3. Conventional Encryption Models

3.1 The basic Communication Scenario for Cryptography

In the basic communication scenario, depicted in figure 3.1, there are two parties, we'll call them Alice and Bob, who want to communicate with each other. A third party, Oscar, is a potential eavesdropper. When Alice wants to send a message, called the **plaintext,** to Bob, she encrypts it using a method prearranged with Bob. Usually, the encryption method is assumed to be know to Oscar; what keeps the message secret is a **key.** When Bob receives the encrypted message, called the **ciphertext,** he changes it back to the plaintext using a decryption key. Oscar could have one of the following goals:

1. Read the message.
2. Find the key and thus read all messages encrypted with that key.
3. Corrupt Alice's message into another message in such a way that Bob will think Alice sent the alternated message.
4. Masquerade as Alice, and thus communicate with Bob even though Bob believes he is communicating with Alice.

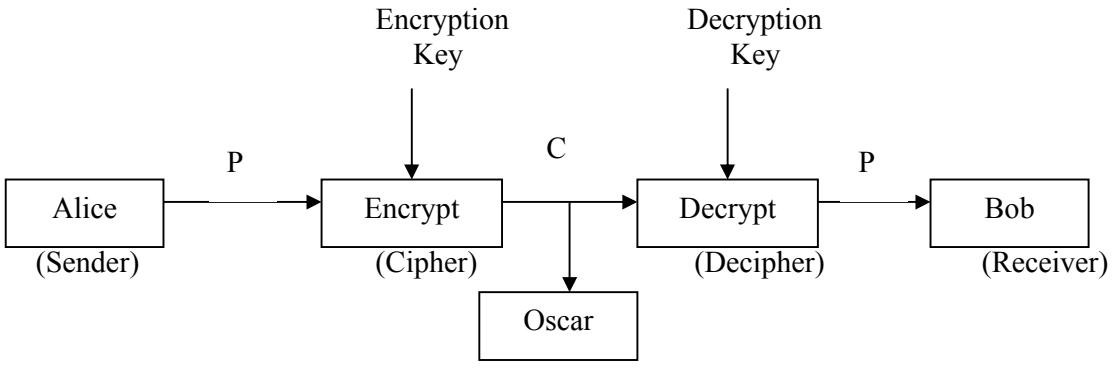

Figure3.1. The basic Communication Scenario for Cryptography

P- plaintext
C- ciphertext
$C = e_k(P)$
$\mathrm{P} = \mathrm{d}_\mathrm{k}(\mathrm{C})$
$d_k = decryptrom\,key$
A more active and malicious adversary, corresponding to cases (3) and (4), is sometimes called Mallory in the literature. More passive observers (as in cases (1) and (2)) are sometimes named Oscar. We'll generally use only Eve, and assume she is as bad as the situation allows.

3.2 Goals of Cryptography

-***Fundamental objective***- to enable Alice and Bob to communicative over an insecure channel such that Oscar can not understand what is being said;
1. Confidentiality- secrecy of data (historical goal); that the data is not understood by anyone other than the intended receiver

2. Data Integrity- prevents unauthorized alteration of data; Bob must be able to detect data manipulation (i.e., deletion, substitution).
3. Authentication- identification of both parties (the sender and the receiver should identify each other) and of information (origin of data, data content, time sent, etc.)
- *data origin authentication*- verifies the source of data, information about origin of the data, creator, time of creation.
-*entity authentication*- verifies the identity of the other party; i.e., ensures that you are not talking to an impostor
4. Non-repudiation- prevents a party from denying previous actions.
Example Alice can not claim she does not send message.

3.3 Security Attacks

Security Attacks- specifies whether the adversary interface or not with the information
-*passive*- the goal is to obtain the information transmitted
-release of message contents - e.g., from a telephone conversation, e-mail, transferred files, etc.
-traffic analysis - e.g., location and identity of communicating hosts, frequency and length of messages, the nature of messages.
-*active attacks* - involves some modification of the data stream
-masquerade - pretending to be a different entity
-reply - passive capture of a data unit and subsequent retransmission
-modification of messages
-denial of service
Passive attacks are difficult to detect but easy to prevent whereas active attracts are easy to detect but difficult to prevent.
Security attacks are can also be divided into *on-line* and *off-line.*

Types of attacks – specifies the information available to the adversary
-*ciphertext – only* – the adversary possesses only a string of chiphertext
-*known paintext* – the adversary possesses a string of plaintext and the corresponding ciphertext.
-*chosen ciphertext*- the adversary selects a string of plaintext and then obtains the corresponding plaintext.
The attacks can also be classified by the approach used unto
-*cryptanalysis*- when the attack relies on the nature of the algorithm plus some information as the ones above and
-*brute force*- when all keys (on average half) are tired until a good one is found. Below are some estimates on the time needed by brute force attacks for various key sizes and speeds.

Key size (bits)	Number of keys	Time (1 encryption/ μs)	Time (10^6 encryption/ μs)
32	$2^{32} \approx 4.3\times10^{9}$	$2^{31}\mu s \approx 35.8$ min	$\approx 2.15\mu s$
56	$2^{56} \approx 7.2\times10_{16}$	$2^{55}\mu s \approx 1142$ years	≈ 10.01 hours
128	$2^{128} \approx 3.4\times10^{38}$	$2^{127}\mu s \approx 5.4\times10^{24}$ years	$\approx 5.4\times10^{18}$ years
168	$2^{168} \approx 3.7\times10^{50}$	$2^{167}\mu s \approx 5.9\times10^{36}$ years	$\approx 5.9\times10^{30}$ years
26 characters	$26! \approx 4\times10^{26}$	$2\times10^{26}\mu s \approx 6.4\times10^{12}$ years	$\approx 6.4\times10^{6}$ years

It is important to mention that trying a key does not mean only decrypting using that key but also identifying whether the obtained plaintext is the valid one. For instance, if a random (meaningless) sequence of bits is encrypted, then it is impossible to decrypt simply because even after all keys are tired the attacker does not know which one is the correct plaintext.

Adversarial goal – specifies what it means for the adversary to 'break' the system
-*complete break*- find out the key
-*partial break*- decrypt some ciphertex
-*distinguish ability*- distinguish between valid ciphertext and random strings

Security Level – specifies the computational resources available to the adversary. An encryption scheme is **unconditionally secure** if the ciphertex generated by the scheme does not contain enough information to determine uniquely the corresponding plaintext. That is, no matter how much time an opponent has, it is impossible to decrypt the chiphertext. With the exception of a scheme known as the one-time pad (described later in this chapter).
An encryption scheme is said to be **computationally secure** if the foregoing two criteria are met.

- The cost of breaking the cipher exceeds the value of the encrypted information.
- The time required to break the cipher exceeds the useful lifetime of the information.

-*provable secure*- the difficulty of breaking a system is shown to be essentially as difficult as solving a well-known (supposedly) difficult problem (usually number-theoretic). One of the most important assumption in modern cryptography is that the adversary can establish encryption method (Kerckhoffs Principle)

Types of ciphers
-by types of operations
-Substitution- each element of the plaintext (bit, letter, group of bits or letter) is mapped into another element
-transpositions- elements of plaintext are rearranged
-by number of keys used
-one for both sender and receiver- symmetric encryption (see below)
-two different keys- public – key encryption (see below)
-by the way the plaintext is processed
-*block cipher*- one block of the input prod uses one block in the output
-*stream cipher*- the input is processed continuously producing one elements of the output at a time.

3.4. Symmetric-key (Private-key) encryption

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). This was the only kind of encryption publicly known until June 1976.

-for any pair (e_K, d_K), it is computationally easy to determine d_k knowing only e_k
-both must be secret.
$e_k=d_k$ or $d_k=F(e_k)$

It is important to note that security of depends encryption on the secrecy of the key, not the servecy of algorithm. We do not need to keep the algorithm secrecy; we need to keep only the secrecy key.

In symmetric key algorithm, the encryption and decryption keys are known to both Alice and Bob. For example the encryption key is shared and the decryption key is easily calculated from it. In many cases the encryption key and the decryption key are the same. All of the classical (pre-1970) cryptosystems are symmetric.

The modern study of symmetric-key ciphers relates mainly to the study of block ciphers, stream ciphers and application of hash function

. A block ciphers take as input a block of plaintext and a key, and output a block of ciphertext of the same size.. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are symmetric encryption techniques which have been designated cryptography standards by the US government (though DES's designation was finally withdrawn after the AES was adopted).

Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit-by-bit or character-by-character. RC4 is a widely used stream cipher.

Cryptographic hash functions are a third type of cryptographic algorithm. They take a message of any length as input, and output a short, fixed length hash which can be used in (for example) a digital signature. For good hash functions, an attacker cannot find two messages that produce the same hash. MD4 is a long-used hash function which is now broken; MD5, a strengthened variant of MD4, is also widely used but broken in practice.

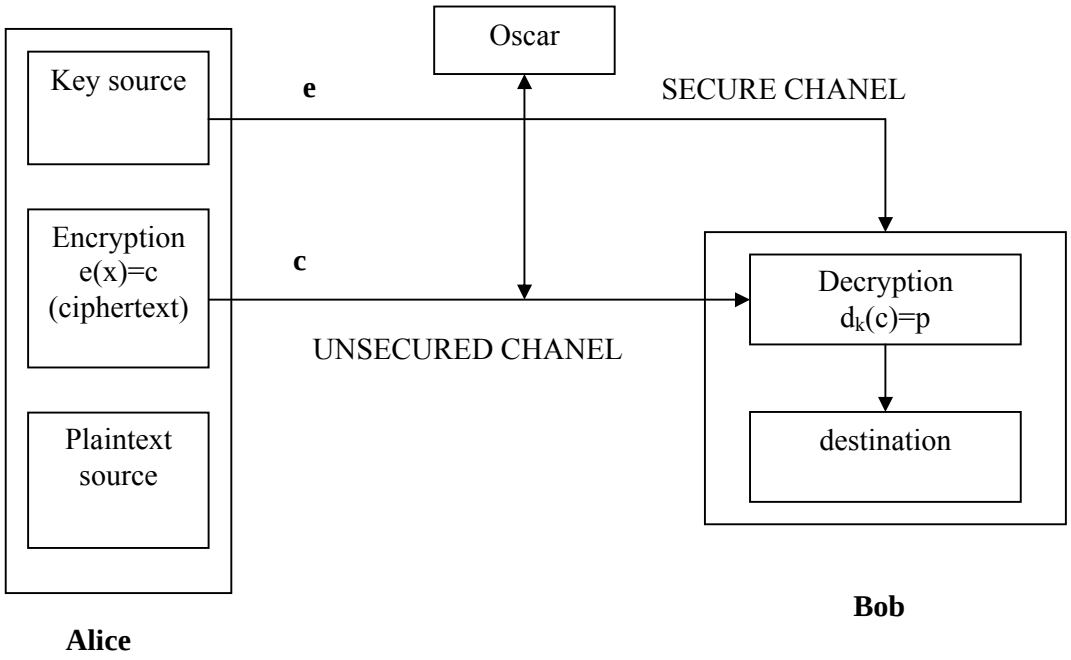

Figure 3. 2. Symmetric key encryption

3.5. Public-key encryption

The public key algorithm introduced in 1976 by Diffie and Helman.

-for any pair (e_k, d_k), it is computationally infeasible to determine d_k knowing e_k. public key cryptography uses two keys, in contrast to the symmetrical encryption which uses only one key.
-e_k can be made public. Only Bob can decrypt
$c=e_k(P)$

d_k= Private

$P=d_k(C)$ e_k= Public

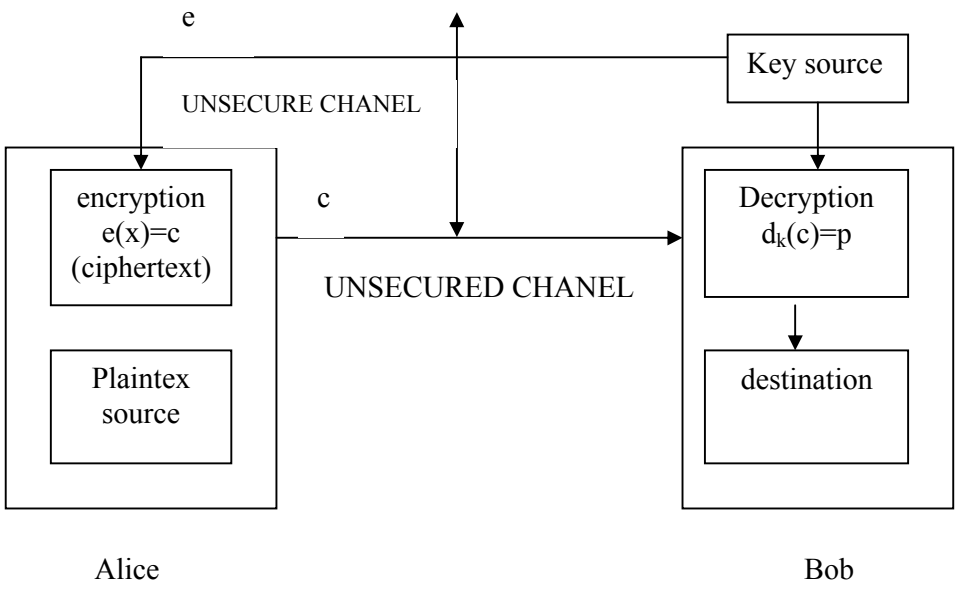

Figure 3. 3. Encryption using public-key techniques

A significant disadvantage of symmetric ciphers is the key management necessary to use them securely. Each distinct pair of communicating parties must, ideally, share a different key, and perhaps each ciphertext exchanged as well. The number of keys required increases as the square of the number of network members, which very quickly requires complex key management schemes to keep them all straight and secret. The difficulty of securely establishing a secret key between two communicating parties, when a secure channel doesn't already exist between them, also presents a chicken-and-egg problem which is a considerable practical obstacle for cryptography users in the real world.

In a groundbreaking 1976 paper, Whitfield Diffie and Martin Hellman proposed the notion of *public-key* (also, more generally, called *asymmetric key*) cryptography in which two different but mathematically related keys are used — a *public* key and a *private* key. A public key system is so constructed that calculation of one key (the 'private key') is computationally infeasible from the other (the 'public key'), even though they are necessarily related. Instead, both keys are generated secretly, as an interrelated pair. The historian David Kahn described public-key cryptography as "the most revolutionary new concept in the field since polyalphabetic substitution emerged in the Renaissance".

In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. The *public key* is typically used for encryption, while the *private* or

secret key is used for decryption. Diffie and Hellman showed that public-key cryptography was possible by presenting the Diffie-Hellman key exchange protocol.

In 1978, Ronald **R**ivest, Adi **S**hamir, and Len **A**dleman invented RSA, another public-key system, known as RSA-encryption algorithm.

The Diffie-Hellman and RSA algorithms, in addition to being the first publicly known examples of high quality public-key algorithms, have been among the most widely used. Others include the ElGamal encryption, and various elliptic curve techniques.

In addition to encryption, public-key cryptography can be used to implement digital signature schemes. A digital signature is reminiscent of an ordinary signature; they both have the characteristic that they are easy for a user to produce, but difficult for anyone else to forge. Digital signatures can also be permanently tied to the content of the message being signed; they cannot then be 'moved' from one document to another, for any attempt will be detectable. In digital signature schemes, there are two algorithms: one for *signing*, in which a secret key is used to process the message (or a hash of the message, or both), and one for *verification,* in which the matching public key is used with the message to check the validity of the signature. RSA and DSA are two of the most popular digital signature schemes. Digital signatures are central to the operation of public key infrastructures and many network security schemes.

Public-key algorithms are most often based on the computational complexity of "hard" problems, such as **trapdoor one-way functions**. For example, the hardness of RSA is related to the integer factorization problem, while Diffie-Hellman and DSA are related to the discrete logarithm problem. More recently, *elliptic curve cryptography* has developed in which security is based on number theoretic problems involving elliptic curves. Because of the difficulty of the underlying problems, most public-key algorithms involve operations such as modular multiplication and exponentiation, which are much more computationally expensive than the techniques used in most block ciphers, especially with typical key sizes. As a result, public-key cryptosystems are commonly hybrid cryptosystems, in which a fast high-quality symmetric-key encryption algorithm is used for the message itself, while the relevant symmetric key is sent with the message, but encrypted using a public-key algorithm. Similarly, hybrid signature schemes are often used, in which a cryptographic hash function is computed, and only the resulting hash is digitally signed.

Remark: The **trapdoor one-way** function
-one-way- $y=f(x)$ is easy to compute but $f^{-1}(y)$ is computationally infeasible
-trapdoor one-way- a one-way function with the property that given some additional information (trapdoor information) it becomes feasible to compute $f^{-1}(y)$,
Exaple: discrete logarithm problem
find x from $a^x = \log_m$ if a, b, m, known factorization of prime numbers.
Ex: $2^k = 19 \text{MOD} 19$ $k=1,2,3,\ldots, 2^k \bmod 19 = 14$

Example 2: factorization of prime numbers.
-what are the factors of 2624653723? (answer: 48611 and 53993)