Electronic Payment Systems

Robert B. Austenfeld, Jr.

(Received on May 20, 1998)

1. Introduction

The purpose of this paper is to review the different methods of electronic payment, particularly those for use on the World Wide Web (WWW or "Web"). The motivation for this paper is the rapid rise in electronic commerce (Austenfeld, 1998) coupled with the problem of, until recently, there not being any convenient and secure means of making payments over the Web. Even today, a great deal of effort is going into developing electronic replacements for the traditional means of payment: cash, check, and, especially recently, credit card. For this reason, much of what I write today will be different by the time you read this. However, this "snapshot" will give the reader an idea of the different forms of electronic payment and a rudimentary knowledge of how they work. Besides replacements for the traditional cash, check, and credit card methods, a new system is being developed: one to take care of micropayments; that is, payments so small that the transaction costs would normally exceed the payment itself. The need for such a system has arisen due to the vast amount of information that now is becoming available over the Internet. It is not uncommon for users of the Internet to want to download, say, a file for a nominal charge. And, given the number of Internet users out there, this can be a lucrative business for information providers once an economically feasible payment system is developed.

This paper will describe typical systems being developed for each type of payment: cash, check, credit card, and micropayments. Because it will probably be one of the most used, the front-runner system for electronic credit card payments will be discussed first. This will be followed by cash, check, and micropayments.

Before describing these systems, it is worth noting that development of a system is one thing and its actual implementation and widespread acceptance is another. That is, some of the systems described herein may never become widely used. Also I want to note, as will soon become evident, that I have relied heavily on a book by O'Mahony, Peirce, and Tewari (1997) called *Electronic Payment Systems*; one of the few books that has comprehensively looked at the subject.

One last point: although the following descriptions may, at times, sound very complicated and labor intensive, in reality most of the "work" is done by computers and associated software.

2. Credit Card Electronic Payment Systems

Although there have been several credit card-based systems developed, according to O'Mahony, Peirce, and Tewari (1997) the Secure Electronics Transactions (SET) system is expected to "supplant other schemes, and become the basis of all network transactions involving payment cards in the not too distant future" (p. 122). For this reason, this section will concentrate on the SET system. Anyone interested in some of the other credit card-based systems that have led up to SET[1] are referred to O'Mahony et al.

To solve the problem of consumers not trusting the Internet for sending credit card information, both MasterCard and Visa, the two biggest credit card companies, began developing separate security systems. MasterCard was working on the Secure Electronic Payment Protocol (SEPP) and Visa on the Secure Transaction Technology (STT). Fortunately, the companies had the good sense to begin joint development of a system and, in January, 1996, announced that system would be SET (O'Mahony

[1] Some of the credit card-based systems that have led up to SET include: First Virtual, Collect All Relevant Information (CARI), CyberCash, the *i*-Key protocol (*i*KP), and the Secure Electronic Payment Protocol (SEPP). See O'Mahony et al. for descriptions of all these systems.

et al, 1997). The system is expected to begin seeing widespread use in 1998.

The best way to describe SET is to explain the series of messages that take place as the customer, merchant, and financial network process the payment. In this regard, it is noted that SET is primarily concerned with payment for goods and services, not the preliminary steps involved in such a transaction such as showing the customer what's available and the placement of the order. Once the order is established, the first thing the customer does using SET is to send a message to begin the "initialization" of the payment. Table 1 shows this and the other messages that comprise the

Table 1 The messages associated with the SET payment process

Message Name	From	To	Purpose
Payment Initialization Request (PIReq)	Cardholder (C)	Merchant (M)	To tell the M the C is ready to pay.
Payment Initialization Response (PIRes)	M	C	To let the C know the M is an accredited retailer (can take credit card payments).
Purchase Request (PReq)	C	M	To make the actual payment as far as the C is concerned.
Authorization Request (AReq)	M	Financial Network (FN)	To verify with the issuer (the C's bank), via the acquirer (the M's bank), that the C has sufficient funds.
Authorization (ARes) Response	FN	M	To let the M know the payment will be honored by the issuer and provide the M with a redeemable "capture token."
Capture Request (CReq)	M	FN	To request redemption of capture tokens.
Capture Response (CRes)	FN	M	To redeem capture tokens by transferring funds from C accounts to the M's account.
Purchase Response (PRes)	M	C	To tell the C that either the transaction has been completed (normal) or that the C should inquire later about status.
Inquiry Request (IReq)	C	M	To inquire about the status of a previous transaction. This inquiry could be done several times for the same transaction.
Inquiry Response (IRes)	M	C	To respond to an Inquiry Request.

SET suite and the source and destination of each message.

Payment Initialization Request (PIReq). The PIReq message includes such things as the brand of the credit card (Visa, Mastercard, etc.), a "local identifier," and a list of certificates held. A certificate is a digital message constructed by a trusted third party (TTP) that lets the user know that a particular cryptographic key is really associated with another party. For example, when the cardholder receives the response from the merchant (the PIRes message), it will be encrypted in what is called the merchant's secret key. By using the merchant's readily available public key[2], the cardholder can then decrypt and read the PIRes message. However, the problem is how does the cardholder know that the public key he has is really that for the merchant; it might be a bogus key of someone posing as the merchant. This is where the certificate comes in. Since the certificate can only be "opened" and read using the public key of the TTP, a party both the cardholder and merchant trust, the cardholder knows any information in that certificate can be trusted. The certificate, in fact, contains the name of the merchant (in this case) and the merchant's public key thus assuring the cardholder of the validity of that public key.

Payment Initialization Response (PIRes). Once the merchant has received the cardholder's PIReq, he or she returns a PIRes message that contains, among

[2] Public-key encryption, also known as asymmetric encryption, solves the problem of how to get the "key" to the recipient so he or she can "open" (decrypt) the secret message. Traditional encryption techniques (symmetric encryption) involve scrambling a message with a key that must be held by the recipient to reverse the process and unscramble the message. In general the only way to safely deliver such a key is through physical means or over some super secure communications system. Therefore, providing it to more than a relatively small number of people is infeasible. The public-key encryption algorithm is such that once a message is encrypted using the secret-key part of the algorithm, only the public-key part of the algorithm can decrypt the message. This assure the recipient that the message truly came from that person. Conversely, if a person wishes to send a message only to Bob (say), they will use Bob's public key. This assures them that only Bob will be able to read this message with his secret key.

other things, a "transaction identifier" formed from the cardholder's local identifier. This will uniquely identify this transaction in all subsequent related messages. The merchant also sends any certificates the cardholder will need that are not already held by the cardholder. Since the merchant has digitally "signed" this message, the cardholder knows it came from the merchant and that it has not been tampered with en route. The digital signature works this way: the merchant (in this case), using a "hashing"[3] algorithm, creates a digest of the message and encrypts that with his/her secret key. (To encrypt the entire message would be too expensive and/or slow due to the computational-intensiveness of the public-key algorithm.) Using the same hashing algorithm, the recipient creates another digest of the message. Using the sender's public key, the recipient decrypts the digest sent with the message and compares it with the one he or she created. If they match, the message is authentic and did come from the sender indicated.

Purchase Request (PReq). Now that the cardholder and merchant have assured themselves that they are really dealing with who they think they are dealing, the cardholder is ready to send the PReq message. This message, in effect, is the cardholder actually committing to the purchase of the goods or services. This message is the most involved and contains two parts: the order information (OI) part and the payment instruction (PI) part. The OI part contains information needed only by the merchant to fill the order such as the transaction identifier and the order description. The PI part contains the actual card data and is never seen by the merchant since it is encrypted using the public key of the acquirer (the merchant's bank). A "dual signature" is created and sent with both with the OI data and the PI

[3] A hashing algorithm is one that, when applied to a long message, creates a digest of that message that can be encrypted and sent with the message. The recipient then applies the same algorithm to the message and sees if that digest (or "hash) matches the decrypted digest; if so, the message has not been tampered with and is authentic. More generally, to hash a number means to make a digest of it. One of the most popular hashing algorithms is MD5 developed by Ron Rivest, the "R" in RSA, the public-key algrorithm.

data. The purpose of a dual signature is to conclusively link two messages without revealing the contents of both; that is, the merchant sees the OI data and, using the dual signature, can be assured that the (unreadable) PI data is validly linked to his/her OI data. Similarly, the acquirer can be assured that his/her PI data is validly linked to the (unreadable) OI data. This tells both the merchant and the acquirer that both data parts (OI and PI) were signed by the cardholder at the same time.

Authorization Request (AReq). After verifying the cardholder's electronic digital signature using appropriate certificates[4)], the merchant will normally send a AReq message to the acquirer. This message is encrypted with the acquirer's public key and contains information needed to identify the order and the amount for which authorization is being requested. A hash of the order details is also included. This same hash has also been included in the PI data which the merchant has not seen.

Upon receipt of the AReq message, the acquirer does the necessary decryptions and verification of signatures. He or she also checks that the hashes of the order details by the merchant and the cardholder (the latter being part of the PI data) are in agreement. Also the amount for which the merchant is requesting authorization will be checked against the purchase amount. If there is a difference, the acquirer will see if it falls within acceptable policy limits. If everything checks out OK, the acquirer goes ahead and seeks authorization from the issuer (the cardholder's bank) through the financial network.

Authorization Response (ARes). After receiving authorization from the issuer via the financial network, the acquirer returns an ARes message digitally signed and

[4)] Since having one certificate authority (CA) issuing certificates to everyone is obviously impractical, a CA tree has been establish with a root CA at the top and subordinate CAs as the tree is traversed to each certificate holder. For this reason, several of certificates may be associated with a particular user so that the person trying to verify that user's validity will first open the root certificate using the root trusted third party's (TTP) public key. Then, using the next public key revealed continue opening certificates until coming to that of the user.

encrypted using the merchant's public key. The SET scheme allows two options for final settlement (called "capture"): (1) capture performed at the time of authorization or (2) capture performed later. If capture is performed at the time of authorization, only information to that effect is included in the ARes. Otherwise, a "capture token," which can be redeemed later by the merchant, is included with the ARes. This capture token is signed by the acquirer and encrypted with the merchant's public key so only the merchant can use it. Once the merchant receives authorization, he or she can be confident of payment and go ahead with delivery the goods and services.

Capture Request (CReq). Assuming the merchant has received capture tokens instead of having the capture take place during authorization, he or she must now use a CReq message to request redemption of the capture tokens held. Typically this would occur at the end of a business day and include tokens from many different transactions. For each token an amount and unique identifier are included which should match the information encrypted within the token itself. The CReq is signed and encrypted by the merchant and sent to the acquirer.

Capture Response (CRes). After verifying the information in the CReq message, the acquirer credits the merchant's account and deducts any transaction fees. Then a CRes is sent back to the merchant advising of a successful settlement and of the capture amount. This message is also signed/encrypted.

Purchase Response (PRes). If the merchant has not already done so, he or she now sends the PRes message to the cardholder advising that the payment is complete. This message is digitally signed by the merchant.

Inquiry Request (IReq). In those cases where the merchant sends the PRes message before authorization and/or capture, the cardholder must use the IReq message to get status on his/her purchase. This message can be sent several times as desired. This message contains the unique transaction identifier and is signed by the cardholder.

Inquiry Response (IRes). For each IReq message the merchant sends back a IRes message which will advise the cardholder of the exact status of his/her purchase by including appropriate authorization and capture codes. This message is signed by the merchant.

To begin using SET, all parties (cardholder, merchant, and acquirer) must have whatever key certificates are needed for signing and encryption purposes. To obtain these keys, a special message is sent (by the cardholder, for example) to the appropriate certificate authority (CA) requesting a registration form. The CA's response includes the registration form and necessary certificates to verify that the form and CA are authentic. The cardholder then fills out the registration form including his or her credit card number and the expiration date. Since this is sensitive information, it is returned to the CA using "extra-strong" encryption along with the completed registration form and signed public keys that are to be certified by the CA. All this information is sent to the CA as a certificate request message. After verifying the card data with the issuer through the financial network, the CA then generates the necessary certificates for the cardholder's public keys and sends these to the cardholder. The cardholder is now ready to participate in the SET payment system.

3. Electronic Check Payment Systems

When discussing electronic checking, there are two current approaches: one that builds on the existing financial infrastructure and another that essentially is a "stand alone" system. The first is nicely represented by one of the ongoing efforts of the Financial Services Technology Consortium (FSTC), the other by the NetCheck scheme. The FSTC is "a group of American banks, research agencies, and government organizations, formed in 1993, that have come together to assist in enhancing the competitiveness of the U.S. financial service industry" (O'Mahony et al., 1997, p. 126). Let's take a look at each of these approaches. But, before we do, here are some of the benefits of electronic checking as suggested by the electronic check portion of FSTC's home page:

- Provides an electronic payment alternative for trading over public data networks.
- Enables banks to gather deposits electronically.
- By retaining the essential features of the existing system, can be rapidly adopted.
- Has great flexibility in that it can support a variety of other instruments: certified checks, cashiers checks, etc.
- Can be used by all market segments from the individual to the corporation.
- Will permit automatic posting of account information; e.g., by being able to be integrated into other applications such as one for accounts receivable.
- Will provide a secure, trusted instrument.
- Will allow the secure integration of the existing financial payments infrastructure with the public networks.
- Makes authentication of checks much easier through the use of public-key certificates.
- Due to the use of cryptographically protected electronic signatures, will eliminate most of the common causes of bad paper checks. (http://www.fstc.org)

Financial Services Technology Consortium (FSTC). In the normal scheme of making a payment by check, the payer writes a check and gives it to the payee who deposits it in his/her account. The payee's bank then uses the established clearing mechanism, such as the automated clearing house (ACH), to clear the check so the funds can be made available to the payee's account. Even with the advances in clearing over the last 20 years or so, there is still a lot of "transaction overhead" in processing check paper. Just the logistics of handling the paper plus the inherent time delays are reasons enough to move to an electronic, paperless system.

Borrowing from O'Mahony et al., Figure 1 shows the FSTC's concept for an electronic check transaction. As can be seen, the payer can now send the check electronically to the payee using, for example, the Internet. Using appropriate cer-

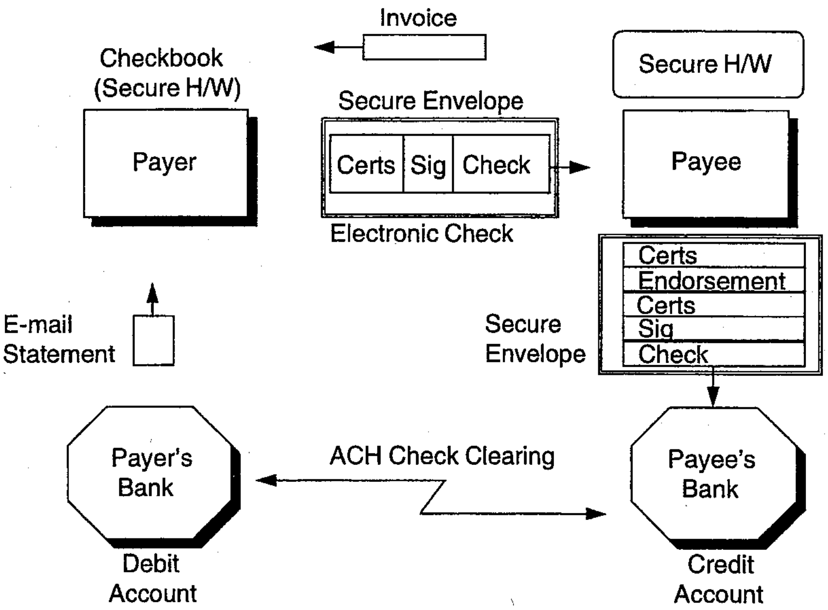

Figure 1. The FSTC electronic check concept (from O'Mahony et al, 1997)

tificates and digital signatures for identity purposes, the electronic check is sent to the payee (perhaps a merchant) in a "secure envelope." This envelope could be secure e-mail or "an encrypted interactive dialogue between the two parties." Similarly, after electronic endorsement, the payee forwards the check to his/her bank for processing. At this point, the normal ACH sort of clearing takes place. As will be noted from Figure 1, both the payer and payee will have to be equipped with the necessary secure hardware to maintain certificates and keys, and information on checks received and sent.

According to O'Mahony et al., the FSTC envisions four types of transactions: deposit-and-clear, cash and transfer, lockbox, and funds transfer. Table 2 shows the steps involved in each type.

NetCheck. Another type of electronic check scheme, which doesn't use the existing financial system, is NetCheck developed by Clifford Neumann of the University of Southern California (Kalakota & Whinston, 1997). This scheme relies on a Kerberos network authentication system for validating the identities of the various users and providing a means for encryption of sensitive information.

Table 2 The steps involved in each of the four FSTC transactions

Step	deposit-and-clear	cash and transfer	lockbox	funds transfer
1	payer sends check to payee	payer sends check to payee	payer sends check to *payee's bank*	payer sends check to *his/her bank*
2	payee endorses check and sends to his/her bank	payee endorses check and sends to *payer's bank*	payee's bank clears check	payer's bank transfers funds to payee's bank
3	payee's bank clears check	payer's bank provides payee formal acknowledgement		
4		payer's bank transfers funds to payee's bank		
comment >>	all parties must be able to process electronic checks	payee can accept electronic checks but his/her bank can't	this "lockbox" is a service offered by U.S. banks to corporate clients	only the payer's bank needs to be able to process electronic checks

NetCheck works as follows:

- When a customer (payer) wishes to send a check to a merchant (the payee), he or she creates an electronic check consisting of the amount, unit of currency, date, account number, and payee. All this information is unencrypted.
- Then the customer obtains what is called a "ticket" from a Kerberos server. This ticket, along with an "authenticator," contains information that will allow positive identification by his/her bank once the electronic check gets there.
- Now the customer creates the authenticator to which the ticket will be appended. The authenticator contains a checksum of the check's contents and a session key shared with the bank. (The ticket also contains this session key, and has been encrypted by the Kerberos server with the bank's secret key. This means that, once the electronic check reaches the bank, the bank can open the ticket (using its public key), get the session key to open the authenticator, and use the information in the authenticator to authenticate the payee

and the contents of the check.) This ticket/authenticator combination constitute the customer's signature on the electronic check.

- The check along with the customer's signature are then forwarded to the merchant, usually by some secure means (e.g., using another Kerberos ticket to get a session key for this purpose).
- Upon receipt of the check, the merchant verifies the unencrypted part (amount, unit of currency, etc.) and electronically endorses it following the same procedures used by the customer. That is, obtaining a ticket, creating an authenticator, and appending these to the check and forwarding it to his/her bank.
- If this is the same bank as the customer's, the bank can open the tickets and verify the validity of the merchant's endorsement and the customer's signature and make the appropriate transfer of funds.
- If this is not the same bank as that of the customer's, another endorsement, in the same way as the merchant endorsed the check, is "placed" on the check and it continues through the banking system this way until reaching the customer's bank. Upon reaching the customer's bank, the series of endorsements can be traced back to the merchant's account for crediting purposes.

O'Mahony et al. (1997) believe schemes developed by universities, such as NetCheck, will probably not be adopted by the public unless first adopted by financial organizations already known and trusted. However, according to Kalakota & Whinston (1997) one of the promising potential uses of the NetCheck scheme is as an internal accounting system within large companies with the electronic checks serving as a form of internal cash:

> With the advent of Intranets, companies are looking at electronic payment technology as a crucial element of internal resource management. For instance, a company can provide digital money to its employees and charge for using the various networks, servers, and databases in order to ensure fair resource allocation. Each user in the organization could be given an account and be

> billed for his use of various resources, a measure that would allow organizations to use accounting mechanisms to manage resources more effectively. Also, with increased cross-functional activities, payment and settlement could become an integral part of Intranet-based commerce, where departments exchange goods and services among themselves. (p. 166)

4. Electronic Cash Payment Systems

Cash is by far the most popular form of payment. According to O'Mahony et al. (1997) this is because:

- Cash is acceptable almost anywhere.
- Cash guarantees the payment; once handed over, the payment transaction is complete.
- There are no transaction charges.
- Cash provides for anonymity.

Although no electronic cash system has yet achieved all of these, there are several that are coming very close. In this section we'll examine three: Ecash, CyberCoin, and Mondex.

Ecash. Ecash was developed by a company based in Holland and the United States called DigiCash. Ecash was launched on a trial basis in October 1994. The first bank to issue Ecash was the Mark Twain Bank of St. Louis, Missouri. Now several other banks and Internet service providers have started issuing Ecash (O'Mahony et al., 1997).

To use Ecash, the client (customer) and merchant must have accounts at the same bank (this may change later). To acquire electronic coins, the client withdraws them from the bank. However, the coins to be withdrawn are first partially created by the client's cyberwallet software by assigning a serial number to each coin. This serial number is then forwarded to the bank blinded; that is, by applying a "blinding factor" to the serial number so that it will not be possible for the bank to know that number. The reason for this is to maintain anonymity; if the bank knew the serial numbers, when the coins are finally spent and returned to the bank, the bank could determine who originally owned the coins.

Now the bank, using an appropriate signature key, assigns value to the coins. The value of the coins is that requested by the client. Since the bank uses different signature keys for each value (e.g., one-cent, five-cent, ten-cent, etc.), the value of each coin is limited to the amount represented by its signature key. The client can easily verify coin values by using the appropriate public key sent from the bank. The request for coins from the client can be for many coins of different denominations. This request is signed with the client's secret key and encrypted with the bank's public key. The response to the request includes all the coins requested, each signed with the appropriate value key. This response is also signed by the bank but it is not encrypted since only the client can unblind the coins before using them.

Once an order has been decided upon, the merchant will send a payment request to the client's cyberwallet giving all necessary information such as a description of the order, order amount, merchant's bank account, etc. The cyberwallet, in turn, shows this information to the client and asks if payment is desired. If so, the wallet gathers the coins needed to pay the *exact* amount (change from the merchant might compromise anonymity since the merchant could record those serial numbers). If more coins are needed, the wallet can withdraw more. The coins are then sent to the merchant encrypted with the bank's public key to prevent theft or tampering. Along with the coins, other information (called *payment info*) is included such that the merchant can verify that the client agrees with the order. Also included with the payment information is a hash of *payer_code*, a secret generated by the client. The purpose of this secret and the hash which eventually goes to the bank, is to provide subsequent proof of payment should the merchant dispute this. For this to work, a hash of *payment info* is also included in the information sent to the bank via the merchant. This ensures the bank that *payment info* (and the hash of *payer_code*) was not tampered with by the merchant.

The merchant now sends everything received from the client (*payment info*, the encrypted coins, and a hash of *payment info*) to the bank with the merchant's signature and encrypted with the bank's public key. The bank checks for double spending by verifying that the coins have not already been deposited. This prevention of double spending requires the bank to keep a large database of spent coins. Once coins have expired, they can be removed from this database. The cyberwallet automatically returns unspent coins to the bank before they expire.

Once the coins are verified by the bank, they are deposited to the merchant's account and a message, signed by the bank, is returned to the merchant. Then the merchant, assured he or she has received good coins, delivers the goods and receipt to the client. Should the merchant fail to do this, the client can reveal the *payer_code* secret as proof of payment.

CyberCoin. Another form of electronic cash is CyberCoin from CyberCash, Inc. of Reston, Virginia. This company also provides a secure credit card service and, more recently, a secure checking service called PayNow (CyberCash home page, 1998). CyberCoin, unlike Ecash, does not involve the use the individual coins. Instead, when the customer buys cash an account is set up with the CyberCash server. The sever is linked with the banking network making it easy to transfer value from a customer's bank account or transfer value to a merchant's bank account. According to O'Mahony et al. (1997), "Setup messages using public-key cryptography are used to load a CyberCash wallet [on the customer's computer] with CyberCoin cash and to set up symmetric key material" (p. 182). The symmetric key material is then used to secure the transaction between the customer and the merchant, making it more efficient than the computationally intensive Ecash process. The merchant verifies with the CyberCash server that the necessary amount of electronic cash is available from the customer. Once verification is received back from the CyberCash server, the goods can be delivered and the appropriate transfer of CyberCoin cash can be made from the customer's account to the merchant's account via the CyberCash server.

Since the details of CyberCoin remain proprietary, the description of this system must necessarily be somewhat general. However, the main differences between CyberCoin and Ecash would seem to be:

- With CyberCoin there are no electronic tokens as with the Ecash where individual coins are minted.
- With CyberCoin there is no anonymity since the CyberCash server records each user's transaction.
- With CyberCoin, before accepting the coins the merchant must get verification from the CyberCash server. However, this is similar to the check made by the bank with the Ecash system to ensure that a coin hasn't already been spent.

Mondex. The third electronic cash system we will look at is called Mondex. It was developed by a British banking organization, NatWest, in 1990 and, according to O'Mahony et al. (1997), is now being promoted internationally. Mondex takes a completely different approach to electronic cash than Ecash or CyberCoin. With Mondex, a sophisticated prepayment smart card is used to store the cash. O'Mahony et al. describe the card this way:

> The Mondex payment scheme relies on the use of a contact chip card[5], the core of which contains a chip based on a modified Hitachi H8/310 microcontroller. This is an 8-bit microprocessor with an on-chip RAM, ROM, and EEPROM[6]

[5] A chip (or "smart") card is similar to the traditional credit card but has an integrated circuit embedded within it. A contact chip card requires an external power supply and clock to drive the integrated circuit and a physical path for the input and output of data. Therefore, a contact card must make physical contact with a card reader of some sort. A contactless card, on the other hand, does not require direct contact with the reader; relying instead on some sort of electrical coupling.

[6] The Mondex card is, in effect, a special purpose, very small computer. The microprocessor does all the data processing while the RAM (random access memory), ROM (read-only ↗

> as well as a serial communications controller to allow it to converse with the outside world. The control program for the Mondex payment scheme is implemented in the ROM of this microcontroller and allows value to be transferred from one Mondex chip to another using a proprietary (and secret) chip-to-chip protocol. (p. 183)

A key feature of such cards is the use of a personal identification number (PIN) ensuring that only the legitimate holder can use the card.

To load the card with value, the user goes to an automated teller machine (ATM) or special telephone equipped with a Mondex *interface device* (IFD). The IFD allows the inserted Mondex card to connect and dialog with the bank's value box. The value box is made up of a large number of Mondex cards that hold value on behalf of the Mondex card holders. A *value control and management system* monitors all value transfers to and from the value box and makes appropriate adjustments to the holders' bank accounts.

In a similar fashion, the card holder can use the cash loaded to the Mondex card to purchase something by inserting it into the merchant's *value transfer terminal*. This terminal has a Mondex IFD to, again, facilitate the transfer of value from the customer's card to the merchant's card. There is no need for the bank to be involved during the purchase transaction; later the merchant can transfer value to the bank's value box and have it credited to his or her account.

Some other features of Mondex are: (1) a keyring sized reader to tell the holder how much value is left, (2) hardware to permit vending machines to accept the card, and (3) a PC card[7] device that would allow a computer to accept the card for pay-

↘ memory), and EEPROM (electrically erasable programmable read-only memory) provide the microprocessor the memories it needs to function. Of course, this computer also needs a way to communicate with the outside world and this is provided by the communications controller.

[7] A PC card is a small card about the size of a credit card that can be inserted into a slot on a personal computer.

ments over the Internet.

In the interests of fraud prevention, the system does remember the last 10 transactions each card makes and the last 300 transactions made by the merchant's terminal. This makes the system lose a lot of its anonymity; one of the benefits of traditional cash.

As can be seen, the use of a smart card for electronic cash offers a great deal of convenience as long as the merchant is equipped to accept it. Since there is no need to "check first" with the bank, the transfer of payment value can take place instantly.

5. Electronic Micropayment Systems

Even though a system like Ecash can make payments as small as one-cent, there is a need for a means to efficiently handle even smaller payment amounts. This is especially true with the burgeoning of electronic commerce where a customer might wish to make a simple database query such as getting a quotation on a share of stock or copying a portion of a financial report. In these cases we might be talking about thousands of transactions in a relatively short time so that, although each is for a small amount, the total revenue to a company could be substantial if there is an economical way to handle these payments. Let's look at a couple of these systems: MilliCent and PayWord.

MilliCent. MilliCent is a micropayment system developed by Digital Equipment Corporation. This system is meant to handle payments under $10.00 down to a minimum of $0.001. According Digital's MilliCent home page (http://www. millicent.digital. com):

> The MilliCent system is designed to be extremely flexible in meeting current and future business requirements for three segments of the evolving online marketplace:
>
> - For software developers and distributors, MilliCent software can support the sale or rental of Java applets, ActiveX controls, software add-ons, and games.

- In the information marketplace, MilliCent software can be used to sell stock quotes, database queries, articles, research, cartoons, clip art, music, and video.
- On corporate intranets, MilliCent software can meter access to applications, services, databases, and information resources. (White Papers: The MilliCent Microcommerce System: Defining a New Internet Business Model)

It is worth noting that besides providing an efficient means of making micropayments over the Internet, MilliCent, as indicated by the third segment just cited, can be used over internal corporate "Internets," called intranets, to allocate and monitor usage of scarce corporate resources.

The MilliCent system is made up of three parts: a broker, the customer, and the vendor. The broker, according to O'Mahony et al., will normally be a financial institution or network service provider. The purpose of the broker is to act as a go-between between the customer and the vendor. If the customer had to buy scrip (the "currency" of MilliCent) from *each* vendor, at least three problems arise: (1) the customer has to maintain a number of vendor accounts (assuming the customer will be dealing with several vendors), (2) the vendor has to maintain a large number of customer accounts, and (3) the time required for a customer to use enough scrip from a specific vendor to make a macropayment economical will be unduly long. The broker solves all these problems by buying vendor script from many vendors and selling it to the customers as needed. The system can be made even more efficient by having the vendor license the broker (as a trusted partner) to produce a specific amount of vendor scrip to sell to customers.

Initially the customer, using a macropayment system such as SET (see the Credit Card Electronic Payment Systems section above), buys *broker scrip* from the broker. For example, the customer might buy $5.00 worth of broker scrip; maybe enough to last about a week. To make a purchase, the customer buys scrip for a specific vendor by trading in some of his or her broker scrip. In the O'Mahony example, the customer buys 20 cents worth *vendor scrip* using $5.00 of broker scrip.

The broker returns the change of $4.80 in broker scrip along with the vendor scrip. Now, without further reference to the broker, the customer can buy from the vendor by sending vendor scrip. Again following the O'Mahony example, the customer sends the 20 cents to the vendor with the purchase request and buys something worth one cent. The vendor returns 19 cents in vendor scrip change. The customer can continue to use this "change" until it is completely used up. Because these repeated payments can be made without reference to a third party, the operation is very efficient in terms of network connections.

The scrip contains the following information:

- The vendor's identification (the script can be used only at this vendor).
- How much the scrip is worth.
- A unique identification number to prevent double spending. The vendor checks that this number has not already been spent.
- A customer identification number. This number is unique to every customer and is used to create the customer secret which the customer and vendor share for encryption of the network connection between them.
- The date the scrip becomes invalid. The purpose of this is to limit the number of identification numbers the vendor must retain.
- Other information such as the customer's age or country.
- A certificate to prove the authenticity of the scrip. This certificate is generated by the vendor (or trusted broker) using a secret only the vendor (or trusted broker) knows. The certificate is generated by making a hash of the secret plus other information in the scrip. Upon receiving the scrip, the vendor does another hash using the secret and compares it with the certificate received from the customer. If they match, the vendor knows the scrip is valid.

So we now know how the vendor prevents double spending (checks that the unique identification number on the scrip has not been spent) and how the authenticity of the scrip is verified (through the use of the certificate that is part of the scrip). Also, we know that, if required, the network connection can be encrypted using something called the customer secret. However, if network connection encryption is not required, there is another MilliCent protocol that, although forsaking privacy, will ensure that the scrip cannot be stolen. This is accomplished by the customer generating something called a *request signature*. The request signature is generated by hashing the scrip, customer secret, and purchase request together. Then the scrip, request, and request signature are sent to the vendor. The vendor then uses the scrip, request and his or her customer secret (shared with the customer so it should be the same) to do a hash and get another request signature for comparing with that received from the customer. If they match, the vendor knows the scrip came from a valid customer and was not stolen.

In summary: "MilliCent is an efficient, lightweight, flexible micropayment system. It can support multiple brokers and vendors and can be extended for use with many applications [for example, it could be used to meter the amount of access an employee has to a particular server within an organization]" (O'Mahony, et al., 1997, p. 208).

PayWord. Another micropayment system is PayWord. This system was developed by Ron Rivest and Adi Shamir of the famous Rivest, Shamir, and Adleman trio; the inventors of the RSA algorithm which has become the de facto standard for public-key cryptography. As with many electronic payment schemes, there is a broker involved with PayWord. The broker maintains the user and vendor accounts just as with the MilliCent scheme. As opposed to MilliCent, however, PayWord is a credit-based system so that what is exchanged between the customer and the vendor must be redeemed with the broker who has vouched for the customer's credit. This vouching is accomplished by the broker issuing a PayWord *certificate* once the customer has open an account using some sort of macropayment. The certificate is digitally signed by the broker and provides the follow information: the broker, the user, the user's delivery address, the user's public key, the expiration date of the certificate (typically one month to limit fraud), and optional information such as credit limits, etc. With the broker's public key, the vendor, can "open" this certificate and know the customer has been authorized by the broker to generate value-representing *paywords* and that these paywords will be redeemed by that broker.

To begin the payment process, the customer generates a *commitment.* This commitment is to a specific *payword chain*; a series of hashes (paywords) that represent value. By signing the commitment, the customer is telling the vendor and broker that he or she is "committing" to honor the specific payword chain identified within the commitment. This way the vendor knows it is OK to accept hashes within the chain as payment and the broker knows it is OK to redeem these hashes by transferring money from the customer's account to the broker's account. The commitment contains the following information: the vendor at which the payword chain is valid (the chain is vendor-specific), the certificate described above, something called the *root* of the payword chain (W_0), the expiration date of the commitment, and additional information such as the length of the payword chain. As mentioned, the commitment is digitally signed by the customer.

The key to the payword scheme is the generation of the payword chain by the customer. Once the length (n) of the chain is decided, the customer will hash a random number, say W_n, to get a new number W_{n-1}. Then Wn-1 is hashed to get W_{n-2} and so forth until W_0, the root, is obtained. W_1, W_2, W_3, W_4, … W_n are paywords each worth some set amount; e.g., one cent. The number n is determined by the number of paywords the customer thinks he or she will spend, say, in a day at that vendor. For example, if that amount is 10 cents (and each payword is worth one cent), n will be 10. Now suppose the customer wants to spend one cent at the vendor. In that case, payword W_1 is sent to the vendor. The vendor, having already received the commitment containing, among other things, the value of W_0, can hash W_1 to get W_0. If the W_0 the vendor gets is the same as the W_0 in the commitment, then the vendor knows the payword is from a valid customer. What if the customer wants to spend more than one payword? For example, if the customer wants to spend five paywords, then he or she sends the highest of the next five paywords; e.g., if the last payword spent was W_1, the payword W_6 is sent along with its number (6) so the vendor knows how many times to hash W_6 to get the last payword received (in this case, W_1). By hashing W_6 five times the vendor should get a match with W_1. It is the vendor's responsibility to keep track of the last payword spent by each customer. If all the paywords are not used up by the expiration date, they can be simply discarded since this is a credit-based system.

To redeem the paywords with the broker, the vendor, maybe at the end of each day, will send to the broker a signed user commitment for that chain and the highest payword spent (for example, W_6). The broker performs the required number of hashes on the payword and checks to see if W_0 matches the W_0 of the commitment; if it does, then the appropriate amount is transferred from the customer's account to the vendor's account.

Because the PayWord system is based mostly on computationally easy hashes, it is highly efficient for many, small micropayments. In comparing PayWord with Millicent, O'Mahony et al. make these observations:

> Unlike the Millicent system, a broker does not have to be contacted for a new vendor payment nor is there any need for scrip change or the returning of unused vendor-specific scrip to the broker. However, PayWord's credit scheme provides more opportunity for user fraud than Millicent, especially if a user's secret key is compromised. (p. 220)

6. Conclusion

This paper has been an attempt to give the reader a taste for some of the different schemes being proposed and, in some cases implemented, for making electronic payments. The reason I believe this subject is of interest is the incredible growth of the Internet and Internet commerce. This growth has tended to outstrip efficient and safe means for making payments; hence the growing interest in electronic payment systems. I have drawn heavily on a single book: *Electronic Payment Systems* by O'Mahony, Peirce, and Tewari (1997) as this seems to be one of the few out there that tackles the subject in any comprehensive way. I want to officially acknowledge and thank these authors for an excellent and timely book which helped me a great deal and understanding the subject and writing this paper.

References

Austenfeld, R. B., Jr. (1998, March). Electronic Commerce: An Overview. *Papers of the Research Society of Commerce and Economics–Hiroshima Shudo University*, pp. 87–127.

CyberCash home page, 1998, http://www.cybercash.com.

Digital Equipment Corporation's MilliCent home page, 1997, http://www.millicent.digital.com.

Financial Services Technology Consortium (FSTC) home page, Electronic Check Project Details, 1998, http://www.fstc.org.

Kalakota, R. & Whinston, A. B. (1997). *Electronic Commerce: A Manager's Guide*. Reading, MA: Addison Wesley Longman, Inc.

O'Mahony, D., Peirce, M., & Tewari, H. (1997). *Electronic Payment Systems*. Norwood, MA: Artech House, Inc.