly selected notes between withdrawals, so that the exact amount held by each payer cannot be determined by others. In addition to the techniques presented here, it is of course also possible for each payer to have two or more accounts that are unlinkable to one another; in the limiting

case, each note would be deposited to and then immediately withdrawn from an account for just that one note.

2.4. Collision of Note Numbers

If two people choose the same note number n, then only one of them can successfully deposit it, since the bank's list ensures that no note number is ever accepted for deposit more than once. Clearly, there is no advantage to paying to withdraw the same note number twice; but it might be unfair if two people independently chose to withdraw the same note, and one of them was ultimately unable to spend it.

This problem may never arise in practice, however. The probability that any two people independently generate the same number uniformly at random is the well-known "birthday problem." Suppose that each person on earth today generates a million note numbers a day for a million years, and that the numbers have random parts (see §4.2) of one hundred decimal digits. Then the probability of even one number being chosen more than once over the entire million years is less than 10^{-50}.

3. The Remaining Consent Conditions

A protocol for the payer untraceability consent condition has been presented above. This section considers the remaining three consent conditions mentioned in the introduction: payee untraceability, mutual untraceability, and bilateral untraceability.

3.1. Payee Untraceability

The following protocol ensures that the payee's consent is necessary for tracing, even if the payer and bank were to cooperate:

e: $n \leftarrow$ *special*
e: $r \leftarrow$ *random*
$[1]\, e \rightarrow p$: nr^b

p: $[2] = [1]$
$[2]\, p \rightarrow b$: nr^b

b: "withdraw from account of p"
b: $[3] = [2]^{\bar{b}}$
$[3]\, b \rightarrow p$: $n^{\bar{b}}r$

p: $[3]^b \doteq [1]$
p: $[4] = [3]$
$[4]\, p \rightarrow e$: $n^{\bar{b}}r$

e: $[5] = [4]r^{-1}$
e: $[5]^b \triangleq n$
e: "wait"
$[5]\, e \rightarrow b$: $n^{\bar{b}}$

b: *check*$[5]^b$
b: "$[5]^b$ not used before?"
b: "mark $[5]^b$ used"
b: "deposit to account of e"

The protocol is in six parts: (1) the payee forms the blinded note and supplies it to the payer; (2) the payer in turn supplies it to the bank; (3) the bank withdraws from the payer's account, signs the blinded note, and returns it to the payer; (4) the payer verifies the signature and returns the note to the payee; (5) the payee unblinds the note, verifies that it was properly signed, and after a possible delay, supplies it for deposit to the bank; and (6) the bank verifies the signature, verifies that the note has was not been deposited before, records the notes as deposited, and deposits to the account of the payee.

This protocol differs from the payer untraceability protocol in that note numbers are created by the person to whose account they are ultimately deposited. Several advantages resulting from this are discussed later in §§4.1.2, 4.2.2, & 6.3. A related advantage is that, unlike the payee in payer untraceability, the payee here can wait safely, knowing that the signature is valid and that the note cannot be deposited by anybody else.

3.2. Mutual Untraceability

In the third consent condition mentioned in the introduction, both payer and payee must consent before a transaction can be traced; each party to the payment wishes to protect its untraceability from the other. An example might be payments for things advertised on public bulletin boards. Mutually untraceable payments can be accomplished with the mechanisms outlined so far, if there is a third party known and trusted to both untraceable parties: the untraceable payer pays the third party using the payer untraceability protocol, then the third party pays the untraceable payee using the payee untraceability protocol. Even if the third party is the bank, it still cannot trace either the payer or the payee.

Another approach to mutual untraceability does not involve such an intermediary; it is based on a modified version of the payee untraceability protocol. In the latter protocol, the blinded note [2] submitted to the bank for withdrawal by the payer would usually be of the form nr^b, and thus would be known to the payee. But the protocol could be modified by the payer, who could apply an additional layer of blinding before the note is signed and then remove the layer before returning the note to the payee. The technique, which is mentioned again in §4.1.2, might be called "double blinding." The payer forms $[2] = [1]x^b = nr^b x^b$, where x is chosen by the payer independently at

random and uniformly from 1 to $m-1$. When the signature is returned as $[3] = n^{\bar{b}} rx$, the payer multiplies it by x^{-1} before returning it to the payee. The payee is unable to determine x or [2], and thus is unable to allow the bank to learn the account of the payer.

3.3. Bilateral Untraceability

In the fourth and final consent condition, either the payer or the payee can allow the bank to verify which two accounts were involved in the transaction. An example might be payments between two individuals, where each individual should be able to seek remedies if a problem arises after the transaction. This condition can be realized by extension of the basic payer or payee untraceability protocols. The untraceable party in the underlying protocol will provide the other party with a copy of the data (detailed in §4.1) that can be used by the untraceable party to sacrifice untraceability.

4. Precursors Protecting Untraceable Party

Because the untraceable party is able to choose r and n, this party can create them as the result of applying a one-way function on randomly generated precursors r' and n', respectively. Each kind of precursor has a variety of uses, covered separately in the following two subsections.

4.1. Precursors to r

Because r serves as a blinding factor, every value for r should be possible, and each should be equally likely. Thus, it might be computed by the untraceable party as $r = f(r')$, where f is a preferably bijective one-way function on residues modulo m, with the precursor r' chosen independently and uniformly from the domain of f. The utility of the untraceable party being able to supply r' to the bank is considered separately below for payer and for payee untraceability.

4.1.1. Substantiating Being the Payer

The untraceable payer may benefit from forfeiting untraceability, thereby revealing both parties to the transaction, in a variety of special circumstances. The payer may, for instance, wish to reveal the payee without the payee's cooperation in the following cases: the payee refuses to give a receipt for payment; notes stolen from the payer have been spent before copies could be deposited by the victimized payer; or the payer wishes to provide incriminating evidence against a seller on a black market, an extortionist, or an acceptor of bribes. (If a trace has been placed with the bank before the notes are cleared, then funds need not actually be transferred and it may be possible to catch the spender on the spot.) When the payer returns unspent notes to the payer's own account, the payer may also wish, for reasons such as taxes, to

substantiate a claim that the funds are being recycled and are not coming from another account.

What is needed is a way to substantiate a claim that a particular n was actually withdrawn from the payer's account. The problem is that a payer knowing a signed note withdrawn from someone else's account can always compute an r that would make one of the payer's own withdrawals seem to correspond to the signed note. This is because a corresponding r can easily be computed as the quotient of any signed blinded note and any signed note (as already mentioned in §2.1). Thus, showing an r does little to substantiate a claim to the bank about the account of withdrawal. A solution is for the payer to provide the precursor r'. While an r can always easily be constructed, finding the corresponding r' is prevented by the one-way property.

For these techniques and those mentioned next to be usable, there must be some record of the blinded notes [1] withdrawn. One approach is simply for the bank to keep an archive of all these notes. (See §6.2 for other uses of a similar archive.) Another is for the payer to obtain withdrawal receipts that are signed by the bank. Such a withdrawal receipt might, for example, be of the form $f([1], \text{'withdrawn by'}, p)^{\bar{b}}$.

4.1.2. Substantiating Being the Payee

In the case of payee untraceability, the payee may benefit from allowing tracing under some circumstances. If the payee is audited unexpectedly, for example, the payee may need to substantiate the claimed source of certain funds deposited by a payee untraceability protocol, like wages or gifts.

One difficulty with achieving this by means of the payee untraceability protocol described so far is that the payee cannot prevent the payer from using "double blinding," as already mentioned in the case of mutual untraceability §3.2. A solution is for the payer to provide the payee with a copy of the withdrawal receipt described in the previous sub-subsection. This allows the payee to verify that no double-blinding took place: that is, that [1] = [2]. Then the payee can substantiate the claimed source of a deposit by showing the precursor r'.

4.1.3. A Fallback System

If signature forgery were detected, recovery might be possible, assuming that the one-way function remains secure: the only payments accepted would be those that include an r', an n satisfying *check*, and an indication of which archived [1] satisfies $[1] = f(r')^b n$. If fallback were ever actually instituted, though, payers would sacrifice untraceability in payments from then on.

4.2. Precursors to n

The special number n is partly chosen at random. The example scheme mentioned in §1.3 forms n by concatenating the binary representation of a suitably chosen random number to itself and then applying an agreed mixing transformation. Another possible scheme would be for the first half of the digits of the special number to be chosen at random and for the second half to be a one-way function of the first half. A further variation on this latter scheme might form the final first half from the bitwise exclusive-or of the original first half and the mapped second half. In such cases, the original first half of the binary representation will be called the "random component" of the special number. More generally, the *random component* refers to the randomly chosen part of the special number when stripped of all redundancy. The precursor of the random component of n will be shown as n' and called simply the precursor of n. Two examples illustrating the usefulness of the untraceable party being able to supply n' are presented below.

4.2.1. Untraceable Payer Designates Payee

An untraceable payer may wish to ensure that the bank allows deposit to only the intended account. For example, when payment is given to an intermediary, such as a waiter or clerk, the funds may ultimately be deposited to the wrong account. Perhaps more important, when some way to substantiate a claim against the payee is needed, the payer may want protection against the possibility that a receipt will not be provided, the funds diverted to an unintended payee, and the payment disavowed.

If the payee is known before the blinded note [1] is supplied to the bank, then a designation of the payee, such as the payee's account number, can be encoded in the note number in some standard way (see §6.3). If the payee is unknown before withdrawal, however, the payer can achieve the same effect by forming the note number so that it contains a public key. The note number might, for example, be the result of applying a suitable one-way function to an RSA modulus. Having decided on a payee, the payer would form a signature on a designation of that payee. This signature would be provided to the payee along with the signed note and could be verified with the public key contained in the note. For notes containing public keys, the bank should verify the signature on the accompanying designation of payee, using the public key in the note, before accepting the deposit. (See §6.7 for consideration of the computational requirements of such signatures.)

The need for a *check* predicate, and other needs for precursors to n discussed in the remainder of this subsection, can conveniently be met using this scheme. The *check* predicate is applied not, as before, on the note number itself but rather on the result of using the public key contained in the note to test the designating signature also supplied. This test additionally verifies the designating signature as a precursor to the note number.

4.2.2. Bankproof Payer Untraceability

The following is a payer untraceability protocol extended to protect the payer who presents a note against the bank falsely claiming to have already deposited the same note to another account:

p: r, $n' \leftarrow$ *random*
p: $n = f(n')$
$[1]p \rightarrow b$: nr^b

b: "withdraw from account of p"
b: $[2] = [1]^{\bar{b}}$
$[2]b \rightarrow p$: $n^{\bar{b}} r$

p: $[3] = [2]r^{-1}$
p: $[3]^b \triangleq n$
p: "wait"
$[3]p \rightarrow e$: $n^{\bar{b}}$

e: $[4] = [3]$
$[4]e \rightarrow b$: $n^{\bar{b}}$

b: $n = [4]^b$
b: "n not pending and not used?"
b: "mark n pending from e until date d"
$[5]b \rightarrow e$: $f(n,$ 'accepted from', e, 'until', $d)^{\bar{b}}$

e: $[5]^b \triangleq f(n,$ 'accepted from', e, 'until', $d)$
$[6]e \rightarrow p$: $[5]$

p: $[6]^b \triangleq f(n,$ 'accepted from', e, 'until', $d)$
$[7]p \rightarrow e$: n'

e: $f([7]) \triangleq [3]^b$
e: $[8] = [7]$
$[8]e \rightarrow b$: n'

b: $f([8]) \triangleq n$
b: "n pending from e?"
b: "change n from pending to used"
b: "retain [8]"
b: "deposit to account of e"

The first four parts of the protocol are essentially the same as the first four of the basic payer untraceability protocol, except that the precursor n' is explicitly developed by the payer. The remainder of the protocol is in parts five through nine: (5) upon receiving the intent to pay, the bank verifies that n has not been supplied previously, records that n was supplied by the payee, and provides the payee with a statement of acceptability, which includes an expiration date d; (6) the payee verifies the signed statement of acceptability and supplies a copy to the payer; (7) the payer also verifies the bank's statement and provides the precursor to the payee; (8) the payee verifies the precursor and forwards it to the bank; (9) the bank also verifies the precursor, verifies that the precursor is for the note accepted and for the payee, records the note as deposited, retains a copy of the precursor, and deposits to the account of the payee.

The bank cannot give improper responses to the payer without allowing the payer to substantiate this fact to third parties. If the bank refuses to give a proper acceptability statement [5], then an arbiter must decide in favor of the payee, unless the bank produces n'. If the bank refuses to give a receipt for deposit, even though an unexpired acceptability statement [5] is held by the payer, and the valid n' is shown, then an arbiter will again decide in the payee's favor. The date stamp in the acceptability statement protects the bank from having to accumulate records of aborted requests indefinitely. The dates stamped must, however, be set far enough in the future to allow redress if the bank fails to honor an acceptability statement.

The protocol also allows a kind of pre-clearing before the final payment. First, the protocol is completed up to but not including the sending of message [7]. Having received [7] from the payer, the payee has only to compare its image under the one-way function with n in order to ensure that payment will ultimately be received. Payments can thus be cleared in advance and consummated with only a small local computation and no communication.

The basic payee anonymity protocol is made bankproof by encoding the payee's account number in the note. The notes are then like the "deposit receipts" mentioned in §1.4, in that if the bank disputes the payee's balance, an arbiter must count them in the payee's favor.

5. Time Value of Money

Current consumer payment systems can be characterized broadly as either "debit" or "credit." In both types of system, a payment causes an amount to be subtracted from the balance of the payer's account and the same amount to be added to the payee's account. But a debit system requires that the transfer leave the payer's balance above zero, whereas a credit system requires only that the transfer leave the payer's balance above some negative limit. (Funds withdrawn from a credit account of course create an obligation for the payer to repay the bank.) The holder of a debit account may earn interest on the balance of the account, say, each day; the holder of a credit account may be charged interest on the amount outstanding each day.

In one approach, the value of notes changes over time; in a second, the bank learns when changes in balances occur and can thereby calculate the interest. The two approaches are described in separate subsections, each of which considers the debit and credit cases for both payer and payee untraceability.

5.1. Dynamic Note Values

First, consider debit accounts with payer untraceability. Suppose that the bank encodes the year of issue or "mint date," possibly along with the denomination, in the type of signature made on each blinded note (see §6.2 for details of such restricted encoding of data in signatures). When a one-dollar denomination note, for example, is withdrawn by a payer, the current value of one dollar is taken from the payer's account (i.e. what a dollar would be worth if it had been earning interest since January first of the mint year). When such a note is used to make payment, it is in effect sold; its "selling price" is determined as the current value at the time of payment, thereby giving the payer the correct interest earned for the period the payer has held the note.

Now consider the credit case for payer untraceability. An expiration date must be included along with the mint date. Notes whose total current value equals the payer's credit limit are provided to the payer by the bank. At the expiration date, all notes the payer does not return to the bank are considered to have been spent, and the payer is charged for the current value of the notes at that time. The difference between what the payer receives for a note at the time it is spent, and the higher price that must be repaid to the bank on the expiration date, is the correct interest charged for the period of the effective loan.

The interest rate for the credit case is usually higher than that for the debit case. This creates an incentive for payees to cheat by holding credit notes as long as they can so as to receive the higher rate. Such abuse can be discouraged if payers routinely obtain from the payee a receipt specifying the note numbers in question; the receipt could, for instance, be used to entrap a payee who cheats. If payers include the date of payment in a designation of payee account (§4.2.1), they prevent payees from cheating. Collusion between a payer and a payee would of course render such measures ineffective. But a payee who tries to cheat may, in any case, find that the payer has already spent copies of the notes elsewhere.

With the approach outlined so far, the payee would learn whether the payer used a debit or a credit account. But the payer may prefer to reveal to the payee only the amount of payment, and to conceal the account type from everyone but the bank. The payer untraceability protocol could be modified slightly to accomplish this: the payer encrypts the notes and any associated information with the bank's public key; the payee forwards the encrypted payment to the bank without being able to decrypt or verify it and waits for the bank to indicate the amount deposited.

The final case involving dynamic note values is that of credit and payee untraceability. Credit is applicable only to payers. Since the payer here is not untraceable, credit charges can be handled just as in a fully traceable system. Nevertheless, if the payer pays with dynamic-value debit notes, the payee can obtain the interest earned for the period between the date of payment and the date of deposit.

5.2. Fixed Note Values

Consider again first the case of payer untraceability from a debit account. Along with each basic payer anonymity note used in a payment, the payer supplies an unsigned blinded "receipt note," which would have a form like that of message [1] in the payer untraceability protocol. When the bank accepts the payment notes for deposit, it signs the blinded receipt notes with signatures encoding the date and the corresponding denominations. The type of signature used on the receipt notes is of course different from that used on the payment notes. When the receipt notes are returned to the payer by the payee, the payer unblinds them.

To recover the interest, the payer supplies the signed receipt notes to the bank. The payer should earn interest from the date the notes were withdrawn to the date they were spent. The bank knows the date the notes were spent, since this is encoded in the receipt notes. But because of the blinding, the bank may not know the date the particular notes were withdrawn. To calculate appropriate interest on each note, though, the bank need merely consider a receipt note as matching the payer's most recent unmatched withdrawal of the same denomination. This approach gives the bank information not provided by the other protocols: namely, the amounts spent on each day.

The credit case with payer untraceability is similar. On an issue date, the bank provides the payer with notes whose value is the full limit. As in the debit case, a receipt is obtained for each note when it is spent. On a settlement date, the payer must provide the bank with the receipts and any remaining unspent notes. (In case the payer tries to cheat by returning notes and receipts having a total value that is below the limit issued, the bank considers the missing value as spent on the issue date, thereby calling for the maximum interest charge on it.) Thus the bank can compute both the amount and the interest due. The settlement date for one period could be the issue date for the next, and such dates should coincide for all payers so as to minimize linking.

Finally, consider payee untraceability. Since the payer is traceable, interest for this party can again be handled as with a traceable system. If the signatures made by the bank during withdrawal encode the date, they allow the payee's account to earn retroactive interest that is paid when the note is deposited.

6. Efficiency and Performance

This section first introduces several considerations related to the efficiency of the payment system: choices for the denomination scheme, advance provision of blinded notes to the bank, ways for payees to protect against notes being spent more than once, and economies resulting from knowing the payee or

amount before deposit. Then the space and time requirements of a large-scale consumer payment system are considered for both the consumer and the bank. Finally, some possible arrangements for multiple financial institutions participating in the same system are considered.

6.1. Denominations

It would of course be quite inconvenient to pay for everything using the correct number of pennies. On the other hand, if each note were signed by a signature type indicating the exact amount of payment, such as $31.84, then the bank could easily trace notes with unusual amounts. The introduction mentioned a compromise solution using notes of standard denominations, such as pennies, nickels, dimes, and so on.

The choice of a set of denominations is constrained by two boundary conditions: one denomination and all possible denominations. A range of denominations can reduce the number of notes needed, but untraceability suffers somewhat, because more is revealed about which payments could have come from which accounts. The choice of denominations depends not only on the desired tradeoff between these considerations, but on the expected distribution of payment amounts.

Consider, for example, denominations based on the powers of two (i.e. 1, 2, 4, 8, ... cent notes). Each denomination corresponds to a single bit of a binary representation of the amount to be paid; for each one bit in the amount, a note of the corresponding denomination is sent, and for each zero bit, nothing is sent. A set containing j notes, each of a different such denomination, is sufficient to pay any amount in exact cents from one cent up to 2^j-1 cents. For example, fourteen denominations allow payments up to $163.83, and sixteen allow payments to $655.35. Assuming uniformly distributed amounts, and that the maximal amount is 2^j-2, the average number of notes in a payment is $j/2$. Distribution of amounts, however, tends to be heavily skewed toward small amounts. It has been estimated, for example, that in 1975, half of all payments in the United States were for under a dollar <Arthur D. Little, Inc.>. Thus, the average number of notes might actually be less than $j/2$.

A useful property of powers-of-two denominations is that any payment can be made without using more than a single note of each denomination. This guarantees that a payer with l notes of each of the first k denominations can make at least l payments for amounts less than denomination $k+1$. Of course, the average number of payments possible from a wallet would in general be greater. Furthermore, a payer lacking the exact complement of denominations needed for a particular payment might pay a larger amount and receive change back. When a payer needs to receive change but wishes to remain untraceable, the payer can take the role of payee in a round of the payee anonymity protocol. If the controls of §4.1.1 or those of the next subsection are to be enforced, however, change should only be given by a deposit to and matching withdrawal from the payer's account.

6.2. Pre-Submitted Note Numbers

It might be desirable to keep consumers from being able to make payee untraceability payments. This would, for example, prevent black marketeers from obtaining such payments, thereby preventing them from avoiding possible tracing or even incrimination by a payer using the "substantiating being the payer" techniques (§4.1.1). Such a constraint cannot be enforced without special provisions, however, since a withdrawal for a payee untraceability payment can be presented to the bank as if it were for payer untraceability: the bank has no way of knowing if the blinded note presented as message [1] of the payer untraceability protocol has been supplied to the withdrawer by the payee and is in fact message [2] of the payee untraceability protocol. Such abuse can be discouraged by requiring consumer account holders to submit all blinded note numbers substantially in advance. The bank would retain the blinded notes and sign them at its leisure, returning signed copies only as requested during withdrawals. This would prevent the consumer from being able to make payee untraceability payments unless they were arranged before the notes were originally submitted. It would also increase efficiency: the payer would not need to transmit blinded notes during withdrawal, and the bank could reduce its peak signing capacity.

Some possible uses for different "types" of signatures have been mentioned in §§5 & 6.1, but it has not been necessary until now to distinguish between different ways to realize them. One way is for the bank to use a different modulus for each type of signature. This has the disadvantage that all the moduli must be stored by all users of the system, but it could reduce the untraceable party's computation somewhat, by allowing use of exponent 3 only (see §6.6). In the second and perhaps more attractive approach, the bank always uses the same modulus but different odd public exponents for each type of signature. Each signature type might, for example, be defined by a distinct prime as its public exponent.

When blinded notes are formed for advance submission to the bank, which signature type each will receive may not be known; nevertheless, the payer must be able to unblind the signed notes when they are withdrawn. A simple extension of the basic protocol allows this, provided the bank uses a single modulus and makes known a modest-sized set of possible public exponents before the blinded notes are created. For example, if $[1] = nr^B$, where $b \mid B$, and the bank forms $[2] = [1]^{\bar{b}} = n^{\bar{b}} r^{B/b}$, then the payer can easily compute the usual message $[3] = [2]r^{-B/b}$. This does require, however, at least one bit of exponent (and hence at least one extra multiply) for each signature type to be anticipated (since each public exponent should both divide B and contain at least one distinct prime factor). Thus, such a scheme may be impractical if the number of signature types needed is large. But

"unanticipated blind signature" techniques <Chaum 87> allow an unlimited number of signature types, while requiring only about as much computation as does forming two signatures.

6.3. Anticipated Payments

Sometimes the amount of payment can be anticipated in advance: regularly scheduled payments, tolls, parking, public transit fares, favorite newspapers, planned purchases, and so on. Consider a single such payment that requires one of each of several denominations to achieve the exact amount. Separate blinded notes would be withdrawn for each of these denominations; they would all be formed using the same n, but each would have an independent r. During withdrawal, the different r's would hide from the bank the fact that several notes have the same n. Since each note would have a different signature type corresponding to its denomination, the bank would be able to accept each note for deposit, even though they all use the same n. The bank can accept the full deposit after making only a single look-up in the list of cleared note numbers, because only one n is used.

A further advantage to such a collection of notes with the same n is that it can be transformed into a single note with multiple signatures, if the same modulus is used for all the denomination signature types (§6.2). This offers economy in communicating the payment to the bank, and in storing it before it is sent. Consider, for example, two notes $n^{\overline{v}}$ and $n^{\overline{w}}$, where v and w are the corresponding prime public exponents. These can be combined to form $n^{\overline{vw}}$, which would be worth the sum of the denominations. To do this, c is first set to the multiplicative inverse of w modulo v (i.e., $c \equiv w^{-1} \pmod{v}$) and d to the integer $(cw-1)/v$. Then the combined note is formed $n^{\overline{vw}} = n^{\overline{v}c} n^{-\overline{w}d}$. This can be readily generalized to any number of denominations. The same effect could of course be obtained by getting a note signed with one denomination, unblinding and reblinding it, and getting it signed with a second denomination.

Often the payee is also known in advance, which can yield further advantages. The payee can keep a list of notes that it has already accepted: if a note has a valid signature, has the payee's account designation encoded within it, and is not on the payee's list, then the payee can accept it without further verification. This is because, if the bank were to dispute the payee's balance, such notes must be counted in the payee's favor by an arbiter, as also mentioned in §4.2.2. Thus the payee never need request verification of such notes before honoring them.

6.4. Online, Offline, and Hybrid Clearing

When the payee and amount are known before withdrawal, the techniques of the previous subsection remove the need for clearing; otherwise, clearing is required. Online clearing is where the payee gives goods or services to the payer only after receiving confirmation of the deposit from the bank; other clearing will be called offline.

Offline clearing is adequate for payments against ongoing accounts, such as utilities, loans, and so on; cases in which things paid for are provided at a later time, such as down payments or prepaid orders; or payments for which nothing is directly provided in return, such as donations and contributions.

A special kind of note <Chaum, Fiat, & Naor> allows offline clearing in any payment situation where eventual detection is a sufficient deterrent to abuse. In spending such a note, the payer must answer a random numeric query; spending the note a second time would require an answer to a different query. Two answers for the same note reveal enough for a payer's account to be easily and irrefutably traced, but one answer alone does not compromise the payer's unconditional untraceability.

Where the payer receives valuable goods or information, particularly items that could easily be resold, it may be prudent to require online clearing. This may be somewhat more expensive than offline clearing in terms both of communication costs and of the capacity for computing and accessing storage that the bank would need in order to meet peak demand. A possible compromise would be for the payee to clear a randomly selected subset of payments immediately, and the rest overnight. This may be adequate for some relatively low-value transactions, where the probability of immediate and possibly on-the-spot detection (rather than the certainty of being traced later, as with the spendable-once money referenced above) outweighs the potential payoff from spending notes more than once. Alternatively, accumulated batches could be sent at random intervals—saving communication costs in a connection-oriented network while providing the benefit of random sampling. Still another variation is online clearing of the most valuable note, and deferred clearing of all but a random sampling of the lower-valued notes.

6.5. Memory Requirements

The untraceable party, the bank, and the clearing function each have different data storage needs.

6.5.1. Untraceable Party Memory

The untraceable party must hold the signed note numbers before they are provided to the payee or bank (in payer or payee untraceability, respectively). The RSA system as originally proposed recommended numbers of 200 decimal digits. Thus, there is plenty of room in such an n to encode a payee account (§§4.1.2, 4.2.2, & 6.3), and the private key needed to designate the payee (§4.2.1) could be chosen to just fit.

Storing a supply of such numbers on a home computer disk would take up only a modest amount of its capacity. But card computers, possibly no

bigger than a credit card, might be used by consumers to carry note numbers and would have far less storage capacity. Currently, 64k bit chips are widely used in the nonvolatile, electrically changeable EEPROM technology—but other chip technologies can already store megabits. Even a 64k bit chip would allow an untraceable payer to make at least 8 payments in exact cents up to $655.35 each (§6.1), and still make 3 payments for preselected arbitrary amounts (§6.3). These last could each be used to refill the card by getting change (§6.1).

6.5.2. Account Records

A fully traceable payment system would, in principle, require one access to the payer's account information and one access to the payee's for each payment. With payer anonymity, the updating of a consumer account is aggregated into a withdrawal that is capable of spanning many payments. This can give a savings approaching, in the limit, one access per transaction. Similarly, for payee anonymity, several payments may be aggregated into a single deposit transaction.

When the payee is untraceable (§3.1) or anticipated (§6.3), the payee account number is built into note numbers; thus, only the distinguishing parts of each such note number need be stored in the payee's account record. When the payee account number is not built in, however, clearing is required.

6.5.3. Clearing

If notes are encoded with expiration dates, they can be purged from the bank's list of cleared notes as they expire. Expiration dates can be encoded in the signature type, as suggested for determining interest amounts in §5. Alternatively, they can be included in otherwise unused digits of the note number itself. But if the expiration date code can be predicted too far into the future by account holders, then the bank cannot be sure how many still-valid but uncleared notes remain outstanding, because lost or destroyed notes will remain on the books indefinitely. The bank could address this problem by announcing randomly chosen 100-bit expiration-date codes only for the near future. Under either approach to obsolete notes, expired notes could be stored on archival storage media, such as magnetic tape, which have lower cost per bit stored but higher access time. The bank might assess a surcharge, possibly increasing with the note's age, on the clearing of expired notes.

Assume that already accepted note numbers are to be stored on moving-head disks. A well-known approach is to use a "hash" function of the note number to select a disk drive and disk address. With such hashing techniques, enough of the disk is left unused to provide an acceptably small ratio of "collisions," and a variety of strategies for resolving collisions are available. In principle, write-only-once memory technology could be used, because expiration dates periodically allow complete purging of current storage.

Only the random components (§4.2) need be stored, and they might be only a few hundred bits each (§2.4). Even with currently available disks, such a note number slot costs less than a quarter of a cent. For example, an IBM 3380K4 has a 7.5 gigabyte capacity, costs less than $200,000 and can make about 40 unoptimized random accesses per second with each of four arms. Bankproof payer untraceability requires precursors to be stored (§4.2.2), but archival media can be used for this, since precursors need only be accessed in case of dispute. If notes are valid for only a few months, and the cost of the disk drives is spread over several years, then the cost per transaction is below a tenth of a cent. This is far lower than the real cost of paper money, check, or credit card transactions.

Inevitably, some disks will fail. Permanent data loss could be prevented by conventional logging and backup practices. When a disk fails, the bank may wish to continue to clear notes that should be stored on that disk, in order not to inconvenience its customers. But if this situation were detected by an attacker, a large number of previously used notes whose numbers were stored on the failed disk could be cleared for a second time. This threat can be reduced by using a cryptographic hash function to decide the disk on which each note should be stored. Thus an adversary in this setting would have poor odds of successfully clearing any already cleared note. While the failed disk is being restored, all newly submitted notes that would have been stored on the failed disk would be stored on a standby disk to prevent their repeated use.

6.6. Computation Requirements

Today's microcomputers could readily perform the computation required of payers and payees in the basic system, while the bank may need special hardware. If the bank's public exponent is small, 3 for instance, then the work of the untraceable party includes only a few modular multiplies, and the even more modest modular multiplicative inverse calculation. Such computations take only a small fraction of a second on an ordinary microprocessor. To sign a blinded note, however, the bank must raise it to a large power in the modular arithmetic system. Chips currently in production and available in some countries make such signatures at roughly the rate of disk-arm accesses. Chips should continue to cost far less than disk arms, and since the quantity of notes signed is the same as that stored in clearing, signing should not add significantly to the system cost.

When withdrawal requests (§1.4) are to be used, the payer must form digital signatures. One signature per withdrawal transaction would be adequate, and it could be pre-computed, along with the blinding factors needed, well in advance of when the bank is contacted. If the untraceable-payer-designates-payee protocol (§4.2.1) is used, the creation of the public key for each note can also be done in advance, but the payer must form a digital signature during each payment.

6.7. Multiple Financial Institutions

A large-scale consumer payment system based on the approach presented here is likely to involve more than one financial institution. There might be a shared entity, called the "mint," which performs only the actual signing of notes for its member institutions, while charging each institution's account accordingly. A second shared entity, called the "clearing center," might maintain the list of cleared note numbers submitted by its members, while crediting each member's account according to that member's submissions. The clearing center need not even verify signatures on most notes; it could merely ask for signatures on a random sampling of notes. There might be more than one mint location, for robustness and economy in communication. There might also be several clearing centers: coverage of the note numbers could be divided among centers as some possibly secret function of the note numbers and duplication of coverage could provide extreme robustness.

The mint and the clearing center periodically determine the amounts of settlement between the financial institutions. The separate financial institutions manage all the rest. In particular, they maintain the accounts of individuals and other account holders, issue various receipts (§§1.4, 4.1, & 4.2.2), verify and maintain records of the designations of payee (§4.2.1) and the precursors of the bankproof payer untraceability protocol (§4.2.2), make credit decisions (§5), and possibly even set their own interest rates (§5.2).

Conclusion

Practical payment systems have been introduced that are secure for all parties and provide a variety of new possibilities for privacy in the upcoming age of digital payments.

Acknowledgements

It is a pleasure to thank all those who have commented on this paper since it was distributed at CRYPTO '84, particularly Jan-Hendrik Evertse.

References

(1) Arthur D. Little, Inc. for NSF, *The Consequences of Electronic Funds Transfer,* U.S. Government Printing Office, 038-000-00249-0, May 1975.

(2) Chaum, D., "Blind Signature Systems," Abstract in: *Advances in Cryptology: Proceedings of CRYPTO 83,* D. Chaum, ed., Plenum, p. 153.

(3) Chaum, D., "Security without Identification: Transaction Systems to make Big Brother Obsolete," *Communications of the ACM,* Vol 28, No. 10, October 1985, pp. 1030–1044.

(4) Chaum, D. and de Jonge, W., "Attacks on Some RSA Signatures," *Advances in Cryptology—CRYPTO '85,* H.C. Williams, Ed., Lecture Notes in Computer Science no. 218, Springer-Verlag, pp. 18–27.

(5) Chaum, D., "Blinding for Unanticipated Signatures," *Advances in Cryptology—EUROCRYPT '87,* D. Chaum and W.L. Price, eds., Lecture Notes in Computer Science no. 304, Springer-Verlag, pp. 277–233.

(6) Chaum, D., Fiat A. and Naor, M., "Untraceable Electronic Cash," to appear in *Advances in Cryptology—CRYPTO '88,* Lecture Notes in Computer Science, Springer-Verlag.

(7) Diffie, W. and Hellman, M., "New Directions in Cryptography," *IEEE Transaction on Information Theory,* vol. 22, no. 6, November 1976, pp. 644–654.

(8) Rivest, R., Shamir, A. and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Communications of the ACM,* vol. 21, no. 2, February 1978, pp. 120–126.