CRYPTOGRAPHY & SECURITY

A SEMINAR REPORT

Submitted by

VIBJAN KOLAPATI

in partial fulfillment of requirement of the Degree of

Bachelor of Technology (B.Tech)

IN

COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF ENGINEERING

COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

KOCHI - 682022

SEPTEMBER 2008

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
KOCHI-682022

Certificate

Certified that this is a bonafide record of the seminar entitled

CRYPTOGRAPHY & SECURITY

Presented by the following student

VIBJAN KOLAPATI

of the VII semester, Computer Science and Engineering in the year 2008 in partial fulfillment of the requirements in the award of Degree of Bachelor of Technology in Computer Science and Engineering of Cochin University of Science and Technology.

Mrs. Sheena Mathew
Seminar Guide

Dr. David Peter S.
Head of the Division

Date:

Acknowledgement

Many people have contributed to the success of this. Although a single sentence hardly suffices, I would like to thank Almighty God for blessing us with His grace. I extend my sincere and heart felt thanks to **Dr. David Peter, Head of Department,** Computer Science and Engineering, for providing us the right ambience for carrying out this work. I am profoundly indebted to my seminar guide, **Ms. Sheena Mathew** for innumerable acts of timely advice, encouragement and I sincerely express my gratitude to her.

I express my immense pleasure and thankfulness to all the teachers and staff of the Department of Computer Science and Engineering, CUSAT for their cooperation and support.

Last but not the least, I thank all others, and especially my classmates who in one way or another helped me in the successful completion of this work.

VIBJAN KOLAPATI

ABSTRACT

Electronic computers have evolved from exiguous experimental enterprises in the 1940s to prolific practical data processing systems in the 1980s. As we have come to rely on these systems to process and store data, we have also come to wonder about their ability to protect valuable data.

Data security is the science and study of methods of protecting data in computer and communication systems from unauthorized disclosure and modification. The goal of this seminar is to introduce the mathematical principles of data security and to show how these principles apply to ATM,Smart cards,e-commerce and other purposes.

Data security has evolved rapidly since 1975. Exciting developments in cryptography: public-key encryption, digital signatures, the Data Encryption Standard (DES), key safeguarding schemes, and key distribution protocols. We have developed techniques for verifying that programs do not leak confidential data, or transmit classified data to users with lower security clearances. We have come to a better understanding of the theoretical and practical limitations to security.

TABLE OF CONTENTS

\

LIST OF TABLES

LIST OF FIGURES

1.INTRODUCTION

Cryptography, art and science of preparing coded or protected communications intended to be intelligible only to the person possessing a key. Cryptography (Greek *kryptos,* "secret"; *graphos,* "writing") refers both to the process or skill of communicating in or deciphering secret writings (codes, or ciphers) and to the use of codes to convert computerized data so that only a specific recipient will be able to read it using a key (*see* Encryption). Cryptographers call an original communication the cleartext or plaintext. Once the original communication has been scrambled or enciphered, the result is known as the ciphertext or cryptogram. The enciphering process usually involves an algorithm and a key. An encryption algorithm is a particular method of scrambling—a computer program or a written set of instructions. The key specifies the actual scrambling process. The original communication may be a written or broadcast message or a set of digital data.

In its broadest sense, cryptography includes the use of concealed messages, ciphers, and codes. Concealed messages, such as those hidden in otherwise innocent text and those written in invisible ink, depend for their success on being unsuspected. Once they are discovered, they frequently are easy to decipher. Codes, in which predetermined words, numbers, or symbols represent words and phrases, are usually impossible to read without the key codebook. Cryptography also includes the use of computerized encryption to protect transmissions of data and messages.

Today most communication leaves some kind of recorded trail. For example, communications over telephone lines, including faxes and e-mail messages, produce a record of the telephone number called and the time it was called. Financial transactions, medical histories, choices of rental movies, and even food choices may be tracked by credit card receipts or insurance records. Every time a person uses the telephone or a credit card, the telephone company or financial institution keeps a record of the number called or the transaction amount, location, and date. In the future, as telephone networks become digital, even the actual conversations may be recorded and stored. All of this amounts to a great privacy. The ability to encrypt data, communications, and other information gives individuals the power to restore personal privacy.

Cryptography is important for more than just privacy, however. Cryptography protects the world's banking systems as well. Many banks and other financial institutions conduct their business over open networks, such as the Internet. Without the ability to protect bank transactions and communications, criminals could interfere with the transactions and steal money without a trace.

1.1 TYPES OF CRYPTOGRAPHY

There are many types of cryptography, including codes, steganography (hidden or secret writing), and ciphers. Codes rely on codebooks. Steganography relies on different ways to hide or disguise writing. Ciphers include both computer-generated ciphers and those created by encryption methods. The different types of ciphers depend on alphabetical, numerical, computer-based, or other scrambling methods.

1.1.1 Codes and Codebooks

A well-constructed code can represent phrases and entire sentences with symbols, such as five-letter groups, and is often used more for economy than for secrecy. A properly constructed code can give a high degree of security, but the difficulty of printing and distributing codebooks—books of known codes—under conditions of absolute secrecy limits their use to places in which the books can be effectively guarded. In addition, the more a codebook is used, the less secure it becomes.

Imagine a codebook with two columns. In the first column is a list of all the words that a military commander could possibly need to use to communicate. For example, it contains all the possible geographic areas in a region, all possible times, and all military terms. In the other column is a list of plain words. To create a coded message, the encoder writes down the actual message. He then substitutes words in the codebook by finding matches in the second column for the words in the message and using the new words instead. For example, suppose the message is *Attack the hill at dawn* and the codebook contains the following word pairs: attack = bear, the = juice, hill = orange, at = calendar, and dawn = open. The encoded message would read *Bear juice orange calendar open*.

If the coded message fell into enemy hands, the enemy would know it was in code, but without the codebook the enemy would have no way to decrypt the message. Codebooks lose some of their value over time, however. For example, if the coded message fell into enemy hands and the next day the hill was attacked at dawn, the enemy could link the event to the coded message. If another message containing the word *orange* were captured, and the following day, something else happened on the hill, the enemy could assume that orange = hill is in the codebook. Over time, the enemy could put together more and more code word pairs, and eventually crack the code. For this reason, it is common to change codes often.

1.1.2 Steganography:

Steganography is a method of hiding the existence of a message using tools such as invisible ink, microscopic writing, or hiding code words within sentences of a message (such as making every fifth word in a text part of the message). Cryptographers may apply steganography to electronic communications. This application is called transmission security.

Steganography, or secret writing, seems to have originated almost as early as writing itself did. Even in ancient Egypt, where writing itself was a mystery to the average person, two distinct forms of writing were used. Hieratic or sacred writing was used for secret communication by the priests, and demotic writing was used by other literate people. The ancient Greeks and Romans, as well as other civilizations that flourished at around the same time, used forms of steganography. The invention of the first shorthand system was presumably intended as a form of secret writing. Shorthand first came into wide use in ancient Rome, with *notae Tironianae* ('Tironian notes'), a system invented by Marcus Tullius Tiro in 63 BC.

1.1.3 Ciphers

Ease of use makes ciphers popular. There are two general types of ciphers. Substitution ciphers require a cipher alphabet to replace plaintext with other letters or symbols. Transposition ciphers use the shuffling of letters in a word to make the word incomprehensible.

Ciphers are the secret codes used to encrypt plaintext messages. Ciphers of various types have been devised, but all of them are either substitution or transposition ciphers. Computer ciphers are ciphers that are used for digital messages. Computer ciphers differ from ordinary substitution and transposition ciphers in that a computer application performs the encryption of data. The term *cryptography* is sometimes restricted to the use of ciphers or to methods involving the substitution of other letters or symbols for the original letters of a message.

1.2 Computer Ciphers & Encryption

Government agencies, banks, and many corporations now routinely send a great deal of confidential information from one computer to another. Such data are usually transmitted via telephone lines or other nonprivate channels, such as the Internet. Continuing development of secure computer systems and networks will ensure that confidential information can be securely transferred across computer networks.

In 1978 three American computer scientists, Ronald L. Rivest, Adi Shamir, and Leonard Adleman, who later founded the company RSA Data Security, created the Rivest-Shamir-Adleman (RSA) system. The RSA system uses two large prime numbers, p and q, multiplied to form a composite, n. The formula $n = pq$, capitalizes on the very difficult problem of factoring prime numbers.

As more and more information is transferred over computer networks, computer scientists continue to develop more secure, complex algorithms. In 1997 the NIST began coordinating development of a replacement for DES called Advanced Encryption Standard (AES). AES will use a more complex algorithm, based on a 128-bit encryption standard instead of the 64-bit standard of DES. This 128-bit algorithm will make AES impossible to decrypt with current technology.

Another encryption system based on 128-bit segments is called International Data Encryption Algorithm, or IDEA. The Swiss Federal Institute of Technology developed the IDEA standard in the 1990s. Computer scientists have also proposed alternatives such as public-key cryptosystems (PKCs), which use two types of keys, a public key and a private key. The public key encrypts data, and a corresponding private key decrypts it. The user gives the public key out to other users, and they can use the public key for encrypting messages to be sent to the user. The user keeps the private key secret and uses it to decrypt received messages.

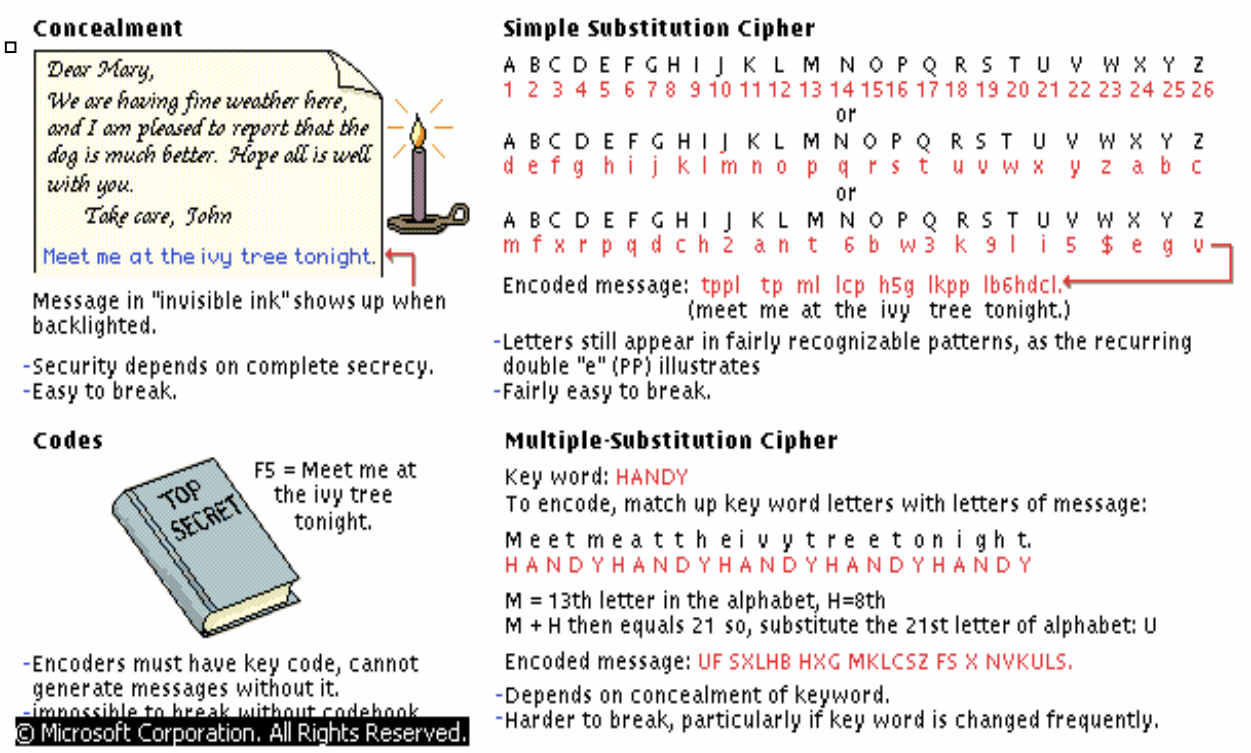
1.3 Cryptanalysis

Cryptanalysis is the art of analyzing ciphertext to extract the plaintext or the key. In other words, cryptanalysis is the opposite of cryptography. It is the breaking of ciphers. Understanding the process of code breaking is very important when designing any encryption system. The science of cryptography has kept up with the technological explosion of the last half of the 20th century. Current systems require very powerful computer systems to encrypt and decrypt data. While cryptanalysis has improved as well, some systems may exist that are unbreakable by today's standards.

Today's cryptanalysis is measured by the number and speed of computers available to the code breaker. Some cryptographers believe that the National Security Agency (NSA) of the United States has enormous, extremely powerful computers that are entirely devoted to cryptanalysis.

The substitution ciphers described above are easy to break. Before computers were available, expert cryptanalysts would look at ciphertext and make guesses as to which letters were substituted for which other letters. Early cryptanalysis techniques included computing the frequency with which letters occur in the language that is being intercepted. For example, in the English language, the letters *e, s, t, a, m,* and *n* occur much more frequently than do *q, z, x, y,* and *w*. So, cryptanalysts look at the ciphertext for the most frequently occurring letters and assign them as candidates to be *e, s, t, a, m,* and *n*. Cryptanalysts also know that certain combinations of letters are more common in the English language than others are. For example, *q* and *u* occur together, and so do *t* and *h*. The frequency and combinations of letters help cryptanalysts build a table of possible solution letters. The more ciphertext that is available, the better the chances of breaking the code.

In modern cryptographic systems, too, the more ciphertext that is available to the code breaker, the better. For this reason, all systems require frequent changing of the key. Once the key is changed, no more ciphertext will be produced using the former key. Ciphertext that is produced using different keys—and frequently changed keys—makes the cryptanalyst's task of code breaking difficult.



Selected Ciphers and Codes

Secret messages may be hidden or disguised in many ways. Encrypting, or coding, a message means changing it from words everyone can see and understand into a special set or particular order of symbols known only to a few. Concealment is a simple kind of cryptography, because the message is written normally and merely hidden. Although they are hard to break, codes are also easy to use because words and symbols are predetermined. (The reason codes are so difficult to break is that there is no way to figure them out logically. There is no clear link between F5 and the message.) In substitution ciphers, messages are completely rewritten. A set of new letters or numbers is assigned to the alphabet (upper right) or the numerical value of letters may be used with a repeating key word (lower right).

Figure 1.1 Paradigm of cryptography

1.4 Security services

Security Requirements:

Confidentiality: Protection from disclosure to unauthorised persons,Integrity: Maintaining data consistency,Authentication: Assurance of identity of person or originator of data.Non-repudiation: Originator of communications can't deny it later,Availability: Legitimate users have access when they need it,Access control: Unauthorised users are kept out.
These are often combined: User authentication used for access control purposes, Non-repudiation combined with authentication.

1.5 Security Threats

Information disclosure/information leakage,Integrity violation,Masquerading,Denial of service Illegitimate use,Generic threat: Backdoors, trojan horses, insider attacks,Most Internet security problems are access control or authentication ones: Denial of service is also popular, but mostly an annoyance.

2.SECURITY MECHANISMS

Three basic building blocks are used:

Encryption is used to provide confidentiality, can provide authentication and integrity protection Digital signatures are used to provide authentication, integrity protection, and non-repudiation Checksums/hash algorithms are used to provide integrity protection, can provide authentication One or more security mechanisms are combined to provide a security service.

2.1 ENCRYPTION

Encryption and decryption

Data that can be read and understood without any special measures is called *plaintext* or *cleartext*. The method of disguising plaintext in such a way as to hide its substance is called *encryption.* Encrypting plaintext results in unreadable gibberish called *ciphertext*. You use encryption to ensure that information is hidden from anyone for whom it is not intended, even thosewho can see the encrypted data. The process of reverting ciphertext to its original plaintext is called *decryption*.

Figure 2-1 illustrates this process.

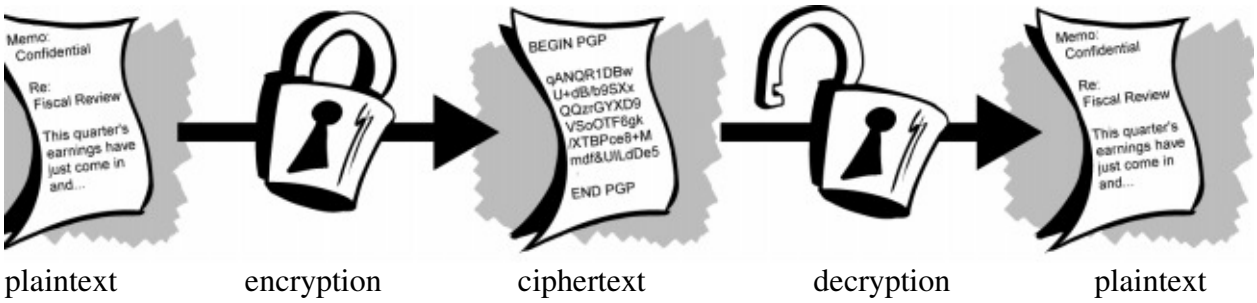
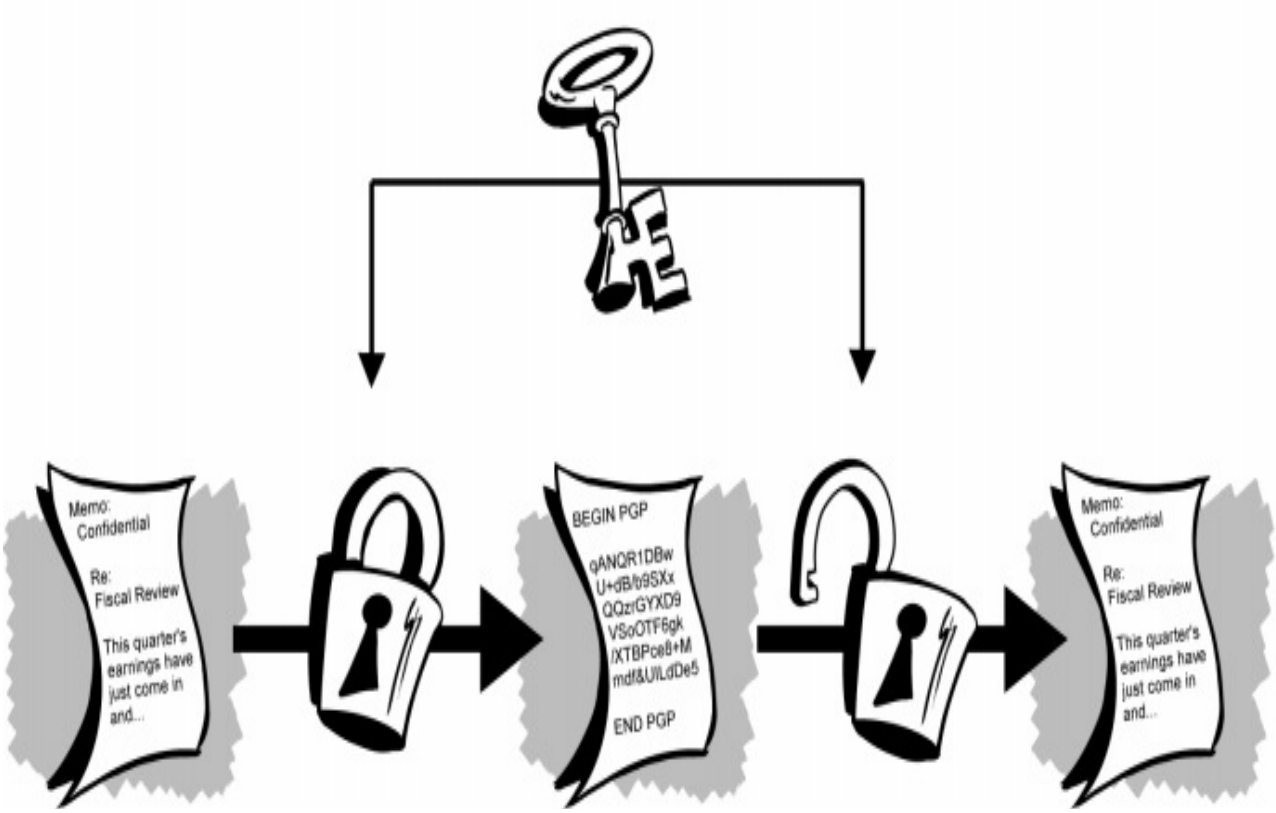


Figure 2-1. Encryption and decryption

In conventional cryptography, also called *secret-key* or *symmetric-key* encryption, one key is used both for encryption and decryption. The Data Encryption Standard (DES) is an example of a conventional cryptosystem that is widely employed by the Federal Government.

Figure 2-2 is an illustration of the conventional encryption process.



plaintext encryption ciphertext decryption plaintext

Figure 2-2. Conventional encryption

Public key cryptography is an asymmetric scheme that uses a *pair* of keys for encryption: a *public key*, which encrypts data, and a corresponding *private,* or
secret key for decryption. You publish your public key to the world while keeping your private key secret. Anyone with a copy of your public key can then encrypt information that only you can read. Even people you have never met.
It is computationally infeasible to deduce the private key from the public key.
Anyone who has a public key can encrypt information but cannot decrypt it. Only the person who has the corresponding private key can decrypt the Information.

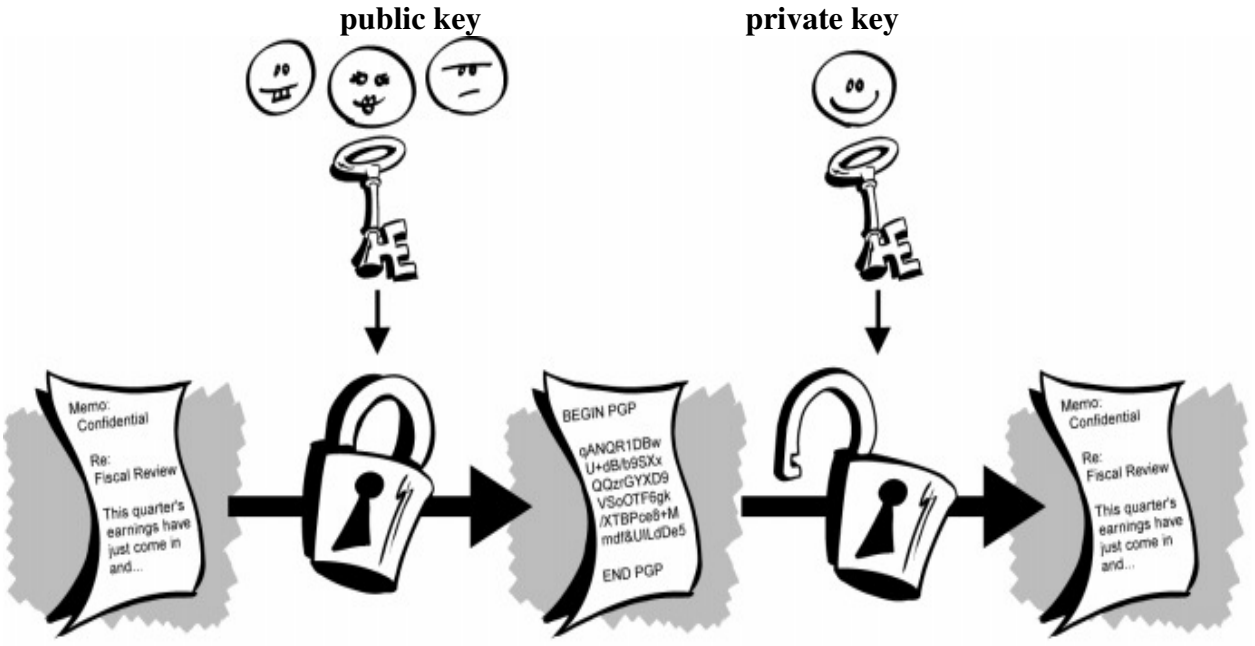
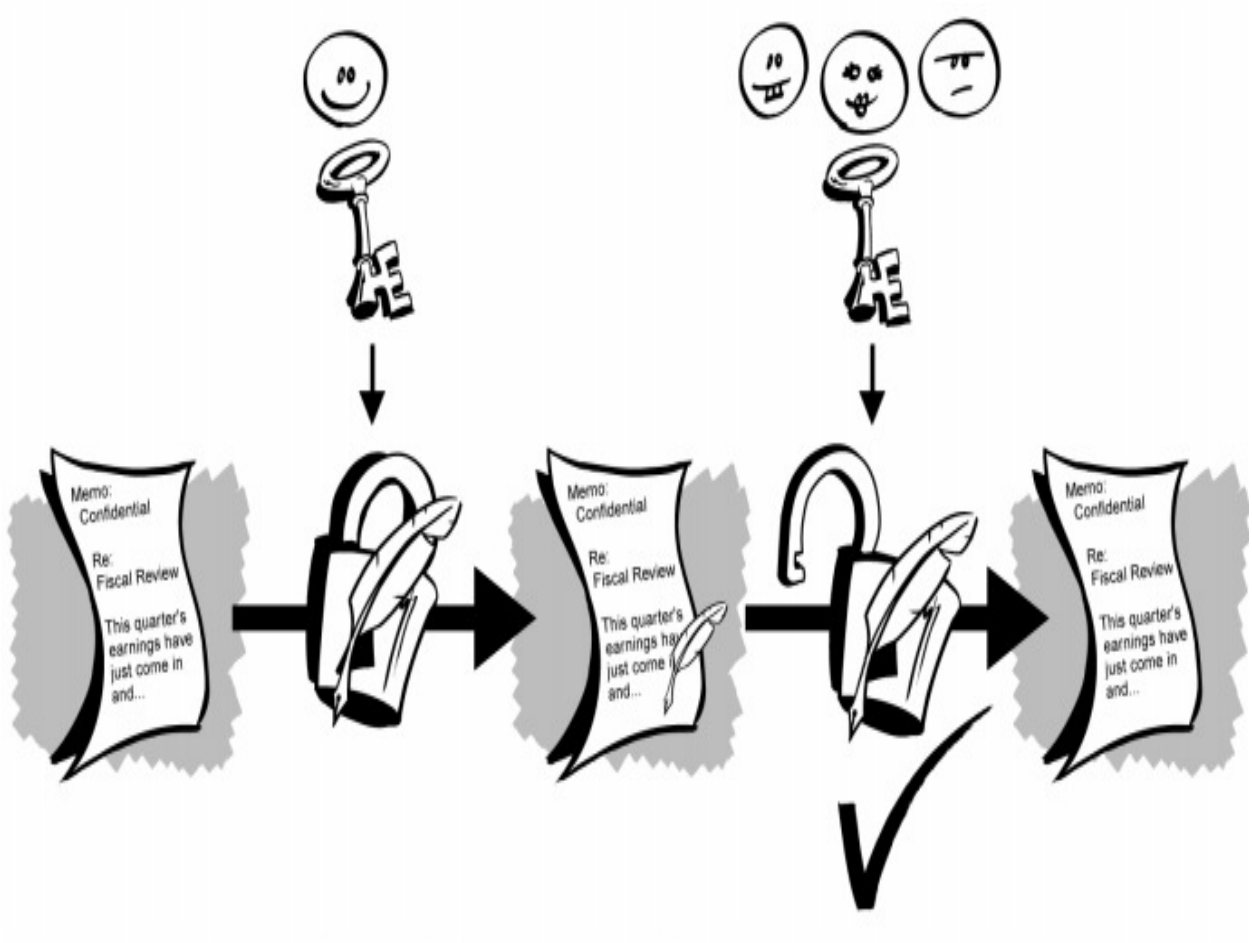


Figure2-3.Public key encryption

2.2 DIGITAL SIGNATURES

A major benefit of public key cryptography is that it provides a method for employing *digital signatures*. Digital signatures enable the recipient of information to verify the authenticity of the information's origin, and also verify that the information is intact. Thus, public key digital signatures provide *authentication* and data *integrity*. A digital signature also provides *nonrepudiation*, which means that it prevents the sender from claiming that he or she did not actually send the information. These features are every bit as fundamental to cryptography as privacy, if not more.A digital signature serves the same purpose as a handwritten signature.However, a handwritten signature is easy to counterfeit. A digital signature is superior to a handwritten signature in that it is nearly impossible to counterfeit, plus it attests to the contents of the information as well as to the identity of the signer.Some people tend to use signatures more than they use encryption. For example, you may not care if anyone knows that you just deposited $1000 in your account, but you do want to be darn sure it was the bank teller you were dealing with.

The basic manner in which digital signatures are created is illustrated in *Figure 2-4*. Instead of encrypting information using someone else's public key, you
encrypt it with your private key. If the information can be decrypted with your
public key, then it must have originated with you.



original text signing signed text verifying verified text

Figure 2-4. Simple digital signatures

2.3 HASH ALGORITHMS

The system described above has some problems. It is slow, and it produces an enormous volume of data—at least double the size of the original information. An improvement on the above scheme is the addition of a one-way *hash function* in the process. A one-way hash function takes variable-length input—in this case, a message of any length, even thousands or millions of bits—and produces a fixed-length output; say, 160-bits. The hash function ensures that, if the information is changed in any way—even by just one bit—an entirely different output value is produced.

PGP uses a cryptographically strong hash function on the plaintext the user is signing. This generates a fixed-length data item known as a *message digest.*
(Again, any change to the information results in a totally different digest.)

Then PGP uses the digest and the private key to create the "signature." PGP transmits the signature and the plaintext together. Upon receipt of the message, the recipient uses PGP to recompute the digest, thus verifying the signature. PGP can encrypt the plaintext or not; signing plaintext is useful if some of the recipients are not interested in or capable of verifying the signature.

As long as a secure hash function is used, there is no way to take someone's signature from one document and attach it to another, or to alter a signed message in any way. The slightest change in a signed document will cause the digital signature verification process to fail.

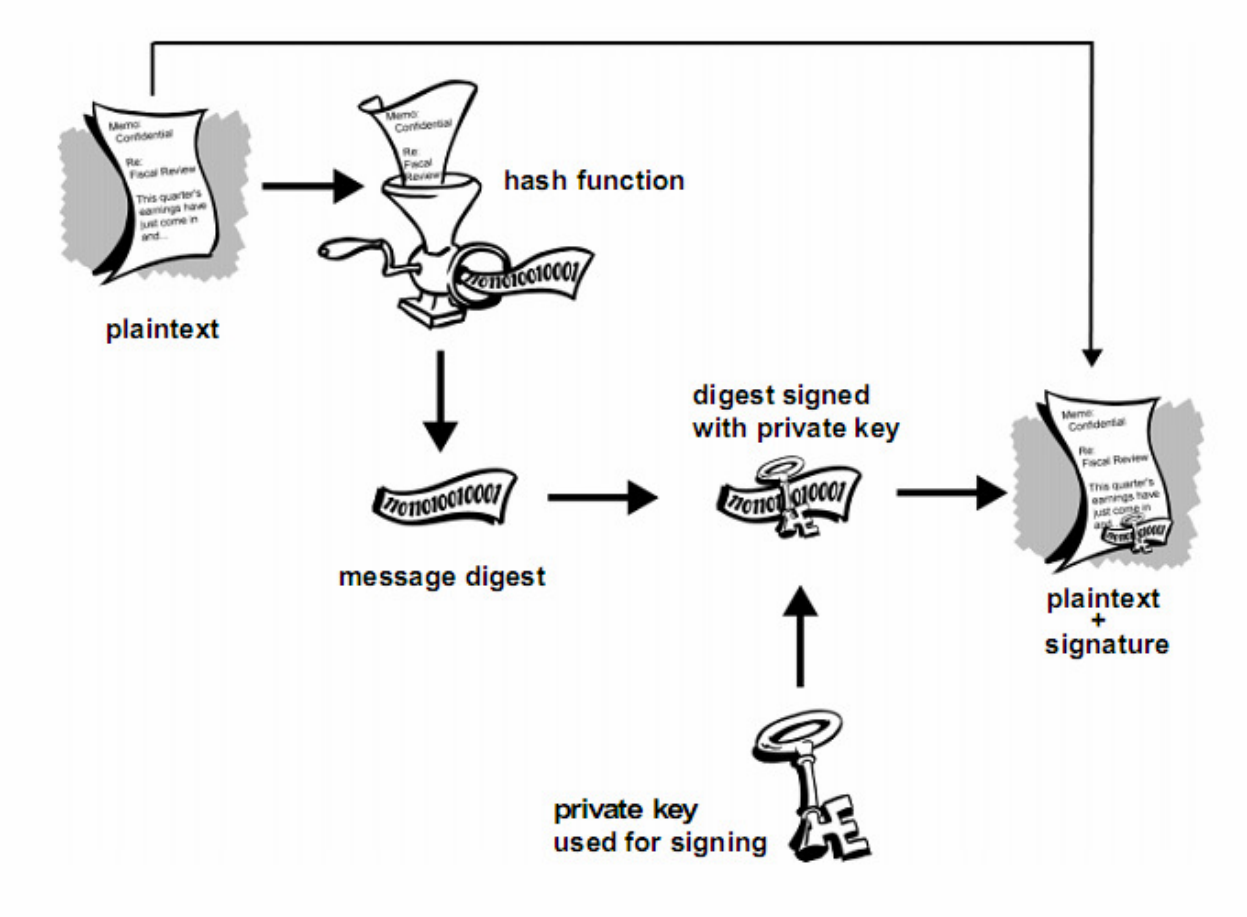


Figure 2-5. Secure Digital Signatures

A hash function h is

- A one-way hash function if it is computationally infeasible to determine the message m given the hash-of-message h(m).
- A collision-resistant hash function if given the hash-of-message h(m) it is computationally infeasible to determine any other message m* with the same hash value h(m)= h(m*).

A message digest is a hash function that derives a fixed-length hash value for every message in some message domain.

3. WATER MARKING

Watermarking is a method in computer security by which identifiers of sources or *copyright owners* of digital or analog signals are embedded into the respective signals themselves in order to keep track of where a signal comes from or who the copyright owners are. Thus, watermarking is for copy protection of electronic content, predominantly, digital content. The signal carrying the content before a watermark is embedded is sometimes called the *cover-signal*, and the piece of data carrying the copyright ID (and other optional information)called the *watermark*. The term "watermark" dates back to the 13th century when paper makers used traditional watermarks to differentiate their products from each other. Thus, traditional watermarks serve as authentication tags of print media, while electronic watermarks serve as copyright tags of the content itself.The demand for watermarking comes mainly from the movie and music industry, which is trying to limit the pirating of their digital audio, video and artwork. The characteristic security requirements on digital watermarking schemes, also called *electronic copyright marking schemes* (ECMS), are as follows:

Unobstrusiveness: Watermarks should not degrade the quality of the cover-signal in a way that annoys consumers when listening to or watching the content. The watermark may still be visible though, like, for example, the logo of a TV station that is continuously visible in one of the corners of the screen.

Robustness: Watermarks should be embedded into the content, in such a way that any signal transform of reasonable strength cannot remove the watermark. Examples of transformations are digital-to-analog conversion, compression
or decompression, or the addition of small amounts of noise. Hence, pirates trying to remove a watermark will not succeed unless they degrade the quality of the cover-signal so much that the result is of little or no commercial interest
any more.Note how watermarking is different from digital steganography, which was originally introduced as *invisible communication* by Gus Simmons.
In steganography, the sender hides a message in a cover signal in order to conceal the *existence* of a message being transferred. The main goal is imperceptability of the hidden message.Unobstrusiveness and robustness are no primary concerns in steganography. The cover signal may be degraded to a certain extent and no precautions are taken to avoid losing the hidden message
should the cover signal be manipulated. Watermarking and steganography are different topics in the area of *information hiding*.Watermarking is also different from authentication and non-repudiation . On the one hand, robustness is a stronger requirement than unforgeability, because an attacker against robustness is already successful if he destroys or removes a watermark, whereas a forger is only successful if he comes up with an authentication tag that verifies against the verification key of an existing sender. In another sense, robustness is a weaker requirement than nonrepudiation, because there are watermarking schemes where all decoders use the same secret that is used to create a watermark. Thus any decoder could be used to pick up a watermark from some cover signal *A* and embed it to another signal *B*, thus making the impression that the

copyright owner of signal *A* also owns a copyright for signal *B*. There are watermarking mechanisms, though, that depend not only on the sender ID, but also on the cover-signal itself, and thus can be used for authentication purposes.
Important classes of digital watermarking schemes are the following:
Blind watermarking (sometimes called *public watermarking*) means that a decoder can detect a watermark without a need to look at the cover signal. Only the sender's secret key is required in order to re-construct some random sequence that the sender used to construct the watermark in the first place. These types of schemes can be used easily in mass market electronic equipment or software. In some cases you may need extra information to help your decoder (in particular to synchronise its random sequence on the possibly distorted test signal).
Non-blind watermarking (sometimes called *private watermarking*) means that a decoder requires the sender's secret key and the original cover signal in order to detect the watermark in it.
Asymmetric watermarking (sometimes called *public-key watermarking*) means that the sender uses a private key to construct the watermark in a similar way as a digital signature is created by using a private key, and any decoder who has access to the corresponding public key can detect and recognize the respective watermark (see also public key cryptography). The public key would neither allow to reconstruct the sender's private key (under certain complexity theoretic assumptions), nor to forge a watermark, nor to remove an embedded watermark.No sufficiently robust and secure asymmetric watermarking schemes are known .Prominent examples of electronic copyright
marking schemes are the following:
DVD (*Digital Video Disk* later reissued as *Digital Versatile Disk*).
This technique has been cryptanalyzed by Petitcolas et al.
SDMI (*Secure Digital Music Initiative*) .
This technique uses *echo hiding*, which was cryptanalyzedby Craver et al.

4.APPLICATIONS OF CRYPTOGRAPHY

There are so many applications in cryptography.some of them are:

The UNIX crypt(3) password protection;Automated teller machine transactions(ATM);Facility access cards;Smart cards;The Web's Secure Socket Layer protocol.

4.1 PROTECTING ATM TRANSACTIONS

In the 1960s, the banking industry considered offering certain electronic banking services to be performed at unattended banking terminals now referred to a automated teller machines (ATM).

The advantages of ATMs to the industry were significant:

- Customers would be able to perform certain banking transactions –deposits,withdrawals, account queries, account-to-account transfers – at any hour of the day.
- The bank would save on the considerable cost of processing checks; ATM terminals do not require medical benefits, they can be discharged at will.
- Electronic transactions would not require human supervision or intervention, permitting labor savings.

Two conflicting forces have influenced the design of electronic banking systems:

- Profitability – the desire by the bank to improve their bottom line;
- Security – the fear that individuals might learn how to penetrate the system, for example, to empty the ATM of cash in a largely invisible manner.

The considerable experience of banks with credit card transactions pointed to certain risks,including the use of counterfeit, lost, or stolen banking cards.
It was decided that a valid transaction would therefore require a customer to offer two bonafides in establishing a customer's identity:

- The banking card recording the user primary account number (PAN) on the card's third stripe;
- A separate identifying element.

Possession of an ATM card alone would not permit a customer to enter into a transaction.

4.1.1 Customer Authentication

If two quantities (Q1, Q2) are required for a customer to be authenticated to the system, possible choices of the second identifier Q2 might be
1. The customer's signature;
2. The customer's voiceprint;
3. The customer's fingerprint;
4. A password assigned to the customer.
Signatures and voiceprints vary under stress; indeed, handwriting and voiceprints
vary too much under stress to provide a reliable identification method and were too costly to implement in the 1960s. Fingerprints have some connotation of criminality

User_ ID	PAN	PIN
Janu	17894567	8974
Sonu	76654321	7860

Table 4.1 ATM PAN-PIN table

that might affect the marketability of ATM systems adversely. The least expensive
solution involves a password or personal identification number (PIN).
In an ATM transaction, a customer would

- Insert the banking card into the ATM's card reader; the primary account number (PAN= Q1) would be read;
- Enter the PIN (= Q2) at the ATM's keyboard.

To establish the authenticity of a customer, the system must have a mechanism for checking if the offered identifiers (Q1, Q2) are properly related. One possible authenticity protocol would reference a table maintained by the bank; the customer's account number (Q1) is recorded on the banking card and the user enters the PIN (Q2) at the banking terminal. The ATM terminal transmits the transaction request to the institution's computing system where (Q1, Q2) are checked by consulting a table stored somewhere in the system (Table 4.1) whenever authentication is required.
With this protocol, the PIN can be selected either by the customer or institution. The former possibility is attractive for marketing the system as it makes the customer feel that she or he is participating in the security of the system – and, if something goes wrong, the customer can be made to feel at least partially responsible!

There are possible threats to this authentication protocol, including the following:

1. The contents of the table might be compromised by a system's programmer; either information revealed, allowing Mr Green to pretend to be Mr Janu, or information added to the system corresponding to a fictitious user.
2. The communications between the ATM and the computing system might be wiretapped so that the signals corresponding to Q2 might be learned. The manufacture of counterfeit plastic banking cards or the alteration of stolen cards is not technically demanding.

There are remedies:

- The table might be enciphered and/or made write-protected to make it difficult even for a bank's system's programmer to read or modify its contents.
- Communications between the ATM and computing system might be enciphered to mitigate against wiretapping.

None of these is a complete solution; a portion of the enciphered table has to be logically "in the clear" when the authentication takes place, and during this time, it is exposed. On the other hand, the goal of an authentication protocol is not to make it impossible for an opponent to succeed, but to make it very difficult and not cost-effective. One way, is to limit the amount of cash that can be withdrawn in a 24-hour period.However, an additional feature was insisted upon by the banking community, which still further complicated the authentication problem.

4.1.2 On-Line/Off-Line Operation

The reliability of computing systems and the need for periodic system maintenance in the 1960s almost mandated the use of banking systems with two modes of operation:

- On-Line: identification of a user is performed remotely by the institution's computing system;
- Off-Line: identification of a user is performed locally at the banking ATM.

The banks intended to allow both modes of operation to coexist; during normal operation,the authentication would be performed at the institution's computing system. When the system was down for repair or maintenance, authentication would be carried out at the ATM.

The limited capability of ATMs and the fact that the list of customers might grow to several millions of customers4 implies that tables such as those described before cannot be stored locally at an ATM. There is a significant logistics problem; the list of customers changes each day. New customers are added and some are dropped. If, say, the 100 Bank of America ATMs in Los Angeles had to be updated daily, the cost advantage of ATMs would be lost. Moreover, banks wanted to cross state boundaries and form networks,like Interlink, the PLUS SYSTEM, and CIRRUS, which would require changes to be made nationally. It might be possible to make these changes by teleprocessing the table changes from the bank's computing system, but this exposes the system to wiretapping.

The solution was to make Q1 and Q2 functionally related,

$$Q2 = f(Q1);$$

and to check the relationship at the ATM during a customer transaction.
What kind of relationship f? Suppose Q1 and Q2 are decimal numbers and are related by

$$Q2 = f(Q1) = 1{,}000{,}000{,}000 - Q1$$

so that Janu's PIN is

$$Q2(\text{Janu}) = 1{,}000{,}000{,}000 - 17{,}894{,}567 = 999{,}982{,}105{,}433$$

This relationship f is unacceptable; first, it requires a customer to remember a 12-digit key.It is likely that the customer will write the PIN on the card instead of committing it to memory, thus negating the entire purpose of a separate identifying element. However,more importantly, the relationship f in the equation above is too simple. Customers might learn how Q1 and Q2 are related and this would enable them (or others) to counterfeit card–PIN pairs, which would be accepted by an ATM terminal during off-line operation.What is required is a "complicated" relationship f that cannot be easily discovered by the users.
The solution – encipherment!
Suppose Q2 is some encipherment of the account number (Q1) Q2= EK{Q1}. If the cryptographic algorithm EK{--- } is sufficiently strong, then knowledge of the pair (Q1,Q2) or even a large number of pairs
({Q1^ (i), Q2^ (i)) : $1 \leq i \leq N$} might not permit a customer easily to deduce the secret key K.
To authenticate a customer, the ATM must check if the relationship Q2 =EK{Q1}is satisfied. This means that the authentication key K must reside at each ATM. This poses a risk and the bank must be careful to safeguard revealing the key.Each ATM contains a high-security module (HSM), a tamper-resistant coprocessor that performs the PIN-validation; the ATM-key resides securely in what is believed to be the tamper-proof HSM.
The IBM Corporation developed an ATM protocol for Lloyd's Banking, initially based on LUCIFER but later retrofitted to the DES algorithm. The authentication protocol used in the IBM LIBERTY banking system is a version of the protocol.
If the PAN(User_ID) is assigned by the bank and PIN(User_ID)=
EK {PAN(User_ID)} is calculated by the card-issuer, it follows that the customer is not able to idependently select the PIN(User_ID). A solution to permit the user to select a UPIN((User_ID)) was devised in 1957 by Chubb Integrated Systems, a British firm that marketed an early ATM system. Chubb introduced a PINOffset, which is magnetically recorded on the card. The PIN(User_ID), PINOffset(User_ID), and U-PIN(User_ID) in the IBM 3624 system are related by

$$U\_PIN(\text{User ID}) = \text{Left16}[EK\{PAN(\text{User ID})\}] + PINOffset(\text{User ID})$$

where Left + 16[. . .] denotes the leftmost 16 bits of . . . .
In an ATM Transaction,
1. A customer inserts the ATM card into the ATM terminal's card reader,
2. The user keys in U-PIN(User_ID),
3. The PAN(User_ID) and PINOffset(User_ID) are read from the ATM card,
and

4. The U-PIN(User_ID) = Left16[EK{PAN(User_ID)}] + PINOffset(User_ID) computation is made at the terminal and the validity of the relationship U-PIN(User_ID)=Left16[EK {PAN(User_ID)}] + PINOffset(User_ID) is checked.
One drawback of this scheme is that the 4-hex digit U-PIN(User_ID) may include 0,1,. . .,9,A,B,. . .,F and the characters A,B,. . .,F are not normally on the ATM keyboard.

To solve this problem, a decimalization table mapping the U-PIN(User_ID) into the decimal digits is introduced. The default table is presented in Table 4.2. The PIN verification test is performed at the ATM module on an HSM. The IBM "Common Cryptographic Architecture" is an application program interface (API) for HSM with syntax Encrypted_PIN_Verify(. . .), which returns a YES/NO value. In addition to the PAN, one of the inputs is the decimalization table.

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

TABLE 4.2 Standard Decimalization Table

4.2 ATM TRANSACTIONS

The most successful commercial application of cryptography has been in facilitating transactions involving ATMs. Originally intended to be used for transactions at a single banking institution, ATMs have evolved to provide truly international banking. The steps in an ATM transaction are:

1. PAN(User_ID) and PINOffset(User_ID) is read from the ATM card;

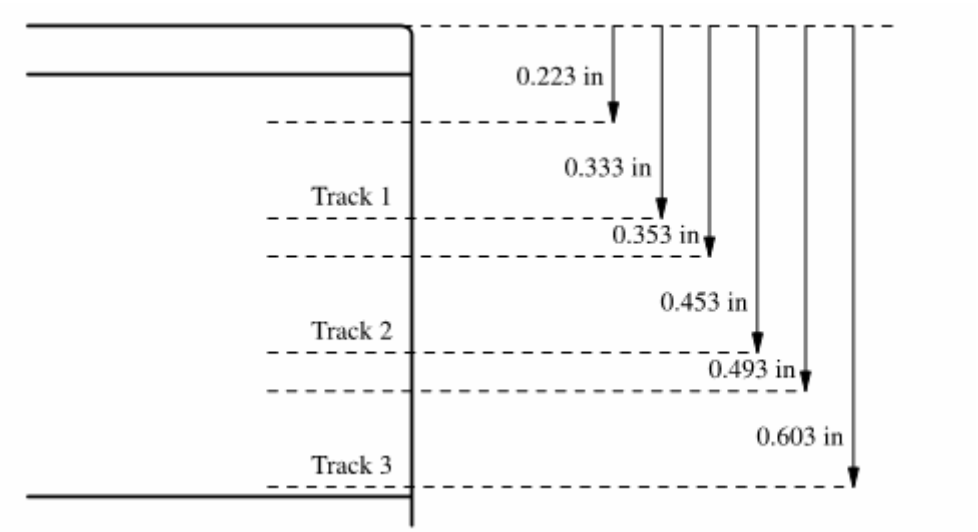


Figure 4.2.1 ANSI X9.1 Three-track credit/debit card format.

2. U-PIN(User_ID) is entered at the keyboard;

3. The transaction request containing this information is forwarded to the local ATM bank processing system.
4. The financial institution of the cardholder is identified (from card-data) and the transaction request is forwarded to it.
5. The cardholder's financial institution verifies the cardholder's ability to perform the transaction;

Account balance sufficient?
Credit-line? or
Stolen card?
and authorization to the local bank to carry out the transaction is forwarded to it.
The same sequence is followed when a credit card is offered as payment at a point-of-sale (POS) system.

4.3 SMART CARD

A smart card is a banking card containing an embedded processor; compared to a PC, the smart card's computational power and memory are significantly limited. The ISO standard 7810 specifies the physical details of the smart card designated as ID-1. The dimensions are 85.60 mm (L) x 53.98 mm (W) x 0.80 mm (T). Even the corner radius of 3.18 mm is specified.

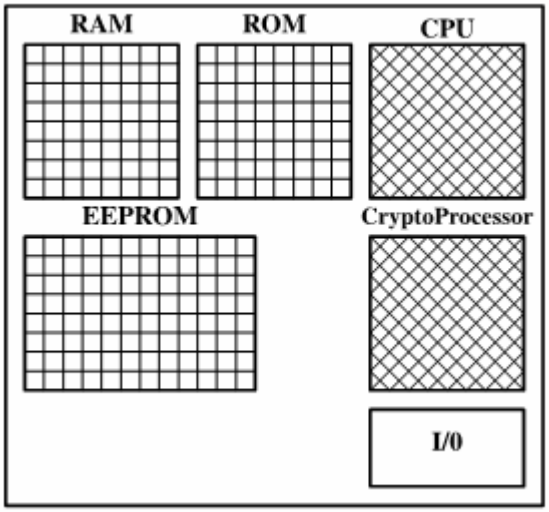


Fig 4.1 smart card memory

4.3.1 Smart Card Memory

Figure 4.1 illustrates the different types of memory contained on smart cards.

- ROM (read-only memory) – 6–24 Kbytes storing the operating system;
- RAM (random access memory) – 256–1024 bytes used as working memory; RAMis volatile, meaning that its contents are lost when power to the smart card is removed.
- EEPROM (electrically erasable programmable memory) – 1–16 Kbytes of memory that

  – can be written to externally,
  – can be erased externally by an electrical charge, and
  – retains its state when the power is removed.

4.3.2 External Interface of Smart Cards

Most smart cards require an external source of energy. One standard method to transfer data is to use a card acceptor device (CAD), which allows for the half-duplex exchange of data at the rate of 9600 b/s. The ISO standard 7816/3 provides either six or eight connection points for (external) power to the smart card. ISO 7816, Part 1 [ISO, 1998]

describes the locations and functions of the contacts on the smart card (Table 4.1).

Position		Function
C1	Vcc	Voltage supply
C2	RST	Reset
C3	CLK	Clock frequency
C4	RFU	Reserved for future use
C5	GND	Ground
C6	Vpp	External voltage
C7	I/O	Serial I/O
C8	RLU	Reserved for future use

Table 4.1 ISO 7816 Smart Card Contacts

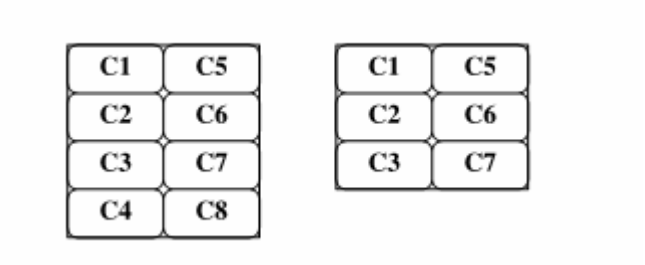


Figure 4.2 Smart card memory interface.

Some newer cards are contactless and exchange data over a small distance by inductive orcapacitive coupling. The smart card/terminal interface (Fig. 18.7) supports only half-duplex data transmission.

4.3.2 Smart Card Processing

A smart card typically contains an 8-bit microprocessor running at 5 MHz. The operating system is required to handle a small number of tasks, including:

- Half-duplex data transmission;
- Control and execution of instruction sequences;
- Running of management functions;
- Protecting access to data on the card;
- Memory and file management;
- Execution of cryptographic application programs (API).

As a smart card is not intended to be a general-purpose processor, it does not supply an interface for users.

Smart Card Functionalities

The cryptographic and related functions on a smart card include RSA with 512, 768, or 1024 bit keys; The digital signature algorithm (DHA); DES and triple-DES; Random number generation (RNG).

The Electronic Purse:

The advantages of a cashless society have been discussed for some time. One application of the smart card is the electronic wallet or electronic purse. The owner of the smart card deposits at his/her bank a sum. An entry is made (by the bank) on the smart card, which is used as cash.

When a purchase is made using the smart card, the amount is debited on the card. What a creative idea for the bank! Perhaps you might even receive interest on the money deposited at the bank, but certainly not at the annual rate of 18%/year.

Smart Card Vendors

Several different vendors have introduced smart cards, including

PC/SC: Microsoft for personal computers; Open Card: Java-based standard for POS (point-of-sale), laptops;.JavaCard: Proposed as a standard by Schlumberger.

4.3.3 The Role of the Smart Card

The smart card will provide proof of identity when a user is communicating with a remote server. Secure transactions involving a smart card will require cryptography. If the identification process is based on public-key cryptography, then

- The key will need to be stored in the EEPROM,. The smart card will need to read-protect the key, and
- The owner of the card will need to use a PIN to prove identity to the card.

Various physical attacks on the information stored in a card have been proposed. One is based on the observation that the contents of the EEPROM can be erased or modified by modifying the voltage applied to the card's contacts. Paul Kocher refers to variants of these attacks as differential power analysis (DPA. Other physical attacks involve heat and UV light.

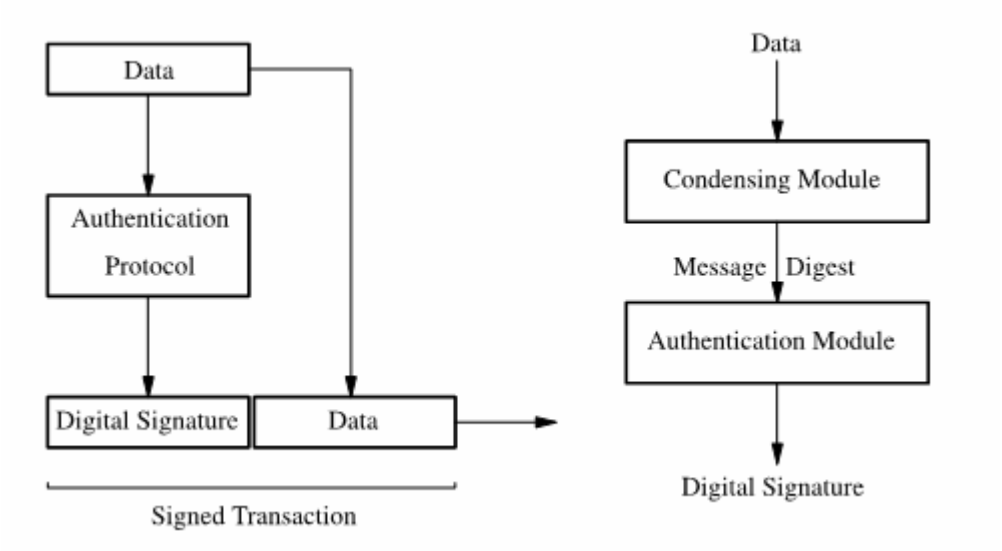


Figure 4.3 TRASEC protocol.

4.3.4 Protocols for Smart Cards

The two articles by Ph. van Heurck [1987, 1989] are among the earliest proposing the application of smart cards. C.I.R.I. is an association of banks in Belgium. These banks created TRASEC in 1987 to develop and maintain a system to develop and implement electronic TRAnsactions in a SECure manner.

Several authentication schemes are described; in one scheme, data are suffixed witha digital signature using the protocol in Figure 4.3.

4.4 CRYPTOGRAPHY APPLICATION BLOCK(CAB)

The Enterprise Library Cryptography Application Block simplifies how developers incorporate cryptographic functionality in their applications. Applications can use the application block for a variety of tasks, such as encrypting information, creating a hash from data, and comparing hash values to verify that data has not been altered.

The Cryptography Application Block has the following features:

It reduces the requirement to write boilerplate code to perform standard tasks; it does this by providing implementations that can be used to solve common application cryptography problems.

- It helps maintain consistent cryptography practices, both within an application and across the enterprise.
- It eases the learning curve for developers by using a consistent architectural model across the various areas of functionality that are provided.
- It provides implementations that you can use to solve common application cryptography problems.
- It is extensible; this means it supports custom implementations of cryptography providers.

4.4.1 Design of the CAB

The Cryptography Application Block includes support for the following features:

- Encryption algorithms
- Hashing algorithms
- Multiple cryptography providers
- Additional implementations of cryptography providers
- Key protection with DPAPI

Design Goals

The Cryptography Application Block was designed to achieve the following goals:

- Provide a simple and intuitive interface to the commonly required functionality.
- Encapsulate the logic that is used to perform the most common application cryptography tasks.
- Present a standard consistent model for common cryptography tasks, using common names for algorithms.
- Exert minimal or negligible performance impact compared to manually written cryptography code that accomplishes the same functionality.

Design Highlights
Figure 1 illustrates the design of the Cryptography Application Block.

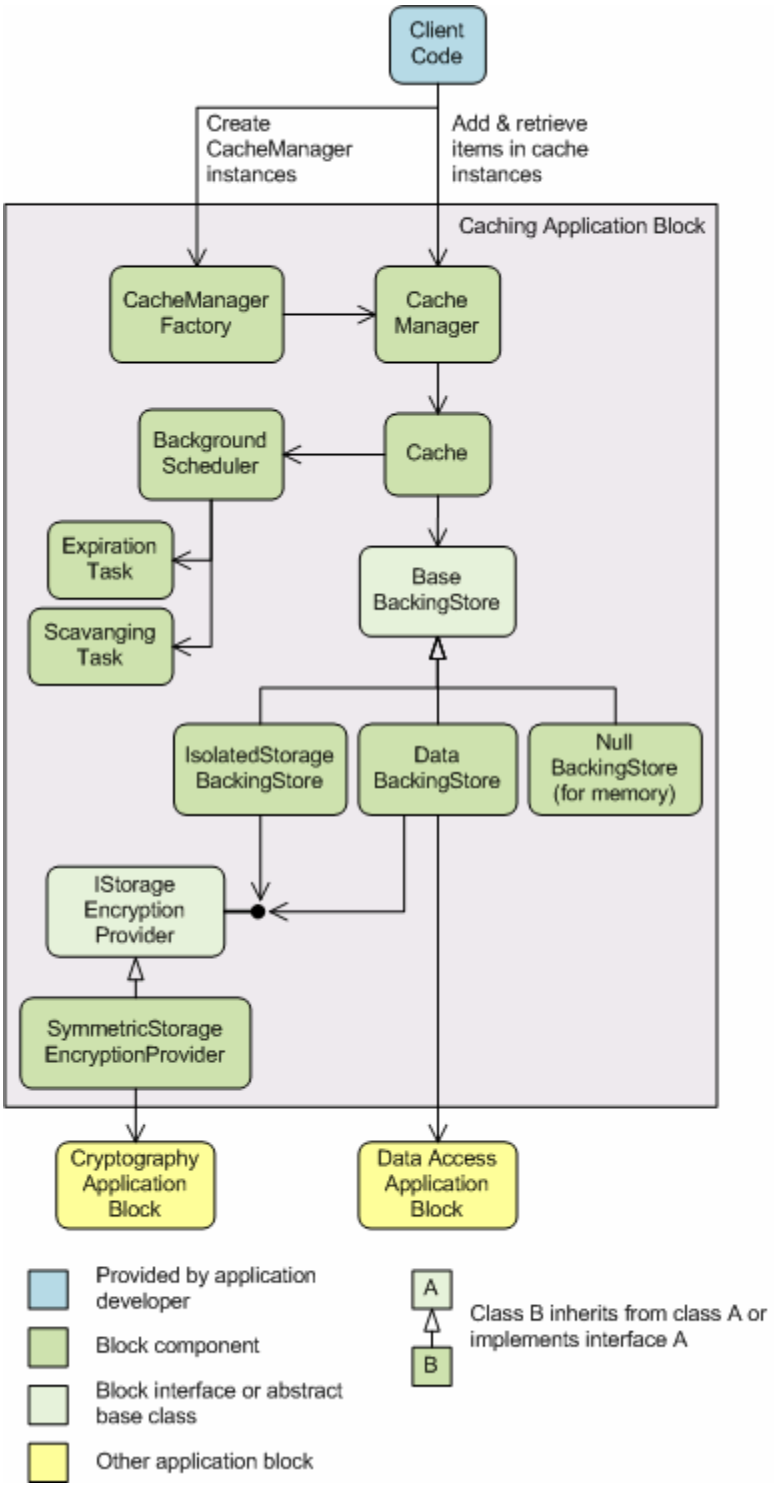


Figure 4.1 ***Design of the Cryptography Application Block***

The Cryptography Application Block separates decisions about how cryptographic functions are implemented from how an application uses them. The application block is designed so you change the behavior of a cryptography provider without changing the application code.

The **Cryptographer** class is a façade that mediates between the client code and the Cryptography Application Block's cryptographic functions. The client code calls static methods on the **Cryptographer**class to create hashes, compare hashes, encrypt data, and decrypt data. Unless you are using the Unity Integration approach, each static method instantiates a factory class and passes the configuration source to the factory class's constructor. The factory uses the configuration data to determine the type of the provider to create.

The **DpapiCryptographer** class uses DPAPI to encrypt and decrypt data. DPAPI uses logon credentials to encrypt data. The logon credentials can either be a user's logon credentials or the local computer's logon credentials. If you use the local computer's logon credentials, DPAPI allows all applications that run under those credentials to decrypt that data. To counteract this, you can use an additional secret to protect the data. This additional secret is named *entropy*. The **DpapiCryptographer** class has overloads of the **Encrypt** and **Decrypt** methods that accept an entropy value.

The **SymmetricCryptographer** class encapsulates provider implementations that derive from the abstract base class **SymmetricAlgorithm**, which is located in the .NET Framework's**System.Security.Cryptography** namespace. This means that you can use the**SymmetricCryptographer** class with any of the .NET Framework symmetric algorithms, such as the Rijndael symmetric encryption algorithm. The application block uses DPAPI to encrypt and decrypt the symmetric algorithm key.

4.4.2 Key Management Model

The configuration tools are used to select a cryptographic provider algorithm. If the algorithm requires a key, the configuration tools prompt to select an existing key or to create a new key. Create a new key, the configuration tools use the Cryptography Application Block to encrypt the key, and then store the encrypted key in its own text file. The application block uses DPAPI to encrypt the keys. When application executes, the application block uses DPAPI to decrypt the key, and then it uses the key to encrypt or decrypt your data.

The Cryptography Application Block's design-time component includes the Cryptographic Key Wizard. We can use this wizard to either create a new key or to use an existing key. We use an existing key by selecting a file that contains a key encrypted with DPAPI.

When we export a key, the configuration tools prompt you to supply a password to use to encrypt the key. The application block**KeyManager** class calls the **KeyReaderWriter** class to encrypt the key and create the file. The file contains a version number, salt value, and the encrypted key.

5.CHALLENGES

There are two kinds of cryptography in this world: cryptography that will stop your kid sister from reading your files, and cryptography that will stop major governments from reading your files.

For many years, this sort of cryptography was the exclusive domain of the military. The United States' National Security Agency (NSA), and its counterparts in the former Soviet Union, England, France, Israel, and elsewhere, have spent billions of dollars in the very serious game of securing their own communications while trying to break everyone else's. Private individuals, with far less expertise and budget, have been powerless During the last 20 years, public academic research in cryptography has exploded. While classical cryptography has been long used by ordinary citizens, computer cryptography was the exclusive domain of the world's militaries since World War II. Today, state–of–the–art computer cryptography is practiced outside the secured walls of the military agencies. The layperson can now employ security practices that can protect against the most powerful of adversaries—security that may protect against military agencies for years to come.

Do average people really need this kind of security? Yes. They may be planning a political campaign, discussing taxes, or having an illicit affair. They may be designing a new product, discussing a marketing strategy, or planning a hostile business takeover. Or they may be living in a country that does not respect the rights of privacy of its citizens. They may be doing something that they feel shouldn't be illegal, but is. For whatever reason, the data and communications are personal, private, and no one else's business,to protect their own privacy against these governments.

Clipper and Digital Telephony do not protect privacy; they force individuals to unconditionally trust that the government will respect their privacy.

6.CONCLUSION

Cryptology presents a difficulty not found in normal academic disciplines: the need for the proper interaction of cryptography and cryptanalysis. This arises out of the fact that in the absence of real communications requirements, it is easy to propose a system that appears unbreakable. Many academic designs are so complex that the would–be cryptanalyst doesn't know where to start; exposing flaws in these designs is far harder than designing them in the first place. The result is that the competitive process, which is one strong motivation in academic research, cannot take hold.

Many applications are useful in real-time and daily life that are implementd by cryptography through implicit or explicit concept of it.For example banking system,ATM cards,Smart cards,Magnetic strip technology,National Security Agency(NSA) to trace information through RADAR and with well equipped material,E-commerce,E-economics,business information,operating systems,databases and finally in System Protection.

In this way Cryptography has many roles and many application.

7.REFERENCES

1.Microsoft Encarta encyclopedia

2.Cryptography Encyclopedia

3.Cryptography and Data security by Dorothy Elizabeth,Amazon.com

4.Computer-security and Cryptography by Alan konheim,ACM portal.

5.Cryptography & Data security by Denning,Amazon.com

6. Bloom, Jefrey A., Ingemar J. Cox, Ton Kalker, Jean-Paul M.G. Linnartz, Matthew L. Miller, C., andBrendan S. Traw (1999). "Copy protection for DVDvideo." Proceedings of the IEEE,87 (7), 1267–1276.

7.Applied Cryptography and Data security by Prof. Christof Paar

8.www.msdn.microsft.com