# Reusable Off-line Electronic Cash Using Secret Splitting

Hitesh Tewari, Donal O'Mahony, Michael Peirce

Networks & Telecommunications Research Group,
Computer Science Department, Trinity College,
Dublin 2, Ireland.

{Hitesh.Tewari, Donal.OMahony, Michael.Peirce}@cs.tcd.ie

***Abstract***

The idea of electronic cash, as a payment instrument is appealing, but has yet to be widely deployed commercially. We outline the properties of two major approaches to the provision of electronic cash, and discuss their strengths and weaknesses. We then propose a new scheme based on secret splitting that allows for off-line spending of coins with a tailored level of anonymity and allowing reuse of the same coin many times. We then discuss a number of practical issues such as the trade off between coin size and the ability to detect double spending.

## Introduction

The rapid growth of data communications networks in recent years has led to a massive growth in electronic commerce. Large organizations such as banks are always looking for means for speeding up and simplifying the financial transaction process. The cost associated with printing, transporting and securing cash is quite high. This has led to a great deal of research on the concept of a *cashless society*, which would result in elimination of all forms of paper based cash. However in the digital world, it is easy to duplicate a piece of information that is stored in a computer's memory or hard disk. Security mechanisms have to be put in place that will prevent unscrupulous users from defrauding the system.

There is a need to invent new electronic payment protocols with strong cryptographic algorithms that will eventually replace present day paper based cash schemes. Over the years a number of electronic cash schemes have been proposed, which try to mimic today's cash based schemes. There are a number of features, that one might expect from an electronic cash scheme. We outline some of these below.

**Anonymity** – Privacy advocates believe that it should not be easy for a bank or governments to be able to trace serial numbers of notes to a particular transaction or individual. It should also not be possible to monitor the activities of individual users and build profiles about their spending habits. On the other hand, from the point of view of governments fully anonymous cash is not

desirable, as it could be used as a tool by criminals for illegal activities such as money laundering.

**Reusability** – One should be able to reuse the same *piece* of cash multiple times. It can pass from person to person before it is destroyed or deleted by the issuer. This feature in turn can help with the previous requirement of anonymity. The more times the cash is transferred, the more difficult it becomes to trace it to the person to whom it was issued. However, the ability to reuse electronic currency also increases the risk of double spending and fraud in the system.

**Off-line** – The ability for two parties to complete a transaction without the intervention of a third party such as the issuing bank or an authentication server is a highly desirable feature.

**Scalability** –There should be no reliance on a centralized architecture, as this can become a point of failure in the system. It also acts as bottleneck during periods of high usage. For any scheme to be successful for mass deployment, it must be capable of distributed operation.

In this paper, we present an electronic cash scheme that exhibits all the above properties. The paper begins with a discussion of a number of important cash-like payment schemes that have gained significant importance in the area. We then focus on the cryptographic techniques that we have made use of in our proposed scheme and then go on to describe the payment protocol in detail. Finally we discuss the advantages and disadvantages of the features of our scheme and make a number of conclusions.

# Related Work

A number of electronic payment schemes have been proposed in the recent past. They can be classified in two broad categories. Schemes that are pure software solutions and do not make use of any specialized hardware such as smart cards e.g. eCash, and schemes that rely on secure hardware for their security such as CAFE. We also describe a payment scheme in widespread use called Mondex, which though it may not be classified as a *pure* electronic cash scheme, exhibits some cash like properties. The analysis of these systems helps us to highlight some of the features that we think should be incorporated into an electronic cash scheme for it to be universally acceptable.

## Digital Cash

A fully anonymous digital cash protocol was proposed by David Chaum in [1, 2, 3]. The scheme is essentially an on-line software solution. This means that a buyer can spend coins with a merchant, which must be validated by the issuer before the purchase can be completed. From examining the coins, neither the issuer nor the merchant can deduce the identity of the customer. The protocols are designed such that a issuer is not able detect the serial numbers of coins that it issues to users of the system, even if it colludes with the other participants in the system. The entities in the system are shown in figure 1.

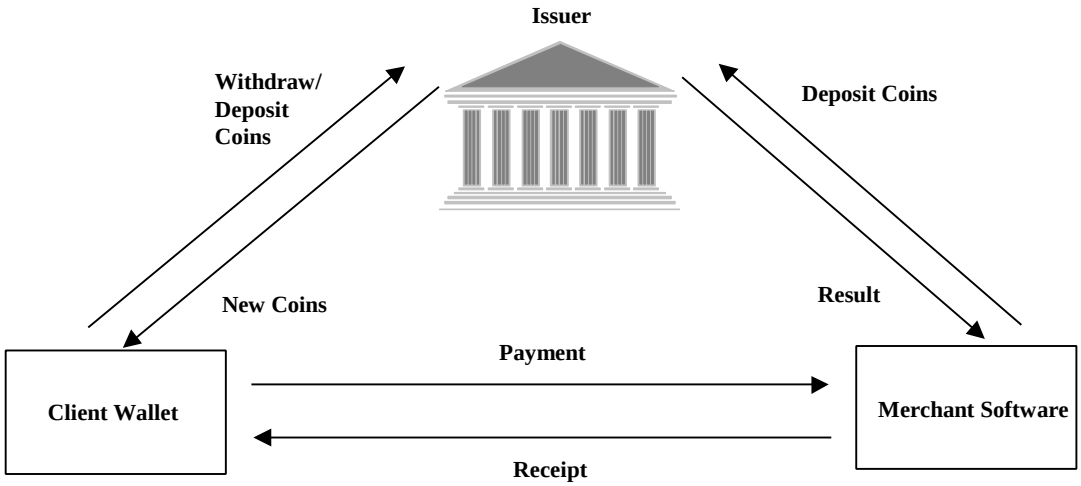


**Figure 1: The Payment Model**

A client can withdraw coins from a bank against an existing account. The scheme uses what is referred to as the *blind signature* protocol [2]. This protocol allows the user to mint a number of coins and forward these *unsigned* coins to the bank. As long as these coins meet certain criteria, the bank signs these with its secret-key without the knowledge of the serial numbers associated with the coins. This feature allows for fully anonymous cash.

On receiving the coins back from the bank, the user removes the blinding factor and can use the coins to pay for goods from any merchant that is participating in the system. On receipt of coins, a merchant must immediately forward them to the issuer for verification. The issuer maintains a database of serial numbers of all coins that have been spent in the system and is thus able to detect double spending.

Advantages

- A software only solution that does not require the use of secure tamper-resistant hardware e.g. smart cards
- Provides user anonymity and untraceable electronic cash

Disadvantages

- The merchant has to maintain an on-line link with the bank to verify the authenticity of any coins that he accepts. This is an unwanted communications overhead
- The bank has to maintain an ever growing database of serial numbers of coins that have been spent in the system
- The scheme is difficult to scale, as it requires the spent coins database to be replicated and synchronization between bank servers
- Coins can only be spent once, after which they must be deposited with the bank

## CAFE

CAFE (Conditional Access for Europe) was a project funded under the European ESPRIT program [6, 7] to develop a secure, anonymous, off-line electronic payment system. The CAFE

protocols are based on the idea of untraceable electronic cash proposed by Chaum and are designed to allow for multi-party security. The security of each entity in the system is guaranteed without the need to trust a third party. CAFE employs two types of security mechanisms to protect the system against attack. The first line of defense is the use of secure tamper-resistant devices with *observers* to store cryptographic keys and to perform all cryptographic transactions. An observer is an embedded integrated circuit that is a trusted entity of the issuing bank. The observer acts on behalf of the issuing organization, ensuring that the smart card in which it is placed dos not deviate from the prescribed protocols.

Protection against double spending is guaranteed as long as the tamper resistance of the devices involved is not compromised. In situations where there is a hardware security breach, CAFE has a cryptographic fallback mechanism, which allows the financial institutions to detect double spending of coins and blacklist suspected users. These lists are then distributed to all merchants in the system.

A coin in the CAFE scheme is constructed from two parts, each of which contains part of the user identity. When a coin is used in a payment transaction, the payer opens one half of the coin. This on its own does not reveal the identity of the user to whom the coin was issued. However, if the user tries to double spend the coin, he will have to open the second half of the coin. Combining the two halves results in the identity of the user being revealed.

Advantages

- Does not rely on secure hardware alone for protection of system security
- It is an off-line scheme
- Banks can detect double spending by users
- Under normal circumstances, payments cannot be linked to a user even if there is collaboration between the merchants and the banks

Disadvantages

- Need to maintain large databases of spent coins. Can lead to scalability problems
- Reusability of coins is not supported
- Makes use of strong cryptographic techniques to ensure the security of the system

## Mondex

The concept of the Mondex [7] card was developed at NatWest, a major UK banking organization in 1990. In July 1996, Mondex International (MXI) was incorporated in the UK. The company is owned by MasterCard International and 28 major organizations world wide including banks in US, Canada, Australia and Asia.

Mondex is an off-line smart card electronic-purse scheme that allows card-to-card transfer of electronic cash. To use the system, a user loads the card with cash, using either a Mondex automated teller machine (ATM) or a specially adapted telephone. Each merchant is equipped with a *value transfer terminal*. This device can be used to transfer cash from the customer to a card in the merchant's device.

The Mondex payment scheme is a closed scheme, as the company has not disclosed the details of the payment protocol. The scheme is presently in operation in a number of countries around the world.

Advantages

- Value can be transported between cards multiple times
- User-to-user transfer of electronic cash without the intervention of a trusted third party or an on-line connection

Disadvantages

- Little is known about the internal working of the system (Does not give much confidence to end-users)
- Not an anonymous system

To summarize, CAFE and Mondex are electronic payment schemes, which exhibit some cash like properties. However, none of these schemes is able to fulfill all our requirements for an ideal electronic cash payment scheme. In the next section, we describe in detail the functionality of our payment scheme.

# Cryptographic Design

In this section, we describe two cryptographic techniques that we have used in the design of our electronic cash scheme. The first is a mechanism to split the identity of a user into two parts using a technique called *secret splitting*. The second technique makes use of this to encode the identity of a user into a coin.

## Secret Splitting

Secret splitting is a technique that allows a message to be divided up into *n* parts. Each part on its own has no meaning. However, when they are combined together, they result in the original message. A simple implementation is to XOR (exclusive-OR) the message *M* with a one-time pad *R* as follows:

$$M \oplus R = S$$

Knowledge of *R* or *S* on its own is not sufficient to generate *M*

In [4], Schneier proposes a digital cash protocol that makes use of the above technique to encode the identity of a user. Below we outline a simplified version of his protocol that is used in our proposed electronic cash scheme.

A coin in our scheme consists of a number of fields, such as the serial number, denomination, validity period and a *transaction list*. This transaction list comprises of a variable number of *transactions items*, where each transaction item is the result of a transfer of the coin between two users in the system. A transaction item consists of *n* identity strings ($I_1$, $I_2$, …, $I_n$). Each of these identity strings is generated, by splitting the identity of the user using the secret splitting protocol.

Transaction Item:

$$I = (I_{1L}, I_{1R})$$
$$I = (I_{2L}, I_{2R})$$

$$I = (I_{nL}, I_{nR}) \quad \text{(where } I \text{ is the identity of a user)}$$

XORing the left and right corresponding halves of any pair will result in the identity of the user ($I$). The last transaction item in the list is unblinded and contains the identity of the coin's current owner. When a coin is transferred from one user to another, the entity acting on behalf of the bank *randomly* blinds the left or right half of each of the identity strings. It then adds a new transaction item (encoding the identity of the recipient) to the transaction list, cryptographically binds the new transaction list to the coin and forwards the coin to its new owner.

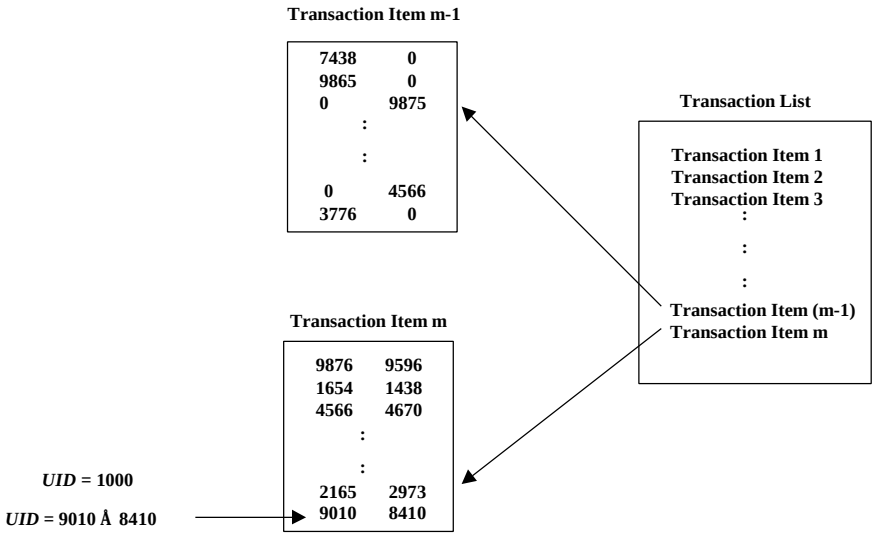


**Figure 2: A Transaction Item List**

Figure 2 shows a transaction list of a coin whose second last transaction item has been blinded, while the last transaction item remains in the clear. If a user tries to spend this coin twice, during each individual payment transaction the left or right halves of the coin will be randomly blinded. There is a high probability that one or more corresponding left and right halves of the transaction will remain unblinded. When two coins with the same serial numbers are deposited back to the issuer, it will be able to look through the transaction list and combine the two halves to reveal the identity of the user that committed the fraud.

# System Overview

In this section, we give an overview of our system. It consists of a number of entities, which communicate with each other to complete various stages of the payment process.

## System Entities

The entities that comprise the system are:

**Issuer** – Responsible for the minting and detection of double spending of coins. It maintains a database of coins that have been deposited and are no longer in circulation in the system. Since the coins are reusable, the size of the database is significantly smaller compared to some of the

schemes we have previously outlined. Each coin also has validity period associated with it. This also helps in reducing the overall size of the database

**Bank** – Maintains end-user accounts. It distributes and accepts coins from end-users

**Buyer** – Users have accounts with banks from which they can withdraw and deposit coins. They store coins in an electronic purse or a software wallet

**Merchant** – Users that can accept coins in exchange for hard goods

**Point of Sale (POS) Device** – Secure off-line devices that act as intermediary's during the transfer of coins between two users. Black lists are downloaded to the device from a central server at periodic intervals, along with the serial numbers of coins that have been deposited more than once at the issuer

Figure 3 shows the various steps that make up the payment process. A bank may periodically ask the issuer to mint a large quantity of coins. When a request is made by an authorized user for new coins, the bank will bind the coins to the users identity and forward the same. A user can then use the coins to pay for services from any merchant in the system. A POS will be involved in the transfer of coins between the two.

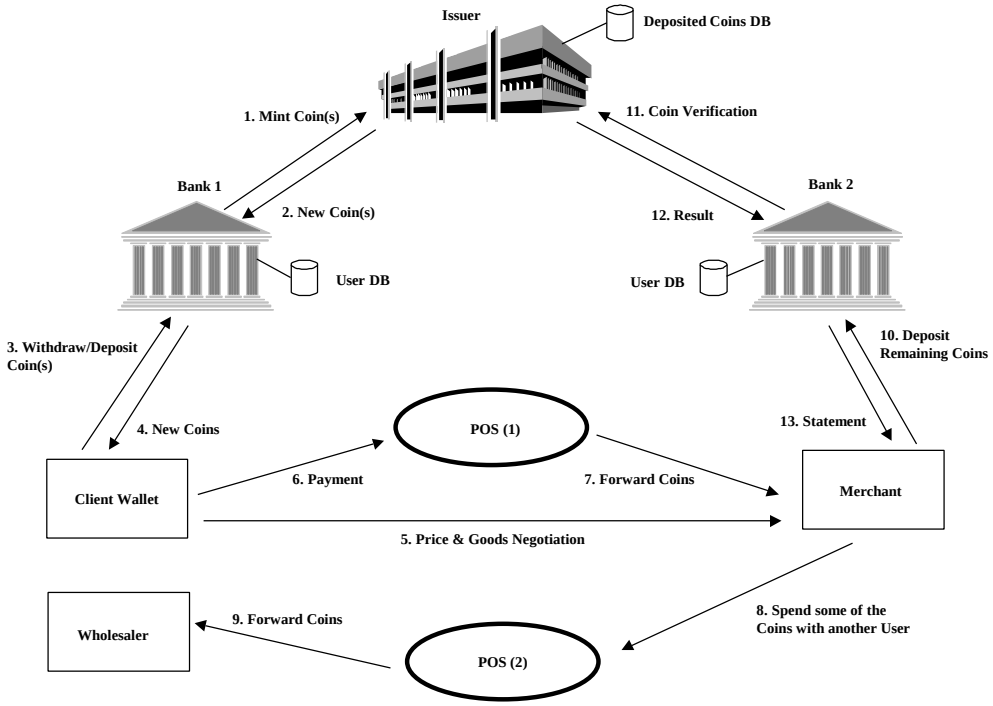


**Figure 3: Overview of the System**

The POS will verify the authenticity of the coins. It will blind the last transaction item in the transaction list and add a new transaction item to the end of the list. This will bind the coin to the recipient (e.g. merchant). The merchant can immediately reuse the coins to pay for services from other users in the system e.g. buying supplies from a wholesaler. At some stage in the process, an end-user will have to deposit the coins with his bank. The bank will forward the coins to the issuer who will check for double spending and add the serial numbers of the spent coins to its database.

There are a number of public-key pairs in the system, which are used for minting coins and appending transaction items. *PK* is denoted as the public-key and *SK* as the secret-key.

- Public key pair for minting and verifying the authenticity of coins

$$(PK_{Issuer}, SK_{Issuer})$$

- Public key pair for appending and verifying transaction items to a coin

$$(PK_{Trans}, SK_{Trans})$$

Table 1 shows the keys which each entity in the system possess

| | | |
|---|---|---|
| Issuer | $(PK_{Issuer}, SK_{Issuer})$ | $(PK_{Trans}, SK_{Trans})$ |
| Bank | $PK_{Issuer}$ | $(PK_{Trans}, SK_{Trans})$ |
| POS | $PK_{Issuer}$ | $(PK_{Trans}, SK_{Trans})$ |
| Users | $PK_{Issuer}$ | $PK_{Trans}$ |

**Table 1: The System Keys**

The issuer is able to mint and verify the authenticity of coins. The banks and the point of sale devices can verify the authenticity of coins and are able to append transaction items to coins. They are not able to mint coins.

Each end-user is assigned an identifier (*UID*) which is signed by the issuer. A digital signature on a given value consists of the value, along with a hash of the same, which is encrypted with the secret-key of the signer.

$$(UID)Sig_{Issuer} = \{UID, (H(UID))SK_{Issuer}\}$$

A *UID* is used by the banks and the POS devices in the system to verify the identity of end-users and to bind coins to the recipient of the same.

## Structure of a Coin

A coin in the system consists of the following parameters:

$$\{\text{SN, Denom, Val}\}Sig_{Issuer}, (H(\text{SN, Denom, Val}), \text{TranList})Sig_{Trans}\}$$

- **Serial Number (SN)** – A unique system wide identifier
- **Denomination (Denom)** – The face value of the coin
- **Validity (Val)** – The period of time for which this coin remains valid. This parameter limits the size of the database at the issuer
- **Transaction List (TranList)** – A list of transaction items associated with each time the coin is transferred from one user to another. There is an upper-limit to the number of transaction items that can be contained in a transaction list

The first three fields of the coin are signed with the secret-key of the issuer ($SK_{Issuer}$). To bind a coin and its corresponding transaction list together, a hash of the first three fields along with the

transaction list is signed with the secret-key ($SK_{Trans}$). This last part of the coin can be modified by the banks and the POS devices in the system, to bind the coin to a recipient during a coin transfer phase.

# The Payment Protocol

We now describe in detail the messages that are used to effect the various stages of the payment process. They are the withdrawal, payment and deposit phases. We also describe how the system is able to detect double spending of coins by users.

## Withdrawal

When a user wishes to withdraw some coins from his bank

1. He forwards his signed *UID* along with the amount of coins required to the bank

The bank:

- Verifies the user identity and account details
- May have a number of coins in storage or it may have to request coins from the issuer
- Uses the secret splitting technique on the *UID* to generate the *first* transaction item for the coin. It adds this *unblinded* transaction item to the transaction list and signs it with ($SK_{Trans}$). This binds the user to the coin

2. The bank forwards the coins to the user

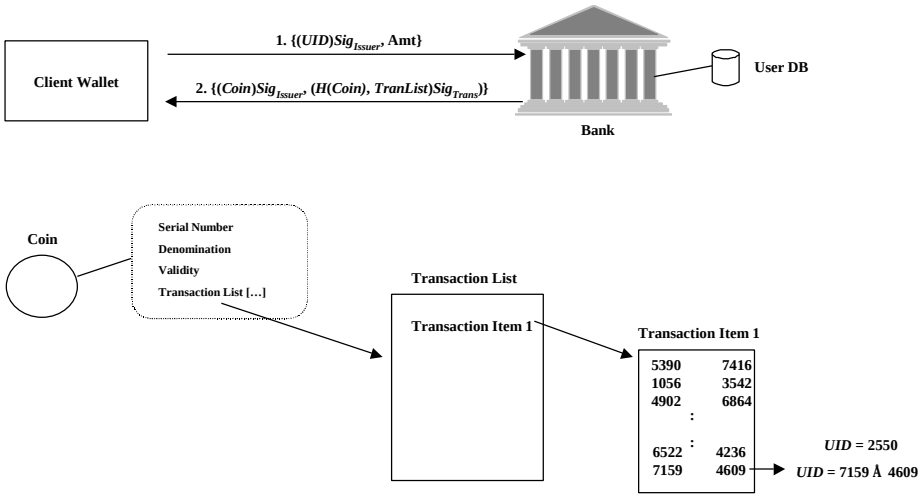


**Figure 4: Withdrawal of Coins**

The network link between the client and the bank can be secured in a number of ways e.g. using a shared secret, Secure Socket Layer (SSL) [8, 9] or using the Diffie-Hellman [4, 5] key-agreement algorithm.

## Payment

When a customer wishes to purchase goods, he negotiates a price with the merchant. The process of price negotiation and selection of merchandise is beyond the scope of this discussion. Once both parties agree on the purchase details, the transfer of coins can take place. This occurs in the presence of a POS device. This is a secure off-line device, whose role is to verify the authenticity of the coins being transferred and to bind the coins to the recipient's identity. Figure 5 gives an overview of the process.

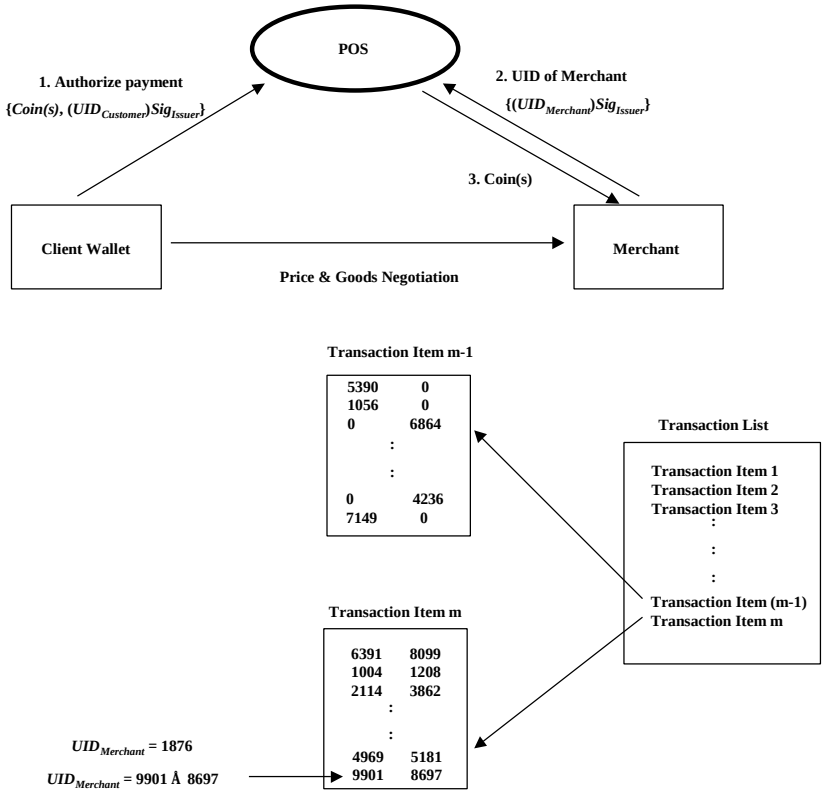


**Figure 5: Payment for Goods**

The steps, which constitute the main payment process, are as follows

1. The customer forwards coins and his signed *UID* to the POS

2. The merchant also forwards his signed *UID* to the POS

For each coin that the POS receives from the customer it:

- Verifies the authenticity of the coin and checks the validity period
- Consults its local database of black listed users and ensures that both parties to the transaction are valid users of the system
- Verifies that the last transaction item in the transaction list belongs to that of the customer. This gives confidence to the POS device that the coin has not been stolen
- If either of the above steps fail the POS will reject the coin
- Randomly blinds the left or right half on individual entries in the last transaction item. This prevents the new owner of the coins from identifying the customer

- Creates a new unblinded transaction item that binds the coin to the recipient (merchant) and signs it with ($SK_{Trans}$)

3. Finally, it forwards the coins to the merchant

To prevent a user from masquerading as a merchant and accepting coins on his behalf, each POS device in the system can be personalized for the merchant to whom it is issued. The merchant would then share a secret with the POS e.g. password, which is used to authenticate him to the device.

Employing the above method prevents the transfer of coins between two end-users. If such functionality were desired, one would require all users and POS devices in the system to share a secret. Alternatively, one could assign public/secret key pairs and associate certificates to all users of the system.

## Deposit

An end-user may be required to deposit coins with his bank if the number of transaction items has reached its maximum limit or if the validity period of a coin is about to expire. This ensures that the size of the coin does not become too large. It also helps in minimizing the risk of not being able to detect coins that have been double spent, for long periods.

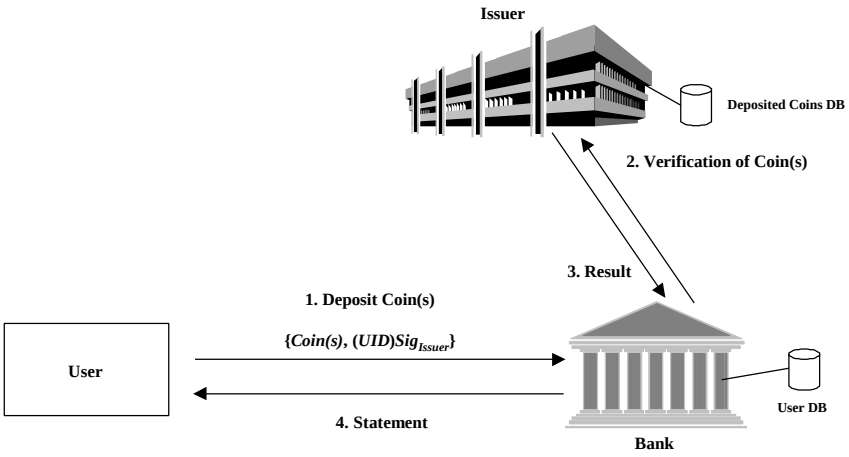


**Figure 6: A Deposit Transaction**

1. The user forwards the coins along with his signed *UID* to the bank

2. The bank compares the *UID* of the user against the last transaction item in the transaction list. This gives the bank confidence that the coins were not stolen. It forwards the coins to the issuer. This step can be performed immediately or at a later date when the bank has a batch of coins to clear

For each coin deposited with the issuer:

- It consults the deposited coins database and checks to see that the coin has not been deposited previously
- If not, it then adds the serial number of coin to the database

3. The issuer returns a result to the bank

4. The bank credits the user's account and forwards a statement to the user

## Tracing Double Spenders

Since a coin is just a piece of digital data, it is possible for a user to duplicate this information and attempt to spend a coin multiple times. There is a post-fact detection mechanism built into the system so that double spending can be detected when the coins are being deposited at the issuer with a high probability of identifying the offending party.

If the issuer detects a coin with the same serial number as one previously recorded in its database, it will assume that the coin has been double spent. It will try to determine the *UID* of the user that spent the coin multiple times.

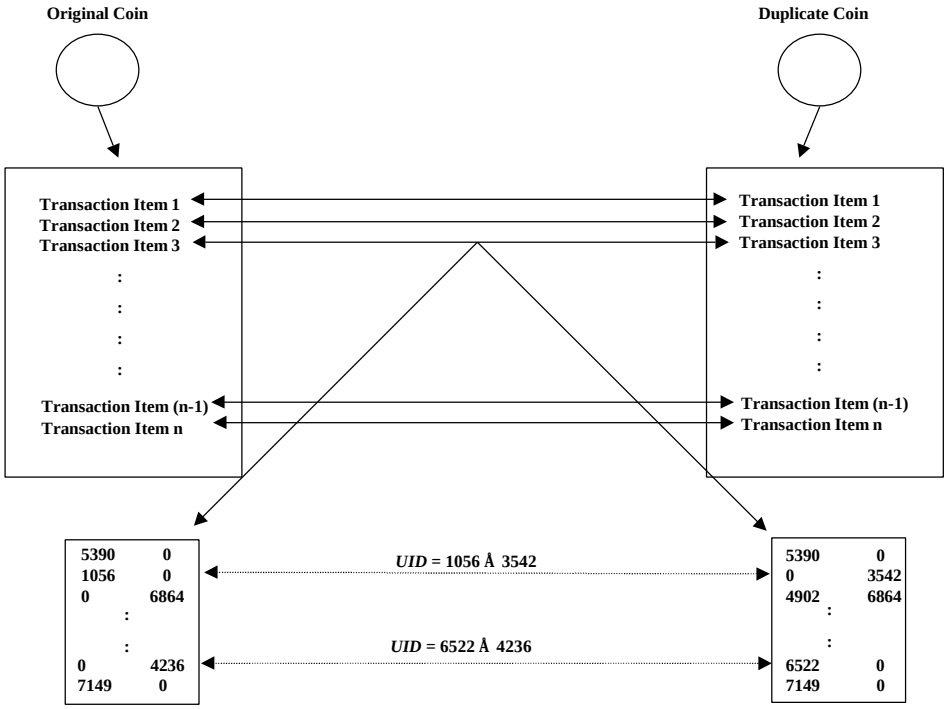


**Figure 7:Post-fact Detection of Double Spending**

It does this by traversing through the transaction items in the transaction list of the two coins, and identifying corresponding transaction items where different left and right half entries have been opened. Remember that each POS device will randomly blind the right and left halves of a transaction item. Once the issuer finds an entry where the corresponding left and right halves are in the clear in the two coins, it does a simple XOR of the two values to reveal the *UID* of the culprit.

Users that are caught double spending are eliminated from the system. Their *UID* is distributed in black lists to all POS devices in the system along with the serial numbers of the coins.

# Critique

We have presented a cash-like electronic payment scheme combines the benefits of off-line functionality with detection of double spending of coins by users. The two main concerns that one might express about the scheme are to do with anonymity of transactions and the size of coins. The latter impacts on the type of storage that may be used for the system. We address each of these concerns below.

## Anonymity

Under normal operation, the scheme allows for anonymous cash transactions. The initial (withdrawal) and final (deposit) transactions allow a bank to associate a coin's serial number with the *UID* of the end-user. All intermediate transactions involving POS devices will hide the identity of the last owner of the coin. Only if a user double-spends a coin will his identity be revealed.

However, it is also possible to relax this option by allowing the POS devices in the system to maintain a log of each transaction. This log could then be transferred to the issuer, who could use it to create an audit trail for each coin.

## Storage Requirements & Double Spending

Each of the coins in the system we propose consists of a fixed header component together with a variable length transaction list that is enlarged each time the coin is spent. The size of this variable length component depends on a number of factors:

- The Users Identity (***U***) - Users in the system are issued with numerical identities. These are then encoded into the coin by the point of sale device
- The number of entries in each transaction item (***T***)
- The maximum number of times a coin can be spent (***M***)

When a coin is generated, the transaction list contains a single users identity encoded as a pair of (unblinded) numbers. Thus

$$\text{Minimum Transaction list size} = 2 * T * U$$

Each time the coin is used (up to *M* times), a blinded transaction item is added. Given that one of the two numbers in each entry is blinded, this can be compactly represented using just one number, and a single bit indicating whether it is the left or the right component of the pair.

$$\text{Maximum Transaction list size} = 2 * T * U + (M - 1)\,(U + 1) * T$$

Some typical values for the above parameters and the consequent coin sizes are shown in the following table:

| Parm Values | Min Size | Max Size | Avg Size |
|---|---|---|---|
| U=64bits,T=6,M=10 | 96 bytes | 535 bytes | 315 bytes |
| U=64bits,T=3,M=5 | 46 bytes | 146 bytes | 97 bytes |
| U=32bits,T=3,M=5 | 24 bytes | 74 bytes | 49 bytes |

It is clear that coin sizes of up to half a kilobyte does mean that a significant amount of storage is required to store a reasonable amount. For example, using the first set of parameters in table 2 : maintaining $40 in an electronic purse in $1 denominations would occupy 12.6 kilobytes. If the same amount were stored using 1 cent coins, the storage requirement would exceed 1 Mbyte. These values rule out the use of current generation smart cards, but other portable hardware devices could cope with these storage requirements.

Minimising *T* and *M* will limit the maximum size of each coin, but these values also have a major bearing on the systems ability to detect and block double spending. Once a coin has been double spent, it will continue to be used until all of its *M* transactions have been used up. Thus the value of *M* is determines how long it will take before the coins return to the issuer where the double spending may be detected. Fraudsters are likely to choose relatively 'fresh' coins to maximise this time.

When a coin is returned to the bank, any cases of double spending will cause the issuer to match the transaction lists of the coins in question. If there is only one entry per transaction list (*T*=1), and the coin has been spent twice, the probability that the fraudsters identity can be recovered is 50%. If *T*=2, then this increases to 75% while with larger values such as *T*=6 there is a 98.5% chance of detection.

If the coin is spent more than twice, the probability of detection goes up very quickly indeed. In general, if the coin is used *K* times, the probability of detection is:

$$\text{Probability of Detection} = 1 - 1/2^{T(K-1)}$$

So, for example, a fraudster spending a single coin with *T*=6, 3 times has a 99.97% chance of having his identity discovered.

Balancing the penalty of increased size with choosing large values for *T* and *M* against the increased protection, issuers may choose to use larger values for coins with high values and opt for smaller coins to represent low values.

## Conclusions

This paper presents a lightweight electronic cash payment scheme. It has the flexibility that one would expect from present day paper based cash schemes, without the complexity that is sometimes associated with electronic payment systems. It combines the benefits of off-line payments, user anonymity and reusability of coins. The scheme has a post-fact detection of double spending mechanism built into it, and is well suited for real-time payments in today's increasingly digital world.

## References

[1] D. Chaum, A. Fiat and N. Naor, "Untraceable Electronic Cash", *Proceedings of Crypto '88*, Springer-Verlag, v. 403, 1990, pp. 319-327

[2] D. Chaum, "Security without Identification: Transaction Systems to Make Big Brother Obsolete", *Communications of the ACM*, v. 28, n. 10, Oct 1985, pp. 1030-1044

[3] D. Chaum, "Achieving Electronic Privacy", *Scientific American*, v. 267, n. 2, Aug. 1992, pp. 76-81

[4] B. Schneier, "Applied Cryptography", 2nd Ed., *John Wiley & Sons, Inc*, 1996

[5] W. Diffie and M.E. Hellman, "New Directions in Cryptography", *IEEE Transactions on Information Theory*, v. IT-22, n. 6, Nov 1976, pp. 644-654

[6] J. P. Boly et al., "The ESPRIT Project CAFE – High Security Digital Payment Systems", *ESORICS '94*, Springer-Verlag, v. 875, 1994, pp. 217-230

[7] D. O'Mahony, M. Peirce, H. Tewari, "Electronic Payment Systems", *Artech House Publishers*, Boston, 1997

[8] K. Hickman, "The SSL Protocol", *Netscape Communications Corp.*, 501 E. Middlefield Rd., Mountain View, CA 94043, Feb. 1995, http://home.netscape.com/newsref/std/SSL.html

[9] P. Kocker, A. Freier and P. Karlton, "The SSL Protocol Version 3.0", *Netscape Communications Corp.*, March 1996, http://home.netscape.com/eng/ssl3/index.html

## URLs for this article:

**Ecash** – www.digicash.com

**Mondex** – www.mondex.com

**NTRG** – ntrg.cs.tcd.ie

**NTRG Payments Page** – ntrg.cs.tcd.ie/mepeirce/project.html