How To Make A Mint_ The Cryptography of Anonymous Electronic Cash

Anonymous: Fried, Frank got NSA's permission to make this report available. They have offered to make copies available by contacting them at <21stCen@ffhsj.com> or (202) 639-7200. See: http://www.ffhsj.com/bancmail/21starch/961017.htm

Received October 31, 1996

With the Compliments of Thomas P. Vartanian
Fried, Frank, Harris, Schriver & Jacobson
1001 Pennsylvania Avenue, N.W.
Washington, D.C. 20004-2505
Telephone: (202) 639-7200

HOW TO MAKE A MINT: THE CRYPTOGRAPHY OF ANONYMOUS ELECTRONIC CASH

Laurie Law, Susan Sabett, Jerry Solinas
National Security Agency Office of Information Security Research and Technology
Cryptology Division
18 June 1996

CONTENTS

INTRODUCTION

With the onset of the Information Age, our nation is becoming increasingly dependent upon network communications. Computer-based technology is significantly impacting our ability to access, store, and distribute information. Among the most important uses of this technology is *electronic commerce*: performing financial transactions via electronic information exchanged over telecommunications lines. A key requirement for electronic commerce is the development of secure and efficient electronic payment systems. The need for security is highlighted by the rise of the Internet, which promises to be a leading medium for future electronic commerce.

Electronic payment systems come in many forms including digital checks, debit cards, credit cards, and stored value cards.

The usual security features for such systems are *privacy* (protection from eavesdropping), *authenticity* (provides user identification and message integrity), and *nonrepudiation* (prevention of later denying having performed a transaction) .

The type of electronic payment system focused on in this paper is *electronic cash*. As the name implies, electronic cash is an attempt to construct an electronic payment system modelled after our paper cash system. Paper cash has such features as being: portable (easily carried), recognizable (as legal tender) hence readily acceptable, transferable (without involvement of the financial network), untraceable (no record of where money is spent), anonymous (no record of who spent the money) and has the ability to make "change." The designers of electronic cash focused on preserving the features of untraceability and anonymity. Thus, electronic cash is defined to be an electronic payment system that provides, in addition to the above security features, the properties of user anonymity and payment untraceability..

In general, electronic cash schemes achieve these security goals via *digital signatures*. They can be considered the digital analog to a handwritten signature. Digital signatures are based on *public key cryptography*. In such a cryptosystem, each user has a secret key and a public key. The secret key is used to create a digital signature and the public key is needed to verify the digital signature. To tell who has signed the information (also called the message), one must be certain one knows who owns a given public key. This is the problem of key management, and its solution requires some kind of authentication infrastructure. In addition, the system must have adequate network and physical security to safeguard the secrecy of the secret keys.

This report has surveyed the academic literature for cryptographic techniques for implementing secure electronic cash systems. Several innovative payment schemes providing user anonymity and payment untraceability have been found. Although no particular payment system has been thoroughly analyzed, the cryptography itself appears to be sound and to deliver the promised anonymity.

These schemes are far less satisfactory, however, from a law enforcement point of view. In particular, the dangers of money laundering and counterfeiting are potentially far more serious than with paper cash. These problems exist in any electronic payment system, but they are made much worse by the presence of anonymity. Indeed, the widespread use of electronic cash would increase the vulnerability of the national financial system to Information Warfare attacks. We discuss measures to manage these risks; these steps, however, would have the effect of limiting the users' anonymity.

This report is organized in the following manner. Chapter 1 defines the basic concepts surrounding electronic payment systems and electronic cash. Chapter 2 provides the reader with a high level cryptographic description of electronic cash protocols in terms of basic authentication mechanisms. Chapter 3 technically describes specific implementations that have been proposed in the academic literature. In Chapter 4, the optional features of transferability and divisibility for off-line electronic cash are presented. Finally, in Chapter 5 the security issues associated with electronic cash are discussed.

The authors of this paper wish to acknowledge the following people for their contribution to this research effort through numerous discussions and review of this paper: Kevin Igoe, John Petro, Steve Neal, and Mel Currie.

1. WHAT IS ELECTRONIC CASH?

We begin by carefully defining "electronic cash." This term is often applied to any electronic payment scheme that superficially resembles cash to the user. In fact, however, electronic cash is a specific kind of electronic payment scheme, defined by certain cryptographic properties. We now focus on these properties.

1.1 Electronic Payment

The term *electronic commerce* refers to any financial transaction involving the electronic transmission of information. The packets of information being transmitted are commonly called *electronic tokens*. One should not confuse the token, which is a sequence of bits, with the physical media used to store and transmit the information.

We will refer to the storage medium as a *card* since it commonly takes the form of a wallet-sized card made of plastic or cardboard. (Two obvious examples are credit cards and ATM cards.) However, the "card" could also be, e.g., a computer memory.

A particular kind of electronic commerce is that of *electronic payment*. An electronic payment protocol is a series of transactions, at the end of which a payment has been made, using a token issued by a third party. The most common example is that of credit cards when an electronic approval process is used. Note that our definition implies that neither payer nor payee issues the token.l

The electronic payment scenario assumes three kinds of players:2

- a *payer* or consumer, whom we will name Alice.
- a *payee*, such as a merchant. We will name the payee Bob.
- a *financial network* with whom both Alice and Bob have accounts. We will informally refer to the financial network as the Bank.

---

1 In this sense, electronic payment differs from such systems as prepaid phone cards and subway fare cards, where the token is issued by the payee.

2 In 4.1, we will generalize this scenario when we discuss transfers.

1.2 Security of Electronic Payments

With the rise of telecommunications and the Internet, it is increasingly the case that electronic commerce takes place using a transmission medium not under the control of the financial system. It is therefore necessary to take steps to insure the security of the messages sent along such a medium.

The necessary security properties are:

- *Privacy*, or protection against eavesdropping. This is obviously of importance for transactions involving, *e.g.*, credit

card numbers sent on the Internet.

- *User identification*, or protection against impersonation. Clearly, any scheme for electronic commerce must require that a user knows with whom she is dealing (if only as an alias or credit card number).
- *Message integrity*, or protection against tampering or substitution. One must know that the recipient's copy of the message is the same as what was sent.
- *Nonrepudiation*, or protection against later denial of a transaction. This is clearly necessary for electronic commerce, for such things as digital receipts and payments.

The last three properties are collectively referred to as *authenticity*.

These security features can be achieved in several ways. The technique that is gaining widespread use is to employ an *authentication infrastructure*. In such a setup, privacy is attained by enciphering each message, using a private key known only to the sender and recipient. The authenticity features are attained via *key management*, *e.g.*, the system of generating, distributing and storing the users' keys.

Key management is carried out using a *certification authority*, or a trusted agent who is responsible for confirming a user's identity. This is done for each user (including banks) who is issued a digital *identity certificate*. The certificate can be used whenever the user wishes to identify herself to another user. In addition, the certificates make it possible to set up a private key between users in a secure and authenticated way. This private key is then used to encrypt subsequent messages. This technique can be implemented to provide any or all of the above security features.

Although the authentication infrastructure may be separate from the electronic-commerce setup, its security is an essential component of the security of the electronic-commerce system. Without a trusted certification authority and a secure infrastructure, the above four security features cannot be achieved, and electronic commerce becomes impossible over an untrusted transmission medium.

We will assume throughout the remainder of this paper that some authentication infrastructure is in place, providing the four security features.

1.3 Electronic Cash

We have defined privacy as protection against eavesdropping on one's communications. Some privacy advocates such as David Chaum (see [2],[3]), however, define the term far more expansively. To them, genuine "privacy" implies that one's history of purchases not be available for inspection by banks and credit card companies (and by extension the government). To achieve this, one needs not just privacy but *anonymity*. In particular, one needs

- *payer anonymity* during payment,
- *payment untraceability* so that the Bank cannot tell whose money is used in a particular payment.

These features are not available with credit cards. Indeed, the only conventional payment system offering it is cash. Thus Chaum and others have introduced *electronic cash* (or *digital cash*), an electronic payment system which offers both features. The sequence of events in an electronic cash payment is as follows:

- *withdrawal*, in which Alice transfers some of her wealth from her Bank account to her card.
- *payment*, in which Alice transfers money from her card to Bob's.
- *deposit*, in which Bob transfers the money he has received to his Bank account.

(See Figure 1.)

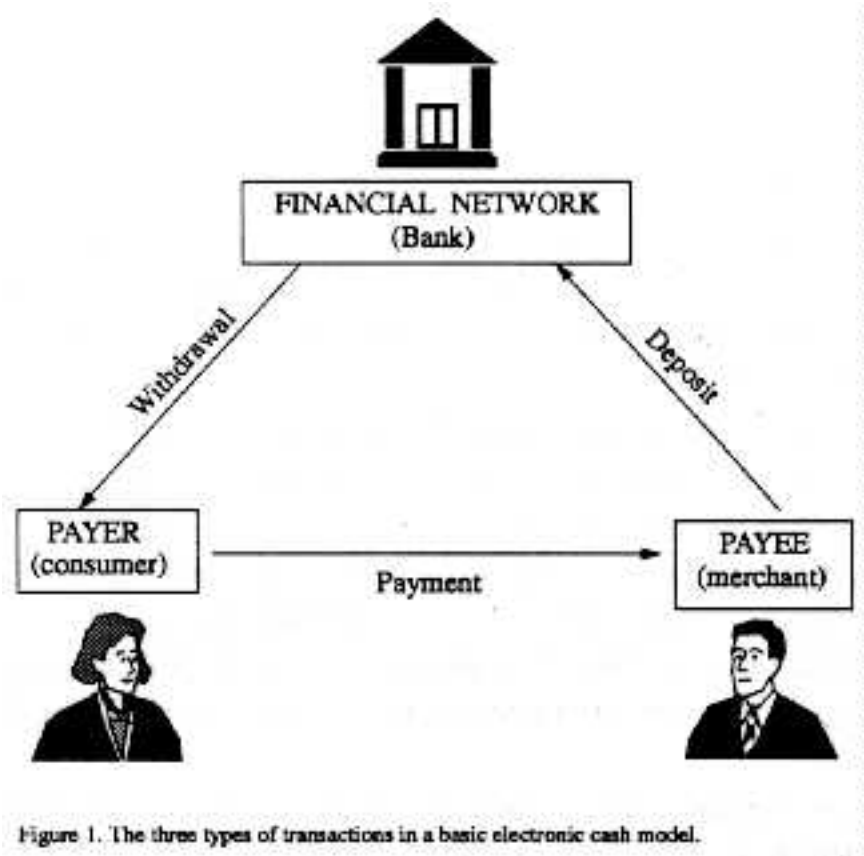

Figure 1. The three types of transactions in a basic electronic cash model.

Figure 1. The three types of transactions in a basic electronic cash model.

These procedures can be implemented in either of two ways:

- *On-line* payment means that Bob calls the Bank and verifies the validity of Alice's token3 before accepting her payment and delivering his merchandise. (This resembles many of today's credit card transactions.)
- *Off-line* payment means that Bob submits Alice's electronic coin for verification and deposit sometime after the payment transaction is completed. (This method resembles how we make small purchases today by personal check.)

Note that with an on-line system, the payment and deposit are not separate steps. We will refer to *on-line cash* and *off-line cash* schemes, omitting the word "electronic" since there is no danger of confusion with paper cash.

3 In the context of electronic cash, the token is usually called an electronic coin.

1.4 Counterfeiting

As in any payment system, there is the potential here for criminal abuse, with the intention either of cheating the financial system or using the payment mechanism to facilitate some other crime. We will discuss some of these problems in 5. However, the issue of *counterfeiting* must be considered here, since the payment protocols contain built-in protections against it.

There are two abuses of an electronic cash system analogous to counterfeiting of physical cash:

- *Token forgery*, or creating a valid-looking coin without making a corresponding Bank withdrawal.
- *Multiple spending*, or using the same token over again. Since an electronic coin consists of digital information, it is as valid-looking after it has been spent as it was before. (Multiple spending is also commonly called *re-spending*, *double spending*, and *repeat spending*.)

One can deal with counterfeiting by trying to *prevent* it from happening, or by trying to *detect* it after the fact in a way that identifies the culprit. Prevention clearly is preferable, all other things being equal.

Although it is tempting to imagine electronic cash systems in which the transmission and storage media are secure, there will certainly be applications where this is not the case. (An obvious example is the Internet, whose users are notoriously vulnerable to viruses and eavesdropping.) Thus we need techniques of dealing with counterfeiting other than physical security.

- To protect against token forgery, one relies on the usual *authenticity* functions of user identification and message integrity. (Note that the "user" being identified from the coin is the issuing Bank, not the anonymous spender.)
- To protect against multiple spending, the Bank maintains a database of spent electronic coins. Coins already in the database are to be rejected for deposit. If the payments are on-line, this will prevent multiple spending. If off-line, the best we can do is to detect when multiple spending has occurred. To protect the payee, it is then necessary to identify the payer.

Thus it is necessary to disable the anonymity mechanism in the case of multiple spending.

The features of authenticity, anonymity, and multiple-spender exposure are achieved most conveniently using public-key cryptography. We will discuss how this is done in the next two chapters.

2. A CRYPTOGRAPHIC DESCRIPTION

In this chapter, we give a high-level description of electronic cash protocols in terms of basic authentication mechanisms. We begin by describing these mechanisms, which are based on public-key cryptography. We then build up the protocol gradually for ease of exposition. We start with a simplified scheme which provides no anonymity. We then incorporate the payment untraceability feature, and finally the payment anonymity property. The result will be a complete electronic cash protocol.

2.1 Public-Key Cryptographic Tools

We begin by discussing the basic public-key cryptographic techniques upon which the electronic cash implementations are based.

One-Way Functions. A *one-way function* is a correspondence between two sets which can be computed efficiently in one direction but not the other. In other words, the function *phi* is one-way if, given *s* in the domain of *phi*, it is easy to compute t = *phi*(*s*), but given only *t*, it is hard to find *s*. (The elements are typically numbers, but could also be, *e.g.*, points on an elliptic curve; see [10].)

Key Pairs. If *phi* is a one-way function, then a *key pair* is a pair *s*, *t* related in some way via *phi*. We call *s* the *secret key* and *t* the *public key*. As the names imply, each user keeps his secret key to himself and makes his public key available to all. The secret key remains secret even when the public key is known, because the one-way property of *phi* insures that *t* cannot be computed from *s*.

All public-key protocols use key pairs. For this reason, public-key cryptography is often called *asymmetric cryptography*. Conventional cryptography is often called *symmetric cryptography*, since one can both encrypt and decrypt with the private key but do neither without it.

Signature and Identification. In a public key system, a user identifies herself by proving that she knows her secret key without revealing it. This is done by performing some operation using the secret key which anyone can check or undo using the public key. This is called *identification*. If one uses a message as well as one's secret key, one is performing a *digital signature* on the message. The digital signature plays the same role as a handwritten signature: identifying the author of the message in a way which cannot be repudiated, and confirming the integrity of the message.

Secure Hashing. A *hash function* is a map from all possible strings of bits of any length to a bit string of fixed length. Such functions are often required to be *collision-free*: that is, it must be computationally difficult to find two inputs that hash to the same value. If a hash function is both one-way and collision-free, it is said to be a *secure hash*.

The most common use of secure hash functions is in digital signatures. Messages might come in any size, but a given public-key algorithm requires working in a set of fixed size. Thus one hashes the message and signs the secure hash rather than the message itself. The hash is required to be one-way to prevent *signature forgery*, *i.e.*, constructing a valid-looking signature of a message without using the secret key.4 The hash must be collision-free to prevent *repudiation*, *i.e.*, denying having signed one message by producing another message with the same hash.

---

4 Note that *token forgery* is not the same thing as *signature forgery*. Forging the Bank's digital signature without knowing its secret key is one way of committing token forgery, but not the only way. A bank employee or hacker, for instance, could "borrow" the Bank's secret key and validly sign a token. This *key compromise* scenario is discussed in 5.3.

2.2 A Simplified Electronic Cash Protocol

We now present a simplified electronic cash system, without the anonymity features.

PROTOCOL 1: *On-line electronic payment.*

Withdrawal:

Alice sends a withdrawal request to the Bank.
Bank prepares an electronic coin and digitally signs it.
Bank sends coin to Alice and debits her account.

Payment/Deposit:

Alice gives Bob the coin.
Bob contacts Bank5 and sends coin.
Bank verifies the Bank's digital signature.
Bank verifies that coin has not already been spent.
Bank consults its withdrawal records to confirm Alice's withdrawal. *(optional)*
Bank enters coin in spent-coin database.
Bank credits Bob's account and informs Bob.
Bob gives Alice the merchandise.

---

5 One should keep in mind that the term "Bank" refers to the financial system that issues and clears the coins. For example, the Bank might be a credit card company, or the overall banking system. In the latter case, Alice and Bob might have

separate banks. If that is so, then the "deposit" procedure is a little more complicated: Bob's bank contacts Alice's bank, "cashes in" the coin, and puts the money in Bob's account.

PROTOCOL 2: *Off-line electronic payment.*

Withdrawal:
Alice sends a withdrawal request to the Bank.
Bank prepares an electronic coin and digitally signs it.
Bank sends coin to Alice and debits her account.

Payment:
Alice gives Bob the coin.
Bob verifies the Bank's digital signature. *(optional)*
Bob gives Alice the merchandise.

Deposit:
Bob sends coin to the Bank.
Bank verifies the Bank's digital signature.
Bank verifies that coin has not already been spent.
Bank consults its withdrawal records to confirm Alice's withdrawal. (optional)
Bank enters coin in spent-coin database.
Bank credits Bob's account.

The above protocols use digital signatures to achieve authenticity. The authenticity features could have been achieved in other ways, but we need to use digital signatures to allow for the anonymity mechanisms we are about to add.

2.3 Untraceable Electronic Payments

In this section, we modify the above protocols to include payment untraceability. For this, it is necessary that the Bank not be able to link a specific withdrawal with a specific deposit.6 This is accomplished using a special kind of digital signature called a *blind signature*.

We will give examples of blind signatures in 3.2, but for now we give only a high-level description. In the withdrawal step, the user changes the message to be signed using a random quantity. This step is called "blinding" the coin, and the random quantity is called the *blinding factor*. The Bank signs this random-looking text, and the user removes the blinding factor. The user now has a legitimate electronic coin signed by the Bank. The Bank will see this coin when it is submitted for deposit, but will not know who withdrew it since the random blinding factors are unknown to the Bank. (Obviously, it will no longer be possible to do the checking of the withdrawal records that was an optional step in the first two protocols.)

Note that the Bank does not know what it is signing in the withdrawal step. This introduces the possibility that the Bank might be signing something other than what it is intending to sign. To prevent this, we specify that a Bank's digital signature by a given secret key is valid only as authorizing a withdrawal of a fixed amount. For example, the Bank could have one key for a $10 withdrawal, another for a $50 withdrawal, and so on.7

_____

6 In order to achieve either anonymity feature, it is of course necessary that the pool of electronic coins be a large one.

7 0ne could also broaden the concept of "blind signature" to include interactive protocols where both parties contribute random elements to the message to be signed. An example of this is the "randomized blind signature" occurring in the Ferguson scheme discussed in 3.3.

PROTOCOL 3: *Untraceable On-line electronic payment.*

Withdrawal:
Alice creates an electronic coin and blinds it.
Alice sends the blinded coin to the Bank with a withdrawal request.
Bank digitally signs the blinded coin.
Bank sends the signed blinded coin to Alice and debits her account.
Alice unblinds the signed coin.

Payment/Deposit:
Alice gives Bob the coin.
Bob contacts Bank and sends coin.
Bank verifies the Bank's digital signature.
Bank verifies that coin has not already been spent.
Bank enters coin in spent-coin database.
Bank credits Bob's account and informs Bob.
Bob gives Alice the merchandise.

PROTOCOL 4: *Untraceable Off-line electronic payment.*

Withdrawal:
Alice creates an electronic coin and blinds it.
Alice sends the blinded coin to the Bank with a withdrawal request.

Bank digitally signs the blinded coin.
Bank sends the signed blinded coin to Alice and debits her account.
Alice unblinds the signed coin.

Payment:
Alice gives Bob the coin.
Bob verifies the Bank's digital signature. *(optional)*
Bob gives Alice the merchandise.

Deposit:
Bob sends coin to the Bank.
Bank verifies the Bank's digital signature.
Bank verifies that coin has not already been spent.
Bank enters coin in spent-coin database.
Bank credits Bob's account.

2.4 A Basic Electronic Cash Protocol

We now take the final step and modify our protocols to achieve payment anonymity. The ideal situation (from the point of view of privacy advocates) is that neither payer nor payee should know the identity of the other. This makes remote transactions using electronic cash totally anonymous: no one knows where Alice spends her money and who pays her.

It turns out that this is too much to ask: there is no way in such a scenario for the consumer to obtain a signed receipt. Thus we are forced to settle for *payer anonymity*.

If the payment is to be on-line, we can use Protocol 3 (implemented, of course, to allow for payer anonymity). In the off-line case, however, a new problem arises. If a merchant tries to deposit a previously spent coin, he will be turned down by the Bank, but neither will know who the multiple spender was since she was anonymous. Thus it is necessary for the Bank to be able to identify a multiple spender. This feature, however, should preserve anonymity for law-abiding users.

The solution is for the payment step to require the payer to have, in addition to her electronic coin, some sort of *identifying information* which she is to share with the payee. This information is split in such a way that any one piece reveals nothing about Alice's identity, but any two pieces are sufficient to fully identify her.

This information is created during the withdrawal step. The withdrawal protocol includes a step in which the Bank verifies that the information is there and corresponds to Alice and to the particular coin being created. (To preserve payer anonymity, the Bank will not actually see the information, only verify that it is there.) Alice carries the information along with the coin until she spends it.

At the payment step, Alice must reveal one piece of this information to Bob. (Thus only Alice can spend the coin, since only she knows the information.) This revealing is done using a *challenge-response protocol*. In such a protocol, Bob sends Alice a random "challenge" quantity and, in response, Alice returns a piece of identifying information. (The challenge quantity determines which piece she sends.) At the deposit step, the revealed piece is sent to the Bank along with the coin. If all goes as it should, the identifying information will never point to Alice. However, should she spend the coin twice, the Bank will eventually obtain two copies of the same coin, each with a piece of identifying information. Because of the randomness in the challenge-response protocol, these two pieces will be different. Thus the Bank will be able to identify her as the multiple spender. Since only she can dispense identifying information, we know that her coin was not copied and re-spent by someone else.

PROTOCOL 5: *Off-line cash.*

Withdrawal:
Alice creates an electronic coin, including identifying information.
Alice blinds the coin.
Alice sends the blinded coin to the Bank with a withdrawal request.
Bank verifies that the identifying information is present.
Bank digitally signs the blinded coin.
Bank sends the signed blinded coin to Alice and debits her account.
Alice unblinds the signed coin.

Payment:
Alice gives Bob the coin.
Bob verifies the Bank's digital signature.
Bob sends Alice a challenge.
Alice sends Bob a response (revealing one piece of identifying info).
Bob verifies the response.
Bob gives Alice the merchandise.

Deposit:
Bob sends coin, challenge, and response to the Bank.
Bank verifies the Bank's digital signature.
Bank verifies that coin has not already been spent.
Bank enters coin, challenge, and response in spent-coin database.
Bank credits Bob's account.

Note that, in this protocol, Bob must verify the Bank's signature before giving Alice the merchandise. In this way, Bob can be

sure that either he will be paid or he will learn Alice's identity as a multiple spender.

3. PROPOSED OFF-LINE IMPLEMENTATIONS

Having described electronic cash in a high-level way, we now wish to describe the specific implementations that have been proposed in the literature. Such implementations are for the off-line case; the on-line protocols are just simplifications of them. The first step is to discuss the various implementations of the public-key cryptographic tools we have described earlier.

3.1 Including Identifying Information

We must first be more specific about how to include (and access when necessary) the identifying information meant to catch multiple spenders. There are two ways of doing it: the *cut-and-choose* method and *zero-knowledge proofs*.

Cut and Choose. When Alice wishes to make a withdrawal, she first constructs and blinds a message consisting of K pairs of numbers, where K is large enough that an event with probability 2^{-K} will never happen in practice. These numbers have the property that one can identify Alice given both pieces of a pair, but unmatched pieces are useless. She then obtains signature of this blinded message from the Bank. (This is done in such a way that the Bank can check that the K pairs of numbers are present and have the required properties, despite the blinding.)

When Alice spends her coins with Bob, his challenge to her is a string of K random bits. For each bit, Alice sends the appropriate piece of the corresponding pair. For example, if the bit string starts 0110. . ., then Alice sends the first piece of the first pair, the second piece of the second pair, the second piece of the third pair, the first piece of the fourth pair, etc. When Bob deposits the coin at the Bank, he sends on these K pieces.

If Alice re-spends her coin, she is challenged a second time. Since each challenge is a random bit string, the new challenge is bound to disagree with the old one in at least one bit. Thus Alice will have to reveal the other piece of the corresponding pair. When the Bank receives the coin a second time, it takes the two pieces and combines them to reveal Alice's identity.

Although conceptually simple, this scheme is not very efficient, since each coin must be accompanied by $2K$ large numbers.

Zero-Knowledge Proofs. The term *zero-knowledge proof* refers to any protocol in public-key cryptography that proves knowledge of some quantity without revealing it (or making it any easier to find it). In this case, Alice creates a key pair such that the secret key points to her identity. (This is done in such a way the Bank can check via the public key that the secret key in fact reveals her identity, *despite the blinding*.) In the payment protocol, she gives Bob the public key as part of the electronic coin. She then proves to Bob via a zero-knowledge proof that she possesses the corresponding secret key. If she responds to two distinct challenges, the identifying information can be put together to reveal the secret key and so her identity.

3.2 Authentication and Signature Techniques

Our next step is to describe the digital signatures that have been used in the implementations of the above protocols, and the techniques that have been used to include identifying information.

There are two kinds of digital signatures, and both kinds appear in electronic cash protocols. Suppose the signer has a key pair and a message M to be signed.

- *Digital Signature with Message Recovery*. For this kind of signature, we have a signing function S_{SK} using the secret key SK, and a verifying function V_{PK} using the public key PK. These functions are inverses, so that

(*) $V_{PK}(S_{SK}(M)) = M$

- The function V_{PK} is easy to implement, while S_{SK} is easy if one knows SK and difficult otherwise. Thus S_{SK} is said to have a *trapdoor,* or secret quantity that makes it possible to perform a cryptographic computation which is otherwise infeasible. The function V_{PK} is called a *trapdoor one-way function*, since it is a one-way function to anyone who does not know the trapdoor.
- In this kind of scheme, the verifier receives the signed message $S_{SK}(M)$ but not the original message text. The verifier then applies the verification function V_{PK}. This step both verifies the identity of the signer and, by (*), recovers the message text.

- *Digital Signature with Appendix*. In this kind of signature, the signer performs an operation on the message using his own secret key. The result is taken to be the signature of the message; it is sent along as an appendix to the message text. The verifier checks an equation involving the message, the appendix, and the signer's public key. If the equation checks, the verifier knows that the signer's secret key was used in generating the signature.

We now give specific algorithms.

RSA Signatures. The most well-known signature with message recovery is the RSA signature. Let N be a hard-to-factor integer. The secret signature key s and the public verification key v are exponents with the property that

$$M^{sv} = M \pmod N$$

for all messages M. Given v, it is easy to find s if one knows the factors of N but difficult otherwise. Thus the "vth power (mod N)" map is a trapdoor one-way function. The signature of M is

$$C := M^s \pmod N;$$

to recover the message (and verify the signature), one computes

$M := Cv \pmod N$.

Blind RSA Signatures. The above scheme is easily blinded. Suppose that Alice wants the Bank to produce a blind signature of the message M. She generates a random number r and sends

$rv \cdot M \pmod N$

to the Bank to sign. The Bank does so, returning

$r \cdot Ms \pmod N$

Alice then divides this result by r. The result is $Ms \pmod N$, the Bank's signature of M, even though the Bank has never seen M.

The Schnorr Algorithms. The Schnorr family of algorithms includes an identification procedure and a signature with appendix. These algorithms are based on a zero-knowledge proof of possession of a secret key. Let p and q be large prime numbers with q dividing $p - 1$. Let g be a generator; that is, an integer between 1 and p such that

$gq = 1 \pmod p$.

If s is an integer (mod q), then the *modular exponentiation* operation on s is

$phi : s \text{ -> } gs \pmod p$.

The inverse operation is called the *discrete logarithm* function and is denoted

log$g$ t <- t.

If p and q are properly chosen, then modular exponentiation is a one-way function. That is, it is computationally infeasible to find a discrete logarithm.
Now suppose we have a line

(**) $y = mx + b$

over the field of integers (mod q). A line can be described by giving its slope m and intercept b, but we will "hide" it as follows. Let

$c = gb \pmod p$,

$n = gm \pmod p$.

Then c and n give us the "shadow" of the line under *phi*. Knowing c and n doesn't give us the slope or intercept of the line, but it does enable us to determine whether a given point (x, y) is on the line. For if (x, y) satisfies (**), then it must also satisfy the relation

(***) $gy = nx \cdot c \pmod p$.

(Conversely, any point (x, y) satisfying (***) must be on the line.) The relationship (***) can be checked by anyone, since it involves only public quantities. Thus anyone can check whether a given point is on the line, but points on the line can only be generated by someone who knows the secret information.
The basic Schnorr protocol is a zero-knowledge proof that one possesses a given secret quantity m. Let n be the corresponding public quantity. Suppose one user (the "prover") wants to convince another (the "verifier") that she knows m without revealing it. She does this by constructing a line (**) and sending its shadow to the verifier. The slope of the line is taken to be secret quantity m, and the prover chooses the intercept at random, differently for each execution of the protocol. The protocol then proceeds as follows.

Schnorr proof of possession:

1. Alice sends c (and n if necessary) to Bob.
2. Bob sends Alice a "challenge" value of x.
3. Alice responds with the value of y such that (x, y) is on the line.
4. Bob verifies via (**) that (x, y) is on the line.

Bob now knows that he is speaking with someone who can generate points on the line. Thus this party must know the slope of the line, which is the secret quantity m.
An important feature of this protocol is that it can be performed only once per line. For if he knows any two points (xo, yo) and (x1, y1) on the line, the verifier can compute the slope of the line using the familiar "rise over the run" formula

$m = y$o - y1 / x1 - x1 (mod q),

and this slope is the secret quantity *m*. That is why a new intercept must be generated each time. We call this the *two-points-on-a-line principle*. This feature will be useful for electronic cash protocols, since we want to define a spending procedure which reveals nothing of a secret key if used once per coin, but reveals the key if a coin is spent twice.

Schnorr identification. The above protocol can be used for identification of users in a network. Each user is issued a key pair, and each public key is advertised as belonging to a given user. To identify herself, a user needs only prove that she knows her secret key. This can be done using the above zero-knowledge proof, since her public key is linked with her identity.

Schnorr Signature. It is easy to convert the Schnorr identification protocol to produce a digital signature scheme. Rather than receiving a challenge from an on-line verifier, the signer simply takes *x* to be a secure hash of the message and of the shadow of the line. This proves knowledge of his secret key in a way that links his key pair to the message.

Blind Schnorr Signature. Suppose that Alice wants to obtain a blind Schnorr signature for her coin, which she will spend with Bob. Alice generates random quantities (mod *q*) which describe a change of variables. This change of variables replaces the Bank's hidden line with another line, and the point on the Bank's line with a point on the new line. When Bob verifies the Bank's signature, he is checking the new point on the new line. The two lines have the same slope, so that the Bank's signature will remain valid. When the Bank receives the coin for deposit, it will see the protocol implemented on the new line, but it will not be able to link the coin with Alice's withdrawal since only Alice knows the change of variables relating the two lines.

Chaum-Pederson Signature. A variant of Schnorr's signature scheme given in [6] is used in electronic cash protocols. This modified scheme is a kind of "double Schnorr" scheme. It involves a single line and point but uses two shadows. This signature scheme can be blinded in a way similar to the ordinary Schnorr signature.

Implementations of the Schnorr Protocols. We have described the Schnorr algorithms in terms of integers modulo a prime *p*. The protocols, however, work in any setting in which the analogue of the discrete logarithm problem is difficult. An important example is that of *elliptic curves* (see [10]). Elliptic curve based protocols are much faster, and require the transmission of far less data, than non-elliptic protocols giving the same level of security.

3.3 Summary of Proposed Implementations

We can now present summaries of the main off-line cash schemes from the academic literature. There are three: those of Chaum-Fiat-Naor [4], Brands [1], and Ferguson [9].

Chaum-Fiat-Naor. This was the first electronic cash scheme, and is the simplest conceptually. The Bank creates an electronic coin by performing a blind RSA signature to Alice's withdrawal request, after having verified interactively that Alice has included her identifying information on the coin. The prevention of multiple spending is accomplished by the cut-and-choose method. For this reason, this scheme is relatively inefficient.

Brands. Brands' scheme is Schnorr-based.8 Indeed, a Schnorr protocol is used *twice:* at withdrawal, the Bank performs a blind Chaum-Pederson signature, and then Alice performs a Schnorr possession proof as the challenge-and-response part of the spending protocol.

The withdrawal step produces a coin which contains the Bank's signature, authenticating both Alice's identifying information and the shadow of the line to be used for the possession proof. This commits Alice to using that particular line in the spending step. If she re-spends the coin, she must use the same line twice, enabling the Bank to identify her.

The Brands scheme is considered by many to be the best of the three, for two reasons. First, it avoids the awkward cut-and-choose technique. Second, it is based only on the Schnorr protocols, and so it can be implemented in various settings such as elliptic curves.

Ferguson. Ferguson's scheme is RSA-based like Chaum-Fiat-Naor, but it uses the "two-points-on-a-line" principle like Brands. The signature it uses is not the blind RSA signature as described above, but a variant called a *randomized blind RSA signature*. The ordinary blind RSA scheme has the drawback that the Bank has absolutely no idea what it is signing. As mentioned above, this is not a problem in the cut-and-choose case, but in this case it can allow a payer to defeat the mechanism for identifying multiple spenders. The randomized version avoids this problem by having both Alice and the Bank contribute random data to the message. The Bank still doesn't know what it is signing, but it knows that the data was not chosen maliciously.

The rest of the protocol is conceptually similar to Brands' scheme. The message to be signed by the Bank contains, in addition to the random data, the shadow of a line whose slope and intercept reveal Alice's identity. During payment, Alice reveals a point on this line; if she does so twice, the Bank can identify her.

Although Ferguson's scheme avoids the cut-and-choose technique, it is the most complicated of the three (due largely to the randomized blind RSA signature). Moreover, it cannot be implemented over elliptic curves since it is RSA-based.

8 For ease of exposition, we give a simplified account of Brands' protocol.

4. OPTIONAL FEATURES OF OFF-LINE CASH

Much of the recent literature on off-line cash has focused on adding features to make it more convenient to use. In this

chapter we will discuss two of these features.

4.1 Transferability

Transferability is a feature of paper cash that allows a user to spend a coin that he has just received in a payment without having to contact the Bank in between. We refer to a payment as a *transfer* if the payee can use the received coin in a subsequent payment. A payment system is *transferable* if it allows at least one transfer per coin. Figure 2 shows a maximum length path of a coin in a system which allows two transfers. The final payment is not considered a transfer because it must be deposited by the payee. Transferability would be a convenient feature for an off-line cash system because it requires less interaction with the Bank. (A transferable electronic cash system is off-line by definition, since on-line systems require communication with the Bank during each payment.)

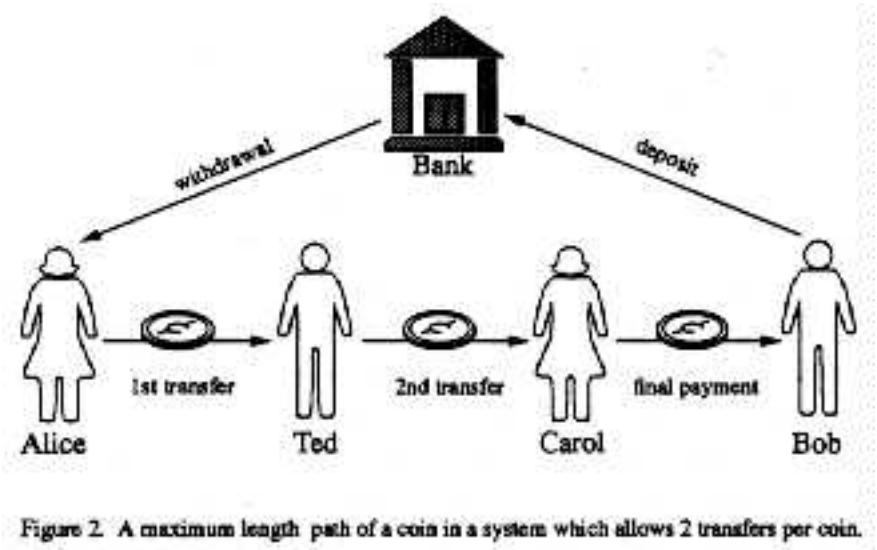

Figure 2. A maximum length path of a coin in a system which allows 2 transfers per coin.

Transferable systems have received little attention in academic literature. The schemes presented in 3.3 are not transferable because the payee cannot use a received coin in another payment - his only options are to deposit or to exchange it for new coins at the Bank. Any transferable electronic cash system has the property that the coin must "grow in size" (i.e., accumulate more bits) each time it is spent. This is because the coin must contain information about every person who has spent it so that the Bank maintains the ability to identify multiple spenders. (See [5].) This growth makes it impossible to allow an unlimited number of transfers. The maximum number of transfers allowed in any given system will be limited by the allowable size of the coin.

There are other concerns with any transferable electronic cash system, even if the number of transfers per coin is limited, and we remove the anonymity property. Until the coin is deposited, the only information available to the Bank is the identity of the individual who originally withdrew the coin. Any other transactions involving that withdrawal can only be reconstructed with the cooperation of each consecutive spender of that coin. This poses the same problems that paper cash poses for detecting money laundering and tax evasion: no records of the transactions are available.

In addition, each transfer delays detection of re-spent or forged coins. Multiple spending will not be noticed until two copies of the same coin are eventually deposited. By then it may be too late to catch the culprit, and many users may have accepted counterfeit coins. Therefore, detection of multiple spending after the fact may not provide a satisfactory solution for a transferable electronic cash system. A transferable system may need to rely on physical security to prevent multiple spending. (See 5.1.)

4.2 Divisibility

Suppose that Alice is enrolled in a non-transferable, off-line cash system, and she wants to purchase an item from Bob that costs, say, $4.99. If she happens to have electronic coins whose values add up to exactly $4.99 then she simply spends these coins. However, unless Alice has stored a large reserve of coins of each possible denomination, it is unlikely that she will have the exact change for most purchases. She may not wish to keep such a large reserve of coins on hand for the some of the same reasons that one doesn't carry around a large amount of cash: loss of interest and fear of the cash being stolen or lost. Another option is for Alice to withdraw a coin of the exact amount for each payment, but that requires interaction with the Bank, making the payment on-line from her point of view. A third option is for Bob to pay Alice the difference between her payment and the $4.99 purchase price. This puts the burden of having an exact payment on Bob, and also requires Alice to contact the Bank to deposit the "change."

A solution to Alice's dilemma is to use *divisible* coins: coins that can be "divided" into pieces whose total value is equal to the value of the original coin. This allows exact off-line payments to be made without the need to store a supply of coins of different denominations. Paper cash is obviously not divisible, but lack of divisibility is not as much of an inconvenience with paper cash because it is transferable. Coins that are received in one payment can be used again in the next payment, so the supply of different denominations is partially replenished with each transaction. (Imagine how quickly a cashier would run out of change if paper cash were not transferable and each payment was put in a separate bin set aside for the next bank deposit!)

Three divisible off-line cash schemes have been proposed, but at a cost of a longer transaction time and additional storage. Eng and Okamoto's divisible scheme [7] is based on the "cut and choose" method. Okamoto [11] is much more efficient and is based on Brands' scheme but will also work on Ferguson's scheme. Okamoto and Ohta [12] is the most efficient of the three, but also the most complicated. It relies on the difficulty of factoring and on the difficulty of computing discrete logarithms.

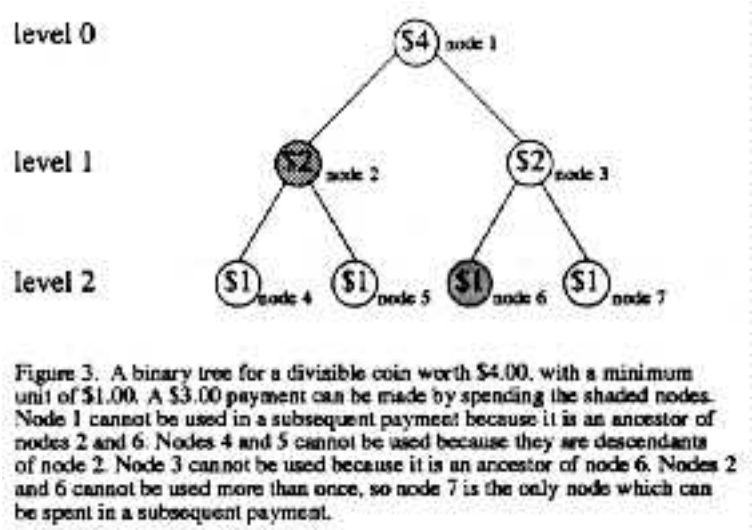

Figure 3. A binary tree for a divisible coin worth $4.00, with a minimum unit of $1.00. A $3.00 payment can be made by spending the shaded nodes. Node 1 cannot be used in a subsequent payment because it is an ancestor of nodes 2 and 6. Nodes 4 and 5 cannot be used because they are descendants of node 2. Node 3 cannot be used because it is an ancestor of node 6. Nodes 2 and 6 cannot be used more than once, so node 7 is the only node which can be spent in a subsequent payment.

Figure 3. A binary tree for a divisible coin worth $4.00, with a minimum unit of $1.00. A $3.00 payment can be made by spending the shaded nodes. Node 1I cannot be used in a subsequent payment because it is an ancestor of nodes 2 and 6. Nodes 4 and 5 cannot be used because they are descendants of node 2. Node 3 cannot be used because it is an ancestor of node 6. Nodes 2 and 6 cannot be used more than once, so node 7 is the only node which can be spent in a subsequent payment.

All three of these schemes work by associating a binary tree with each coin of value $*w*. (See Figure 3). Each node is assigned a monetary value as follows: the unique root node (the node at level 0) has value $*w*, the two nodes at level 1 each have value $*w*/2, the four nodes at level 2 each have value $*w*/4, etc. Therefore, if *w* = 2*l*, then the tree has *l*+ 1 levels, and the nodes at level *j* each have value $*w*/2*j*. The *leaves* of the tree are the nodes at level *l*, and have the minimum unit of value.
To spend the entire amount of value $*w*, the root node is used. Amounts less than $*w* can be spent by spending a set of nodes whose values add up to the desired amount.
Initially, any whole dollar amount of up to $*w* can be spent. Subsequent payments are made according to the following rules:

1. Once a node is used, all its descendant and ancestor9 nodes cannot be used.
2. No node can be used more than once.

These two rules insure that no more than one node is used on any path from the root to a leaf. If these two rules are observed, then it will be impossible to spend more than the original value of the coin. If either of these rules are broken, then two nodes on the same path are used, and the information in the two corresponding payments can be combined to reveal the identity of the individual that over-spent in the same way that the identity of a multiple spender is revealed.
More specifically, in the Eng/Okamoto and Okamoto schemes, each user has a secret value, *s*, which is linked to their identity (uncovering *s* will uncover their identity, but not vice-versa.) Each node *i* is assigned a secret value, *ti*. Hence, each node *i* corresponds to a line

$$y = sx + ti$$

When a payment is made using a particular node *n*, *ti* will be revealed for all nodes i that are ancestors of node *n*. Then the payee sends a challenge *xi* and the payer responds with

$$y1 = sx1 + tn .$$

This reveals a point (*x*1, *y*1) on the line $y = sx + tn$, but does not reveal the line itself. If the same node is spent twice, then responses to two independent challenges, *x*1 and x2, will reveal two points on the same line: (*x*1, *y*1) and (*x*2, *y*2). Then the secret value *s* can be recovered using the two-points-on-a-line principle described in 3.2.
If someone tries to overspend a coin, then two nodes in the same path will be used. Suppose that nodes *n* and *m* are in the same path, and node *n* is farther from the root on this path. Spending node *n* will reveal *tm*, since node *m* is an ancestor of node *n*. Now if node *m* is also spent, then the response to a challenge *x*1 will be $y1 = sx1 + tm$. But *tm* was revealed when *tn* was spent, so *sx*1 and hence *s* will be revealed. Therefore, spending two nodes in the same path will reveal the identity of the over-spender. The Okamoto/Ohta divisible scheme also uses a binary tree with the same rules for using nodes to prevent multiple and over-spending, but when nodes are used improperly, a different technique is used to determine the identity of the spender. Instead of hiding the user's identifying secret in a line for which a point is revealed when a coin is spent, the user's identifying secret is hidden in the factorization of an RSA modulus. Spending the same node twice, or spending two nodes on the same path will provide enough information for the Bank to factor the modulus (which is part of the coin) and then compute the user's secret identifying information.
Although these three divisible schemes are untraceable, payments made from the same initial coin may be "linked" to each

other, meaning that it is possible to tell if two payments came from the same coin and hence the same person. This does not reveal the payer's identity if both payments are valid (follow Rules 1 and 2, above), but revealing the payer's identity for one purchase would reveal that payer's identity for all other purchases made from the same initial coin.

These are three examples of off-line cash schemes that have divisible coins. Although providing divisibility complicates the protocol, it can be accomplished without forfeiting untraceability or the ability to detect improper spenders. The most efficient divisible scheme has a transaction time and required memory per coin proportional to the logarithm of N, where N is the total coin value divided by the value of the minimum divisible unit. More improvements in the efficiency of divisible schemes are expected, since the most recent improvement was just presented in 1995.

9 A *descendant* of a node n is a node on a path from node n to a leaf. An *ancestor* of node n is a node on the path from node n to the root node.

5. SECURITY ISSUES

In this section we discuss some issues concerning the security of electronic cash. First, we discuss ways to help prevent multiple spending in off-line systems, and we describe the concept of wallet observers. We also discuss the consequences of an unexpected failure in the system?s security. Finally, we describe a solution to some of the law enforcement problems that are created by anonymity.

5.1 Multiple Spending Prevention

In 1.3, we explained that multiple spending can be prevented in on-line payments by maintaining a database of spent electronic coins, but there is no cryptographic method for preventing an off-line coin from being spent more than once. Instead, off-line multiple spending is detected when the coin is deposited and compared to a database of spent coins. Even in anonymous, untraceable payment schemes, the identity of the multiple-spender can be revealed when the abuse is detected. Detection after the fact may be enough to discourage multiple spending in most cases, but it will not solve the problem. If someone were able to obtain an account under a false identity, or were willing to disappear after re-spending a large sum of money, they could successfully cheat the system.

One way to minimize the problem of multiple spending in an off-line system is to set an upper limit on the value of each payment. This would limit the financial losses to a given merchant due to accepting coins that have been previously deposited. However, this will not prevent someone from spending the same small coin many times in different places.

In order to prevent multiple spending in off-line payments, we need to rely on physical security. A "tamper-proof" card could prevent multiple spending by removing or disabling a coin once it is spent. Unfortunately, there is no such thing as a truly "tamper-proof" card. Instead, we will refer to a "tamper-resistant" card, which is physically constructed so that it is very difficult to modify its contents. This could be in the form of a smart card, a PC card10, or any storage device containing a tamper-resistant computer chip. This will prevent abuse in most cases, since the typical criminal will not have the resources to modify the card. Even with a tamper-resistant card, it is still essential to provide cryptographic security to prevent counterfeiting and to detect and identify multiple spenders in case the tamper-protection is somehow defeated. Also, setting limits on the value of off-line payments would reduce the cost-effectiveness of tampering with the card.

Tamper-resistant cards can also provide personal security and privacy to the cardholder by making it difficult for adversaries to read or modify the information stored on the card (such as secret keys, algorithms, or records).

10 Formerly PCMCIA, or Personal Computer Memory Card International Association.

5.2 Wallet Observers

All of the basic off-line cash schemes presented in 3.3 can cryptographically detect the identity of multiple spenders, but the only way to prevent off-line multiple spending is to use a tamper-resistant device such as a smart card. One drawback of this approach is that the user must put a great deal of trust in this device, since the user loses the ability to monitor information entering or leaving the card. It is conceivable that the tamper-resistant device could leak private information about the user without the user's knowledge.

Chaum and Pedersen [6] proposed the idea of embedding a tamper-resistant device into a user-controlled outer module in order to achieve the security benefits of a tamper-resistant device without requiring the user to trust the device. They call this combination an electronic wallet (see Figure 4). The outer module (such as a small hand-held computer or the user's PC) is accessible to the user. The inner module which cannot be read or modified is called the "observer." All information which enters or leaves the observer must pass through the outer module, allowing the user to monitor information that enters or leaves the card. However, the outer module cannot complete a transaction without the cooperation of the observer. This gives the observer the power to prevent the user from making transactions that it does not approve of, such as spending the same coin more than once.

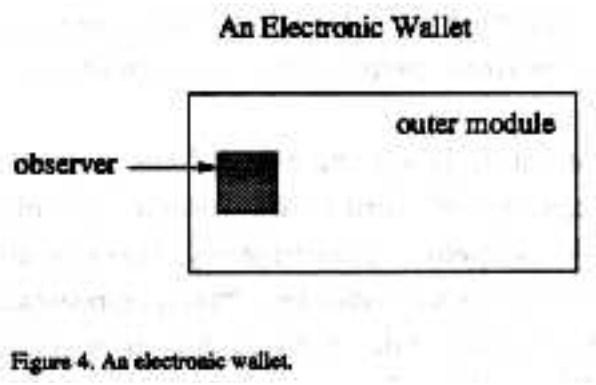

Figure 4. An electronic wallet.

Figure 4. An electronic wallet.

Brands[1] and Ferguson[8] have both shown how to incorporate observers into their respective electronic cash schemes to prevent multiple spending. Brands' scheme incorporates observers in a much simpler and more efficient manner. In Brands' basic scheme, the user's secret key is incorporated into each of his coins. When a coin is spent, the spender uses his secret to create a valid response to a challenge from the payee. The payee will verify the response before accepting the payment. In Brands' scheme with wallet observers, this user secret is shared between the user and his observer. The combined secret is a modular sum of the two shares, so one share of the secret reveals no information about the combined secret. Cooperation of the user and the observer is necessary in order to create a valid response to a challenge during a payment transaction. This is accomplished without either the user or the observer revealing any information about its share of the secret to the other. It also prevents the observer from controlling the response; hence the observer cannot leak any information about the spender.

An observer could also be used to trace the user's transactions at a later time, since it can keep a record of all transactions in which it participates. However, this requires that the Bank (or whoever is doing the tracing) must be able to obtain the observer and analyze it. Also, not all types of observers can be used to trace transactions. Brands and Ferguson both claim that they can incorporate observers into their schemes and still retain untraceability of the users' transactions, even if the observer used in the transactions has been obtained and can be analyzed.

5.3 Security Failures

Types of failures.

In any cryptographic system, there is some risk of a security failure. A security failure in an electronic cash system would result in the ability to forge or duplicate money. There are a number of different ways in which an electronic cash system could fail.

One of the most serious types of failure would be that the cryptography (the protocol or the underlying mathematics) does not provide the intended security.11 This could enable someone to create valid looking coins without knowledge of an authorized bank's secret key, or to obtain valid secret keys without physical access to them. Anyone who is aware of the weakness could create coins that appear to come from a legitimate bank in the system.

Another serious type of failure could occur in a specific implementation of the system. For example, if the bank's random number generator is not a-good one, one may be able to guess the secret random number and use it to compute the secret keys that are used to create electronic money.

Even if the cryptography and the implementation are secure, the security could fail because of a physical compromise. If a computer hacker, thief, dishonest bank employee, or a rogue state were to gain access to the bank's secret key they could create counterfeit money. If they gain access to a user's secret key they could spend that user's money. If they could modify the user or bank's software they could destroy the security of the system.

The above failure scenarios apply, not only to the electronic cash system, but also to the underlying authentication infrastructure. Any form of electronic commerce depends heavily on the ability of users to trust the authentication mechanisms. So if, for example, an attacker could demonstrate a forgery of the certification authority's digital signature, it would undermine the users' trust in their ability to identify each other. Thus the certification authorities need to be secured as thoroughly as do the banks.

Consequences of a failure.

All three of the basic schemes described in this paper are anonymous, which makes it impossible for anyone to connect a deposited coin to the originating banks withdrawal record of that coin. This property has serious consequences in the event of a security failure leading to token forgery. When a coin is submitted for deposit, it is impossible to determine if it is forged. Even the originating bank is unable to recognize its own coins, preventing detection of the compromise. It is conceivable that the compromise will not be detected until the bank realizes that the total value of deposits of its electronic cash exceeds the amount that it has created with a particular key. At this point the losses could be devastating.

After the key compromise is discovered, the bank will still be unable to distinguish valid coins from invalid ones since deposits and withdrawals cannot be linked. The bank would have to change its secret key and invalidate all coins which were signed with the compromised key. The bank can replace coins that have not yet been spent, but the validity of untraceable coins that have already been spent or deposited cannot be determined without cooperation of the payer. Payment untraceability prevents the Bank from determining the identity of the payer, and payer anonymity prevents even the payee from identifying the payer.

It is possible to minimize this damage by limiting the number of coins affected by a single compromise. This could be done by changing the Bank's public key at designated time intervals, or when the total value of coins issued by a single key exceeds a designated limit. However, this kind of compartmentation reduces the anonymity by shrinking the pool of withdrawals that

could correspond to a particular deposit and vice versa.

11 We are unaware of anything in the literature that would suggest this type of failure with the protocols discussed in this paper.

5.4 Restoring Traceability

The anonymity properties of electronic cash pose several law enforcement problems because they prevent withdrawals and deposits from being linked to each other. We explained in the previous section how this prevents detection of forged coins. Anonymity also makes it difficult to detect money laundering and tax evasion because there is no way to link the payer and payee. Finally, electronic cash paves the way for new versions of old crimes such as kidnapping and blackmail (see [13]) where money drops can now be carried out safely from the criminal's home computer.12

One way to minimize these concerns is to require large transactions or large numbers of transactions in a given time period to be traceable. This would make it more difficult to commit crimes involving large sums of cash. However, even a strict limit such as a maximum of $100 a day on withdrawals and deposits can add up quickly, especially if one can open several accounts, each with its own limit. Also, limiting the amount spent in a given time period would have to rely on a tamper-resistant device.

Another way to minimize these concerns is to provide a mechanism to restore traceability under certain conditions, such as a court order. Traceability can be separated into two types by its direction. For~ard traceability is the ability to identify a deposit record (and hence the payee), given a withdrawal record (and hence the identity of the payer). In other words, if a search warrant is obtained for Alice, forward tracing will reveal where Alice has spent her cash. Back~ard traceability is the ability to identify a withdrawal record (and hence the payer), given a deposit record (and hence the identity of the payee). Backward tracing will reveal who Alice has been receiving payments from.

A solution that conditionally restores both forward and backward traceability into the cut-and-choose scheme is presented by Stadler, Piveteau, and Camenisch in [14]. In the basic cut-and choose scheme, an identifying number is associated with each withdrawal record and a different identifying number is associated with each deposit record, although there is no way to link these two records to each other. To provide a mechanism for restoring backward traceability, the withdrawal number (along with some other data which cannot be associated with the withdrawal) is encrypted with a commonly trusted entity's public key and incorporated into the coin itself. This encrypted withdrawal number is passed to the payee as part of the payment protocol, and then will be passed along to the bank when the coin is deposited by the payee. The payer performs the encryption during the withdrawal transaction, but the bank can insure that the encryption was done properly. If the required conditions for tracing are met, the payment or deposit can be turned over to the trusted entity holding the secret key to decrypt the withdrawal number. This withdrawal number will allow the bank to access its withdrawal records, identifying the payer.

To provide a mechanism for restoring forward traceability, the payer must commit to a deposit number at the time that the coin is withdrawn. The payer encrypts this deposit number with a commonly trusted entity's public key (along with some other data that cannot be associated with the deposit) and must send this value to the bank as part of the withdrawal protocol. The bank is able to determine that the payer has not cheated, although it only sees the deposit number in encrypted form. If the required conditions for tracing are met, the withdrawal record can be turned over to the trusted entity holding the secret key to decrypt the deposit number. The bank can use this deposit number to identify the depositor (the payee).

Stadler et al. have shown that it is possible to provide a mechanism for restoring traceability in either or both directions. This can be used to provide users with anonymity, while solving many of the law enforcement problems that exist in a totally untraceable system. The ability to trace transactions in either direction can help law enforcement officials catch tax evaders and money launderers by revealing who has paid or has been paid by the suspected criminal. Electronic blackmailers can be caught because the deposit numbers of the victim's ill-gotten coins could be decrypted, identifying the blackmailer when the money is deposited.

The ability to restore traceability does not solve one very important law enforcement problem: detecting forged coins. Backwards tracing will help identify a forged coin if a particular payment or deposit (or depositor) is under suspicion. In that case, backwards tracing will reveal the withdrawal number, allowing the originating bank to locate its withdrawal record and verify the validity of the coin. However, if a forged coin makes its way into the system it may not be detected until the bank whose money is being counterfeited realizes that the total value of its electronic cash deposits using a particular key exceeds the values of its withdrawals. The only way to determine which deposits are genuine and which are forged would require obtaining permission to decrypt the withdrawal numbers for each and every deposit of electronic cash using the compromised key. This would violate the privacy that anonymous cash was designed to protect.

Unfortunately, the scheme of [14] is not efficient because it is based on the bulky cut-and-choose method. However, it may be possible to apply similar ideas to restore traceability in a more efficient electronic cash scheme.

12 We will not focus on such crimes against individuals, concentrating instead on crimes against the Government, the banking system, and the national economy.

CONCLUSION

This report has described several innovative payment schemes which provide user anonymity and payment untraceability. These electronic cash schemes have cryptographic mechanisms in place to address the problems of multiple spending and token forgery. However, some serious concerns about the ability of an electronic cash system to recover from a security failure have been identified. Concerns about the impact of anonymity on money laundering and tax evasion have also been discussed.

Because it is simple to make an exact copy of an electronic coin, a secure electronic cash system must have a way to protect

against multiple spending. If the system is implemented on-line, then multiple spending can be prevented by maintaining a database of spent coins and checking this list with each payment. If the system is implemented off-line, then there is no way to prevent multiple spending cryptographically, but it can be detected when the coins are deposited. Detection of multiple spending after-the-fact is only useful if the identity of the offender is revealed. Cryptographic solutions have been proposed that will reveal the identity of the multiple spender while preserving user anonymity otherwise.

Token forgery can be prevented in an electronic cash system as long as the cryptography is sound and securely implemented, the secret keys used to sign coins are not compromised, and integrity is maintained on the public keys. However, if there is a security flaw or a key compromise, the anonymity of electronic cash will delay detection of the problem. Even after the existence of a compromise is detected, the Bank will not be able to distinguish its own valid coins from forged ones. Since there is no way to guarantee that the Bank's secret keys will never be compromised, it is important to limit the damage that a compromise could inflict. This could be done by limiting the total value of coins issued with a particular key, but lowering these limits also reduces the anonymity of the system since there is a smaller pool of coins associated with each key.

The untraceability property of electronic cash creates problems in detecting money laundering and tax evasion because there is no way to link the payer and payee. To counter this problem, it is possible to design a system that has an option to restore traceability using an escrow mechanism. If certain conditions are met (such as a court order), a deposit or withdrawal record can be turned over to a commonly trusted entity who holds a key that can decrypt information connecting the deposit to a withdrawal or vice versa. This will identify the payer or payee in a particular transaction. However, this is not a solution to the token forgery problem because there may be no way to know which deposits are suspect. In that case, identifying forged coins would require turning over all of the Bank's deposit records to the trusted entity to have the withdrawal numbers decrypted.

We have also looked at two optional features of off-line electronic cash: transferability and divisibility. Because the size of an electronic coin must grow with each transfer, the number of transfers allowed per coin must be limited. Also, allowing transfers magnifies the problems of detecting counterfeit coins, money laundering, and tax evasion. Coins can be made divisible without losing any security or anonymity features, but at the expense of additional memory requirements and transaction time.

In conclusion, the potential risks in electronic commerce are magnified when anonymity is present. Anonymity creates the potential for large sums of counterfeit money to go undetected by preventing identification of forged coins. Anonymity also provides an avenue for laundering money and evading taxes that is difficult to combat without resorting to escrow mechanisms. Anonymity can be provided at varying levels, but increasing the level of anonymity also increases the potential damages. It is necessary to weigh the need for anonymity with these concerns. It may well be concluded that these problems are best avoided by using a secure electronic payment system that provides privacy, but not anonymity.

REFERENCES

1. Stefan Brands, *Untraceable Off-Line Cash in Wallets with Observers*, Advances in Cryptology CRYPTO '93, Springer-Verlag, pp. 302-318.

2. David Chaum, *Achieving Electronic Privacy*, Scientific American (August 1992), 96-101.

3. David Chaum, *Security without Identification: Transaction Systems to make Big Brother Obsolete*, ACM 28 no. 10 (Oct 1985), 1030-1044.

4. David Chaum, Amos Fiat, and Moni Naor, *Untraceable Electronic Cash*, Advances in Cryptology CRYPTO '88, Springer-Verlag, pp. 319-327.

5. David Chaum and Torben Pedersen, *Transferred Cash Grows in Size*, Advances in Cryptology - EUROCRYPT '92, Springer-Verlag, pp. 390-407.

6. David Chaum and Torben Pedersen, *Wallet Databases with Observers*, Advances in Cryptology CRYPTO '92, Springer-Verlag, pp. 89-105.

7. Tony Eng and Tatsuaki Okamoto, *Single-Term Divisible Electronic Coins*, Advances in Cryptology EUROCRYPT '94, Springer-Verlag, pp. 311-323.

8. Niels Ferguson, *Extensions of Single-term Coins*, Advances in Cryptology - CRYPTO '93, Springer-Verlag, pp. 292-301.

9. Niels Ferguson, *Single Term Off-Line Coins*, Advances in Cryptology - EUROCRYPT '93, Springer-Verlag, pp. 318-328.

10. Alfred J. Menezes, *Elliptic Curve Public Key Cryptosystems*, Kluwer Academic Publishers, Boston, 1993.

11. Tatsuaki Okamoto, *An Efficient Divisible Electronic Cash Scheme*, Advances in Cryptology - CRYPTO '95, Springer-Verlag, pp. 438-451.

12. Tatsuaki Okamoto and Kazuo Ohta, *Universal Electronic Cash*, Advances in Cryptology - CRYPTO '91, Springer-Verlag, pp. 324-337.

13. Sebastiaan von Solms and David Naccache, *On Blind Signatures and Perfect Crimes*, Computers & Security 11 (1992), 581-583.

14. Markus Stadler, Jean-Marc Piveteau, and Jan Camenisch, *Fair Blind Signatures*, Advances in Cryptology - EUROCRYPT '95, Springer-Verlag, pp. 209-219.

[End]

Thanks to the authors, Thomas Vartanian and anonymous others.

Report any transcription mistakes in equations to <jya@pipeline.com>. Check corrections page for updates.