RZ 2791 (# 89269) 02/12/96
Computer Science / Mathematics 10 pages

Research Report

Micro-Payments based on *i*KP

Ralf Hauser, Michael Steiner, and Michael Waidner

IBM Research Division
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

IBM Research Division
Almaden · T.J. Watson · Tokyo · Zurich

Micro-Payments based on *i*KP

Ralf Hauser, Michael Steiner, and Michael Waidner

IBM Research Division, Zurich Research Laboratory, 8803 Rüschlikon, Switzerland

Abstract: Micro-payments are payments too small in amount to warrant the overhead costs of current financial clearing networks. Furthermore one can expect that content servers for the global information infrastructure (GII) will have to process so many of these low value transactions that computationally complex and costly cryptographic protocols will be impractical. This report proposes a micro-payment scheme that can be bootstrapped with the already well-known payment protocols for larger amounts, but does not depend on them for each micro-transaction. Special attention is given to its integration into IBM's Internet Keyed Payment Systems (*i*KP).

1 Introduction

Micro-payments have a broad application area in the marketing of information distributed in an electronic form. Modern network information browsing tools (WWW [1]) enable users/clients to wander arbitrarily through the global networks and obtain such documents.
We assume that a specific client normally is consuming enough low-value documents from a given server that all these low-value transactions can be accumulated in one regular payment transaction with a normal amount. For the case where clients show a *non-repetitive consumption pattern* with respect to the servers/seller they buy goods from, we require the inclusion of a third party such as a *micro-payment broker.*
In this paper we propose a computationally cheap but nevertheless secure and non-repudiable micro-payment scheme which is bootstrapped on payment protocols for larger amounts. A concrete proposal is based on the *Internet Keyed Payments Protocol (i*KP*)* [2, 3] and its extension for a split of authorization and clearance.
In the next section, we cite previous work dealing with micro-payments and discuss the additional mechanism based on secure digest functions. In Section 2 we outline the fundamental architectural types of micro-payments. In Section 4, the solution for repeated micro-payments is discussed and in Section 5, it is adapted to the case where the micro-payments are not repeated. Section 6 outlines further and open issues.

2 Previous Work

This section first sketches current proposals for online micro-payments. It then cites a recent architecture for micro-payments based on an electronic purse. The latter architecture, subsequently described, contains a "coupon" mechanism that is the core around which our micro-payment system was built.

2.1 Software-Only Architectures

The main two existing proposals for online micro-payments are the NetBill Security and Transaction Protocol [4] and Millicent [5]. They all conclude that digital signatures are not affordable and that the repudiable security of shared keys and secure digest functions for Message Authentication Codes (MAC) is sufficient in light of the small monetary sums at stake.
They all advocate third parties with brokerage functions and a trust relationship of that broker either to the buyer or to the seller or to all parties involved in the transaction:

- An account server, called NetBill server, maintains accounts for both buyers and sellers. NetBill acts as an aggregator to combine many small transactions into larger conventional transactions, thus amortizing conventional overhead fees. Therefore, both parties have to trust the third party.

- In Millicent, each vendor only accepts "scrips" it has issued and authorized itself previously. By efficient double-spending detection, it can therefore avoid financial risk. The client, however, must fully trust both the broker providing the scrip valid for a certain server and the vendor who accepts it that the scrip will be honored.

Evaluation of current, software-only broker architectures

- **Efficiency/Code Complexity**: The cost/latency of establishing a connection to a third party to obtain some token most likely alienates the buyer's gain of not having to compute a digital signature for the payment, but this gain persists on the seller's side, who is expected to be the bottleneck in such transactions.
- **Security/Non-Repudiability**: Under the assumption that systematic cheating can be detected, the enforcement of proper business behavior is assumed to be achieved by the market forces.
- The two proposals mentioned so far do not require tamper-resistant hardware like smartcards or electronic wallets at the buyer's site.

If such devices are available, several further scenarios exist.

2.2 Architectures Relying on Tamper-Resistant Hardware

One possibility is to use electronic purse schemes. Typically these schemes rely on fast symmetric cryptography and require tamper-resistant hardware at both the buyer's (smartcard) and the seller's (POS-terminal) site. More advanced schemes use digital signatures. Often, payments are already accumulated at the seller's site, i.e., no individual clearing is necessary. Fast payment plus accumulation at the seller's site would make them very attractive for micro-payments. The main disadvantage is that buyer and seller would need additional hardware.

In an electronic purse scheme each micro-payment would be a complete payment. Another approach was taken by the CAFE payment system [6]. CAFE is also based on tamper-resistant hardware at the buyer's site but uses digital signatures and provides a high degree of anonymity for payments. Micro-payments are considered in CAFE only for the special application of *phone calls* – where the problem is to pay connection costs *tick-by-tick*. The trick applied in CAFE is the same we use in the following (see Section 2.3), namely, bootstrapping a Winternitz signature and performing micro-payments by revealing *pre-images* [7].

2.3 Basic Construction

Our construction for repeated micro-payments is based on a computationally secure one-way function f. Informally, a function is one-way if it is difficult to find a value X for an image Y randomly chosen from the range of f. In fact, we go a bit further and require that f is even one-way on its iterations. Good practical candidates for f are MD5 [8] or SHA [9].

Given such a one-way function f, the buyer will randomly choose a value X and will recursively compute

1. $A^0(X) = X$
2. $A^{i+1}(X) = f(A^i(X))$.

We call the values $A^0, ..., A^{n-1}$ *coupons.* These n coupons will enable the buyer to make n micro-payments of fixed value v to one seller:

Bootstrapping: The buyer forwards A^n to the seller, together with the value v per coupon, and authenticates them both. This authentication is done using an arbitrary payment system that authenticates the amount of the payment (e.g., iKP) by replacing it by (A^n, v, n). All n micro-payments can be authorized at once.

Micro-Paying: The micro-payments themselves are performed by successively revealing A^{n-1}, A^{n-2}, ..., A^0 to the seller.

Note that this mechanism preserves the security of the payment system used to authenticate (A^n, v):

Each individual micro-payment is digitally signed by the buyer with a highly efficient but specialized signature scheme. Thus each of these coupons provides non-repudiation. However, this shows only the fact that the buyer wanted to pay something, but not what he wanted to pay for.

Several applications of this idea are known: In the early 80's, *Winternitz* suggested that chains of coupons can be used to implement efficient one-time signatures[1] [10]. In 1981, *Lamport* applied the idea to the problem of dynamic passwords [11] and most recently *Pedersen* applied the Winternitz idea to micro-payments [7]; as mentioned above the scheme is part of the CAFE payment system. Independent from our work two other groups came up with very similar schemes [12, 13].

Our proposal is very similar to Pedersen, but it provides the following improvements:

- The coupon-chains are securely bootstrapped with *i*KP without the need for secure hardware as in CAFE.
- Means are provided to protect the integrity of the product description for which the coupon is revealed.
- Brokers are introduced that enable the use of the coupon mechanism also for non- or rarely repeated purchase patterns. Compared to other online- and broker-based approaches, the need for trust in this broker is minimal.

3 Market Patterns

The general market model assumed here is the one of *i*KP [2] with a buyer (B), a seller (S), and the existing financial networks subsumed under the notion of an *acquirer* (-gateway) (A); see Figure 1.

If all parties involved have a public and private key pair, it is possible to execute a regular credit card transaction securely over arbitrarily wide-area networks and achieve the goals of non-repudiation and maximally confining the parties, e.g., by providing partial anonymity (need-to-know principle). *i*KP contains an option to authorize an amount first and do the clearing only later. This *guarantee* by the financial network of somebody's ability to pay will become a centerpiece of the following proposals.

We will distinguish between two forms of market behavior patterns:

1. repeated micro-transactions
2. singular micro-transactions.

Under the assumption that no trusted hardware restricts the buyer in his freedom to participate in the protocols, it appears that singular micro-transactions always need a trusted third party[2] - brokerage system.

However, if there exist repeated relations with the same seller, there is no need for a third party. Based on the mechanism sketched in Section 2.3 we will describe how this can be achieved with *i*KP.

[1] In fact, our application can be described as using Winternitz signatures to sign each micro-payment, and to authenticate the public key of the Winternitz signatures (i.e., A^n) like the amount of a payment.

[2] Thus, the acquirer is now the fourth party.

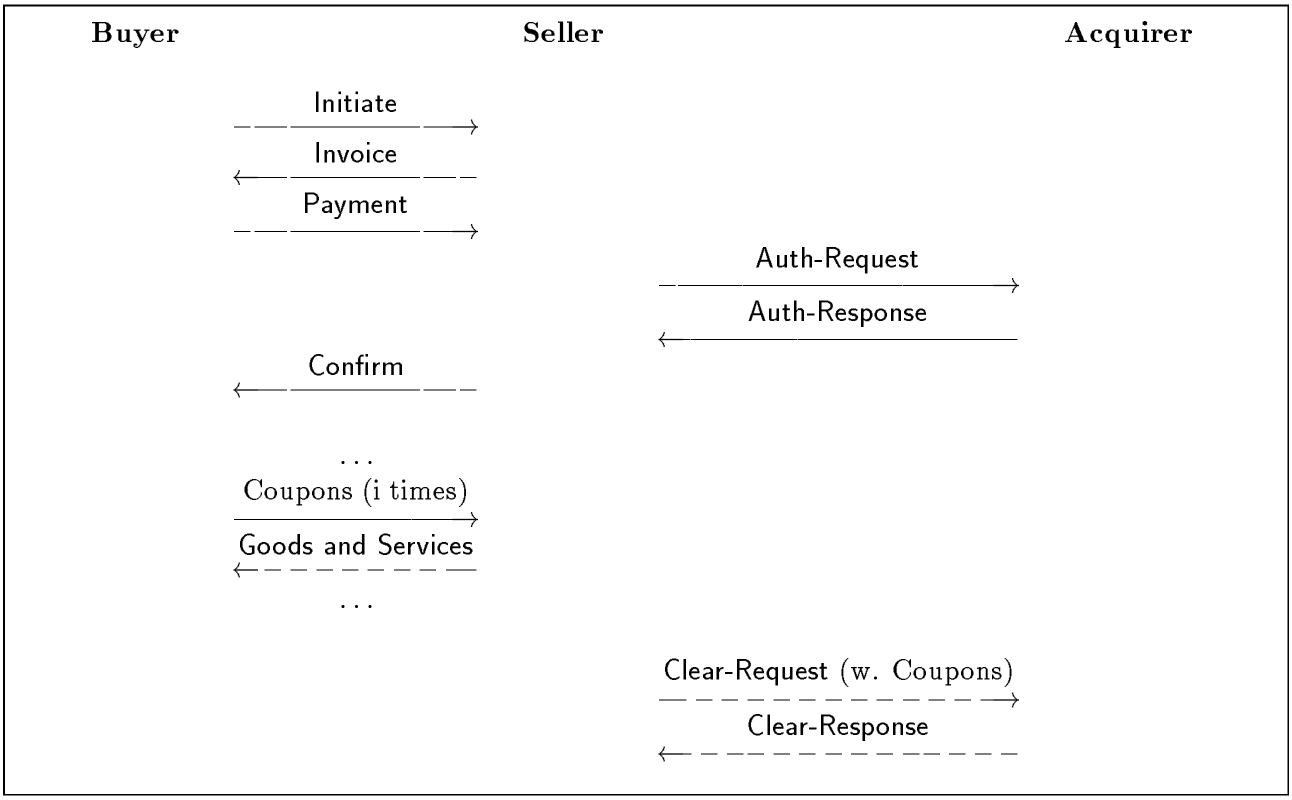

Figure 1: *Framework of* μ-iKP *Protocol*

4 Repeated Micro-Payments

4.1 Initialization of a Repeated Micro-Payment Relationship with μ-iKP

In iKP there is a set of information called COMMON that is shared by all parties involved. It contains the parameter PFLAGS, which must have an additional option to accommodate micro-payments.
The buyer then chooses the "root" of a pre-image chain: P_B. Furthermore, the buyer calculates the n descendants of P_B with the mentioned secure one-way function and stores them as a chain of pre-images[3]. The buyer sends now the 3KP PAYMENT message to the seller to initiate a 3KP whereby the authorization is split from clearance. $A^n(P_B)$ is included in COMMON together with the agreed amount per coupon and the length n of the chain. This way the buyer commits himself to the chosen chain. The seller proceeds with normal authorization. Figure 2 illustrates this scenario.

4.2 Micro-Spending

After successful authorization, the micro-transactions may begin. If the limit is $n = 1000$, the buyer begins by releasing coupon $A^{999}(P_B)$ for the first item to be purchased. For any other micro-payment in a micro-transaction, the lower-numbered subsequent pre-images ($i = 998$, $i = 997$, $i = 996$, ...) are released for payment.

[3] Memory vs. runtime optimization could also advocate that certain parts of the chain be recalculated upon demand.

- **Composite Fields:**

Common	$A^n(P_B)$, n, $Value/Coupon$,$\mathcal{H}(\mathrm{DESC}, \mathrm{SALT}_B)$, ID_S, TID_S, DATE, NONCE_S, $\mathrm{ID_B}$, $\mathcal{H}(V)$, $\mathcal{H}(V')$, PFLAGS
Clear	ID_S, DATE, NONCE_S, $\mathcal{H}(V)$, $A^n(P_B)$, $\mathcal{H}(\mathrm{Common})$
SLIP	n, $Value/Coupon$, $\mathcal{H}(\mathrm{Common})$, BAN, R_B.
EncSlip	$\mathcal{E}_A(\mathrm{SLIP})$
SIG_S	$\mathcal{S}_S(\mathcal{H}(\mathrm{Common}), \mathcal{H}(V)), \mathcal{H}(V')$
SIG_B	$\mathcal{S}_B(\mathrm{EncSlip}, \mathcal{H}(\mathrm{Common}))$

- **Protocol Flows:**

Initiate: B $\xrightarrow{A^n(P_B),\ \mathrm{SALT}_B,\ \mathrm{ID_B},\ \mathrm{CERT}_B}$ S

Invoice: B $\xleftarrow{\overbrace{\mathrm{ID}_S,\ \mathrm{TID}_S,\ \mathrm{DATE},\ \mathrm{NONCE}_S,\ \mathcal{H}(V),\ \mathcal{H}(\mathrm{Common}),}^{\mathrm{Clear}}\ \overbrace{\mathcal{S}_S(\mathcal{H}(\mathrm{Common}), \mathcal{H}(V))}^{\mathrm{SIG}_S}}$ S

Payment: B $\xrightarrow{\overbrace{\mathcal{E}_A(\mathrm{SLIP})}^{\mathrm{EncSlip}},\ \overbrace{\mathcal{S}_B(\mathrm{EncSlip}, \mathcal{H}(\mathrm{Common}))}^{\mathrm{SIG}_B}}$ S

Auth-Request: S $\xrightarrow{\mathrm{Clear},\ \mathcal{H}(\mathrm{DESC}, \mathrm{SALT}_B),\ \mathrm{EncSlip},\ \mathrm{SIG}_S,\ \mathrm{SIG}_B}$ A

Auth-Response: S $\xleftarrow{\mathrm{Y/N},\ \overbrace{\mathcal{S}_A(\mathrm{Y/N}, \mathcal{H}(\mathrm{Common}))}^{\mathrm{SIG}_A}}$ A

Confirm: B $\xleftarrow{\mathrm{Y/N},\ V,\ \mathrm{SIG}_A}$ S

. . .

$Micro - Payments$: B $\xrightarrow{i, A^i(P_B)\ j \text{ times with decreasing } i}$ S

. . .

Clear − Request: S $\xrightarrow{\mathrm{SIG}_A, V', A^{n-j}(P_B)}$ A

Clear − Reply: S $\xleftarrow{\mathrm{Y/N},\ \overbrace{\mathcal{S}_A(\mathrm{Y/N}, \mathrm{SIG}_A, V', A^{n-j}(P_B))}^{\mathrm{SIG}'_A}}$ A

Figure 2: μ-3KP Protocol

4.3 Clearing of Accumulated Micro-Payments With μ-iKP

When the seller receives the last coupon, $A^j(P_B)$, the seller assembles the CLEAR-REQUEST by adding $A^j(P_B)$ to CLEAR – which is worth $n - j$ times the amount of each coupon. The acquirer can verify this pre-image without any further information, and he will store it like the rest of a regular iKP message.

There remain two issues of this CLEAR-REQUEST:

1. The seller must somehow determine when the *last coupon* will be reached. He cannot expect the buyer always to deposit all the pre-images of a chain. Waiting until coupon P_B is reached may cause significant interest or exchange rate losses. If the seller on the other hand can clear intermediary coupons he will suffer from multiple clearing charges from the acquirer. It's up to the sellers discretion to find the optimal strategy to solve this (economic) problem.

2. In the case that $A^i(P_B)$, where $i \neq 0$, can be cleared, fraudulent buyers could replace $A^i(P_B)$ with a *higher-numbered* $A^i(P_B)$ than the one released last. There are two approaches to counter this problem:

   (a) The seller signs CLEAR-REQUEST message. This prevents fraudulent buyers from interfering but adds the cost of an additional expensive cryptographic operation.

   (b) If a wrong $A^j(P_B), j > i$, is cleared, the seller will find out in the CLEAR-RESPONSE and is always able to clear the correct $A^i(P_B)$ later. This increases the seller's transaction cost with the conventional financial networks, but it essentially amounts merely to a denial of service attack . Unless systematic attacks must be expected, it may well be preferable to the seller to omit the expensive signature change and risk the rare occurrence of this quasi-denial of service.

4.4 Protection of Micro-Product Requests and Delivery

The DESC of the authorizing μ-iKP exchange specifies for example a document subtree in a server, but not the exact document to be consumed later. We call such an exact document specifier (e.g a URL [14] for the World Wide Web) a *micro-DESC*. The delivery policy is likely to consist of an obligation of the seller to retransmit[4] a micro-product so many times until a buyer acknowledges the receipt. This appears feasible because the value of one document is small and it is unlikely that an interceptor would find enough other buyers of the same information himself in order to make such fraud profitable[5].

In such a setting, interceptors can change a micro-DESC (HTTP request/URL) coming from the legitimate buyer to the seller within the realm of the DESC. This is essentially *stealing pre-images* and depositing them. When the legitimate buyer later claims to have received a wrong product, the interceptor has already consumed the micro-product.

Sufficient protection appears to be achievable if the buyer and seller establish a session key parallel to the initiating 3KP authorization run. part of common, unlinked SSL) $A^in(P_B)$ and micro-DESC can now be bound together by computing a MAC over both or by encrypting both for privacy reasons. The only remaining source of trouble can be dishonest sellers because

[4] Naturally, for non-reentrant micro-products like current exchange rate, the obligation to give the actual rate at a later point in time as opposed to just replaying the historic rate would be released.

[5] Operating in a micro-payment environment does not prevent sellers from cryptographically strongly marking their products nor does it prevent buyers from employing vending schemes to avoid untrustworthy sellers [15].

the micro-DESC in the requests of the buyers are still not disputable. Practically, however, the losses are small and systematically fraudulent sellers would most likely go out of business due to non-technical reasons such as a bad reputation.

5 Non-Repeated Micro-Payments Through Brokerage Trusted Third Parties

So far we assumed that there is a long-lived enough relationship between buyer and seller to justify the establishment of macro-payment context. Although this is a reasonable assumption in many circumstances there might still be cases there we have to relax this assumption.
The underlying assumption of the approach with a broker is, that by introducing the broker as a trusted third party (TTP), the following holds: The sum of the users of a broker buy form a particular seller so intensively, that this constitutes a *virtual repeated micro-payment* between the broker and the seller. Furthermore the same is assumed for the buyer-broker relation: The buyer is about to non-repeatedly purchase from so many sellers through the same broker that this constitutes a virtual buyer-broker repeated micro-payment.

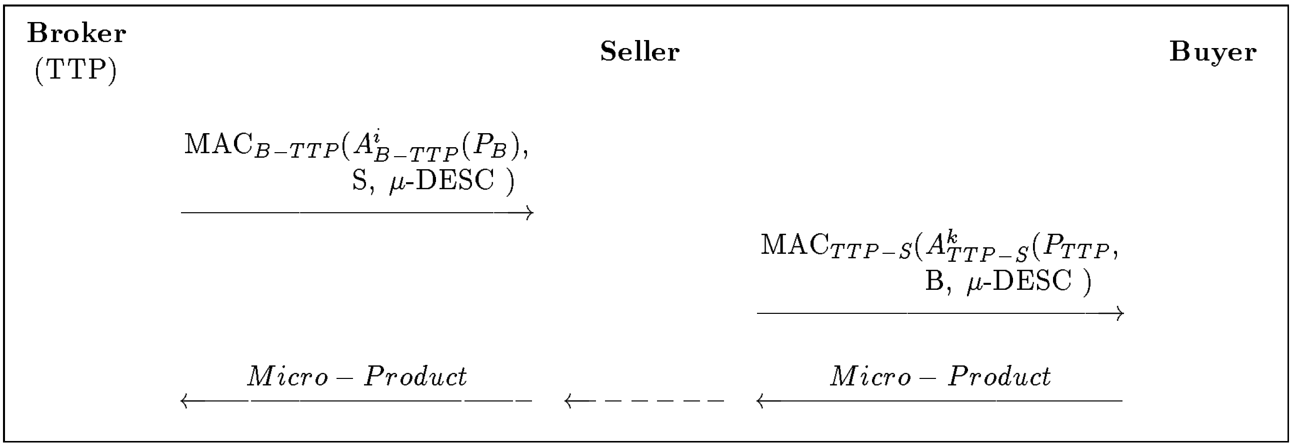
Broker (TTP)
Seller
Buyer
$\text{MAC}_{B-TTP}(A^i_{B-TTP}(P_B),$ S, μ-DESC)
$\text{MAC}_{TTP-S}(A^k_{TTP-S}(P_{TTP},$ B, μ-DESC)
$Micro - Product$
$Micro - Product$

Figure 3: Coupon Translation through Broker TTP

The entire system works as follows (see Figure 3):

1. The buyer establishes a micro-payment relation and a shared session key with the TTP.
2. Whenever the buyer wants to purchase something, he sends the micro-DESC and his $A^i_{B-TTP}(P_B)$ to the TTP protected by the pertinent session key.
3. The TTP then translates $A^i_{B-TTP}(P_B)$ into $A^k_{TTP-S}(P_{TTP})$, adds micro-DESC and the permitted depositor and protects this request with the previously established, shared session key TTP-seller. This token is either sent directly to the seller or returned to the buyer who then transparently forwards it to obtain the desired micro-product.

Evaluation
This approach to employ brokerage TTPs to avoid the problems of non-repetitive buyer behavior provides no security gains. Its main achievement is to simplify the monetary relations

and to avoid situations where the buyer obtains change or wants to redeem coupons as in Millicent [5]. From an efficiency point of view, the advantage of reduced computational complexity at the seller's site (=potential bottleneck) is even increased as a broker-seller relationship is longer-lived and likely involves more transactions than a buyer-seller relationship and therefore more micro-payments per macro-payment can be done.

6 Further and Open Issues

Bottleneck Broker
As the broker is directly involved in each transaction it might easily become a bottleneck. By letting the buyer pre-fetch coupons we could decouple the interaction broker-buyer and buyer-seller. In that case a seller cannot expect the pre-images distributed by the TTP to various buyers to arrive in sequence. The seller therefore must relax the requirement of a strictly consecutive arrival order and maintain a list of potentially later arriving pre-images. Owing to the low financial amounts at stake, it is probably permissible not to add an explicit expiration to the pre-images to inform all parties involved of the urgency to deposit them, but to have informal rules work for the normal case. This problem is exacerbated if it is not the rule that every pre-image of the sequence must be deposited, but that multiple pre-images can be spent in one transaction simply by giving the lowest numbered pre-image.

Fair Exchange
The setting with a broker acting as a mediator lends itself to the idea to use this trusted third party also for fair exchange of goods and payment. The broker would first collect the payment from the buyer and would forward it to the seller only when the seller delivers the good (see [16] on how one might implement it).

Key Management
In the current version of this report, the key management to obtain keys to protect the integrity of the {micro-DESC, coupon} pair and the delivery are considered orthogonal to the technical problem described . This task is delegated, for example, to SSL [17] or SHTTP [18]. If a future *i*KP coupon-based micro-payment system experiences wide-spread use, significant efficiency gains might be realizable if the pertinent key management is integrated into the protocols.

7 Conclusion

This report has shown that *i*KP is well amenable to support micro-payments with coupons with retaining full non-repudiation of payments at low cryptographic costs (one hash per verification of a micropayment) and minimal communication overhead (all micropayment need only one flow and do not require the acquirer to be online). If the buyer's consumption pattern shows locality, minimal changes to *i*KP are sufficient. If the buyer is surfing cyberspace broadly, the complexity-reducing aid of a brokering trusted third party becomes necessary. But even with this third party, the buyer can limit his exposure towards both the broker and the seller to the equivalence of the negligible value of one coupon. It has also been shown how the entire coupon-spending and micro-product delivery can be protected against attacks by message interceptors on the network.

References

[1] Tim Berners-Lee, R. T. Fielding, and H. Frystyk Nielsen. *Hypertext Transfer Protocol*, March 1995. Internet Draft, Expires September 8, 1995.

[2] Mihir Bellare, Juan Garay, Ralf Hauser, Amir Herzberg, Hugo Krawczyk, Michael Steiner, Gene Tsudik, and Michael Waidner. iKP – a family of secure electronic payment protocols. In *First USENIX Workshop on Electronic Commerce* [19].

[3] Mihir Bellare, Juan Garay, Ralf Hauser, Amir Herzberg, Hugo Krawczyk, Marc Linehan, Michael Steiner, Gene Tsudik, and Michael Waidner. Internet keyed payments protocol (iKP). Technical report, IBM Research, 1995. INTERNET-DRAFT draft-tsudik-ikp-00.txt.

[4] Benjamin Cox, J. D. Tygar, and Marvin Sirbu. NetBill security and transaction protocol. In *First USENIX Workshop on Electronic Commerce* [19].

[5] Marc Manassee. Design considerations for lightweight payment protocols. In *First USENIX Workshop on Electronic Commerce* [19].

[6] Jean-Paul Boly, Antoon Bosselaers, R. Cramer, R. Michelsen, S. Mjølsnes, F. Muller, T. Pedersen, B. Pfitzmann, P. de Rooij, B. Schoenmakers, M. Schunter, L. Vallée, and M. Waidner. The ESPRIT project CAFE - high security digital payment systems. In Dieter Gollmann, editor, *Proceedings of the Third European Symposium on Research in Computer Security (ESORICS)*, number 875 in Lecture Notes in Computer Science, Brighton, UK, November 1994. Springer-Verlag, Berlin Germany.

[7] Torben P. Pedersen. Electronic payments of small amounts. Technical report, Aarhus University, Computer Science Department, August 1995.

[8] Ronald Rivest. The MD5 message-digest algorithm, April 1992.

[9] U. S. National Institute of Standards and Technology NIST, Computer Systems Laboratory. Secure Hash Standard. Federal Information Processing Standards Publication (FIPS PUB) 180, May 1993.

[10] Ralph C. Merkle. A certified digital signature. In G. Brassard, editor, *Advances in Cryptology – CRYPTO '89*, number 435 in Lecture Notes in Computer Science 435, pages 218–238, Heidelberg, August 1990. Springer-Verlag.

[11] Leslie Lamport. Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772, November 1981.

[12] Ronald L. Rivest and Adi Shamir. PayWord and MicroMint: Two simple micropayment schemes. Technical report, MIT LCS, November 1995.

[13] Ross Anderson, Harry Manifavas, and Chris Sutherland. A practical electronic cash system. personal communication, Dec 1995.

[14] T. Bernes-Lee, L. Masinter, and M. McCahill. Uniform resource locators (url). Internet Request for Comment RFC 1738, December 1994.

[15] Ralf C. Hauser. Using the Internet to decrease Software Piracy - on Anonymous Receipts, Anonymous ID Cards, and Anonymous Vouchers. In *INET'95 The 5th Annual Conference of the Internet Society The Internet: Towards Global Information Infrastructure*, volume 1, pages 199–204, Honolulu, Hawaii, USA, June 1995.

[16] Holger Bürk and Andreas Pfitzmann. Value exchange systems enabling security and unobservability. *Computers & Security*, 9(8):715–721, 1990.

[17] Kipp E.B. Hickman. The SSL protocol. RFC draft, Netscape Communications Corp., November 1994. Version 1.0.

[18] E. Rescorla and A. Schiffman. The Secure HyperText Transfer Protocol. Internet Draft, July 1995. version 1.1, Expires 1/96.

[19] USENIX. *First USENIX Workshop on Electronic Commerce*, New York, July 1995.