

# Extensions of Single-term Coins

Niels Ferguson

CWI, PO Box 94079 Amsterdam, The Netherlands. E-mail: niels@cwi.nl

**Abstract.** We show how the electronic cash scheme in [Fer93a] can be extended to provide  $n$ -spendable coins. Furthermore, we show how observers can be incorporated in the protocols to provide prior restraint against double spending by the user, instead of just detection after the fact.

## 1 Introduction

In [Fer93a, Fer93b] a coin system is presented that is an order of magnitude more efficient and simpler than earlier systems for electronic cash. In this paper we show two new extensions of this scheme. The first one is the construction of  $n$ -spendable coins (or  $n$ -show credentials) as opposed to the usual 1-spendable coins. These coins can be spent up to  $n$  times without the user being identified, but spending the coin a  $(n + 1)$ 'th time reveals the user's identity. The second extension is the incorporation of observers [CP93a] which allows prior restraint against double-spending while still maintaining all other properties of the coin system. Similar results for a different electronic cash scheme based on discrete logarithms are described in [Bra94, Bra93].

## 2 Multi-spendable coins

Under some circumstances it might be useful to allow the user to spend a specific coin several times. A 5-spendable coin might, for example, be used to represent a 5-trip subway ticket. The same effect can of course be achieved using five 1-spendable coins, but there are a few differences. First of all, a single 5-spendable coin requires less storage than five 1-spendable coins. On the other hand, the uses of the multi-spendable coin can easily be linked together by the Bank, so the unlinkability is lost. This makes  $n$ -spendable coins less useful for electronic cash applications, but for the subway fare the linkability does not provide any problems. It even has the slight advantage of allowing the subway company to gather statistical data on the use of the 5-trip fare.

### 2.1 Original payment protocol

The coins in [Fer93a] are based on two RSA signatures. The factorization of the RSA modulus is known only to the bank. The user Alice has three numbers  $C$ ,  $A$ , and  $B$  of a special form and two signatures  $(C^k A)^{1/v}$  and  $(C^U B)^{1/v}$ . Here,  $k$  is a random number known to the user,  $U$  is the user's identity and  $v$  is a prime

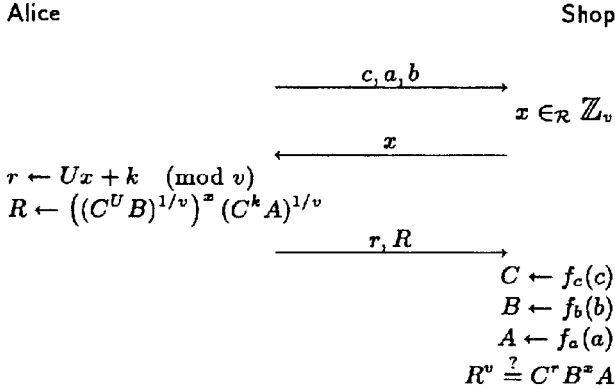


Fig. 1. Payment protocol for 1-spendable coin

large enough to make a birthday attack modulo  $v$  impossible (128 bits). The numbers  $C$ ,  $A$ , and  $B$  are all images of oneway functions on the base numbers  $c$ ,  $a$  and  $b$ . Thus  $C = f_c(c)$ ,  $A = f_a(a)$  and  $B = f_b(b)$ . The payment protocol for this coin system is shown in figure 1. When Alice wants to pay a coin to a shop, she first sends  $c$ ,  $a$  and  $b$ . The shop then generates a random challenge  $x$  which it sends to the user. Finally, Alice sends the number  $Ux + k \pmod{v}$  and the signature  $(C^{Ux+k} B^x A)$ , which she can easily construct from her two given signatures. The main aim of this protocol is to catch Alice if she spends the same coin twice. If Alice spends a coin only once, then only  $C$ ,  $A$ ,  $B$ , and  $Ux + k$  are revealed. As nobody else has any knowledge about  $C$ ,  $A$ ,  $B$  or  $k$  (this is ensured by the withdrawal protocol), these four numbers do not identify her. If she spends the same coin twice, she will receive two different challenges with high probability. If she answers them both, then Alice's identity  $U$  can easily be determined from the two answers. Note that the computations in the exponents are done modulo  $v$ . In the above protocol Alice has to apply a correction factor to  $R$  (which is not shown) to get  $(C^{(Ux+k) \bmod v} B^x A)$  instead of  $C^{Ux+k} B^x A$ . This is accomplished by dividing  $R$  by a proper power of  $C$ . In the rest of this paper we will assume implicitly that all computations of exponents are done modulo  $v$ , and that the necessary corrections are applied to the resulting signatures.

## 2.2 Achieving $n$ -spendability

We now convert these coins to  $n$ -spendable coins. The 1-spendable case uses a secret sharing line. We generalize this to use a higher degree polynomial to hide the identity  $U$  [Sha79]. For an  $n$ -spendable coin Alice stores  $n + 2$  numbers of a special form:  $C, A_0, \dots, A_n$ . She also receives  $n + 1$  signatures during the withdrawal protocol:  $(C^U A_0)^{1/v}, (C^{k_1} A_1)^{1/v}, \dots, (C^{k_n} A_n)^{1/v}$ . The modified

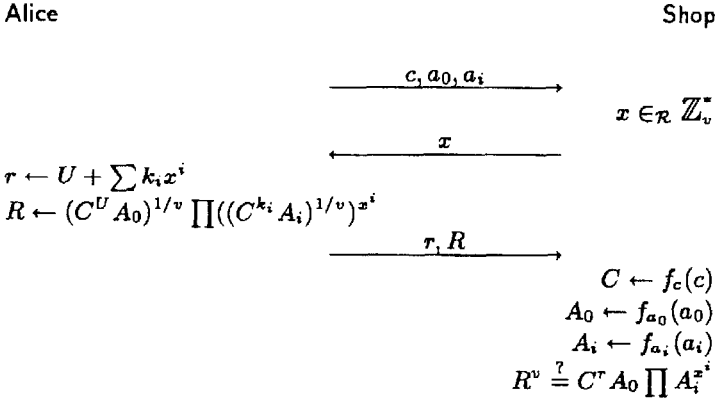


Fig. 2. Payment protocol for n-spendable coin.

payment protocol is shown in figure 2. In this figure all occurrences of  $i$  are assumed to be over the range  $1, \dots, n$ . Alice starts by sending  $c, a_0, \dots, a_n$  to the shop. The shop replies as before with a random challenge  $x$ . Finally Alice sends a point on the polynomial back to the shop, together with an RSA signature that proves it is the correct point. i.e. she sends  $r := U + \sum_{i=1}^n k_i x^i \pmod v$  and the signature  $(C^r A_0 \prod_{i=1}^n A_i^{x^i})^{1/v}$ .

If Alice spends this coin  $l$  times, then she must reveal  $l$  points on the polynomial  $\sum_{i=1}^n k_i x^i + U$ . As long as  $l \leq n$  this does not reveal any information about  $U$  (assuming that the challenge  $x = 0$  is excluded). As soon as Alice reveals  $n + 1$  points on the polynomial, the entire polynomial can easily be constructed, thereby revealing her identity  $U$ .

As always, the payment protocol is the simple part of a coin system. The withdrawal protocol is much more complicated. We show how the withdrawal protocol from [Fer93a] can be modified in a fairly straightforward manner for the  $n$ -spendable case (figure 3). It becomes the original withdrawal protocol if  $n = 1$  is substituted. (In that case,  $A_0$  takes the function of  $B$  and  $A_1$  takes the function of  $A$  when compared with figure 1.) Again, all occurrences of  $i$  should be read as running over the range 1 to  $n$ . For a description of the workings of this protocol the reader is referred to [Fer93a] or [Fer93b]. The numbers  $C, A_0$  and  $A_i$  are of the form  $f_c(c) := c g_c^{f(h_c^c)}$ ,  $f_{a_0}(a_0) := a_0 g_{a_0}^{f(h_{a_0}^{a_0})}$  and  $f_{a_i}(a_i) := a_i g_{a_i}^{f(a_i)}$  respectively, where  $f()$  is a suitable oneway function and the  $g$ 's are publicly known elements of large order in the multiplicative RSA group. The numbers  $h_c$  and  $h_a$  are elements of order  $\omega$  in  $\mathbb{Z}_p^*$  where  $\omega$  is the RSA-modulus and  $p$  is a prime with  $p \bmod \omega = 1$ .

The withdrawal protocol in figure 3 provides unconditional unlinkability between coins. For every transcript that the Bank gets from a withdrawal protocol

Alice

Bank

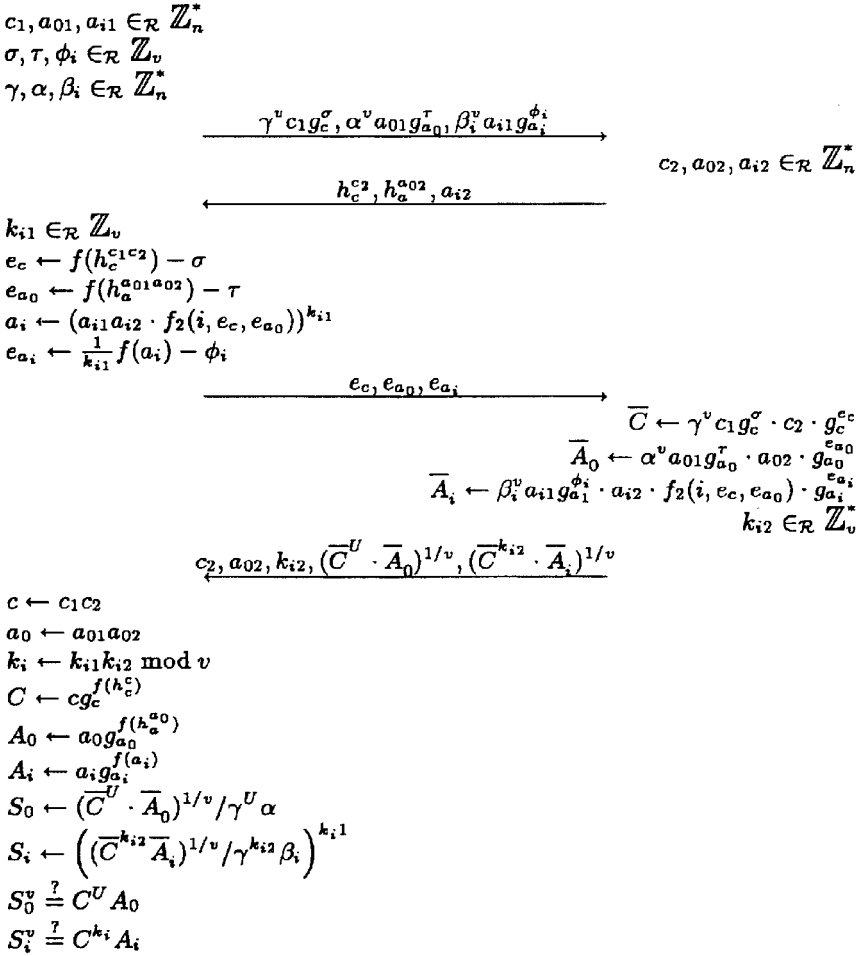


Fig. 3.  $n$ -spendable coin withdrawal protocol

and for every legal coin, there is exactly one possible set of choices out of the random choices that Alice can make that would result in her getting that specific coin from the given transcript.

### 2.3 Efficiency

The  $n$ -spendable coins are in some respects more efficient than  $n$  1-spendable coins. For a 1-spendable coin Alice must store 3 base numbers, 2 signatures and

1 random  $k$ . For an  $n$ -spendable coin these numbers are  $n + 2$ ,  $n + 1$  and  $n$  respectively. The base numbers and the  $k$ 's must be stored outright, but the signatures from different coins (with different  $v$ 's) can be multiplied together as the signatures can easily be separated again (just like in batch-RSA). As an example we will use a 512-bit RSA modulus, a 128 bit prime  $v$  and multiply the signatures together in batches of 4. A 1-spendable coin then requires 240 bytes and a 5-spendable coin requires 624 bytes. Compared to five 1-spendable coins this halves the necessary storage space.

The computational requirements are much higher for an  $n$ -spendable coin. In the 1-spendable case Alice only needs 31 modular multiplications on average to spend a coin (if  $x$  is 20 bits long). If she spends several coins at the same time, then she can use the same  $x$  for all her coins and send the shop the product of all the signatures. This uses only  $29 + 2t$  multiplies (on average) to spend  $t$  coins at the same time. The  $n$ -spendable coins require much more computations during the spending. As the  $x^i$  powers are 128 bits long (they are modulo  $v$ ), Alice needs about  $(n - 1) * 192 + 31$  multiplies on average for each of the  $n$  spendings of the coin. To spend an  $n$ -spendable coin  $n$  times therefore requires about  $192n^2 - 161n$  multiplies. Even for moderate  $n$  this is obviously inefficient compared to the  $31n$  for the  $n$  1-spendable coins.

### 3 Adding Observers

The major problem in electronic cash systems is the double spending. There is no cryptographic way in which we can prevent Alice from spending the same coin twice in an off-line system. Informally this can be shown as follows: Alice can first make a complete backup of the information in her computer and then spend a coin at shop A. She then restores all the information so that the computer is back in the same state as it was before spending the coin at shop A. Alice now spends the same coin again at shop B. The state of Alice's computer is the same for the second spending as it would have been if she never went to shop A. As we are talking about an off-line system, the state of shop B's computer after Alice spent her coin at shop A is the same as the state of shop B's computer before Alice spent her coin at shop A. As both the participants in the second spending are in the same state as they would have been if Alice had not been at shop A at all, the second payment completes successfully. Alice has succeeded in spending the same coin twice.

The only cryptographic protection against this attack is to detect the double-spending and to identify the user who did it. This is the way which is taken by all electronic cash systems [CFN90, CdBvH<sup>+</sup>90, vA90, OO92, Fer93a, Bra93, FY93].

An observer [CP93a, Cha92, Cra92, BCC<sup>+</sup>93, CP93b] is a tamper-resistant module that is incorporated in the user's computer. This is done in such a way that all communications to and from the observer is done via the user's computer. The observer is produced by a central authority and has its own native digital signature scheme. If the observer is incorporated in the electronic cash protocols

in such a way that it prevents double-spending, then this provides prior restraint against the double-spending fraud. If Alice succeeds in breaking the tamper-resistance of the observer, she can double-spend a coin but will still be caught by the underlying coin scheme. The combination of an electronic cash system with observers provides the best of both worlds: the security is only dependent on cryptographic assumptions and Alice is prevented from double-spending her coins by a tamper-resistant device.

We show how observers can be incorporated in the coin scheme of [Fer93a]. As the underlying randomized blind signature protocol for these coins is the same as the ‘validator’ protocols from [Cra92] and [BCC<sup>+</sup>93] we use the same type of construction.

Instead of giving all the information of a coin to Alice, we will keep a vital part of it in the observer. Alice will store the values  $c, a, b$ , the signature  $(C^k A)^{1/v}$  and the blinding factor  $\beta$ . The observer will store the signature  $S := (\beta^v C^U B)^{1/v}$ . The modified payment protocol is shown in figure 4. The protocol is basically the same as in figure 1 except that Alice no longer sends the signature  $(C^r B^x A)^{1/v}$  but executes a Guillou-Quisquater identification protocol [GQ88] to prove that she knows a root of  $X := (C^r B^x A)$ . This is enough to convince the shop of the accuracy of  $r$ . Alice doesn’t know the root of  $X$  by herself, but (as the protocol demonstrates) Alice and her observer together can convince the shop. (From the shops point of view, a standard GQ protocol is executed.) The way in which Alice and the observer cooperate in producing the proof is related to the diverted ‘meta’ protocols in [OO90]. Alice does a lot of additional blinding on the messages to and from the observer; these serve to prevent shared information [Cra92]. It is assumed that the observer might one day be returned to the central authority. In that case we still want to maintain the unlinkability of the payments. The blinding being done by Alice ensures that the observers transcript of a payment protocol cannot be linked to the shops transcript of the payment protocol. Some minor additional modifications have been left out for clarity. For a full discussion of these issues, and proofs of the properties of the second part of the protocol, we refer the reader to [Cra92] and [BCC<sup>+</sup>93].

It remains to be shown how we ensure the distribution of the information over Alice and her observer during the withdrawal protocol. We start again with the withdrawal protocol from [Fer93a] (see figure 3 with  $n = 1$ , and substitute  $A_0 = B, A_1 = A$ ). We want to achieve the following aims:

- Alice can only conduct a withdrawal protocol with the help of an observer.
- At the end of the protocol, Alice is left with  $c, a, b$  and  $(C^k A)^{1/v}$  and  $\beta$  while the observer gets  $\beta(C^U B)^{1/v}$ . The observer should not get any information about the values that Alice gets, and Alice should get no information about the signature that the observer gets.

Alice and the observer create a mutually random number  $\eta$  such that Alice knows  $\eta^v$  and the observer knows  $\eta$ . This is done using the elementary protocol shown in figure 5 which is a slight modification to the coin tossing protocol by Blum [Blu82].

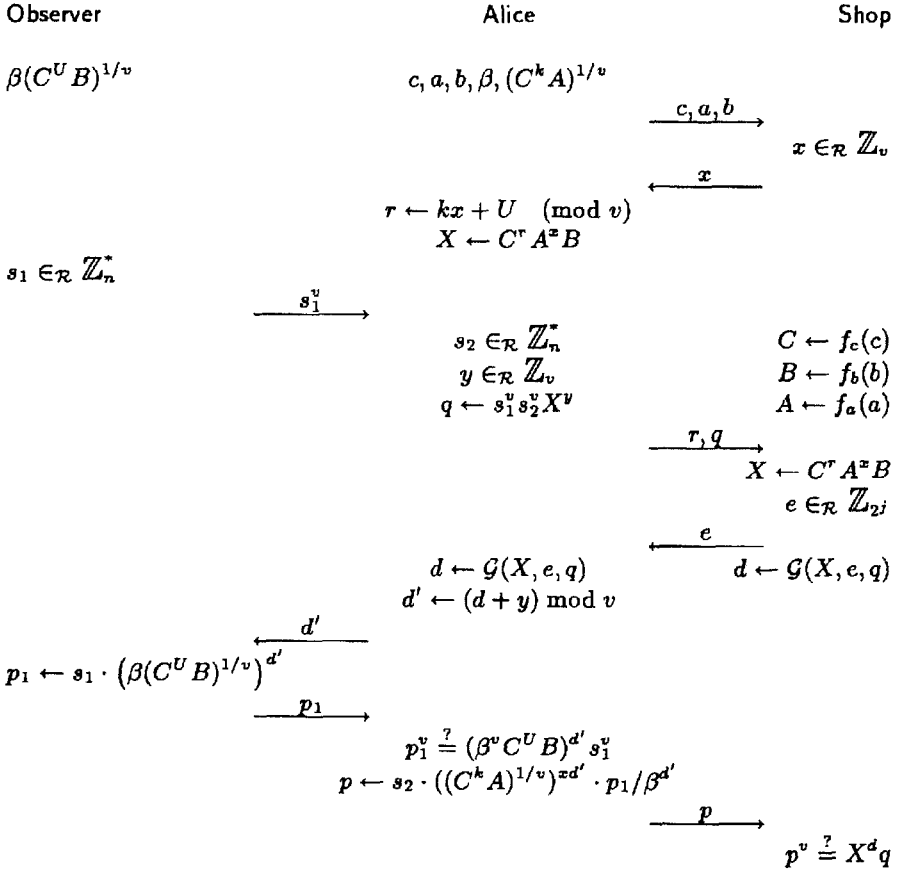
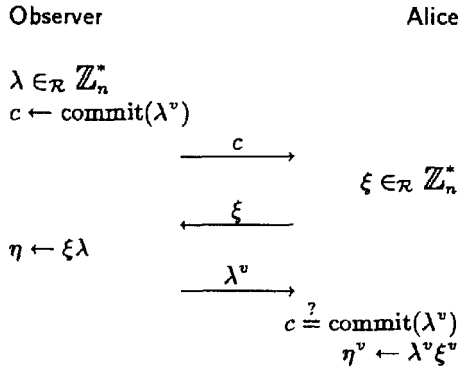


Fig. 4. Payment protocol with observer

The observer signs a message consisting of  $\eta^v$  and the data in the third transmission using its native digital signature scheme. This signature, together with  $\eta^v$  is also sent to the bank in the third transmission. The bank verifies the digital signature to ensure that  $\eta$  was indeed created in cooperation with an observer (and that therefore Alice only knows  $\eta^v$ ). In the final message the bank sends the signature  $(\overline{C^U B} \eta^v)^{1/v}$  instead of  $(\overline{C^U B})^{1/v}$ . Alice divides this by  $\gamma^U$  giving  $\eta(C^U \overline{B})^{1/v}$  which is equal to  $\beta \eta(C^U B)^{1/v}$ . She sends this number to the observer who divides out the  $\eta$  factor resulting in  $\beta(C^U B)^{1/v}$ . As Alice has no knowledge about  $\eta$  except its  $v$ 'th power, she cannot compute the signature  $(C^U B)^{1/v}$  that she needs to spend the coin by herself. During the one spending that the observer will allow, the observer only executes a Guillou-Quisquater protocol with Alice proving the knowledge of the root  $\beta(C^U B)^{1/v}$  which does



**Fig. 5.** Creation of  $\eta$

not help her in an attempt to spend the coin a second time.

There are a few other subtle changes necessary to the withdrawal protocol to get all the desired properties. These are not described here but will be included in the full paper. (For some of the details, see [Cra92, BCC<sup>+</sup>93]).

## 4 Discussion

We have shown two extensions to the single term off-line coins. This coin scheme was the first of what promises to be a new class of far more efficient electronic cash schemes (see for example [Bra93, Bra94]). The efficient implementation of coins eliminate the necessity of using checks [CFN90, vA90] with all the related organizational and security problems of refunds (e.g. [Hir93]). Although checks are often more efficient from the cryptographers point of view, they are more complicated. There is also a difficulty in finding a simple and consistent user-interface for them. Checks need to be of different denomination (if all checks were of large denominations then the user would lose too much money if she ever lost her computer), but it is hard for the average user to predict which sequence of payments will be possible with a given set of checks<sup>1</sup>. Efficient coins solve this problem as it is easy to collect a set of coins for which the computer can say “You have \$ 12.30 and I guarantee that you can make any 7 payments (as long as the total amount is below \$ 12.30)”. Users are already used to the concept of a limited number of payments as exhibited by checkbooks. For practical applications the storage requirements of coin systems are unfortunately still on the large side.

Although  $n$ -spendable coins seem attractive and are more storage-efficient,

<sup>1</sup> To protect the user’s privacy, there can be no change given by the shop



it still remains to be seen if the gained storage is worth the extra computational complexity and linkability problems.

The incorporation of observers into electronic cash protocols improves the overall functionality of the system. Banks do not like to allow their customers to cheat them and then attempt to recover the loss from the perpetrators afterwards. With observers providing the prior-restraint, the security of electronic cash is now better in all respects than any other means of payments (unless factoring is easy :-).

## References

- [BCC<sup>+</sup>93] Stefan Brands, David Chaum, Ronald Cramer, Niels Ferguson, and Torben Pedersen. Transaction systems with observers. Technical report, CWI (Centre for Mathematics and Computer Science), Amsterdam, 1993. To appear.
- [Blu82] M. Blum. Coin flipping by telephone. In *Proc. 24th IEEE Compcon*, pages 133–137, 1982.
- [Bra93] Stefan Brands. An efficient off-line electronic cash system based on the representation problem. Technical Report CS-R9323, CWI (Centre for Mathematics and Computer Science), Amsterdam, 1993. Anonymous ftp: <ftp.cwi.nl/pub/CWIreports/AA/CS-R9323.ps.Z>.
- [Bra94] Stefan Brands. Electronic cash systems based on the representation problem in groups of prime order. In *Proceedings of CRYPTO '93*, 1994. To appear.
- [CdBvH<sup>+</sup>90] David Chaum, Bert den Boer, Eugène van Heyst, Stig Mjølsnes, and Adri Steenbeek. Efficient off-line electronic checks. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology—EUROCRYPT '89*, Lecture Notes in Computer Science, pages 294–301. Springer-Verlag, 1990.
- [CFN90] David Chaum, Amos Fiat, and Moni Naor. Untraceable electronic cash. In S. Goldwasser, editor, *Advances in Cryptology—CRYPTO '88*, Lecture Notes in Computer Science, pages 319–327. Springer-Verlag, 1990.
- [Cha92] David Chaum. Achieving electronic privacy. *Scientific American*, pages 96–101, August 1992.
- [CP93a] David Chaum and Torben Pedersen. Wallet databases with observers. In *Advances in Cryptology—CRYPTO '92*, 1993. To appear.
- [CP93b] R.J.F. Cramer and T.P. Pedersen. Improved privacy in wallets with observers. In *Proceedings of EUROCRYPT '93*, 1993. To appear.
- [Cra92] Ronald Cramer. Shared information in the moderated setting. Master's thesis, Rijksuniversiteit Leiden, Netherlands, August 1992.
- [Fer93a] Niels Ferguson. Single term off-line coins. In *Proceedings of EUROCRYPT '93*, 1993. To appear.
- [Fer93b] Niels Ferguson. Single term off-line coins. Technical Report CS-R9318, CWI (Centre for Mathematics and Computer Science), Amsterdam, 1993. Anonymous ftp: <ftp.cwi.nl/pub/CWIreports/AA/CS-R9318.ps.Z>.
- [FY93] Matthew Franklin and Moty Yung. Secure and efficient off-line digital money. In A. Lingas, R. Karlsson, and S. Carlsson, editors, *Automata, Languages and Programming, 20th International Colloquium, ICALP 93, Lund, Sweden*, Lecture Notes in Computer Science 700, pages 265–276. Springer-Verlag, 1993.

- [GQ88] Louis C. Guillou and Jean-Jacques Quisquater. A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In Christoph G. Günther, editor, *Advances in Cryptology—EUROCRYPT '88*, Lecture Notes in Computer Science, pages 123–128. Springer-Verlag, 1988.
- [Hir93] Rafael Hirschfeld. Making electronic refunds safer. In *Advances in Cryptology—CRYPTO '92*, 1993. To appear.
- [OO90] Tatsuaki Okamoto and Kazuo Ohta. Divertible zero knowledge interactive proofs and commutative random self-reducibility. In J.-J. Quisquater and J. Vandewalle, editors, *Advances in Cryptology—EUROCRYPT '89*, Lecture Notes in Computer Science, pages 134–149. Springer-Verlag, 1990.
- [OO92] Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash. In J. Feigenbaum, editor, *Advances in Cryptology—CRYPTO '91*, Lecture Notes in Computer Science, pages 324–337. Springer-Verlag, 1992.
- [Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.
- [vA90] Hans van Antwerpen. Off-line electronic cash. Master's thesis, Eindhoven University of Technology, department of Mathematics and Computer Science, 1990.