This work has been submitted to the IEEE for possible publication. Copyright may be transferred without notice, after which this version may no longer be accessible.

Electronic Payment Systems*

N. Asokan, Phil Janson, Michael Steiner, Michael Waidner
IBM Research Division
Zurich Research Laboratory, CH-8803 Rüschlikon, Switzerland
{aso,pj,sti,wmi}@zurich.ibm.com

ABSTRACT

As business is moving from face-to-face trading, mail order, and telephone order to electronic commerce over open networks such as the Internet, crucial security issues are being raised. Whereas Electronic Funds Transfer over financial networks is reasonably secure, securing payments over open networks connecting commercial servers and consumer workstations poses challenges of a new dimension.

This article reviews the state of the art in payment technologies, and sketches emerging developments.

Keywords

Electronic commerce, payment systems, open networks, secure transactions, security, privacy.

Introduction

Since the dawn of history, there has been trading between two parties exchanging goods face-to-face. Eventually such trading became complicated and inconvenient; money was invented so that a buyer could acquire something he needed from a seller without necessarily exchanging goods. Security of the monetary systems was guaranteed by the local, regional, national, and eventually international banks controlling the printing of money. In course of time, new ways of payment such as payment orders, cheques, and later 'plastic' money were invented. These allow payment without 'actual' money. Mapping between the payment instrument and real money is still guaranteed by banks through secure financial clearing networks. Eventually, remote payment became possible using those same instruments, although security then started to become a challenge. Verifying a hand-written signature on a cheque or a credit card mail order is impossible when buyer and seller are not face-to-face; in this case the buyer must take the additional risk of sending in a payment before having received his purchase or the seller must send the purchased items before having received the payment. Phone order purchases are riskier since no signature at all can be provided: the seller runs the risk that the buyer may deny having made the purchase and demand a refund even after he has received the goods. Recently, there has been a great deal of interest in facilitating commercial transactions over open computer networks, such as

* This work was partially supported by the ACTS Project AC026, *SEMPER*; however, it represents the view of the authors. *SEMPER* is part of the *Advanced Communication Technologies and Services* (ACTS) research program established by the *European Commission Directorate General XIII*. An earlier version of this report (with a more extensive bibliography) can be found at *http://www.semper.org/info.*

the Internet. The introduction of open networks renders the security issues even more critical.

Clearly then, without new security measures, envisioning electronic commerce over open computer networks is utopic. In this article, we attempt to provide an overview of electronic payment systems focusing on issues related to their security.

Electronic Payment Models

Commerce always involves a **payer** and a **payee** (or **buyer** and **seller**) who exchange money for goods or services, and at least one **financial institution** which links "bits" to "money." In most existing payment systems, the latter role is divided into two parts: an **issuer** (used by the payer) and an **acquirer** (used by the payee).

Electronic payment from payer to payee is implemented by a flow of real money from the payer via the issuer and acquirer to the payee. In **prepaid cash-like** payment systems, a certain amount of money is taken away from the payer (for example, by debiting that amount from the payer's bank account) before a purchase is made. This amount of money can be used for payments later. Card-based electronic purses, electronic cash as well as (certified/guaranteed) bank cheques fall in this category. The typical flows are shown in Figure 1.

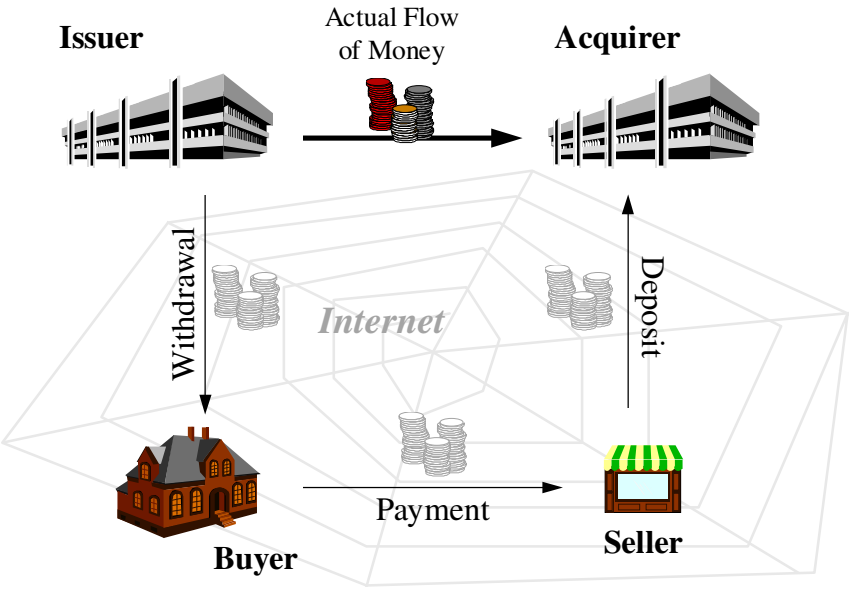

Figure 1 Cash-like Payment System

In **pay-now** payment systems, the payer's account is debited at the time of payment. ATM card based systems (such as the European *EC-Direct* system, *edc*) fall into this category. In **pay-later (credit)** payment systems, the payee's bank account is credited the amount of sale before the payer's account is debited. Credit card systems fall into this category. From a protocol point of view, pay-now and pay-later systems belong to

the same class. Typical flows of these systems are shown in Figure 2. As a payment is always done by sending some sort of "form" from payer to payee (cheque, credit card slip, etc.) we call these systems **cheque-like.**

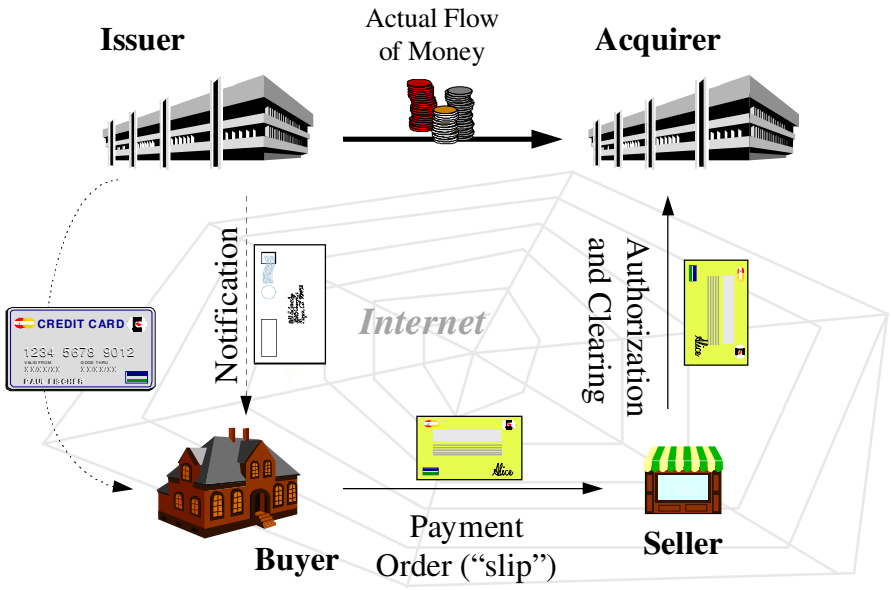

Figure 2 Cheque-like Payment System

Both types of payment systems are *direct* payment systems, i.e., a payment requires an interaction between payer and payee. There are also *indirect* payment systems where either the payer or the payee initiates payment without the other party (payee or payer, respectively) involved on-line. In the context of Internet payments this is usually considered part of "home banking."

Security Requirements

Except for the high-level requirements that nobody wants to lose money, and that everybody wants to be unobservable in their business to some extent, the concrete security requirements of electronic payment systems vary depending on their features and the trust assumptions put on their operation. Generally, though, one or more of the following requirements must be met.

Integrity and Authorisation

For a payment system, integrity means that no money is taken from a party unless a payment is explicitly authorised by that party. Moreover users might require not to receive any money without their explicit consent; this is desirable when a user wants to avoid unsolicited bribery.

Technically, authorisation is often achieved by *authenticating* those messages that cause the transfer of money (withdrawal/payment/deposit in cash-like systems, payment/clearing in cheque-like systems). There are several ways to do this:

- **Out-band authorisation:** The verifying party (typically a bank) sends a notification to the authorising party (the payer) and requires that the authorising party approve or deny the authorisation off-line, using a secure out-band channel (e.g., ordinary mail or phone calls).

  This is the current approach for credit card payments of mail orders and telephone-orders (MOTO): anyone who knows a user's credit card data can initiate such payments in the name of that user. Users must check their credit card statements and must actively complain about false transactions. The absence of a reaction within a certain time (usually 90 days) is interpreted as "approval" by default.

- **Authorisation by passphrase:** The verifying party requires that every message from the authorising party include a cryptographic check value computed using a secret known only to the authorising and verifying parties. This secret can be a PIN or a passphrase, or in general any form of "shared secret" (see sidebar on basic concepts in security and cryptography).

  This achieves security against outsiders, but does not provide **non-repudiation of origin**: a dispute between the authorising and verifying parties about the origin of a well-authenticated message cannot be resolved. A court would not be able to decide from the messages exchanged whether a disputed payment was authenticated by the payer or by dishonest employees of the payer's bank.

  Short shared secrets like 4- or 6-digit PINs are inherently susceptible to various kinds of attacks (see sidebar). They cannot by themselves provide a high degree of security. They should only be used to control access to a physical token like a smartcard (or wallet) that performs the actual authorisation based on secure cryptographic mechanisms.

- **Authorisation by signature:** The verifying party requires a digital signature of the authorising party.

  Digital signatures provide non-repudiation of origin: only the owner of the secret signature key can sign messages (whereas everybody who knows the corresponding public verification key can test signatures). Resolution of disputes when the authorising and verifying parties disagree about the authenticity of a message is more straightforward when digital signatures are used.

Authorisation need not always be based on authentication. In systems with high levels of anonymity and untraceability requirements (explained later in greater detail), authentication may in fact contradict requirements. Authorisation can also be inherent in the payment message if the structure of the message is such that only an authorised person could have generated it and it can be used only once. It is also necessary to ensure the **freshness** (see sidebar) of authenticated messages.

Later, we list some examples of payment systems structured according to the technique used for authorising the payer's transfer order to the bank. This authorisation constitutes the most important relationship in a payment system.

In addition to these requirements, payer and payee might want a certain degree of fairness in the exchange of payment for receipt, or more directly in the exchange of money for goods and services. The most practical way to implement fair exchanges is by

means of a trusted third party. However, fairness of exchange is usually beyond the scope of payment systems.

Confidentiality

Some or all parties involved may wish confidentiality of a transaction. Confidentiality in this context means the restriction of the knowledge about various pieces of information related to a transaction: the identity of payer, payee, purchase content, amount, etc. Typically, the confidentiality requirement dictates that this information be restricted only to the participants involved. Where anonymity or untraceability are desired, the requirement may be to limit this knowledge to certain subsets of the participants only. The section on anonymity and untraceability contains more details on this topic.

Availability and Reliability

All parties are interested in being able to make or receive payments whenever necessary.

Payment transactions must be *atomic*: they occur entirely or not at all, but never hang in an unknown or inconsistent state. No payer would accept a loss of money (the loss of a significant amount, in any case) due to a network crash, or because the payee's server crashed.

Availability and reliability presume that the underlying networking services and all software and hardware components are sufficiently dependable. Recovery from crash failures requires some sort of stable storage at all parties and specific re-synchronisation protocols. These fault tolerance issues are not discussed in the following, because most payment systems do not address them explicitly.

Requirements for Mobile Trusted Hardware

The use of cryptographic techniques to meet security requirements makes it desirable that all parties involved have access to secure key storage. If payments are to be possible from any workstation, the secret key storage of a user must even be portable. This in turn makes it necessary that users be provided with smart cards or similar secure devices. These technology options are discussed in the following section together with some examples.

Technology Overview

	On-line Payment Systems	Off-line Payment Systems
Systems enabling control of payment flow by payment system provider	Credit-card payment systems: • Proposal using no cryptography: First Virtual • Proposals using Cryptography: CyberCash, iKP. • Proposed standard: SET. Micropayments: • Millicent, NetBill, Phone-Ticks, μ-*iKP*, PayWord. Payment switches: • Globe ID(R) by GC Tech. • OpenMarket payment switch. Electronic cheques • FSTC Electronic Check Project	Electronic purses, using smart cards: • Shared key, e.g., Danmont (Denmark), Proton (Belgium) • Public key, e.g., CLIP (Europe-wide) • Not known publicly: Mondex (UK) • Standardisation: CEN Intersector Electronic Purse, EMV Electronic Purse.
Untraceable Systems	• Anonymous "remailers" for change, e.g., NetCash, Anonymous Credit-cards. • Anonymous ("blind") signatures, e-cash.	• Anonymous ("blind") signatures, e.g., CAFE (research project, sponsored by the European Union).

Figure 3 Proposed Technologies for *Internet* Payments

The main design decision for electronic payment systems is how to authorise payments: how to enable the honest payer to convince the payee to accept a legitimate payment while preventing the dishonest payer from making unauthorised payments. For instance, it must not be possible to spend the same money twice by sending the same messages to two different payees. All this must be done in a way that does not violate the privacy of honest payers and payees.

On-line vs. Off-line

Payments can be performed **on-line,** involving an authentication and authorisation server (usually as part of the issuer or acquirer) in each payment, or **off-line,** i.e., without contacting a third party during payment.

The obvious problem with off-line payments is how to prevent payers from spending more money than they actually possess. In a purely digital world, a dishonest payer can easily reset the local state of his system after each payment to the state before the pay-

ment. Therefore off-line payment systems that *prevent* (not merely detect after the fact) double-spending require tamper-resistant hardware, such as **smartcards**, at the payer end. Often, tamper-resistant hardware, such as security modules of point-of-sale (POS) terminals, is also used at the payee end — it is mandatory in the case of shared-key systems, and in cases where the payee does not forward individual transactions but only their totals.

On-line systems obviously require more communication, but not necessarily tamper-resistant hardware. In general, they are considered more secure than off-line systems. Most proposed Internet payment systems are on-line systems.

All proposed payment systems based on electronic hardware are off-line systems. Examples are *Mondex*[1], *CLIP* (by Europay), and *CAFE*[2] (developed by a European ESPRIT project). *Mondex* is the only system that enables off-line transferability: the payee can use the amount received for a new payment, without intermediate deposit — but this seems to be a politically unpopular feature in most countries. It is also the only system considered for high-value payments (e.g., over 100 ECU). *CAFE* is the only system that provides strong payer anonymity and untraceability. Both systems offer payers an electronic wallet, preventing the well-known *fake-terminal attacks* on the payer's PIN. Additionally *CAFE* provides **loss tolerance**, which allows the payer to recover from coin losses (but at the expense of some anonymity). *Mondex, CLIP,* and *CAFE* are **multi currency** purses capable of handling different currencies simultaneously.

All these systems can be used for Internet payments, although none is being used at the time of writing. Plans for deployment on the Internet exist for several systems including *Mondex* and *CLIP*. The main technical obstacle is that they require a smartcard reader attached to the payer's PC or workstation. Inexpensive PCMCIA smartcard readers and standardised infrared interfaces on notebook computers will solve this connectivity problem.

Another system being developed in this spirit is the *FSTC Electronic Check Project* (URL: *http://www.fstc.org/projects/echeck/index.html*), which uses a tamper-resistant PCMCIA card, and implements a cheque-like payment model. The related *FSTC Electronic Commerce Project* deals with unifying the different payment methods at the acquirer's site.

Instead of tamper-resistant hardware, off-line authorisation could be given via preauthorisation. The payee is known to the payer in advance, and the payment is already authorised during withdrawal, in a way similar to a certified bank cheque.

Trusted Hardware

As we saw, most off-line payment systems require a piece of tamper-resistant hardware at the payer's end in order to meet the security requirements of the issuer. In a certain sense, this hardware is a "pocket branch" of a bank and must be trusted by the issuer.

Independent of the issuer's security considerations, it is in the payer's interest to have a secure device that can be trusted to protect his secret keys and to perform the necessary operations. Initially, this could be simply a smartcard. But in the long run, it could become a smart device of a different form factor with secure access to a minimal keyboard and display. This is often called an **electronic wallet**.

Without such a secure device, the payer's secrets, and hence his money, are vulnerable to anybody who can access his machine. This is obviously a problem in multi-user environments. It is also a problem even on single-user machines that may be accessed directly or indirectly by others — for instance a virus surreptitiously installed on such a machine could steal PINs and passwords as they are entered.

Even with a smartcard, the payer cannot be sure that a Trojan horse in his PC cannot reveal the PINs of credit cards by sending mail to a remote attacker, or by simply asking the smartcard to make a payment silently to the attacker's account. Therefore, for true security, trusted input and output channels between user and smartcard must exist.

Cryptography

"Crypto-less" Systems

Using no cryptography at all means relying on "out-band" security: for instance, the goods ordered electronically are not delivered before a fax (or a letter, or a phone call) arrives from the payer confirming the order.

Examples of this kind of system are *First Virtual* and the *Internet Shopping Network*. In both, a user has an account and receives a password in exchange for a credit card number. The password is not protected while travelling over the Internet. Hence, these systems are vulnerable to eavesdropping. *First Virtual* achieves some protection by asking the supposed payer for an acknowledgement of each payment via email, but the actual security of the system is based on the payer's ability to revoke each payment within a certain period. In other words, there is no definite authorisation during payment. Until the end of this period, the payee assumes the entire risk.

Generic "Payment Switch"

A special role is played by the *OpenMarket Payment Switch*[3]: it is an on-line payment system implementing both the pre-paid and pay-later models. Its architecture supports several authentication methods, depending on the payment method chosen, ranging from simple, unprotected PIN-based authentication to challenge-response based systems, where the response is computed, typically by a smartcard.

Actually, *OpenMarket* uses passwords and optionally two types of devices for response generation: *Secure Net Key* and *SecureID*. User authentication therefore is based on shared-key cryptography. However, authorisation is based on public-key cryptography: the *Payment Switch* digitally signs an authorisation message, which is forwarded to the payee.

The *Globe ID(R)* system by *GC Tech* (URL: *http://www.gctec.com/us/Technical*) is very similar to the *OpenMarket* approach. In both solutions, the payment switch is completely trusted by users who use shared-key cryptography (see next section).

Shared-key Cryptography

For authentication based on shared-key cryptography, the prover (e.g., payer) and the verifier (e.g., issuer) need a shared secret like a DES key, or at least a password/PIN.

As both sides have exactly the same secret information, it is not possible to provide non-repudiation. For example, if payer and issuer disagree about a certain payment, there is no way to decide whether this payment was initiated by the payer, or by a dishonest employee of the issuer. Therefore authentication of the payer's transfer order based on shared-keys is not appropriate if the payer bears the risk of forged payments, at least not for high-value payments. Anderson[4] discusses some examples of the consequences of the lack of non-repudiation.

If authentication is to be done off-line (which is desirable at least for low-value payments), each payer-payee pair needs a shared secret key. In practice this means that some sort of master-key is present at each payee end, enabling the payee to derive the payer's key, for example, from the payer's identity. Tamper-resistant security modules in point-of-sale terminals are used to protect this master key.

Off-line systems *Danmont* and *Proton,* and on-line systems *SNPP*, *NetBill*[5], and NetCheque[6], are examples of systems using shared-key cryptography. The *2KP* (a variant of *iKP*[7]) may use a shared secret between payer and issuer/acquirer for authentication. The design of *Mondex* is not public. According to presentations by *Mondex* they are using public key technology, but it is not known where and for what purpose. The actual hardware they are using suggests that the system is primarily a shared-key system.

Public-key Digital Signatures

For authentication based on public-key cryptography, the prover has a secret signature key and a certificate for its corresponding public signature verification key , issued by a well-known authority (in the context of payments, this authority can be the payment system operator or a specific issuer). In most existing and proposed systems, RSA is the underlying public-key technology; but there are several alternatives as well.

Digital signatures can provide non-repudiation so that disputes between sender and recipient of a signed message can be resolved. This should be mandatory if the payer bears the risk of forged payments, at least for high-value payments.

Off-line authentication is no problem here because the payee can easily verify a signature of the payer (and could check the certificate against a local copy of a blacklist of "bad" certificates, if necessary). Authorisation still requires either an on-line connection or trusted hardware at the payer end.

Rather general WWW security schemes using public-key cryptography are *SHTTP* and *SSL*. Neither is a payment technology per se, but they have been proposed for securing payment information as well.

SSL (URL:*http://www.netscape.com/newsref/std/SSL.html*) is an Internet socket-layer communication interface allowing two parties to communicate securely. An extension of *SSL*, called *PCT* (URL:*http://pct.microsoft.com/*) has also been described. Neither *SSL* nor *PCT* support non-repudiation.

SHTTP (URL:*http://www.eit.com/creations/s-http/*) is a secure extension of the *HTTP* protocol used on the Internet World-Wide Web (WWW). It was developed by the *CommerceNet* consortium. *SHTTP* offers several security techniques such as signing and encrypting with RSA. Various payment protocols can be implemented on top of these techniques.

Complete payment systems using public-key cryptography include *ecash*[8], *NetCash*[6], *CyberCash* (URL:*http://www.cybercash.com)*, the *3KP* variants of *iKP*[7], and *SET* (URL:*http://www.mastercard.com/set/set.htm*). These systems are (or aim for) real implementations. The protocol ideas themselves are much older. The use of digital signatures for both on-line and off-line payments, anonymous accounts with digitally signed transfer orders, and anonymous electronic cash were all introduced during the 1980s. Bürk and Pfitzmann[9] have included an early survey of these issues in their paper.

Micropayments

Micropayments are low-value (e.g., less than 1 ECU, less than $1) payments to be made very quickly. Payment of a small amount for each time "tick" using a pre-paid telephone card is an example of a micropayment scenario. Given the small amounts involved and the speed required, micropayment techniques must be both inexpensive and fast. To achieve this one has to make certain assumptions and compromises.

Based on the assumption of repeated payments (e.g., pay-per-view) there have been a number of proposals, beginning with *CAFE* phone-ticks[10], that use **one-way hash functions** to implement micropayments (see sidebar for a more detailed description of one such proposal: μ*iKP*).

On the other hand, assuming the low risk due to the small amounts, one can also reduce the degree of security (e.g., by foregoing non-repudiation). Notable examples are *NetBill* and *NetCheque*, which are founded on the shared-key based *Kerberos* technology. Both implement a cheque-like debit payment model. The use of shared-key technology is justified by the performance required to process many micro-payments in a short time. It has been announced that both will migrate to public key technology.

Anonymity of Payer

Some payment systems provide payer anonymity and untraceability. Both are considered useful for cash-like payments (say, up to $100), with the argument that cash is also anonymous and untraceable, and that payers prefer to keep their everyday payment activities private. Typically they certainly do not want unrelated third parties to observe and track their payments. Often, they prefer the payees (shops, publishers, etc.) and in some cases even banks to be incapable of observing and tracking their payments.

Whereas **anonymity** simply means that the payer's identity is not used in payments, **untraceability** means that the payer cannot be identified, and even that two different payments by the same payer cannot be linked. Both properties hold with respect to the payee only or with respect to both payee and issuer/acquirer.

All payment systems could be made untraceable by outsiders by encrypting all flows between payer and payee, and anonymous to the payee by using pseudonyms instead of real identities. Some electronic payment systems are designed to provides anonymity or even untraceability with respect to the payee (e.g., *iKP* offers this as an option).

Currently, the only payment systems mentioned here that provide anonymity and untraceability against payee and issuer/acquirer are *ecash*[8](on-line) and *CAFE*[2] (off-line,

based on smartcards). Both are based on public-key cryptography (a special form of signatures called blind signatures[8, 11]).

NetCash[6] and *ACC*[12] also provide anonymity and untraceability. But they are based on the use of trusted "mixes" that change electronic money of one representation into another representation, without revealing the relation. Neither *ecash* nor *CAFE* assume the existence of such trusted third parties.

Standardisation

The European Standardisation Organisation, *CEN*, as well as *Europay, MasterCard, VISA* (for this purpose known as *EMV*) are working on standards for smartcard-based electronic payment systems. Although a *CEN* standard for an Intersector Electronic Purse already exists, current *EMV* specifications do not include electronic purse functions. No standardisation efforts towards an untraceable off-line payment system exist.

Two initially competing proposals, *STT* and *SEPP*, for standards for credit-card payment schemes were published by *VISA* and *Mastercard*, respectively, and recently replaced by a common proposal, called *SET* — Secure Electronic Transactions.

In recent months, several proposals for credit-card payment systems were submitted to the *Internet Engineering Task Force* (*IETF*), namely *FirstVirtual, CyberCash,* and *iKP*, and a working group on electronic payment systems was established in December, 1995. As a result of joint statements by *MasterCard* and *VISA*, the IETF working group now works only on negotiation and encoding (transport) aspects, rather than on the protocols themselves. A first step in this direction was the *Universal Payment Preamble* proposal from *CyberCash* (available as an Internet draft). *CommerceNet* and the *W3 Consortium* started a joint project (JEPI) supporting this IETF work (URL:*http://www.w3.org/pub/Payments/Payments/041796.html)*.

Currently, discussions on *SET* dominate the stage of *Internet* payment systems, but there is a parallel demand for international standards of electronic cash-like payment schemes and schemes for micropayments.

Summary

This high-level overview is intended to make clear that the technology necessary for secure electronic Internet payment systems already exists. Achieving security for all parties, including perfect untraceability of the payer, is possible.

The question "Which payment system will be used on the Internet?" will not have a single answer. Several payment systems will coexist:

- Micropayments (less than 1 ECU), low-value payments (1-100 ECU) and high-value payments have significantly different security and cost requirements.

  Possibly, high values will be transferred using non-anonymous, on-line payment systems based on public-key cryptography implementing a cheque-like or credit-card-like (e.g., *SET*) payment model. As soon as smartcard readers are available at PCs and workstations, small amounts might be paid using pre-paid off-line payment systems that provide a certain degree of untraceability (similar to cash).

- Payment systems with and without tamper-resistant hardware at the payer's end will coexist for some time. Ultimately, payment systems based on smartcards and electronic wallets (having secure access to some display and keyboard, and communicating with the buyer's terminal via an infrared interface) will become prevalent, because they clearly provide better security and enable the payer to use untrusted terminals without endangering security.
- A few almost equivalent payment systems will possibly coexist for the same areas of application (i.e., payment model and maximum amounts). The reasons are various "cultural" differences in the business and payment processes (e.g., between the U.S. and Europe), national security considerations that might disable some solutions in some countries, and competition between payment system providers.

A more formal model of secure electronic payment systems is needed to analyse the requirements on them and to compare specific payment systems. This is a focus of our current work.

References

For a collection of WWW pointers on electronic commerce, see <http://www.zurich.ibm.com/Technology/Security/sirene/outsideworld/ecommerce.html>.

1. *R. Rolfe:* **Here Comes Electronic Cash;** Credit Card Management Europe, January/February 1994, 16-20.
2. *J.-P. Boly et al.:* **The ESPRIT Project CAFE - High Security Digital Payment Systems;** ESORICS '94, LNCS 875, Springer-Verlag, Berlin 1994, 217-230. <http://www.zurich.ibm.com/Technology/Security/sirene/publ/BBCM1_94CafeEsorics.ps.gz>
3. *D. K. Gifford, L. C. Stewart, A. C. Payne, G. W. Treese:* **Payment Switches for Open Networks;** IEEE COMPCON, March 95.
4. *R. Anderson:* **Why Cryptosystems Fail;** Communications of the ACM 37/11 (1994) 32-41. <ftp://ftp.cl.cam.ac.uk/users/rja14/wcf.ps.Z>
5. *M. Sirbu, J. D. Tygar:* **NetBill: An Internet Commerce System;** IEEE COMPCON, March 95.
6. *C. Neuman:* **Security, Payment, and Privacy for Network Commerce;** IEEE Journal on Selected Areas of Communications 13/8 (1995) 1523-1531.
7. *M. Bellare et al:* **iKP — A Family of Secure Electronic Payment Protocols;** Usenix 1995. <http://www.zurich.ibm.ch/Technology/Security/publications/1995/ikp.ps.gz>
8. *D. Chaum:* **Privacy Protected Payments — Unconditional Payer and/or Payee Untraceability;** SMART CARD 2000, North-Holland, Amsterdam 1989, 69-93.
9. *H. Bürk, A. Pfitzmann:* **Digital Payment Systems Enabling Security and Unobservability;** Computers & Security 8/5 (1989) 399-416. <http://www.zurich.ibm.com/Technology/Security/sirene/publ/BuePf_89.ps.gz>
10. *T. Pedersen:* **Electronic payments of small amounts;** Technical Report, Aarhus University, Computer Science Department, August 1995.
11. *S. Brands:* **Untraceable Off-line Cash in Wallet with Observers;** Crypto '93, LNCS 773, Springer-Verlag, Berlin 1994,302-318.
12. *S. H. Low, N. F. Maxemchuk, S. Paul:* **Anonymous Credit Cards;** 2nd ACM Conference on Computer and Communication Security, Fairfax 1994.

Sidebar: Basic Concepts in Cryptography and Security

- **Message Authentication:** To authenticate a message is to prove the identity of its originator to its recipient. Message authentication can be achieved using shared-key or public-key cryptography:
  - **Shared-key:** In shared-key cryptography, the prover and the verifier share a common secret. Hence this is also called *symmetric authentication.* A message is authenticated by means of a cryptographic check value, which is a function of both the message itself and the shared secret. This check value is known as the **message authentication code** or **MAC**.
  - **Public-key:** In public-key cryptography, each entity has a related pair of keys. One, known as the **signature key**, is used for computing signatures and is kept secret. The other, known as the **verification key**, is used to verify signatures made with the corresponding signature key; the verification key is made public along with a **certificate** binding an entity's identity to its verification key. Certificates are signed by a well-known authority whose verification key is known *a priori* to a verifier. A message is authenticated by computing a **digital signature** over the message using the prover's private key. Given a digital signature and a certificate for its verification key, a verifier can authenticate the message. Authentication of messages using MACs does not provide **non-repudiation** for the message, whereas authentication using digital signatures does.
- **Attacks:** Electronic payment protocols can be attacked at two levels: the protocol itself or the underlying cryptographic system can be attacked.

  Protocol
  - **Freshness and Replay:** It is often necessary to guarantee the freshness of messages in a protocol. Freshness means that the message provably belongs to the current context only (e.g., the current payment transaction), i.e., that it is not a **replay** of a previous message. A **nonce** is a random value chosen by the verifying party and sent to the authenticating party. As nonces are unpredictable and used only in one context (e.g., one payment transaction), they ensure that a message cannot be reused in later transactions. As nonces do not require synchronisation between the two parties, they are very robust and popular in cryptographic protocol design. In general, nonces are an example of a **challenge-response** technique.
  - **Fake-terminal:** Protocols that perform authentication in only one direction are susceptible to a "fake-terminal" attack. For example, when a customer uses a bank ATM machine, the bank and the machine check the authenticity of the customer by means of the PIN code. The customer, however, cannot be sure whether the ATM is a genuine bank terminal or a fake one installed by an attacker for gathering PIN codes. Using a trusted personal device, such as a smart card or electronic wallet, helps avoid this attack.

  Cryptosystem
  - **Brute force attack:** The space from which cryptographic keys are chosen is necessarily finite. If this space is not large enough, a brute force attack of trying every possible key becomes practical. Four-digit PIN codes

have a total of 10,000 permutations in the key space. If a value *X* is known that is a deterministic function of the PIN, one can use this *X* to search the set of all possible PINs for the correct one. In some applications one can increase the protection against brute force attacks by randomization. Even if the key space is large, the probability distribution of keys is not necessarily uniform (especially for user-chosen PINs, which are likely to be related to the user's birthday, phone number, etc.) and it might be possible to mount **dictionary attacks**.

- **Cryptanalysis:** Another type of attack can explore weaknesses in the cryptosystem itself. Most cryptosystems are not proven secure but rely on heuristics, experience, and careful review and are prone to errors. Even provably secure cryptosystems are based on the intractability of a given mathematical problem (e.g., the difficulty of finding graph isomorphism), which might be solvable one day.

FURTHER READING:

B. Schneier: **Applied Cryptography;** John Wiley & Sons, 1994, 1996.

Sidebar: Proposed Electronic Payment Systems

- **Secure Electronic Transactions (SET)**
  SET is a pragmatic approach to pave the way for easy and rapid enabling of secure electronic transactions over the Internet preserving the existing relationships between merchants and acquirers as well as between consumers and their bank. SET concentrates on securely communicating credit card numbers between a consumer and an acquirer gateway interfacing to the existing financial infrastructure. In our classification, SET falls under the "cheque-like" model. In a first handshake the merchant authenticates itself to the consumer and all offer and payment data are fixed. The consumer then generates a payment slip using a sophisticated encryption scheme which protects the sensitive payment information (e.g., credit card number), limits the encryption to selected fields to ease export approval, cryptographically ties the order information, and minimises exposures of the user's privacy. This slip is then signed by the consumer to authorise the payment and is sent to the merchant, who sends it to its acquirer gateway to authorise and capture the payment. The acquirer checks all signatures and the slip, verifies over the existing network the creditability of the consumer and sends – depending on the outcome of this operation – either a positive or negative signed acknowledgement back to merchant and consumer. *SET* was designed by *GTE*, *IBM*, *Microsoft*, *Netscape*, *SAIC*, *Terisa,* and *Verisign* in addition to *Mastercard* and *Visa*. First prototypes of *SET* toolkits have been built already, and products are expected to be available before the end of 1996. *SET* is likely to be widely adopted for credit card payments over the Internet.

FURTHER READING:

SET public documentation at URL:*http://www.mastercard.com/set/set.htm*

- **DigiCash E-Cash**

DigiCash is a Dutch company founded by David Chaum. Based on Chaum's research on anonymous electronic cash, *DigiCash* has developed **e-cash**, a cash-like payment system providing high levels of anonymity and untraceability. E-cash is based on the concept of **blind signatures.** When an entity *A* wants to obtain a blind signature on a message *m* from an entity *B, A* generates a blinded message *m'* from *m* and requests *B* to sign *m'* and return the blind signature on *m,* $sign_B(m')$, to *A*.. The blinding transformation is such that:
- *B* (and no one else) can construct $sign_B(x)$ given *x* but anyone can verify it
- *A* (and no one else) can derive the signature on *m* (i.e., $sign_B(m)$) given the blind signature on it, $sign_B(m')$.

In an e-cash system, users can withdraw e-cash coins from a bank and use them to pay other users. Each e-cash coin has a serial number. To withdraw e-cash coins, a user prepares a "blank coin" having a randomly generated serial number, blinds it and sends it to the bank. If the user is authorised to withdraw the specified amount of e-cash, the bank signs the blind coin and returns it to the user. The user then unblinds it to extract the signed coin. The signed coin can now be used to pay any other e-cash user. When a payee deposits an e-cash coin, the bank records its serial number to prevent double spending. However, because the bank cannot see the serial number when it signs the coin, it cannot relate the deposited coin to the earlier withdrawal by the payer.

Even though anonymity and untraceability are the most important features of the e-cash system, it is a commercial product with several additional features such as loss-tolerance and support for receipts. It has been used in several trials in Germany, Finland, and the United States.

FURTHER READING:

DigiCash public documentation at URL: *http://www.digicash.com*

D. Chaum: **Security without Identification: Transaction Systems to Make Big Brother Obsolete;** in: Commun. ACM; 28(10) October 1985, 1030-1044.

- **Micropayments: μ-*i*KP**

Micropayments are payments of small amounts that are to be done quickly. Hence the overhead costs of current financial clearing networks are not justified in their case. Content servers in the global information infrastructure will probably have to process such a large number of these low-value transactions that it will be impractical to use computationally complex and expensive cryptographic protocols to secure them. Ideally, a micropayment scheme should be efficient in terms of communication costs as well, especially minimising interactions with third parties. The micropayment proposal for *i*KP, μ-*i*KP, was designed with these goals in mind. It is based on computationally secure **one-way functions.** Informally, a function *f()* is one-way if it is difficult to find the value *x* given the values $y = f(x)$. In this case, *x* is called the **pre-image** of *y*; conversely, *y* is the **image** of *x*. Given such a one-way function, the payer will randomly choose a seed value *X* and recursively compute:

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

The values A^0 ,..., A^{n-1}, known as *coupons,* will enable the payer to make *n* micropayments of a fixed value *v* to one payee in the following way. First, the payer forwards A^n and *v* to the payee in an authenticated manner. Authentication can be

achieved by sending these values to the payee as the payload of a normal *i*KP payment. The payee ensures, possibly via its bank, that A^n does in fact correspond to a good hash pre-image chain that can be used for subsequent micropayments. The micropayments are then carried out by revealing components of the chain A^{n-1} , A^{n-2} , ... , A^0 successively to the payee. To clear the payments, the payee presents the partial chain A^i, ..., A^j ($0 \le i < j \le n$) to its bank in return for a credit of value $v(j-i)$. The overhead of the setup phase is justified only when it is followed by several repeated micropayment transactions. However, non-repeated (or rarely repeated) micropayments are also a likely scenario in the electronic marketplace. A user surfing the World-Wide Web may chance upon a single page which costs 0.01 ECU. Neither the micropayment setup overhead nor the cost of a normal payment is justified in this case. In μ-*i*KP, this problem is solved by a **broker.** An isolated micropayment from payer *P* to payee *Q* is carried out by *P,* who makes one or more micropayments to broker *B,* and then *B* makes an equivalent micropayment to *Q*. In other words, a non-repeating financial relationship between *P* and *Q* is achieved by leveraging on existing relationships between *B* and *P* and between *B* and Q.

FURTHER READING:

R. Hauser, M. Steiner, M. Waidner: **Micro-payments based on *i*KP;** in: Proceedings of SECURICOM 96; Paris, June 1996, 73-82.

(URL: *http://www.zurich.ibm.com/Technology/Security/publications/1996/HSW96.ps.gz)*