

## 1. Introduction

The use of cryptography can be traced back to the time of the Egyptians. Since 1960, it was used by the governments and military to protect secrets and national strategies. With the expansion of the internet in early 1990, cryptography played an important role in the internet security, (safety data communication over internet).

The oxford dictionary gives the following definition of the term cryptography «a secret way of writing, either by arbitrary characters, by using letters or characters other than their ordinary sense or by other method clear only to those possessing the key, also anything like in this way, generally, the art of writing or solving ciphers ». [11]

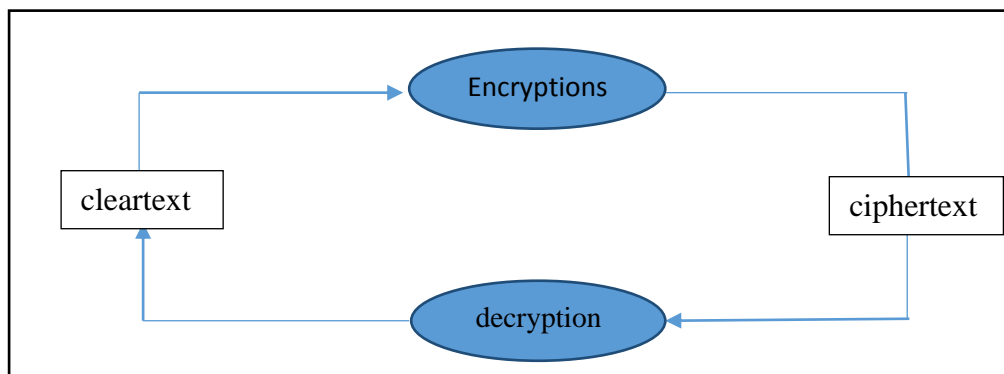
In this chapter, we present and explain the different models of cryptography systems, and we deal with the fundamental rules of cryptography, which are necessary to protect sensitive data and offer effective security, we also study some public-key cryptography system like (RSA, ECC, Elgamal....) and hashing secure technique, which we use in our application.

## 2. Definition

Cryptography is a science of using mathematics to encrypt and decrypt data, so cryptography enables to store sensitive information and transmit it over security network (like the internet), hence that cannot be read by anyone except the intended recipient. [12]

Otherwise cryptography is the science and study of secret writing, where cleartext is transformed into ciphertext. The process of transforming cleartext into ciphertext called encryption the reverse process is called decryption, a couple of process are controlled by cryptographic key or keys.

Cryptanalysis is science and study of methods of breaking ciphers and secure communication. Classical cryptanalysis includes a combination of analytical logic, and application of mathematical tools, pattern finding, determination, and luck. Cryptoanalysts also called attackers. [13]



**Figure 2.1.** Secret writing [13]

### 3. basic Terminology[25]

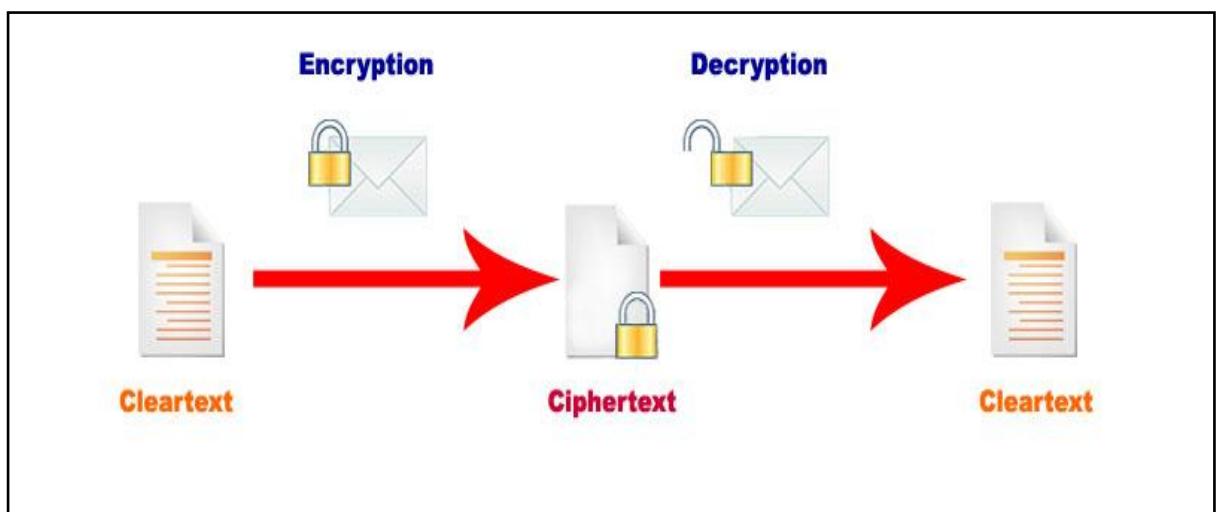
- ✓ Plaintext : the original message
- ✓ Ciphertext : the coded message
- ✓ Cipher : algorithm for transforming plaintext to ciphertext
- ✓ key :info used in cipher known only to sender/receiver
- ✓ Cryptanalysis (codebreaking) : the study of principles/ methods of deciphering ciphertextwithout knowing key
- ✓ Cryptology : the field of both cryptography and cryptanalysis.

### 4. Functionality of cryptography work

A cryptographic algorithm, or cipher, isa process of decoding and encoding a messageso that its meaning isnot clear. A cryptographic algorithm works in combination with a key(word, number, or phrase) to encrypt the plaintext. The same plaintext encrypts to different cipher text with different keys. The security of encrypted data is entirely dependent on two things: the strength of the cryptographic algorithm and the secrecy of the key [12].

### 5. Encryption and decryption

Data that can be read and understood without any special measures is called clear text. The method of disguising plaintext in such a way as to hide its substance is called encryption. Encrypting plaintext results in unreadable gibberish called cipher text. You use encryption to ensure that information is hidden from anyone for whom it is not intended, even those who can see the encrypted data. The process of reverting cipher text to its original plaintext is called decryption.This process is illustrated in (Figure2.2) [12].



**Figure 2.2.**Encryption and decryption.

## 6. Common goals of cryptography[11]

In essence, cryptography concerns four main goals:

### 6.1. Message confidentiality

Only an authorized recipient should be able to extract the contents of the message from its encrypted form. Resulting from steps to hide, stop or delay free access to the encrypted information.

### 6.2. Message integrity

The recipient should be able to determine if the message has been altered.

### 6.3. Authentication

The recipient should be able to verify from the message, the identity of the sender, the origin or the path (or combinations) so to validate claims from emitter or to validate the recipient expectations.

### 6.4. Non-repudiation

The emitter should not be able to reject sending the message. Not all cryptographic systems realize all of the above goals. Some applications of cryptography have different goals, for example, some situations require repudiation where a participant can plausibly deny that they are a sender or receiver of a message, or extend these goals to include variations like:

- ✓ Message access control: Who are the valid recipients of the message?
- ✓ Message availability: By providing means to limit the validity of the message, channel, emitter or recipient in time or space.

## 7. Keys

A key is a value that works with a cryptographic algorithm to produce a specific cipher text, keys are essentially, big number. The key size is measured in bits. The number representing a 1024-bit key is darn huge. In public key cryptography, is the bigger the key is, and the more secure for the cipher text is, however the public key size and the traditional cryptography's secret key size are totally unrelated. A conventional 80-bit key has the equivalent strength of a 1024-bit public key. A conventional 128-bit key is equivalent to a 3000-bit public key, again the bigger the key is the more secure, but the algorithms used for each type of cryptography are very different.

While the public and private key are mathematically related, it is very difficult to derive the private key given only the public key. However deriving the private key is always possible given enough time and computing power. This make it very important to pick keys of the right size, large enough to be secure but small enough to be applied quickly. Additionally you need to consider who might be trying to read your files, how determined they are, how much time they have, and what their resources might be. Larger keys will be cryptographically secure for a longer period of time. If what you want to encrypt needs to be hidden for many years, you might want to use a very large key, of course, who knows how long it will take to determine your key using tomorrow's faster, more efficient computers? There was a time when a 56-bit symmetric key was considered very safe[12].

## 8. Hash functions

### 8.1. Defintion

Hashing produces a value or values from a string of text using a mathematical function. Hashing is one way to enable security during the process of message transmission. When the message is intended for a particular recipient only, the hashing function helps to protect the security of the transmission from unauthorized users. Hashing is also a method of storing sensitive values in a database table in an efficient manner.

When the user sends a secure message, a hash of the intended message is generated and encrypted and sent along with the message, when the message is recieved, the receiver decrypts the hash as well as the message then, the receiver creates another hash from the message if the two hashes are similar when compared, hence we can say a secure transmission has occurred[14].

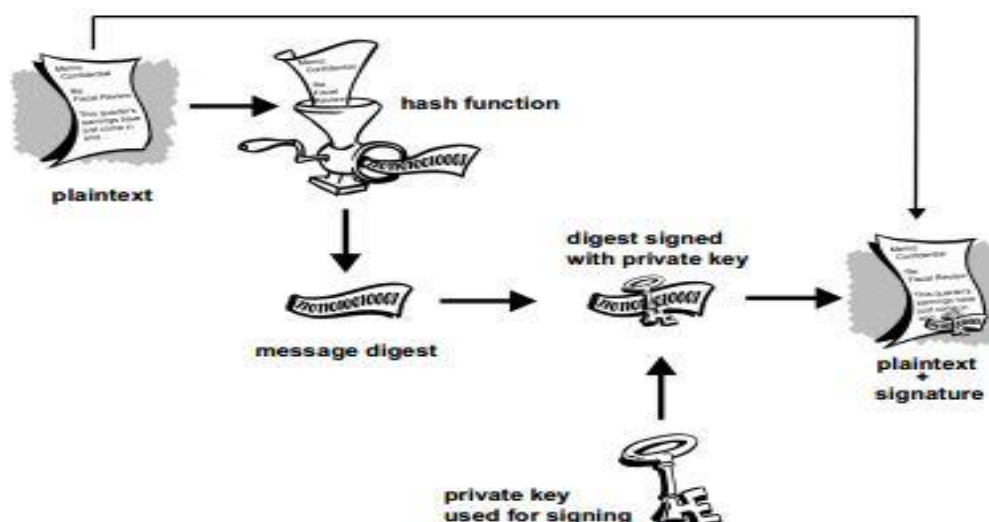


Figure 2.3. Hashing function [12].

## 8.2.The hash function properties[24]

In summary a cryptographic hash function needs to satisfy the following three properties:

- ✓ It should be hard to find a message with a given hash value.
- ✓ It should be hard to find two messages with the same hash value.
- ✓ Given one message it should be hard to find another message with the same hash value.

## 8.3.Application of hash function in cryptography

### 8.3.1.Password Hashing:

A password, in computer science, is a secret sequence of character that one usesto gain access to a file, an application (application web) or a computer system.Today, many web applications use a database to store and retrieve a variety of data including passwords, Fortunately, some web applications generate a hash value of allpasswords and store these hash values, instead of the password itself, in the database.We show below a sample that explains how we use a hashing secure technique in order to store a password in a database and describe the process which happened during the login phase[24]:



**Figure 2.4.**Login interface [24].

All passwords in this application are hashed with the md5 algorithm and the resulting hash value is stored in the database.

id	name	username	email	password	usertype
62	Administrator	admin	admin@email.com	11f953d5321942a7f01e4317d718bec	Super Administrator

**Figure 2.5.**Hash of password stored in database[24].

When a user enters his/her credential at the backend login page, the password entered in the cleartext is first hashed with the md5 algorithm and the output is compared to the value of the hashed password stored in the database for which the usernames are identical. If the two strings match then access is granted, otherwise access is denied.

**Remark:**In this part focus is only on Hash values of password, in order to gain more enough information to secure the password, without ignoring other uses of hashing such as Trusted Digital Time-Stamping, and File Integrity Verification...

## 9. Digital signatures

A digital signature is a property private to a user or process that is used for signing messages. Let B be the recipient of message M signed by A, then A's signature must satisfy these requirements [13,23] :

- ✓ B must be able to validate A's signature on M.
- ✓ It must be impossible for anyone, including B to forge A's signature.

A digital signature, therefore, establishes the sender authenticity, it is similar to a normal written signature. It also establishes data authenticity.

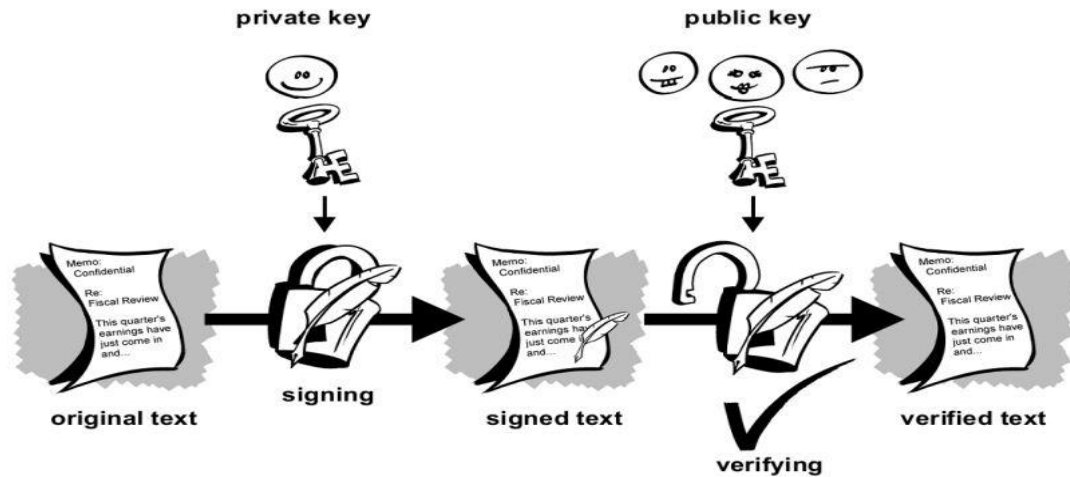
Public-key authentication systems provide a simple scheme for implementing digital signatures. Because the transformation  $D_A$  is private to A,  $D_A$  serves as A's digital signature. The recipient B of a message M signed by A is assured of both sender and data authenticity.

It is impossible for B or anyone else to forge A's signature on another message, and impossible for A to disclaim a signed document. Because the inverse transformation  $E_A$  is public, the receiver B can readily validate the signature.

While traditional systems such as the DES provide data authenticity, they do not in themselves provide sender authenticity. Because the sender and receiver share the same key, the receiver could forge the sender's signature. It is possible to implement digital signatures in conventional systems using a trusted third party S.

The following approach was suggested by Merkle [Merk80]. Each user A registers a pair of private transformations  $E_A$  and  $D_A$  with S, where  $E_A(D_A(M)) = M$  for every message M. To send a signed message M to B, A computes  $C = D_A(M)$ , and transmits C to B. To check the validity of C and obtain M, B sends C to S.

S computes  $E_A(C) = M$  and returns M to B enciphered under B's private transformation.



**Figure 2.6.**Digital signature process. [12]

The RSA encryption algorithm is particularly interesting since it can be used directly as a signature algorithm with message recovery:

- ✓ The sender applies the RSA decryption transformation to generate the signature, by taking the message and raising it to the private exponents.
- ✓ The receiver then, applies the RSA encryption transform to recover the original message.

## 10.Symmetric cryptography

Conventional encryption has benefits, It is very fast and useful for encryption data that is not going anywhere, However, conventional encryption alone as a means for transmitting secure data can be very expensive simply due to the difficulty of secure key to the sender and recipient to communicate securely. By using conventional encryption they must accept a key and keep it secret between themselves. If they are in different physical locations they must trust a courier (bag), the Bat phone or some other secure communication medium to prevent the detection of the secret key during transmission.

Anyone who intercepts the key in transit can later read, modify, and forge all information encrypted or authenticated with that key, from DES to captain midnight's secret decoder ring. The persistent problem with conventional encryption is key distribution: How do you get the key to the recipient without someone intercepting it? [11]

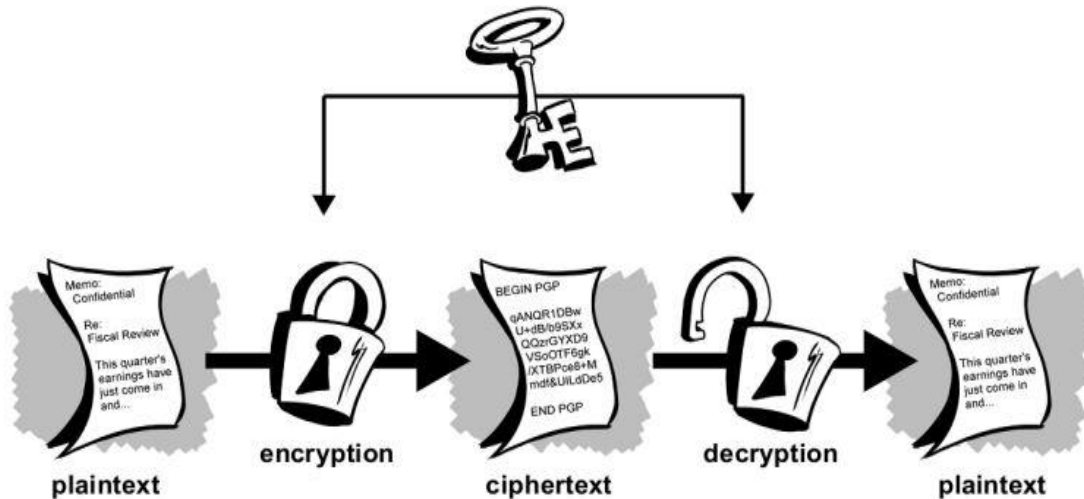


Figure 2.7.Tradional encryption [12]

## 10.1. Key Management

To be able to use symmetric encryption algorithms such as DES or AES, we need a way for the two communicating parties to share the secret key. Hence in this part we discuss the different types of keys[15]:

### 10.1.1. Static (or long-term) Keys

These are keys which are to be in use for a long timeperiod. The exact definition of long will depend on the application, but this could mean from a few hours to a few years. The compromise of a static key is usually considered to be a major problem, with potentially catastrophic consequences.

### 10.1.2. Ephemeral, or Session (or short-term) Keys

These are keys which have a short life-time, from a few seconds to a day. They are usually used to provide confidentiality for the given time period. The compromise of a session key should only result in the compromise of that session's secrecy and it should not affect the long-term security of the system.



## 11.Public key cryptograhy

Public key cryptography is an asymmetric scheme that uses a pair of keys for encryption,a public key which encrypts data and a corresponding private or secret key for decryption.You publish your public key to the world while keeping your private key secret.

Anyone with a copy of your public key can then encrypt information that you can read only.

So, the one 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.

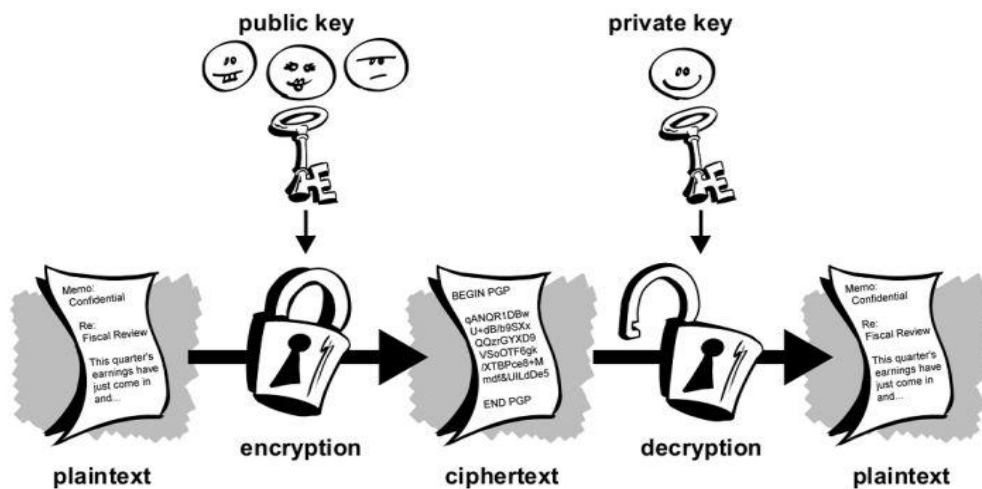


Figure 2.8.public key encryption [12]

The principal goal of the public key cryptography is that it allows people who have no pre-existing security arrangement to exchange messages securely.The need for sender and receiver to share a secret key via some secure channel is eliminated.All communications have only a public key and private key.Some examples of public key cryptosystems are Elgamal, RSA, ECC, NTRU [12].

### 11.1. The RSA cryptosystem[16]

The different observations just stated form the base for the RSA public-key cryptosystem, were invented at MIT in 1977 by Ronald Rivest, Adi Shamir and Leonard Adleman.

The public key in this cryptosystem consists of the value  $n$ , which is called the modulus, and the value  $e$ , which is called the public exponent. The private key consists of the modulus  $n$  and the value  $d$ , which is called the private exponent.

An RSA public-key / private-key pair can be generated by the following steps:

- ✓ Generate a pair of large, Radomprimes  $p$  and  $q$ .

- ✓ Calculate the modulus  $n = p \cdot q$ .
- ✓ Select an odd public exponent  $e$  between 3 and  $n-1$  that is relatively prime to  $p-1$  and  $q-1$ .
- ✓ Compute the private exponent  $d$  from  $e$ ,  $p$  and  $q$ .
- ✓ Output  $(n, e)$  as the public key and  $(n, d)$  as the private key.

#### 11.1.1. The RSA encryption

The encryption operation in the RSA cryptosystem is an exponentiation to the  $e$  the power modulo  $n$ :

$$C = \text{ENCRYPT}(m)^e = m \bmod n.$$

The input  $m$  is the message the output  $c$  is the resulting ciphertext, in practice the message  $m$  is typically some kind of appropriately formatted key to be shared. The actual message is encrypted with the shared key using a traditional encryption algorithm. This construction makes it possible to encrypt a message of any length with only one exponentiation.

#### 11.1.2. The RSA decryption

The decryption operation is exponentiation to the  $d$  the power modulo  $n$ :

$$M = \text{DECRYPT}(C, d) = C \bmod n.$$

The relationship between the exponents  $e$  and  $d$  ensure that encryption and decryption are inverses, so that the decryption operation recovers the original message  $m$ . Without the private key  $(n, d)$  (or equivalently the prime factors  $p$  and  $q$ ), it is difficult to recover  $m$  from  $c$ . Consequently,  $n$  and  $e$  can be made public without compromising security, which is the basic requirement for a public-key cryptosystem.

#### 11.1.3. The RSA signature

The fact that the encryption and decryption operations are inverses and operate on the same set of inputs also means that the operations can be employed in reverse order to obtain a digital signature scheme following Diffie and Hellman's model. A message can be digitally signed by applying the decryption operation to it, i.e. by exponentiating it to the  $d^{\text{th}}$  power:

$$S = \text{SIGN}(m) = m^d \bmod n.$$

#### 11.1.4. The RSA signature verifying

The digital signature can then be verified by applying the encryption operation to it and comparing the result with and/or recovering the message:

$$M = \text{VERIFY}(S) = S^e \bmod n.$$

Key pair Public key : $n=55$ , $e=3$ private key : $n=55$ , $d=7$			Key pair generation Primes : $p = 5$ , $q = 11$ Modulus : $n = p*q = 55$ Public exponent : $e = 3$ Private exponent : $d = 3^{-1} \bmod 20 = 7$			
Message	Encryption $c = m^3 \bmod n$		Decryption $m = c^7 \bmod n$			
M	$m^2 \bmod n$	$m^3 \bmod n$	$C^2 \bmod n$	$C^3 \bmod n$	$C^6 \bmod n$	$C^7 \bmod n$
0	0	0	0	0	0	0
1	1	1	1	1	1	1
2	4	8	9	17	14	2
3	9	27	14	48	49	3
4	16	9	26	14	31	4
5	25	15	5	20	15	5
6	36	51	16	46	26	6

Table.2.1. RSA example.

### 11.2. ECC cryptosystem

Elliptic curve cryptosystem ECC include key distribution, encryption and digital signature algorithms. The key distribution algorithms use to share a secret key, the encryption algorithms enable confidential communication while the digital signature used to authenticate the signer and validate the integrity of the message.

Also, the feature of ECC is the compared with RSA cryptosystems that can provide the same security level with a short key length because of this mathematical property ECC they are faster and need less hardware than RSA. The security of an ECC, however, depends not only on the length of the key but also on the elliptic curve parameters. So, faster parameters generation is important for practical implementation of an ECC[17].

**11.2.1. ECC encryptions/ decryption**

Suppose entity A wants to send an encrypted message  $x$  to entity B. Thus entity B chooses a large prime  $p$  and an integer  $a \bmod p$ . Also, entity B chooses a secret integer  $i$  and computes: Entity B then makes  $c, a, p$  public and keeps  $i$  secret, Entity A chooses a random  $k$  and computes:

$$y_1 \equiv a^k \bmod p$$

$$y_2 \equiv c^k \bmod p$$

Entity A sends  $(y_1, y_2)$  to entity B, which decrypt by calculating  $x = y_1 * y_2 \bmod p$ , Now we describe the elliptic curve version. Entity B chooses an elliptic curve  $E \bmod p$  where  $p$  is a large prime. Entity B chooses a point  $a$  on  $E$  and a secret integer  $i$ . Entity B computes  $c = a * a$  ( $= a + a + \dots + a$ ). The points  $a$  and  $c$  are made public, while  $i$  kept secret.

Entity A expresses its message as a point  $x$  on  $E$ . Then entity A chooses a random integer  $k$ , computes  $y_1 = k * a \bmod p$  and  $y_2 = x + k * c$  then sends the pair  $(y_1, y_2)$  to Entity B that it decrypts by calculating  $x = y_2 - a * y_1$  [11].

**11.2.2. Signature generation [18]**

For signing a message  $m$  by sender A, using private key  $D_a$

- ✓ Calculate  $e = \text{hash}(m)$ , where hash is cryptography hash function such as SH-1.
- ✓ Select a random integer  $K$  from  $[1, n-1]$ .
- ✓ Calculate  $r = x_1 \bmod n$ , where  $(x_1, y_1) = K * G$  ( $G$  is the generator point, an elliptic curve domain parameter).
- ✓ Calculate  $s = k^{-1}(e + d_A r) \bmod n$ .

**11.2.3. Signature verification**

For B to authenticate A's signature, B must have A's public key  $Q_A$

- ✓ Verify that  $r$  and  $s$  are integers in  $[1, n-1]$ . If not, the signature is invalid.
- ✓ Calculate  $e = \text{HASH}(m)$ , where HASH is the same function used in the signature generation.
- ✓ Calculate  $w = s^{-1} \bmod n$ .
- ✓ Calculate  $u_1 = e w \bmod n$  and  $u_2 = r w \bmod n$ .
- ✓ Calculate  $(x_1, y_1) = u_1 G + u_2 Q_A$ .
- ✓ The signature is valid if  $x_1 = r \bmod n$ , invalid otherwise. [18, 17]

### 11.3. TheElgmal cryptosystem

We now consider the Elgmal cryptosystem, which was named by, Taher Elgmal in 1984, which is based on the difficulty of a problem called the discrete logarithm (DL).

The original public key system proposed by Die and Hellman needs interaction of both parties to calculate a common private key, This poses problems if the cryptosystem should be applied to communication systems where both parties are not able to interact in reasonable time due to delays in transmission or unavailability of the receiving party .

Thus ElGamal simplified the Die-Hellman key exchange algorithm by inserting a random exponent  $k$ , This exponent is an replacement for the private exponent of the receiving entity. Due to this simplification the algorithm can be used to encrypt in one direction, without the necessity of the second party to take actively part, The key advance here is that the algorithm can be used for encryption of electronic messages, which are transmitted by the means of public store-and-forward services[22].

#### 11.3.1.DL - Discrete logarithme problem

When we are working with the real numbers, logby is the value  $x$ , such that  $b^x = y$ . We can define an analogous discrete logarithm. Given integers  $b$  and  $n$ , with  $b < n$ , the discrete logarithm of an integer  $y$  to the base  $b$  is an integer  $x$ , such that:

$$b^x \equiv y \pmod{n}$$

The discrete logarithm is also called index and we write:

$$x = \text{ind}_{b,n} y.$$

While it is quite efficient to raise numbers to large powers modulo  $p$  (recall the repeated squaring algorithm), the inverse computation of the discrete logarithm is much harder. The ElGamal system relies on the difficulty of this computation.

#### 11.3.2.Elgame encryption

Let  $p$  be a prime and  $g$  be a generator of  $\mathbb{Z}_p$ . The private key  $x$  is an integer between 1 and  $p - 2$ . Let  $y = g^x \pmod{p}$ . The public key for ElGamal encryption is the triplet  $(p, g, y)$ , if taking discrete logarithms is as difficult as it is widely believed, releasing  $y = g^x \pmod{p}$  does not reveal  $x$ .

To encrypt a plaintext  $M$ , a random integer  $k$  relatively prime to  $p - 1$  is selected, and the following pair of values is computed:

$$a \leftarrow g^k \pmod{p}$$

$$b \leftarrow My^k \pmod{p}$$

The ciphertext  $C$  consists of the pair  $(a, b)$  computed above.

### 11.3.3. Elgamal decryption

The decryption of the cipher text  $C = (a, b)$  in the Elgamal scheme, to retrieve the plaintext  $M$ , is simple:

$$M \leftarrow b/a^x \bmod p$$

In the above expression, the “division” by  $a^x$  should be interpreted in the context of modular arithmetic, that is,  $M$  is multiplied by the inverse of  $a^x$  in  $\mathbb{Z}_p$ . The correctness of the ElGamal encryption scheme is easy to verify. Indeed, we have:

$$\begin{aligned} b/a^x \bmod p &= My^k (a^x) - 1 \bmod p \\ &= Mg^{xk} (g^{kx})^{-1} \bmod p \\ &= M. \end{aligned}$$

### 11.3.4. Elgamal signature

A variation of the above scheme provides a digital signature. Namely, a signature for message  $M$  is a pair  $S = (a, b)$  obtained by selecting a random integer  $k$  relatively prime to  $p - 1$  and computing:

$$\begin{aligned} a &\leftarrow g^k \bmod p \\ b &\leftarrow k^{-1} (M - xa) \bmod (p - 1) \end{aligned}$$

### 11.3.5. Elgamal signature verification

To verify a digital signature  $S = (a, b)$ , we check that:

$$Ya^b \equiv g^M \pmod{p}$$

Example:

Alice chooses  $P_A = 107$ ,  $\alpha_A = 2$ ,  $d_A = 67$ , and she computes  $\beta_A = 2^{67} \equiv 94 \pmod{107}$ . Her public key is  $(P_A, \alpha_A, \beta_A) = (107, 2, 94)$ , and her private key is  $d_A = 67$ . Bob wants to send the message "B" (66 in ASCII) to Alice. He chooses a random integer,  $k = 45$  and encrypts  $M = 66$  as  $(r, t) = (\alpha_A^k, \beta_A^k M) \equiv (2^{45}, 94^{45} 66) \equiv (28, 9) \pmod{107}$ . He sends the encrypted message  $(28, 9)$  to Alice. Alice receives the message  $(r, t) = (28, 9)$ , and using her private key  $d_A = 67$  she decrypts to  $tr^{-d_A} = 9 \cdot 28^{-67} \equiv 9 \cdot 28^{106-67} \equiv 9 \cdot 43 \equiv 66 \pmod{107}$ .

## 12. Conclusion

We presented in this chapter some cryptography systems by taking into consideration their different processes like encryption, decryption and digital signature. We focused on public key cryptography. In addition, we dealt with the conventional cryptography and Hash functions, in preparation for their implementation in our application.