How to Break and Repair a "Provably Secure" Untraceable Payment System

(Extended Abstract)

Birgit Pfitzmann[1], Michael Waidner[2]

Abstract

On Crypto '88, an untraceable payment system with provable security against abuse by individuals was presented by Damgård. We show how to break the untraceability of that system completely.

Next, an improved version of the system is presented. We also augment the system by security for the individuals against loss of money, and we introduce the possibility of receipts for payments. Finally, whereas all this concerned an on-line system, we present a similar construction for untraceable electronic cash.

1 Introduction

We start with a brief overview over untraceable payment systems. Then we give an overview over the rest of the paper.

1.1 Untraceable payment systems in general

The main characteristic of untraceable payment systems is that, like with conventional cash, the system operator (normally called "bank") cannot completely observe the payment behaviour of the individuals. Also, payer and payee may want to be untraceable by each other, or at least one of them. The need for such untraceable payments was discussed, e.g., in [Chau_85].

The first system of this kind, with several variants, was presented in [Chau_83, Chau_85], and in more detail in [Chau_89]. Common characteristics of all the variants are:

[1] Institut für Informatik, Universität Hildesheim, Samelsonplatz 1, W-3200 Hildesheim, FRG; fax +49-5121-860475; phone +49-5121-883-739;
e-mail pfitzb@infhil.uucp (via unido.informatik.uni-dortmund.de).

[2] Institut für Rechnerentwurf und Fehlertoleranz, Universität Karlsruhe, Postfach 6980, W-7500 Karlsruhe 1, FRG; fax +49-721-370455; phone +49-721-608-4024;
e-mail waidner@ira.uka.de.

J. Feigenbaum (Ed.): Advances in Cryptology - CRYPTO '91, LNCS 576, pp. 338-350, 1992.

- They work for on-line payments, i.e. the bank is needed within each payment.
- They assume that individuals have normal non-anonymous bank accounts, but money can be withdrawn from one account (in a form called "electronic coin") and deposited into another account in a way that the bank cannot trace from whom to whom the money was passed.
- The untraceability is unconditional. (Of course, suitable circumstances must be assumed: Enough coins of the same denomination must always have been withdrawn.)
- Different payments are unlinkable. The contrary would be that for several payments, it could be observed that they were made by the same person (although not by whom). Linkability may be dangerous, because it facilitates deanonymization. For instance, if an individual makes linkable payments to several organizations, and is personally known to one of them, they all can put their information together. And even if the individual is originally known to none of them, they may compare their information about the untraceable individuals' spending behaviour with the bank's information about when and how much money each real person withdrew and thus find out who is who.
- The security against abuse by individuals, in particular that they cannot make up coins themselves, or deposit coins twice, relies on a form of RSA.

The variants differ in who is untraceable by whom and in the degree of security against fraud, also by the bank, that they offer. (An efficiency improvement was presented in [Chau3_90].)

Other variants of payment systems achieve even more untraceability by eliminating the fixed accounts (but this may not always be desired, e.g., for taxing), or enable provable security against abuse on the basis of any signature scheme at the cost of a slight decrease in untraceability, or guarantee receipts for payments [BüPf_89].

Untraceable off-line payments exist, too [ChFN_90, CBHM_90]. Their security is necessarily a bit smaller than that of on-line systems can be: In on-line systems, one can prevent individuals from spending the same money twice, whereas in off-line systems, one can only detect this afterwards. (Note that all these systems do not assume the existence of tamper-proof devices.) Thus there is a risk that the individual has no money to pay back; but this risk is also accepted with some non-digital payment systems. Again, the untraceability is unconditional, whereas the security relies on a form of RSA.

On Crypto '88, an untraceable payment system with provable security against abuse by individuals was introduced [Damg_90]. This system should mostly have the same characteristics as the basic system from [Chau_89] (on-line, fixed accounts, unconditional untraceability, but not of the payee against a collusion of payer and bank, unlinkability), but additionally, the security against fraud by individuals should be provably as hard as a well-known cryptographic assumption. The price paid is that this system is far less efficient than all the systems mentioned above. Security against fraud by the bank is not considered. The cryptographic assumption is stated as "claw-free permutation pairs and a trap-door one-way function exist"; actually, the underlying protocol from [ChDG_88] assumes quadratic residuosity, but a construction of a similar protocol on the stated assumptions has been sketched by the same authors [Damg_91].

(Later, an efficient untraceable payment system, again with the same characteristics as the basic system from [Chau_89], but based on the Fiat-Shamir-scheme instead of RSA, was introduced in [OkOh_90]. Security against fraud by the bank is not considered there, either. Off-line systems on similar assumptions are contained in [OkOh1_90, OkOh_91]. However, some of the efficiency in these systems is gained by making payments linkable.)

1.2 Overview over this paper

We first show that Damgård's payment system is not untraceable at all (§2). This section also contains a description of that system. (Note that [Damg_90] also contains a credential system; we do not break that.)

In §3, we present a way to repair the untraceability, while maintaining security against abuse by individuals on the same assumptions as in [Damg_90].

In §4, we add measures to ensure security against fraud by the bank (on the even weaker assumption of any signature scheme). This is a highly desirable property for a payment system anyway; and in particular, if the clients want unconditional security against the bank being curious (or its employees, or the programmers or operators of its computers), they might also appreciate some security against losing their money. We also introduce the option of a secure exchange of a receipt for the payment.

In §5, we sketch extensions to other variants of on-line payment systems, in particular, one with maximal untraceability. We also mention how to make clients unconditionally secure against fraud. In §6, we sketch a provably secure untraceable off-line payment system. The paper ends with a warning (§7).

2 Breaking the Untraceability of Damgård's Payment System

We first describe the payment system as far as we need it (§2.1). Then we show why unconditional untraceability was claimed for it, and why this does not hold (§2.2).

2.1 Description of the system

The basic idea of Chaum's payment systems is that electronic coins are numbers of a special form, with an RSA-signature by the bank. Untraceability is achieved by having the bank sign the coin in a blinded form. Before passing the coin to the payee, the payer takes the "blinding factor" off and obtains a coin that is still signed by the bank, but unconditionally unlinkable to its previous form.

The basic idea of Damgård's system is to use a provably secure signature scheme in the sense of [GoMR_88] instead of RSA. By definition, a local transformation of one signature into a different one is not possible in such schemes. Instead, "blind signing" is performed by a two-party computation protocol between the client A (Alice) and the bank:

Protocol: For a withdrawal, A chooses a random number R as a coin number. They perform the two-party protocol from [ChDG_88] on the signing function, where A enters R as her private input and the bank its secret key SK; they use the version of the protocol where A's input is unconditionally hidden. The output, i.e. the bank's signature on R, is opened to A only.

To spend the coin, A hands R and the signature to the payee C (Chris), who passes it to the bank. If the coin has not been deposited before (the bank keeps a list of coin numbers to check this), it deposits the money to C's account.

2.2 How to trace all payments

We show that the system described above is not untraceable at all: Neither does the theorem proved about it contain what untraceability means in the real world, nor can real untraceability be proved about the system in general, nor is the system secure when it is, as suggested, used with the particular signature schemes from [BeMi_88].

Problem with the theorem: Untraceability in [Damg_90] is said to follow immediately from Theorem 3, which states that during the withdrawal protocol, the bank obtains no information in the Shannon sense about the number R it signs, i.e. the coin number. (And this theorem is true.) Thus, when the coin is deposited, the bank does not recognize it.

However, during the deposit, not only the number R, but also the signature on it is shown. Hence the theorem should contain that the bank does not obtain any information about the signature either. (Thus the actual flaw in [Damg_90] is in the middle of p. 332, in the sentence saying that Theorem 3 is sufficient for Condition 2.)

Problem with the actual systems: First consider the signature scheme from [BeMi_88]. There, each signature is a list, and the i-th signature issued with respect to a certain key is a list of length i. Thus, for each i, the bank only needs to store which individual withdrew the i-th coin, say Alice. Later, when another individual, say Chris, deposits a coin where the signature is a list of length i, the bank knows that Chris received this coin from Alice. Hence there is no untraceability at all.

Since list-like authentication is a rather special and inefficient case, consider the efficient signature scheme from [GoMR_88], too. There, tree-authentication is used. Thus, the number i of a signature can be derived from its position in the tree. (Of course, the signature is passed on without the tree, but for each part of the signature, one can see whether it is a left or right child of its parent.) Therefore, the same attack is possible, and there is no untraceability either.

Problem with the general system: In most provably secure signature schemes, the signer's input is not just the key; in particular, the schemes are not memory-less. As long as the input changes from one signature to the next, information may leak through into the signatures. Thus, in general, signature schemes with memory are unsuitable.

More generally, even if the input were only the secret key, security is not trivial: The bank might try to use different secret keys for different signatures. It is not excluded that several secret keys lead to valid signatures. If the resulting signatures are different, the bank can later see which secret key a deposited coin was signed with. Thus it need only store which key it used when which individual withdrew coins, and untraceability is lost.

3 Repairing the Untraceability

We repair the untraceability in a way that is applicable to any provably secure signature scheme: During the deposit, the signature is not shown. Instead, the payer A proves in zero-knowledge that she has it. Similar ideas can be found in the credential system in [Damg_90] and in §2.1 of [Chau_90]. In contrast to [Chau_90] (where blind RSA-signatures are used), one still needs the two-party protocol to issue the signature: Not only the existence of any signature must be proved, but of one on a coin number that has not been deposited before; hence the coin number must be shown in the deposit. Therefore, it must be hidden during the withdrawal. Thus the protocol is:

Withdrawal: Like in §2.1.

Deposit: A priori, the participants must have agreed upon an encoding of the signatures so that they are of equal length. Now A tells the payee C the coin number R. C tells R and his account number to the bank. Finally, A gives the bank a computationally convincing perfect zero-knowledge proof (or: argument) that she knows the bank's signature on R.

Proof (Sketch): Untraceability: By definition of perfect zero-knowledge, the bank and C obtain no more knowledge than the coin-number R, and that this is one of the coins a signature was given on (more precisely: that A knows a value passing the signature test). Thus, if all honest clients test each signature that they receive during a withdrawal with exactly this test, the coin can correspond to any of the previous withdrawals, since the bank does not obtain any information about the coin number during those.

The security against abuse by individuals is obviously unchanged from [Damg_90] as long as the cryptographical assumption for the soundness of the zero-knowledge proof holds. This can be the same assumption as for the two-party computation.

Alternative: Another way to repair the system is to restrict oneself to memoryless and deterministic provably secure signature schemes. Such schemes exist, see [Gold_87]; the construction can also be applied to obtain a signature scheme based on any one-way permutations [NaYu_89]. Now the secret key of the bank is fixed by commitments (unconditionally committing), and within the two-party signing protocol, it is checked that the key entered fits these commitments. The protocol from [ChDG_88] guarantees unconditional correctness of the output for deterministic computations, if the participant who unconditionally hides his secrets, i.e. A, is honest. Thus the output is a deterministic function of the coin-number and the commitments. Hence no information can leak into the signature. (Note that the commitments would be unnecessary if the signature test admitted just one signature; but this is not the case with the signature schemes considered.)

4 Security for individuals against fraud

In this section, we make the payment system provably secure for all parties for the case considered in [Damg_90]. Thus, in particular, payer and bank can trace the payee. All the measures in Sections 4, 5, and 6 are only sketched in this extended abstract.

Tasks: No security against fraud by the bank is guaranteed in [Damg_90]. (This has not been claimed, thus it is not a mistake; but it is nevertheless a desirable property.)

In particular, the bank can always claim that a coin with a particular number has been deposited before and can therefore not be deposited now. Then the bank can keep this money to itself. For the basic untraceable payment system by Chaum, this problem has already been considered at least in [Chau_89, BüPf_89].

Additionally, one should make the withdrawals secure.

If payer and payee do not trust each other, one must also take care about receipts.

Secure deposits: This problem can be solved in a conceptually simpler way than in the efficient systems: Before a withdrawal, *A* chooses a new key pair (*PK*, *SK*) of the signature scheme used, and uses *PK* as the coin number. (Thus this number can be regarded as a pseudonym of the possessor of the coin.) Now *A* can uniquely designate that she wants to spend the coin, and to whom, by signing a corresponding message with *SK*. The bank confirms the deposit for C by a signed message containing his account number and the coin number.

Proof (sketch): If the bank refuses to accept a coin, one needs a court, where the protocol is repeated. (This can also be an arbiter within the network, via whom all the messages are sent in this case.) First *A* repeats the zero-knowledge proof that she has the bank's signature on her coin, with the bank acting as the verifier. With the proof systems used, the court can check if the bank must be convinced. Also, *A*'s deposit order is tested with *PK*. If the bank now claims that the coin has already been deposited, it can only claim that this has been to C's account, since it is cryptographically hard to forge *A*'s signature on a different deposit order. Now the court asks the bank to give C the signature confirming the deposit again. Even if the coin had already been deposited (i.e. if it was not the bank who tried to cheat, but *A* and C), this does not give C additional money, since he just receives the same signature again. (The bank need not even store the exact signature if one defines that only one deposit confirmation per account number and coin number is valid.)

However, in this protocol, C had to tell *A* his account number. If C wants to be untraceable by *A*, unless the bank helps *A*, one can proceed as follows: Suppose C is known to *A* under a pseudonym C_A, and his account number is C_B. Then *A* signs that she gives the coin to C_A (and hands this to C), and C signs (using a key pair whose-public key is part of C_A) that he deposits this money on the account C_B.

Secure withdrawals: The bank might also just claim that all the money of a client has already been withdrawn. (This problem is omitted in [Chau_89], too. However, in such a system, it can easily be solved by having the payer sign an order for each withdrawal.)

Here, where the withdrawal is a two-party computation, both bank and payer must sign each message of the protocol. (The payer's public key for this purpose must be established together with the account number.) Then in the case of a dispute, the whole withdrawal can be reconstructed by a court.

Receipts: In a system guaranteeing payer untraceability only, this is easy: *A* requires a receipt after the payment is completed. If C refuses this, *A* complains to the bank, who signs a receipt instead. (And if the bank refuses, too, one needs a court again.) In the case where C should normally be untraceable, the bank must use the pseudonym C_A in the receipt, not C_B: Otherwise *A* may falsely claim that C refused a receipt, just to find out C's account number.

5 More General On-line Payments

We first sketch a payment system with maximal untraceability, and provable security against fraud for all parties. Then we show a variant with maximal untraceability if fixed accounts are required. We also note that feasible versions of these systems can be obtained with blind RSA-signatures.

Maximal untraceability and provable security: Assume that both A and C want to be untraceable and secure against fraud even if the other one colludes with the bank. (The bank, however, is still known to everybody.) We first assume there are no fixed accounts (like with cash, where the bank cannot see how much money participants receive and spend either).

Pseudonyms are public keys in the secure signature scheme used. (When someone chooses a pseudonym, of course they choose the complete key pair.) A sentence like "someone sends a message under his pseudonym X" means that the message is signed with the secret key corresponding to X, and that the recipient checks this. Also note (although this has implicitly been used all the time) that people can be addressed under pseudonyms, either because they are anonymously meeting in a shop, or, more likely, on an underlying network offering untraceability (see, e.g., [Chau_81, Chau_88]).

Assume that A is known to C under a pseudonym A_C, and C to A as C_A. Also assume that A owns a coin of a certain denomination under a pseudonym A_B, i.e. she possesses a signature of the bank on A_B.

1. C chooses a new pseudonym C_B for this payment. A and C, under their normal pseudonyms A_C and C_A, tell each other their pseudonyms A_B and C_B, and what kind of transfer they wish.
2. A, under the pseudonym A_B, tells the bank to transfer this coin to C_B. To prove that she owns a coin under this pseudonym indeed, she proves in perfect zero-knowledge that she knows the signature of the bank on A_B.
3. The bank checks that the coin has not been deposited before. Then it sends signed messages confirming the transfer of a coin from A_B to C_B to A and C (under their pseudonyms A_B and C_B). (If the bank refuses, it can be forced by a court, since it is not anonymous; if the court is an arbiter within the network, A and C can appeal under their pseudonyms A_B and C_B.)
4. C, under the pseudonym C_A, sends a receipt of the payment from A_C to A.

   If C refuses, A can use the bank's confirmation from Step 3, together with C's message from Step 1, instead, because they prove that money was transferred from A_B to C_B, and that C_B is another pseudonym of C_A, chosen for such a payment.

If C wishes, A can also send him a confirmation of the payment, indicating just the pseudonyms A_C and C_A. A refusal can be treated as with the receipt.

5. Now C transfers his coin to a pseudonym unknown to A. For this, he chooses a new pseudonym C'_B and executes the withdrawal protocol with the bank (see §2.1) with C'_B as the coin number, and all messages signed under the pseudonym C_B.

When C wants to spend the coin he has just received, he uses C'_B as A_B.

The untraceability is maximal since during the transfer, nobody obtains any information (in the Shannon sense) about a pseudonym of anybody else that will ever be used again, or has ever been used before.

Cryptographical security holds as before.

Variant with fixed accounts: If one wants to force each participant to use a fixed account (e.g. for taxing), but keep the mutual anonymity, one can split Step 5 into a deposit into C's account, and a subsequent withdrawal:

5a. C transfers his coin to a new pseudonym C'_B as above.

5b. C deposits the coin on his account, i.e. he sends a message including the account number under the pseudonym C'_B, and proves that he knows a signature on C'_B.

5c. Now he withdraws the money again, i.e. he executes the withdrawal protocol using a third pseudonym C''_B as the coin number, signing everything under his real name.

To distinguish coins that have just been received, and those that have been deposited in an account already, the bank must use different signatures in Step 5a and 5c.

Feasible RSA-versions: The two systems just described can also easily be implemented with blind RSA-signatures. Of course, the provable security is lost, but the systems become far more efficient. We sketch this for the system with maximal untraceability: In Step 2, the RSA-signature on A_B is shown directly. Instead, C_B is chosen right at the beginning as a blinded version of C'_B. Thus instead of Step 5, C can locally unblind the signature received on C_B to obtain one on C'_B. In this form, these systems are already contained in [PWP_87].

Unconditional security? Since one makes the clients unconditionally untraceable, one might also like to make them unconditionally secure against fraud. As far as signatures are only exchanged between a client and the bank, this can easily be achieved if the clients use signatures where signers are unconditionally secure [PfWa_91, BlPW_91]. Thus it works immediately as long as no receipts are needed.

If one wants to consider receipts between clients, too, or to make the bank unconditionally secure, too, one would need unconditionally secure signatures

[ChRo_90]. Since signatures need not be transferred in our case, the original version can be used. However, with these signatures, a client cannot locally choose a new public key. Hence many steps of the protocol would become multiparty computations.

6 Off-line Payments

The main problem with off-line payments is that digital money can be copied, thus someone may spend the same money several times if there is no on-line bank to check that they don't. This must be detected afterwards, so that the culprit must pay the money back (if he can be found and has enough money). At first look, this seems difficult to achieve when the payer is untraceable. However, the problem was solved in [ChFN_90] using an interactive payment protocol that achieves a very high probability ($1-2^{-k}$ for a security parameter k) that someone can be traced after spending money twice. We adapt that idea.

Now, however, we need that signatures can be passed from the payee C to the bank later, thus we need the withdrawal according to the alternative in §3.

The bank keeps one counter for each individual A, counting the coins that A has withdrawn.

1. Before the withdrawal of the v-th coin, A computes a signature S on the message "I have cheated with coin v" with her standard key and keeps it to herself. (Of course, the message can be coded much shorter.) She also chooses a pseudonym A_v for this coin (i.e. the public key of a new key pair), and k random numbers r_i of the length of S. Now she computes unconditionally hiding commitments x_i on r_i and y_i on $S+r_i$ and forms the coin number
$$R := ((x_1, y_1), \dots ,(x_k, y_k), A_v).$$
2. A and the bank use a two-party protocol to get R signed by the bank as in the remark in §3; within this protocol, it is additionally checked that R is of the proper form (i.e. A must enter the information used to construct R as additional secret inputs). The result of this check is made visible to the bank. Apart from that, R is kept unconditionally secret as above.
3. When A pays the coin to C, she hands R and the signature to C, and signs the payment under the pseudonym A_v. For each index i, C may choose whether A must open x_i or y_i.
4. To deposit the coin, C passes the complete information received from A to the bank. When A paid the same coin to two honest participants, with high probability there will be an index i such that A has opened both x_i and y_i, i.e. shown r_i and $S+r_i$. Hence S can be computed and A punished.

Measures against a collusion of A and C can be taken as in [ChFN_90].

7 Outlook

Using the brute force method of applying digital signatures and general multi- (or at least two-)party protocols, one can probably invent lots of other variants of payment system quickly. Apart from proving that the security of such systems can be based on rather weak assumptions, an advantage may be that most of these systems are conceptually simpler than their more practical counterparts; i.e. they may more easily convince a general public that it is possible to combine untraceability and security. However, since they are not very efficient at present, we stop that here.

Warning: Although we claimed that all our systems are unconditionally untraceable and provably secure, one should be careful. The security definitions we sketched were on the same level of abstractness as the previous one, which turned out to be partially wrong. Thus people would be wise not to be too convinced (even if ours are correct, as we hope), since obviously a wrong definition on this level can seem ok for quite a while. Hence one should look for definitions independent of the particular payment system considered, something like: "A payment system is an x-tuple of algorithms *pay, receive* ... such that ... honest participants never lose money except if they execute *pay* ... ". This is, however, a rather daunting task. (For instance, what is "money" in the general case, and how does one treat the case where a court is needed in between?) Some steps in that direction have been taken in [WaPf_85, ChEv_87], but the former for a simple value exchange problem, the latter rather for a credential mechanism, and both only with algebraic models of cryptographic primitives.

Acknowledgements

We are happy to thank *Andreas Pfitzmann* for helpful discussions and *Ivan Damgård* and *Tatsuaki Okamoto* for helpful letters.

References

BeMi_88 Mihir Bellare, Silvio Micali: How to sign given any trapdoor function; 20th Symposium on Theory of Computing (STOC) 1988, ACM, New York 1988, 32-42.

BlPW_91 Gerrit Bleumer, Birgit Pfitzmann, Michael Waidner: A Remark on a Signature Scheme where Forgery can be Proved; Eurocrypt '90, LNCS 473, Springer-Verlag, Berlin 1991, 441-445.

BüPf_89 Holger Bürk, Andreas Pfitzmann: Digital Payment Systems Enabling Security and Unobservability; Computers & Security 8/5 (1989) 399-416.

CBHM_90 David Chaum, Bert den Boer, Eugène van Heijst, Stig Mjølsnes, Adri Steenbeek: Efficient offline electronic checks; Eurocrypt '89, LNCS 434, Springer-Verlag, Berlin 1990, 294-301.

Chau_81 David Chaum: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms; Communications of the ACM 24/2 (1981) 84-88.

Chau_83 David Chaum: Blind Signatures for untraceable payments; Crypto '82, Plenum Press, New York 1983, 199-203.

Chau_85 David Chaum: Security without Identification: Transaction Systems to make Big Brother Obsolete; Communications of the ACM 28/10 (1985) 1030-1044.

Chau_88 David Chaum: The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability; Journal of Cryptology 1/1 (1988) 65-75.

Chau_89 David Chaum: Privacy Protected Payments – Unconditional Payer and/or Payee Untraceability; SMART CARD 2000: The Future of IC Cards, Proc. of the IFIP WG 11.6 International Conference; North-Holland, Amsterdam 1989, 69-93.

Chau_90 David Chaum: Showing credentials without identification: Transferring signatures between unconditionally unlinkable pseudonyms; Auscrypt '90, LNCS 453, Springer-Verlag, Berlin 1990, 246-264.

Chau3_90 David Chaum: Online cash checks; Eurocrypt '89, LNCS 434, Springer-Verlag, Berlin 1990, 288-293.

ChDG_88 David Chaum, Ivan Bjerre Damgård, Jeroen van de Graaf: Multiparty Computations ensuring privacy of each party's input and correctness of the result; Crypto '87, LNCS 293, Springer-Verlag, Berlin 1988, 87-119.

ChEv_87 David Chaum, Jan-Hendrik Evertse: A secure and privacy-protecting protocol for transmitting personal information between organizations; Crypto '86, LNCS 263, Springer-Verlag, Berlin 1987, 118-167.

ChFN_90 David Chaum, Amos Fiat, Moni Naor: Untraceable Electronic Cash; Crypto '88, LNCS 403, Springer-Verlag, Berlin 1990, 319-327.

ChRo_90 David Chaum, Sandra Roijakkers: Unconditionally Secure Digital Signatures; Crypto '90, 11-15 August 1990, Abstracts, 209-217.

Damg_90 Ivan Bjerre Damgård: Payment Systems and Credential Mechanisms with Provable Security Against Abuse by Individuals; Crypto '88, LNCS 403, Springer-Verlag, Berlin 1990, 328-335.

Damg_91 Ivan Bjerre Damgård: Private communication, Brighton, April 10th 1991.

Gold_87 Oded Goldreich: Two Remarks Concerning the Goldwasser-Micali-Rivest Signature Scheme; Crypto '86, LNCS 263, Springer-Verlag, Berlin 1987, 104-110.

GoMR_88 Shafi Goldwasser, Silvio Micali, Ronald L. Rivest: A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks; SIAM J. Comput. 17/2 (1988) 281-308.

NaYu_89 Moni Naor, Moti Yung: Universal One-way Hash Functions and their Cryptographic Applications; 21st STOC, ACM, New York 1989, 33-43.

OkOh_90 Tatsuaki Okamoto, Kazuo Ohta: Divertible zero-knowledge interactive proofs and commutative random self-reducibility; Eurocrypt '89, LNCS 434, Springer-Verlag, Berlin 1990, 134-149.

OkOh1_90 Tatsuaki Okamoto, Kazuo Ohta: Disposable zero-knowledege authentications and their applications to untraceable electronic cash; Crypto '89, LNCS 435, Springer-Verlag, Heidelberg 1990, 481-496.

OkOh_91 Tatsuaki Okamoto, Kazuo Ohta: Universal Electronic Cash; Crypto '91, Santa Barbara, CA, 11.-15. August 1991, Abstracts, 8.7-8.13.

PfWa_91 Birgit Pfitzmann, Michael Waidner: Fail-stop Signatures and their Application; Securicom 91, Paris 1991, 145-160.

PWP_87 Birgit Pfitzmann, Michael Waidner, Andreas Pfitzmann: Rechtssicherheit trotz Anonymität in offenen digitalen Systemen; Computer und Recht 3/10,11,12 (1987) 712-717, 796-803, 898-904; Revision: DuD 14/5-6 (1990) 243-253, 305-315.

WaPf_85 Michael Waidner, Andreas Pfitzmann: Betrugssicherheit trotz Anonymität. Abrechnung und Geldtransfer in Netzen; Proc. Datenschutz und Datensicherung im Wandel der Informationstechnologien, IFB 113, Springer-Verlag, Berlin 1985, 128-141; Revision: DuD /1 (1986) 16-22.