X-Cash: Executable Digital Cash

(Extended Abstract)

Markus Jakobsson[1] Ari Juels[2]

[1] Information Sciences Research Center, Bell Laboratories
Murray Hill, NJ 07974
markusj@research.bell-labs.com
[2] RSA Laboratories
Bedford, MA 01730
ari@rsa.com

Abstract. In this paper, we propose a new financial instrument known as *executable digital cash*, or *X-cash*. X-cash is a means of binding an offer to the accompanying goods or payment, enabling the processes of searching and paying to be unified. The result is a mechanism by which electronic trades can occur in a highly distributed setting with strong security guarantees. When a party receives an X-cash offer, he or she can verify that it is *bona fide* and can initiate a trade immediately, without contacting the originator directly. X-cash may therefore be used, among other things, to enable mobile agents to carry funds and make payments on-site without running the risk of "pick-pocketing". In this paper, we introduce X-cash, describe some variants, and sketch proofs of its security properties.

1 Introduction

The growth of the Internet and the increasing sophistication and availability of cryptographic tools have promised to bring commerce to new heights of efficiency and international breadth. Efficiency suggests a number of things, including minimized human involvement, improved distribution of goods and information, and more rapid processing of transactions. Ideally, prospective traders should be able to locate one another in a highly automated fashion and then execute trades with strong security guarantees. Until now, two trends in the research area of electronic commerce have been visible. Starting with the introduction of payment schemes to the field of cryptography by Chaum, Fiat and Naor ([7], also see [9],) research contributions have tended either to introduce new features into existing payment paradigms or to address stronger attack models. Among the new features recently introduced are off-line payments [2, 3, 14], divisibility [27, 20], and micro-payments [17, 18, 23, 25, 28, 33]. Examples of stronger attack models or improved protection against attacks include tamper-resistance [10], provable security against forgery [24], fairness [19], probabilistic on-line verification [23, 37], and revocable anonymity [4, 5, 6, 12, 15, 16, 20, 21, 22, 26, 30, 31, 32, 34, 36]. In all of these schemes, however, it has been assumed that we start at a point

where we have two parties who are aware of each other's existence and whereabouts and wish to perform a transfer of funds and merchandise. Whereas this is true for a conventional commercial setting, it is not necessarily true for the type of setting which is the main driving force of electronic commerce-namely one in which there is a large number of uncoordinated and distributed participants potentially willing to engage in barters, but unaware of each other's trade goals. It is possible in such a setting to let prospective trading partners seek each other out and then initiate peer-to-peer transactions. This, however, increases the risk of communications bottlenecks, as communicating with the originator of an offer may require costly traversals of a network. In addition, if the issuer of an offer receives many bids but has limited computational power, this means of commerce could overtax his or her resources. In order to obtain a realistic and efficient solution, we must consider alternative methods of establishing first contact between traders, and develop methods to perform a transaction without peer-to-peer contact when a desirable match is found. To do this, we may consider the *mobile agent* paradigm that has recently become the focus of much attention in the AI and distributed systems communities. Mobile agents are program segments sent across a network which execute on host machines (very much like a friendly virus). Their aim is to perform some task on behalf of the user with a certain degree of autonomy (see [29] for an overview). Proposed uses include bartering, negotiating, entertainment, monitoring, data selection and filtration, searching, and distributed processing. Current suggestions for payment schemes are not well adapted to use with mobile agents: if an agent carries digital cash, for instance, it is vulnerable to "pick-pocketing" [35]. On the other hand, not allowing agents to carry funds to perform commerce requires a reduction to the peer-to-peer setting with its attendant bottlenecks. Our aim is to avoid these two types of problems, and to supply an efficient and practical payment scheme which may be based upon any type of broadcast mechanism, including mobile agents. To this end, we propose a new financial instrument known as *executable digital cash*, or *X-cash*. X-cash is a means of binding an offer to the accompanying goods or payment, enabling the processes of searching and payment to be unified. The result is a mechanism by which electronic trades can occur in a highly distributed setting with strong security guarantees. When a party receives an X-cash offer, he or she can verify that it is *bona fide* and can initiate a trade immediately, without contacting the originator directly. The basic idea is as follows. Alice obtains from her bank a signed certificate bearing her public key PK_A and authorizing her to make payments using a corresponding secret key SK_A. Alice signs a program ω using SK_A. This program ω acts like an agent for Alice (in the usual sense of the word not related to mobile agents). It takes as input some item (e.g., a program, a news article, or frequent flier miles), and outputs the amount which Alice is willing to pay for that item. The program ω along with the certificate constitute a piece of X-cash. If Bob wishes to sell an item Q to Alice, he can take the X-cash and the item Q to Alice's bank. By running the program ω on Q, Alice's bank can determine how much to pay Bob. Alice's bank may then hold the item Q for Alice or otherwise arrange to send

it to her. The trade is thus completed in a secure fashion without any direct contact between Alice and Bob. X-cash may be regarded as an extension of the recently introduced concept of *challenge semantics* [20]. This concept uses the challenge of a payment to indicate the conditions of the barter. In its original version, it only allowed a designation of the payment to be specified. We extend the concept and the use of it by allowing any executable program to be used instead, which enables a solution to the problem of agent-based trade. Our method can be applied to any payment scheme with revocable anonymity controlled by a set of trustees, to certificate-based payment schemes without anonymity (such as [11]), and to payment schemes with on-line redemption (such as [13]).

Organization of paper

The remainder of this paper is organized as follows. Section 2 gives the definitions and notation used in the paper, describes our trust model, and formalizes the goals we are seeking to achieve. Section 3 describes how we achieve these goals using X-cash. We sketch some proofs on the security of our X-cash scheme in section 4. In section 5, we describe some extensions and improvements to the basic X-cash scheme.

2 Definitions, Model, and Goals

Definitions

Informally, an *offer* is a proposal to trade some collection of goods, moneys, or services for another collection of goods, moneys, or services according to a set of well defined terms. An offer may involve either buying and selling: the term in our usage eliminates the distinction between these two activities. Alice might, for instance, make an offer to sell 500 French francs at 5 francs per \$1, or she might make an offer to buy up to 500 French francs at \$1 per 5 francs. A *bid* is a response to an offer. If Alice is selling French francs, and Bob tenders her \$5, then Bob is making a bid. We refer generically to any entity making an offer or a bid as a *trader*. We may describe these ideas more formally in terms of an *offer function*, defined as a function $\omega : S \rightarrow T$. Here $S = \{0,1\}^*$ is the space of possible bids and $T = \{0,1\}^* \cup \phi$ is the space of possible goods, moneys, or services proposed in response to these bids. The symbol ϕ indicates a null response, i.e., the bid is deemed unacceptable. We shall use ω interchangably to indicate an offer function and the code implementing an offer function. We denote by $\omega(Q)$ the output of ω on a bid Q. Observe that ω is stateless. It does not compute, for example, based on the current time or on a history of bids. In advanced protocols which we shall touch on only briefly in this paper, S may be defined to include parameters like the current time and a lists of all bids made in response to an offer. We define an *X-cash coin* Ω to be an expression of an offer ω (as a program or a text description, or in any other form) along with all accompanying signatures, certificates, programs, and instructions. Alice

will transmit or broadcast Ω in order to initiate a trade (by means, e.g., of a mobile agent.) The aim of this paper will be to determine what form the X-cash coin must assume to achieve the flexibility and security guarantees desired in our model for electronic commerce. The system we propose will make extensive use of what we refer to as *negotiable certificates*. A negotiable certificate is an authorization, issued by a financial or other institution, for a trader to make offers using some quantity of assets held by the institution. Let (SK_A, PK_A) denote a secret/public signature key pair held by a trader Alice, and let (SK_F, PK_F) denote a secret/public signature key pair held by Alice's financial institution. A negotiable certificate C assumes the form $\sigma_{SK_F}(PK_A)$, where σ_{SK_F} denotes a signature using the secret key SK_F. (Note that the units of value of the certificate may either be left implicit, or may be specified in an extra field.) If Alice wishes to sign over a quantity m of assets to Bob, she creates the signature $\sigma_{SK_A}(Bob, m)$, and gives it to Bob along with the negotiable certificate C to be redeemed by her financial institution. Thus a negotiable certificate may be loosely regarded as a license to write checks up to a certain amount.

Trust Model

Let us now present the trust model in which we seek to conduct trades. We then give a formal statement of the goals, regarding both security and flexibility, which we are trying to achieve in this model.

Network Alice will broadcast her X-cash coin in an open network (by means, e.g., of a mobile agent which may spawn). We assume the following about this network.

1. An adversary may inject X-cash coins of her own construction into the network (such as a coin Ω' purporting to come from Alice).
2. The X-cash coin Ω may be freely read and executed by any party.
3. An adversary cannot significantly impede normal delivery of an X-cash coin. In particular, let $\mathcal{D}$ denote the total set of delivery points potentially reachable by an X-cash coin Ω. Let $p_t(D)$ be the probability that Ω reaches a delivery point $D \in \mathcal{D}$ after broadcast in a non-adversarial network in time t. Let $p'_t(D)$ be the probability that Ω reaches delivery point D in time t in a setting where at least a constant c-fraction of network servers are honest, but the rest may refuse to deliver any message. Suppose that t is such that $p_t(D) > (1-\epsilon)\lim_{t\to\infty} p_t(D)$ for all $D \in \mathcal{D}$ and for some constant ϵ s.t. $0 < \epsilon < 1$. In other words, t represents a long enough time for almost all of the broadcast to be accomplished under normal circumstances. We require that the probability distributions p_t and p'_t be polynomial time indistinguishable over coin flips of the entities in the network.
4. All parties have unimpeded access to financial institutions.

Parties We assume the following about the parties in our model.

1. Financial institutions may be trusted to act on behalf of their patrons, but not necessarily of other parties.
2. Financial institutions trust one another.[3]
3. Parties other than financial institutions are not necessarily trustworthy.

Computational assumptions We make the following computational assumptions.

1. All parties have conventionally limited computational resources (polynomial in an appropriate security parameter).
2. A digital signature scheme is employed in which it is infeasible to commit existentially forgery of signatures.

Goals of this paper Our goal is to achieve realize electronic commerce with the following properties within the trust model described above:

1. *Entitlement authentication.* Any party considering an offer ω issued by Alice must be able to determine from the X-cash coin Ω whether Alice has been issued the goods, services, or moneys being offered. This should be achievable off-line. Note that this property is different from authentication in the usual sense in that Alice's identity is not of concern (and may not even be known). Note also that entitlement authentication is a guarantee that Alice has been issued, but not necessarily that she *currently* possesses the funds or rights in question: these funds or rights may already have been spent.
2. *Fairness.* No one should be able to engage in any exchange not defined by ω. Moreover, Alice should be able to specify (in her X-cash coin) how many such exchanges she wishes to engage in.
3. *Perfect matchmaking.* Any party that receives the X-cash coin Ω should be able to engage in a fair exchange with Alice. No information beyond publicly available information and that provided by Ω is required.
4. *Integrity.* Any party must be able to verify that the X-cash coin Ω has not been tampered with.
5. *Efficiency.* The X-cash coin Ω should be compact, and offers and bids should be capable of being processed efficiently.

3 Solution

In this section, we provide details of the X-cash protocols used to achieve the goals described above. Before presenting these protocols formally, let us take a brief look at the intuition behind them. Recall that before making an offer, Alice obtains a negotiable certificate C granting her rights to the funds or rights

[3] Note that this assumption is not necessary if we make use of a fair exchange protocol, such as that proposed in, e.g., [1].

she wishes to offer, and enabling her to transfer those rights to another party. The key idea behind X-cash is the following. Alice constructs her X-cash coin Ω in such a way that the transfer of rights using C is conditional on having a suitable bid R as input to a piece of code ω. In other words, instead of signing over funds or rights to an individual, Alice signs them over based on a piece of code ω which evaluates the worth of a bid R. To make a bid, Bob creates a suitable, signed representation R of his bid, and submits it to Alice's financial institution along with Ω. This financial institution verifies that Alice's negotiable certificate still retains sufficient value for the transaction with Bob, and contacts Bob's financial institution to ensure that Bob too has sufficient funds available. The two financial institutions then process the exchange. The formal details of the protocols are given below. Note that for simplicity of notation, we assume that all signatures have full message recovery.

X-cash protocols

Initiation of trade

1. Alice has a negotiable certificate C from her financial institution F_A, attributing to her rights to all goods or moneys in T, the range of the offer function ω to be used in her X-cash. This certificate is issued against public key PK_A for which Alice holds the corresponding private key SK_A.
2. Alice decides what offer she wishes to make, and constructs an offer function $\omega : S \rightarrow T$. Again, $S = \{0,1\}^*$ is the space of possible bids and $T = \{0,1\}^* \cup \phi$ is the space of possible responses to these bids. Alice creates a piece of executable code for her offer function ω.
3. Alice decides what policy she wishes to use in accepting bids. For the sake of simplicity, we might allow three possible policies: (1) She accepts all bids until all rights attributed by C are exhausted; (2) She accepts the first j valid bids; or (3) She accepts the best bid received before date d. Alice encodes her policy choice in a field P.
4. Alice constructs the X-cash coin Ω containing $[\sigma_{SK_A}(\omega, P), C]$.
5. Alice transmits Ω.

Initiation of bid

1. On receiving Alice's offer, Bob verifies the correctness of $\sigma_{SK_A}(\omega, P)$.
2. Bob evaluates Alice's offer ω. (This may involve reading or automatically processing an attached prose description of the offer and/or executing ω on possible bids.)
3. Bob executes ω on input Q, which is his matching bid. He verifies that the output indicates acceptance of the bid, i.e., that $\omega(Q) \neq \phi$ and that the corresponding offer is as desired.
4. Bob obtains from the financial institution F_B a certificate C' bound to a public key PK_B for which Bob holds the corresponding secret key SK_B. (Note that Bob may have to perform this step earlier if ω checks certificates.)

5. Bob creates[4] a bid capsule $R = [\sigma_{SK_B}(\Omega, Q, \omega(Q)), C']$.
6. Bob sends R to financial institution F_A.

Clearing Process

1. On receiving the first bid capsule with the X-cash coin Ω, the financial institution F_A reads the policy P in Ω, verifies that Ω is correctly formed (that all signatures and certificates are valid), and then stores Ω.
2. In accordance with the policy P in Ω, the financial institution F_A collects all valid bid capsules $R_1, R_2, \ldots, R_m$ (containing bids $Q_1, Q_2, \ldots, Q_m$).
3. For each R_i in $\{R_1, R_2, \ldots, R_m\}$, the financial institution F_A does the following:
   (a) F_A checks that R_i is correctly formed.
   (b) F_A then runs ω on the bid Q_i contained in capsule R_i.
   (c) If $\omega(Q_i) \neq \phi$, then F_A checks that Alice has funds worth at least $\omega(Q_i)$ remaining against the negotiable certificate C. If not, F_A does not process R_i.
   (d) F_A checks with the appropriate financial institution F_B that there are funds to back the bid Q_i. If not, then F_A does not process R_i.
4. If Alice has sufficient funds, and there are sufficient funds remaining to support the bid Q_i, then F_A and F_B perform the exchange specified by offer and bid, as explained below.

Performing the exchange
When the two financial institutions, F_A and F_B, have agreed on an exchange as specified by Ω and some bid capsule R_i, the ownership rights need to be exchanged correspondingly. This can be done in a variety of ways, out of which we suggest two: (1) If the same public key is to be used for the newly acquired merchandise, the financial institutions simply re-issue certificates on the public keys corresponding to the new owners of the merchandise. These certificates can then be forwarded by either financial institution to the acquirers, or "picked up" by the same. (2) If a new public key is to be employed, the financial institutions may enter the old public keys of the parties acquiring the merchandise that they certify in a database, and the new owners have to supply a new public key to be certified, and prove knowledge of the secret key corresponding to the old public key in order for the exchange to occur.

4 Proofs

We claim that our basic scheme implements *entitlement authentication* (Theorem 1), *fairness* (Theorem 2), *perfect matchmaking* (Theorem 3), and *integrity* (Theorem 4).

[4] Note that the expected output of ω on Q is included in the bid in order to avoid bait-and-switch attacks in which an offer appears one way when first inspected by Bob, and in another way when redeemed by the bank.

Theorem 1: The basic scheme implements entitlement authentication, i.e., it is possible for a party examining an offer to determine that the party making the offer has been issued the rights to the goods of the offer.

Proof of Theorem 1: *(Sketch)*
Recall that Alice signs the offer using the key associated with the negotiable certificate C. The public key in C is signed by a financial institution, meaning that this institution is responsible for redeeming the value implicitly specified by the public key and certificate. Thus, by examining the signatures, Bob can ascertain that the certifying entity will redeem this value in the case of a transaction if there are funds remaining. It is not possible to forge either of these signatures, by the assumption of existential unforgeability of the corresponding signature schemes. □

Theorem 2: The basic scheme implements fairness, i.e., no one should be able to engage in an exchange not defined by the corresponding offer and bid.

Proof of Theorem 2: *(Sketch)*
First, a bid is made with respect to an offer in a binding way: a matching offer constitutes a pair of offer and bid. In particular, Bob signs both offer and bid together, so that they may not be dissociated without forgery or alteration of his signature. Likewise, Alice protected the integrity of the offer by signing it. Therefore, the scheme implements fairness under the assumption that the financial entities will not steal resources. □

Theorem 3: The basic scheme implements perfect matchmaking.[5] In other words, any party with a strategy for producing valid bids and appropriate access to broadcasts of an offer Ω should be allowed a fair exchange based on Ω.

Proof of Theorem 3: *(Sketch)*
By assumption 3 about the broadcast network, it is not possible for an adversary to impede the broadcast of an X-cash coin Ω significantly. In particular, any party which has access to a distribution point $D \in \mathcal{D}$ such that $p_t(D)$ is significantly large for suitable t will obtain Ω with high probability even in the face of an adversarial attack. By assumption 4 about the broadcast network, bids will arrive at the appropriate financial institution unimpeded. Having collected bids in accordance with the policy P specified in Ω, the financial institution backing the offer will process all matching bids. Selected offers and bids will then be resolved atomically by the financial institutions backing the funds of the offer and the selected bids. By Theorem 2, the resulting trade will be fair. □

Theorem 4: The basic scheme implements integrity, i.e., any party must be able to verify that a given offer capsule has not been tampered with.

[5] We note that if the selection strategies governing how matches are made are very complex, then the computational task of finding the "best fit" is significant. We can only hope for heuristic matchmaking schemes to be "almost perfect". The work of matching received offers and bids, however, is outside the scope of this paper. We assume that there is a mechanism for selection of offers and bids in place, and for simplicity, that this mechanism effects perfect matches.

This follows automatically from the use of digital signatures to authenticate offers; if it is possible to tamper with an offer capsule, this breaks the assumption that the corresponding signature scheme is existentially unforgeable.

5 Extensions

There are a number of possible ways of extending the functionality of X-cash. We will touch briefly on some of these in this section.

5.1 Anonymity

The ability to perform financial transactions anonymously has been of major concern to proponents of digital cash since its inception. Anonymity is of equal or greater importance in X-cash transactions, particularly as a single coin may be viewed openly by many parties. X-cash may be rendered anonymous by essentially the same means as traditional e-cash. Many off-line anonymous cash schemes, however, have mechanisms for protecting against overspending by the use of thresholds. Since redemption of X-cash occurs on-line, these mechanisms are not relevant here. On the other hand, schemes with perfect privacy and on-line redemption (e.g. [8]) are quite suitable for use with X-cash, as are many of the schemes with anonymity controlled by trustees.

5.2 Stateful offers and bids

In the body of this paper, we consider only stateless offers, i.e., offers ω which take as input a single bid. In some situations, though, the party making an offer may wish to take into account the value of multiple bids or other information simultaneously. For this reason, it may be desirable to extend the scope of the offer function ω to allow for a range of possible inputs and outputs, and also to change the policy field P. We sketch a couple of examples here:

- Alice has 50,000 frequent flier miles to sell. She is willing to sell them piecemeal, but wishes to dispose of as many as possible in the next month. Alice therefore indicates in her policy description P that the Bank should collect all bids $Q_1, Q_2, ..., Q_n$ over the next month and then run ω on them, processing all bids output by ω. Alice constructs an offer program ω which finds and outputs the subset of bids among $Q_1, Q_2, ..., Q_n$ whose sum is as close as possible to but not greater than 50,000.
- Alice wishes to sell a 6 ounce gold bar for its market price on the day of sale. She obtains from her Bank a negotiable certificate of entitlement to the gold and constructs an offer program ω. When given a bid Q, the program ω goes out onto the Web, checks the current price per ounce d of gold bullion, and outputs "yes" if $Q \geq 6d$, and "no" otherwise. Alice indicates in P that her Bank should redeem any bid Q which yields a "yes" output. (Note that the state in ω is external in this example.)

5.3 Secret Strategies

We have just demonstrated how it is possible to enhance the offer program to make X-cash more flexible. It is equally possible to make enhancements to the policy statement P. This may be particularly useful if Alice wishes to pursue what we refer to as a *secret strategy*, i.e., if she wishes for her method for selecting among bids to remain concealed from potential trading partners. She may be accomplish this by constructing a piece of X-cash of the form $\Omega = \sigma_{SK_A}(\omega, E_{PK_F}[P], C)$, where PK_F is the public key of Alice's issuing financial institution. Consider the following scenario. Alice wishes to sell one million shares of Mata Hari Crypto Corp., Inc.-a controlling interest-at the price of \$100/share. She does not want anyone to know how large a block of stock is being sold, and wants to avoid having any one individual accumulate too many shares from the offering. Alice may accomplish this by constructing an offer program ω which takes as input a bid $\$Q$ and outputs "$Q/100$ shares". She constructs a policy P stating that any bid for more than 10,000 shares should be rejected. Alice includes an encryption of P in her X-cash coin as described above. Note that if complex policy statements are permitted, then it may be beneficial for P to take the form of a program whose inputs are bids and timestamps associated with these bids and whose outputs are accepted bids.

6 Acknowledgments

The authors wish to express thanks to Burt Kaliski and Marty Wattenberg for their many helpful suggestions on this paper.

References

1. N. Asokan and Victor Shoup, "Optimistic fair exchange of digital signatures (to appear)," In Kaisa Nyberg, editor, Advances in Cryptology - EUROCRYPT '98, number to be assigned in Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 1998.
2. S. Brands, "Untraceable Off-line Cash in Wallets with Observers," Advances in Cryptology - Proceedings of Crypto '93, pp. 302–318.
3. S. Brands, "An Efficient Off-line Electronic Cash Systems Based on the Representation Problem," C.W.I. Technical Report CS-T9323, The Netherlands.
4. E. Brickell, P. Gemmell and D. Kravitz, "Trustee-based Tracing Extensions to Anonymous Cash and the Making of Anonymous Change," Proc. 6th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), 1995, pp. 457–466.
5. J. Camenisch, U. Maurer and M. Stadler, "Digital Payment Systems with Passive Anonymity-Revoking Trustees," Computer Security - ESORICS 96, volume 1146, pp. 33–43.
6. J. Camenisch, J-M. Piveteau and M. Stadler, "An Efficient Fair Payment System," Proceedings of the 3rd ACM Conference on Computer and Communications Security, 1996, pp. 88–94.
7. D. Chaum, A. Fiat and M. Naor, "Untraceable Electronic Cash," Advances in Cryptology - Proceedings of Crypto '88, pp. 319–327.

8. D. Chaum, "Blind Signatures for Untraceable Payments," Advances in Cryptology - Proceedings of Crypto '82, pp. 199-203.
9. D. Chaum, "Achieving Electronic Privacy," Scientific American, August 1992, pp. 96–101.
10. D. Chaum and T. Pedersen, "Wallet databases with observers," Advances in Cryptology - Proceedings of Crypto '92, pp. 89–105.
11. CitiBank and S. S. Rosen, "Electronic-Monetary System," International Publication Number WO 93/10503; May 27 1993.
12. G.I. Davida, Y. Frankel, Y. Tsiounis, and M. Yung, "Anonymity Control in E-Cash Systems," Financial Cryptography 97, pp. 1–16.
13. DigiCash' payment scheme; `http://www.digicash.com`
14. N. Ferguson, "Extensions of Single-term Coins," Advances in Cryptology - Proceedings of Crypto '93, pp. 292–301.
15. Y. Frankel, Y. Tsiounis, and M. Yung, "Indirect Discourse Proofs: Achieving Efficient Fair Off-Line E-Cash," Advances in Cryptology - Proceedings of Asiacrypt 96, pp. 286–300.
16. E. Fujisaki, T. Okamoto, "Practical Escrow Cash System", LNCS 1189, Proceedings of 1996 Cambridge Workshop on Security Protocols, Springer Verlag, pp. 33 – 48.
17. S. Glassman, M. Manasse, M. Abadi, P. Gauthier and P. Sobalvarro, "The Millicent Protocol for Inexpensive Electronic Commerce," In World Wide Web Journal, Fourth International World Wide Web Conference Proceedings, O'Reilly, December 1995, pp. 603–618.
18. R. Hauser, M. Steiner and M. Waidner, "Micropayments Based on iKP," 14th Worldwide Congress on Computer and Communications Security Protection, 1996, pp. 67–84.
19. M. Jakobsson, "Ripping Coins for a Fair Exchange," Advances in Cryptology - Proceedings of Eurocrypt '95, pp. 220–230.
20. M. Jakobsson and M. Yung, "Revokable and Versatile Electronic Money," 3rd ACM Conference on Computer and Communications Security, 1996, pp. 76–87.
21. M. Jakobsson and M. Yung, "Distributed 'Magic Ink' Signatures," Advances in Cryptology - Proceedings of Eurocrypt '97, pp. 450–464.
22. M. Jakobsson and M. Yung, "Applying Anti-Trust Policies to Increase Trust in a Versatile E-Money System," Advances in Cryptology - Proceedings of Financial Cryptography '97, pp. 217–238.
23. S. Jarecki and A. Odlyzko, "An Efficient Micropayment System Based on Probabilistic Polling," Advances in Cryptology - Proceedings of Financial Cryptography '97, pp. 173–191.
24. A. Juels, M. Luby and R. Ostrovsky, "Security of Blind Digital Signatures," Advances in Cryptology - Proceedings of Crypto '97, pp. 150-164.
25. C. Jutla and M. Yung, "Paytree: 'Amortized Signature' for Flexible Micropayments," 2nd USENIX Workshop on Electronic Commerce, November 1996.
26. D. M'Raïhi, "Cost-Effective Payment Schemes with Privacy Regulation," Advances in Cryptology - Proceedings of Asiacrypt '96.
27. T. Okamoto, "An Efficient Divisible Electronic Cash Scheme," Advances in Cryptology - Proceedings of Crypto '95, pp. 438–451.
28. R. Rivest and A. Shamir, "PayWord and MicroMint: Two Simple Micropayment Schemes," Cryptobytes, vol. 2, num. 1, 1996, pp. 7–11.
29. D. Rus, R. Gray and D. Kotz, "Transportable Information Agents", 1st Intl. Conf. Autonomous Agents, 1997.

30. S. von Solms and D. Naccache, "On Blind Signatures and Perfect Crimes," Computers and Security, 11 (1992) pp. 581–583.
31. M. Stadler, "Cryptographic Protocols for Revokable Privacy," PhD Thesis, ETH No. 11651, Swiss Federal Institute of Technology, Zürich, 1996.
32. M. Stadler, J-M. Piveteau and J. Camenisch, "Fair Blind Signatures," Advances in Cryptology - Proceedings of Eurocrypt '95, pp. 209–219.
33. J. Stern and S. Vaudenay, "SVP: a Flexible Micropayment Scheme," Advances in Cryptology - Proceedings of Financial Cryptography '97, pp. 161–171.
34. Y. Tsiounis, "Efficient Electronic Cash: New Notions and Techniques," PhD Thesis, College of Computer Science, Northeastern University, 1997. `http://www.ccs.neu.edu/home/yiannis`
35. B. Venners, "Solve Real Problems with Aglets, a Type of Mobile Agent," Javaworld, May 1997.
36. B. Witter, "The Dark Side of Digital Cash," Legal Times, January 30, 1995.
37. Y. Yacobi, "On the Continuum Between On-line and Off-line E-cash Systems - I," Advances in Cryptology - Proceedings of Financial Cryptography '97, pp. 193–201.

This article was processed using the LaTeX macro package with LLNCS style