INTRODUCTION

Computer data often travels from one computer to another, leaving the safety of its protected physical surroundings. Once the data is out of hand, people with bad intention could modify or forge your data, either for amusement or for their own benefit. Cryptography can reformat and transform our data, making it safer on its trip between computers. The technology is based on the essentials of secret codes, augmented by modern mathematics that protects our data in powerful ways.

- **Computer Security** - generic name for the collection of tools designed to protect data and to thwart hackers
- **Network Security** - measures to protect data during their transmission.
- **Internet Security**- measures to protect data during their transmission over a collection of interconnected networks.

Basic Concepts

Cryptography: The art or science encompassing the principles and methods of transforming an intelligible message into one that is unintelligible, and then retransforming that message back to its original form.

Plaintext: The original intelligible message.

Cipher text: The transformed message.

Cipher: An algorithm for transforming an intelligible message into one that is unintelligible by transposition and/or substitution methods.

Key: Some critical information used by the cipher, known only to the sender& receiver.

Encryption (Encipher or encode): The process of converting plaintext to cipher text using a cipher and a key.

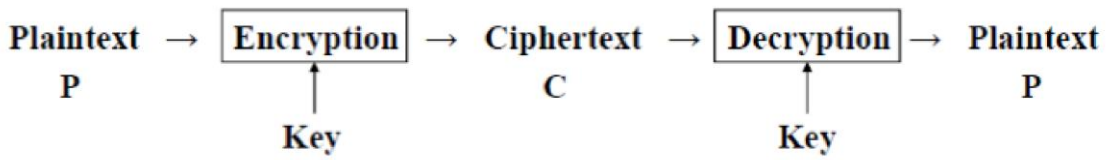
Decryption (Decipher or decode): The process of retrieving the plaintext from the ciphertext using a cipher and a key.

Cryptanalysis (code breaking): The study of principles and methods of transforming an unintelligible message back into an intelligible message *without* knowledge of the proper key.

Cryptology: Both cryptography and cryptanalysis.

Cryptographer: is person who does cryptography.

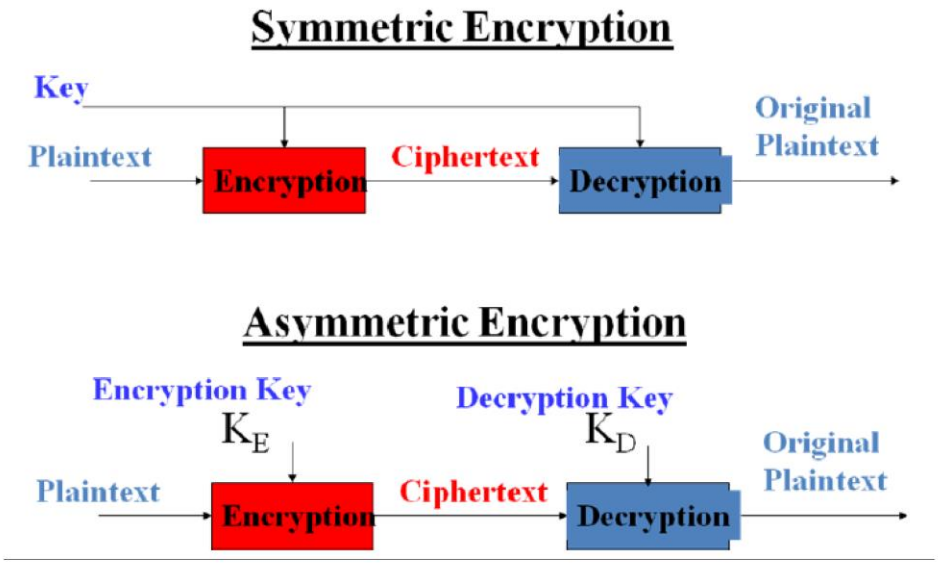
Cryptanalyst: is a person practitioner of cryptanalysis.



Cryptography

Cryptographic systems are generally classified along 3 independent dimensions:

- Type of operations used for transforming plain text to cipher text
  All the encryption algorithms are based on two general principles:
  - ✓ **Substitution**, in which each element in the plaintext is mapped into another element.
  - ✓ **Transposition**, in which elements in the plaintext are rearranged.
- The number of keys used
  - ✓ If the sender and receiver uses same key then it is said to be symmetric key (or secret-key or single key or conventional encryption).
  - ✓ If the sender and receiver use different keys then it is said to be asymmetric (or public key) encryption.

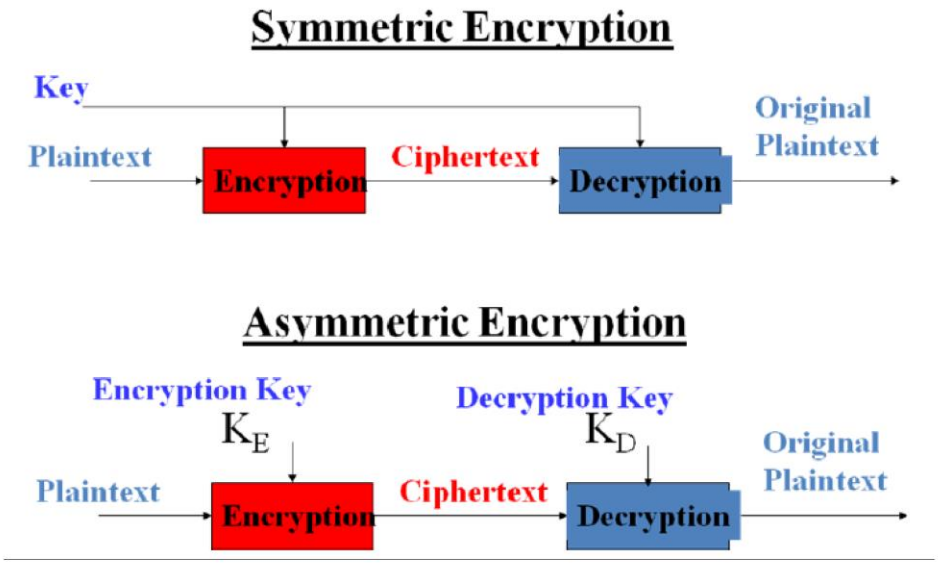


- The way in which the plain text is processed
  - ✓ A block cipher processes the input as block of elements at a time, producing output block for each input block.
  - ✓ A stream cipher processes the input elements continuously, producing output element one at a time, as it goes along.

Cryptography

Cryptographic systems are generally classified along 3 independent dimensions:

- Type of operations used for transforming plain text to cipher text
  All the encryption algorithms are based on two general principles:
  - ✓ **Substitution**, in which each element in the plaintext is mapped into another element.
  - ✓ **Transposition**, in which elements in the plaintext are rearranged.
- The number of keys used
  - ✓ If the sender and receiver uses same key then it is said to be symmetric key (or secret-key or single key or conventional encryption).
  - ✓ If the sender and receiver use different keys then it is said to be asymmetric (or public key) encryption.



- The way in which the plain text is processed
  - ✓ A block cipher processes the input as block of elements at a time, producing output block for each input block.
  - ✓ A stream cipher processes the input elements continuously, producing output element one at a time, as it goes along.

Security Attacks, Services and Mechanisms

To assess the security needs of an organization effectively, the manager responsible for security needs some systematic way of defining the

requirements for security and characterization of approaches to satisfy those requirements. One approach is to consider three aspects of information security:

Security attack – Any action that compromises the security of information owned by an organization.

Security mechanism– A mechanism that is designed to detect, prevent or recover from a security attack.

Security service – A service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks and they make use of one or more security mechanisms to provide the service.

Security Services

The classification of security services are as follows:

1. **Availability**: Requires that computer system assets be available to authorized parties when needed.
2. **Authentication**: - assurance that the communicating entity is the one claimed.
3. **Access Control**:- prevention of the unauthorized use of a resource.
4. **Data Confidentiality**:- protection of data from unauthorized disclosure.
5. **Data Integrity**: - assurance that data received is as sent by an authorized entity.
6. **Non-Repudiation**: - protection against denial by one of the parties in a communication.

Security Mechanisms

One of the most specific security mechanisms in use is cryptographic techniques.

Encryption or encryption-like transformations of information are the most common means of providing security. Some of the mechanisms are

1. Cryptography.

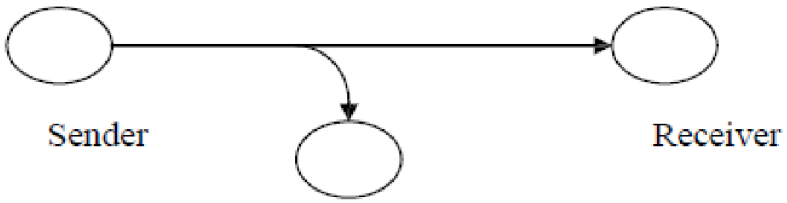
2. Digital Signature.

3. Access Control.

Security Attacks

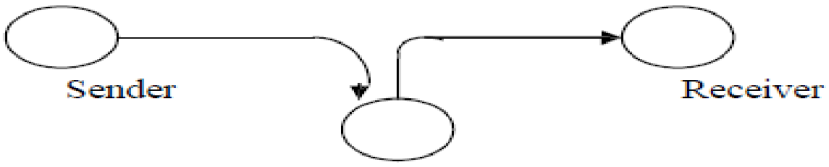
There are four general categories of attack which are listed below:

1. **Interruption**
   An assets of the system is destroyed or becomes unavailable or unusable. This is an attack on availability e.g., destruction of piece of hardware, cutting of a communication line or Disabling of file management system.
2. **Interception**
   An unauthorized party gains access to an asset. This is an attack on confidentiality.
   Unauthorized party could be a person, a program or a computer. e.g., wire tapping to capture data in the network, illicit copying of files.



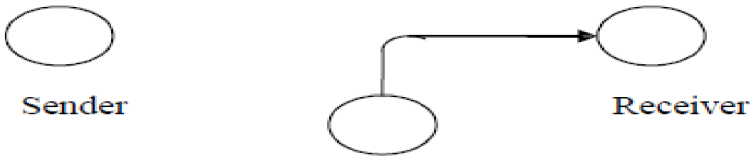
Eavesdropper or forger

3. **Modification**
   An unauthorized party not only gains access to but tampers with an asset. This is an attack on integrity. e.g., changing values in data file, altering a program, modifying the contents of messages being transmitted in a network.



Eavesdropper or forger

4. **Fabrication**
   An unauthorized party inserts counterfeit objects into the system. This is an attack on authenticity. e.g., insertion of spurious message in a network or addition of records to a file.



Eavesdropper or forger

Cryptographic Attacks

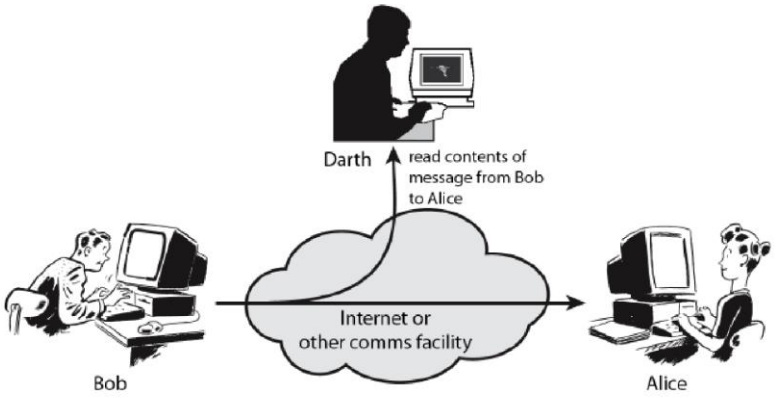
There are two types of cryptographic attacks:

1- Passive Attacks

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. Passive attacks are of two types:

- **Release of message contents**: A telephone conversation, an e-mail message and a transferred file may contain sensitive or confidential information. We would like to prevent the opponent from learning the contents of these transmissions.
- **Traffic analysis**: If we had encryption protection in place, an opponent might still be able to observe the pattern of the message. The opponent could determine the location and identity of communication hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of communication that was taking place.

Passive attacks are very difficult to detect because they do not involve any alteration of data. However, it is feasible to prevent the success of these attacks.



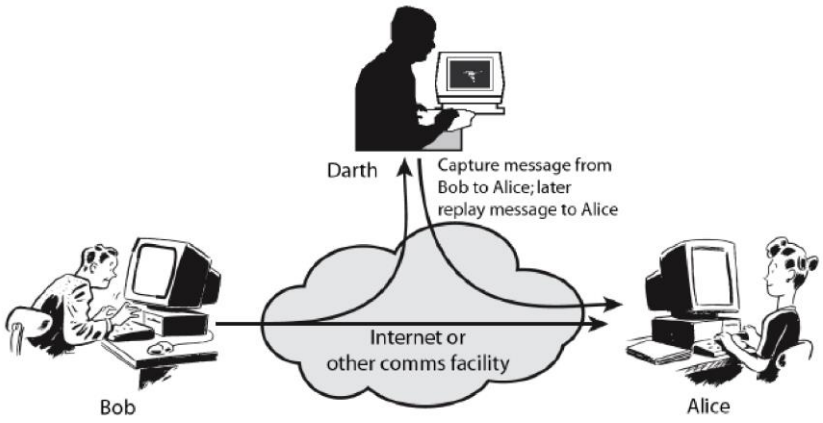
Passive Attacks

2- Active attacks

These attacks involve some modification of the data stream or the creation of a false stream. These attacks can be classified in to four categories:

- **Masquerade** – One entity pretends to be a different entity.
- **Replay** – involves passive capture of a data unit and its subsequent transmission to produce an unauthorized effect.
- **Modification of messages** – Some portion of message is altered or the messages are delayed or recorded, to produce an unauthorized effect.
- **Denial of service** – Prevents or inhibits the normal use or management of communication facilities. Another form of service denial is the disruption of an entire network, either by disabling the network or overloading it with messages so as to degrade performance.

It is quite difficult to prevent active attacks absolutely, because to do so would require physical protection of all communication facilities and paths at all times. Instead, the goal is to detect them and to recover from any disruption or delays caused by them.



Active Attacks

CLASSICAL ENCRYPTION TECHNIQUES

Symmetric encryption : is a form of cryptosystem in which encryption and decryption are performed using the same key. It is also known as conventional encryption or single-key encryption.

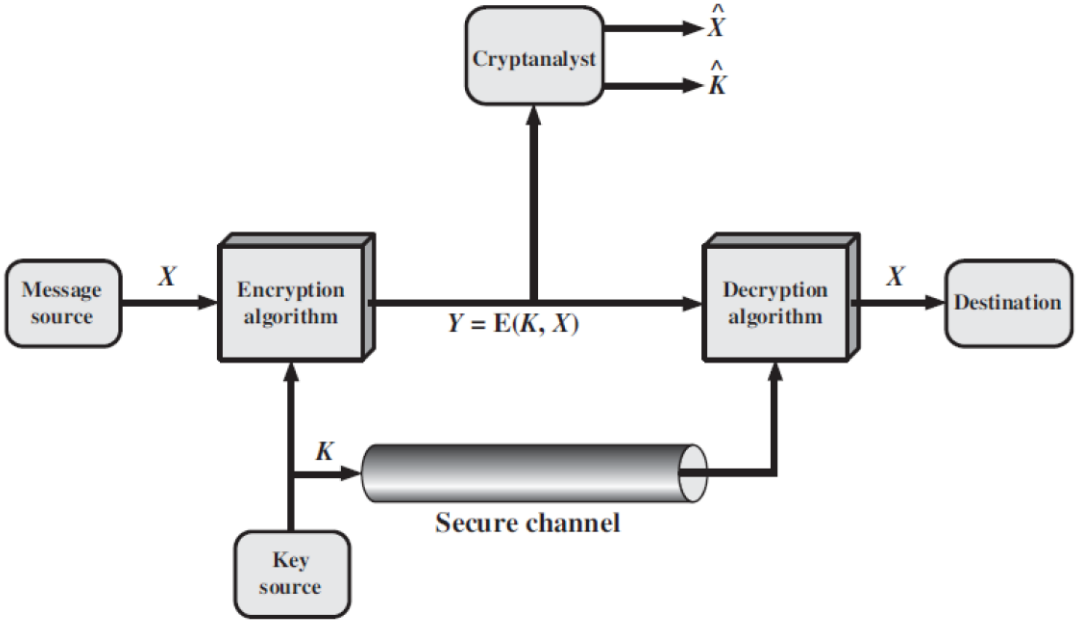


Figure 2.2 Model of Symmetric Cryptosystem

There are two requirements for secure use of conventional encryption (symmetric encryption):

• A strong encryption algorithm

• A secret key known only to sender / receiver

Mathematically:

C = EK(X) or C = E (K, X)

X = DK(C) or X = D (K, C)

X = plaintext, C = ciphertext, K = secret key, E = encryption algorithm, D = decryption

algorithm

Cryptanalysis

- Objective: to recover the plaintext of a ciphertext or, more typically, to recover the secret key.

- Kerkhoff's principle: the adversary knows all details about a cryptosystem except the secret key.
- Two general approaches:

  – brute-force attack

  – non-brute-force attack (cryptanalytic attack)

Brute-Force Attack

– Try every key to decipher the ciphertext.

– On average, need to try half of all possible keys

– Time needed proportional to size of key space

Key Size (bits)	**Number of Alternative Keys**	**Time required at 1 decryption/μs**	**Time required at 10^6 decryptions/μs**
32	$2^{32} = 4.3 \times 10^9$	2^{31} μs = 35.8 minutes	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	2^{55} μs = 1142 years	10.01 hours
128	$2^{128} = 3.4 \times 10^{38}$	2^{127} μs = 5.4×10^{24} years	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	2^{167} μs = 5.9×10^{36} years	5.9×10^{30} years
26 characters (permutation)	$26! = 4 \times 10^{26}$	2×10^{26} μs = 6.4×10^{12} years	6.4×10^6 years

Non-brute-force attack (Cryptanalytic Attacks)

•summarizes the various types of cryptanalytic attacks based on the amount of information

known to the cryptanalyst.

1. ciphertext only : only know algorithm / ciphertext, statistical, can identify plaintext
2. known plaintext : know/suspect plaintext & ciphertext to attack cipher
3. chosen plaintext : select plaintext and obtain ciphertext to attack cipher

4. chosen ciphertext : select ciphertext and obtain plaintext to attack cipher
5. chosen text : select either plaintext or ciphertext to en/decrypt to attack cipher

Classical Ciphers

• Plaintext is viewed as a sequence of elements (e.g., bits or characters)

• There are three basic building blocks of all encryption techniques:

1. Substitution cipher: replacing each element of the plaintext with another element.

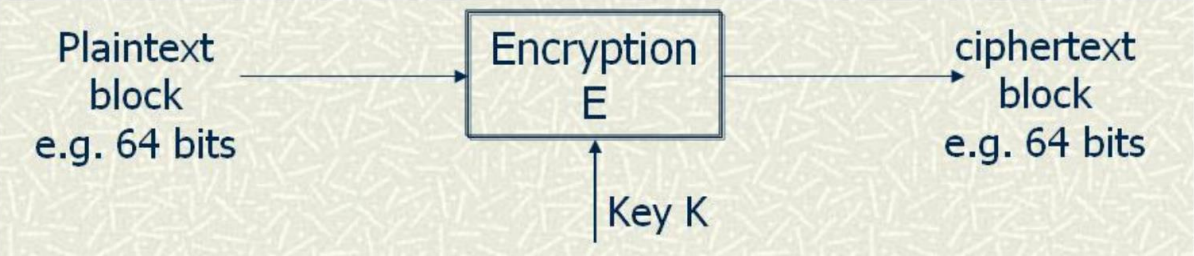
(Caesar Cipher, Monoalphabetic Ciphers, Playfair Cipher, Polyalphabetic Ciphers, Vegenere table, One-Time Pad)

2. Transposition (or permutation) cipher: rearranging the order of the elements of the plaintext. (Rail fence cipher, Row Transposition Ciphers)
3. Product cipher: using multiple stages of substitutions and transpositions.

Block Ciphers

Block Ciphers definition

- Encrypt a block of input to a block of output
- Typically, the two blocks are of the same length
- Most symmetric key systems block size is 64.
- Many block ciphers have a Feistel structure.
- Such a structure consists of a number of identical rounds of processing.
- In each round, a substitution is performed on one half of the data being processed,
- followed by a permutation that interchanges the two halves.
- The original key is expanded so that a different key is used for each round.



Iterated Block Cipher - A block cipher that "iterates a fixed number of times of another block cipher, called round function, with a different key, called round key, for each iteration".

Diffusion – dissipates statistical structure of plaintext over bulk of ciphertext.

Confusion – makes relationship between ciphertext and key as complex as possible.

Feistel Cipher:

Feistel ciphers are a special class of iterated block ciphers , where the ciphertext is calculated from the plaintext by repeated application of the same transformation or round function. In a Feistel cipher, the text being encrypted is split into two halves. The round function f is applied to one half using a subkey and the output of f is exclusive-ored with the other half. The two halves are then swapped. Each round follows the same pattern except for the last round where there is no swap.

A nice feature of a Feistel cipher is that encryption and decryption are structurally identical, though the subkeys used during encryption at each round are taken in reverse order during decryption.

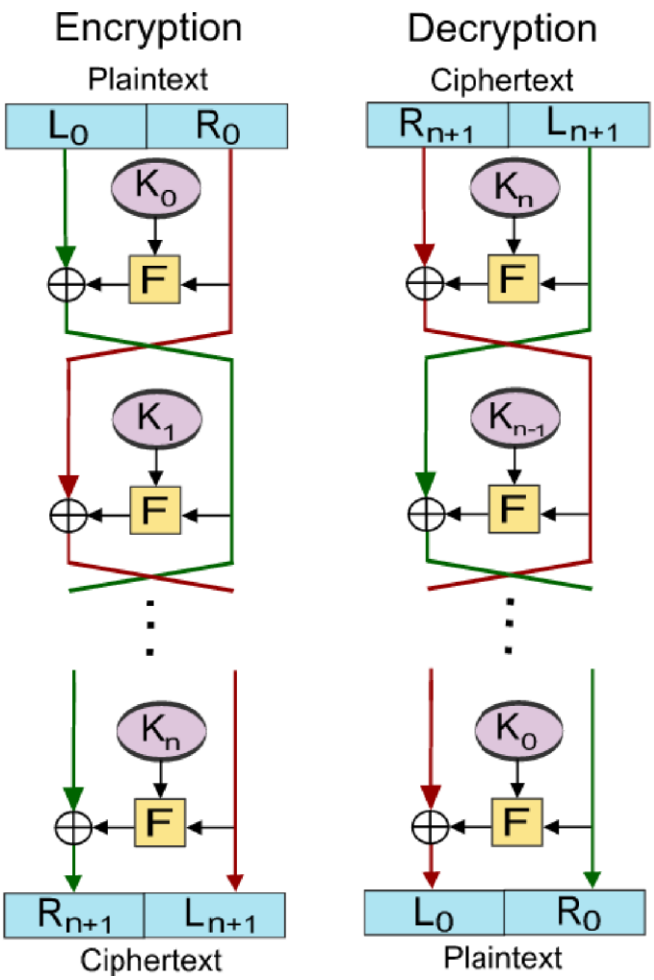
- Several block ciphers are based on the structure proposed byFeistel in 1973.
- A Feistel Network is fully specified given

  - the block size: n = 2w

  - number of rounds: d

  - d round functions $f_1, \ldots, f_d: \{0,1\}^w \rightarrow \{0,1\}^w$

- Used in DES and many other block ciphers.



Construction details

Let F be the round function and let $K_0, K_1, \ldots, K_n$ be the sub-keys for the rounds $0, 1, \ldots, n$ respectively.

Then the basic operation is as follows:

Split the plaintext block into two equal pieces, (L_0, R_0)

For each round , $i = 0, 1, \ldots, n$ compute

$$L_{i+1} = R_i$$
$$R_{i+1} = L_i \oplus \mathrm{F}(R_i, K_i).$$

Then the ciphertext is (R_{n+1}, L_{n+1}).

Decryption of a ciphertext (R_{n+1}, L_{n+1}) is accomplished by computing for $i = n, n-1, \ldots, 0$

$$R_i = L_i.$$
$$L_i = R_{i+1} \oplus \mathrm{F}(L_{i+1}, K_i).$$

Then (L_0, R_0) is the plaintext again.

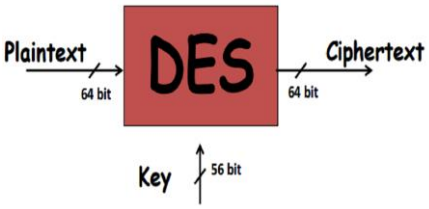
The diagram illustrates both encryption and decryption. Note the reversal of the subkey order for decryption; this is the only difference between encryption and decryption.

Data Encryption Standard (DES)

The Data Encryption Standard (DES), known as the Data Encryption Algorithm (DEA) by ANSI and the DEA-1 by the ISO, has been most widely used block cipher in world, especially in financial industry.

DES Features:

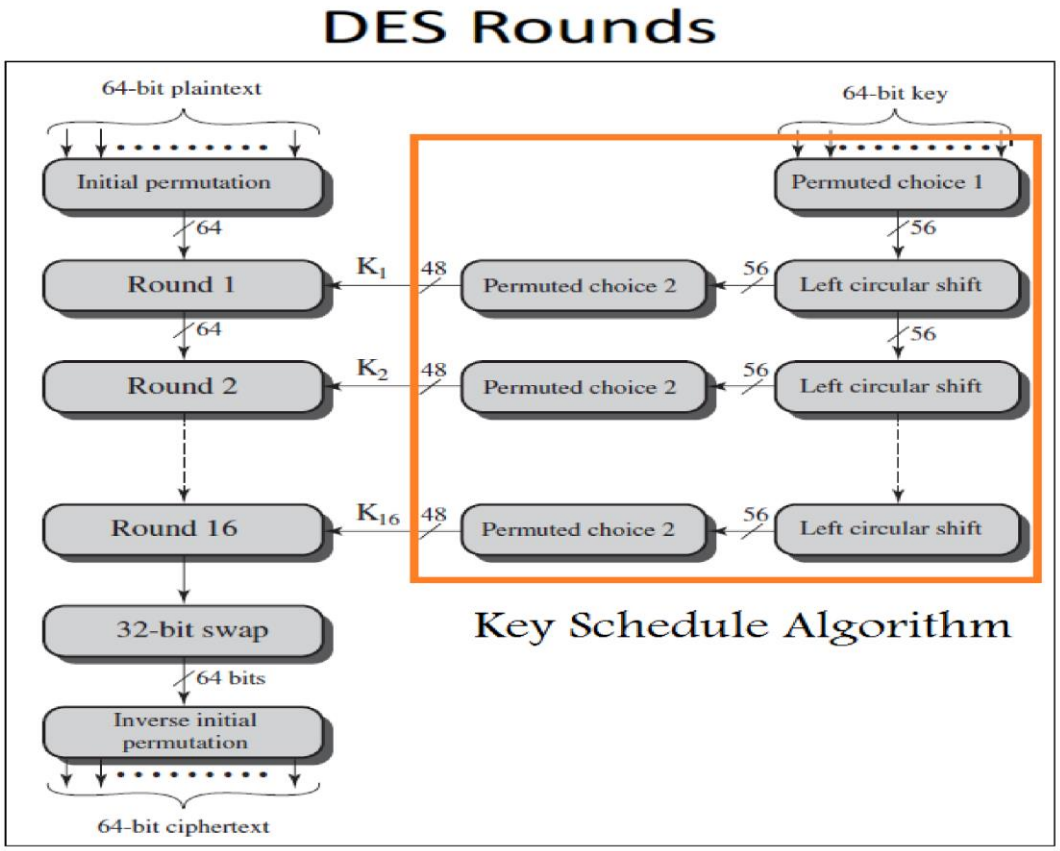
- It encrypts 64-bit data (Block size = 64 bits).
- Key size = 56 bits (in reality, 64 bits, but 8 are used as parity-check bits for error control slide).
- Number of rounds = 16. (a 16-round Feistel cipher with block size of 64 bits.)
- 16 intermediary keys, each 48 bits.
- The algorithm is a combination of the two basic techniques of encryption: confusion and diffusion.



Key length in DES

- In the DES specification, the key length is 64 bit.
- 8 bytes; in each byte, the 8th bit is a parity-check bit

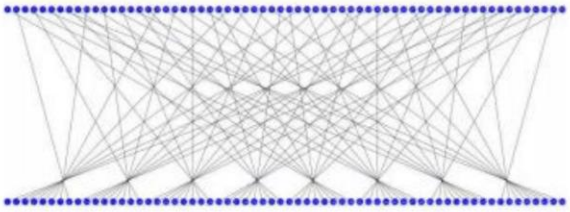
- Each parity-check bit is the XOR of the previous 7 bits



General Depiction of DES Encryption Algorithm

Initial Permutation (IP)

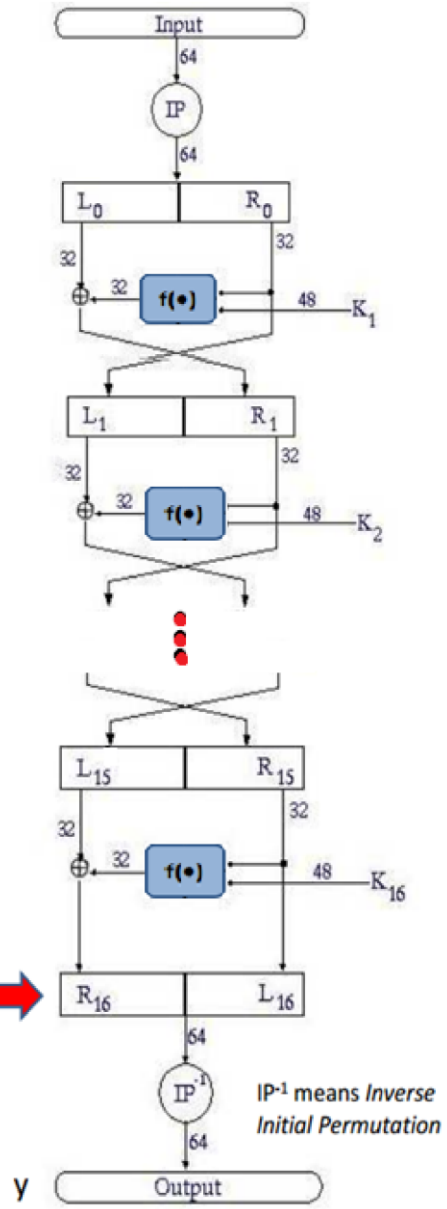
58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7



- This table specifies the input permutation on a 64-bit block.
- The meaning is as follows:
  The first bit of the output is taken from the 58th bit of the input; the second bit from the 50th bit, and so on, with the last bit of the output taken from the 7th bit of the input.
- This information is presented as a table for ease of presentation:
  it is a vector, not a matrix.

DES Rounds in Details

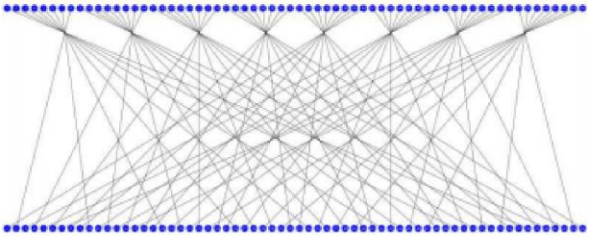
- $IP(x) = L_0R_0$
- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$
- $y = IP^{-1}(R_{16}L_{16})$



- Note that, as usual:
  - $R_{16} = L_{15} \oplus f(R_{15}, K_{16})$
  - $L_{16} = R_{15}$
- ... but they are *switched* in the pre-output

Final Permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25



The final permutation is the inverse of the initial permutation; the table is interpreted similarly.

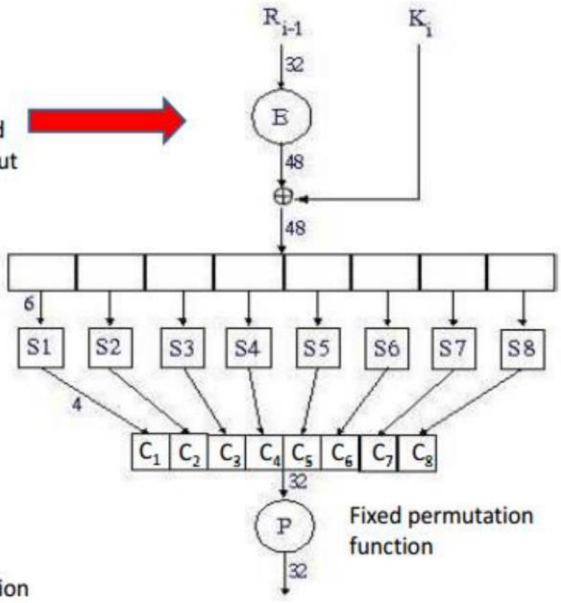
That is, the output of the Final Permutation has bit 40 of the preoutput block as its first bit, bit 8 as its second bit, and so on, until bit 25 of the preoutput block is the last bit of the output.

DES "f(•)" Function

E is an *expansion function* which takes a block of 32 bits as input and produces a block of 48 bits as output

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

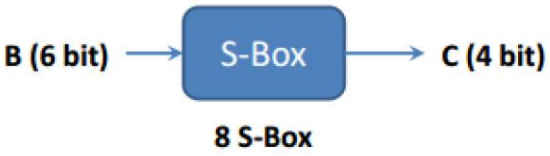
16 bits appear twice, in the expansion



S-boxes

- S-boxes are the only *non-linear* elements in DES design

Each of the unique selection functions $S_1, S_2, \ldots, S_8$, takes a 6-bit block as input and yields a 4-bit block as output



- S = matrix 4x16, values from 0 to 15
- B (6 bit long) = $b_1b_2b_3b_4b_5b_6$
  - b_1b_6 ➔ r = row of the matrix (2 bits: 0,1,2,3)
  - $b_2b_3b_4b_5$ ➔ c = column of the matrix (4 bits:0,1,…15)
- C (4 bit long) = Binary representation of S(r, c)

Example (S1)

Row #	S_1	1	2	3	…	7	…	15	Column #
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

S(i, j) <16, can be represented with 4 bits

Example: B =101111

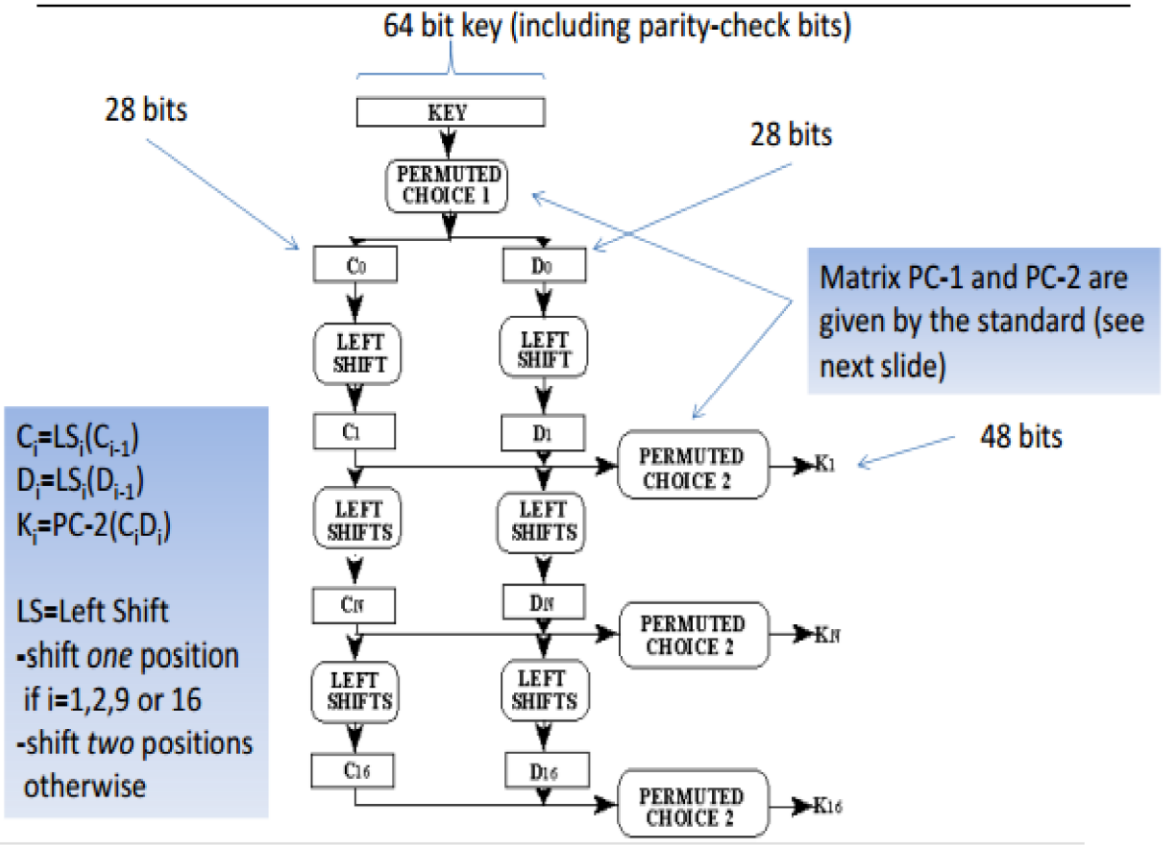
b_1b_6 = 11 = row 3

$b_2b_3b_4b_5$= 0111 = column 7

➔ C=7=0111

Another example: B=011011, C=?

DES Key Generation ($K_1 - K_{16}$)



DES Permuted Choice 1 and 2 (PC-1, PC-2)

Parity-check bits (namely, bits 8,16, 4,32,40,48,56,64) are not chosen, they do not appear in **PC-1**

Left						
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
Right						
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

PC-2 selects the 48-bit subkey for each round from the 56-bit key-schedule state

DES Weak Keys

- DES uses 16 48-bits keys generated from a master 56-bit key (64 bits if we consider also parity bits)
- Weak keys: keys make the same sub-key to be generated in more than one round.
- Result: reduce cipher complexity
- Weak keys can be avoided at key generation.
- DES has 4 weak keys
  - 01010101 01010101
  - FEFEFEFE FEFEFEFE
  - E0E0E0E0 F1F1F1F1
  - 1F1F1F1F 0E0E0E0E

DES Decryption

Decryption uses the same algorithm as encryption, except that the subkeys K 1, K 2, …K16 are applied in reversed order

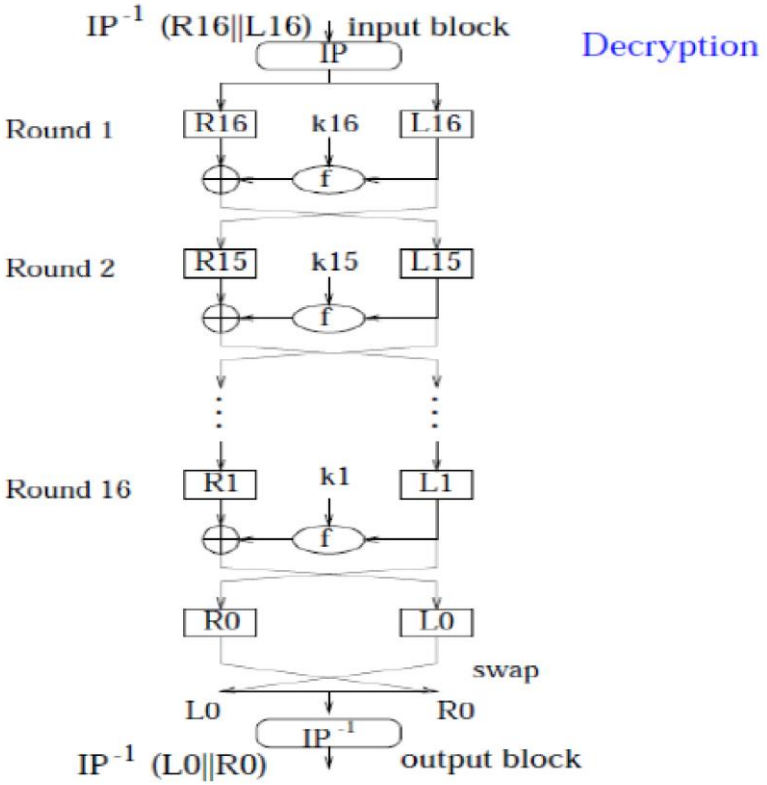
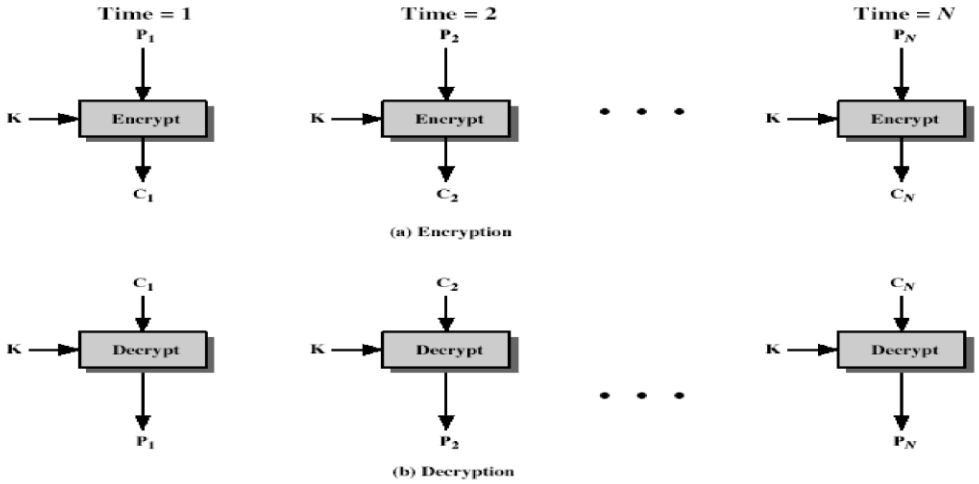


Figure . DES Decryption

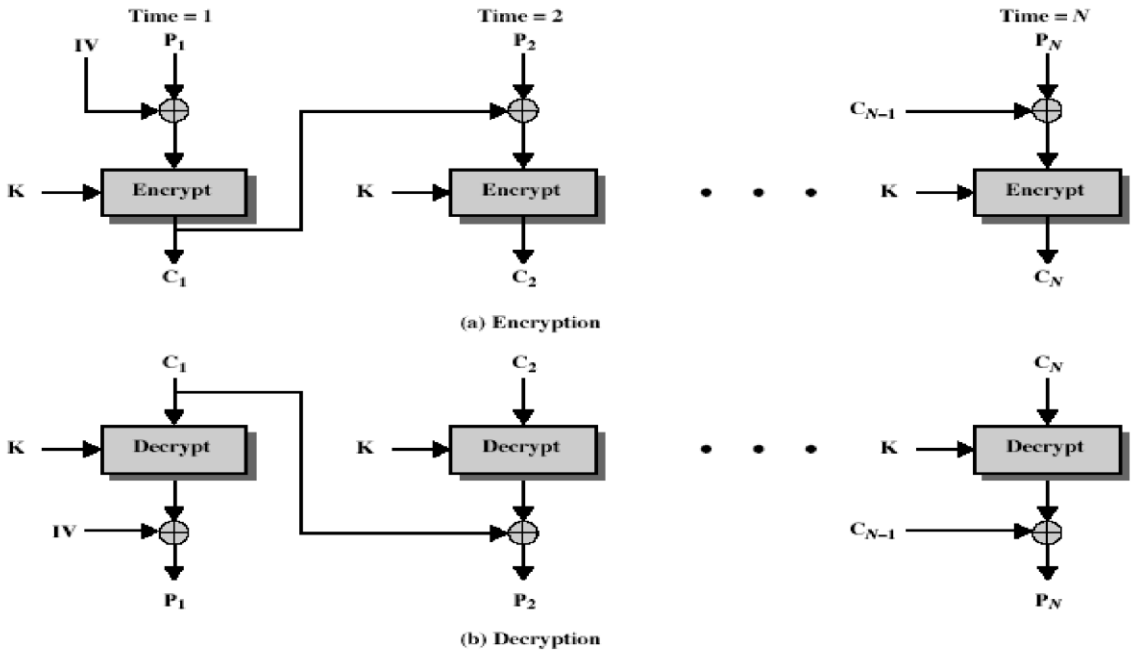
DES Modes

1. ECB (Electronic Code Book)Operation Mode - Blocks of clear text are encrypted independently. Main properties of this mode:
   - Identical clear text blocks are encrypted to identical cipher text blocks.
   - Re-ordering clear text blocks results in re-ordering cipher text blocks.
   - An encryption error affects only the block where it occurs.

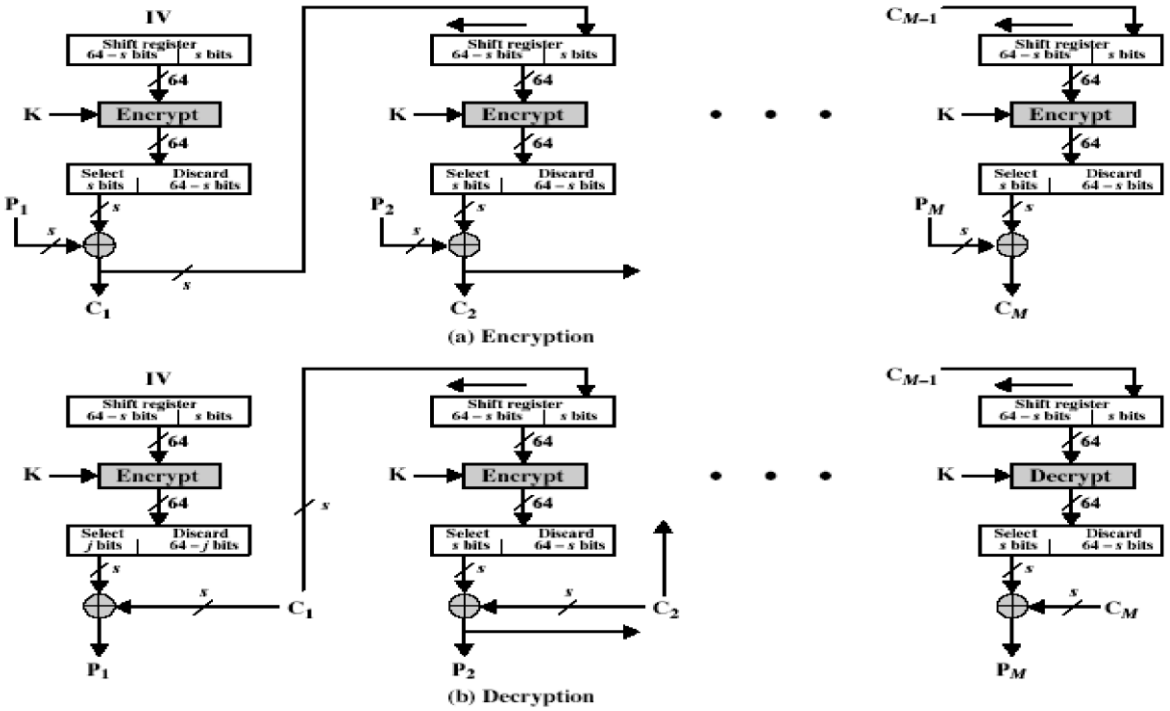


2. **CBC (Cipher Block Chaining) Operation Mode** - The previous cipher text block is XORed with the clear text block before applying the encryption mapping. Main properties of this mode:

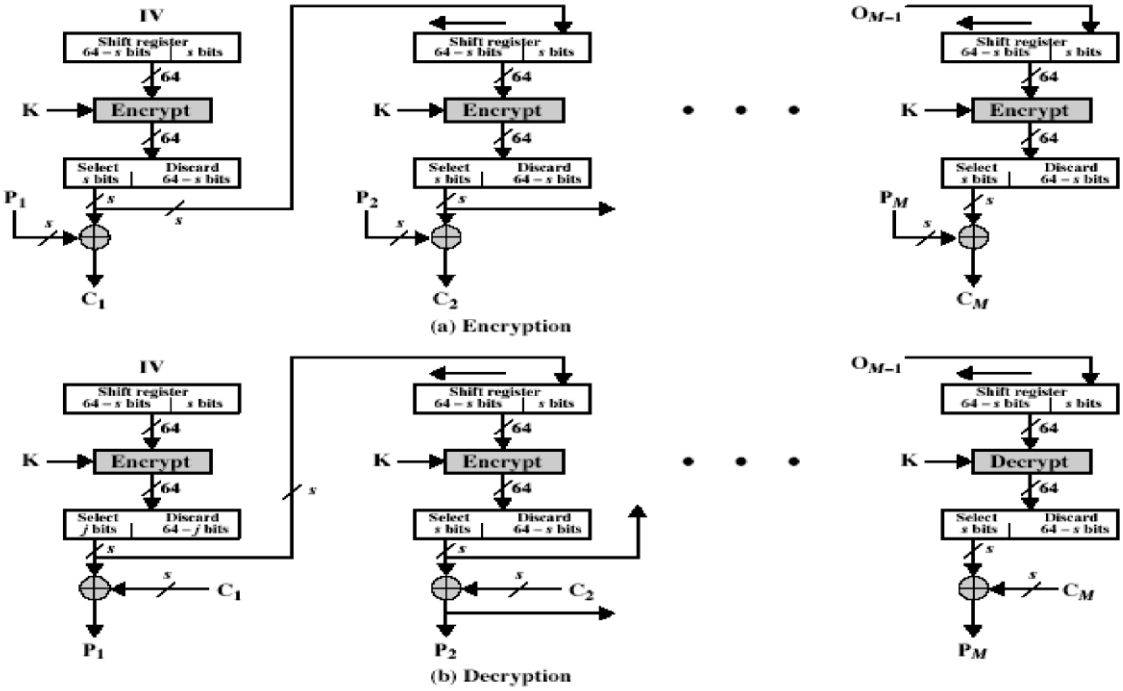
An encryption error affects only the block where is occurs and one next block.



3. **CFB (Cipher FeedBack)** - Message is treated as a stream of bits , Bitwise-added to the output of the block cipher , Result is feedback for next stage (hence name) Cipher FeedBack (CFB) Message is treated as a stream of bits , Bitwise-added to the output of the block cipher , Result is feedback for next stage (hence name).



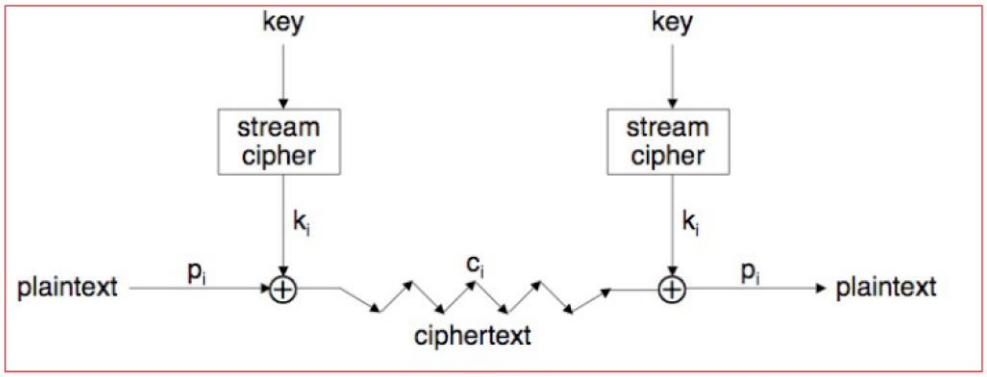
4. **OFM (Output Feedback Mode)**- The block cipher is used as a stream cipher, it produces the random key stream.



Stream Ciphers

A stream cipher is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (keystream). In a stream cipher each plaintext digit is encrypted one at a time with the corresponding digit of the keystream, to give a digit of the ciphertext stream.

- Generalization of one-time pad,



Generic view of stream cipher

- The message is represented as a binary string (a sequence of 0's and 1's using a coding mechanism such as ASCII coding.
- The encryption is done by adding the key to the message modulo 2, bit by bit. This process is often called *exclusive or*, and is denoted by *XOR.*
- Stream cipher is initialized with short **key**.
- Encryption of each digit is dependent on the current state of the cipher
- Traditionally, stream ciphers were based on **shift registers.**
- *Speed*
  - *Initialization*
  - *Keystream generation*
- *Resources – memory, power,* cpu
- *Hardware, software suitability*

True random number generators (TRNGs) are characterized by the fact that their output cannot be reproduced. For instance, if we flip a coin 100 times and record the resulting sequence of 100 bits, it will be virtually impossible for anyone on Earth to generate the same 100 bit sequence. The chance of success is 1/2100, which is an extremely small probability.

Key:

- key can be used only once.
- The key is a **truly random sequence** of 0's and 1's of the same length as the message.

- Key is "stretched" into long **keystream**
- Keystream is used like a one-time pad (XOR to encrypt or decrypt)

Types of stream ciphers

A stream cipher generates successive elements of the keystream based on an internal state. This state is updated in essentially two ways: if the state changes independently of the plaintext or ciphertext messages, the cipher is classified as asynchronous stream cipher. By contrast, self- synchronising stream ciphers update their state based on previous ciphertext digits.

1. Synchronous Stream Ciphers
   - Key-stream is independent of plain and cipher-text.
   - Both sender &receiver must be synchronized.
   - Resynchronization can be needed.
   - No error propagation.
   - Active attacks can easily be detected.

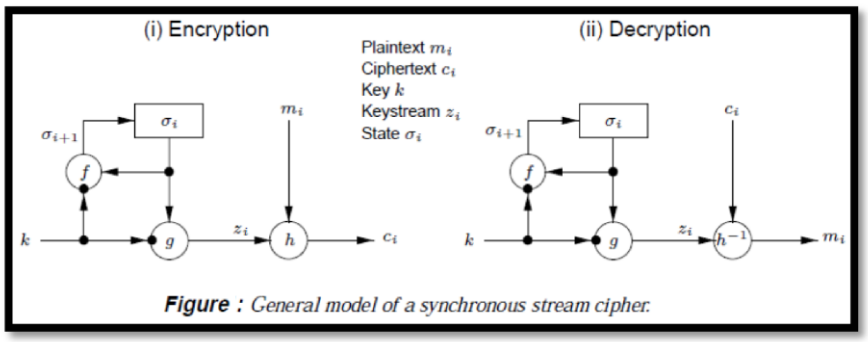


Figure : *General model of a synchronous stream cipher.*

2. Self-Synchronizing Stream Ciphers
   - Key-stream is a function of fixed number t of cipher- text bits.
   - Limited error propagation (up to t bits).
   - Active attacks cannot be detected.
   - At most t bits later, it resynchronizes itself when synchronization is lost.
   - It helps to diffuse plain-text statistics.

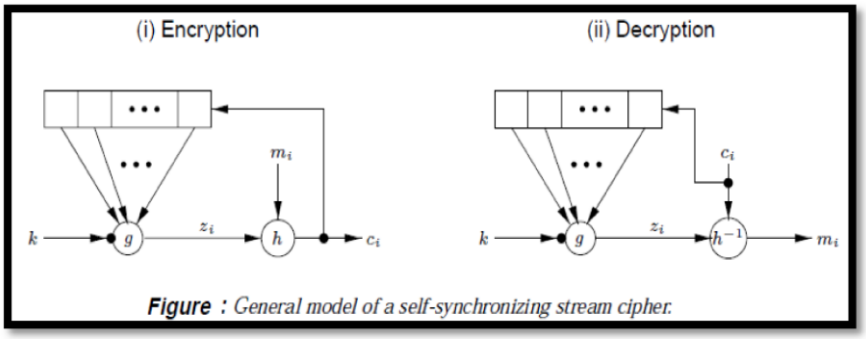
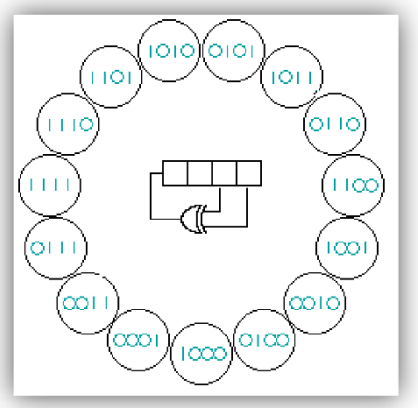


Figure : *General model of a self-synchronizing stream cipher.*

Shift Registers

- Shift register includes:
  - o A series of stages each holding one bit.
  - o The content of stage i is moved to stage i − 1 for each i
  - o A feedback function.
- A linear feedback shift register (LFSR) has a linear feedback function. The new content of stage M − 1 is the feedback bit sj which is calculated by adding together modulo 2 the previous contents of a fixed subset of stages 0, 1,….., M – 1.
- The maximal sequence consists of every possible state except the "0" state.



Linear feedback shift registers

Linear feedback shift registers (LFSRs) are used in many of the keystream generators that have been proposed in the literature. There are several reasons for this:

1. LFSRs are well-suited to hardware implementation;
2. they can produce sequences of large period;
3. they can produce sequences with good statistical properties; and
4. because of their structure, they can be readily analyzed using algebraic techniques.

A Mathematical Description of LFSRs

The general form of an LFSR of degree m is shown in Fig. It shows m flip-flops and m possible feedback locations, all combined by the XOR operation. Whether a feedback path is active or not, is defined by the feedback coefficient $p_0, p_1, \ldots, p_{m-1}$:

- If $p_i = 1$ (closed switch), the feedback is active.
- If $p_i = 0$ (open switch), the corresponding flip-flop output is not used for the feedback.

With this notation, we obtain an elegant mathematical description for the feedback path. If we multiply the output of flip-flop i by its coefficient pi, the result is either the output value if $p_i = 1$, which corresponds to a closed switch, or the value zero if $p_i = 0$, which corresponds to an open switch. The values of the feedback coefficients are crucial for the output sequence produced by the LFSR.

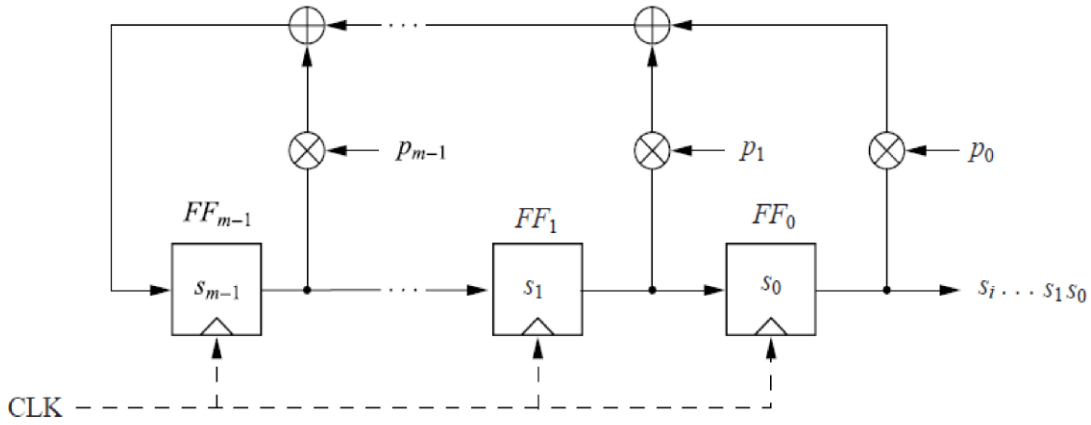


Fig. General LFSR with feedback coefficients p_i and initial values $s_{m-1}, \ldots, s_0$

Let's assume the LFSR is initially loaded with the values $s_0, \ldots, s_{m-1}$. The next output bit of the LFSR s_m, which is also the input to the leftmost flip-flop, can be computed by the XOR-sum of the products of flip-flop outputs and corresponding feedback coefficient:

$$s_m \equiv s_{m-1}\, p_{m-1} + \cdots + s_1 p_1 + s_0 p_0 \bmod 2$$

Due to the finite number of recurring states, the output sequence of an LFSR repeats periodically. This was also illustrated in the following example. Moreover, an LFSR can produce output sequences of different lengths, depending on the feedback coefficients.

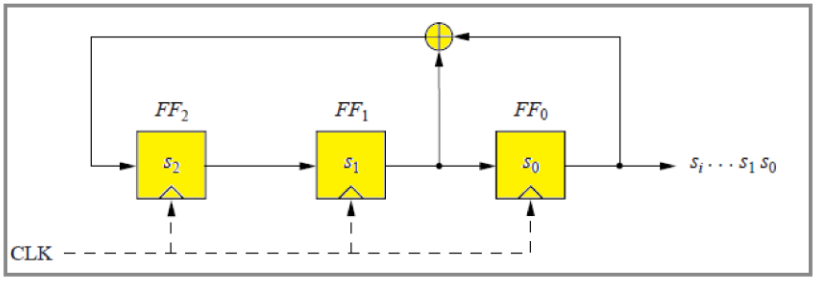


Fig. Linear feedback shift register of degree 3 with initial values s_2, s_1, s_0

Table Sequence of states of the LFSR

clk	FF_2	FF_1	$FF_0 = s_i$
0	1	0	0
1	0	1	0
2	1	0	1
3	1	1	0
4	1	1	1
5	0	1	1
6	0	0	1
7	1	0	0
8	0	1	0

Seed (initial state)

Starts to repeat after clock cycle 6. This means the LFSR output has period of length 7 and has the form:

0010111 0010111 **0010111** . . .

The following theorem gives us the maximum length of an LFSR as function of its degree.

Theorem : *The maximum sequence length generated by an LFSR of degree m is* $2^m - 1$.

Example . LFSR with maximum-length output sequence
Given an LFSR of degree $m = 4$ and the feedback path $(p_3 = 0, p_2 = 0, p_1 = 1, p_0 = 1)$, the output sequence of the LFSR has a period of $2^m - 1 = 15$, i.e., it is a maximum-length LFSR.

Example . LFSR with non-maximum output sequence
Given an LFSR of degree $m = 4$ and $(p_3 = 1, p_2 = 1, p_1 = 1, p_0 = 1)$, then the output sequence has period of 5; therefore, it is not a maximum-length LFSR.

As an example, the notation (0,2,5) refers to the polynomial 1+x2 +x5

Example

Create a linear feedback shift register with 4 cells in which s4 = s1 xor s0. Show the value of output for 20 transitions (shifts) if the seed is (0001)2.

Solution

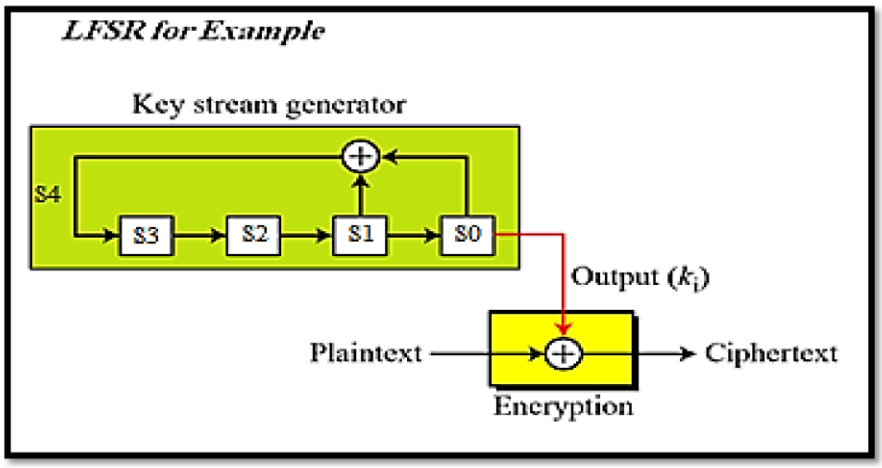


Table Cell values and key sequence for Example.

States	S4	S3	S2	S1	S0	ki
Initial	1	0	0	0	1	
1	0	1	0	0	0	1
2	0	0	1	0	0	0
3	1	0	0	1	0	0
4	1	1	0	0	1	0
5	0	1	1	0	0	1
6	1	0	1	1	0	0
7	0	1	0	1	1	0
8	1	0	1	0	1	1
9	1	1	0	1	0	1
10	1	1	1	0	1	0

Table Continued

States	S4	S3	S2	S1	S0	ki
11	1	1	1	1	0	1
12	0	1	1	1	1	0
13	0	0	1	1	1	1
14	0	0	0	1	1	1
15	1	0	0	0	1	1
16	0	1	0	0	0	1
17	0	0	1	0	0	0
18	1	0	0	1	0	0
19	1	1	0	0	1	0
20	1	1	1	0	0	1

Note that the key stream is 100010011010111 10001…. This looks like a random sequence at first glance, but if we go through more transitions, we see that the sequence is periodic. It is a repetition of 15 bits as shown below:

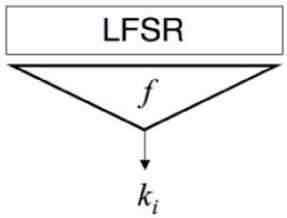
100010011010111 **100010011010111** 100010011010111 **100010011010111** …

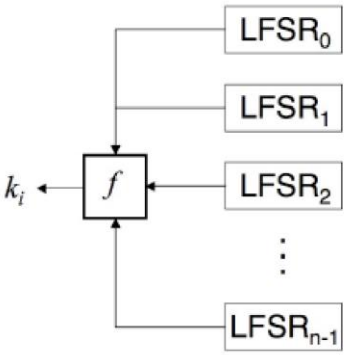
A pseudorandom number generator

A pseudorandom number generator (PRNG), also known as a deterministic random bit generator (DRBG), is an algorithm for generating a sequence of numbers whose properties approximate the properties of sequences of random numbers. The PRNG-generated sequence is not truly random, because it is completely determined by a relatively small set of initial values, called the PRNG's seed (which may include truly random values). Although sequences that are closer to truly random can be generated using hardware random number generators, pseudorandom number generators are important in practice for their speed in number generation and their reproducibility.

Shift Register-Based Stream Ciphers

- Two approaches to LFSR-based stream ciphers
  - One LFSR with nonlinear combining function.
  - Multiple LFSRs combined via nonlinearfunction.





RC4

Description

RC4 generates a pseudorandom stream of bits (a keystream). As with any stream cipher, these can be used for encryption by combining it with the plaintext using bit-wise exclusive-or; decryption is performed the same way (since exclusive-or with given data is an involution). (This is similar to the Vernam cipher except that generated pseudorandom bits, rather than a prepared stream, are used.) To generate the keystream, the cipher makes use of a secret internal state which consists of two parts:

1. A permutation of all 256 possible bytes (denoted "S" below).
2. Two 8-bit index-pointers (denoted "i" and "j").

The permutation is initialized with a variable length key, typically between 40 and 256 bits, using the key-schedulingalgorithm (KSA). Once this has been completed, the stream of bits is generated using the pseudo-random generation algorithm (PRGA).

Key-scheduling algorithm (KSA)

The key-scheduling algorithm is used to initialize the permutation in the array "S". "keylength" is defined as the number of bytes in the key and can be in the range $1 \leq \text{keylength} \leq 256$, typically between 5 and 16, corresponding to a key length of 40 – 128 bits. First, the array "S" is initialized to the identity permutation. S is then processed for 256 iterations in a similar way to the main PRGA, but also mixes in bytes of the key at the same time.

Pseudo-random generation algorithm (PRGA)

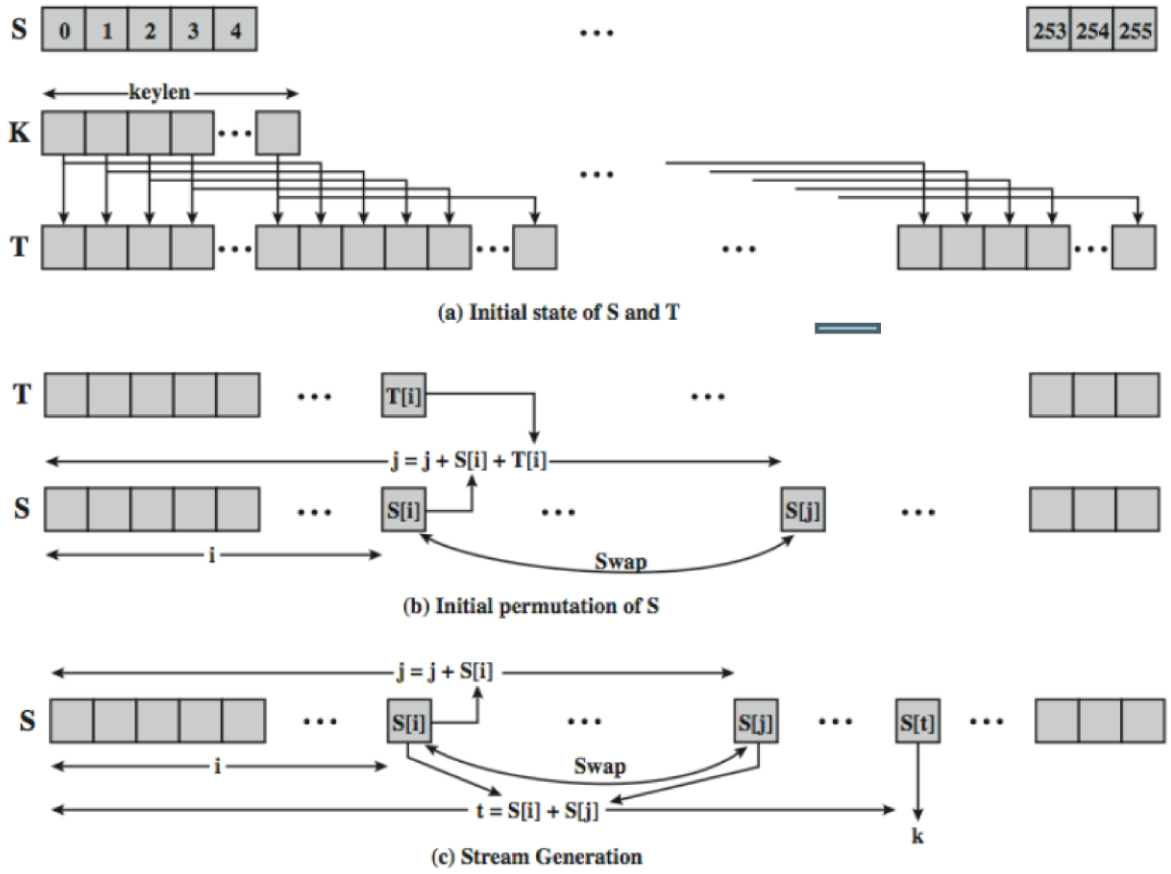
For as many iterations as are needed, the PRGA modifies the state and outputs a byte of the keystream. In each iteration, the PRGA increments i, looks up the ith element of S, S[i], and adds that to j, exchanges the values of S[i] and S[j], and then uses the sum S[i] + S[j] (modulo 256) as an index to fetch a third element of S, (the keystream value K below) which is XORed with the next byte of the message to produce the next byte of either ciphertext or plaintext. Each element of S is swapped with another element at least once every 256 iterations.

RC4 Key Schedule

- Start with an array S of numbers: 0..255
- Use key to well and truly shuffle
- S forms **internal state** of the cipher

```
for i = 0 to 255 do
    S[i] = i
    T[i] = K[i mod keylen])
j = 0
for i = 0 to 255 do
    j = (j + S[i] + T[i]) (mod 256)
    swap (S[i], S[j])
```

RC4 Overview



Homework. :RC4 Example

Assume we use a 4 x 3-bit key, K, and plaintext P as below:

K = [1 2 3 6]

P = [1 2 2 2]

RC4 Encryption

- Encryption continues shuffling array values
- Sum of shuffled pair selects "stream key" value from permutation
- XOR S[t] with next byte of message to en/decrypt

```
i = j = 0
for each message byte Mi
    i = (i + 1) (mod 256)
    j = (j + S[i]) (mod 256)
    swap(S[i], S[j])
    t = (S[i] + S[j]) (mod 256)
    Ci = Mi XOR S[t]
```

University of Mosul \ College of Computer Sciences and Mathematics

Dept.: Computer Sciences

Message Authentication and Hash Functions

Encryption/Decryption

- Provides message confidentiality.
- Does it provide message authentication?

Message Authentication

Message authentication is a mechanism or service used to verify the integrity of a message. Message authentication assures that data received are exactly as sent by sender (i.e., contain no modification, insertion, deletion, or replay and that the purported identity of the sender is valid).

- ***B*** receives a message from ***A***, he wants to know
  - ✓ (Data origin authentication) whether the message was really sent by ***A***;
  - ✓ (Data integrity) whether the message has been modified.

- ***Solutions:***
  - ✓ ***A*** attaches ***a message authentication code*** (MAC) to the message.
  - ✓ Or ***A*** attaches ***digital signature*** to the message.

Hash function can be used with ***a message authentication code*** (MAC) and ***digital signature***.

Hashing Functions

- condenses arbitrary message to fixed size
- usually assume that the hash function is public and not keyed
  - MAC which is keyed
- used to detect changes to message.
- can use in various ways with message.
- most often to create a digital signature.

Hash Function Properties

- a Hash Function produces a fingerprint of some file/message/data

  h = H(M)

  - condenses a variable-length message M
  - to a fixed-sized fingerprint

- assumed to be public

Requirements for Hash Functions

1. can be applied to any sized message M
2. produces fixed-length output h
3. is easy to compute h=H(M) for any message M
4. given h is infeasible to find x s.t. H(x)=h
   - one-way property
5. given x is infeasible to find y s.t. H(y)=H(x)
   - weak collision resistance
6. is infeasible to find any x,y s.t. H(y)=H(x)
   - strong collision resistance

Simple Hash Functions

All hash functions operate using the following general principles. The input (message, file, etc.) is viewed as a sequence of n-bit blocks. The input is processed one block at a time in an iterative fashion to produce an n-bit hash function.

One of the simplest hash functions is the bit-by-bit exclusive-OR (XOR) of every block. This can be expressed as follows:

$$C_i = b_{i1} \oplus b_{i1} \oplus \ldots \oplus b_{im}$$

where

C_i = ith bit of the hash code, $1 \leq i \leq n$
m = number of n-bit blocks in the input
b_{ij} = ith bit in jth block
$\oplus$ = XOR operation

	bit 1	bit 2	• • •	bit n
block 1	b_{11}	b_{21}		b_{n1}
block 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
block m	b_{1m}	b_{2m}		b_{nm}
sh code	C_1	C_2		C_n

Figure . Simple Hash Function Using Bitwise XOR

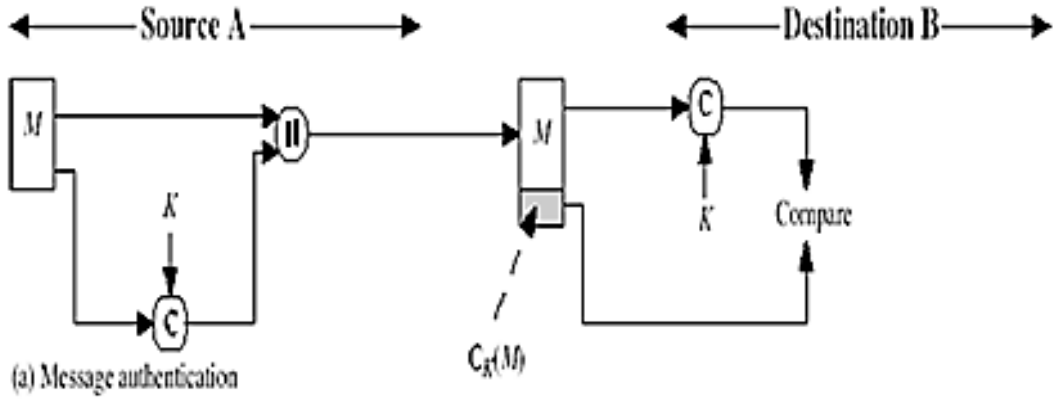
This operation produces a simple parity for each bit position and is known as a longitudinal redundancy check. It is reasonably effective for random data as a data integrity check.

A Message Authentication Code (MAC)

A message authentication code (MAC) is an algorithm that requires the ***use of a secret key***. A MAC takes a variable-length message and a secret key as input and produces an authentication code. A recipient in possession of the secret key can generate an authentication code to verify the integrity of the message.

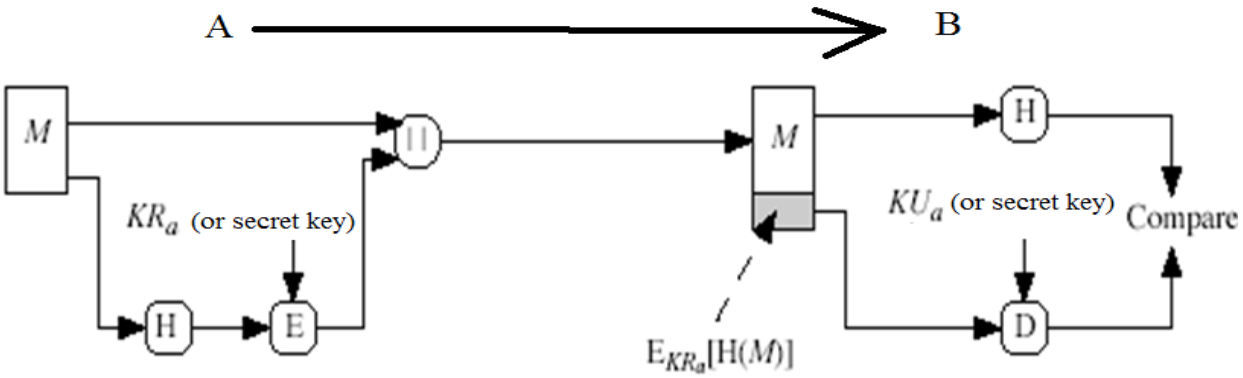
- generated by an MAC function C that creates a small fixed-sized block
  - depending on both message M and a shared secret key K, MAC=C_K(M)
  - MAC is appended to the message M
- receiver performs same computation on message and checks it matches the MAC
- provides assurance that message is unaltered and comes from sender

- can also use encryption for secrecy
  - generally use separate keys for each
  - can compute MAC either before or after encryption
  - is generally regarded as better done before.



Requirements for MACs

- taking into account the types of attacks
- need the MAC to satisfy the following:
  - knowing a message and MAC, is infeasible to find another message with same MAC
  - MACs should be uniformly distributed
  - MAC should depend equally on all bits of the message
- Hash function can be used. See the following figure..



Digital Signature

Digital signature is an authentication mechanism that enables the creator of a message to attach a code that acts as a signature (information about sender). Typically the signature is formed by taking the hash of the message and encrypting the message with the creator's private key. The signature guarantees the source and integrity of the message.

Digital Signature Properties

1. must depend on the message being signed
2. must use information unique to sender
   - to prevent both forgery and denial
3. must be relatively easy to produce
4. must be relatively easy to recognize & verify
5. be computationally infeasible to forge
   - with new message for existing digital signature
   - with fraudulent digital signature for given message
6. be practical save a copy of the digital signature in storage

The Digital Signature Mechanism

Example of RSA encryption\decryption (***security & authentication***)

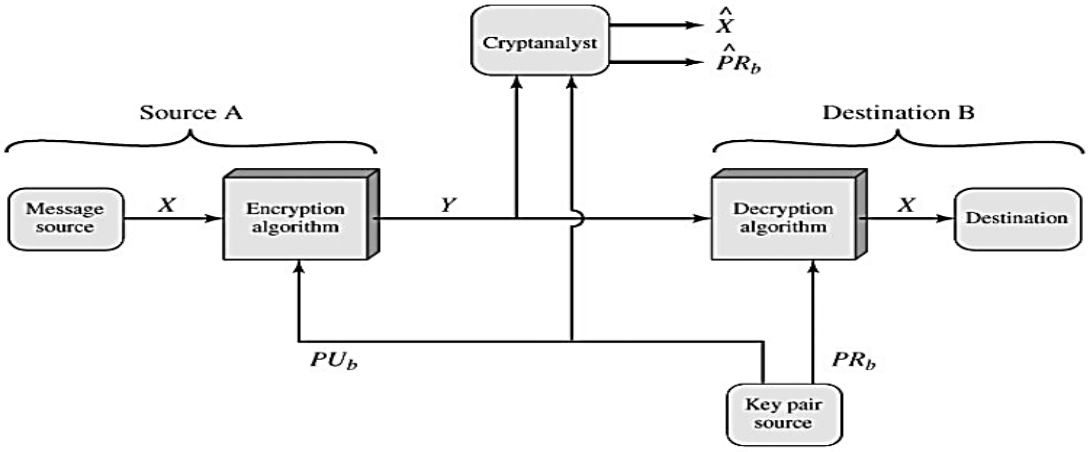


Figure : Public-Key Cryptosystem: Secrecy

The most important development from the work on public-key cryptography is the digital signature. The digital signature provides a set of security capabilities that would be difficult to implement in any other way.

A can sign a message using a digital signature generation algorithm. The inputs to the algorithm are the message and **A**'s private key. Any other user, say **B**, can verify the signature using a verification algorithm, whose inputs are the message, the signature, and **A**'s public key, see the following figures.

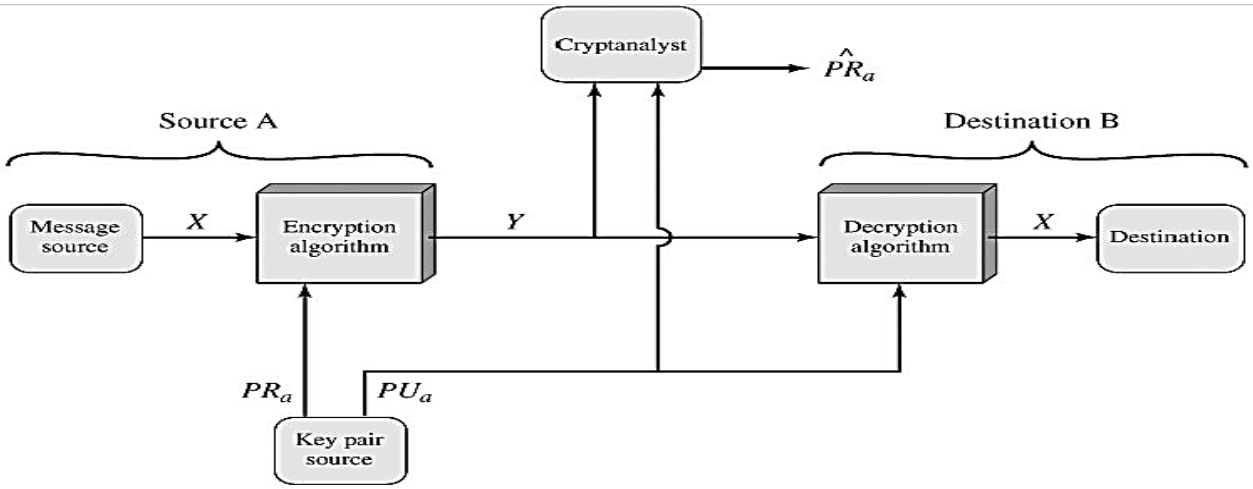


Figure : Public-Key Cryptosystem: Authentication

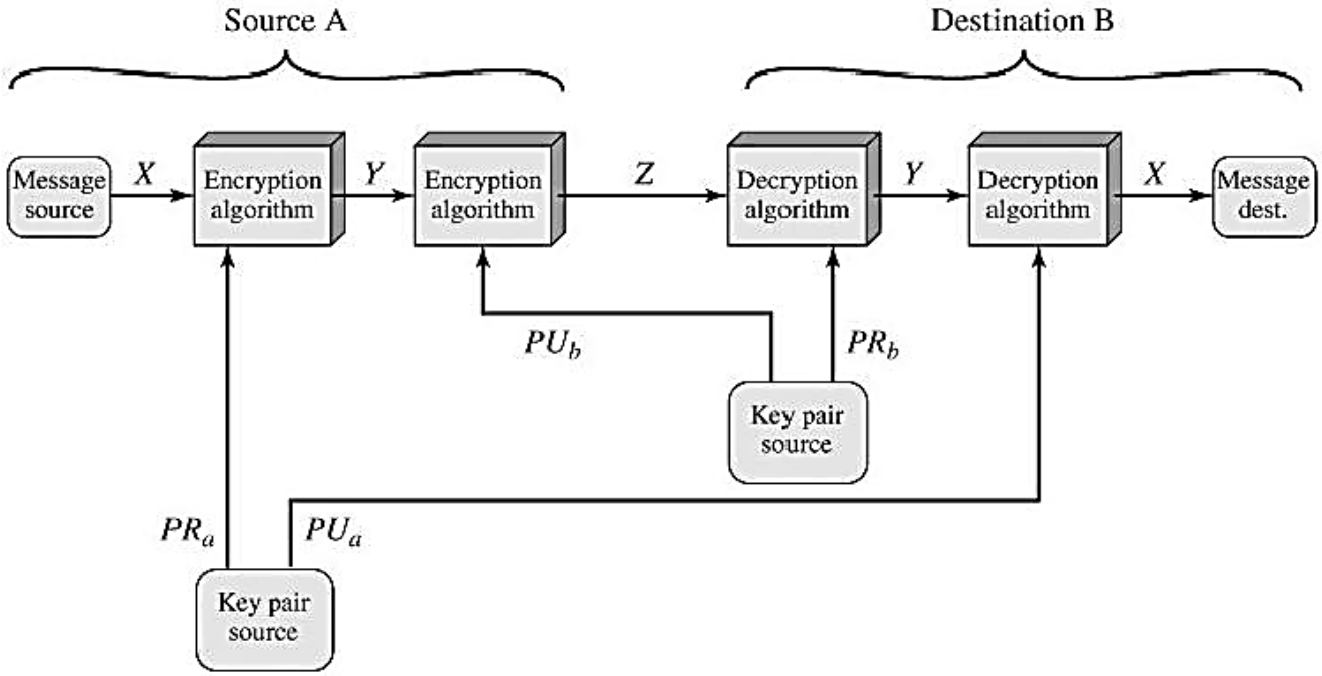
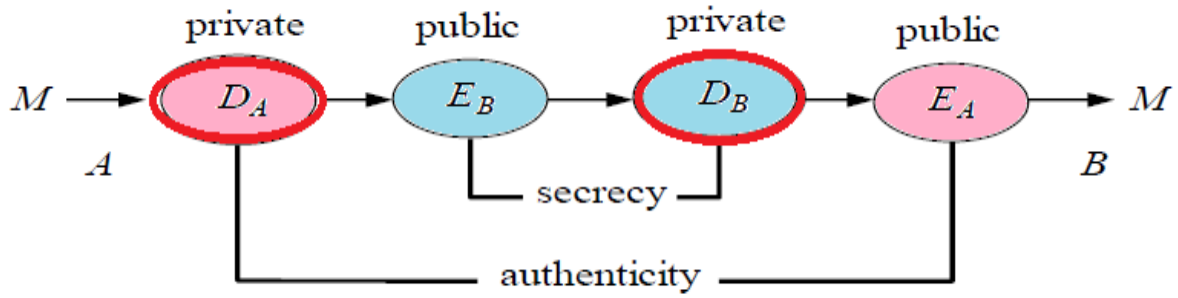


Figure:Public-Key Cryptosystem: Authentication and Secrecy

In simplified terms, the following figure shows how to use the public key to the achievement of security and authentication.



H.W :

What are the differences between the security and authentication?

Information Hiding

Steganography and Watermarking

- It employs technologies offered by numerous science disciplines:
  - Digital Signal Processing (Images, Audio, Video)
- Information Hiding is a branch of computer science that deals with concealing the existence of a message.
  It is related to cryptography whose intent is to render messages unreadable except by the intended recipients.

  - Cryptography
  - Information Theory\Coding Theory
  - Data Compression
  - Human Visual/Auditory perception

- There are two important sub-disciplines of Information Hiding
  - Steganography
  - Watermarking

  Information Hiding is applied to: (or carrier files) Images (Most frequently)

  - Audio
  - Video
  - Text
  - Executable programs

Steganography

- Steganography literally means “covered writing”.
- Steganography’s primary goal is to hide data within some other data (carrier file) such that the hidden data cannot be detected even if it is being sought.

Steganography is the science that serves to hide a specific message in a suitable cover file without making a noticeable changing with the cover that bring an attention of HSS (Human Sense Systems). Steganography simply takes one piece of information and hides it within another Computer files (images, sounds recordings, even disks) contain unused or insignificant areas of data. Steganography takes advantage of these areas, replacing them with information. The files can then be exchanged without anyone knowing what really lies inside of them.

Steganography types:

There are three basic types of Steganography:

1- ***Pure Steganography:*** Pure Steganography does not require the prior exchange of a stego-key, so both sender and receiver have to access the embedding and extraction algorithms.If an outsider knows the extraction algorithm, he can extract the secret message out of every cover sent between the two parties

   1. The embedding process can be described as the mapping:
      E: C x M C., where C is Cover, M is Message
   2. The Extraction process consists of mapping:
      D: C M

2- ***Secret Key Steganography:*** Secret key Steganography uses stego-key to embed the secret message into a cover and extracts the secret message using the same stego-key. Both parties could agree on the key before sending the secret message.

   1. The embedding process can be described as :
      EK: C x M x K C
      (where K is the key and M is the Message and C is Cover).
   2. The Extraction process consists of:
      DK: C x K M

      Secret Key Steganography requires the exchange of some keys, although transmission of additional secret information subverts the invisible communication.

3- ***Public Key Steganography:*** Public key Steganography requires a public key to embed the secret message and a private key in reconstruct process.

Least significant bit (LSB) insertion.

It is a common, simple approach to embedding information in image.

24-bit images: These images have a 24 bit value for each pixel in which each 8 bit value refer to the colors RED BLUE and GREEN. We can embed 3 bits of information in each pixel one in each LSB position of the three 8 bit values in 24 bit value. Increase or decrease of the value by changing the Least Significant bit doesn't change the appearance of the image much so the resulted Stego image looks exactly same as the cover image.

8-bit images: In these images 1 bit of information can be hidden in each pixel. The pointers to entries in the palette are changed. A change of even one bit could mean the difference between a shade of red and a shade of blue. Such a change would be noticeable on the displayed image, for this reason data-hiding experts recommend using grey-scale palettes.

Example of LSB

insertion : hide the letter G in a carrier file

G in ASCII is the binary string

01000111 suppose a sequence of 8 bytes

had the values

01010100 11010101 11001100 11110001 00011101 01010001 11001100 11001000 hiding the 8 bits representing G in the LSB of the eight carrier bytes

results in

01010100 11010101 11001100 11110000 00011100 01010001 11001101 11001000 - this changed 4 bits (in italics); in general, about 50% of the bit values change

Watermarks

1. Watermarks have been proposed for Copyright Protection of digital images, audio and video and, extensively, multimedia products.

2. Watermarks are digital signals that are embedded into other digital signals (carriers file).

3. The carrier signal is not affected strongly by such an embedding (watermarks are invisible).

4. A watermark should represent exclusively the copyright owner of the product and can be detected only by him/her.

5. Watermarks must be robust to any product modification that does not degrade its quality.

6. Resistance against any intentional attack is required

7. In a watermarking scheme can distinguish between ***<u>three fundamental stages</u>***

- Watermark generation, aims at producing the watermark pattern.
- Watermark embedding, can be considered as a superposition of watermark signal on the original image.
- Watermark detection, performed using watermark correlators or hypothesis testing

What are the differences between watermarking and steganography?

- Digital watermarking is a concept closely related to steganography, in that they both hide a message inside a carrier file.
- However, what separates them is their goal.

- Watermarking tries to hide a message related to the actual content of the carrier file,
- while in steganography the carrier file has no relation to the message, and it is merely used as a cover to hide its existence.

Steganography/Watermarking versus Cryptography

The purpose of both is to provide secret communication.

Cryptography hides the contents of the message from an attacker, but not the existence of the message.

Steganography / watermarking even hide the very existence of the message in the communicating data.

Consequently, the concept of breaking the system is different for cryptosystems and stegosystems (watermarking systems).

♦ cryptographic system is broken when the attacker can read the secrete message.

♦ Breaking of a steganographic /watermarking system has two stages:

1. The attacker can detect that steganography/watermarking has been used.
2. The attacker is able to read, modify or remove the hidden message.

A steganography/watermarking system is considered as insecure already if the detection of steganography/watermarking is possible.

.