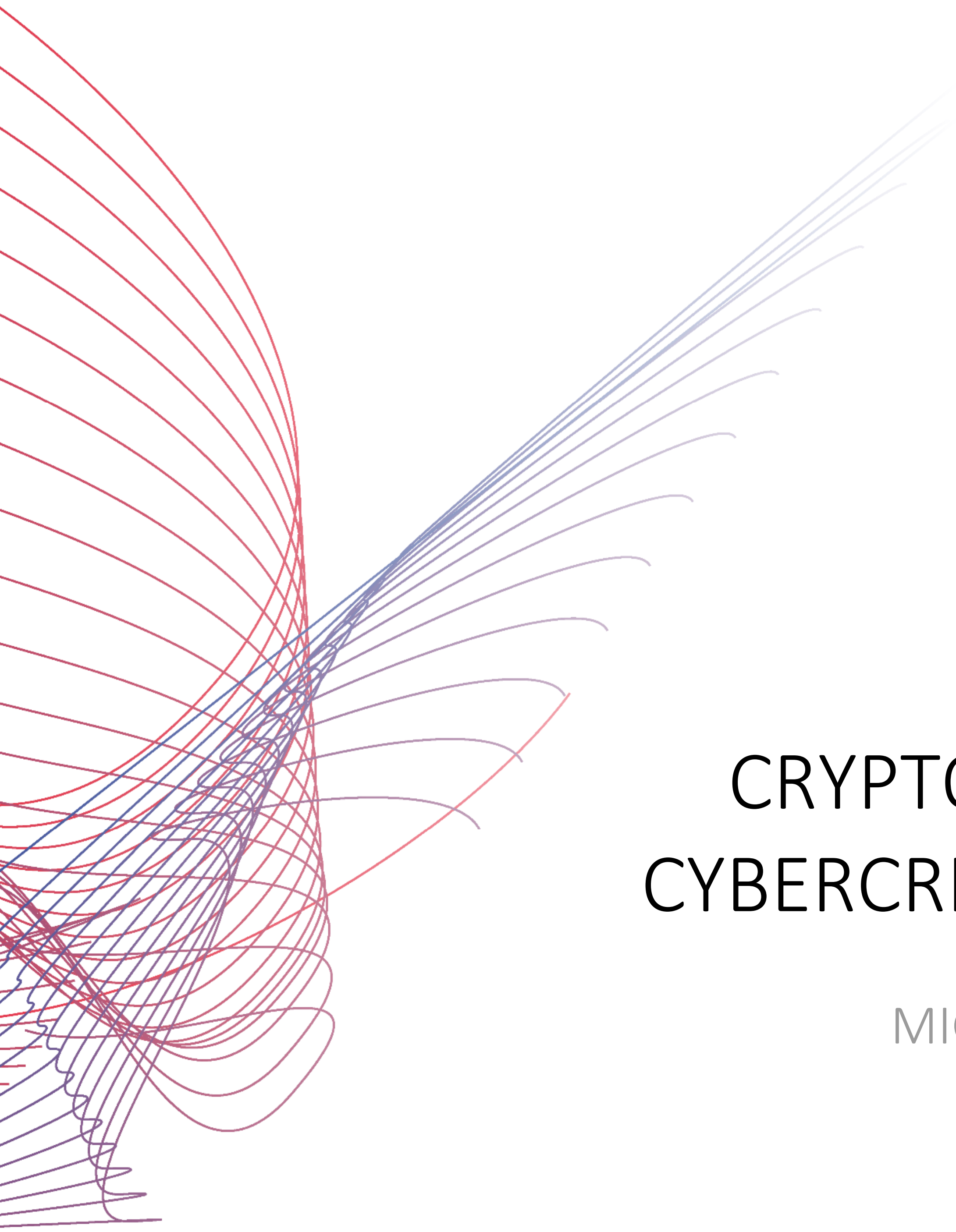
en
cs de
sk es

TECH
pedia



CRYPTOGRAPHY, CYBERCRIMINALITY

MIGUEL SORIANO

Title: Cryptography, cybercriminality
Author: Miguel Soriano
Published by: Czech Technical University of Prague
Faculty of electrical engineering
Contact address: Technicka 2, Prague 6, Czech Republic
Phone Number: +420 224352084
Print: (only electronic form)
Number of pages: 39
Edition: 1st Edition, 2017

ISBN 978-80-01-06202-9

TechPedia
European Virtual Learning Platform for
Electrical and Information Engineering

http://www.techpedia.eu

This project has been funded with support from the European Commission.
This publication reflects the views only of the author, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

Explanatory Notes

$E=m\cdot c^2$	Definition
⚑	Interesting
i	Note
🧪	Example
🧩	Summary
+	Advantage
−	Disadvantage

ANNOTATION

This module contains some necessary information for the basic orientation of students in the field of cryptography and cybercriminality.

OBJECTIVES

This module provides some basic information about cryptography and cybercriminality. The first part of the course is designed to introduce the students and help it become better acquainted with the possibilities that cryptography can offer to provide information security. Therefore, the course includes a brief overview of public-key and secret-key cryptography and algorithms. The second part is devoted to introduce the concept of cybercrime, and a classification of the attack techniques. Finally, some basic prevention tips are provided

LITERATURE

[1] Bruce Schneier: Applied Cryptography. John Kiley & Sons, Inc., New York, 1994

[2] William Stallings: Cryptography and Network Security. Principles and Practices. Prentice Hall, New Jersey, 2003

[3] Vesna Hassler: Security Fundamentals for E-Commerce. Artech House, Boston, 2001

[4] Rolf Oppliger: Internet and Intranet Security. Artech House, Boston, 2002

[5] Michael Goodrich, Roberto Tamassia: Introduction to Computer Security, 2010

[6] John R. Vacca: Computer and Information Security Handbook (Morgan Kaufmann Series in Computer Security), 2009

[7] Jason Andress: The Basics of Information Security: Understanding the Fundamentals of InfoSec in Theory and Practice, Elsevier, 2011

Index

1 Basics of cryptography

Cryptography is a strong mathematical tool for protecting information in computer systems. Many security applications are, in fact, based on the use of cryptography to encrypt and decrypt data. Thanks to cryptography, sensitive data can be safely transmitted through telecommunication networks without the threat of the information being intercepted and, subsequently, compromised. Encryption can be defined as the process of making information indecipherable and useless to all except those who are the intended recipients of such information. Decryption is converting data back to its original form.

This technique is used in everyday actions, such as making or receiving a call from a mobile phone, paying with a credit or debit card, withdrawing money from an ATM, logging on to a computer with a password, ... Cryptography enables to store sensitive information or transmit it across insecure networks (like the Internet) so that it cannot be read by anyone except the intended recipient. Cryptography has now become an industry standard for providing information security, trust, controlling access to resources, and electronic transactions. However, it is important to point out that cryptography by itself is not sufficient to deal with all threats to information security.

A cryptographic algorithm, or cipher, is simply just some sequences of processes for performing both an encryption, and the corresponding decryption. is a mathematical formula designed specifically to obscure the value and content of data. Most valuable cipher algorithms work in combination with one or several keys as part of the process. The same plaintext can be encrypted to different ciphertext when different keys are used. There must be no way to find the plaintext (clear data) if the key is unknown, except brute force, i.e. by trying all possible keys until the right one is found. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key

The number of possible keys must be so large that it is computationally infeasible to actually stage a successful brute force attack a reasonable length of time. Many cipher algorithms increase their protection by increasing the size of the keys they use. However, the larger the key, the more computing time is needed to encrypt and decrypt data. So it is important to choose a cipher algorithm that strikes a balance between your protection needs and the computational cost of protecting the data.

Modern Cryptographic algorithms can be divided by two criteria: by type of key used, and by the manner they operate on the data.

Regarding the type of key used, ciphers can be classified into:

a) Symmetric key or secret key algorithms. Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). The Advanced Encryption Standard (AES) is an example of a conventional cryptosystem widely employed.

b) Public key cryptography or asymmetric key cryptography is a scheme that uses a pair of keys for encryption: a public key, which encrypts data, and a corresponding private, or secret, key for decryption. Obviously, the two keys of the same pair are mathematically linked; nevertheless it is computationally infeasible to derive the private key from the public key. A user or entity publishes their public key while keeping their private key secret. 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.

The primary benefit of public key cryptography is that it allows people who have no preexisting security arrangement to exchange messages securely. The sender and the receiver do not need to share any secret keys via secure channels; all communications involve only public keys, and no private key is ever transmitted or shared.

Regarding the manner in which the algorithms operate on the data, ciphers can also be classified into,

a) Block Ciphers operate on fixed-length groups of bits, called blocks, with a keyed unvarying transformation. They divide the message into blocks and encrypt one block at a time. When the block cipher is considered secure then the resulting ciphertext of a single block is of course also secure - when analyzed independently. Nevertheless, when multiple messages are encrypted with the same key, then ECB is not secure, since identical message blocks will result in identical ciphertext block. Therefore an attacker could easily detect block repetitions in the message, so direct use of a block cipher is inadvisable. Different operations modes are used in order to avoid this problem.

b) Stream ciphers convert one symbol of plaintext directly into a symbol of ciphertext, They work by producing a keyed pseudorandom sequence that operates as a cryptographic keystream. This keystream is basically a stream of bits that is combined with plaintext to encrypt one bit or byte at a time, producing the ciphertext.

During the module, the following terminology will be used:

- Plaintext is the message that has to be transmitted to the recipient. It is also commonly referred to as cleartext.
- Ciphertext is the output that is generated after encrypting the plaintext.
- Encryption is the process of changing the content of a plaintext in such a manner that it hides the actual message.
- Decryption is the reverse of encryption; it is the process of retrieving the plaintext message from its encrypted form (ciphertext). This process converts ciphertext to plaintext.
- Key is a word, number, or string that is used to encrypt the plaintext or to decrypt the ciphertext.
- Cryptanalysis is the science of breaking codes and ciphers.

- Hash algorithm is an algorithm that converts a text string of arbitrary length into a string of fixed length.
- Cipher is a cryptographic algorithm, i.e. a mathematical function used for encryption and decryption.
- Decipher converts enciphered text to the equivalent plain text by means of a cipher system.
- Key Management - Process by which key is generated, stored, protected, transferred, loaded, used, and destroyed.

2 Symmetric key cryptography

The process of encryption and decryption of information by using a single key is known as secret key cryptography or symmetric key cryptography. In symmetric key cryptography the keys used to encrypt the plaintext and to decrypt the ciphertext may be identical (usual situation) or there may be a simple transformation to go between the two keys. The main problem with symmetric key algorithms is that the sender and the receiver have to agree on a common key. A secure channel is also required between the sender and the receiver to exchange the secret key.

Both parties must protect the key; the disclosure of the key by either party can result in compromise of the information

The process of using symmetric-key cryptography is as follows: User A wants to send a message to User B and wants to ensure that only User B is able to read the message. To secure the transmission, User A generates a secret key, encrypts the message with this key, and sends the message to User B. User B needs that secret key to read the encrypted message. User A can give the secret key to User B by using any means available. After User B receives the secret key, he or she can decrypt the message to retrieve the original message.

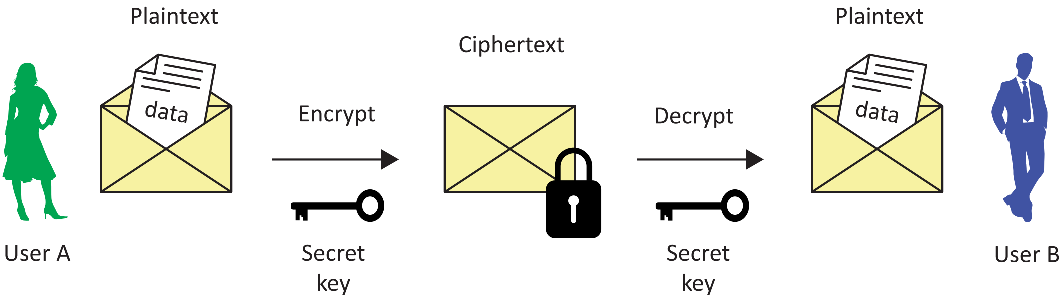


Fig 2.1. Symmetric-key encryption model

The properties that a cipher algorithm must fulfill are the following:

- Diffusion: each bit of the plaintext influence many ciphertext bits and each ciphertext bit is affected by many plaintext bits,
- Confusion: it is necessary to avoid structured relationships (especially linearity) between plaintext and ciphertext/key that are exploited in known attacks.
- Ciphertext should be random-looking and have good statistical properties.
- Simplicity.
- Efficiency: extremely fast in hardware & software on wide variety of platforms.

The major problem with symmetric cryptography is that the process of transferring keys to the recipient is prone to security risks. Transferring the secret key over the Internet in an e-mail message is insecure. Verbally communicating the key over a phone line runs the risk of eavesdropping. Similarly, snail mail runs the risk of possible interception.

The security risks that are involved in secret key cryptography have been overcome to a large extent by using public key cryptography. Secret-key cryptography is often used to encrypt data on hard drives. The person encrypting the data holds the key privately and there is no problem with key distribution.

As we mentioned in the previous section, an important distinction among secret key algorithms is between stream and block ciphers. Nowadays, block ciphers are more used than stream ones.

2.1 Block ciphers algorithms

Block ciphers transform the group of symbols of the plaintext into the group of symbols of the cipher text. So the encryption is realized block by block of the plaintext.

Symbols of the plaintext are grouped into the block of the plaintext and cryptographic transformation is applied to this block based on the key. The result of the encryption is the block of the cipher text with the same size as the size of the plaintext block.

It is possible that the plaintext size will not be exactly a multiple of the block length; in this case usually a padding-scheme is applied to fill up the last block. However, depending on the operation mode, padding might not be needed. The principle of the encryption and decryption based on the block cipher is shown in the following figure.

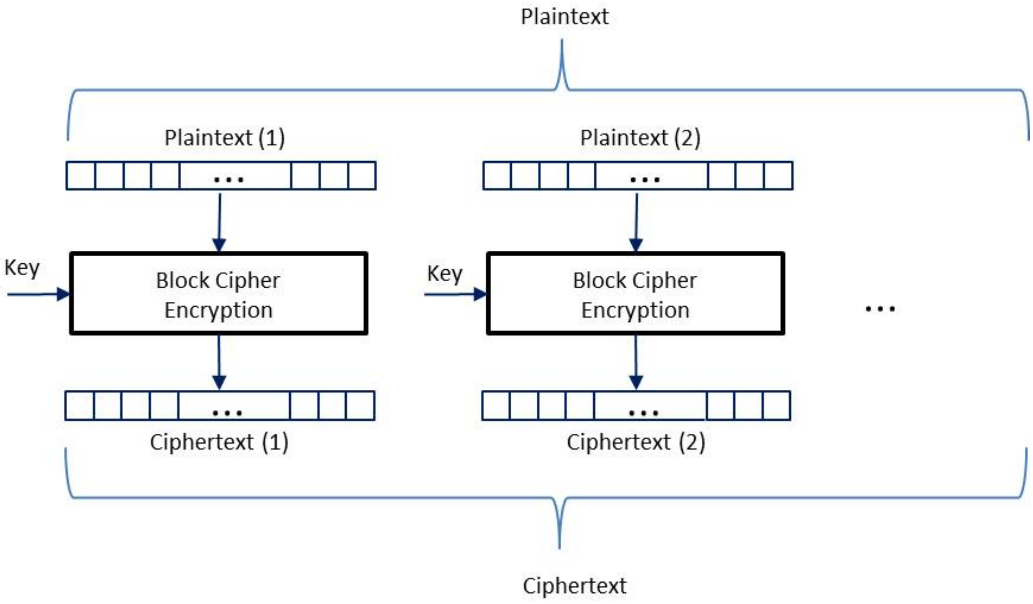


Fig. 2.2. Block cipher scheme

Most block ciphers are based on the concept of an iterated product cipher. These ciphers carries out the encryption in multiple rounds, each of them repeats a series of operations on the data applying a different subkey derived from the original key. The operations applied in each round normally comprises: substitution, permutation and key-mixing. These ciphers are known as substitution-permutation networks (SPN) and Feistel ciphers. Since the substitution box is the only non-linear part of most ciphers, the substitution boxes in must be chosen very carefully to protect against cryptanalysis attacks.

Decryption is realized by analogical approach. This transformation is applied on the ciphertext block by using the same key *k* (in the case of secret key algorithms) as was used in encryption process. The result of this process is decrypted plaintext block.

Typically, the size of the plaintext block is 64 or 128 bits and the ciphertext block is the same size.

The advantages of block ciphers:

- High level of diffusion
- Immunity to tampering: difficult to insert symbols without detection.

The most widely used block secret key algorithms include:

- Data Encryption Standard (DES)
- Advanced Encryption Standard (AES)

It is not recommended to use the same secret key bits for encrypting the same plaintext parts. If an algorithm is used for some number of identical plaintext blocks, the result is some number of identical ciphertext blocks. There are ways to blur and mix plaintext blocks with ciphertext blocks, preventing blockwise modification attacks. These methods are called the block cipher modes of operations.

Modes of operation

Block Ciphers can be used in a variety of ways, with different secrecy properties and error recovery properties. These modes apply to almost all of different block ciphers in existence. The choice of encryption mode affects the speed, the security against adversaries and the error propagation.

ECB: Electronic Code Book

It is the basic cipher, without any modification. The message is split into blocks, and each plaintext block is encrypted separately, independently of the others. Therefore, there is no interdependency between blocks and in consequence this mode is not recommended. The use of this mode introduces some drawbacks:

The plaintext structural information is exposed

It is susceptible to attacker blockwise modification: blocks can be reordered and the reordering or repetition of blocks can change the message.

Any ciphertext encrypted with the same key can be used as source material for the attacker.

A typical example of weakness of encryption using ECB mode is encoding a bitmap image (for example a .bmp file). Even a strong encryption algorithm that uses ECB mode cannot blur efficiently its content.

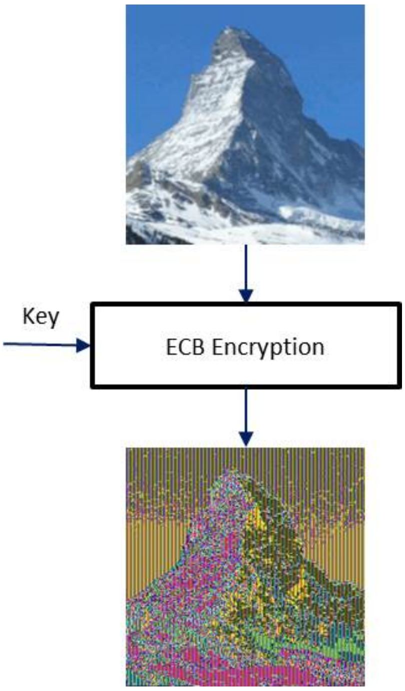


Fig 2.3. Plaintext bmp image and the corresponding ECB encrypted

The Cipher Block Chaining Mode

The Cipher Block Chaining (CBC) mode combines (“chaining”) plaintext blocks with the previous ciphertext blocks. It requires an IV to combine with the first plaintext block.

In the encryption process, the IV is exclusive-ORed with the first plaintext block before encryption. The result is then encrypted and the output is the first block of the ciphertext. For later blocks the prior ciphertext is used instead of the IV. The consequence of the chaining operation is that ciphertext block c_j depends on plaintext block p_j and the previous ciphertext block c_{j-1}. It is easy to see that this dependence is equivalent to say that c_j depends on the actual and all the preceding plaintext blocks.

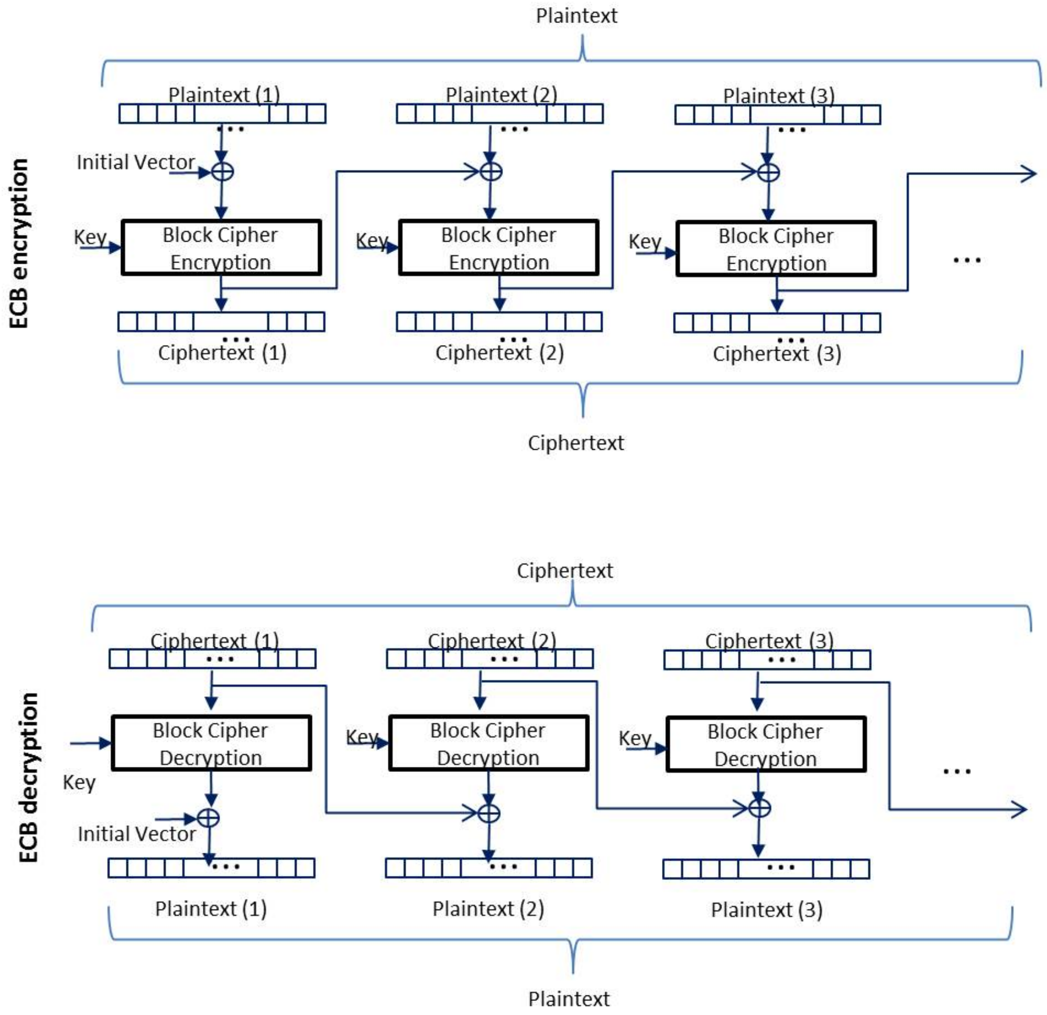


Fig 2.4 CBC encryption and decryption mode

The use of CBC solves the disadvantages of ECB, but introduces the following two drawbacks

- Parallel encryption is not possible: it is not possible to encrypt block p_{j+1} before or during encryption of block p_j, since c_j is needed. Nevertheless, decryption is parallelizable; plaintext block p_j requires ciphertext blocks c_j and c_{j-1}.

- Error-propagation. A single bit error during transmission of c_j will result in failure of not only p_j but of the next block p_{j+1} as well. Nevertheless, only one bit of p_{j+1} block will be flipped. This is called “limited error-propagation”.

The Cipher Feedback Mode (CFB)

According to NIST definition, “Cipher Feedback (CFB) mode is a confidentiality mode that features the feedback of successive ciphertext segments into the input blocks of the forward cipher to generate output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa”. An important parameter in this mode is s, an integer such that $1 \leq s \leq L$, being L the block length of the block.

The first input block is the IV. Basically, the encryption process in CFB mode take as input the *L-s* least significant bits of the previous input concatenated with the *s* bits of most recent ciphertext, encrypts this new input, and then the s most significant bits are exclusive-ORed with the corresponding *s* bits of the plaintext block to generate the next ciphertext block. Next figure illustrates this operation mode

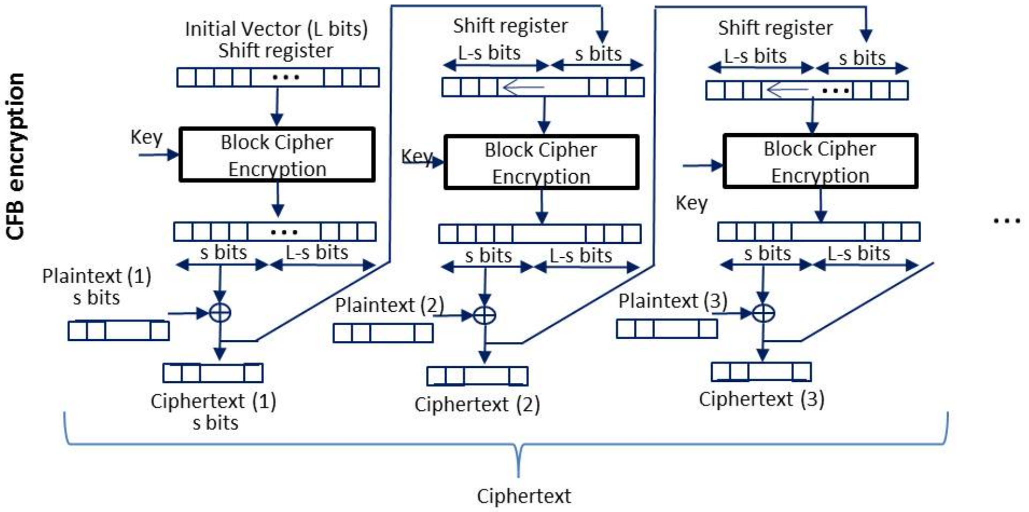


Fig 2.5 CFB encryption mode

In the case s=1, CFB makes a block cipher into a stream cipher by encrypting single bits

The possibility to apply this mode of operation in parallel is the same as in CBC mode; that is to say, multiple forward cipher operations cannot be performed in parallel, but the CFB decryption can be performed in parallel some plaintext values instantly one at a time, for which ciphertext feedback is a method.

Regarding error propagation, a single bit error on c_j may flip the corresponding bit on p_j, but changes p_{j+1} significantly.

The Output Feedback Mode (OFB)

The output feedback mode operates in the following way:

The first input block is the initial vector (IV). The corresponding input block is encrypted and the leftmost *s* bits of the output of this encryption are used for two different functions. On the one hand is the input of the next block, and in the other hand the *s* bits are exclusive-ORed with the *s* bits of the plaintext block to generate the ciphertext block. Thus, the successive output blocks are produced from applying the forward cipher function to the previous output blocks, and the output blocks are exclusive-ORed with the corresponding plaintext blocks to generate the ciphertext blocks.

In fact, OFB is a form of stream cipher. Next figure illustrate the procedure

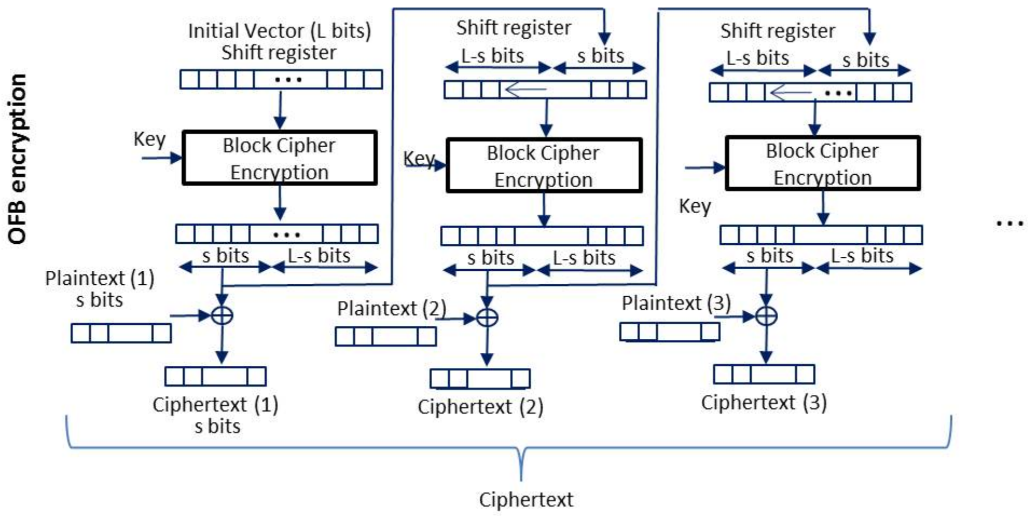


Fig 2.6 OFB encryption mode

It is easy to prove that this mode does not propagate errors; a single bit error on c_j only affects the corresponding bit of p_j.

Regarding CFB, the main advantage of OFB is the following:

+

If the IV is known, it is possible to preprocess the output blocks before knowing the plaintext (or ciphertext data in decryption)

And the drawbacks are:

–

Neither the encryption nor the decryption can be performed in parallel, since each input block depends on the results of the previous cipher function

An active attacker can make controlled changes to plaintext since there is no error propagation

The Counter Mode (CTR)

This mode is based on the encryption of a set of input blocks called counters. The output blocks are exclusive-ORed with the plaintext to generate the ciphertext and vice versa. In general, given the initial counter block for a message, the successive counter blocks are derived by applying an incrementing function. Usually, the counter is split into two sections: message number and block number within the message. It is essential that counter never repeat for any given key. CTR mode is illustrated in the following figure.

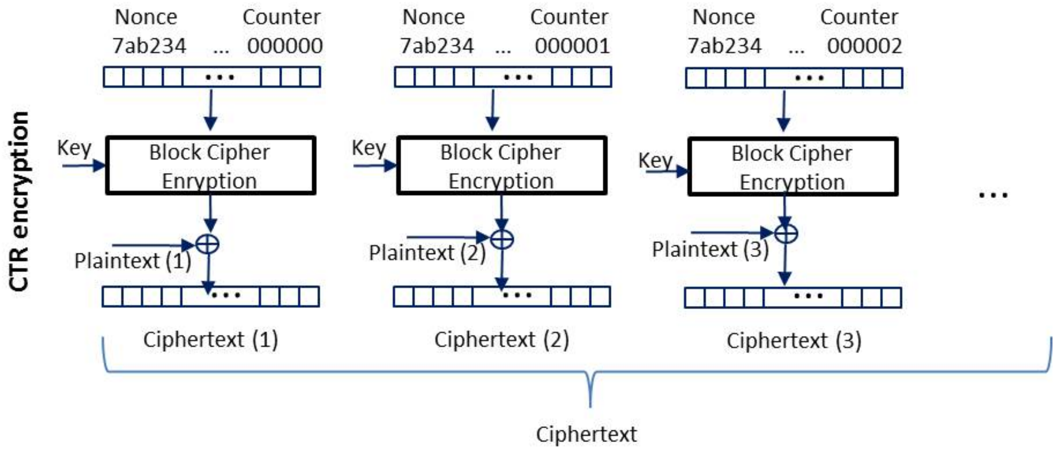


Fig 2.7 CTR encryption mode

It is easy to prove that this mode does not propagate errors; if a block is modified due to a transmission error; only this block will be decrypted erroneously.

The main advantages of this mode are:

Both CTR encryption and CTR decryption are highly parallelizable; there are no linkage between stages.

Preprocessing is possible: the cipher functions can be applied before knowing the plaintext (or ciphertext data in decryption)

The main drawback is

An active attacker can make controlled changes to plaintext

General comments

CBC mode is the most suitable for general file or packet encryption. When there are important requirements of high-speed data, CTR is the best option. In the case that error propagation is not desirable and the transmission line is noisy, OFB is a good option. And, in the case of risk of byte or bit deletion, a good choice is the use of CFB with s=8, or s=1.

2.2 Stream cipher algorithms

A stream cipher is a symmetric cipher which operates with a time-varying transformation on individual digits of the plaintext. This is achieved by adding a digit from a keystream to a plaintext digit. The keystream, also called the running-key, is a pseudo-random sequence (a sequence that appear like a random sequence to an attacker) produced by a finite state automaton whose initial state is determined by a secret key and a public parameter.

The security of a stream cipher completely depends on the keystream. The keystream must be unpredictable to prevent a successful attack.

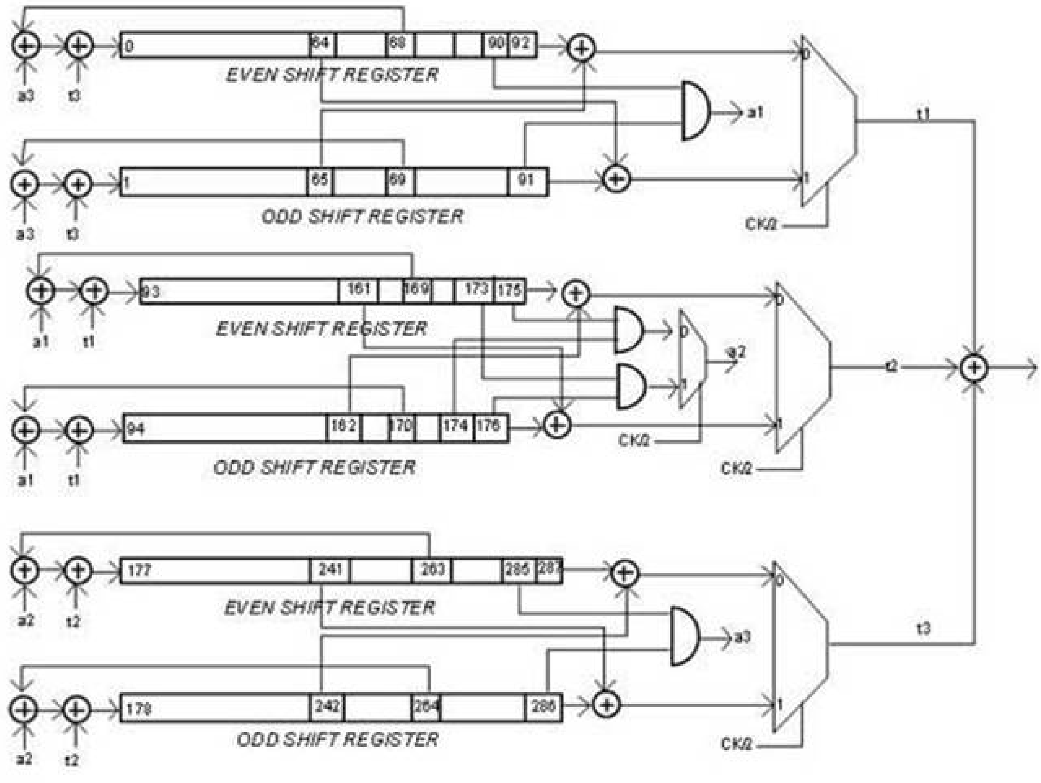


Fig 2.8 Example of a stream cipher: the Trivium

Stream ciphers sometimes require fewer resources, e.g., code size or chip area, for implementation than block ciphers, and they are attractive for use in constrained environments such as cell phones.

Stream ciphers are less popular than block ciphers in most domains such as Internet security. There are exceptions, for instance, the popular stream cipher RC4.

Types of stream ciphers

A stream cipher generates successive elements of the keystream based on an internal state. In a synchronous stream cipher, the state update mechanism is updated independently of the plaintext and the ciphertext. By contrast, self-synchronising stream ciphers update their state based on previous ciphertext digits.

Synchronous stream ciphers

A synchronous stream cipher is a stream cipher, in which the keystream is generated independently of the plaintext and of the ciphertext. The keystream is usually produced by a pseudorandom generator, parameterized by a key, which is the secret key of the whole scheme.

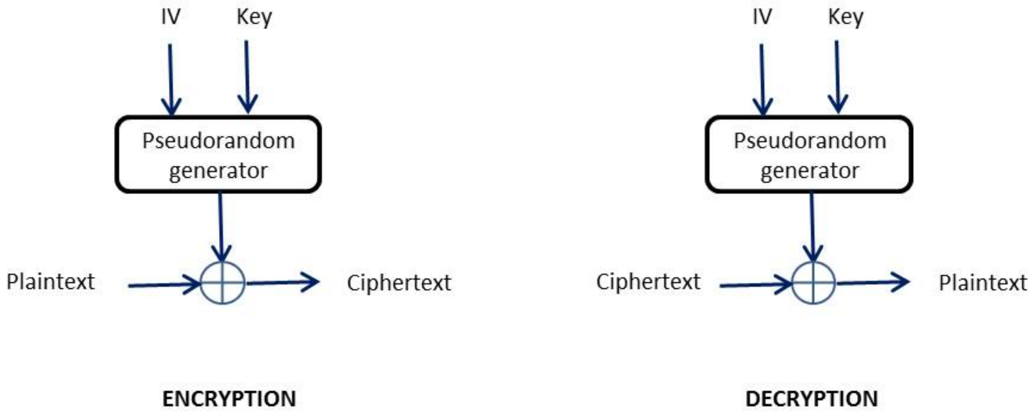


Fig 2.9 Synchronous stream cipher

Some important properties of synchronous stream ciphers include the following:

- No error propagation: a single bit error on c_j only affects the corresponding bit of p_j. The transmission error will not affect the decryption of other digits
- In order to get a proper decryption, sender and receiver need to be synchronized. If a digit is inserted or deleted during transmission, sender and receiver need to be resynchronized

Self-Synchronizing Stream Cipher

In a self-synchronizing, or asynchronous, stream cipher, the keystream depends on the secret key of the scheme, but also of a fixed number, say t, of ciphertext digits (that have already been produced, or read; this distinguishes it from a synchronous stream cipher).

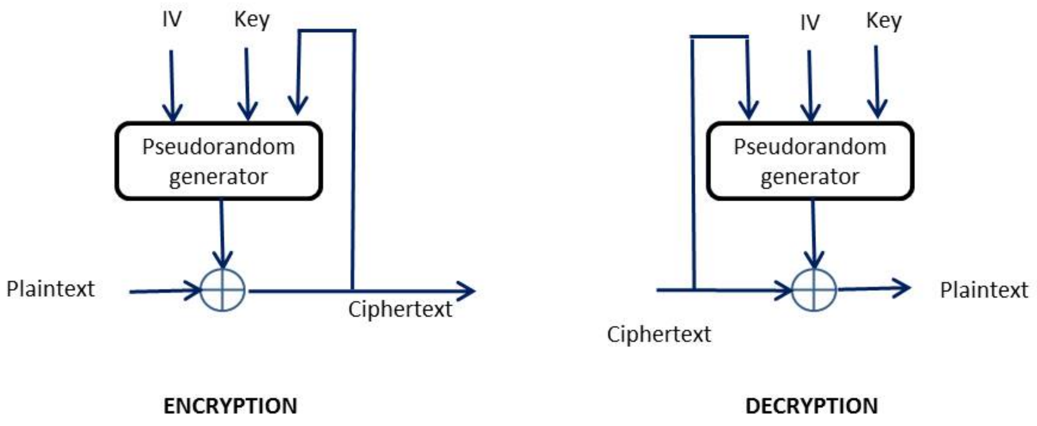


Fig 2.9 Self-synchronizing stream cipher

Self-synchronizing The main properties of this scheme are the following:

- Self-synchronization: if some digits of the ciphertext are deleted, inserted or flipped, the cipher is able to automatically resume proper decryption after some digits.
- Limited error propagation: the effect of single digit errors is limited, only some ciphertext digits will be wrongly decrypted.

3 Public key cryptography

The public key cryptography evolved to address the security issues posed by symmetric cryptography. This method solves the problem of secret key cryptography by using two keys instead of a single key. Public key cryptography uses a pair of keys. In this process, one key is used for encryption, and the other key is used for decryption.

This process is known as public key cryptography or asymmetric cryptography because both the keys are required to complete the process. These two keys are collectively known as the key pair. In asymmetric cryptography, one of the keys is freely distributable. This key is called the public key. Hence, this method of encryption is also called public key encryption. The second key is the secret or private key. The private key is not distributable. This key, like its name suggests, must remain confidential to its respective owner. Because the key pair is mathematically related, whatever is encrypted with a public key may only be decrypted by its corresponding private key and vice versa. It is important to point out that it is virtually impossible to deduce the private key from the knowledge of the public key.

There is a basic flaw in public-key encryption: given enough time and computing power, it will be possible figure out the secret key from the public key and decrypt the message. For this reason public-key encryption relies on keys that are really big, usually the keys are made up of 1024 or 2048 bits. The longer the used keys (that is, the more bits they have), the tougher the encryption.

The algorithms for public key cryptography are based on mathematical problems that currently admit no efficient solution. It is computationally easy for a user to generate a public and private key-pair and to use it for encryption and decryption. These complex mathematical problems make it extremely difficult for a properly generated private key to be determined based only on the knowledge of the public key. The security of public key cryptography is ensured in this way; the strength of the algorithm lies in such difficulty. Thus the public key may be published without compromising security. Security depends only on keeping the private key private. Public key algorithms, unlike symmetric key algorithms, do not require a secure for the initial exchange of one (or more) secret keys between the parties.

Two common uses of public key cryptography are public key encryption and digital signatures. Public key encryption is when a secret message is encrypted using a public key but only the person who possesses the secret key can decode and read the secret message. Digital signature is a message that is signed with a sender's private key and can be verified by anyone with access to the sender's public key. Both of these applications are examples of confidentiality and authenticity of public key encryption.

Public key algorithms are slow compared with symmetric ones. Often, in order to solve this problem, public key is used to distribute the symmetric key. This symmetric key will be used to cipher the user information.

While key management is much simpler in public key cryptography compared with symmetric cryptography, there is a common misconception that key management is trivial with public key cryptography. Moreover, some users mistakenly believe that public key cryptography encryption is more secure from cryptanalysis than symmetric encryption. In fact the security of any system depends on key length and the computational work involved in breaking the cipher.

The most common public key algorithm is ***RSA.***

3.1 How does public key cryptography work?

Using public key encryption to provide confidentiality

Let us take an example where User_B wants to send a message to User_A. User_B encrypts the message with User_A's public key, and User_A decrypts the message using his or her private key. Since the key pairs are complementary, only User_A's private key can decrypt this file. If someone else intercepts the ciphertext, they will be unable to decrypt it, because only User_A's private key can be used for decryption. This method does not provide any authentication that the message is coming from User_B, because User_A's public key is known to the world. However, it does provide confidentiality to the message, as only User_A can decrypt the message.

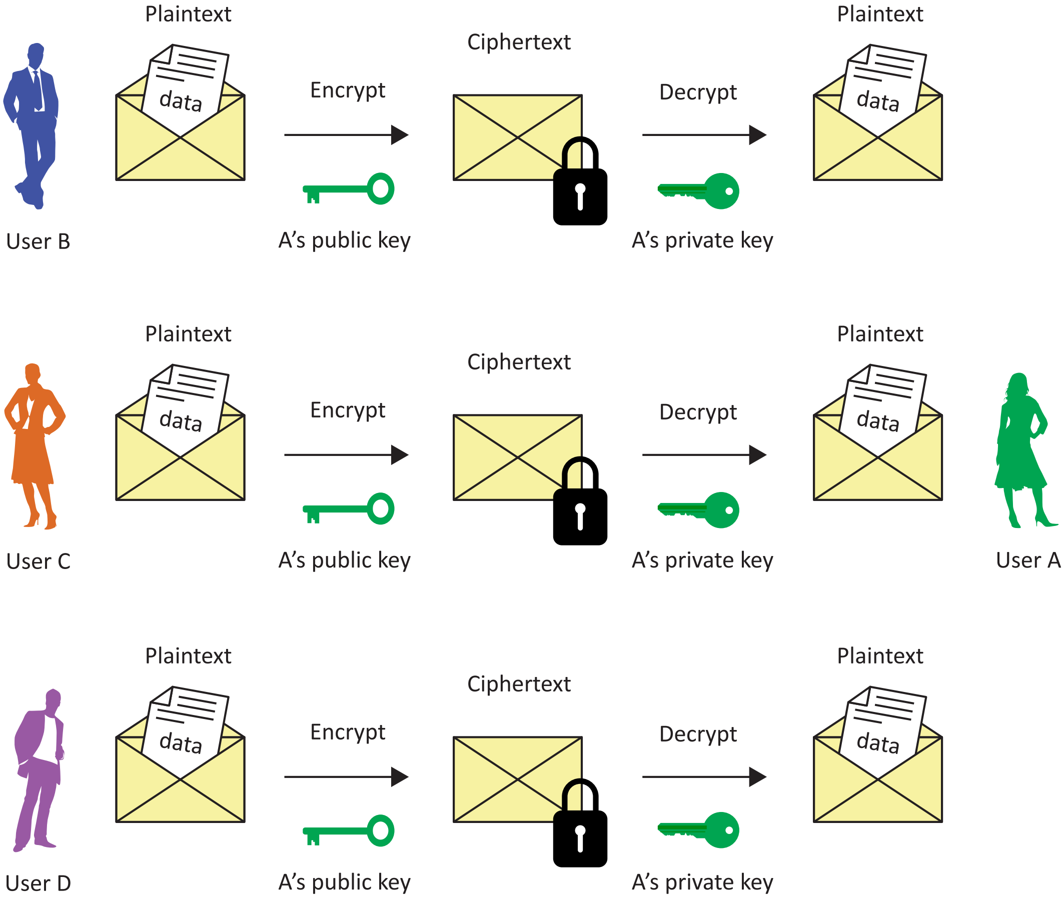


Fig 3.1 Public-key encryption model (to provide confidentiality)

This method very clearly indicates that the data you send to a user can only be encrypted by the public key of the recipient if confidentiality is required. Similarly, the decryption can be done only by the private key, which is supplied by the recipient of the data. Therefore, messages can be exchanged securely. The sender and receiver do not need to share a key, as required for symmetric encryption. All communications involve only public keys, and no private key is ever transmitted or shared.

Using public key encryption to provide authentication

To provide authentication, User_A must encrypt the message with his or her private key and User_B will decrypt the message with User_A's public key. This method will provide authentication that the message is coming from User_A but it does not provide confidentiality, because User_A's public key is known to all. Hence, anybody possessing User_A's public key could decrypt the message.

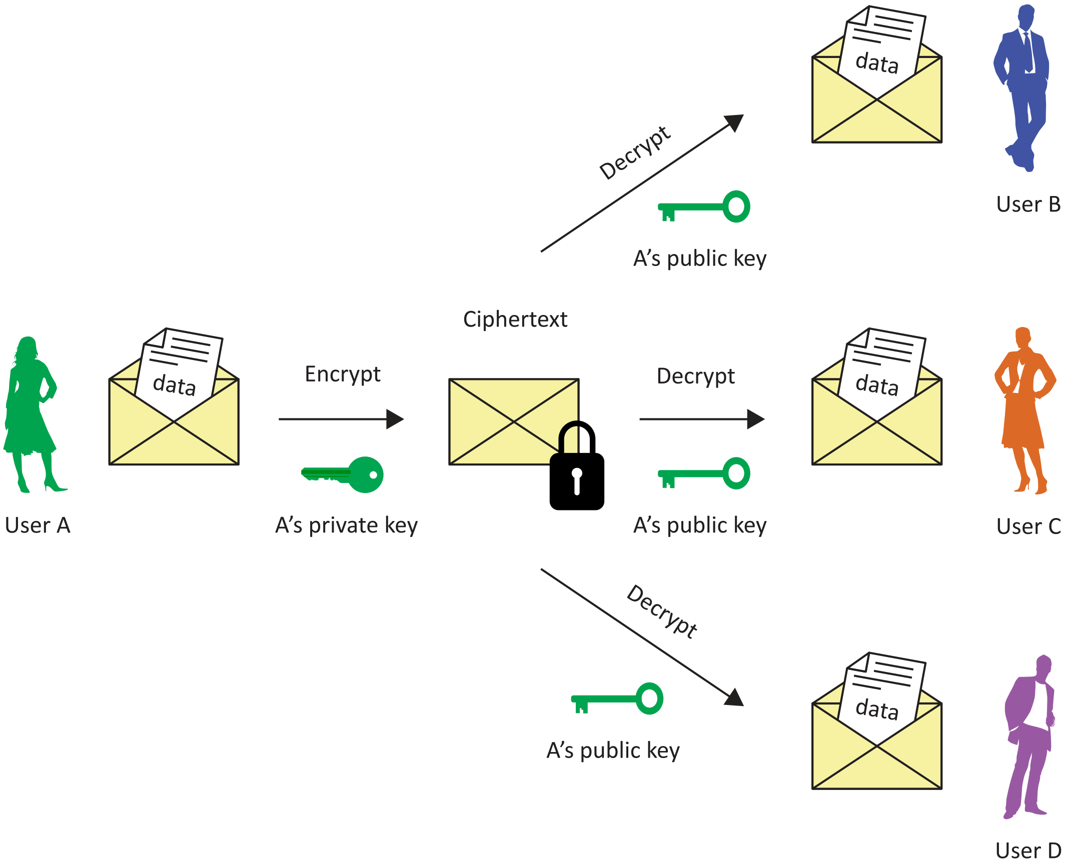


Fig 3.2 Public-key encryption model (to provide authentication)

Using public key encryption to provide authentication and confidentiality

To provide both confidentiality and authentication, User_B will need to encrypt the plaintext first with his or her private key, which will provide authenticity. Then, User_B will use User_A's public key to encrypt the message, which will provide confidentiality.

The disadvantage of the system is that it will be very time consuming and complex as public key encryption and decryption has to be done four times, and the key length of the public key is large (1024 bits to 4094 bits).

4 Hybrid system: Combining Symmetric and Asymmetric Encryption

The disadvantage of using *public key encryption* is that it is *quite a slow process*, as key lengths are large (1024 bits to 4094 bits). When both processes are compared, *symmetric key encryption* is significantly *faster*, as the key length is smaller (40 bits to 256 bits). On the other hand, there is a problem in transferring the key in secret key encryption. Both these techniques can be used together to provide a better method of encryption. This way one can make use of the combined advantages and overcome the disadvantages.

Specifically, the hybrid system uses a public key algorithm in order to safely share the symmetric encryption system's secret key. The real message is then encrypted using this key and then sent to the recipient. Since the key sharing method is secure, the symmetric key used for the encryption changes for each message sent. For this reason it is sometimes called the session key. This means that if the session key was intercepted, the interceptor would only be able to read the message encrypted with that key. In order to decrypt other messages the interceptor would have to intercept other session keys.

The session key, encrypted using the public key algorithm, and the message being sent, encrypted with the symmetric algorithm, are automatically combined into a single package. The recipient uses his or her private key to decrypt the session key and then uses the session key to decrypt the message. Many applications use this system.

The steps in data transaction within a combined technique are:

1. Encrypt the plaintext using a symmetric encryption and a random key.

2. Encrypt only this random key with the recipient’s public key using asymmetric encryption. Now send the encrypted random key to the recipient. The recipient, at his or her end, can now decrypt the random key using his/her private key.

3. Next, send the actual encrypted data. The encrypted data can be decrypted using the key that was encrypted by using the public key from the asymmetric key pair.

The combined technique of encryption is widely used. For instance, it is used in *Secure Shell (SSH)* to secure communications between the client and the server and in *PGP (Pretty Good Privacy)* for sending messages. Above all, it is the heart of *Transport Layer Security (TLS),* which is widely used by Web browsers and Web servers to maintain a secure communication channel with each other.

The next figure illustrates the process.

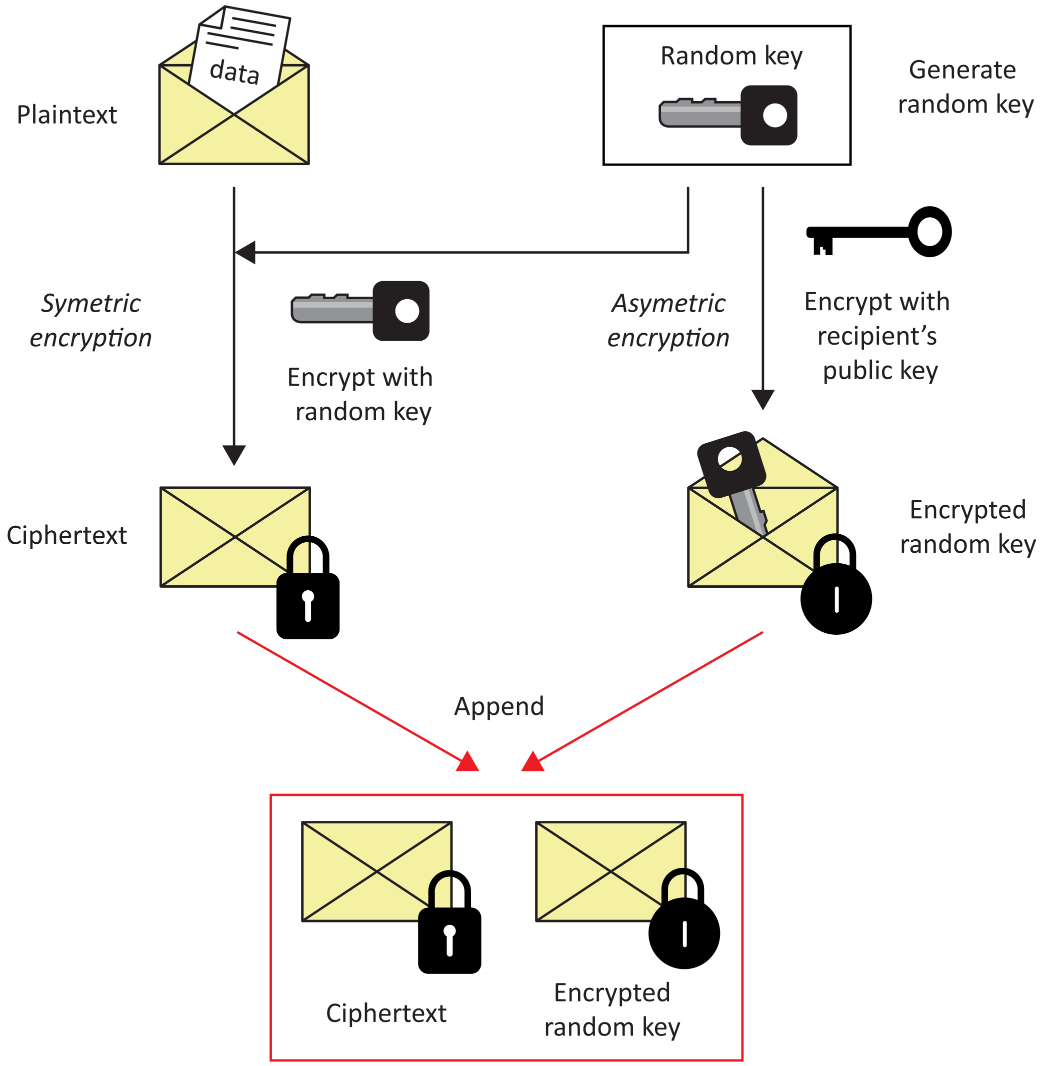


Fig 4.1 Hybrid encryption model (to provide confidentiality)

5 Hash functions

The term hash function originates historically from computer science, where it denotes a function that compresses a string of arbitrary input to a string of fixed length. Any change to the input data will (with very high probability) change the hash value. Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties. Cryptographic hash functions can be used to provide message integrity, to protect information authenticity, to protect against the threat of repudiation and to secure passwords. Unlike secret key and public key algorithms, hash functions, also called message digests, have no key.

The basic requirements for a cryptographic hash function are:

- the input can be of any length,
- the output has a fixed length,
- it is easy to compute the hash value for any given message,
- hash functions are one-way, that is to say, it is computationally infeasible to generate a message that has a given hash,
- it is infeasible to modify a message without changing the hash,
- Collision resistant, that is to say, it is computationally infeasible to find two distinct messages that hash to the same result.

The hash value represents concisely the longer message or document from which it was computed. One can think of a message digest as a "digital fingerprint" of the larger document.

The main role of a cryptographic hash function is in the provision of *digital signatures*. Additionally, a digest can be made public without revealing the contents of the document from which it is derived.

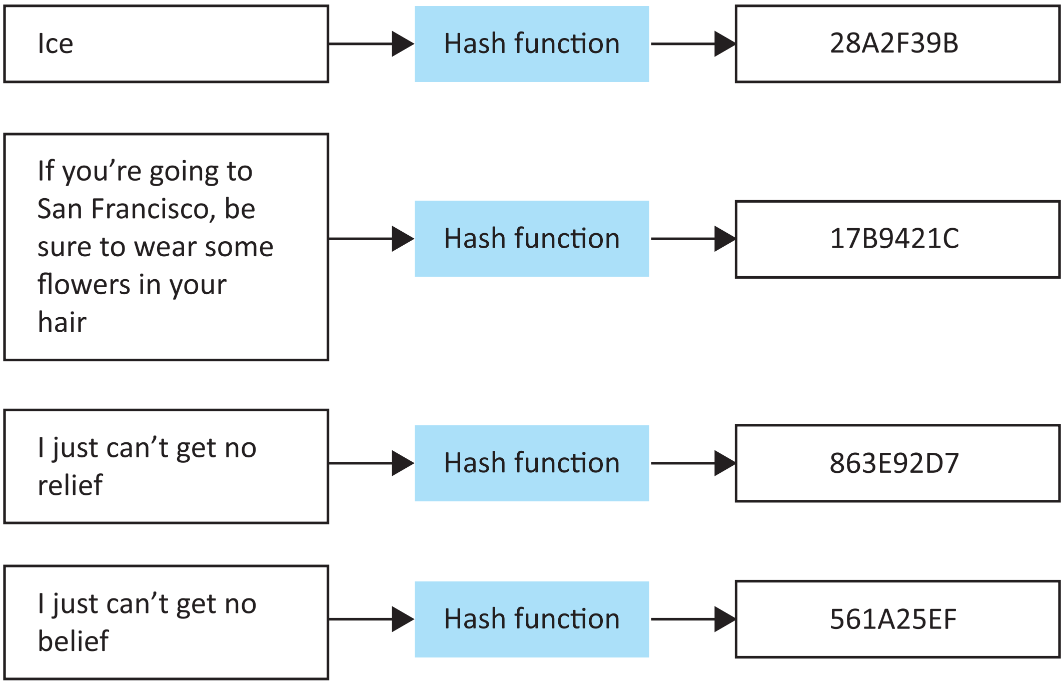

Ice
Hash function
28A2F39B
If you're going to San Francisco, be sure to wear some flowers in your hair
Hash function
17B9421C
I just can't get no relief
Hash function
863E92D7
I just can't get no belief
Hash function
561A25EF

Fig 5.1 Hash function

6 Digital signature

Digital signatures are the most important development from the work on public-key cryptography, and provide a set of security capabilities that would be difficult to implement in any other way. A digital signature is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document, and possibly to ensure the integrity of the message. Digital signatures are easily transportable and cannot be imitated by someone else. The ability to ensure that the original signed message arrived means that the sender cannot easily repudiate it later.

Digital signatures are based on handwritten signatures, which are used for ownership rights or message content confirmation.

Handwritten signatures should have the following properties:

- *The signature is secure* – the signature should not be imitated and any potential attempt at signature forgery should be detected easily.
- *The signature facilitates authentication* – the signature identifies uniquely the signature keeper, who signed the document without restraint and wittingly.
- *The signature is untransferable* – the signature is part of the document and an unauthorized subject is unable to transfer the signature to another document.
- *The signed document is unchangeable* – the document cannot be changed and modified after the signature.
- *The signature is undeniable* – the keeper of the signature cannot deny the subscription of the signed document.

In practice, none of these features is consistently fulfilled in handwritten signatures and can be discredited or corrupted. All these features should have digital signatures too. However, there are some problems associated to the practical realization of digital signatures. Digital files can be easily copied and part of a document can be transmitted to another document and the signed document can be easily modified.

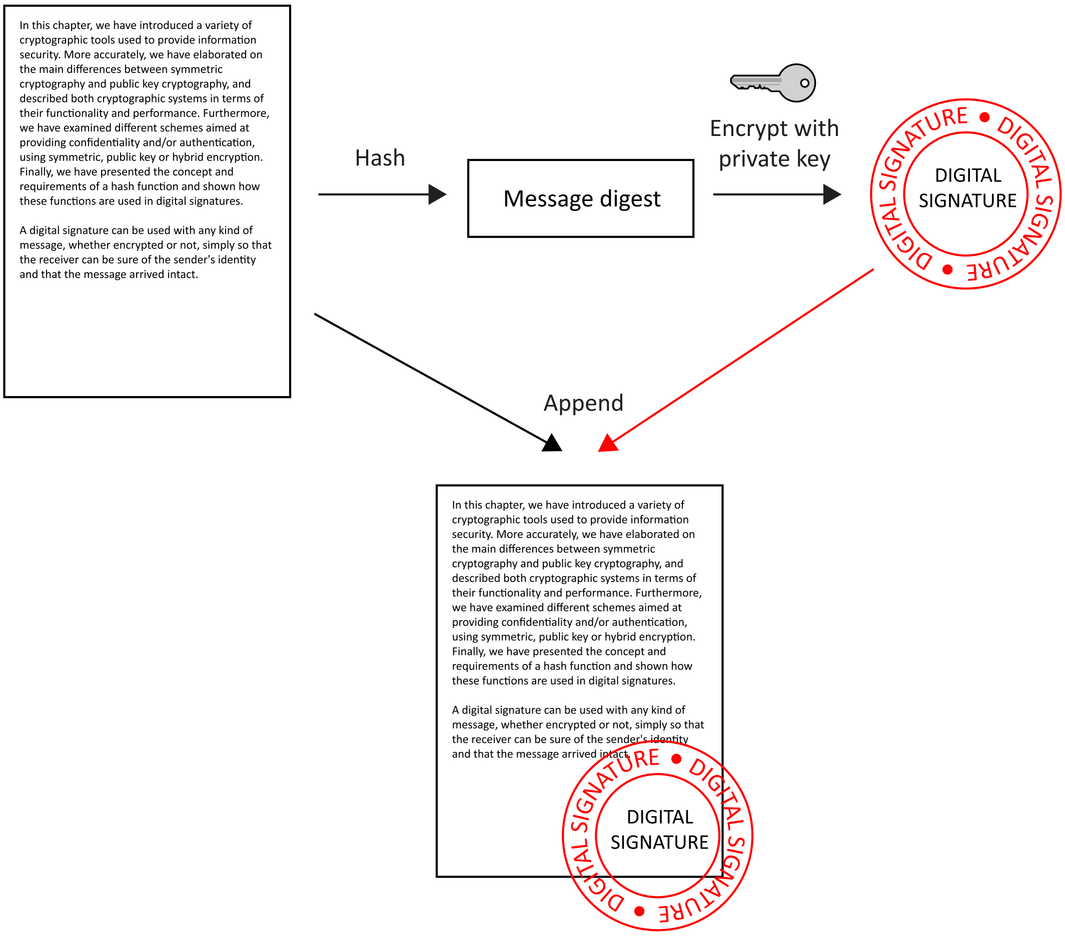
The following requirements can be formulated for a digital signature:

- The signature must be a bit pattern that *depends on the message* being signed.
- The signature *must use some information unique to the sender*, to prevent both forgery and denial.
- The realization and implementation of the digital signature must be *relatively easy.*
- The *forgery* of the digital signature must be *computationally infeasible*, either by constructing a new message for an existing digital signature or by constructing a fraudulent digital signature for a given message.
- It must be practical to retain a copy of the digital signature in storage.

A digital signature can be used with any kind of message, whether encrypted or not, simply so that the receiver can be sure of the sender's identity and that the message arrived intact.

There are several possible schemes for digital signatures. Among others, one of the most accepted schemes is based on hash functions. In this case, if a user wants to digitally sign a document, the steps that he/she has to follow are:

1. Evaluate the hash of the document to be signed.
2. Using asymmetric encryption, encrypt the hash with the sender's private key to obtain the digital signature.
3. Append the digital signature to the document.



In this chapter, we have introduced a variety of cryptographic tools used to provide information security. More accurately, we have elaborated on the main differences between symmetric cryptography and public key cryptography, and described both cryptographic systems in terms of their functionality and performance. Furthermore, we have examined different schemes aimed at providing confidentiality and/or authentication, using symmetric, public key or hybrid encryption. Finally, we have presented the concept and requirements of a hash function and shown how these functions are used in digital signatures.

A digital signature can be used with any kind of message, whether encrypted or not, simply so that the receiver can be sure of the sender's identity and that the message arrived intact.

In this chapter, we have introduced a variety of cryptographic tools used to provide information security. More accurately, we have elaborated on the main differences between symmetric cryptography and public key cryptography, and described both cryptographic systems in terms of their functionality and performance. Furthermore, we have examined different schemes aimed at providing confidentiality and/or authentication, using symmetric, public key or hybrid encryption. Finally, we have presented the concept and requirements of a hash function and shown how these functions are used in digital signatures.

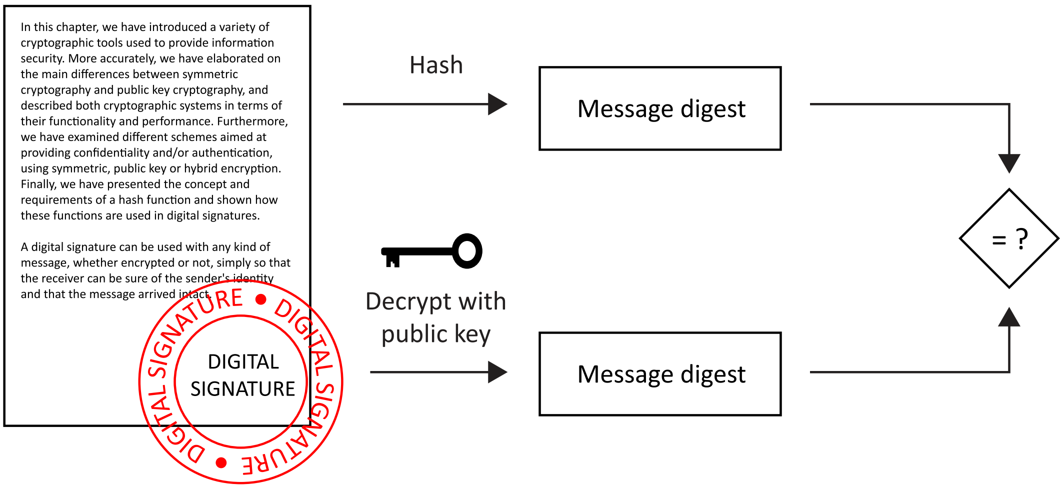
A digital signature can be used with any kind of message, whether encrypted or not, simply so that the receiver can be sure of the sender's identity and that the message arrived intact.

Fig 6.1 Hash based digital signature

The receiver can verify the authenticity of this digital signature following the steps below:

1. Evaluate the hash of the document (excluding the digital signature).
2. Using asymmetric encryption, decrypt the digital signature with the sender's public key to obtain a message digest.

3. Compare the results obtained in the two previous steps



In this chapter, we have introduced a variety of cryptographic tools used to provide information security. More accurately, we have elaborated on the main differences between symmetric cryptography and public key cryptography, and described both cryptographic systems in terms of their functionality and performance. Furthermore, we have examined different schemes aimed at providing confidentiality and/or authentication, using symmetric, public key or hybrid encryption. Finally, we have presented the concept and requirements of a hash function and shown how these functions are used in digital signatures.

A digital signature can be used with any kind of message, whether encrypted or not, simply so that the receiver can be sure of the sender's identity and that the message arrived intact.

Fig 6.2 Verification process of a hash-based digital signature

If the message digests obtained in the two steps are the same, the recipient will know that the signed data has not been changed.

7 Key Exchange. Digital Certification

Digital signatures represent one of the primary uses of public-key cryptography. For messages sent through an insecure channel, a properly implemented digital signature gives the receiver reason to believe the message was sent by the claimed sender. In many aspects, digital signatures are equivalent to traditional handwritten signatures, but properly implemented digital signatures are more difficult to forge than the handwritten type. In order to verify a digital signature, the sender's knowledge of the public key is required. Therefore, a key distribution mechanism is totally needed.

The most accepted approach is based on the usage of digital certificates, which enables the realization of the key exchange.

A digital certificate is an electronic document used to identify an individual, a server, a company, or some other entity and to associate that identity with a public key. It incorporates a digital signature *that binds together a public-key with an identity* — information such as the name of a person or an organization, their address, and so forth. The certificate can be used to verify that a public key belongs to an individual. Certificates help prevent the use of fake public keys for impersonation. Only the public key certified by the certificate will work with the corresponding private key possessed by the entity identified by the certificate.

A digital certificate is a data structure which contains the public key of a subject or certificate holder, as well as the identification data of the certificate holder, a time stamp related to the certificate validity and other data from the certification authority. This structure is signed with the private key of a *certification authority (CA)* and every user is able to check the authenticity of the certificate content by using the public key of the certification authority. Certification authorities are the entities that issue certificates and validate identities.

The next figure shows the structure of a digital certificate:

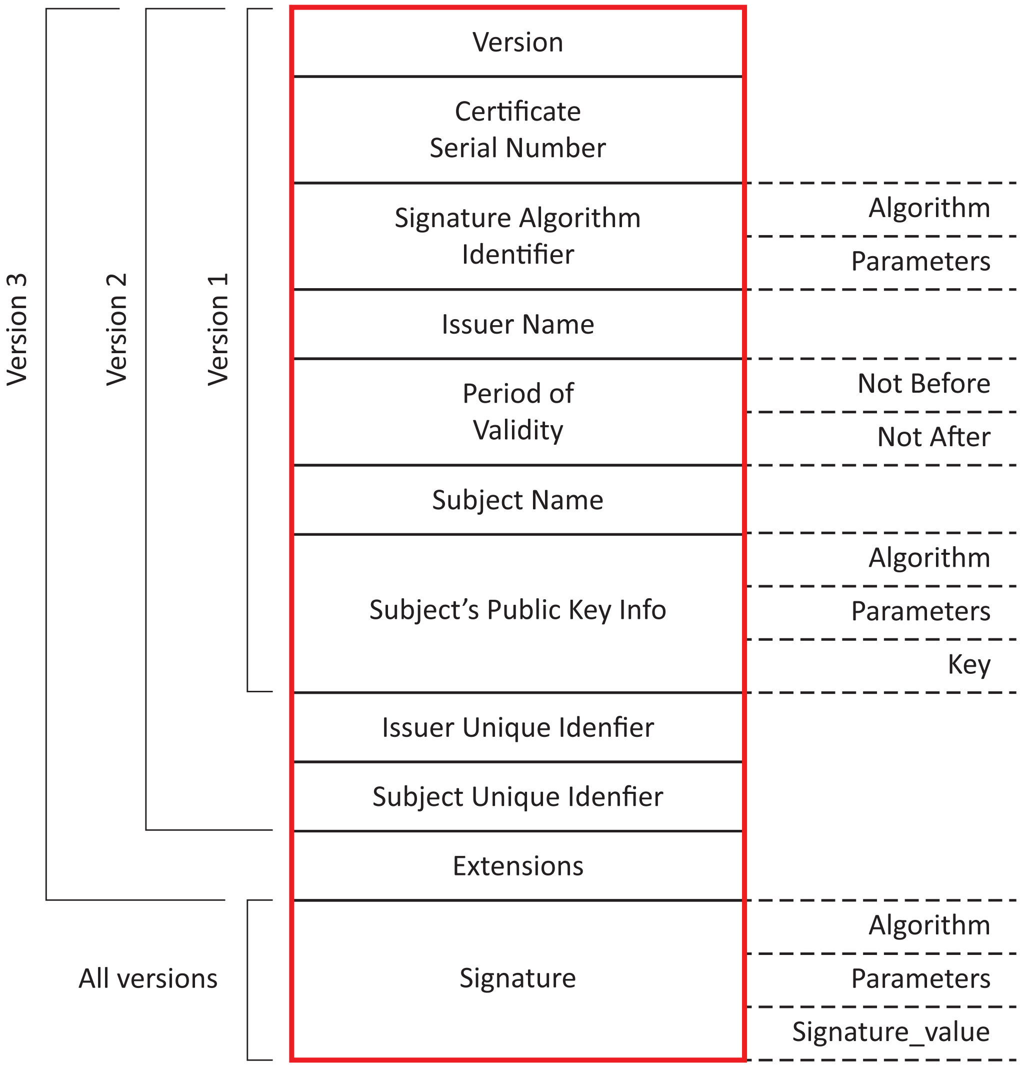


Fig 7.1 Digital certificate structure

8 Cybercriminality: Introduction

Cybercrime or computer crime is any criminal activity involving computers and networks. It can range from fraud to unsolicited emails (spam). These crime cases include attacks against computer data and systems, identity theft, the distribution of child sexual abuse images, internet auction fraud, the penetration of online financial services, as well as the deployment of viruses, botnets, and various email scams such as phishing.

One of the best ways to avoid being a victim of cybercrimes and protecting the sensitive information is by making use of impenetrable security that uses a unified system of software and hardware to authenticate any information that is sent or accessed over the Internet.

Cybercrimes are defined as: "Offences that are committed against individuals or groups of individuals with a criminal motive to intentionally harm the reputation of the victim or cause physical or mental harm to the victim directly or indirectly, using modern telecommunication networks such as Internet and mobile phones (SMS/MMS)". Such crimes may threaten the security and financial health of a state. Issues surrounding these types of crime have become high-profile, particularly those surrounding cracking, copyright infringement, child pornography, and child grooming. There are also problems of privacy when confidential information is lost or intercepted, lawfully or otherwise.

It is important to be conscious that is quite impossible to recognize every cyber criminal activity before it affects the targeted entities. For this reason, it is crucial to have a mature approach to cyber security that emphasizes the aspects of early detection and recovery.

An effective incident response procedure includes the following steps:

- Identification of the threat agent which hit the infrastructure.
- Containment of the threat, preventing it from moving laterally within the targeted infrastructure.
- Forensic investigation to identify the affected systems and the way the threat agent has penetrated the computer system.
- Remediate/Recover by restoring IT infrastructure back online and in production once forensics investigation are complete.
- Report and share threat data to higher management and share the data on the incident through dedicated platforms that allow rapid sharing of threat data with law enforcement and other companies.

Unfortunately, the process described is rarely followed. Up until now, the containment and remediation process has been a primary manual human process that makes it non-responsive and inefficient.

9 Attack techniques

Security attacks can be characterized as the different sorts of systematic activities aimed at decreasing or corrupting the security. From this perspective, an attack can be defined as a systematic threat generated by an entity in an artificial, deliberate and intelligent way. Computer networks may be vulnerable to many threats along many avenues of attack, including:

- Social engineering, wherein someone tries to gain access through social means (pretending to be a legitimate system user or administrator, tricking people into revealing secrets, etc.)
- War dialing, wherein someone uses computer software and a modem to search for desktop computers equipped with modems that answer, providing a potential path into a corporate network.
- Denial-of-service attacks, including all types of attacks intended to overwhelm a computer or a network in such a way that legitimate users of the computer or network cannot use it.
- Protocol-based attacks, which take advantage of known (or unknown) weaknesses in network services.
- Host attacks, which attack vulnerabilities in certain computer operating systems or in how the system is set up and administered.
- Password guessing; passwords are sequences of symbols, usually associated with a user name, which provide a mechanism for identification and authentication of a particular user. On almost all machines, the users themselves choose the passwords. This places the burden of security on end users who either do not know, or, sometimes do not care about sound security practices. As a general rule, passwords that are simple to remember, are, likewise, easy to guess. Attackers have several venues of guessing passwords and overcoming this obstacle.
- Eavesdropping of all sorts, including stealing e-mail messages, files, passwords, and other information over a network connection by listening in on the connection.

Security attacks may be divided into these two main categories:

- Passive attacks
- Active attacks

9.1 Passive attacks

Passive attacks attempt to learn or make use of information from the system but do not affect system resources. A passive attack is one where the attacker only monitors the communication channel. A passive attacker only threatens the confidentiality of data.

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted.

Two types of passive attacks are related to message contents and traffic analysis:

- Eavesdropping. In general, the majority of network communications occur in an unsecured or "plaintext" format, which allows an attacker who has gained access to data paths in the network to "listen in" or interpret (read) the data exchanged over the network. The ability of an eavesdropper to monitor the network is generally the biggest security problem that administrators face in an enterprise. Without strong encryption services that are based on cryptography, the data can be read by others as it traverses the network.
- Traffic analysis. It refers to the process of intercepting and examining messages in order to deduce information from patterns in communication. It can be performed even when the messages are encrypted and cannot be decrypted. In general, the greater the number of messages observed, or even intercepted and stored, the more can be inferred from the traffic.

9.2 Active attacks

Active attacks attempt to alter system resources or affect their operation. This type of attack is one where the adversary attempts to delete, add, or in some other way alter the transmission on the channel. An active attacker threatens data integrity and authentication as well as confidentiality.

Active attacks involve some modification of the data stream or the creation of a false stream and can be divided into six categories:

- Masquerade. It is a type of attack where the attacker pretends to be an authorized user of a system in order to gain access to it or to gain greater privileges than they are authorized for.

- Replay. In this kind of attack, a valid data transmission is maliciously or fraudulently repeated or delayed. This is carried out either by the originator or by an adversary [http://en.wikipedia.org/wiki/Adversary_(cryptography)] who intercepts the data and retransmits them, possibly as part of a masquerade attack [http://en.wikipedia.org/wiki/Spoofing_attack] .

- Modification of messages. The attacker removes a message from the network traffic, alters it, and reinserts it.

- Man in the Middle (MitM). In this kind of attacks, an intruder intercepts communications between two parties, usually an end user and a website. The attacker can use the information accessed to commit identity theft or other types of fraud.

- Denial of Service (DoS) and Distributed Denial of Service (DDoS). A denial of service (DoS) attack is an incident in which a user or organization is deprived of the services of a resource they would normally expect to have. In a distributed denial-of-service, large numbers of compromised systems (sometimes called a botnet [http://searchsecurity.techtarget.com/definition/botnet]) attack a single target.

- Advanced Persistent Threat (APT). It is a network attack in which an unauthorized person gains access to a network and stays there undetected for a long period of time. The intention of an APT attack is to steal data rather than cause damage to the network or organization. APT attacks target organizations in sectors with high-value information, such as national defense, manufacturing and the financial industry.

10 Prevention tips

Cybercrime prevention can be straight-forward - when armed with a little technical advice and common sense, many attacks can be avoided. In general, online criminals are trying to make their money as quickly and easily as possible. The more difficult you make their job, the more likely they are to leave you alone and move on to an easier target. Probably, the best line of defense is still the end-user. The less risks she takes, the lower her chance of being the victim of a computer attack. The tips below provide basic information on how online fraud can be prevented.

- Keep the computer system up with the latest patches and updates. Vendors usually release patches for their software when vulnerability has been discovered. One of the best ways to keep attackers away from a computer is to apply patches and other software fixes when they become available. Most product documentation offers a method to get updates and patches. Some applications will automatically check for available updates, otherwise it is absolutely necessary to check periodically for updates. In any case, by regularly updating your computer, you block attackers from being able to take advantage of software flaws (vulnerabilities) that they could otherwise use to break into your system. While keeping your computer up-to-date will not protect the computer from all attacks, it makes it much more difficult for hackers to gain access to it, blocks many basic and automated attacks completely, and might be enough to discourage a less-determined attacker to look for a more vulnerable computer elsewhere.

- Make sure the computer is configured securely. Installing a system right out of the box and leaving it with the default configuration is probably one of the most common mistakes that people make when setting up a network. When a computer is installed, it is important to pay attention not just to making it run, but also focus on making it work securely. Default configurations often have default administrative accounts and passwords that hackers the world over know. Configuring Internet applications such as Web browser and email software is one of the most important areas to focus on.

- Choose strong passwords and keep them safe. Passwords are often the only protection used on a system. A user ID is only a name and does not verify identification, but the password associated with the user ID works as an identifier. Firewalls and intrusion detection systems mean nothing if your passwords are compromised. A strong password is one that cannot be found in any dictionary. It also means a password that is not easily guessed.

- Protect the computer with security software. Several types of security software, including firewall and antivirus are necessary for basic online security. A firewall is a software or hardware product that screens the information coming into and leaving your computer to ensure that there is no unauthorized access to your computer, providing in this way the first line of defense. The next line of defense many times is the antivirus software, a computer program that can be used to scan files to identify and eliminate computer viruses and other malicious software (malware). Strictly speaking, a virus is a program that can replicate itself and is designed to spread from one computer to another, doing things the end-user doesn't want and/or doesn't know about. Malware is a broader term, short for malicious software, and there are many different forms, including viruses, Trojan horses, keyloggers, worms, adware, and spyware, to name a few.

- Protect the personal information. Identity Theft has become a major problem with people using the Internet for cash transactions and banking services. In this cyber crime, a criminal accesses data about a person’s bank account, credit cards, Social Security, debit card and other sensitive information to siphon money or to buy things online in the victim’s name. It can result in major financial losses for the victim and even spoil the victim’s credit history. Therefore, caution is required when sharing personal information such as name, home address, phone number, and email address online. To take advantage of many online services, you will inevitably have to provide personal information in order to handle billing and shipping of purchased goods.