

1. Introduction

The recent digital revolution and widespread access to telecommunication networks have enabled the emergence of e-commerce, and e-government. This proliferation of digital communications and the transition of social interactions into the cyberspace have raised new concerns in terms of security and trust, like confidentiality, privacy and anonymity; data integrity; protection of intellectual property and digital rights management; threats of corporate espionage, and surveillance system, etc. These issues are interdisciplinary in their essence, drawing from several fields: algorithmic number theory, cryptography, network security, signal processing, software engineering, legal issues, any many more.

In this first chapter, we discuss the technical and applicative aspects of cryptology with its two branches: cryptography and cryptanalysis.

2. Security Properties

Also known as service security, basic requirements ... etc, it is all about what the users expect from the informatic systems to provide. The following elements are some of those properties:

2.1. Authentication

Using a cryptographic system, we can establish the identity of a remote user (or system). A typical example is the SSL certificate of a web server providing proof to the user that he or she is connected to the correct server. The identity is not of the user, but of the cryptographic key of the user. Having a less secure key lowers the trust we can place on the identity. [1].

2.2. Non-Repudiation

The concept of non-repudiation is particularly important for financial or e-commerce applications. Often, cryptographic tools are required to prove that a unique user has made a transaction request. It must not be possible for the user to refute his or her actions. For example, a customer may request a transfer of money from her account to be paid to another account. Later, she claims never to have made the request and demands the money be refunded to the account. If we have non-repudiation through cryptography, we can prove –

usually through digitally signing the transaction request, that the user authorized the transaction. [1].

2.3. Confidentiality

More commonly, the biggest concern will be to keep information private. Cryptographic systems were originally developed to function in this capacity. Whether it be passwords sent during a log on process, or storing confidential medical records in a database, encryption can assure that only users who have access to the appropriate key will get access to the data. [1].

2.4. Integrity

We can use cryptography to provide a means to ensure data is not viewed or altered during storage or transmission. Cryptographic hashes for example, can safeguard data by providing a secure checksum. [1].

3. Cryptology

Cryptology is the mathematical science that has two branches: cryptography and cryptanalysis. Cryptography is a science of using mathematics to encrypt and decrypt data. It is also the study of mathematics of techniques related to aspects of Information security (Privacy, integrity and authenticity). Cryptanalysis is the study of encrypted information in order to discover the secret. The cryptanalysts are also called pirates. Those two branches will be explained latter.

In this domain, there are keywords used to define its many applications, they are as follow:

- **Plaintext:** readable and understandable data without specific intervention.
- **Encryption:** method to conceal the plaintext by hiding its contents. This operation allows ensuring that only persons to whom information is intended can access them.
- **Ciphertext:** unintelligible text resulting from encryption.
- **Decryption:** inverse transformation process of the encrypted plaintext.
- **Cryptosystem:** it is all possible keys (key space) of possible plaintexts and ciphertexts associated with a given algorithm.

Tuple (P, C, K, E, D), as: P: set of plain text.C: finite set of ciphertexts. K: Key space.

For each $k \in K$, there is an encryption function $e_k \in E$, and a corresponding decryption function $d_k \in D$, such as:

$$d_k(e_k(x)) = x, \text{ with } x \in P. \quad [2]. \quad (1)$$

4. Cryptography

Cryptography is the ancient science of encoding messages so that only the sender and receiver can understand them. It is now available to the entire world thanks to the development of modern computers, which can perform more mathematical operations in a second than a human being could do in a lifetime. An ordinary PC can produce codes of such complexity that the most powerful supercomputer using the best available attack algorithms would not break them in a million years. Cryptography is used to secure telephone, Internet, and email communication and to protect software and other digital property. [3]. Cryptography is made up of two halves: symmetric key encryption and public key encryption.

4.1. Symmetric Key Encryption

4.1.1. Principle

In Symmetric Key Encryption (also known as secret key, single key, shared key, one key or private key encryption), both the sender and the receiver share the same key used for both encryption and decryption of the data. In fact, the two keys may be identical or trivially related (i.e. there is a very simple transformation required to go between the two). In real life usage, a secret is being shared by two or more parties that can be used for the maintenance of a private link for communication. AES (Advanced Encryption Standard) is a very popular algorithm, which belongs to the family of symmetric key encryption algorithms. [4].

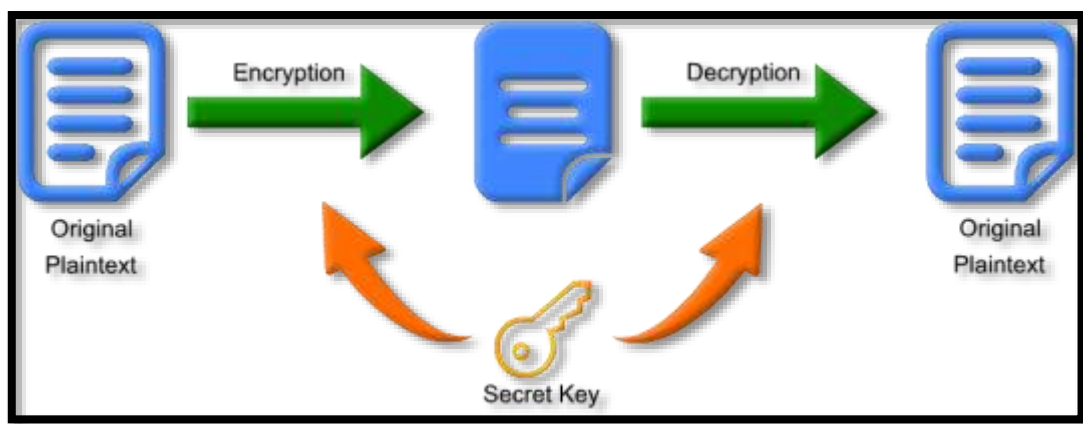


Figure I.1: Symmetric key encryption.

4.1.2. Small taxonomy of classic symmetric encryption

4.1.2.1. Caesar cipher

The Caesar Cipher is one of the simplest and most widely known encryption techniques. It is a form of substitution cipher in which each letter of the plaintext is replaced by a letter some fixed number of positions further down the alphabet. This technique is named after Julius Caesar, who used it with a left shift of 3 to protect messages of military significance.

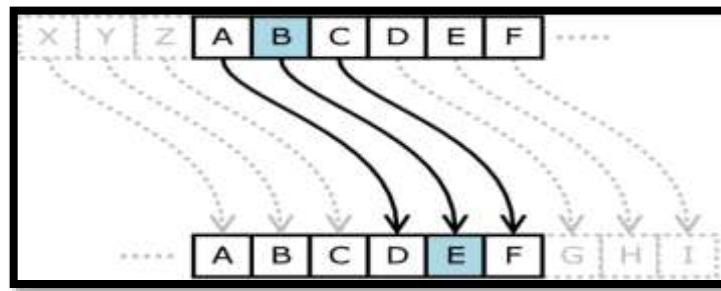


Figure I.2: Caesar cipher.

Decryption occurs by performing a shift of the same magnitude but in the opposite direction on each letter of the ciphertext. [5].

4.1.2.2. Vigenere cipher

The Vigenere Cipher is an encryption method that is based on a variation of the Caesar Cipher. It works by applying a series of different Caesar Ciphers on the plaintext, based on the letters of a so-called keyword. It is in fact a simple form of polyalphabetic substitution. [6]. The Vigenère Cipher exists in different forms, such as a rectangular matrix with 26 shifted alphabets (tabula recta) and as two concentric discs with a full alphabet each. The letters of the keyword determine how many places the inner disc should be shifted.



Figure I.3: Vigenere cipher.

During the course of history, the Vigenere Cipher has been reinvented many times. It was falsely attributed to Blaise de Vigenere as it was originally described in 1553 by Giovan Battista Bellaso. [6].

4.1.3. Substitution and transposition (or confusion and diffusion)

The substitutions include replacing symbols or groups of symbols by other symbols or groups of symbols in order to create confusion. Transposition consists of mixing symbols or groups of symbols of a clear message according to predefined rules to create the diffusion. These rules are determined by the encryption key. A series of transpositions forms a permutation.

4.1.4. Modern and powerful techniques

4.1.4.1. DES (Data Encryption Standard)

DES is the classic among cryptographic procedures and represents the creation of modern cryptography. It was developed at IBM as a result of a tender of the National Bureau of Standards (NBS, today NIST) for a uniform encryption standard and was presented in 1976. This presentation was a small sensation at that time, as DES was the first standardized and, above all, publicly made known encryption procedure for really high security requirements. Every 5 years, NIST certified the DES algorithm. The last certification was carried out in 1999, yet under the condition that the DES version Triple-DES is used, as DES no longer meets today's security requirements. [7].

The Data Encryption Standard (DES) was developed by an IBM team around 1974 and adopted as a national standard in 1977. Triple DES is a minor variation of this standard. It is three times slower than regular DES but can be billions of times more secure if used properly. Triple DES enjoys much wider use than DES because DES is so easy to break with today's rapidly advancing technology. In 1998 the Electronic Frontier Foundation, using a specially developed computer called the DES Cracker, managed to break DES in less than 3 days. And this was done for under \$250,000. The encryption chip that powered the DES Cracker was capable of processing 88 billion keys per second. [7].

In addition, it has been shown that for a cost of one million dollars a dedicated hardware device can be built that can search all possible DES keys in about 3.5 hours. This

just serves to illustrate that any organization with moderate resources can break through DES with very little effort these days. No sane security expert would consider using DES to protect data. [7].

4.1.4.2. Triple DES

Triple DES was the answer to many of the shortcomings of DES. Since it is based on the DES algorithm, it is very easy to modify existing software to use Triple DES. It also has the advantage of proven reliability and a longer key length that eliminates many of the shortcut attacks that can be used to reduce the amount of time it takes to break DES. However, even this more powerful version of DES may not be strong enough to protect data for very much longer. The DES algorithm itself has become obsolete and is in need of replacement. To this end the National Institute of Standards and Technology (NIST) is holding a competition to develop the Advanced Encryption Standard (AES) as a replacement for DES. Triple DES has been endorsed by NIST as a temporary standard to be used until the AES is finished sometime in 2001. [7].

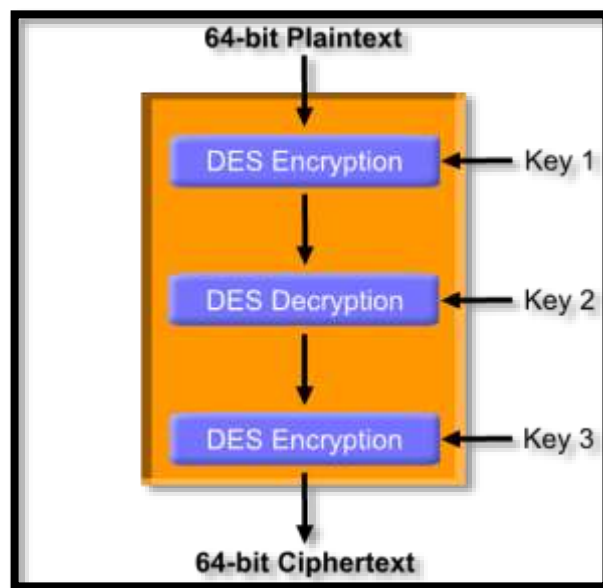


Figure I.4: Triple DES.

The AES will be at least as strong as Triple DES and probably much faster. Many security systems will use both Triple DES and AES for at least the next five years. After that, AES may supplant Triple DES as the default algorithm on most systems if it lives up to its

expectations. But Triple DES will be kept around for compatibility reasons for many years after that. So the useful lifetime of Triple DES is far from over, even with the AES near completion. For the foreseeable future Triple DES is an excellent and reliable choice for the security needs of highly sensitive information. [7].

4.1.4.3. AES (Advanced Encryption Standard)

The National Institute of Standards and Technology (NIST) established a DES successor as the new standard for symmetric encryption procedures: AES, Advanced Encryption Standard. The main objective is the establishment of a Federal Information Processing Standard (FIPS). On 12 September 1997, a formal tender for algorithms has been published. [7]. Certain conditions for the algorithm included, among others:

- a symmetric cryptographic procedure.
- a block cipher.
- a block length of 128 bits.
- a key size of 128, 192 and 256 bits.

The AES finalists of the third and final round were RC6, Rijndael, Serpent and Twofish. From these finalists Rijndael was selected on 2 October 2000 as AES. [7].

AES is a Federal Information Processing Standard (FIPS) and has been approved to be used by United States government organizations to protect sensitive, unclassified information. It is also widely adopted both commercially and globally. The AES implementation provided by Altera has been validated as conforming to the FIPS-197 standard. The National Security Agency (NSA) reviewed all the AES finalists, including Rijndael, and stated that all of them were secure enough for U.S. Government non-classified data. In June 2003, the U.S. Government announced that AES may be used to protect classified information. [7].

The design and strength of all key lengths of the AES algorithm (i.e., 128, 192 and 256) are sufficient to protect classified information up to the SECRET level. TOP SECRET information will require use of either the 192 or 256 key lengths. The implementation of AES in products intended to protect national security systems and/or information must be reviewed and certified by NSA prior to their acquisition and use. [7].

4.2. Public Key Encryption

In Public Key Encryption, two different but mathematically related keys are used. Public key encryption encrypts data using the recipient's public key, and it cannot be decrypted without using a matching private key. In other words, you need one key to lock (encrypt the plaintext) and another key to unlock (decrypt the ciphertext). Important thing is that one key cannot be used in the place of the other. Depending on which key is published, public key encryption can be used for two purposes. If the locking key is made public, then this system can be used by anybody to send private communication to the holder of the unlocking key. If it is the other way around, the system makes it possible to verify documents locked by the owner. Public key encryption is an asymmetric key algorithm. But only some asymmetric key algorithms have the special property of being unable to reveal one key with the knowledge of the other. So, the asymmetric key algorithms with this special property are called public key encryption algorithms. [4].

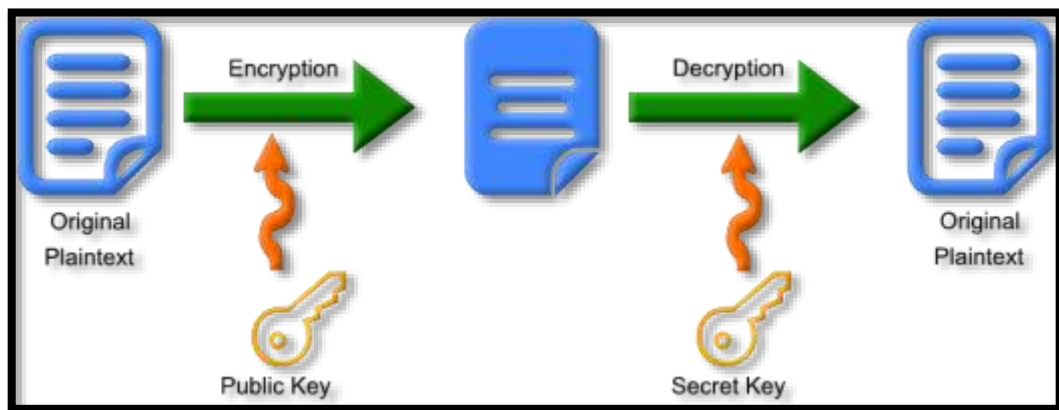


Figure I.5: Public Key Encryption.

4.3. Key security

The Key transmission security relies on two things: key distribution, which consists of distributing key on several parts, and key exchange that is done only between two parts. As explained in what follow.

4.3.1. Key Distribution

It depends on the type of Encryption method (symmetric key encryption or public key encryption)

4.3.1.1. Case of Symmetric Key Encryption

The users who use a common secret key:

- **Physically:** with a meeting, protected transmission channel..., etc.
- **A trusted third party:** it is chosen and it provide the key
- **Automatic key distribution to a user demand:** it requires total confidence in the system.
- **Use an old key to encrypt a new one:** the old key is related with the new one, so user can generate a new key by using old one.

4.3.1.2. Case of Public Key Encryption

It solves the problem of secret key distribution, but the problem continuous (user authentication related to the key). Four solutions allow the transfers:

- **Public announcement:** broadcast, mailing list.
- **Publicly available directory:** safer than the public announcement, but problem of vulnerability.
- **Public Key Authority:** enhance control of distribution from the directory. Request to an authority.
- **Public Key Certificates:** it is an electronic document used to prove ownership of a public key. [2].

4.3.2. Key Exchange

4.3.2.1. Diffie-Hellman protocol:

The Diffie-Hellman protocol is a method for two computer users to generate a shared private key with which they can then exchange information across an insecure channel. First, the two users agree on two prime numbers g and p , where p is large (typically at least 512 bits) and g is a primitive root modulo p . (In practice, it is a good idea to choose p such that $(p-1)/2$ is also prime). The numbers g and p need not to be kept secret from other users. Now First user chooses a large random number a as his private key and second user similarly chooses a large number b . first user then computes $A = g^a \pmod{p}$, which he sends to second user, and second user computes $B = g^b \pmod{p}$, which he sends to first user. Now both users compute their shared key $K = g^{ab} \pmod{p}$, which the first user computes as

$$K = B^a \pmod{p} = (g^b)^a \pmod{p}. \quad (2)$$

and the second user computes as

$$K = A^b \pmod{p} = (g^a)^b \pmod{p}. \quad (3)$$

The two users can now use their shared key K to exchange information without worrying about other users obtaining this information. In order for a potential eavesdropper (Eve) to do so, first user would first need to obtain $K = g^{ab} \pmod{p}$ knowing only g , p , $A = g^a \pmod{p}$ and $B = g^b \pmod{p}$. This can be done by computing a from $A = g^a \pmod{p}$ and b from $B = g^b \pmod{p}$. This is the discrete logarithm problem, which is computationally infeasible for large p . Computing the discrete logarithm of a number modulo p takes roughly the same amount of time as factoring the product of two primes the same size as p , which is what the security of the RSA cryptosystem relies on. Thus, the Diffie-Hellman protocol is roughly as secure as RSA. [8].

4.3.2.2. Quantum key Distribution:

Quantum Key Distribution (QKD) uses the properties of individual particles of light (photons) to establish a digital key between two communication partners. Based on the principles of quantum physics the information carried by a photon cannot be extracted unnoticed by the legitimate partners, as any extraction of information requires a measurement that modifies the properties of the photon. Additionally, the principles of quantum physics dictate that an identical photon cannot be produced by an eavesdropper. Therefore, any attempt to intercept will be detected by both parties and it is proven that the distribution of digital keys is absolutely secure (Information Theoretic Secure (ITS)) against eavesdropping.

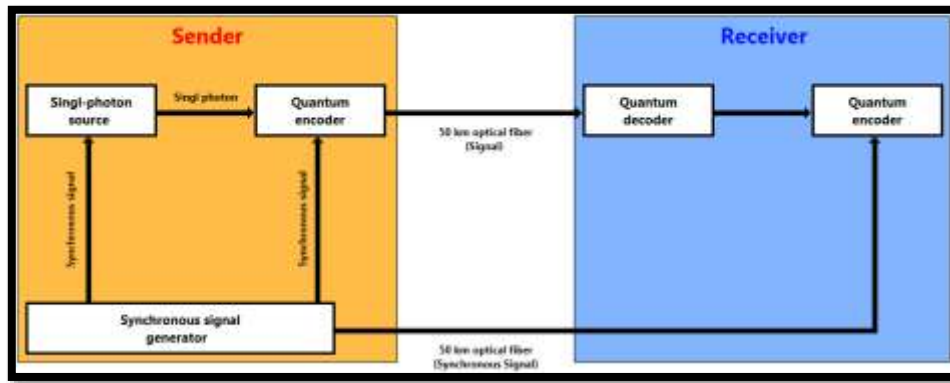


Figure I.6: Quantum key Distribution.

Consequently, messages encrypted with these keys using One-Time Pad encryption can withstand any attacks from arbitrary powerful computers (and even Quantum Computers). In addition secret keys generated with QKD can be used for ITS message authentication. Thus QKD gives us a tool for absolutely secure communication within the digital world. [9].

4.4. Difference between Symmetric Key and Public Key Encryption

There are two fundamental ways to use keys or secrets for encryption (symmetric and asymmetric). Symmetric encryption uses the identical key to both encrypt and decrypt the data. Symmetric key algorithms are much faster computationally than asymmetric algorithms as the encryption process is less complicated. The length of the key size is critical for the strength of the security. NIST has recommendations on how long a key should be— in general, 160-512 bits. There are inherent challenges with symmetric key encryption in that the key must somehow be managed. Distributing a shared key is a major security risk. [10].

Asymmetric encryption uses two related keys (public and private), and takes away the security risk of key sharing. The private key is never exposed. A message that is encrypted by using the public key can only be decrypted by applying the same algorithm and using the matching private key. Likewise, a message that is encrypted by using the private key can only be decrypted by using the matching public key. [10]. The next table demonstrate clearly the difference between the symmetric and asymmetric encryption:

Characteristics	Symmetric key	Public Key
Key used for encryption / decryption	Same key used for encryption and decryption	One key used for encryption and another, different key is used for decryption
Key size	Less than 512 bits	Could be more than 512 bits
Speed of encryption / decryption	Very fast	Slower
Size of resulting encrypted text	Usually same as or less than the original clear text size	More than the original clear text size
Key agreement / exchange	A big problem	No problem at all
Usage	Mainly for encryption and decryption (confidentiality), cannot be used for digital signatures (integrity and non-repudiation checks)	Can be used for encryption and decryption (confidentiality) as well as for digital signatures (integrity and non-repudiation checks)

Table I.1: Difference between Symmetric Key and Public Key.

4.5. Complexity of cryptographic algorithms

The complexity theory provides a mechanism to analyze the computational complexity of the cryptographic algorithms, which makes it possible to make a comparison between their security levels. This theory helps cryptanalysts to make predictions about the time required to break the cryptographic algorithms. The computational complexity of an algorithm is measured by two factors:

- T: time complexity.
- S: The space complexity (the necessary memory space).

The complexity generally depends on a parameter n that represents the size of the input. [11].

4.6. Encryption modes

There are two different types of symmetric algorithms in the way of dealing with plaintext and the ciphertext, and how they are treated and manipulated. They are explained as follow:

4.6.1. Block cipher mode

The basic idea of a block cipher is to divide text in relatively large blocks, typically 64 or 128 bits long, and encode each block separately. The same encryption key is used for each block and it is the encryption key that determines the order in which substitution, transportation and other mathematical functions are performed on each block. Strong algorithms mean that reverse engineering the cipher, or determining which functions were performed on each block, in which order, virtually impossible. [12]. There are five different methods used on block ciphers, they are called operation modes:

4.6.1.1. Electronic Code Book (ECB) mode

It is the simplest mode of encryption. Each plaintext block is encrypted separately. Similarly, each ciphertext block is decrypted separately. Thus, it is possible to encrypt and decrypt using many threads simultaneously. In this mode, the created ciphertext is not blurred.

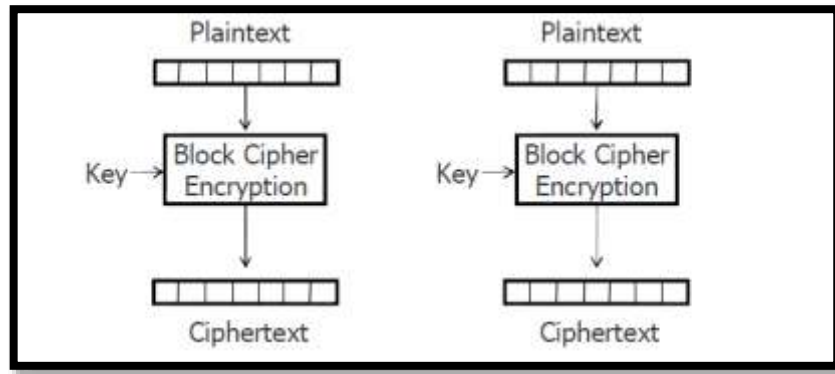


Figure I.7: Electronic Code Book (ECB) mode.

Messages that are encrypted using ECB mode should be extended until a size that is equal to an integer multiple of the single block length. The popular method of aligning the length of the last block is about appending an additional bit equals to 1 and then filling the rest of the block with bits equal to 0. It allows to determine precisely the end of the real message. Ciphers that are used in ECB mode are more vulnerable to replay attacks. [13].

4.6.1.2. Cipher Block Chain (CBC) mode

The CBC mode of encryption was invented by IBM in 1976. It is about to add XOR each subsequent plaintext block to a ciphertext block that was previously received. The result is encrypted using a cipher's algorithm in the usual way. Each subsequent ciphertext block depends on the previous one. The first plaintext block is added XOR to a random initialization vector (commonly referred to as IV). The vector has the same size as all plaintext blocks. Encryption in CBC mode can be performed only using one thread. Despite this disadvantage, it is a very popular way of encrypting, which is used in various applications. During decrypting ciphertext blocks, one should add XOR output data from decryption algorithm to previous ciphertext blocks. The receiver knows all ciphertext blocks just after obtaining encoded the message, thus he can decrypt the message using many threads simultaneously.

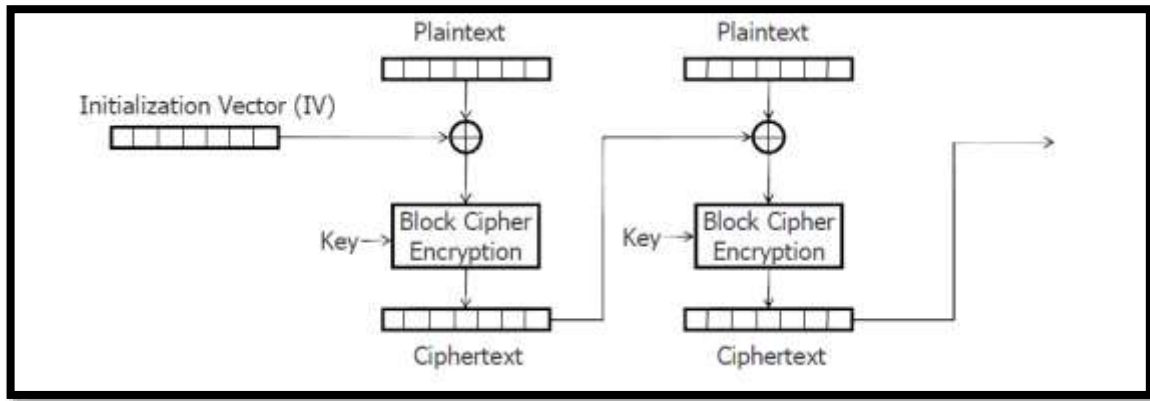


Figure I.8: Cipher Block Chain (CBC) mode.

If one bit of a plaintext message is damaged (for example because of transmission error), all subsequent ciphertext blocks will be damaged and it will not be possible to decode the ciphertext in the future. As opposed to that, if one ciphertext bit is damaged, only two received plaintext blocks will be damaged. A message that is to be encrypted using CBC mode, should be extended until a size that is equal to an integer multiple of the single block length (as during using ECB mode). [13].

4.6.1.3. Cipher FeedBack (CFB) mode

The CFB mode is similar to the previously described CBC mode. The main difference is that one should encrypt mixed data from the previous round (so not plaintext blocks) and then add to plaintext bits. It does not affect the security strength but it results in using cipher's encryption algorithms (the same that were used for encrypting plaintext) during decryption process.

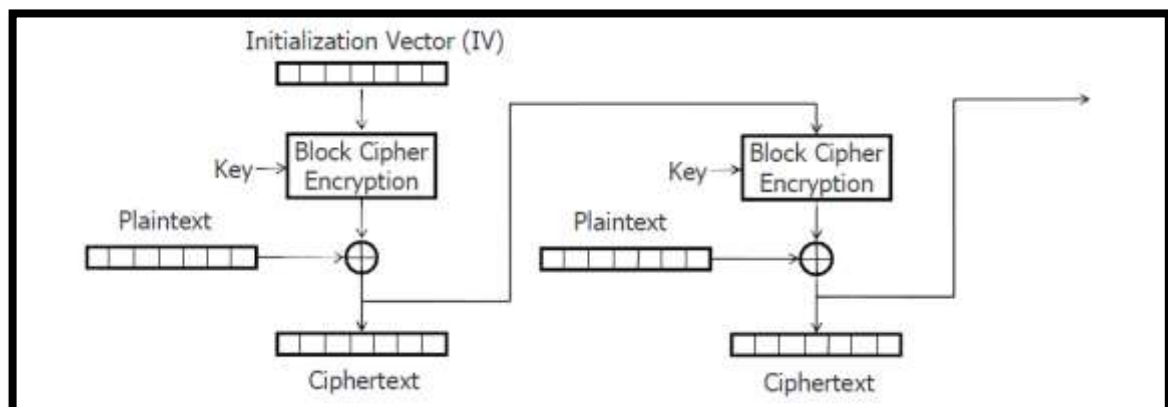


Figure I.9: Cipher FeedBack (CFB) mode.

If one bit of a plaintext message is damaged, the corresponding ciphertext block and all subsequent ciphertext blocks will be damaged. Encryption in CFB mode can be performed only using one thread. On the other hand, as in CBC mode, one can decrypt ciphertext blocks using many threads simultaneously. Similarly, if one ciphertext bit is damaged, only two received plaintext blocks will be damaged. As opposed to the CBC mode, the encrypted message does not need to be extended until a size that is equal to an integer multiple of the single block length. [13].

4.6.1.4. Output FeedBack (OFB) mode

Algorithms that work in the OFB mode create keystream bits that are used for encryption subsequent data blocks. In this regard, the way of working of the block cipher becomes similar to the way of working of a typical stream cipher.

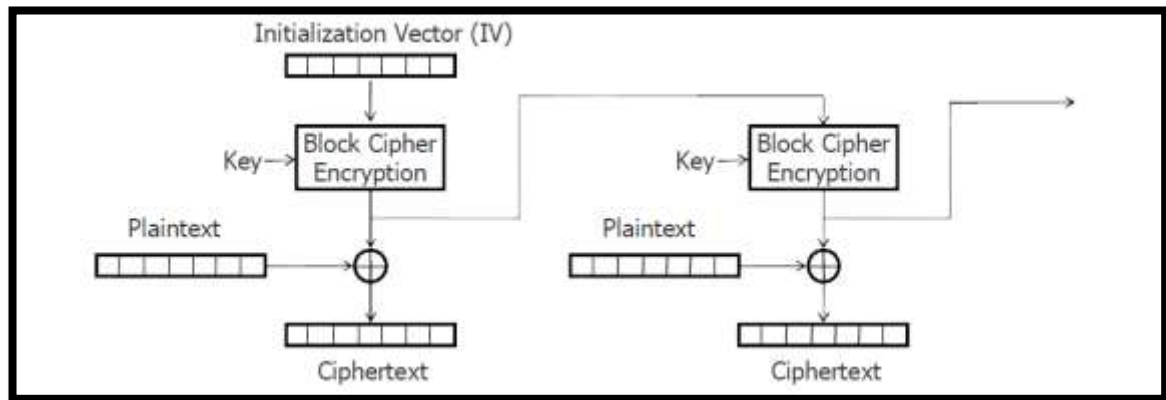


Figure I.10: Output FeedBack (OFB) mode.

Because of the continuous creation of keystream bits, both encryption and decryption can be performed using only one thread at a time. If one bit of a plaintext or ciphertext message is damaged (for example because of transmission error), only one corresponding ciphertext or respectively plaintext bit is damaged as well. It is possible to use various correction algorithms to restore the previous value of damaged parts of the received message. [13].

4.6.1.5. Counter (CTR) mode

Using the CTR mode makes block ciphers' way of working similar to stream ciphers' way of working. As in the OFB mode, keystream bits are created regardless of content of encrypted data blocks. In this mode, subsequent values of an increasing counter are added

to a nonce value and the results are encrypted as usual. The nonce plays the same role as initialization vectors in the previous modes.

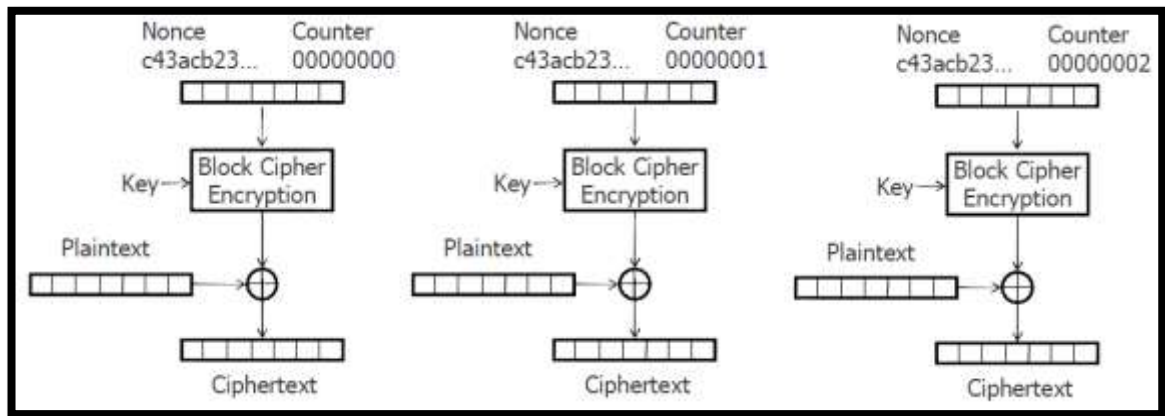


Figure I.11: Counter (CTR) mode.

It is one of the most popular block ciphers modes of operation. Both encryption and decryption can be performed using many threads at the same time. If one bit of a plaintext or ciphertext message is damaged, only one corresponding output bit is damaged as well. Thus, it is possible to use various correction algorithms to restore the previous value of damaged parts of received messages. [13].

4.6.2. Stream cipher mode

The basic idea of a stream cipher is to divide text into very small blocks, one bit or one byte long, and encode each block depending on many previous blocks. Stream ciphers use a different encryption key -- a value which must be fed into the algorithm -- for each bit or byte, so the same bit or byte produces different ciphertext each time it is encrypted. Some stream ciphers use a keystream generator, which produces a random, or nearly random, stream of bits. The cipher performs a Boolean operation, known as an exclusive OR, between the bits in the keystream and the bits in the plaintext to produce ciphertext. [12].

4.7. Hash function

Hash is a kind of process or function, which is responsible for translating information into a cryptic value. The concept of hash and encryption is almost same. In practical view Hash is an algorithm that takes an arbitrary block of data and returns a fixed-size bit string.

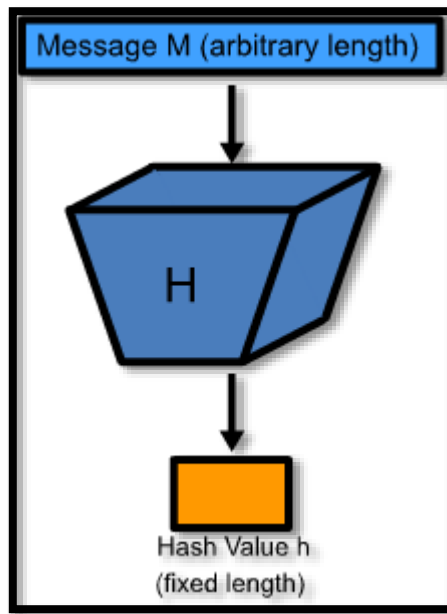


Figure I.12: Hash function.

Hashing is also known for its unidirectional process because it does not require rehashing or decrypting to get back data. In hashing the data which is needed to be encoded is often called the message, and the outcome of hash value after processing is sometimes called the message digest or simply digests. While hashing message, an algorithm is utilized which works to map input values to a series of known output values. So given the same series of input values, a hash algorithm always produces the same output values. Hashing is an industry supported standard similar to encryption. [2].

4.8. Digital signature

A digital signature is the term used for marking or signing an electronic document, by a process meant to be analogous to paper signatures, but which makes use of a technology known as public key cryptography. Additional security properties are required of signatures in the electronic world. This is because the probability of disputes rises dramatically for electronic transactions without face-to-face meetings, and in the presence of potentially undetectable modifications to electronic documents. Digital signatures address both of these concerns, and offer far more inherent security than paper signatures. Compared to all other forms of signatures, digital signatures are by far the most easily verified and the most reliable with respect to providing document integrity. [14].

5. Cryptanalysis

Cryptanalysis is the science of cracking codes and decoding secrets. It is used to violate authentication schemes, to break cryptographic protocols, and, more benignly, to find and correct weaknesses in encryption algorithms. [15]. Among the types of attacks are:

5.1. Ciphertext only attacks (COA)

A ciphertext only attack (COA) is a case in which only the encrypted message is available for attack, but because the language is known a frequency analysis could be attempted. In this situation the attacker does not know anything about the contents of the message, and must work from ciphertext only. [15].

5.2. Known plaintext attacks (KPA)

In a known plaintext attack (KPA) both the plaintext and matching ciphertext are available for use in discovering the key. [15].

5.3. Chosen plaintext attacks (CPA)

A chosen plaintext attack (CPA) occurs when the attacker gains access to the target encryption device - if, for example, it is left unattended. The attacker then runs various pieces of plaintext through the device for encryption. This is compared to the plaintext to attempt to derive the key. [15].

5.4. Chosen ciphertext attacks (CCA)

In a chosen ciphertext attack (CCA), the cryptanalyst can choose different ciphertexts to be decrypted and has access to the decrypted plaintext. This type of attack is generally applicable to attacks against public key cryptosystems. [15].

5.5. Man-in-the-middle attacks (MITMA)

Abbreviated as MITMA, a man-in-the-middle attack is an attack where a user gets between the sender and receiver of information and sniffs any information being sent. In some cases, users may be sending unencrypted data, which means the man-in-the-middle (MITM) can obtain any unencrypted information. In other cases, a user may be able to obtain information from the attack, but have to unencrypt the information before it can be read. In the picture below is an example of how a man-in-the-middle attack works. The attacker

intercepts some or all traffic coming from the computer, collects the data, and then forwards it to the destination the user was originally intending to visit. [16].

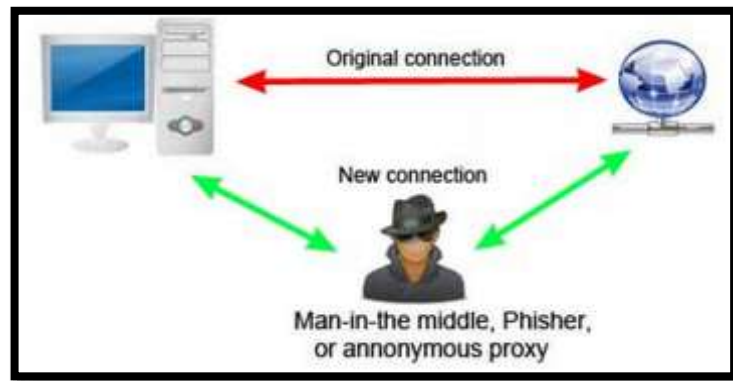


Figure I.13: Man-in-the-middle attacks (MITMA).

5.6. Side Channel Attacks (SCA)

Side channel attacks are closely related to the existence of physically observable phenomena caused by the execution of computing tasks in present microelectronic devices. For example, microprocessors consume time and power to perform their assigned tasks. They also radiate an electromagnetic field, dissipate heat, and even make some noise. As a matter of fact, there are plenty of information sources leaking from actual computers that can consequently be exploited by malicious adversaries. [17]. Some of the most common "side-channel" attack vectors are:

- Timing
- Power monitoring
- Error handling analysis

5.7. Brute Force Attacks (BFA)

A brute force attack involves trying all possible keys until hitting on the one that results in plaintext. This can involve significant costs related to the amount of processing required to try quadrillions (in the case of DES) of keys. The time required is a factor of how many keys can be tried per unit of time, which is a factor of how many computers can be assigned to the task in parallel. [15].

6. Conclusion

In this first chapter, we presented the fundamental aspects of cryptology with its two disciplines: cryptography and cryptanalysis, as well as its main objectives. Then we discussed the symmetric and asymmetric cryptography, while we explain their operation through examples of ancient and modern cryptographic algorithms. Then we presented the design of cryptographic methods of block encrypting. Finally, we presented some cryptanalytic attacks, which are considered the most powerful tools to measure the robustness of cryptographic methods.