RECIPIENT SPECIFIC ELECTRONIC CASH
A Scheme for Recipient Specific Yet Anonymous and Tranferable Electronic Cash

Chittaranjan Mandal
School of Information Technology
IIT Kharagpur, WB 721302, India
Email: chitta@sit.iitkgp.ernet.in

Chris Reade
Kingston Business School
Kingston University
Email: Chris.Reade@king.ac.uk

Keywords: Electronic cash, anonymous payments, double payments, zero-knowledge proof

Abstract: A new scheme for electronic money is described where e-cash is created for a specific recipient in any transaction. This has benefits for the efficiency of implementing measures against double spending. Details of the scheme are provided to show that anonymity and transferability are still possible with recipient specific e-cash. The scheme ensures both authentication and integrity of the electronic instrument. A method for giro payments based on the scheme is also discussed.

1 Introduction

We will describe a new scheme for electronic cash to be used for electronic payments. Electronic payments may be classified as either *notational* in which electronic communication is used to access notational money stored in bank accounts to effect transfers or *token-based* where digital tokens representing stored value are transferred directly between payer and payee. The former covers credit card and debit card transactions and payment orders initiated over the Internet, whilst the later group includes use of tokens stored on prepayment cards or in electronic wallets. The scheme introduced here is for token-based (stored value) payments where the terms *electronic money* /*electronic cash*/ *e-cash* refer to the digital tokens that are stored and exchanged in transactions.

A key rationale for electronic cash schemes is that they can provide privacy and anonymity of payments as is the case with conventional cash. In contrast, notational payments allow the identity of the payer to be traced and a person's transactional history can be kept by their bank. With the combination of a rapid rise in electronic commerce and in the use of mobile devices, heavy reliance on notational electronic payments, is becoming a serious problem for privacy. Another reason for preferring electronic cash is to reduce the cost of transactions. It is desirable that the cost (often the time taken) of the transaction should be commensurate with the value being transferred. Usually the cost is determined by the particular payment scheme being used and is independent of the value being transferred. This makes many existing payment schemes unsuitable for transferring small amounts of electronic money.

A recent survey of developments in electronic money and internet and mobile payments (Committee on Payment and Settlement Systems, 2004) shows that there are a large number of different electronic payment systems either in use or under development, and several reported in earlier surveys (see e.g (Pilioura, 1998)) that are now discontinued.

After considering the context of this work by reviewing related work in the next section, the mechanisms for cash generation, payment and encashment are introduced in section 3. Subsequent sections address transferability (section 5) and double spending prevention (section 5.4) before concluding.

2 Related work

There is a considerable body of work on electronic cash mechanisms since the pioneering work of Chaum (Chaum, 1983; Chaum et al., 1990). Okamoto and Ohta (Okamoto and Ohta, 1992) list the key required properties as (i) independence (cash is se-

cure wherever it resides), (ii) security against double spending, (iii) privacy (keeping anonymity or untraceability of spenders), (iv) off-line payment, (v) transferability, and (vi) divisibility. Cryptographic techniques have been developed for many of the desired properties. In addition to common uses of cryptography for authentication, integrity and confidentiality of information, it also plays a part in ensuring anonymity of electronic money and untraceability of payers.

The key problems of independence and anonymity were addressed in the early papers (Chaum, 1983; Chaum et al., 1990). In particular, Chaum introduced the use of blind signature techniques (Chaum, 1983) for anonymity and he has several patents for these.

Double spending is a problem only for off-line payments since, for purely online systems, double spending can be detected immediately through banks keeping records of spent cash. In the latter case, double spending can be prevented rather than just detected. For partially off-line systems, a method for either preventing or at least detecting and tracing double spenders is required. This can be done easily by compromising anonymity and using a spender's credentials when cash is spent, but solutions which retain anonymity (for honest spenders) also exist. Preventing double spending with off-line systems requires hardware such as electronic purses (wallets with observers (Chaum and Pedersen, 1993b; Brands, 1994)) to control the transfer of electronic cash. However, even in these situations, it is necessary to have traceability of double spenders in case the hardware is compromised. The first approaches to traceability involved use of one-show blind signatures [8] but were problematic for efficient implementation. Stephen Brands (Brands, 1993; Brands, 1994) introduced a new technique of restrictive blind signatures to resolve the efficiency problem. This involves a method of blinding which ensures that certain information is retained in the blinded cash. This information is enough to reveal the credentials of a spender if and only if they spend more than once. Our mechanism is similar in nature, but uses different techniques to identify the double spender.

A general method for adding transferability to electronic cash systems was considered in (Chaum and Pedersen, 1993a). The latter paper showed that all proposals for transferring money must inevitably grow the size of the money and it was also proved that recognising cash that has been seen before is always theoretically possible.

Divisibility was addressed in (Okamoto and Ohta, 1992) with a more efficient mechanism proposed in (Okamoto, 1995).

This paper introduces an alternative approach to handling double spending to that proposed in previous work (e.g. (Brands, 1994)). Current schemes effectively identify double spenders but do not block double spending. This scheme of recipient-specific cash is designed with a view to blocking double spending. We do not consider divisibility techniques in this paper.

3 Generating e-cash for payment

We describe details of a payment where $\mathcal{A}$ is to pay a sum of money v to $\mathcal{B}$.

Apart from an account number $a_{\mathcal{B}}$ (which may not be confidential), we assume $\mathcal{B}$ shares a secret key $x_{\mathcal{B}}$ with his bank $\mathcal{B}^{\star}$. For a new payment, $\mathcal{B}$ first creates a nonce n, from which he can compute the following data:

$$u_{\mathcal{B}} = H(a_{\mathcal{B}}||x_{\mathcal{B}}||n) \tag{1}$$

where H is a suitable one-way hash function chosen for the scheme. (It must at least be collision-intractable). The value $u_{\mathcal{B}}$ will be used in the creation of e-cash. Note that the bank $\mathcal{B}^{\star}$ will also be able to calculate this value once it is provided with knowledge of n, but no-one else can whilst the secret key remains a secret to all but these two parties. The pair of values n and $u_{\mathcal{B}}$ thus act as credentials for $\mathcal{B}$ to the bank without direct transmission of the secret $x_{\mathcal{B}}$.

In the sequel, however, we need a more complicated form for $u_{\mathcal{B}}$ to cater for zero-knowledge proofs used for off-line payments and discussed later in section 5.3. For the new version, we additionally assume that two numbers number g and h are publicly available where g is a suitably chosen base for a group and h is a suitably chosen modulus to enable use of discrete logarithm problems (Odlyzko, 1984).Then,

$$u_{\mathcal{B}} = H(g^{(a_{\mathcal{B}}||x_{\mathcal{B}}||n)} \pmod h) \tag{2}$$

For offline and giro payments considered later, it is convenient to assume that $\mathcal{B}$ has a supply of signed values of the form $u_{\mathcal{B}}$ (each with a different nonce n). These do not have any intrinsic monetary value on their own.

To receive a payment, $\mathcal{B}$ (the recipient) creates a new secret for the payment ($s_{\mathcal{B}}$) and uses it along with $u_{\mathcal{B}}$ to compute a serial number for the e-cash $p = H(u_{\mathcal{B}}||s_{\mathcal{B}})$

The serial number then needs to be signed by $\mathcal{A}$'s bank ($\mathcal{A}^{\star}$) with a signature associating a monetary value of v. First we consider the unblinded case.

The signing process uses the bank's private key ($d = KR_{\mathcal{A}^{\star},v}$) appropriate for the chosen amount. A

corresponding amount is deducted from $\mathcal{A}$'s bank account. $\mathcal{B}$ then receives the payment from $\mathcal{A}$ which is $P = \langle p, \{p\}_d \rangle$. The payment P is thus a a pair of a serial number p and a signature of that number $\{p\}_d$, signed using the private key ($d = KR_{\mathcal{A}^\star,v}$) of the bank $\mathcal{A}^\star$ for the denomination of v. [We assume a digital signature is always accompanied by a certificate which both identifies the owner of the key used in the signature and validates the ownership.]

An important property of P is that anyone can check the signature with the bank's public key and hence verify that it has a valid form for e-cash. Another property that we will discuss later (in section 4) is that encashing P will also require knowledge of $u_\mathcal{B}$, $s_\mathcal{B}$, $x_\mathcal{B}$ and n. The use of nonces (n) ensures that different values of $u_\mathcal{B}$ are used for each payment to $\mathcal{B}$.

3.1 Blinding the payment

A technique for obtaining blind signatures and then unblinding them was first introduced by David Chaum (Chaum, 1983; Chaum et al., 1990). Anonymity of the e-cash collected from the bank by $\mathcal{A}$ can be ensured if the bank $\mathcal{A}^\star$ does not get to know the serial number of the money. Similarly anonymity can be maintained for the e-cash paid to $\mathcal{B}$.

The blinding technique for RSA is essentially a transform. For any RSA private key d and an appropriate random blinding number r there is a function blind_r and an inverse unblind_r (derived from the public key used to check signatures made with d) with the additional property that

$$\text{unblind}_r(\{\text{blind}_r(p)\}_d) = \{p\}_d$$

This means that a signature of p (namely $\{p\}_d$) can be obtained indirectly by first blinding p, then getting a signature of the blinded value $\{\text{blind}_r(p)\}_d$ and then unblinding. RSA blindings can also be chained using the further property that $\text{unblind}_r \circ \text{unblind}_s$ is an unblinding inverse for blinding with $\text{blind}_s \circ \text{blind}_r$.

In the payment, the bank can sign a, possibly multiple, blinded version of p using the key $d = KR_{\mathcal{A}^\star,v}$, so it does not get to see p. That is, $\mathcal{B}$ first blinds p to p' and passes p' on to $\mathcal{A}$. $\mathcal{A}$ in turn blinds p' to p'' (optionally) and then gets her bank ($\mathcal{A}^\star$) to sign this serial number to create the blinded payment P''. $\mathcal{A}$ (optionally) unblinds P'' to P' and returns that to $\mathcal{B}$. $\mathcal{B}$ in turn unblinds P' to get $P = \langle p, \{p\}_d \rangle$.

4 Simple encashment of the payment

$\mathcal{A}$ sends the payment P to $\mathcal{B}$. $\mathcal{B}$ would now like to encash/deposit the payment by sending P to his bank $\mathcal{B}^\star$ for deposit into account $a_\mathcal{B}$.

Recall that

$$P = \langle p, \{p\}_d \rangle, \text{ where} \quad (3)$$
$$d = KR_{\mathcal{A}^\star,v} \text{ and}$$
$$p = H(u_\mathcal{B} || s_\mathcal{B})$$

$$u_\mathcal{B} = H(a_\mathcal{B} || x_\mathcal{B} || n) \quad (4)$$

$\mathcal{B}$ is also required to send the following tuple to his bank $\mathcal{B}^\star$ to establish his own identity and knowledge of the secret used in the cash as well as to let the bank know the value n.

$$\langle H(x_\mathcal{B}), \{u_\mathcal{B} || n || s_\mathcal{B}\}_{x_\mathcal{B}}, H(u_\mathcal{B} || n || s_\mathcal{B} || x_\mathcal{B}) \rangle$$

The reasoning behind this choice is discussed below.

The hash value $H(x_\mathcal{B})$ is used by $\mathcal{B}^\star$ to identify $\mathcal{B}$, which assumes the bank maintains a sorted table of the hashes of the secret numbers of account holders. $\mathcal{B}$ needs to communicate n and $u_\mathcal{B}$ (encrypted) to the bank so the bank can verify knowledge of $(a_\mathcal{B} || x_\mathcal{B} || n)$ and thus establish his credentials. Furthermore, $\mathcal{B}$ needs to pass the secret value $s_\mathcal{B}$ (encrypted). The quantity $\{u_\mathcal{B} || n || s_\mathcal{B}\}_{x_\mathcal{B}}$ uses $x_\mathcal{B}$ as a symmetric key to encrypt the secret associated with the payment before passing it to the bank. The final element of the tuple is essentially a digest to ensure integrity of the other components of the tuple. Note that $H(x_\mathcal{B})$ is susceptible to the birthday attack (Bellare and Kohno, 2004). It can be made resistant to this attack by choosing $x_\mathcal{B}$ as a prefix of a longer string $X_\mathcal{B}$. When hash functions are computed, $X_\mathcal{B}$ in lieu of $x_\mathcal{B}$ would be used.

At the bank, the value $u_\mathcal{B}$ is checked then the decrypted value $s_\mathcal{B}$ is used with this to verify the serial number p. The bank then verifies the signature in the payment and goes on to perform its clearing.

If $\mathcal{A}^\star \neq \mathcal{B}^\star$, then $\mathcal{B}^\star$ needs to send the information $s_\mathcal{B}$ and $u_\mathcal{B}$ to $\mathcal{A}^\star$ to request the transfer of money of value v to itself. The generating bank ($\mathcal{A}^\star$) needs to keep track of whether a payment has already been honoured (and check this when a request is made). Time limits are needed to avoid banks storing this information indefinitely and this is achieved easily by assigning a "use-by date" to the e-cash, using a signature ($d = KR_{\mathcal{A}^\star,v}$, above) with a finite expiry date. If $\mathcal{A}^\star = \mathcal{B}^\star$, then the transfer step is redundant.

The generating bank only verifies $p = H(u_\mathcal{B} || s_\mathcal{B})$ and its signature of p in P. It needs to be supplied with $u_\mathcal{B}$ and $s_\mathcal{B}$ separately rather than just p because simply checking the signature of an arbitrary serial number directly is unsafe. For example, the number could be chosen so that its signature can be computed easily. If RSA signatures are used, then choosing $p = s^e \pmod{h_{\text{RSA}}}$, where $e = KU_{\mathcal{A}^\star,v}$ (using a proper RSA modulus h_{RSA}), ensures that it will have s as its RSA signature.

After successful verification, the required amount of money is transferred from the generating bank to the receiving account. Note that encashment necessarily associates the e-cash with the receiving account, but the payer remains anonymous to the bank because of blinding.

5 Transfer payments

Here $\mathcal{B}$ wishes to transfer the payment to $\mathcal{C}$ instead of encashing it.

The transfer currency serial has the form

$$q = H(u_C||s_C), \tag{5}$$

$\mathcal{C}$ blinds q to q' and passes that on to $\mathcal{B}$.

The essence of the transfer operation is to mark P as transferred and then to stamp Q as bearing the value of P (where Q is the signed version of q).

5.1 Online transfer payments

Here we assume that $\mathcal{B}$ is online with his bank $\mathcal{B}^\star$. In this case the following operations can be performed. The bank $\mathcal{B}^\star$ is given P, the currency from which Q is being derived. It first checks that P has not already been encashed or transferred and then verifies a proof of the knowledge of either $(u_C||s_C)$ or $(a_\mathcal{B}||x_\mathcal{B}||n)$ from $\mathcal{B}$. The quantity $(u_C||s_C)$ is enough to verify knowledgable possession of the currency P. This could be treated as sufficient for the bank to transfer the value of the currency to Q, in which case the transfer takes place without the bank learning the identity of $\mathcal{B}$. If it should be desirable to identify the payer $\mathcal{B}$, then the second set of values $((a_\mathcal{B}||x_\mathcal{B}||n)$ from $\mathcal{B})$ need to be verified by the bank. Identification of the payer could, for example, be a governmental regulation.

After the above step, the bank knows that P has not yet been used and that the payer has the required knowledge of the currency. It can then update its database to indicate that P has been transferred and then sign q with its signature for the denomination of the currency P. Q is the resulting signed serial number.

Note that if P bears the signature of a bank different to $\mathcal{B}^\star$, then $\mathcal{B}^\star$ can approach the bank that had signed P to get the value of P which it is transferring to Q.

Two important characteristics of this online transfer process are: (i) that the payer and the payee achieve the transfer without getting to know the serial numbers of each other's currencies (Thus anonymity of both the payer and the payee is well preserved); (ii) that there is no possibility of double spending taking place as the bank ensures that the currency is marked as transferred.

5.2 Offline transfer payments

If the payer is not online with the bank at the time of transfer, then the above online scheme cannot be used. In general, offline schemes cannot prevent double spending but the scheme described below ensures that the double spender can be detected and identified after the act. In this scheme the payee will learn the serial number of the payer's currency but not his identity. The bank of the payer will get to know the identity of the payer. Neither the payer nor the payer's bank will learn the serial number of the payee's currency.

If $\mathcal{B}$ is not online with $\mathcal{B}^\star$ when transferring the currency to $\mathcal{C}$, it is not enough for $\mathcal{B}$ to just pass on $u_\mathcal{B}$ and $s_\mathcal{B}$ to $\mathcal{C}$ because these parameters are not enough to identify $\mathcal{B}$. Such identification is needed in case $\mathcal{B}$ double spends P. The quantity $u_\mathcal{B}$ for p should be of the form $u_\mathcal{B} = H(g^{(a_\mathcal{B}||x_\mathcal{B}||n)} \pmod h)$ as in equation (2). A certified version of $u_\mathcal{B}$ was mentioned in section 3. A certified version of $u_\mathcal{B}$ was mentioned in section 3 which we will denote $U_\mathcal{B}$ here. The need for these will be explained further in section 5.5. $\mathcal{B}$ now passes on $q, U_\mathcal{B}, s_\mathcal{B}$ and also a zero-knowledge proof[1](Goldreich et al., 1991) of $(a_\mathcal{B}||x_\mathcal{B}||n)$ to $\mathcal{C}$. This proof enables $\mathcal{C}$ to verify the validity of the paramenters of the currency he is receiving. Later, when $\mathcal{C}$ is online, he can pass on $P, s_\mathcal{B}$ and this proof to the bank $(\mathcal{B}^\star)$ to get q signed by the bank as before to obtain Q. The zero-knowledge proof mentioned above can be made unique easily (with very high probability), so that if $\mathcal{B}$ were to double spend a copy of his money to $\mathcal{C}'$, then that zero-knowledge proof would be a distinct one. Without this property the double spender could blame the payees for colluding to show up copies of a proof constructed for a bonafide payment. The way in which double spending can be identified is discussed in section 5.4. For now we assert that if $\mathcal{B}$ were to double spend in trying to transfer the payment offline, then he could be identified. Also, it is necessary to link up p to q. The necessity for doing this and the method employed are explained in section 5.5.

There is evidently an asymmetry between the way the first payment and then subsequent transfer payments are made. This asymmetry is easily removed by considering the first payer in the chain as making

[1] In cryptography, a zero-knowledge proof is an interactive method for one party to prove to another that an assertion is true, without actually revealing it.

the first payment to herself ($\mathcal{A}$). Thus, $\mathcal{A}$ first generates electronic money payable to $\mathcal{A}$'s own account. In order to pay $\mathcal{B}$, $\mathcal{A}$ transfers this money to $\mathcal{B}$ using the methods described above. All the actual payments then work out as transfer payments.

5.3 Zero-knowledge proof

The zero-knowledge proof scheme utilizes the hardness of the discrete logarithm problem (Odlyzko, 1984), using a suitable (publicly known) base g and a modulus h.

Consider a line $y = mx + e$, where m is a secret and e is a uniquely chosen intercept. If the owner of the secret is challenged with x_0, then he can respond with $y_0 = mx_0 + e$. In this case the challenger can only verify y_0 knowing m and e. However, if the exponents $M = g^m \pmod h$ and $E = g^e \pmod h$ are made known to the challenger, the challenger can verify that $Y_0 = g^{y_0} = M^{x_0}E \pmod h$, without needing to know m (or e).

For the transfer payment, we need a zero-knowledge proof for m where $m = (a_{\mathcal{B}}||x_{\mathcal{B}}||n)$. Let $m' = a_{\mathcal{B}}||x_{\mathcal{B}}$ and let n be represented in l bits, then

$$(a_{\mathcal{B}}||x_{\mathcal{B}}||n) = (a_{\mathcal{B}}||x_{\mathcal{B}})2^l + n = m'k + n, \quad (6)$$

where $k = 2^l$, a constant. We have $y_0 = (m'k+n)x_0 + e = m'(kx_0) + (nx_0 + e)$. Thus,

$$Y_0 = M'^{kx_0}(g^n)^{x_0}E \pmod h \text{ where}$$

$$M' = g^{m'} \pmod h \quad (7)$$

M', g^n, E and d are disclosed to $\mathcal{C}$ for use in checking zero-knowledge proofs. In addition, we can require M' to be signed by the bank $\mathcal{B}^\star$. This is possible because m' is a fixed quantity known to both $\mathcal{B}$ and his bank $\mathcal{B}^\star$. This allows the recipient $\mathcal{C}$ to verify $\mathcal{B}$'s knowledge of $(a_{\mathcal{B}}||x_{\mathcal{B}}||n)$ and then to verify $p = H(u_{\mathcal{B}}||s_{\mathcal{B}})$. $\mathcal{C}$ can pass on this proof to $\mathcal{B}^\star$ at a later stage when he is online with $\mathcal{B}^\star$ to get the required special signature.

Everything that is done in the online transfer method also needs to be done in the offline transfer method. The only difference is that that transfer signature from the bank is taken later, when the bank becomes online and then verification of the knowledge of $(a_{\mathcal{B}}||x_{\mathcal{B}}||n)$ is done by replaying the zero-knowledge proof to the bank, in the absence of the payer ($\mathcal{B}$).

5.4 Identification of the double spender

The online process requires the bank ($\mathcal{B}^\star$) to identify the party ($\mathcal{B}$) transferring the currency and to check that there is not an attempt at double spending. Thus $\mathcal{B}$ cannot commit double spending without taking recourse to the offline transfer mechanism. In the latter case, we noted in the explanation above, that $m' = a_{\mathcal{B}}||x_{\mathcal{B}}$ is fixed for $\mathcal{B}$ and $M' = g^{m'} \pmod h$ is required in the zero-knowledge proof which is essential to the offline transfer process. Now, for all the account holders ($\mathcal{X}$), the bank can enforce a one-to-one correspondence between their $m'(\mathcal{X}) = a_{\mathcal{X}}||x_{\mathcal{X}}$, $M'(\mathcal{X}) = g^{m'(\mathcal{X})} \pmod h$ and $a_{\mathcal{X}}$ values. The bank can, therefore, efficiently associate a received $M'(\mathcal{X})$ with the corresponding $a_{\mathcal{X}}$ and hence, the account holder $\mathcal{X}$. Double spending occurs if the bank is called upon to honour a credit request for a currency with a serial number that it has already either credited or transferred (by the online process). In either case the bank has the $M'(\mathcal{X})$ (or additionally the $a_{\mathcal{X}}$) value of the double spender $\mathcal{X}$, for the $u_{\mathcal{X}}$ and also p values of the doubly spent currency. Thus, if a double payment does occur then this scheme will definitely identify the culprit with the help of his bank. This is an improvement on some other double payment prevention schemes that only identify the culprit with high probability (Tewari et al., 1998). Those schemes often have a high computation penalty or a reliance on tamper resistant devices which is not the case with this scheme.

5.5 Safety of offline transfer payments

In section 5.2 it was noted that the value of $u_{\mathcal{B}}$, as defined in equation (2) needs to be properly signed. This is because anyone who has received a transfer payment from $\mathcal{B}$ (say) knows M' (in equation (7)) and can generate new values of $u_{\mathcal{B}}$. This enables the recepient to now manufacture serial numbers for currency which can be used for spurious payments that can be traced back to $\mathcal{B}$. This is prevented as follows. $u_{\mathcal{B}}$ is signed by $\mathcal{B}$ (with a special signature key $d_{T,\mathcal{B}}$ for such transfers). This signature is again signed by $\mathcal{B}^\star$. While signing the bank needs to be sure that it is signing $u_{\mathcal{B}}$ for $\mathcal{B}$. The bank cannot be shown n, until the time the money is encashed. Therefore, $\mathcal{B}^\star$ injects the idenity of $\mathcal{B}$ by signing $\{u_{\mathcal{B}}\}_{(d_{T,\mathcal{B}})}M'$, where M' is defined in equation (7). It was discussed in section 5.3 that M' embeds the identity of $\mathcal{B}$. To prevent replays of this signature $\mathcal{B}$ now ties up the serial number p of the currency to be transferred to the blinded value of serial number q' of the new currency, by signing pq' as $\{pq'\}_{(d_{T,\mathcal{B}})}$. $\mathcal{C}$ unblinds this to get $\{pq\}_{(d_{T,\mathcal{B}})}$. This signature from $\mathcal{B}$ certifies that q was derived from p. No one else can produce such a signature and so this prevents spurious transfer currencies from being manufactured and circulated.

At the time of accepting a transfer payment the payee should verify the signature of $\{u_{\mathcal{B}}\}_{(d_{T,\mathcal{B}})}M'$ and $\{pq\}_{(d_{T,\mathcal{B}})}$ from $\mathcal{B}^{\star}$. $\mathcal{B}^{\star}$ signs the transfer currency q only if all the signatures and expressions checkout correctly.

6 Conclusions

A new scheme for electronic money has been proposed that differs from existing schemes in that e-cash is created for a specific recipient in any transaction. Details of the scheme were provided to demonstrate that both anonymity and transferability are possible with recipient specific e-cash using variations of well established mechanisms such as blinding (Chaum, 1983). Although the basic scheme is an online one, an offline version was also discussed along with details of how this would work. The mechanisms discussed ensure both authentication and integrity of the electronic instrument and support transferability both offline and online. Details of how payment confidentiality, anonymity and untracebility can be maintained by both variants were also discussed.

The online scheme naturally prevents double spending while the offline scheme identifies the double spender. Details were provided to indicate how such an identification can be made. Zero-knowledge proofs were employed as a mechanism to enable offline transfers without revealing information that could compromise anonymity. The schemes do not rely on secret splitting as discussed in (Chaum et al., 1990) and are computationally more efficient than schemes that do use secret splitting.

REFERENCES

Bellare, M. and Kohno, T. (2004). Hash function balance and its impact on birthday attacks. In *EUROCRYPT*, pages 401–418.

Brands, S. (1994). Untraceable off-line cash in wallets with observers (extended abstract). In Stinson, D. R., editor, *CRYPTO '93: Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, volume 773 of *Lecture Notes in Computer Science*, pages 302–318, Santa Barbara, California, USA. Springer.

Brands, S. A. (1993). An efficient off-line electronic cash system based on the representation problem. In *246*, page 77. Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

Chaum, D. (1983). Blind signatures for untraceable payments. In *CRYPTO '82*, pages 199–203, New York, USA. Plenum Press.

Chaum, D., Fiat, A., and Naor, M. (1990). Untraceable electronic cash. In *CRYPTO '88: Proceedings on Advances in cryptology*, pages 319–327, Santa Barbara, California, USA. Springer.

Chaum, D. and Pedersen, T. P. (1993a). Transferred cash grows in size. In *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, volume 658, pages 390–407, Santa Barbara, California, USA. Springer.

Chaum, D. and Pedersen, T. P. (1993b). Wallet databases with observers. In *CRYPTO '92: Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, volume 658, pages 89–105, Santa Barbara, California, USA. Springer.

Committee on Payment and Settlement Systems (2004) Survey of developments in electronic money and internet and mobile payments. Bank for International Settlements. CPSS Publications, number 62, March 2004.

Goldreich, O., Micali, S., and Wigderson, A. (1991). Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):690–728.

Odlyzko, A. M. (1984). Discrete logarithms in finite fields and their cryptographic significance. In *Theory and Application of Cryptographic Techniques*, volume 209, pages 224–314. Springer-Verlag, Berlin.

Okamoto, T. (1995). An efficient divisible electronic cash scheme. *Lecture Notes in Computer Science*, 963:438–451.

Okamoto, T. and Ohta, K. (1992). Universal electronic cash. In *CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pages 324–337, Santa Barbara, California, USA. Springer.

Pilioura, T. (1998). Electronic payment systems on open computer networks: a survey. In Tsichritzis, D., editor, *Electronic Commerce Objects*, pages 197–228. Centre Universitaire d'Informatique, University of Geneva.

Tewari, H., O'Mahony, D., and Peirce, M. (1998). Reusable off-line electronic cash using secret splitting. Technical report, Trinity College, Department of Computer Science, Trinity College, Dublin.