

Introduction to Cryptography

UNIT I

Problems Addressed by Cryptography

- Key Agreement
- Secure Communication

Key Agreement

- Exchanging a key among two parties to access confidential message

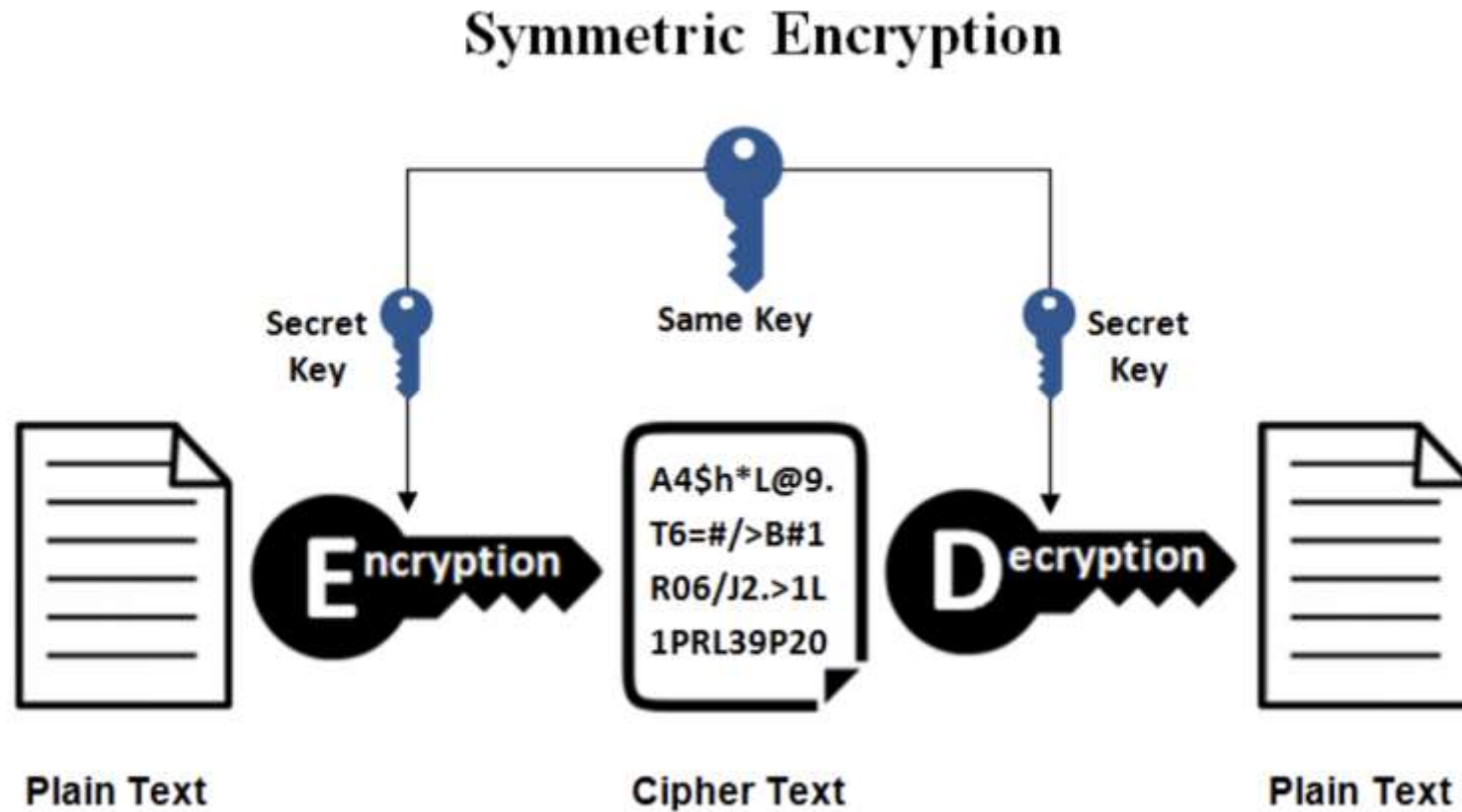
Secure Communication

- There is challenge to maintain
 - Confidentiality
 - Integrity

Types of Cryptographic Primitives

- Symmetric Key primitives (private-key cryptography)
 - Same Key (k) used at both ends
 - Computationally efficient
 - Key Agreement a Big issue
- Asymmetric Key primitives (public-key cryptography)
 - Different Key (k) used at both the ends
 - Computationally inefficient
 - No key agreement required

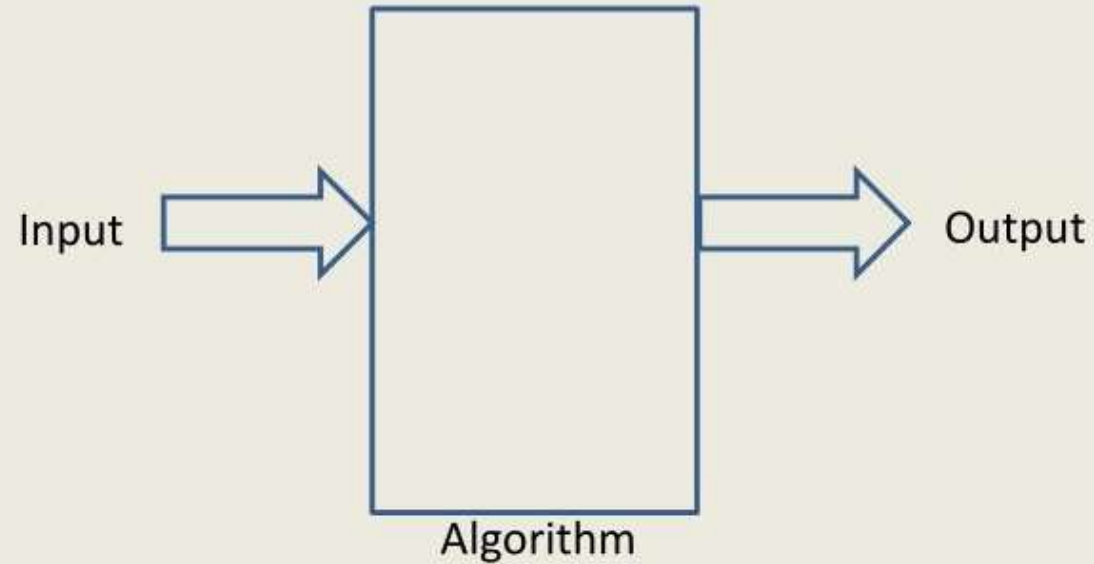
Symmetric Key Cryptography



Cryptographic Algorithms

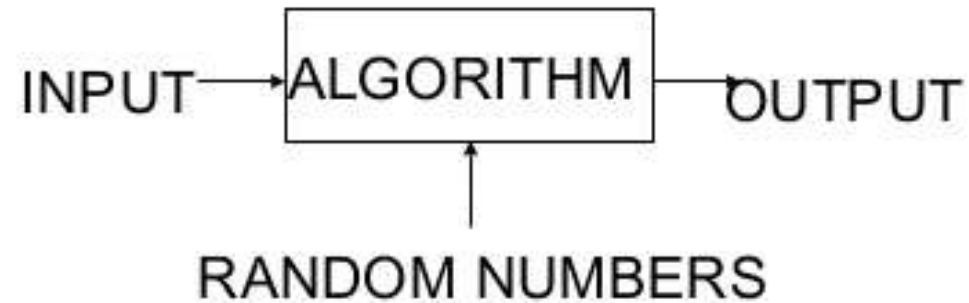
- Deterministic
- Randomized

Deterministic Algorithm



- The **output** as well as the **running time** are functions only of the input.

Randomized Algorithms



- In addition to input, algorithm takes a source of random numbers and makes random choices during execution;
- Behavior can vary even on a fixed input;

Algorithms in Symmetric Key Cryptography

- GEN
 - To Generate a randomized key (K)
- ENC
 - To convert message (plaintext) to Ciphertext using key (K)
- DEC
 - To convert Ciphertext to message (plaintext) using key (K)

Properties expected from a Secure Cipher

- Correctness

- $DEC_k(\underline{ENC_k(m)}) := m$

- Privacy

- Ciphertext **c** should not reveal anything about **m**

Challenges to formalize Privacy

- Definition 1

- An Encryption process is secure if the ciphertext does not reveal the underlying key.
- Consider an ENC algorithm which always outputs plaintext as the ciphertext.

- Definition 2

- An Encryption process is secure if the ciphertext does not reveal the underlying plaintext.
- Consider an ENC algorithm where the first 1% of the ciphertext is the same as the first 1% of the plaintext.

Challenges to formalize Privacy Definition

- Definition 3

- An Encryption process is secure **if the ciphertext does not reveal any character of the underlying plaintext.**
- Consider an ENC algorithm where the ciphertext reveals whether the underlying plaintext is less than or greater than a certain value.

- Definition 4

- An Encryption process is secure **if the ciphertext does not reveal any meaningful information about the underlying plaintext.**
- The notion of the meaningful information varies from application to application.

Challenges to formalize Privacy Definition

- Definition 5

- An Encryption process is secure **if the ciphertext does not help to compute any function of the underlying plaintext.**
- Precisely what we expect from a secure cipher. But there are certain loopholes in the above definition.
 - How to formalize whether a given ciphertext helps to compute a given function of the underlying plaintext?
 - What is the underlying adversary / attack model?
 - Is the adversary passive or malicious?
 - Does the adversary have access to any kind of “additional Information”?

Attack Models

- Ciphertext Only Attack (COA)
- Known plaintext Attack (KPA)
- Chosen plaintext Attack (CPA)
- Chosen Ciphertext Attack (CCA)

Note: In all attack models, the goal of the adversary is to compute some function of the underlying plaintext from the ciphertext.

Ciphertext Only Attack (COA)

- Simplest possible attack
- Attacker have access to the ciphertext
- No other additional knowledge is available other than the ciphertext.

Known Plaintext Attack (KPA)

- Attack has access to several (plaintext, ciphertext) pairs under the same k
 - E.g. the first word in an email is usually “hello” / “dear”, etc

Chosen Plaintext Attack (CPA)

- Attacker gets “encryption oracle” service --- active attack
 - Gets **encryption** of the plain-texts of **its choice**, without the knowledge of sender / receiver

Chosen Ciphertext Attack (CCA)

- It is the strongest possible attack model
- Attacker gets “encryption oracle” plus “decryption oracle” service.
 - Gets decryption of ciphertexts of its choice.

Keys and Kerckhoffs' Principle

- To main security, key should be definitely a secret
- What about ENC and DEC algorithm?
 - More security by keeping them private too?
- Kerckhoffs' Principle:
 - “A cryptosystem should be secure even if everything about the system, except the key, is a public knowledge”.

Importance of Kerckhoffs' Principle

- Maintaining the privacy of a key is relatively easier.
 - Key size \approx 100 bits, Program size : 1000 times larger
 - Algorithms can be leaked, reverse engineered
- Easy to replace a key if the key is exposed.
- Infeasible to assign a secret pair of algorithm for every pair of parties.
- Published designs undergo public scrutiny and so likely to be more secure.

Note: **Dangerous to use a Proprietary Encryption Scheme**

Classical Cypher & their Cryptoanalysis

Classical Cipher

- Shift cipher
- Mono-alphabetic substitute cipher
- Poly-alphabetic substitution (Vigenere) cipher

Shift Cipher

- Plain-text and cipher-text character $\in \{a, \dots, z\}$
 - Encryption: shift each plain-text character by k positions “forward”
 - Decryption: shift each cipher-text character by k positions “backward”
 - $k \in \{0, \dots, 25\}$ and randomly selected by the key-generation algorithm
- Mathematical interpretation of the shift cipher
 - Interpret the set $\{a, \dots, z\}$ as $\{0, \dots, 25\}$
 - $K = \{0, \dots, 25\}$ and $M = C = \text{set of strings over } \{0, \dots, 25\}$
- GEN
 - $k \in_R K$
- ENC
 - Input: $m_i \in M$ and k
 - Process: $c_i := (m_i + k) \bmod 26$
 - Output: $c_i \in C$
- DEC
 - Input: $c_i \in C$ and k
 - Process: $c_i := (m_i - k) \bmod 26$
 - Output: $m_i \in M$

Cryptoanalysis of Shift Cipher

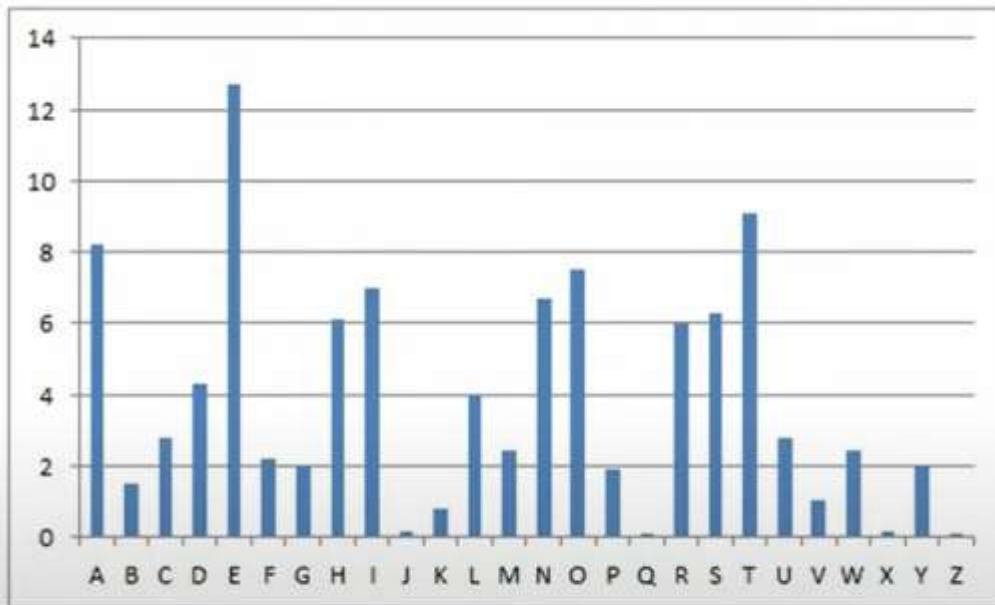
- Plaintext: $m = (m_1, \dots, m_l)$
- Ciphertext: $c_i = (c_1, \dots, c_l)$
- Attack model: Ciphertext Only Attack (COA)
- Information known to attacker
 - Ciphertext
 - Process through which the ciphertext is generated i.e. $c_i := (m_i + k) \bmod 26$
- Attack
 - An attacker can try to decrypt c *with all possible k* (brute force)
 - Easy to mount: only 26 candidate keys
- Learning
 - **Sufficient key-space principle:** Any secure cipher must have a key-space that is not vulnerable to exhaustive search.

Mono-alphabetic Substitution Cipher

- Map each plain-text character to an arbitrary cipher-text character in a one-to-one fashion
 - Key: A secret permutation (determined by the key generation algorithm)
- Is mono-alphabetic substitution cipher secure?
 - Brute-force attack is impractical
 - Number of candidate keys = $(26!) \approx 2^{88}$

Cryptoanalysis of Mono-alphabetic substitution Cipher

- Frequency Analysis: applicable when plaintext space is a natural language.
 - Idea: exploit the redundancy present in the underlying natural language.



Average English letter frequency

Bigram	Percentage	Bigram	Percentage
TH	3.15	HE	2.51
AN	1.72	IN	1.69
ER	1.54	RE	1.48
ES	1.45	ON	1.45
EA	1.31	TI	1.28
AT	1.24	ST	1.21
EN	1.20	ND	1.18

Average English
bigram frequency

THE, ING, AND, HER, ERE, ENT, THA, NTH,
WAS, ETH, FOR

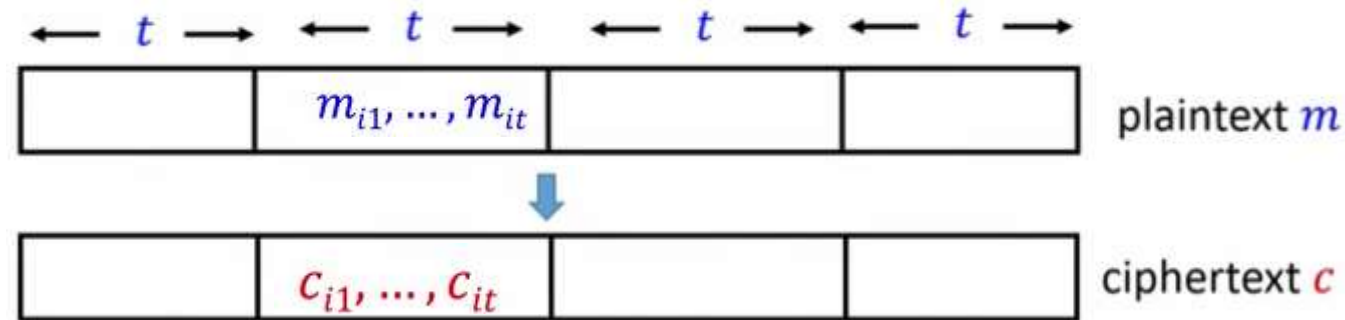
Popular English trigram in
decreasing order

Cryptoanalysis of Mono-alphabetic substitution Cipher

- Most frequently occurring character/ bigram/ trigram in the ciphertext, corresponds to most frequently occurring character / bigram/ trigram in the plaintext

Poly-alphabetic Substitution (Vignere) Cipher

- Idea: invoke multiple instances of shift cipher
 - In each instance, a plain-text character is mapped to a different ciphertext character.
- $M = C = \{0, \dots, 25\}^*$ $K = \{0, \dots, 25\}^t$, t randomly chosen by GEN
- Key-generation algorithm: output a uniformly random key $k = (k_1, \dots, k_t)$
- Encryption:



$$c_{ij} = (m_{ij} + k_j) \bmod 26$$

Example: Vigenere Cipher

- Key $k = CIPHER = (2,8,15,7,4,17)$ $t = 6$
- Plaintext $m = thiscryptosystemisnotsecure$

← 6 →						← 6 →						← 6 →						← 6 →						← 3 →			
19	7	8	18	2	17	24	15	19	14	18	24	18	19	4	12	8	18	13	14	19	18	4	2	20	17	4	
+ mod 26						+ mod 26						+ mod 26						+ mod 26						+ mod 26			
2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	
21	15	23	25	6		8	0	23	8	21	22	15	20	1	19	19	12	9	15	22	8	25	8	19	22	25	19

Cryptoanalysis of Vigenere Cipher

- Two stage approach
 - stage 1: Determine the length t of the unknown key k
 - Kasiski's method, index of coincidence method
 - Stage 2: Try to determine the characters k_1, k_2, \dots, k_t of the key k
- Stage II
 - t independent instances of letter frequency analysis

Learnings from cryptanalysis of Classical Ciphers

- Can be broken by launching a ciphertext-only attack
 - They can be even badly broken through stronger attack models
- Sufficient Key-space principle
 - Key space should be sufficiently large to make brute-force attack infeasible
- Designing secure cipher a tough task
-

Classical vs Modern Cryptography

- Classical cryptography was an art
 - No scientific foundation --- end result : disaster
- Modern Cryptography
 - Strong scientific foundations and principles
- Principle 1
 - Formal security definition
- Principle 2
 - Precisely stating any assumption used in construction
- Principle 3
 - Rigorous proof of security

OSI Security Architecture

An International Standard

Introduction

- Security architecture for OSI offers a systematic way of defining security requirements and characterizing the approaches to achieve these requirements.
- It was developed as an international standard.

Need for OSI Security Architecture

- To assess the security needs, of an organization effectively and choose various security products and policies.
- The need for some systematic way of defining the requirements for security and characterizing the approaches to satisfy those requirements.
- This is difficult enough in a centralized data-processing environment, and with the use of local area and wide area network, the problems are compounded.

The OSI security Architecture

- Such a systematic approach is defined by ITU-T (The International Telecommunication Union – Telecommunication Standardization Sector)
- It is a United Nation (UN) sponsored agency that develops standards, called recommendations, relating to telecommunication and to Open System Interconnection (OSI) recommendations X.800, security Architecture for OSI.

Benefits

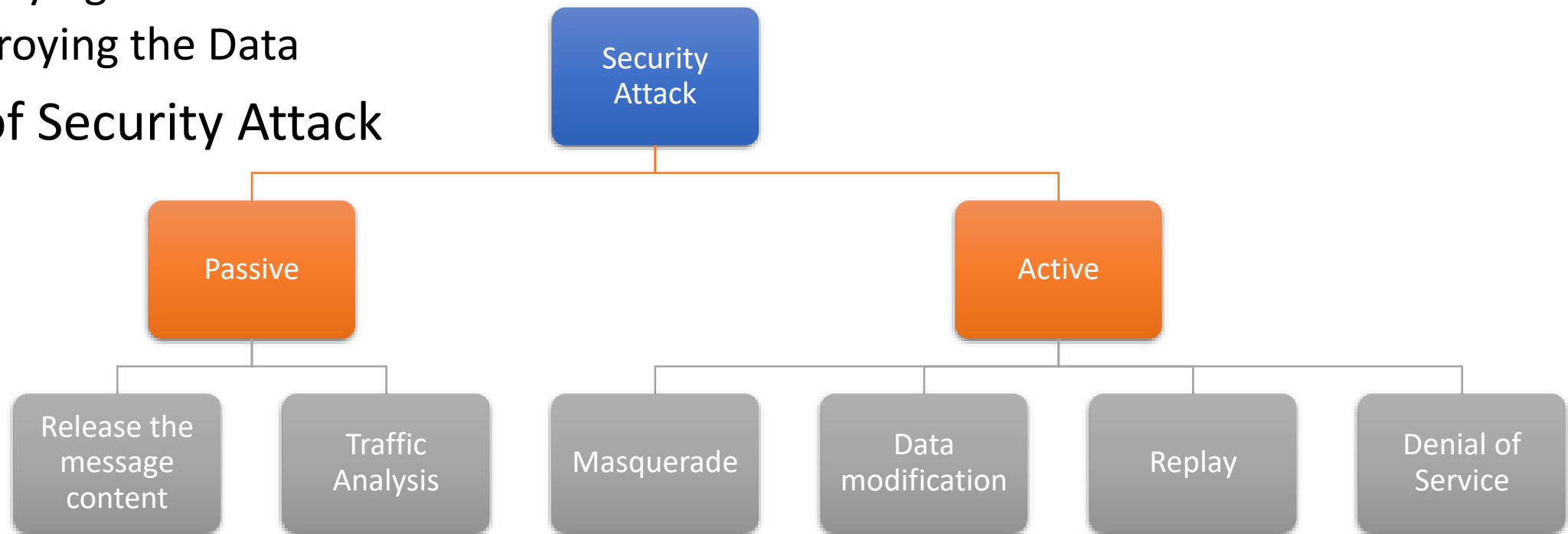
- The OSI security architecture is useful to managers as way of organization the task of providing security
- Furthermore, because this architecture was developed as international standards, computer and communications vendors have developed security feature for their products and services that relate to this structured definition of services and mechanisms.

OSI Security Architecture focus

- Security Attack
 - Any action that compromise the security of information owned by an organization.
- Security Mechanism
 - A process that is designed to detect, prevent or recover from a security attack
 - And security mechanism is a method which is used to protect your message from unauthorized entity.
- Security Services
 - Security Services is the services to implement security policies and implemented by security mechanism.

Security Attack

- Security attack is a process of gaining an access of data by unauthorized user
 - Accessing the Data
 - Modifying the Data
 - Destroying the Data
- Types of Security Attack



Security Mechanism



Encipherment

Digital Signature

Traffic padding

Notarization

Security Services

- Confidentiality
 - Ensures that the information in a computer system and transmitted information are accessible only for reading by authorized parties.
- Authentication
 - Ensures that the origin of a message or electronic document is correctly identified, with an assurance that the identity is not false
- Integrity
 - Ensures that only authorized parties are able to modify computer system assets and transmitted information
- Non-repudiation
 - Requires that neither the sender nor the receiver of a message be able to deny transmission.
- Access control
 - Requires that access to information resources may be controlled by or for the target system.
- Availability
 - Requires that computer system assets be available to authorized parties when needed.

Perfect Security

Introduction

- In 1949, Claude Shannon published a paper entitled “**Communication Theory of Secrecy Systems**” in the Bell Systems Technical Journal.
- This paper had a great influence on the scientific study of cryptography.
- Claude Shannon is also known as father of information theory.

Perfect Secrecy

- The notion of perfect security is also called as unconditional security, information –theoretic security.
- The attack model considered in the definition of perfect security is **Ciphertext-only attack**
- It is assumed that the attacker is computationally unbounded
- Informal definition
 - “irrespective of any prior info”. The attack has about m , the cipher-text c should not leak **no additional information** about the plain-text.

Basic Definition & properties

- $\Pi = (GEN, ENC, DEC)$, A publicly known cipher
- For the attacker, Π induces a probability distribution on \mathcal{M}, \mathcal{K} and \mathcal{C}
 - \mathcal{M} : *Plaintext space*
 - \mathcal{K} : *Key space*
 - \mathcal{C} : *Ciphertext space*
- Probability distribution on \mathcal{K} : *induced by GEN*
 - Almost always a uniform distribution
 - $\Pr[\mathbf{K} = k]$: *probability that GEN output the key k* \rightarrow typically $\frac{1}{|\mathcal{K}|}$


Basic Definition & properties

- Probability distribution on \mathcal{M} : *induced by any prior information about the underlying plaintext*
 - Ex: plaintext can be “Attack” with prob 0.7 or “Retreat” with prob. 0.3
 - $\Pr[\mathbf{M} = m]$: *probability that underlying plaintext is m*
- Probability distribution on \mathcal{C} : determined by the probability distribution over M, K and the steps of ENC.
 - $\Pr[\mathbf{C} = c]$: probability that ENC outputs the ciphertext c


Formal Definition

- An encryption scheme (GEN,ENC,DEC) over a plaintext space \mathcal{M} is perfectly-secure if for every probability distribution over \mathcal{M} and \mathcal{K} every plaintext $m \in \mathcal{M}$ and every ciphertext $c \in \mathcal{C}$, the following holds:

- $$\Pr [\mathbf{M} = m \mid \mathbf{C} = c] = \Pr [\mathbf{M} = m]$$



Posteriori probability that m
is encrypted in c



a-priori probability that m
might be communicated

Observing the cipher-text c **does not change** the
attacker's knowledge about the distribution of plaintext

Alternate Definitions

- Original definition: Probability of knowing a plain-text remains the same before and after seeing the cipher-text.

$$\Pr [\mathbf{M} = m \mid \mathbf{C} = c] = \Pr [\mathbf{M} = m]$$

- Alternate Definition:

- For every probability distribution over \mathcal{M} and \mathcal{K} , every plain-text $m_0, m_1 \in \mathcal{M}$ and every cipher-text $c \in \mathcal{C}$, the following holds

$$\Pr[\mathbf{C} = c \mid \mathbf{M} = m_0] = \Pr [\mathbf{C} = c \mid \mathbf{M} = m_1]$$

- Meaning: probability distribution of cipher-text is independent of plain-text

Proof that both definition are similar

$$\text{If } \Pr[\mathbf{M} = m \mid \mathbf{C} = c] = \Pr[\mathbf{M} = m], \quad \forall m \in \mathcal{M}, c \in \mathcal{C}$$

$$\text{Then } \Pr[\mathbf{C} = c \mid \mathbf{M} = m_0] = \Pr[\mathbf{C} = c \mid \mathbf{M} = m_1], \quad \forall m_0, m_1 \in \mathcal{M}, c \in \mathcal{C}$$

Proof: Let $m_0, m_1 \in \mathcal{M}, c \in \mathcal{C}$ be arbitrary plain-texts and cipher-text

$$\text{Given that } \Pr[\mathbf{M} = m_0 \mid \mathbf{C} = c] = \Pr[\mathbf{M} = m_0]$$

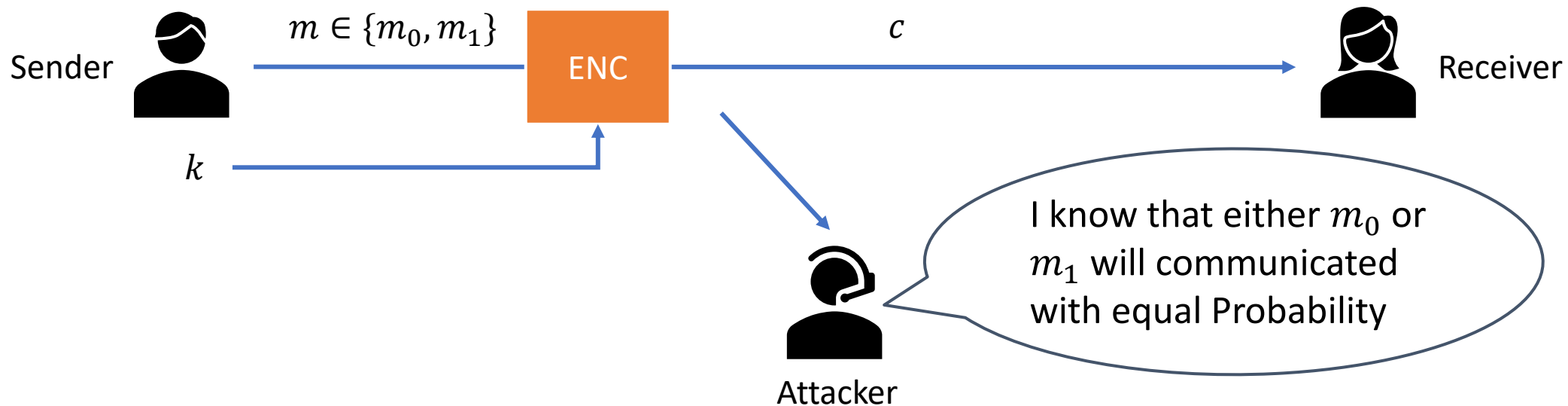
$$\frac{\Pr[\mathbf{C} = c \mid \mathbf{M} = m_0] \cdot \Pr[\mathbf{M} = m_0]}{\Pr[\mathbf{C} = c]} = \Pr[\mathbf{M} = m_0] \quad (\text{Expanding LHS by Bayes theorem})$$

$$\therefore \Pr[\mathbf{C} = c \mid \mathbf{M} = m_0] = \Pr[\mathbf{C} = c]$$

$$\text{Similarly, given that } \Pr[\mathbf{M} = m_1 \mid \mathbf{C} = c] = \Pr[\mathbf{M} = m_1]$$

$$\Pr[\mathbf{C} = c \mid \mathbf{M} = m_1] = \Pr[\mathbf{C} = c] \quad (\text{Expanding LHS by Bayes theorem and simplifying as above})$$

Second equivalent Definition

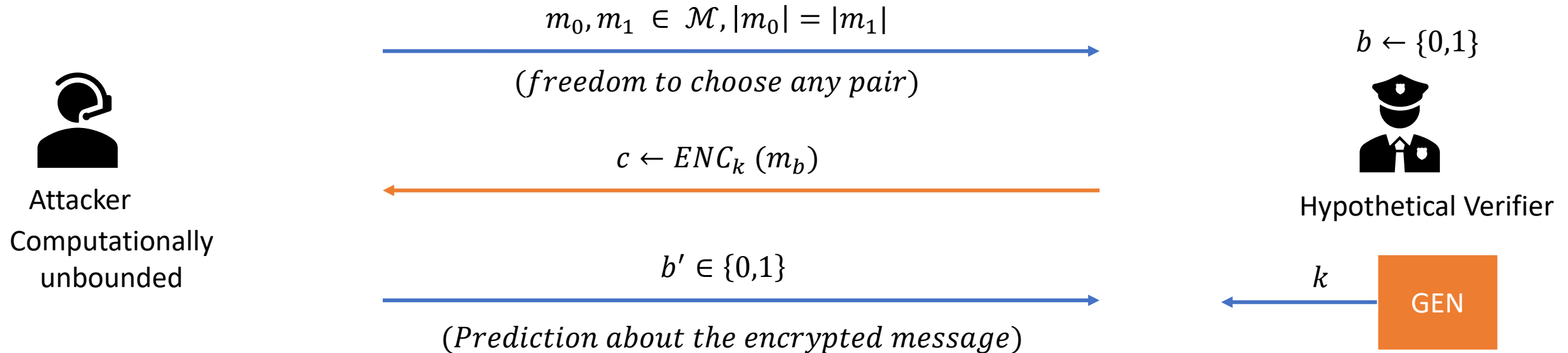


Perfect Secrecy: adversary should not get “any advantage” by seeing c

- Attacker should learn the underlying m from c only with probability $\frac{1}{2}$
- No better than guessing m

Second equivalent definition

- Definition is formalized with the help of challenge-response game (experiment)



Second Equivalent Definition

- Experiment output:
 - 1, if $b=b' \rightarrow$ interpretation : Attacker identified underlined message
 - 0, if $b \neq b' \rightarrow$ interpretation: Attacker failed to identify the underlying message

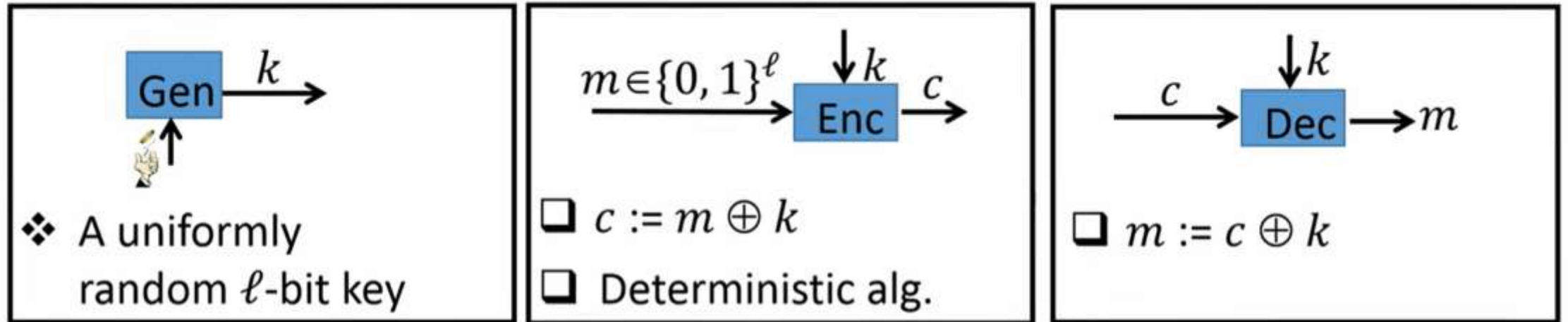
- $\Pi = (GEN, ENC, DEC)$ over \mathcal{M} is perfectly – indistinguishable if for every \mathcal{A}

$$\Pr(\text{PrivK}_{\mathcal{A}, \Pi}^{eav} = 1) = \frac{1}{2}$$

Limitations of perfect Security

Vernam Cipher (One-time pad encryption)

$$\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0, 1\}^\ell$$



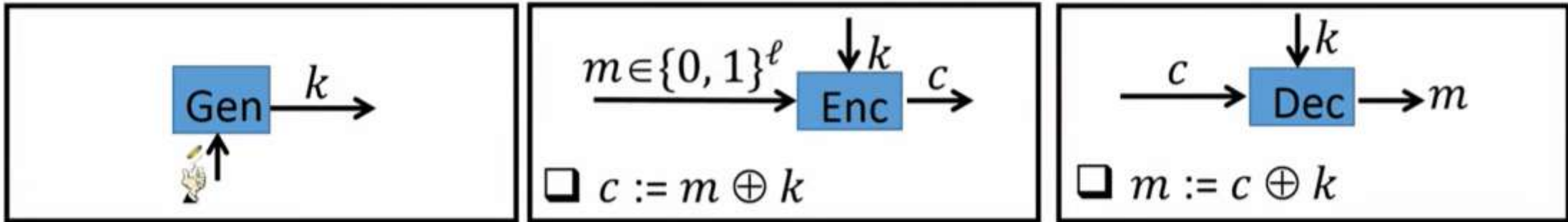
□ The bit-wise XOR operation is defined as follows:

$$0 \oplus 0 = 0$$

$$1 \oplus 1 = 0$$

$$0 \oplus 1 = 1$$

Vernam Cipher (One-time pad encryption)



❑ **Limitation I** : Key k , should be as **large** as plaintext m

- ❖ To securely communicate 1GB file, sender and receiver need to agree on a uniformly random key of size 1GB

❑ **Limitation II** : Key k **cannot** be used to encrypt **more than one message**

- ❖ A **fresh key** for each instance of encryption

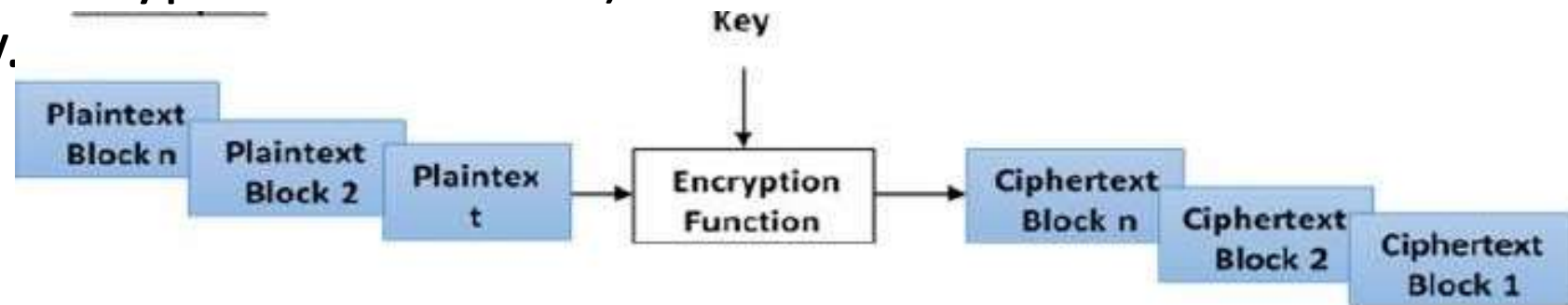
- Let key k is used to encrypt **two different messages** $m_0, m_1 \in \{0, 1\}^\ell$
- An eavesdropper on seeing $c_0 := m_0 \oplus k$ and $c_1 := m_1 \oplus k$ can compute

$$c_0 \oplus c_1 = m_0 \oplus m_1$$

Modern Symmetric Key Encryption

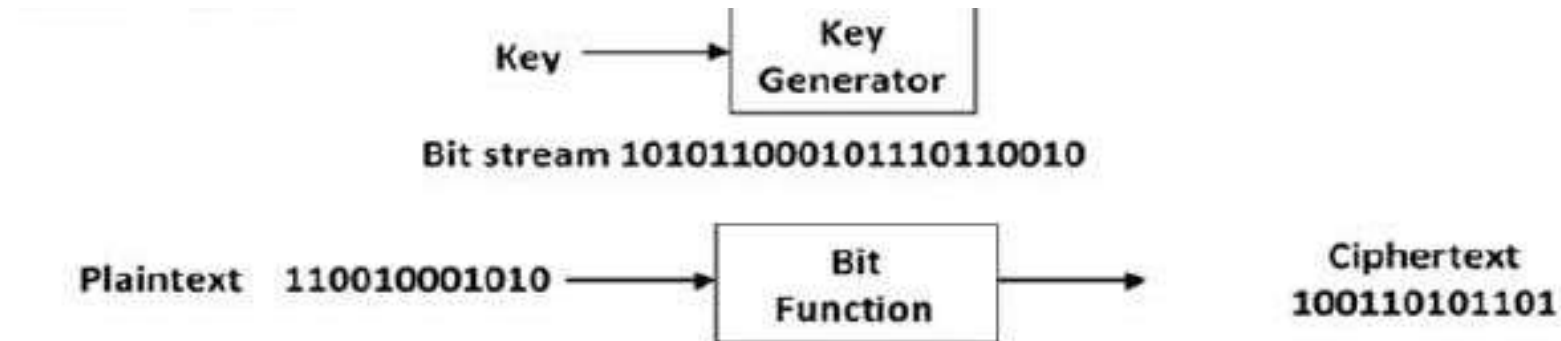
Block Ciphers

- In this scheme, the plain binary text is processed in blocks (groups) of bits at a time; i.e. a block of plaintext bits is selected, a series of operations is performed on this block to generate a block of ciphertext bits.
- The number of bits in a block is fixed.
- For example, the schemes DES (Data Encryption Standard) and AES (Advanced Encryption Standard) have block sizes of 64 and 128, respectively.



Stream Ciphers

- In this scheme, the plaintext is processed one bit at a time i.e. one bit of plaintext is taken, and a series of operations is performed on it to generate one bit of ciphertext.
- Technically, stream ciphers are block ciphers with a block size of one bit.



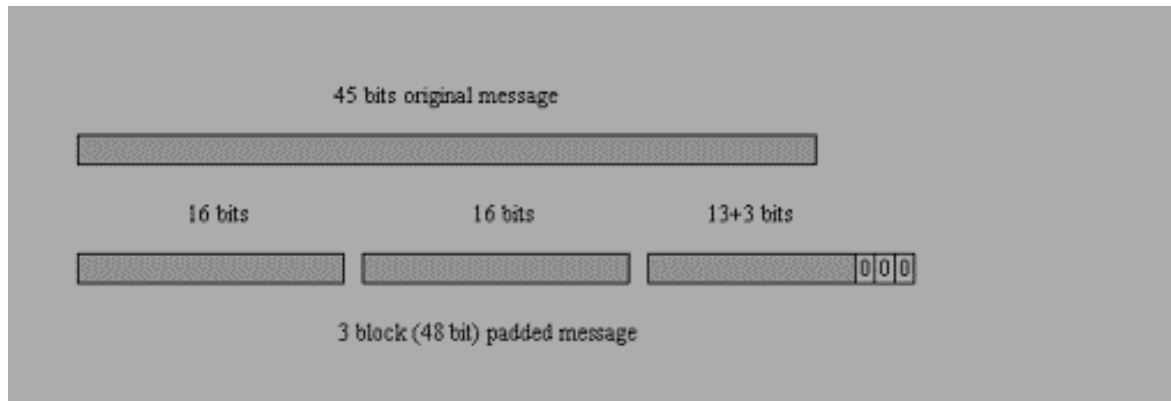
Block Ciphers

Block Size

- Avoid very small block size
 - Say a block size is m bits. Then the possible plaintext bits combinations are then 2^m . If the attacker discovers the plain text blocks corresponding to some previously sent ciphertext blocks, then the attacker can launch a type of 'dictionary attack' by building up a dictionary of plaintext/ciphertext pairs sent using that encryption key.
 - A larger block size makes attack harder as the dictionary needs to be larger.
- Do not have very large block size
 - With very large block size, the cipher becomes inefficient to operate. Such plaintexts will need to be padded before being encrypted.
- Multiples of 8 bit
 - A preferred block size is a multiple of 8 as it is easy for implementation as most computer processor handle data in multiple of 8 bits.

Padding in Block Cipher

- Block ciphers process blocks of fixed sizes (say 16 bits). The length of plaintexts is mostly not a multiple of the block size.
 - For example, a 45-bit plaintext provides two blocks of 16 bits each with third block of balance 13 bits. The last block of bits needs to be padded up with redundant information so that the length of the final block equal to block size of the scheme. In our example, the remaining 13 bits need to have additional 03 redundant bits added to provide a complete block.
- The process of adding bits to the last block is referred to as **padding**.



Block Cipher Schemes

- Digital Encryption Standard (DES)
 - The popular block cipher of the 1990s. It is now considered as a 'broken' block cipher, due primarily to its small key size.
- Triple DES
 - It is a variant scheme based on repeated DES applications. It is still a respected block cipher but inefficient compared to the new faster block ciphers available.
- Advanced Encryption Standard (AES)
 - It is a relatively new block cipher based on the encryption algorithm Rijndael that won the AES design competition.

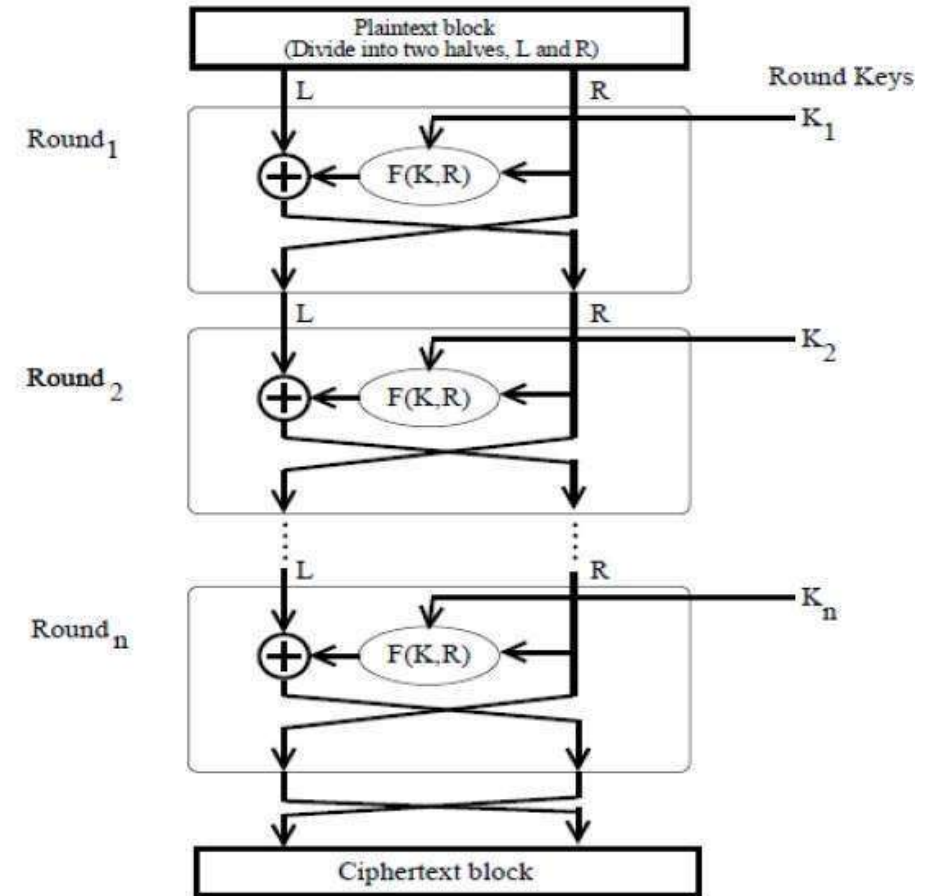
Feistel Block Cipher

Introduction

- Feistel Cipher is not a specific scheme of block cipher.
- It is a design model from which many different block ciphers are derived.
- DES is just one example of a Feistel Cipher.
- A cryptographic system based on Feistel cipher structure uses the same algorithm for both encryption and decryption.

Feistel Block Encryption Process

- The input block to each round is divided into two halves that can be denoted as L and R for the left half and the right half.
- In each round, the right half of the block, R, goes through unchanged. But the left half, L, goes through an operation that depends on R and the encryption key. First, we apply an encrypting function 'f' that takes two input – the key K and R. The function produces the output $f(R, K)$. Then, we XOR the output of the mathematical function with L.
- In real implementation of the Feistel Cipher, such as DES, instead of using the whole encryption key during each round, a round-dependent key (a subkey) is derived from the encryption key. This means that each round uses a different key, although all these subkeys are related to the original key.
- The permutation step at the end of each round swaps the modified L and unmodified R. Therefore, the L for the next round would be R of the current round. And R for the next round be the output L of the current round.
- Above substitution and permutation steps form a 'round'. The number of rounds are specified by the algorithm design.
- Once the last round is completed then the two sub blocks, 'R' and 'L' are concatenated in this order to form the ciphertext block.



Decryption Process

- The process of decryption in Feistel cipher is almost similar. Instead of starting with a block of plaintext, the ciphertext block is fed into the start of the Feistel structure and then the process thereafter is exactly the same.
- The process is said to be almost similar and not exactly same. In the case of decryption, the only difference is that the subkeys used in encryption are used in the reverse order.
- The final swapping of 'L' and 'R' in last step of the Feistel Cipher is essential. If these are not swapped then the resulting ciphertext could not be decrypted using the same algorithm.

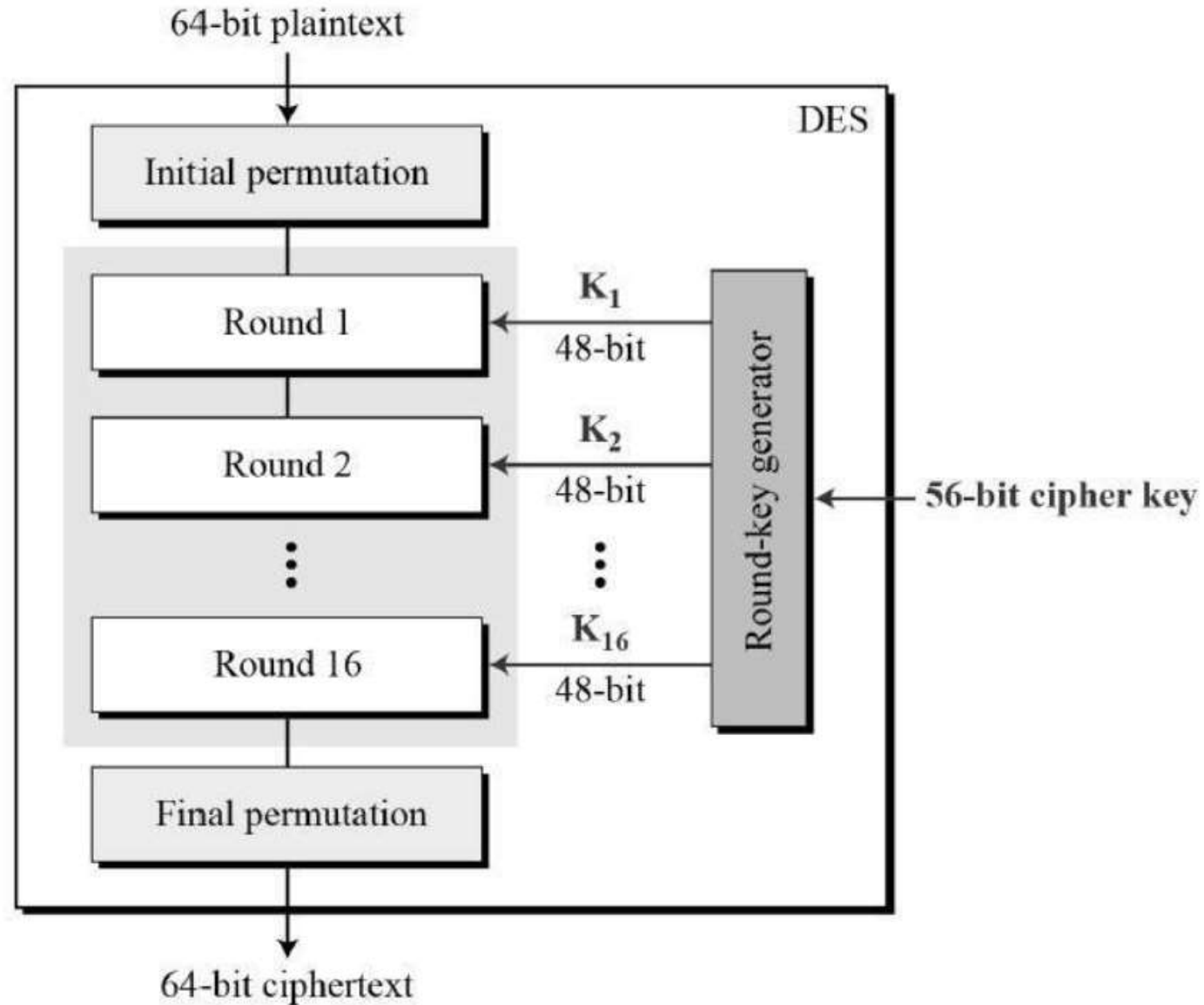
Data Encryption Standard

Introduction

- The Data Encryption Standard (DES) is a symmetric-key block cipher created in early 1970s by an IBM team and later adopted by the National Institute of Standards and Technology (NIST).
- The algorithm takes the plain text in 64-bit blocks and converts them into ciphertext using 48-bit keys.
 - Since it's a symmetric-key algorithm, it employs the same key in both encrypting and decrypting the data.
- DES is based on the Feistel block cipher, called LUCIFER, developed in 1971 by IBM cryptography researcher Horst Feistel. DES uses 16 rounds of the Feistel structure, using a different key for each round.

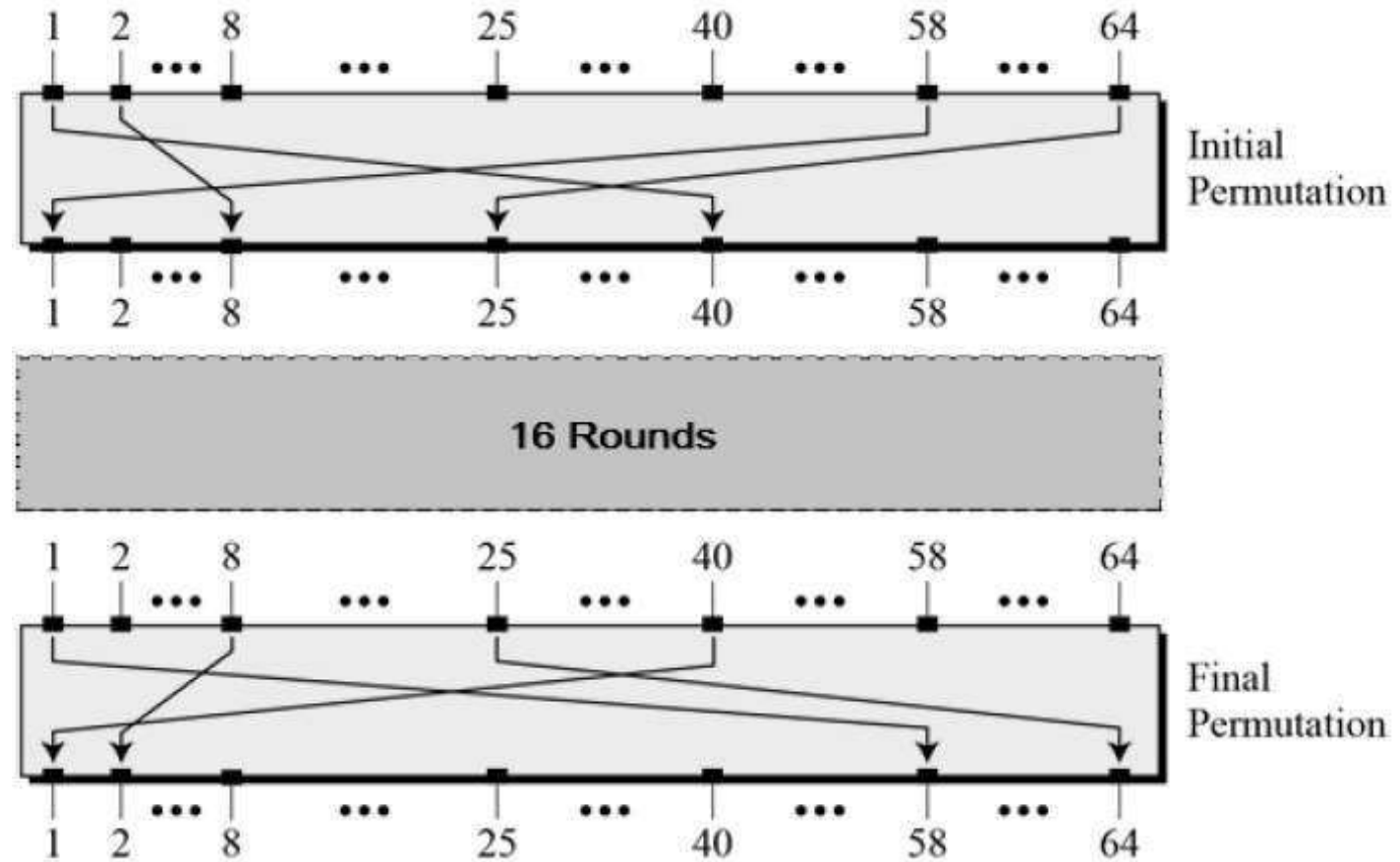
General Structure of DES

- All that is required to specify DES is –
 - Round Function
 - Key Schedule
 - Any additional processing
 - Initial Permutation
 - Final Permutation



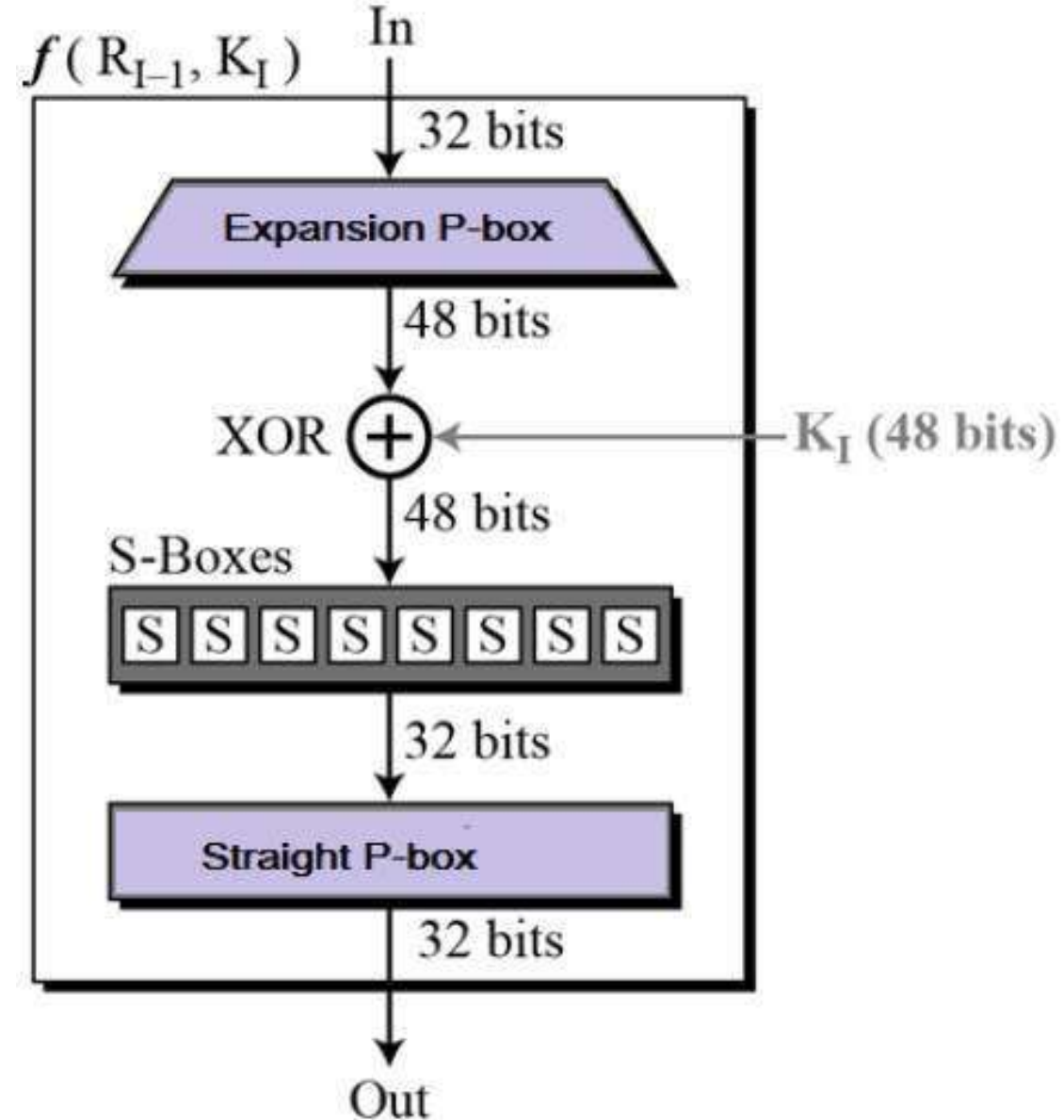
Initial and Final Permutation

- The initial and final permutation are straight Permutation boxes (P-boxes) that are inverses of each other.
 - They have no cryptography significance in DES.



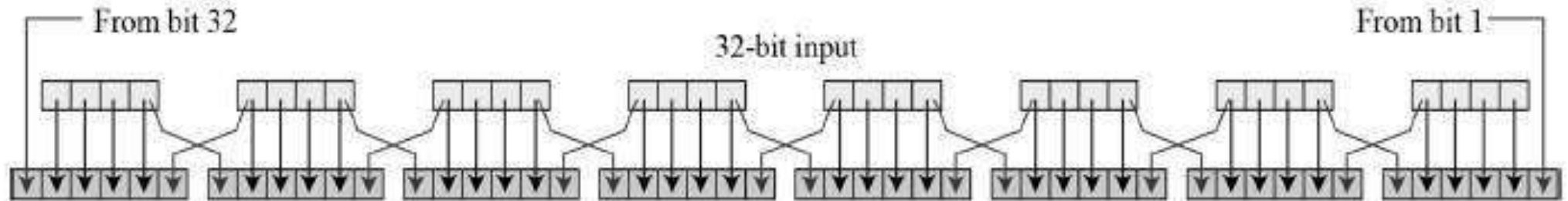
Round Function

- The heart of this cipher is the DES function, f . The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



Round Function

- Expansion Permutation Box
 - Since right input is 32-bit and round key is a 48-bit, we first need to expand right input to 48 bits. Permutation logic is graphically depicted in the following illustration –



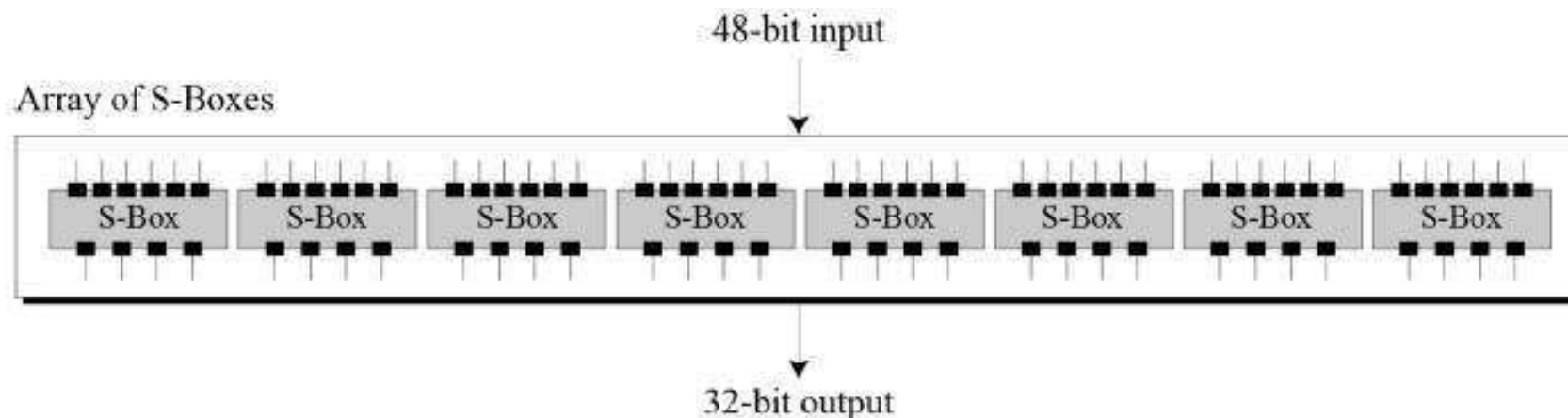
Round Function

- **XOR (Whitener)**

- After the expansion permutation, DES does XOR operation on the expanded right section and the round key. The round key is used only in this operation.

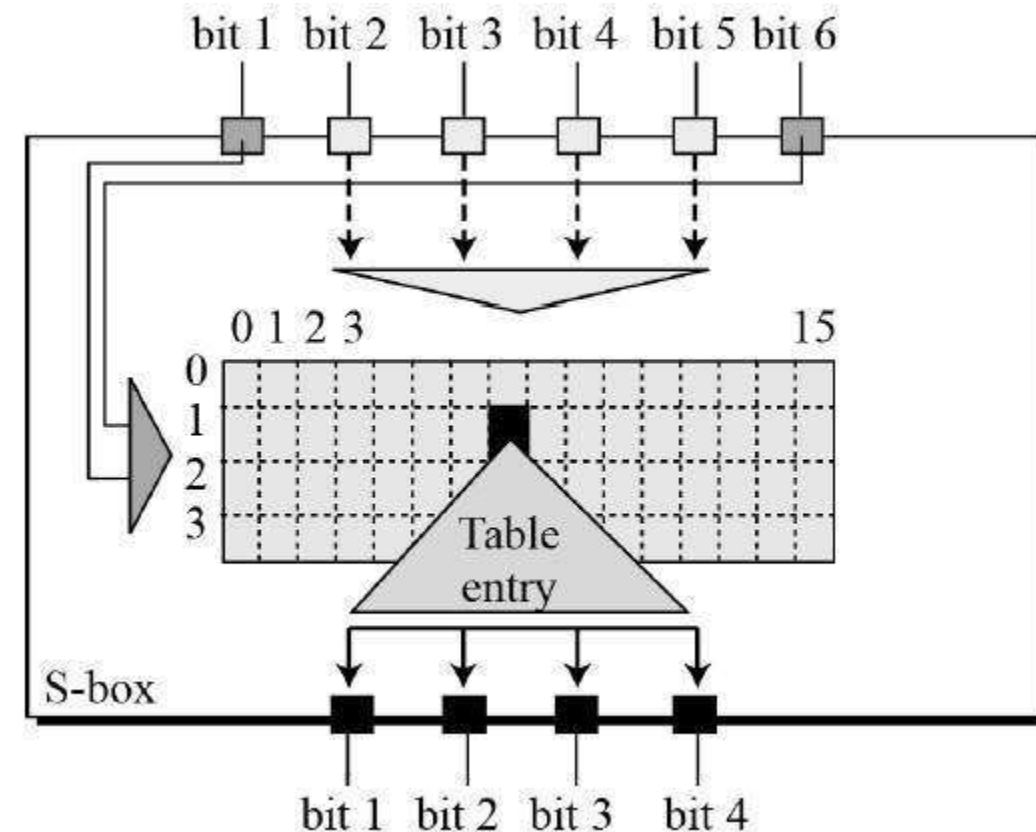
- **Substitution Boxes**

- The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the following illustration –



Round Function

- S-box rule
 - There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.



Round Function

- Straight Permutation
 - The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:

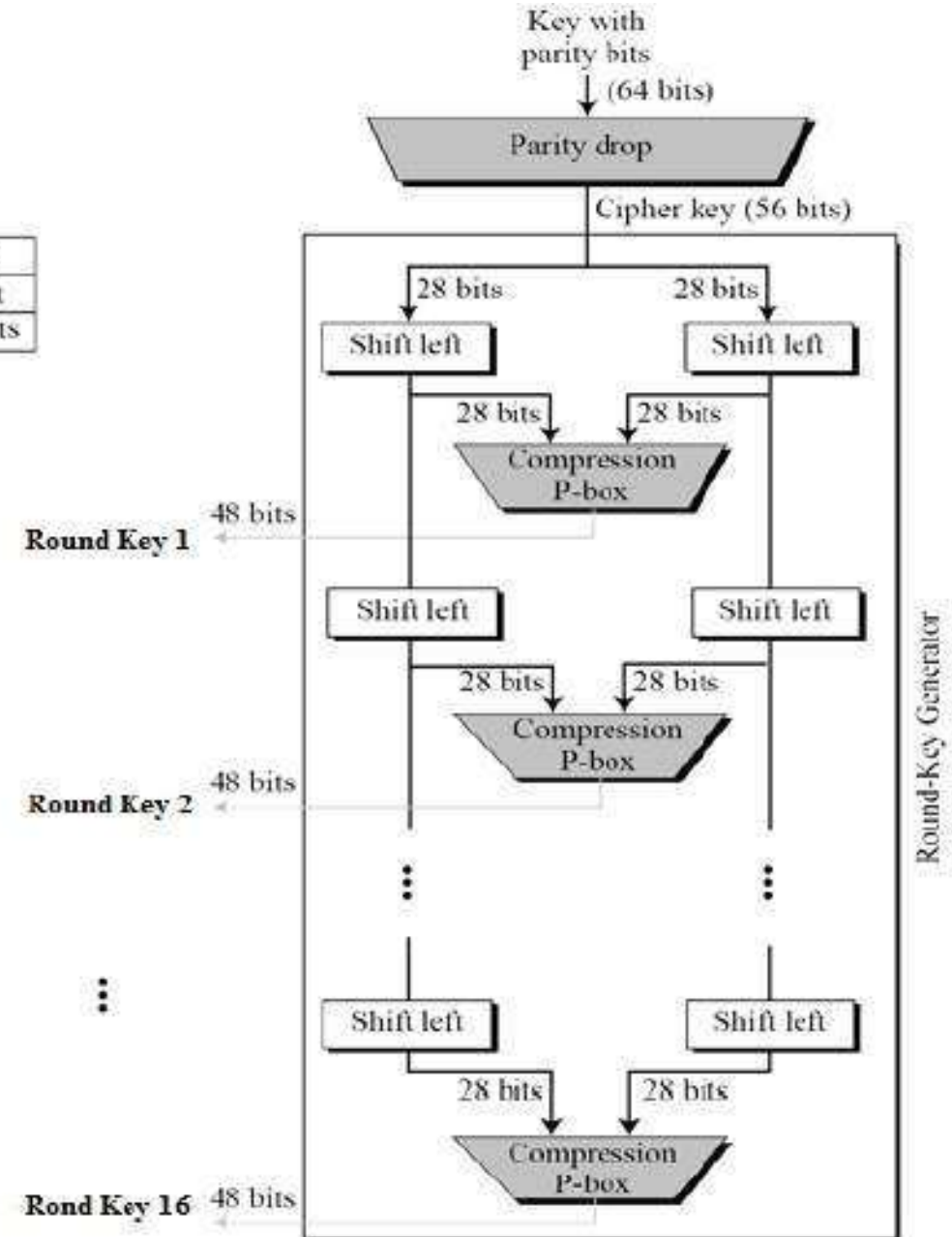
16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

- Working of DES Algorithm: https://youtu.be/8TET_mmwJaM

Key Generation

- The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key. The process of key generation is depicted in the following illustration –

Shifting	
Rounds	Shift
1, 2, 9, 16	one bit
Others	two bits



DES Analysis

- The DES satisfies both the desired properties of block cipher. These two properties make cipher very strong.
 - Avalanche effect – A small change in plaintext results in the very great change in the ciphertext.
 - Completeness – Each bit of ciphertext depends on many bits of plaintext.
- During the last few years, cryptanalysis have found some weaknesses in DES when key selected are weak keys. These keys shall be avoided.
- DES has proved to be a very well designed block cipher. There have been no significant cryptanalytic attacks on DES other than exhaustive key search.

Security of DES

Introduction

- DES, as the first important block cipher, has gone through much scrutiny. Among the attempted attacks, three are of interest:
 - Brute-force
 - The weakness of short cipher key in DES. Combining this weakness with the key complement weakness, it is clear that DES can be broken using 2^{55} encryptions.
 - Differential cryptanalysis
 - It has been revealed that the designers of DES already knew about this type of attack and designed S-boxes and chose 16 as the number of rounds to make DES specifically resistant to this type of attack.
 - Linear cryptanalysis.
 - Linear cryptanalysis is newer than differential cryptanalysis. DES is more vulnerable to linear cryptanalysis than to differential cryptanalysis. S-boxes are not very resistant to linear cryptanalysis. It has been shown that DES can be broken using 2^{43} pairs of known plaintexts. However, from the practical point of view, finding so many pairs is very unlikely.

Strength of DES

Introduction

- Since its adoption as a federal standard, there have been lingering concerns about the level of security provided by DES. These concerns, by and large, fall into two areas:
 - Key size
 - Nature of the algorithm.

Key Size

- With a key length of 56 bits, there are 2^{56} possible keys, which is approximately 7.2×10^{16} . Thus, on the face of it, a brute-force attack appears impractical. Assuming that, on average, half the key space has to be searched, a single machine performing one DES encryption per microsecond would take more than a thousand years to break the cipher.

Nature of DES Algorithm

- Another concern is the possibility that cryptanalysis is possible by exploiting the characteristics of the DES algorithm.
- The focus of concern has been on the eight substitution tables, or S-boxes, that are used in each iteration.
 - Because the design criteria for these boxes, and indeed for the entire algorithm, were not made public, there is a suspicion that the boxes were constructed in such a way that cryptanalysis is possible for an opponent who knows the weaknesses in the S-boxes.
 - This assertion is tantalizing, and over the years a number of regularities and unexpected behaviors of the S-boxes have been discovered. Despite this, no one has so far succeeded in discovering the supposed fatal weaknesses in the S-boxes.

Block Cipher Design Principles

Introduction

- The three critical aspects of block cipher design: the number of rounds, design of the function F , and key scheduling.
- DES Design Criteria
 - The criteria used in the design of DES focused on the design of the S-boxes and on the P function that takes the output of the S boxes. The criteria for the S-boxes are as follows:
 - No output bit of any S-box should be too close a linear function of the input bits.
 - Each row of an S-box should include all 16 possible output bit combinations.
 - If two inputs to an S-box differ in exactly one bit, the outputs must differ in at least two bits.
 - If two inputs to an S-box differ in the two middle bits exactly, the outputs must differ in at least two bits.
 - If two inputs to an S-box differ in their first two bits and are identical in their last two bits, the two outputs must not be the same.
 - For any nonzero 6-bit difference between inputs, no more than 8 of the 32 pairs of inputs exhibiting that difference may result in the same output difference.
 - This is a criterion similar to the previous one, but for the case of three S-boxes.

Criteria for Permutation P

- Criteria 1:
 - The four output bits from each S-box at round i are distributed so that two of them affect “middle bits” of round $(i + 1)$ and the other two affect end bits. The two middle bits of input to an S-box are not shared with adjacent S-boxes. The end bits are the two left-hand bits and the two right-hand bits, which are shared with adjacent S-boxes.
- Criteria 2:
 - The four output bits from each S-box affect six different S-boxes on the next round, and no two affect the same S-box.
- Criteria 3:
 - For two S-boxes j, k , if an output bit from S_j affects a middle bit of S_k on the next round, then an output bit from S_k cannot affect a middle bit of S_j . This implies that for $j = k$, an output bit from S_j must not affect a middle bit of S_j .

Number of Rounds

- The cryptographic strength of a Feistel cipher derives from three aspects of the design
 - the number of rounds
 - the function F
 - the key schedule algorithm.
- The greater the number of rounds, the more difficult it is to perform cryptanalysis, even for a relatively weak F .
- The heart of a Feistel block cipher is the function F . In DES, this function relies on the use of S-boxes.
- The more nonlinear F , the more difficult any type of cryptanalysis will be. There are several measures of nonlinearity.
 - In rough terms, the more difficult it is to approximate F by a set of linear equations, the more nonlinear F is.

S-box Design

- For larger S-boxes, such as 8×32 , the best method of selecting the S-box entries :
 - Random
 - Use some pseudorandom number generation or some table of random digits to generate the entries in the S-boxes.
 - This may lead to boxes with undesirable characteristics for small sizes (e.g., 6×4) but should be acceptable for large S-boxes (e.g., 8×32).
 - Random with testing
 - Choose S-box entries randomly, then test the results against various criteria, and throw away those that do not pass.
 - Human-made
 - This is a more or less manual approach with only simple mathematics to support it. It is apparently the technique used in the DES design.
 - This approach is difficult to carry through for large S-boxes.
 - Math-made
 - Generate S-boxes according to mathematical principles. By using mathematical construction, S-boxes can be constructed that offer proven security against linear and differential cryptanalysis, together with good diffusion.

Key Schedule Algorithm

- A final area of block cipher design, and one that has received less attention than S-box design, is the key schedule algorithm.
- With any Feistel block cipher, the key is used to generate one subkey for each round.
 - In general, we would like to select subkeys to maximize the difficulty of deducing individual subkeys and the difficulty of working back to the main key.
- No general principles for this have yet been promulgated.

Block Cipher modes of Operation

Block Cipher modes of Operation

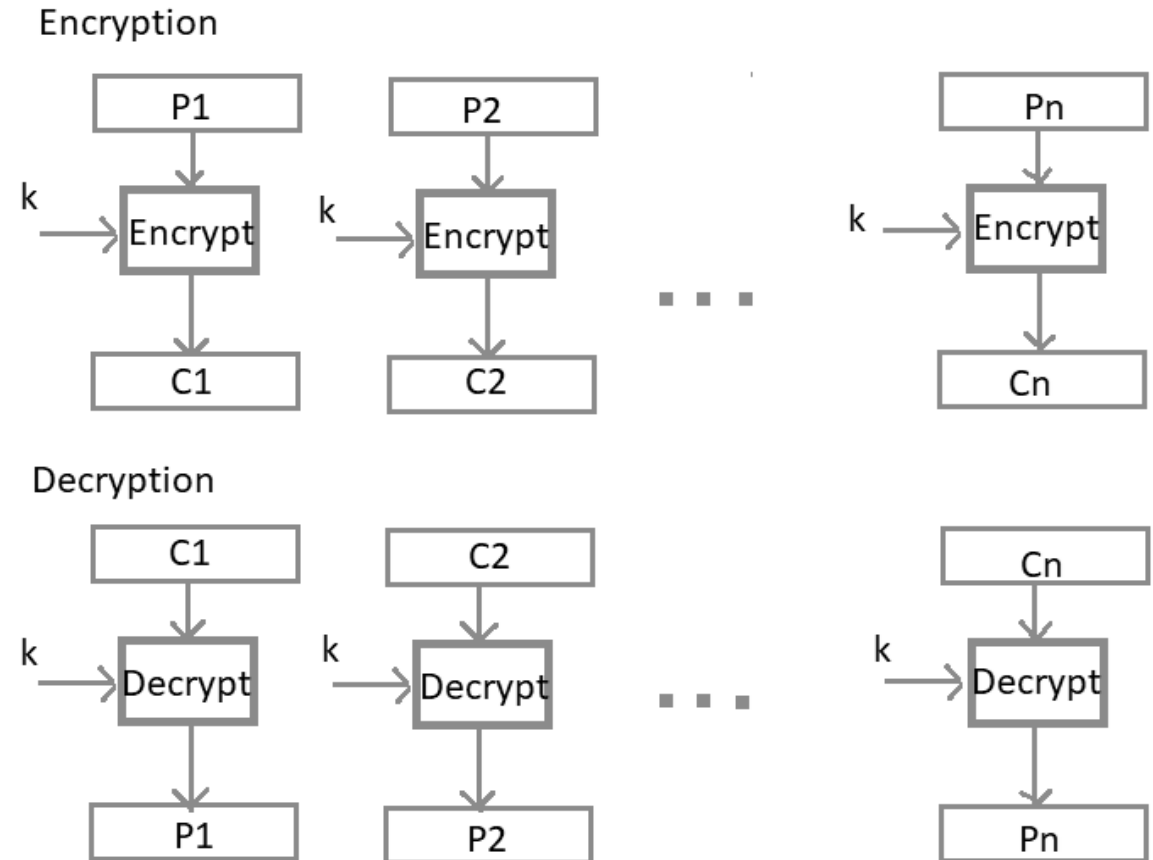
- Encryption algorithms are divided into two categories based on input type:
 - Block cipher
 - Stream cipher.
- Block cipher is an encryption algorithm which takes fixed size of input say b bits and produces a ciphertext of b bits again. If input is larger than b bits it can be divided further.
- For different applications and uses, there are several modes of operations for a block cipher.

Modes of Operation

- Electronic Code Book (ECB)
- Cipher Block Chaining (CBC)
- Cipher Feedback Mode (CFB)
- Output Feedback Mode (OFM)
- Counter Mode CTR

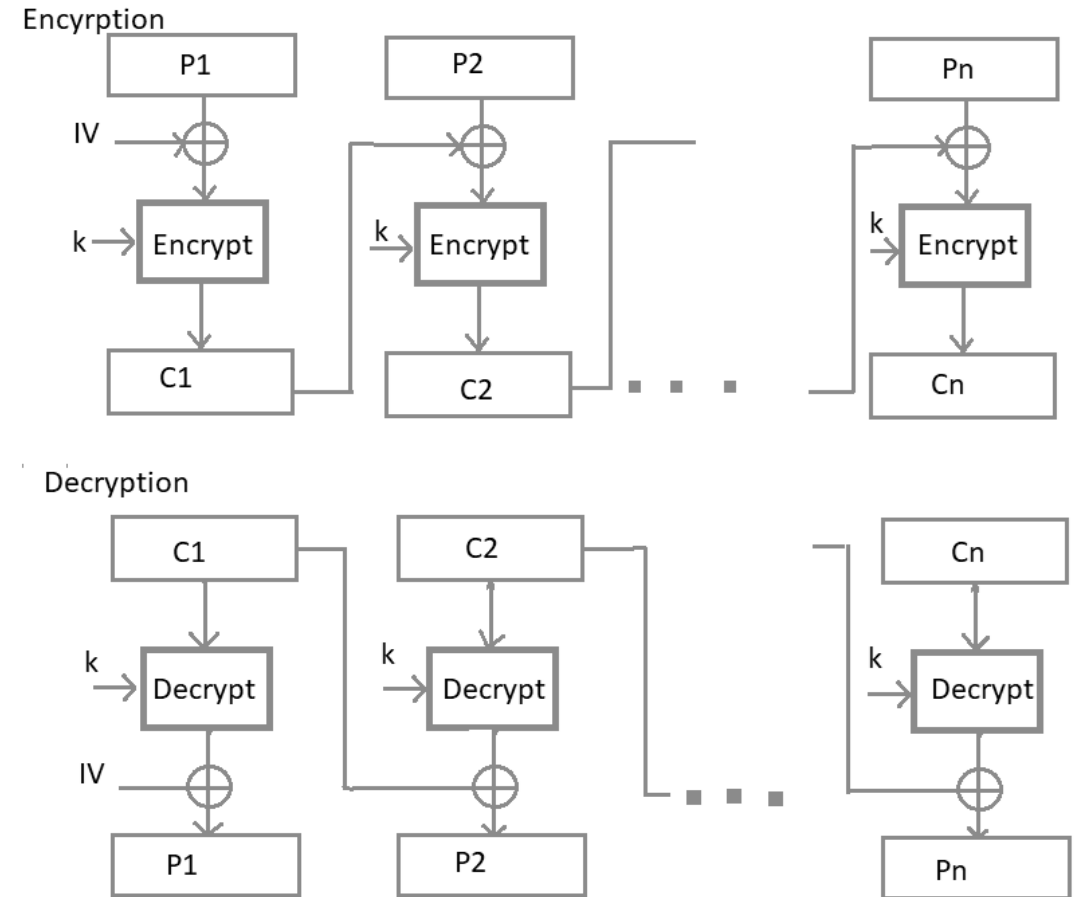
Electronic Code Book (ECB)

- Electronic code book is the easiest block cipher mode of functioning.
 - It is easier because of direct encryption of each block of input plaintext and output is in form of blocks of encrypted ciphertext.
 - Generally, if a message is larger than b bits in size, it can be broken down into bunch of blocks and the procedure is repeated.
- Advantages
 - Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.
 - Simple way of block cipher.
- Disadvantages
 - Prone to cryptanalysis since there is a direct relationship between plaintext and ciphertext.



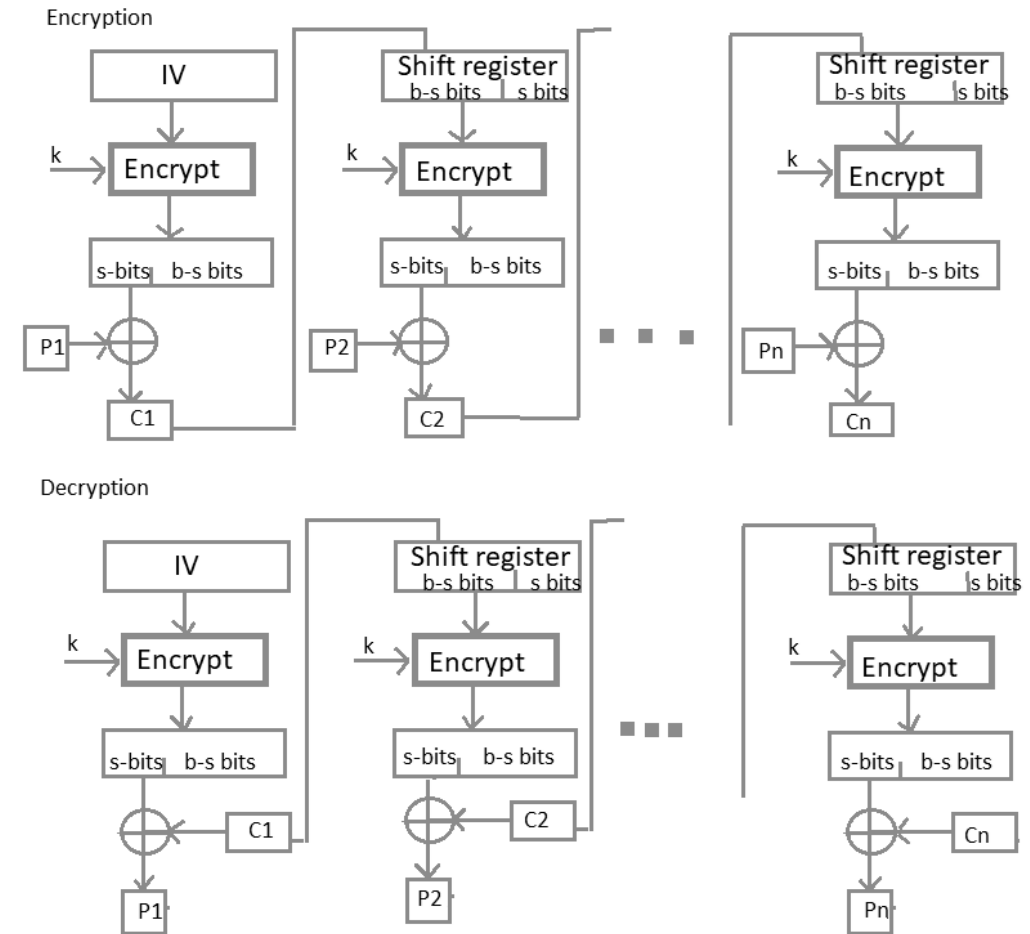
Cipher Block Chaining (CBC)

- Cipher block chaining or CBC is an advancement made on ECB since ECB compromises some security requirements.
 - In CBC, previous cipher block is given as input to next encryption algorithm after XOR with original plaintext block.
 - In a nutshell here, a cipher block is produced by encrypting a XOR output of previous cipher block and present plaintext block.
- Advantages
 - CBC works well for input greater than b bits.
 - CBC is a good authentication mechanism.
 - Better resistive nature towards cryptanalysis than ECB.
- Disadvantages
 - Parallel encryption is not possible since every encryption requires previous cipher.



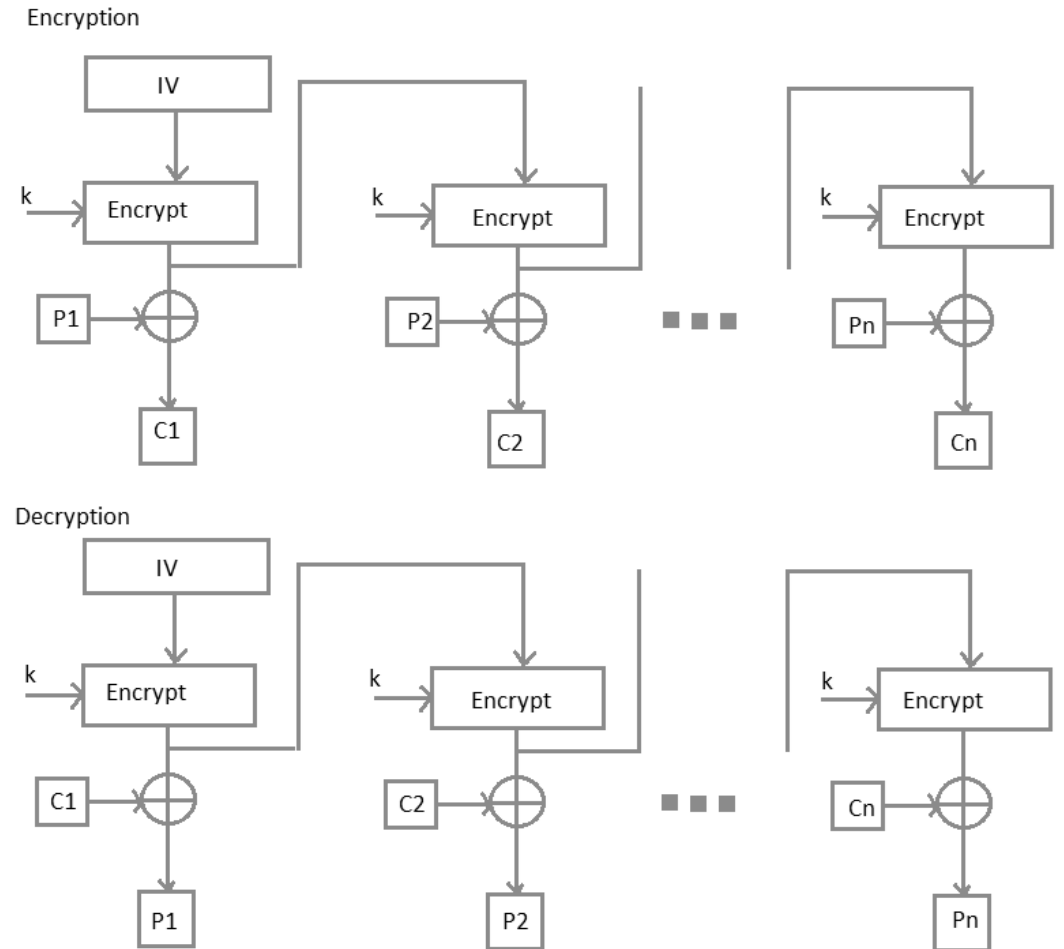
Cipher Feedback Mode (CFB)

- In this mode the cipher is given as feedback to the next block of encryption with some new specifications:
 - first an initial vector IV is used for first encryption and output bits are divided as set of s and $b-s$ bits the left hand side s bits are selected and are applied an XOR operation with plaintext bits.
 - The result given as input to a shift register and the process continues.
- Advantages
 - Since, there is some data loss due to use of shift register, thus it is difficult for applying cryptanalysis.



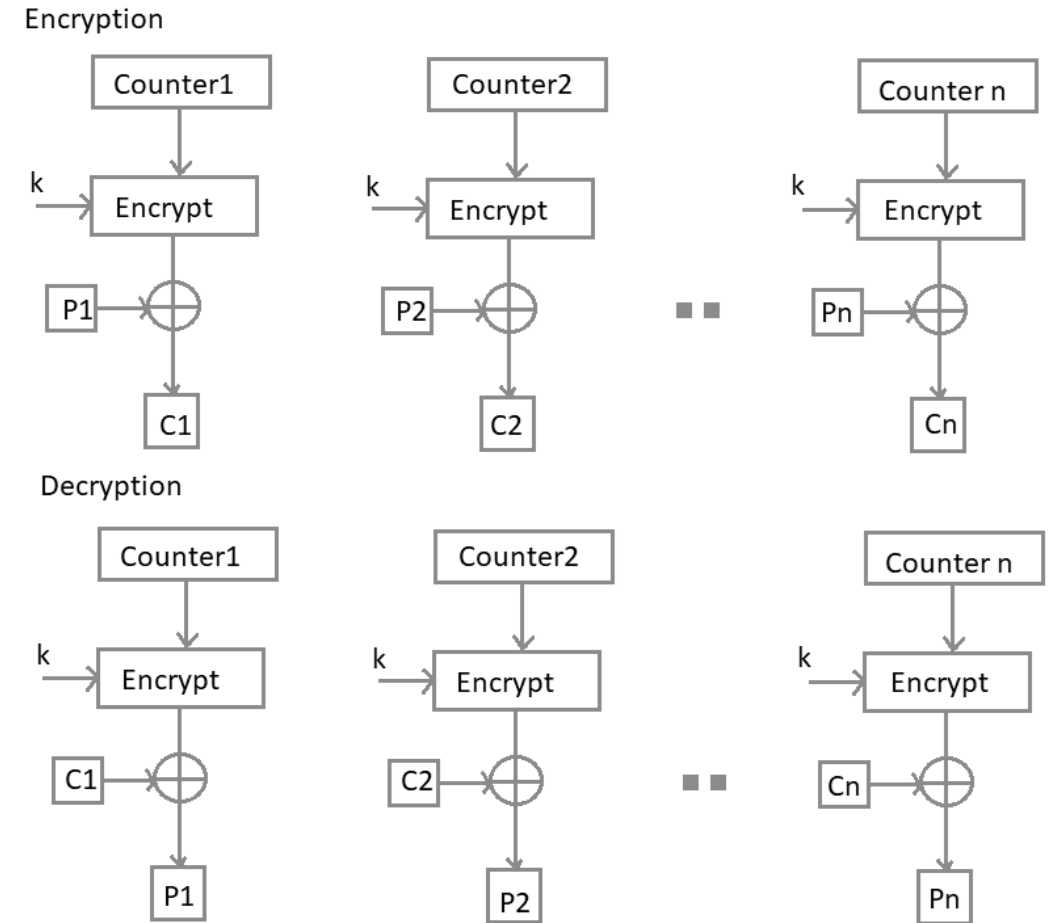
Output Feedback Mode

- The output feedback mode follows nearly same process as the Cipher Feedback mode except that it sends the encrypted output as feedback instead of the actual cipher which is XOR output.
 - In this output feedback mode, all bits of the block are sent instead of sending selected s bits.
 - The Output Feedback mode of block cipher holds great resistance towards bit transmission errors. It also decreases dependency or relationship of cipher on plaintext.



Counter Mode

- The Counter Mode or CTR is a simple counter based block cipher implementation.
- Every time a counter initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block.
- The CTR mode is independent of feedback use and thus can be implemented in parallel.



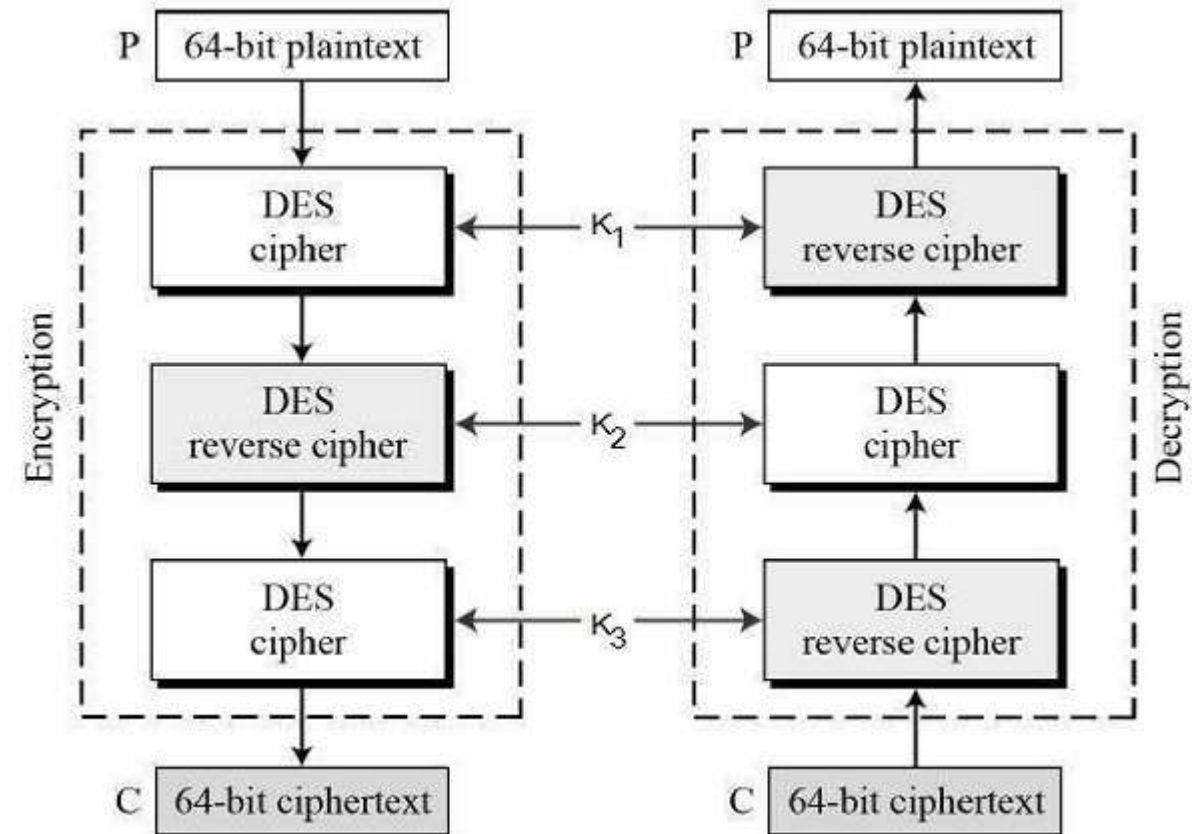
Triple DES

Introduction

- The speed of exhaustive key searches against DES after 1990 began to cause discomfort amongst users of DES.
 - However, users did not want to replace DES as it takes an enormous amount of time and money to change encryption algorithms that are widely adopted and embedded in large security architectures.
- The pragmatic approach was not to abandon the DES completely, but to change the manner in which DES is used.
 - This led to the modified schemes of Triple DES (sometimes known as 3DES).
- There are two variants of Triple DES
 - 3-key Triple DES (3TDES)
 - 2-key Triple DES (2TDES)

3-KEY Triple DES (3TDES)

- Before using 3TDES, user first generate and distribute a 3TDES key K , which consists of three different DES keys K_1 , K_2 and K_3 .
 - This means that the actual 3TDES key has length $3 \times 56 = 168$ bits.
- The encryption-decryption process is
 - Encrypt the plaintext blocks using single DES with key K_1 .
 - Now decrypt the output of step 1 using single DES with key K_2 .
 - Finally, encrypt the output of step 2 using single DES with key K_3 .
 - The output of step 3 is the ciphertext.
 - Decryption of a ciphertext is a reverse process. User first decrypt using K_3 , then encrypt with K_2 , and finally decrypt with K_1 .
- Due to this design of Triple DES as an encrypt–decrypt–encrypt process, it is possible to use a 3TDES (hardware) implementation for single DES by setting K_1 , K_2 , and K_3 to be the same value. This provides backwards compatibility with DES.



2-KEY Triple DES (2TDES)

- Second variant of Triple DES (2TDES) is identical to 3TDES except that K_3 is replaced by K_1 . In other words, user encrypt plaintext blocks with key K_1 , then decrypt with key K_2 , and finally encrypt with K_1 again.
 - Therefore, 2TDES has a key length of 112 bits.
- Triple DES systems are significantly more secure than single DES, but these are clearly a much slower process than encryption using single DES.

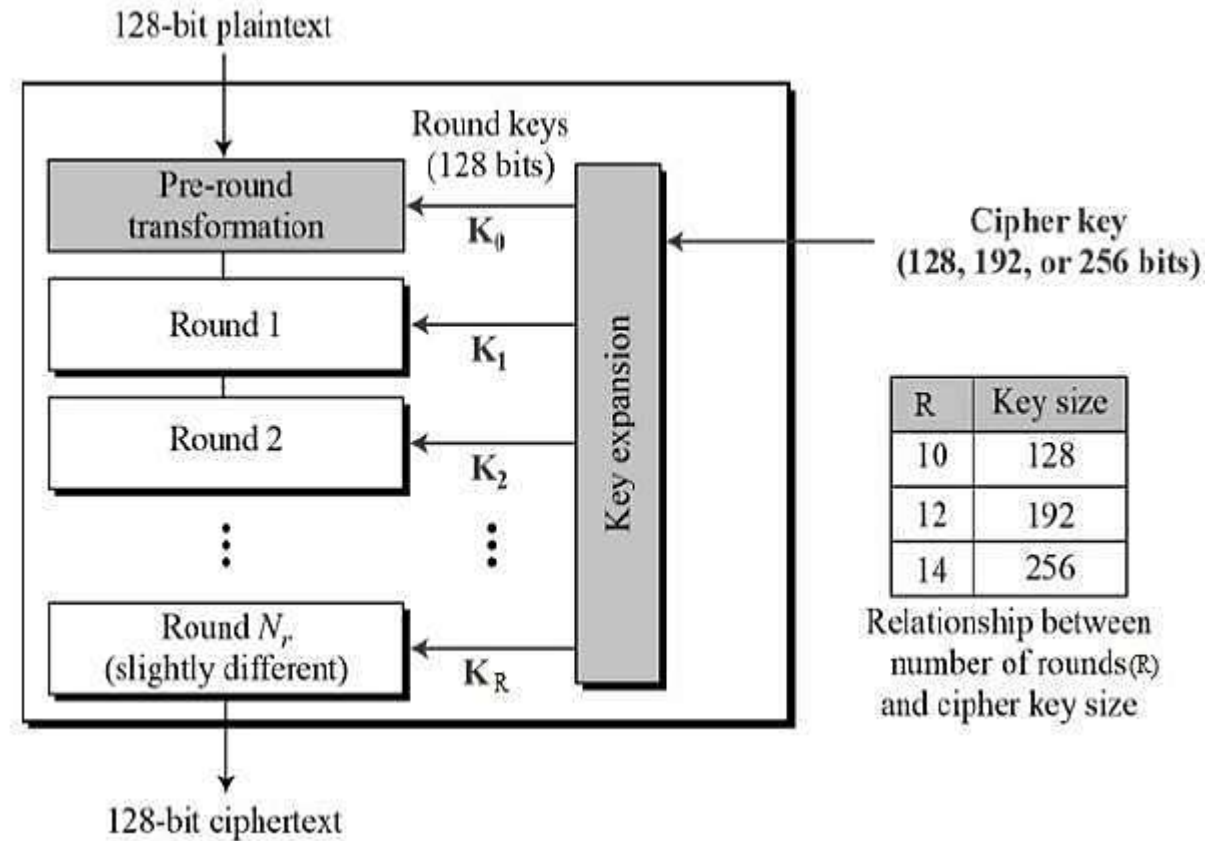
Advanced Encryption Standard (AES)

Introduction

- The more popular and widely adopted symmetric encryption algorithm likely to be encountered nowadays is the Advanced Encryption Standard (AES). It is found at least six times faster than triple DES.
- A replacement for DES was needed as its key size was too small. With increasing computing power, it was considered vulnerable against exhaustive key search attack. Triple DES was designed to overcome this drawback but it was found slow.
- The features of AES are as follows –
 - Symmetric key symmetric block cipher
 - 128-bit data, 128/192/256-bit keys
 - Stronger and faster than Triple-DES
 - Provide full specification and design details
 - Software implementable in C and Java

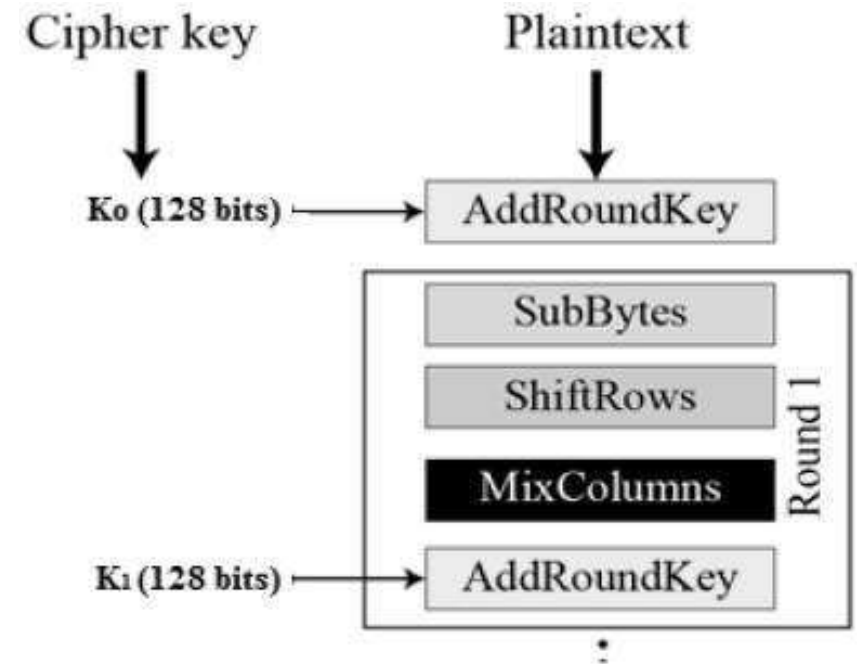
Operation of AES

- AES is an iterative rather than Feistel cipher.
 - It is based on 'substitution-permutation network'.
 - It comprises of a series of linked operations, some of which involve replacing inputs by specific outputs (substitutions) and others involve shuffling bits around (permutations).
- Interestingly, AES performs all its computations on bytes rather than bits.
 - Hence, AES treats the 128 bits of a plaintext block as 16 bytes.
 - These 16 bytes are arranged in four columns and four rows for processing as a matrix –
- Unlike DES, the number of rounds in AES is variable and depends on the length of the key.
 - AES uses 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys.
 - Each of these rounds uses a different 128-bit round key, which is calculated from the original AES key.



Encryption Process

- Here, we restrict to description of a typical round of AES encryption.
- Each round comprise of four sub-processes.
 - Byte Substitution (SubBytes)
 - The 16 input bytes are substituted by looking up a fixed table (S-box) given in design. The result is in a matrix of four rows and four columns.
 - ShiftRows
 - Each of the four rows of the matrix is shifted to the left. Any entries that 'fall off' are re-inserted on the right side of row.
 - MixColumns
 - Each column of four bytes is now transformed using a special mathematical function.
 - This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column.
 - The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.
 - AddRoundKey
 - Each column of four bytes is now transformed using a special mathematical function.
 - This function takes as input the four bytes of one column and outputs four completely new bytes, which replace the original column.
 - The result is another new matrix consisting of 16 new bytes. It should be noted that this step is not performed in the last round.



Decryption Process

- The process of decryption of an AES ciphertext is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order –
 - AddRound Key
 - MixColumns
 - ShiftRows
 - Byte Substitution (SubBytes)
- Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms need to be separately implemented, although they are very closely related.

AES Analysis

- In present day cryptography, AES is widely adopted and supported in both hardware and software. Till date, no practical cryptanalytic attacks against AES has been discovered.
 - Additionally, AES has built-in flexibility of key length, which allows a degree of 'future-proofing' against progress in the ability to perform exhaustive key searches.
- However, just as for DES, the AES security is assured only if it is correctly implemented and good key management is employed.

Evaluation Criteria for AES

Introduction

- AES has been subjected to more scrutiny than any other encryption algorithm over a longer period of time, and no effective cryptanalytic attack based on the algorithm rather than brute force has been found.
 - Accordingly, there is a high level of confidence that 3DES is very resistant to cryptanalysis.
 - If security were the only consideration, then 3DES would be an appropriate choice for a standardized encryption algorithm for decades to come.
- The principal drawback of 3DES is that the algorithm is relatively sluggish in software.
 - The original DES was designed for mid-1970s hardware implementation and does not produce efficient software code.
 - 3DES, which has three times as many rounds as DES, is correspondingly slower.
 - A secondary drawback is that both DES and 3DES use a 64-bit block size. For reasons of both efficiency and security, a larger block size is desirable.
- Because of these drawbacks, 3DES is not a reasonable candidate for long-term use. As a replacement, NIST in 1997 issued a call for proposals for a new Advanced Encryption Standard (AES), which should have security strength equal to or better than 3DES and significantly, improved efficiency.
 - In a first round of evaluation, 15 proposed algorithms were accepted.
 - A second round narrowed the field to 5 algorithms.
 - NIST completed its evaluation process and published a final standard (FIPS PUB 197) in November of 2001. NIST selected Rijndael as the proposed AES algorithm.
 - The two researchers who developed and submitted Rijndael for the AES are both cryptographers from Belgium: Dr. Joan Daemen and Dr. Vincent Rijmen.
- Ultimately, AES is intended to replace 3DES, but this process will take a number of years. NIST anticipates that 3DES will remain an approved algorithm (for U.S. government use) for the foreseeable future.

AES Evaluation

- The criteria span the range of concerns for the practical application of modern symmetric block ciphers.
- In fact, two set of criteria evolved. When NIST issued its original request for candidate algorithm nominations in 1997. The three categories of criteria were as follows:
 - Security
 - This refers to the effort required to cryptanalyze an algorithm.
 - The emphasis in the evaluation was on the practicality of the attack. Because the minimum key size for AES is 128 bits, brute-force attacks with current and projected technology were considered impractical.
 - Therefore, the emphasis, with respect to this point, is cryptanalysis other than a brute-force attack.
 - Cost
 - NIST intends AES to be practical in a wide range of applications. Accordingly, AES must have high computational efficiency, so as to be usable in high-speed applications, such as broadband links.
 - Algorithm and implementation characteristics
 - This category includes a variety of considerations, including flexibility; suitability for a variety of hardware and software implementations; and simplicity, which will make an analysis of security more straightforward.

Traffic Confidentiality

Introduction

- Mostly users are concerned about security from traffic analysis. Even in commercial applications, traffic analysis may yield information that the traffic generators would like to conceal. The following types of information that can be derived from a traffic analysis attack:-
 - Identities of partners
 - How frequently the partners are communicating
 - Message pattern, message length, or quantity of messages that suggest important information is being exchanged
 - The events that correlate with special conversations between particular partners
- A covert channel is a means of communication in a fashion unintended by the designers of the communications facility. Typically, the channel is used to transfer information in a way that violates a security policy.

Link Encryption Approach

- In link encryption, network-layer headers are encrypted, reducing the opportunity for traffic analysis.
- However, it is still possible in those circumstances for an attacker to assess the amount of traffic on a network and to observe the amount of traffic entering and leaving each end system.
- An effective countermeasure to this attack is traffic padding.

Traffic Padding Encryption Device

- Traffic padding produces cipher text output continuously, even in the absence of plaintext. A continuous random data stream is generated. When plaintext is available, it is encrypted and transmitted.
- When input plaintext is not present, random data are encrypted and transmitted. This makes it impossible for an attacker to distinguish between true data flow and padding and therefore impossible to deduce the amount of traffic.
- Traffic padding is essentially a link encryption function. If only end-to-end encryption is employed, then the measures available to the defender are more limited.
- For example, if encryption is implemented at the application layer, then an opponent can determine which transport entities are engaged in dialogue. If encryption techniques are housed at the transport layer, then network-layer addresses and traffic patterns remain accessible.

