State of the Art in Electronic Payment Systems*

N. Asokan†
Nokia Research Center, Helsinki, Finland

P. Janson, M. Steiner, M. Waidner
IBM Research Division, Zurich Research Lab, Rüschlikon, Switzerland

October 4, 1999

1 Introduction

The exchange of goods conducted face-to-face between two parties dates back to before the beginning of recorded history. As trade became more and more complicated, increasingly abstract representations of value were devised, progressing from barter through bank notes, payment orders, checks, credit cards, and now electronic payment systems.

Traditional means of payment suffer from various well-known security problems: Money can be counterfeited, signatures can be forged, and checks can bounce. Electronic means of payment retain the same drawbacks. In addition, they entail some additional risks: Unlike paper, digital "documents" can be copied perfectly and arbitrarily often; digital signatures can be produced by anybody who knows the secret cryptographic signing key; a buyer's name can be associated with every payment, eliminating the anonymity of cash.

Thus without new security measures, widespread electronic commerce is not viable. On the other hand, properly designed electronic payment systems can actually provide better security than traditional means of payments, in addition to flexibility of use. This paper provides an overview of electronic payment systems, focusing on issues related to security.

A wide variety of payment systems have been proposed and/or implemented in the last few years. Our aim is not to cover all of these payment system. Instead, we focus on systems that satisfy one of the following criteria:

- historical importance (pioneering role),
- popularity and/or widespread current deployment, and
- novelty and/or potential for widespread future deployment.

The rest of the paper is organized as follows. In Section 2, we outline an informal model of electronic payment systems. In Section 3 we identify desirable security requirements for electronic payments. In Section 4, we survey the state of the art in the technological tools which can help meet the requirements identified earlier. In Section 5 we present some example payment systems and discuss their properties. In the next two sections, we touch upon some advanced issues. In Section 6 we motivate the need for a general framework within which multiple payment systems can be used. We also describe research efforts in building such frameworks. In Section 7 we refer to the problem of handling disputes, which will inevitably arise when electronic payment systems are deployed widely. In Section 8, we describe the current state of standards activity. Finally, in Section 9 we conclude with some observations about the current status and future possibilities regarding electronic payment systems. Appendix A is a brief overview of some of the relevant basic concepts in security and cryptography.

*An earlier version of this paper appeared in IEEE Computer [1].

†Part of the work done while at the IBM Zurich Research Laboratory.

2 Electronic Payment Models

Commerce always involves a **payer** and a **payee** who exchange money for goods or services. There is also 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 is implemented by a flow of money from the payer via the issuer and acquirer to the payee.

Figure 1 shows typical flows of money in the case of pre-paid, **cash-like** payment systems. In these systems, a certain amount of money is taken away from the payer before purchases are made. This amount of money can be used for payments later. Smartcard-based electronic purses, electronic cash, and bank checks (such as certified checks) fall into this category.

Figure 2 shows typical flows of money in the case of bank account based systems, which include pay-now systems and pay-later systems. In pay-now payment systems, the payer's account is debited at the time of payment. Automated-teller-machine (ATM) cards 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: Because a payment is always done by sending a "form" from the payer to the payee (whether it be a check or a credit card slip or some other form), we call these systems **check-like**.

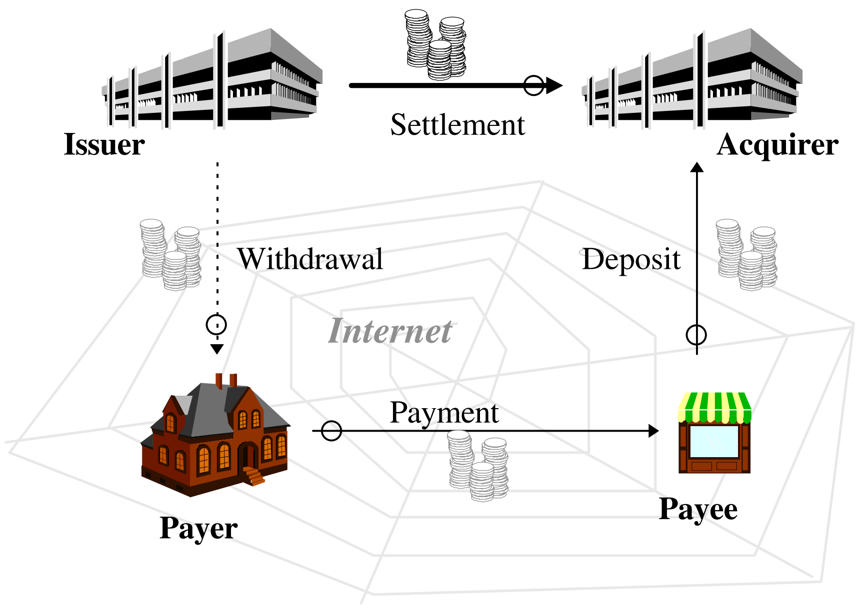

Figure 1: Money flow in a cash-like payment system (circle in arrow indicates party initiating the flow)

Both types of payment systems introduced above are **direct** payment systems: A payment requires a direct interaction between payer and payee. There are also **indirect** payment systems in which either the payer or the payee initiates payment without the other party involved being online. Electronic funds transfer is one example of an indirect payment system.

While we can consider electronic funds transfer over financial networks reasonably secure, we face challenges of a new dimension in securing payments over open networks like the Internet. This paper surveys the

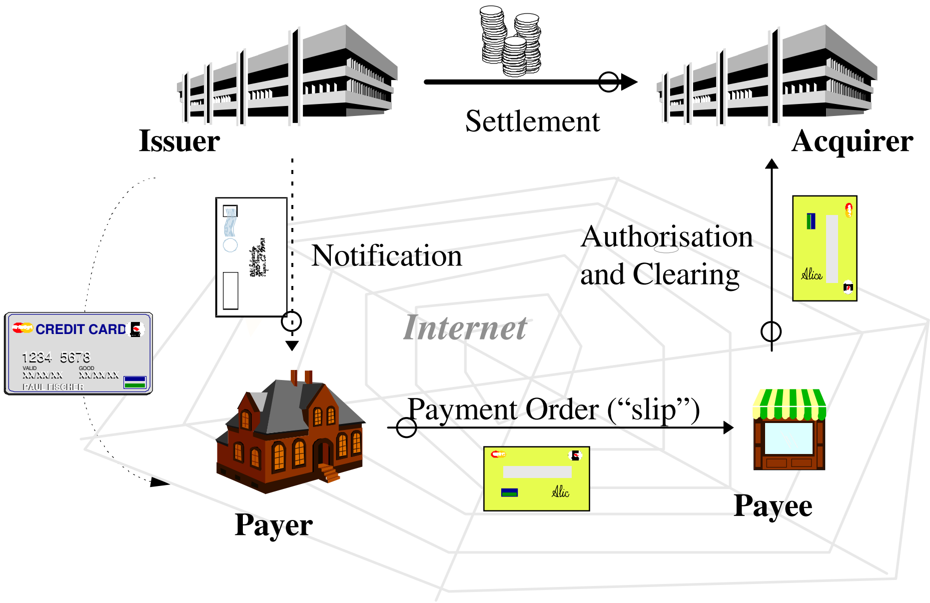

Figure 2: Money flow in a check-like payment system (circle in arrow indicates party initiating the flow)

state of the art in payment technologies and sketches emerging developments, focusing mainly on protocols suitable for use in open networks.

3 Security Requirements

The concrete security requirements of electronic payment systems vary, depending both on their features and the trust assumptions placed on their operation. In general, however, electronic payment systems must exhibit integrity, authorization, confidentiality, availability, and reliability.

The interests of the payer, payee and bank contradict each other to some degree. Therefore the trust which a system requires from the parties in each other should be as little as possible. This requirement is often referred to as **multi-party security**. One way of reducing trust is to make actions, in particular authorization, accountable. Digital signatures are a core ingredient to achieve **accountability**.

3.1 Integrity and authorization

The integrity requirement at a global level is that the payment system should guarantee that no new monetary value is created by any player within the digital system. The integrity requirement from an individual player is that no money is taken from him without his explicit authorization. A user may also require that receipt of payments without his explicit authorization be disallowed in order to prevent, for example, unsolicited bribery. Payment can be authorized in three ways: via out-band authorization, using passwords, and using signatures. The latter two are based on sender authentication of authorization messages.

- **Out-band authorization**: In this approach, the verifying party (typically a bank) notifies the authorizing party (the payer) of a transaction. The latter is required to approve or deny the payment using a secure, out-band channel such as surface mail or the phone.

This is the current approach used for credit card transactions for mail orders and phone orders: Anyone who knows a user's credit card data can initiate transactions. Legitimate users must carefully check their account statements and complain about any unauthorized transactions. If the user does not complain within a certain time, transactions are considered "approved" by default.

- **Password authorization**: In the password-based approach, every message from the authorizing party includes a cryptographic check value computed using a secret known only to the authorizing and verifying parties. This secret can be a personal identification number, a password, or any form of shared secret. See Appendix A.1.1 for more explanation on message authentication based on shared secrets.
- **Signature authorization**: In this type of transaction, the verifying party requires a digital signature of the authorizing party. Digital signatures also provide **nonrepudiation** of origin. See Appendix A.1.2 for more explanation on digital signatures.

3.2 Confidentiality

Some parties involved may wish confidentiality of transactions. 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, and so on. Typically, the confidentiality requirement dictates that this information be restricted only to the parties involved. Where **anonymity** is desired, the requirement may be to further limit this knowledge to certain subsets of the parties only, as described later.

3.3 Availability and reliability

All parties require the ability to make or receive payments whenever necessary. Payment transactions must be atomic: They occur entirely or not at all, but they never hang in an unknown or inconsistent state. No payer would accept a loss of money (not a significant amount, in any case) due to a network or system crash.

Availability and reliability presume that the underlying networking services and all software and hardware components are sufficiently dependable. Recovery from crash failures requires some form of stable storage be available to all parties as well as specific resynchronization protocols. These fault tolerance issues are not discussed here: Most payment systems do not address them explicitly and the standard mechanisms to increase availability and reliability [2] can be applied to most protocols in a rather straightforward manner.

4 Technology Overview

As we saw, electronic payment systems must enable an honest payer to convince the payee to accept a legitimate payment and at the same time prevent a dishonest payer from making unauthorized payments, all the while ensuring the privacy of honest participants. In this section, we describe some of the technologies that can be used to meet the security requirements.

4.1 Online versus offline

Offline payments involve no contact with a third party during payment – the transaction involves only the payer and payee. The obvious problem with offline payments is that it is difficult to prevent payers from spending more money than they actually possess. In a purely digital world a dishonest payer can, after each payment, easily reset the local state of his system to a prior state. This is known as the **double spending** problem. Typically, double spending is prevented by the use of **tamper-resistant** hardware (see Section 4.2). Another way to achieve protection against double spending is to use preauthorization: The payee is known to the payer in advance, and the payment is already authorized during withdrawal, as in the case of certified bank checks. A weaker solution is *detect* rather than prevent double spending. Double spending detection techniques guarantee that if a user spends money more than allowed, he can be identified and made accountable. Identification of double spenders can be achieved even if the user was (unconditionally) anonymous during "legal" transactions [3, 4]! Techniques for detection of double spending combined with prevention based on tamper-resistant hardware and can serve as a second line of defense.

Online payments involve an authorization server (usually as part of the issuer or acquirer) in each payment. Online systems obviously require more communication. In general, they are considered more secure than offline systems. Most proposed Internet payment systems are online.

4.2 Trusted hardware

As we saw, prevention of double spending requires tamper-resistant hardware at the payer end. The smartcard is an example. Tamper-resistant hardware may also be used at the payee end. An example is the security module of a point-of-sale (POS) terminal. This is mandatory in the case of shared-key systems and in cases where the payee forwards only the total volume of transactions (rather than every individual transaction) to the bank. In a certain sense, tamper-resistant hardware is a "pocket branch" of a bank and must be trusted by the issuer. However, such trust might not always be justified because smartcards are vulnerable to a number of attacks [5, 6, 7, 8, 9]. Therefore utmost care is necessary in designing protocols which rely on the tamper resistance of a device for their security.

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 correctly. Initially, this could simply be a smartcard. But in the long run, it should become a smart device of a different form factor with secure access to a minimal keyboard and display. Such a device is usually referred to as an **electronic wallet**.

Without such a secure device, the payers' secrets and hence their money are vulnerable to anyone who can access their computers. This is obviously a problem in multiuser environments. It is also a problem even on single-user computers which can be accessed directly or indirectly by others. A virus, for example, installed on a computer could steal personal identification numbers (PINs) and passwords as they are entered. This is an instance of the more general **fake-terminal** attack (see Appendix A.2.1). Even when a smartcard is available to store keys, a virus program may directly ask the smartcard to make a payment to an attacker's account. Thus for true security, trusted input/output channels between the user and the smartcard must exist [10].

4.3 Micropayments

Micropayments are low-value payments (probably less than $1). In some scenarios, such as paying for each tick of a phone call, micropayments need to be made repeatedly and very quickly. Given these constraints, micropayment techniques must be both inexpensive and fast. Achieving both requires certain compromises, either by reducing the (multi-party) security or in making assumption on the usage of such systems. We return to micropayments in Section 5.3.

4.4 Cryptography

Using no cryptography at all implies reliance on out-band security: For example, goods ordered electronically may not be delivered until a fax arrives from the payer confirming the order. The first proposals for Internet payment systems were of this type. Later proposals rely on cryptography to provide reasonable security guarantees.

A wide variety of cryptographic techniques have been developed for user and message authentication, secret communication, and nonrepudiation. They are essential tools in building secure payment systems over open networks that have little or no physical security. See Appendix A for a quick introduction of basic concepts in cryptography and security.

4.4.1 Shared-key cryptography

Authentication based on shared-key cryptography requires that the prover (the payer) and a verifier (the issuer) both have a shared secret. The shared key may be, for example, a password or a PIN or a key in a shared-key cryptosystem. The most commonly used shared-key cryptosystem is the Data Encryption

Standard (DES) and its stronger variant triple DES [11]. However, the US National Institute of Standards Technology (NIST) is in the process of selecting a new standard for shared-key encryption[1]. Once finalized, this Advanced Encryption Standard (AES) will become the most commonly used shared-key cryptosystem.

Because both sides have exactly the same secret information, shared-key cryptography does not provide nonrepudiation. If payer and issuer disagree about a payment, there is no way to decide if the payment was initiated by the payer or by an employee of the issuer. Authenticating a transfer order on the basis of shared keys is therefore not appropriate if the payer bears the risk of forged payments [12].

If authentication is to be done offline, *each* payer-payee pair needs a shared secret. In practice this means that a common master key is present at each payee end, to enable the payee to derive the payer's key. Tamper-resistant security modules are required to protect the master key.

4.4.2 Public-key digital signatures

Authentication based on public-key cryptography requires that the prover has a secret signing key and a certificate for its corresponding public signature verification key. The certificate is issued by a well-known authority. Most systems now use RSA [13] signatures, but there are several alternatives.

Digital signatures can provide nonrepudiation – disputes between sender and receiver can be resolved. Digital signatures should be mandatory if the payer bears the risk of forged payments. A special form of signatures called **blind signatures** [14, 4] is used to construct payment systems that provide user anonymity (see Section 4.5). A blind signature on some message is made in such a way that the signer does not know the exact content of the message.

4.5 Payer anonymity

Payers prefer to keep their everyday payment activities private. Certainly they do not want outsiders not involved in their transactions to be able to observe and track their payments. Often, they prefer the payees (shops, publishers, and the like) and in some cases even banks to be incapable of observing and tracking their payments.

There are two levels of anonymity of a player: **untraceability** simply means that an adversary cannot determine a player's identity in a run of the payment protocol; **unlinkability** means that, in addition, participation by the same player in two different payments cannot be linked. Some payment systems provide untraceability and unlinkability of the payer. Both are considered useful for cash-like payments since cash is also untraceable and unlinkable. By encrypting the contents and addressing information in all message flows between payer and payee, any payment system can be made unlinkable by outsiders. Payer untraceability with respect to the payee can be achieved by using pseudonyms instead of real identities. Some electronic payment systems are designed to provide untraceability or even unlinkability with respect to the payee (iKP [15], for example, offers this as an option).

5 Representative Payment Systems

Table 1 lists some examples of payment systems along with an indication of relevant properties. Pointers to more up-to-date information on these payment systems and others may be found at `http://www.semper.org/sirene/outsideworld/ecommerce.html`.

A rather general security scheme that uses public-key signatures is Secure Socket Layer (SSL). SSL is a socket-layer communication interface that allows two parties to communicate securely over the Internet. It is not a payment technology per se (and hence does not fit into Table 1), but is often used as a means to secure payment messages. SSL does not support nonrepudiation.

Many of the complete payment systems (of all models) were proposed and implemented during the 1990s. But the protocol ideas themselves are much older. The use of digital signatures for both online and offline payments, anonymous accounts with digitally signed transfer orders, and anonymous electronic cash were all introduced during the 1980s (see [23] for an early survey).

[1] `http://www.nist.gov/aes`

System[a]	**Online/Offline**	**Anonymity**	**Cryptography**	**Trusted h/w**
Check-like payment systems				
First Virtual [16]	online	none	none	none
iKP [15]	online	none	public-key	none
OpenMarket [17]	online	none	public-key	none
echeck [18, 19][b]	offline	none	public-key	tamper-resistant
SET [20]	online	none	public-key	proposed
Cash-like payment systems				
e-cash[c]	online	unlinkable	blind signatures	none
CAFE [21]	offline	unlinkable	blind signatures	tamper-resistant
Mondex[d]	offline	none	public-key	tamper-resistant
CEPS[e]	offline	none	public-key	tamper-resistant
NetCash [22]	online	unlinkable	shared-key only	none
Anonymous credit cards [23, 24]	online	unlinkable	public-key	none
Micropayments				
NetBill [25]	online	none	shared-key only	none
CAFE Ticks [26]	offline	none	OWHF[f]	none
MilliCent [27]	online	none	shared-key+OWHF	none
μ-iKP [28]	offline	none	public-key+OWHF	none
IBM Micro Payments [29][g]	offline	none	public-key	none

Table 1: Information Sources for Representative Payment Systems

[a]proposed or existing standards are indicated in boldface.
[b]by FSTC, `http://www.echeck.org`
[c]by *Digicash*
[d]`http://www.mondex.com/`
[e]Common Electronic Purse Specifications, `http://www.europay.com/SmartCard/html/Index_ceps.html`
[f]one-way hash function
[g]previously known as Minipay

5.1 Check-like Systems

First Virtual was an example of a payment system that did not use cryptography. A user had an account and received a password in exchange for a credit card number. But the password was not protected as it traversed the Internet. Such a system is vulnerable to eavesdropping. First Virtual achieved some protection by asking the payer for an acknowledgment of each payment via e-mail, but the actual security of the system was based on the payer's ability to revoke each payment within a certain period. In other words, there was no definite authorization during payment. Until the end of this period, the payee assumed the entire risk.

The realization that cryptographic techniques are necessary to provide a reasonable level of security for Internet credit card payments led to several proposals. In particular iKP [15] set the theoretical foundations for such protocols in clearly specifying the involved security requirements and providing a systematic and modular approach to satisfy them. These efforts culminated in the proposed standard SET Secure Electronic Transactions, described in Section 8.

A payment switch is an online payment system that implements both the pre-paid and pay-later models, as exemplified by the OpenMarket payment switch [17]. OpenMarket's architecture supports several authentication methods, depending on the payment method chosen. The methods range from simple, unprotected PIN-based authentication to challenge-response-based systems. In the latter case, it is possible to use a smartcard or other hand-held devices may be used for computing the response.User authentication therefore is based on shared-key cryptography. However, authorization is based on public-key cryptography: The OpenMarket payment switch digitally signs an authorization message, which is forwarded to the payee. The payment switch is completely trusted by users who use shared-key cryptography.

The FSTC (Financial Services Technology Consortium) Electronic Check project [18] uses a tamper-resistant PCMCIA card and implements a check-like payment model. A primary design principle in the FSTC electronic check mechanism appears to be to mirror the steps in paper check transactions as much as possible.

5.2 Cash-like Systems

There are two main motivations for cash-like systems: Avoiding the need to have an account with a financial institution (e.g., as in pre-paid phone cards) and/or the possibility of untraceable, and possibly even unlinkable, transactions. So far, the only payment systems mentioned here that provide untraceability and unlinkability against payee and issuer are Digicash e-cash, an online system and CAFE (Conditional Access For Europe) [21], an offline system. Both are cash-like payment system based on blind signatures.

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" that has a randomly generated serial number, blinds it, and sends it to the bank. If the user is authorized 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.

NetCash [22] and anonymous credit cards [23, 24] also provide untraceability and unlinkability. But they are based on the use of trusted intermediate systems that change electronic money of one representation into another representation, without revealing the relation. Neither e-cash nor CAFE assume the existence of such trusted third parties.

Most proposed payment systems based on electronic hardware are offline cash-like systems. All these systems can be used for Internet payments. But none are actually being used for Internet payments, except in small-scale trials, at the time of this writing. The main technical obstacle is that they require a smartcard reader attached to the payer's computer. Inexpensive PCMCIA smartcard readers and standardized infrared interfaces on notebook computers will solve this connectivity problem.

CEPS is a proposed standard for smartcard-based cash-like systems. It evolved from systems already deployed such as Geldkarte[2] or Proton[3]. Mondex is the only system that enables offline transferability: The

[2] `http://www.zka.de/`
[3] `http://www.protonworld.com`

payee can use the amount received to make a new payment himself, without having to go to the bank in between. However, this seems to be a politically unpopular feature as the risks are rather hard to control in such systems. CAFE is the only offline system that provides strong payer untraceability and unlinkability. CAFE also provides loss tolerance, which allows the payer to recover from coin losses (but at the expense of some anonymity in case of loss) [30]. CEPS, Mondex and CAFE are all capable of handling multiple different currencies simultaneously.

A class of payment systems known as "fair" cash-like payment systems support untraceability for the payer (like e-cash or CAFE) but allow a third party to trace selected payments under certain conditions [31, 32, 33]. The development of such systems is motivated by law enforcement related concerns that completely unlinkable payment systems can aid in money laundering, or in committing "perfect crimes" [34, 35]. The latter means that in untraceable payment systems it is possible to remotely blackmail users or the bank without leaving any traces. So far none of these systems has left the stage of theoretical designs.

5.3 Micropayments

Content servers on the Internet will probably have to process such a large number of low-value transactions that it will be impractical to use computationally complex and expensive cryptographic protocols to secure them. One solution is to use flat-fee subscription services. Another more flexible approach is the use of micropayment systems. As mentioned in Section 4.3 such micropayments achieve efficiency only based on some compromises.

Two very prominent examples of micropayment systems are MilliCent [27] and IBM Micro Payments [29] (previously known as "Minipay"): While the former gains its speed by putting complete trust in the payment system provider, the latter achieves its efficiency relying on the assumption that in the case of small payments *a posteriori* fraud detection is sufficient.

A number of proposals [26, 28, 36, 37] were especially designed for situations where a sequence of small payments need to be made to the same merchant. Such situations occur in pay-per-web-page services, or in streaming data services such as video-on-demand. These systems amortize the cost of an initial setup operation over extremely efficient subsequent payment operations, without any loss of multi-party security. They can even be used in paying for network connectivity from anonymous Internet kiosks similar to the use of pre-paid cards in phone booths.

A noteworthy side effect of such scenarios of payment for streaming data services is that the risks of loss for both parties is very small: at most one payment or chunk of streaming data is outstanding, both of them can easily be made of negligible value by shortening the period of unit price. So we achieve highly efficient fair exchange [38] of payments for electronic goods.

All of these proposals use computationally secure one-way hash functions to implement micropayments. In this section, we describe μ-iKP [28], the micropayment proposal for iKP. Informally, a function $f()$ is one-way if it is difficult to find the value x given the value $y = f(x)$. The value x is the **preimage** of y. Given such a one-way function, the payer will randomly choose a seed value X and recursively compute:

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

The values A_0, ..., A_{n-1}, known as coupons, will enable the payer to make n micropayments of a fixed value v to the same payee: 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 iKP payment. The payee ensures, possibly via its bank, that A_n does in fact correspond to a good preimage chain that can be used for subsequent micropayments. The micropayments are then carried out by revealing components of the chain $A_{n-1}, A_{n-2}, \ldots, A_0$ successively to the payee. To clear the payments, the payee presents the partial chain

$$A_i, \ldots, A_j (0 \leq i < j \leq n)$$

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 micropayments. However, nonrepeated (or rarely repeated) micropayments are also a likely scenario in the electronic marketplace: A user surfing the Web may chance upon a single page that costs $0.01. Neither the micropayment setup overhead nor the cost of a normal payment is justified in this case.

μ-iKP solves this problem with a broker: An isolated micropayment from payer P to payee Q is carried out by P making one or more micropayments to broker B. Broker B then 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.

On the other hand, if the amount of the transaction is small, developers can assume a lower risk and so opt to reduce security (for example, by foregoing nonrepudiation). A notable example is NetBill, which is founded on the shared-key technology Kerberos. It implements a check-like debit-payment model. The use of shared-key technology is justified by the performance required to process many micropayments in a short time.

6 Generic Payment Service

As we saw, a large number of electronic payment systems have been proposed in the past few years. The many different payment systems are incompatible with each other. Each individual party will have the ability to use only a subset of these payment systems. When a party wants to make a payment, he must first identify what payment systems he has in common with the payee, and then pick a suitable one from among them for the payment.

For application developers this implies the need to understand the details of different systems, to adapt the code as soon as new payment systems are introduced, and also to provide a way of picking a suitable payment instrument for every transaction. Ideally, an application should be able to make use of any of the several common means of payment available to payer and payee. A unifying framework enabling business applications to use different payment systems in a transparent manner can greatly ease the task of business application developers by relieving them from having to

- make sure that the application knows how to use all the various different payment systems its users are likely to have available, and
- in case multiple payment instruments are available, provide a way of choosing one of them.

Several projects aiming at building such a framework have been reported in the literature [39, 40, 41]. One of the earliest and most complete of these efforts is the Generic Payment Service Framework developed as part of the European Commission project SEMPER [42, 43, 44]. An important aspect of this framework is a set of generic APIs which define common electronic payment services.

An important part of a common payment system framework is the negotiation of mutually acceptable payment system and other related parameters between the parties. This implies a language to represent the necessary meta-information and a protocol for negotiation of parameters. There have been several efforts in this direction. Notable among them are Joint Electronic Payments Initiative by the World Wide Web consortium, Open Trading Protocol[4] by an industry consortium (and the related Internet Open Trading Protocol by the *trade* working group of the Internet Engineering Task Force), and Electronic Commerce Modeling Language[5], also by an industry consortium. None of these efforts have led to a widely accepted standard yet. But their very presence underlines the importance of the problem.

7 Handling Disputes

Support for handling disputes is a crucial aspect of any system providing accountability [45]. However, many payment systems limit themselves only to the generation and collection of evidence. It is assumed that such

[4] `http://www.otp.org`
[5] `http://www.ecml.org`

evidence can be used in some dispute resolution procedure external to the system. Such procedures are usually left unspecified.

A discussion of the issues can be found in [46]. This paper also discusses how the SEMPER generic payment service framework could provide *dispute services* while retaining its generality: To the extent possible, users and applications must be allowed to initiate or respond to payment disputes in a payment-system independent manner. In general, dispute claims can be expressed in terms of *statements* about a (possibly alleged) transaction. The terms of these statements correspond to the parameters used during the invocation of payment services themselves (such as the identities of the parties involved and the amount of transaction) and a finite set of other “environmental” terms (such as the time of transaction). [46] contains the grammar of a proposed language to express payment dispute claims.

8 Standards

Two proposals, Visa’s Secure Transaction Technology (STT) and MasterCard’s Secure Electronic Payment Protocol (SEPP), began as competing standards for credit-card-based online payment schemes. The SET Secure Electronic Transactions standard, designed by MasterCard, Visa, GTE, IBM, Microsoft, Netscape, SAIC, Terisa, and Verisign, has replaced these competing proposals. SET is likely to be widely adopted for credit card payments over the Internet.

SET is a pragmatic approach that paves the way for easy, fast, secure transactions over the Internet. It seeks to preserve the existing relationships between merchants and acquirers as well as between payers and their bank. SET concentrates on securely communicating credit card numbers between a payer and an acquirer gateway interfacing to the existing financial infrastructure.

In our classification, SET falls under the check-like model. The transaction is initiated with a handshake, with the merchant authenticating itself to the payer and fixing all payment data. The payer then uses a sophisticated encryption scheme to generate a payment slip. The goal of the encryption scheme is to protect sensitive payment information (such as the credit card number); limit encryption to selected fields (to ease export approval); cryptographically bind the order information to the payment message; and maximize user privacy. Next the payment slip is signed by the payer and is sent to the merchant. The merchant sends the slip to its acquirer gateway, to authorize and capture the payment. The acquirer checks all signatures and the slip, verifies the creditability of the payer, and sends either a positive or negative signed acknowledgment back to merchant and payer.

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.

Europay, MasterCard, and Visa (known collectively as EMV), are working on standards for smartcard-based electronic payment systems. In 1999 Europay, Visa and others published the first version of the Common Electronic Purse Specifications (CEPS). First implementations of CEPS are expected for 2001. There are currently no efforts to standardize an unlinkable, offline payment system.

9 Status and Outlook

The most frequent form of payment supported by Internet businesses today is to send the credit card number in an SSL encrypted channel from the payer’s web browser to the merchant server. The reasons for this are twofold: first, being able to receive payments in this manner involves very minimal set up effort by the merchant, and second, implementations of the other, more secure, electronic payment systems have been perceived as too complex. However, SSL based Internet payments cannot provide nonrepudiation of origin. There has been at least one case where an Internet pornography merchant claimed fraudulent payments on a massive scale by exploiting the fact that most Internet credit card payments today do not provide nonrepudiation [47].This case is also a striking example that even though the monetary damage to a single user is small ($39.90 in this case), the actual damage to their lives can be much greater.

The technology to provide high grade security for Internet payments already exists. It is now possible to achieve security for all parties, including the perfect unlinkability of the payer. The correct political and business conditions for the widespread deployment of highly secure payment systems are yet to arrive. No one system will prevail; several payment systems will coexist:

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

- High values will be transferred using non-anonymous, online payment systems based on public-key cryptography implementing a check-like payment model. The most promising candidates are SET and the FSTC Electronic Checks.

  SET has had a slower start than expected. But within the next three to five years, it will nevertheless become the predominant method for credit card purchases on the Internet. Initially it is implemented in software only, but will later be supported by smartcards. For some time, the currently preferred method of using SSL to encrypt payment details on their way from payer to payee will coexist with SET.

  While it is likely that FSTC Electronic Checks will be deployed within the US[6] it is not clear if the FSTC design will ever be used, or indeed can be used, internationally.

- Within the next few years, smartcard readers will become widely available on PCs and workstations. This will enable payments of small amounts using pre-paid, offline payment systems that provide a certain degree of unlinkability.

  Several smartcard-based electronic purse schemes are already deployed on a national basis, in particular in Europe, and within a few years most of us will carry smartcards that can be used to buy things offline and in shops, as well as over the Internet. There are good chances that CEPS will become an accepted standard but so far most purses cannot be used for cross-border payments.

- Several micropayment systems will be used with microservice providers, but it is not clear yet whether there will be a single winner in the end.

It is difficult to predict the future of privacy protecting payment systems. CAFE and Digicash's e-cash demonstrated the technical feasibility of strongly untraceable payment systems. But neither has been successful on the market yet. The increasing awareness for privacy risks, in particular on the Internet, might create the necessary demand for such systems.

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 for two reasons: They enable mobility of users and they clearly provide better security, allowing the payer to use untrusted terminals without endangering security.

A few almost equivalent payment systems with the same scope (in terms of the payment model and maximum amounts) will possibly coexist. The reasons are various cultural differences in the business and payment processes, national security considerations that might disqualify some solutions in some countries, and competition between payment system providers.

References

[1] N. Asokan, Phil Janson, Michael Steiner, and Michael Waidner. State of the art in electronic payment systems. *IEEE Computer*, 30(9):28–35, September 1997. A Japanese translation of the article appeared in pp 195-201, *Nikkei Computer* (http://nc.nikkeibp.co.jp/jp/) issue of March 30, 1998.

[2] P. A. Lee and T. Anderson. Fault tolerance – principles and practice. 2nd rev. ed., Dependable Computing and Fault-Tolerant Systems Vol. 3, Springer-Verlag, Wien 1990.

[6] There is currently a larger trial involving the U.S. Department of the Treasury, the Defence Finance and Accounting System and the Federal Reserve Bank.

[3] David Chaum, A. Fiat, and M. Naor. Untraceable electronic cash. In S. Goldwasser, editor, *Advances in Cryptology – CRYPTO '88*, number 403 in Lecture Notes in Computer Science, pages 319–327, Santa Barbara, CA, USA, 21-25 Aug. 1988 1990. Springer-Verlag, Berlin Germany.

[4] Stefan Brands. Untraceable off-line cash in wallet with observers. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 302–318. Springer-Verlag, Berlin Germany, 1993.

[5] Ross Anderson and Markus Kuhn. Tamper resistance – a cautionary note. In *Second USENIX Workshop on Electronic Commerce* [48], pages 1–11.

[6] Dan Boneh, Richard A. DeMillo, and Richard J. Lipton. On the importance of checking cryptographic protocols for faults. In *Eurocrypt '97*, volume 1233 of *Lecture Notes in Computer Science*, pages 37–51. Springer-Verlag, Berlin Germany, 1997.

[7] Paul Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Wiener [49], pages 399–397.

[8] Suresh Chari, Charanjit Jutla, Josyula R. Rao, and Pankaj Rohatgi. A cautionary note regarding evaluation of AES candidates on smart-cards. In *Second Advanced Encryption Standard (AES) Candidate Conference*, Rome, Italy, March 1999. `http://csrc.nist.gov/encryption/aes/round1/conf2/aes2conf.htm`.

[9] Suresh Chari, Charanjit Jutla, Josyula R. Rao, and Pankaj Rohatgi. Towards sound approaches to counteract power-analysis attacks. In Wiener [49], pages 398–412.

[10] Andreas Pfitzmann, Birgit Pfitzmann, Matthias Schunter, and Michael Waidner. Trusting mobile user devices and security modules. *IEEE Computer*, 30(2):61–68, February 1997.

[11] National Institute of Standards Technology. Data Encryption Standard (DES). Draft Federal Information Processing Standards Publication 46-3 (FIPS PUB 46-3), January 1999.

[12] Ross Anderson. Why cryptosystems fail. *Communications of the ACM*, 37(11):32–41, November 1994.

[13] Ron L. Rivest, A. Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Journal of the ACM*, 21(2):120–126, February 1978.

[14] David Chaum. Privacy protected payments: Unconditional payer and/or payee untraceability. In D. Chaum and I. Schaumueller-Bichl, editors, *Smartcard 2000*, pages 69–93. Elsevier Science Publisher North Holland, 1989.

[15] Mihir Bellare, Juan Garay, Ralf Hauser, Amir Herzberg, Hugo Krawczyk, Michael Steiner, Gene Tsudik, Els Van Herreweghen, and Michael Waidner. Design, implementation and deployment of a secure account-based electronic payment system. Research Report RZ 3137, IBM Research Division, June 1999. A revised version is to appear in the IEEE JSAC Special Issue on Network Security, 2000.

[16] Nathaniel S. Borenstein. Perils and pitfalls of practical cybercommerce. *Communications of the ACM*, 39(6):36–44, June 1996.

[17] D. Gifford, L. Stewart, C. Payne, and G. Treese. Payment switches for open networks. In *IEEE COMPCON '95*, San Francisco, March 1995. IEEE.

[18] Milton M. Anderson. The electronic check architecture. Technical Report Version 1.0.2, Financial Services Technology Consortium, September 1998.

[19] Jeff Kravitz (ed.). FSML – Financial service markup language. Technical Report Version 1.17.1, Financial Services Technology Consortium, October 1998.

[20] Mastercard and Visa. *SET Secure Electronic Transactions Protocol*, version 1.0 edition, May 1997. Book One: Business Specifications, Book Two: Technical Specification, Book Three: Formal Protocol Definition. Available from `http://www.setco.org/set_specifications.html`.

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

[22] Gennady Medvinsky and B. Clifford Neuman. NetCash: A design for practical electronic currency on the internet. In Victoria Ashby, editor, *1st ACM Conference on Computer and Communications Security*, pages 102–106, Fairfax, Virginia, November 1993. ACM Press.

[23] Holger Bürk and Andreas Pfitzmann. Digital payment systems enabling security and unobservability. *Computers & Security*, 8(5):399–416, August 1989.

[24] S. Low, N. Maxemchuk, and Sanjoy Paul. Anonymous credit cards. In Jacques Stern, editor, *2nd ACM Conference on Computer and Communications Security*, pages 108–117, Fairfax, Virginia, November 1994. ACM Press.

[25] Benjamin Cox, J. D. Tygar, and Marvin Sirbu. NetBill security and transaction protocol. In *First USENIX Workshop on Electronic Commerce*, New York, July 1995. USENIX.

[26] Torben P. Pedersen. Electronic payments of small amounts. In *Cambridge Workshop on Security Protocols* [50], pages 59–68.

[27] Steve Glassman, Mark Manasse, Martin Abadi, Paul Gauthier, and Patrick Sobalvarro. The MilliCent protocol for inexpensive electronic commerce. In *Fourth International Conference on the World-Wide Web*, MIT, Boston, December 1995.

[28] Ralf Hauser, Michael Steiner, and Michael Waidner. Micro-payments based on iKP. Research Report 2791 (# 89269), IBM Research, February 1996.

[29] A. Herzberg and H. Yochai. Mini-pay: Charging per click on the web. In *Sixth International Conference on the World-Wide Web*, Santa Clara, California, April 1997.

[30] Birgit Pfitzmann and Michael Waidner. Strong loss tolerance of electronic coin systems. *ACM Transactions on Computer Systems*, 15(2):194–213, May 1997.

[31] Markus Jakobsson and Moti Yung. Revokable and versatile electronic money. In Clifford Neuman, editor, *3rd ACM Conference on Computer and Communications Security*, pages 76–87, New Delhi, India, March 1996. ACM Press.

[32] Yair Frankel, Yiannis Tsiounis, and Moti Yung. "Indirect Discourse Proofs": achieving efficient fair off-line e-cash. Asiacrypt '96, LNCS 1163, Springer-Verlag, Berlin 1996, 286-300.

[33] Jan Camenisch, Ueli Maurer, and Markus Stadler. Digital payment systems with passive anonymity-revoking trustees. ESORICS '96 (4th European Symposium on Research in Computer Security), Rome, LNCS 1146, Springer-Verlag, Berlin 1996, 33-43.

[34] Sebastiaan von Solms and David Naccache. On blind signatures and perfect crimes. Computers & Security 11/6 (1992) 581-583.

[35] Holger Petersen and Guillaume Poupard. Efficient scalable fair cash with off-line extortion prevention. 1st International Conference on Information and Communications Security (ICICS), LNCS 1334, Springer-Verlag, Berlin 1997, 463-477.

[36] Ronald L. Rivest and Adi Shamir. PayWord and MicroMint: Two simple micropayment schemes. In *Cambridge Workshop on Security Protocols* [50], pages 69 – 88.

[37] Ross Anderson, Harry Manifavas, and Chris Sutherland. NetCard – a practical electronic cash system. In *Cambridge Workshop on Security Protocols* [50].

[38] Matthias Schunter. Fair exchange: A new paradigm for e-commerce. In Lacoste et al. [51].

[39] Alireza Bahreman. Generic electronic payment services: Framework and functional specification. In *Second USENIX Workshop on Electronic Commerce* [48], pages 87–103.

[40] Steven P. Ketchpel, Hector Garcia-Molina, Andreas Paepcke, Scott Hassan, and Steve Cousins. U-PAI: A universal payment application interface. In *Second USENIX Workshop on Electronic Commerce* [48], pages 105–121.

[41] Sun Microsystems. The Java Wallet (TM) architecture white paper, March 1998. Available from `http://java.sun.com/products/commerce/docs/whitepapers/arch/architecture.ps`.

[42] J. L. Abad-Peiro, N. Asokan, M. Steiner, and M. Waidner. Designing a generic payment service. Research Report RZ 2891 (# 90839), IBM Research, December 1996. A later version is also available [43].

[43] J. L. Abad-Peiro, N. Asokan, Michael Steiner, and Michael Waidner. Designing a generic payment service. *IBM Systems Journal*, 37(1):72–88, January 1998.

[44] N. Asokan and Michael Steiner. The payment framework. In Lacoste et al. [51].

[45] B. Pfitzmann and M. Waidner. Properties of payment systems - general definition sketch and classification. Research Report RZ 2823 (#90126), IBM Research, May 1996.

[46] N. Asokan, Els Van Herreweghen, and Michael Steiner. Towards a framework for handling disputes in payment systems. In *Third USENIX Workshop on Electronic Commerce*, pages 187–202, Boston, Mass., September 1998. USENIX. Available from `http://www.zurich.ibm.com/Technology/Security/publications/1998/AvHS98b%.ps.gz`.

[47] Paul Wallich. How to steal millions in chump change. *Scientific American*, 281(2):21, August 1999. More information on this fraud case at `http://www.labmed.umn.edu/~john/ccfraud.html`.

[48] USENIX. *Second USENIX Workshop on Electronic Commerce*, Oakland, California, November 1996.

[49] Michael Wiener, editor. *Advances in Cryptology – CRYPTO '99*, number 1666 in Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, August 1999.

[50] *Cambridge Workshop on Security Protocols*, volume 1189 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin Germany, April 1996.

[51] Gerard Lacoste, Birgit Pfitzmann, Michael Steiner, and Michael Waidner, editors. *SEMPER – Secure Electronic Marketplace for Europe*. Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, (to be published).

[52] Bruce Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, Inc, 1996.

[53] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. CRC Press series on discrete mathematics and its applications. CRC Press, 1997. ISBN 0-8493-8523-7.

[54] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. HMAC: Keyed-hashing for message authentication. Internet RFC 2104, Feb 1997.

[55] NIST National Institute of Standards and Technology (Computer Systems Laboratory). Secure hash standard. Federal Information Processing Standards Publication FIPS PUB 180-1, April 1995.

[56] N. Asokan, Hervé Debar, Michael Steiner, and Michael Waidner. Authenticating public terminals. *Computer Networks*, 31(8):861–870, May 1999.

A Basic Concepts in Cryptography and Security

Cryptographic techniques are essential tools in securing payment protocols over open, insecure networks. Here we outline some relevant basic concepts. Additional information on cryptographic techniques can be found in excellent reference works on the subject [52, 53].

A.1 Message authentication

To authenticate a message is to prove the identity of its originator to its recipient. Authentication can be achieved by using shared-key or public-key cryptography.

A.1.1 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* (MAC). The MAC function is such that it is infeasible to compute a valid MAC for a new messagewithout knowing the shared secret. This means that we can detect any modification to a message by an attacker.HMAC [54] has become the standard way to construct MAC functions and is based on a suitable one-way hash function such as SHA-1 [55].

Shared secrets that are short – like a six-digit personal identification number (PIN) – are inherently susceptible to various kinds of attacks. 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 a wallet) that performs the message authentication using secure cryptographic mechanisms, such as digital signatures (see next Section).

A.1.2 Public-key cryptography

Authentication using public-key cryptography is based on digital signatures. A digital signature system consists of a signature algorithm and a verification algorithm. In addition, each entity that needs to sign has a matching 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 signing 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 all verifiers. To sign a message, the prover executes the signature algorithm with the message and his signing key as input. Thus, the signature can be generated only by the entity that knows the signing key. Given a digital signature and a certificate for its verification key, a verifier can authenticate the message by executing the matching signature verification algorithm. Any verifier who knows the corresponding public verification key can verify the authenticity of a signature; once a signature has been verified, the verifier can be sure that the signer cannot repudiate having signed the message. This property is known as nonrepudiation of origin. Authentication of messages using MACs does not provide nonrepudiation of origin for the message, whereas authentication using digital signatures does.

A.2 Attacks

Electronic payment protocols can be attacked at two levels: The protocol itself or the underlying cryptosystem.

A.2.1 Protocol-level attacks

Protocol attacks exploit weaknesses in the design and/or implementation of the high-level payment system. Even if the underlying cryptographic techniques are secure, their inappropriate use may open up vulnerabilities that an attacker can exploit.

Freshness and replay. A protocol may be attacked by replaying some messages from a previous legitimate run. The standard countermeasure is to guarantee the freshness of messages in a protocol. Freshness means that the message provably belongs to the current context only (that is, the current payment transaction) and 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 to be included in its reply. Because nonces are unpredictable and used in only one context, they ensure that a message cannot be reused in later transactions. Nonces do not require synchronization of clocks between the two parties. Consequently, they are very robust and popular in cryptographic protocol design. In general, nonces are an example of the challenge-response technique.

Fake-terminal. Protocols that perform authentication in only one direction are susceptible to the fake-terminal attack. For example, when a customer uses an ATM, the bank and the machine check the authenticity of the customer using a PIN. The customer, however, cannot be sure whether the ATM is a genuine bank terminal or a fake one installed by an attacker for gathering PINs. Using a trusted personal device, such as a smartcard or electronic wallet, helps avoid this attack. See [56] for a more detailed discussion on the fake-terminal problem and potential solutions.

A.2.2 Cryptosystem attacks

Cryptosystem attacks exploit weaknesses in the underlying cryptographic building blocks used in the payment system.

Brute force attack. The straight-forward cryptosystem attack is the brute force attack of trying every possible key. The space from which cryptographic keys are chosen is necessarily finite. If this space is not large enough, a brute force attack becomes practical. Four-digit PIN codes have a total of 10,000 permutations in the key space. If a value X is known to be the result of applying a deterministic transformation to 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, and so on). It might then be possible to mount dictionary attacks. Instead of trying every possible key as in the brute force attack, the attacker will only try the keys in "dictionary" of likely words, phrases, or other strings of characters.

Cryptanalysis. More sophisticated attacks, called cryptanalysis, attempt to 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 (such as the difficulty of finding graph isomorphism), which might be solvable one day.