Sarvasiddhant Education Society's
SWAMINARAYAN SIDDHANTA INSTITUTE OF TECHNOLOGY
Affiliated to Rashtrasant Tukdoji Maharaj Nagpur University
Nagpur Katol Highway Road, Khapri (Kothe),
Tal.Kalmeshwar, Nagpur, Maharashtra 441501
Department Of Computer Engineering
Session 2023-2024

CNS NOTES

CRYPTOGRAPHY & NETWORK SECURITY

SEMESTER: 7TH SEM (FINAL YEAR)

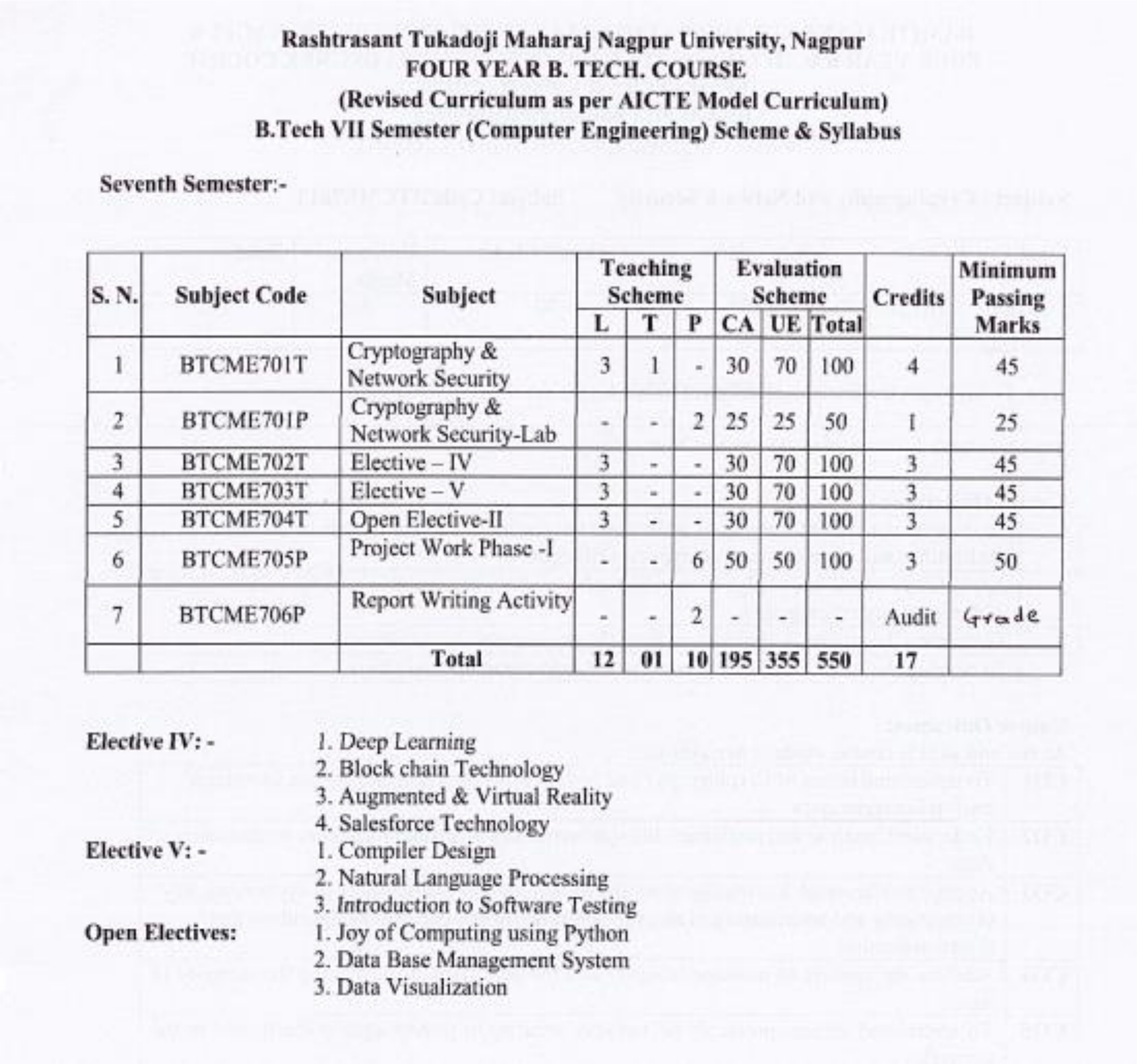
Subject Incharge: Prof. Ashvini Bais

Asst. Prof. (CE Dept.)

NOTES

CRYPTOGRAPHY & NETWORK SECURITY

Semester: 7th Sem (Final Year)



Rashtrasant Tukadoji Maharaj Nagpur University, Nagpur
FOUR YEAR B. TECH. COURSE
(Revised Curriculum as per AICTE Model Curriculum)
B.Tech VII Semester (Computer Engineering) Scheme & Syllabus

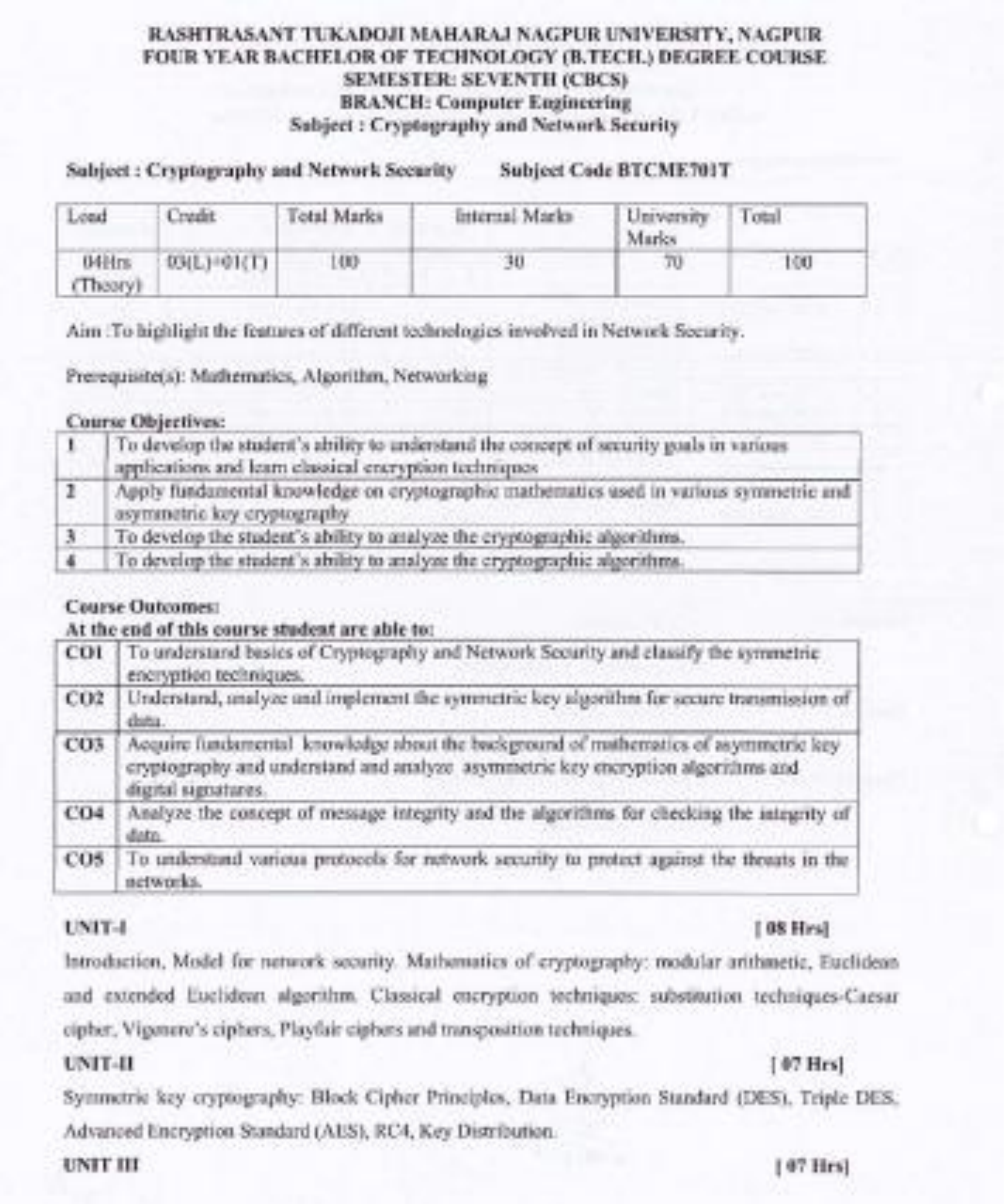
Seventh Semester:-

S. N.	Subject Code	Subject	Teaching Scheme			Evaluation Scheme			Credits	Minimum Passing Marks
			L	T	P	CA	UE	Total		
1	BTCME701T	Cryptography & Network Security	3	1	-	30	70	100	4	45
2	BTCME701P	Cryptography & Network Security-Lab	-	-	2	25	25	50	1	25
3	BTCME702T	Elective – IV	3	-	-	30	70	100	3	45
4	BTCME703T	Elective – V	3	-	-	30	70	100	3	45
5	BTCME704T	Open Elective-II	3	-	-	30	70	100	3	45
6	BTCME705P	Project Work Phase -I	-	-	6	50	50	100	3	50
7	BTCME706P	Report Writing Activity	-	-	2	-	-	-	Audit	Grade
		Total	**12**	**01**	**10**	**195**	**355**	**550**	**17**	

Elective IV: -
1. Deep Learning
2. Block chain Technology
3. Augmented & Virtual Reality
4. Salesforce Technology

Elective V: -
1. Compiler Design
2. Natural Language Processing
3. Introduction to Software Testing

Open Electives:
1. Joy of Computing using Python
2. Data Base Management System
3. Data Visualization



RASHTRASANT TUKADOJI MAHARAJ NAGPUR UNIVERSITY, NAGPUR
FOUR YEAR BACHELOR OF TECHNOLOGY (B.TECH.) DEGREE COURSE
SEMESTER: SEVENTH (CBCS)
BRANCH: Computer Engineering
Subject : Cryptography and Network Security

Subject : Cryptography and Network Security   Subject Code BTCME701T

Load	Credit	Total Marks	Internal Marks	University Marks	Total
04Hrs (Theory)	03(L)+01(T)	100	30	70	100

Aim :To highlight the features of different technologies involved in Network Security.

Prerequisite(s): Mathematics, Algorithm, Networking

Course Objectives:

1	To develop the student's ability to understand the concept of security goals in various applications and learn classical encryption techniques
2	Apply fundamental knowledge on cryptographic mathematics used in various symmetric and asymmetric key cryptography
3	To develop the student's ability to analyze the cryptographic algorithms.
4	To develop the student's ability to analyze the cryptographic algorithms.

Course Outcomes:
At the end of this course student are able to:

CO1	To understand basics of Cryptography and Network Security and classify the symmetric encryption techniques.
CO2	Understand, analyze and implement the symmetric key algorithm for secure transmision of data.
CO3	Acquire fundamental knowledge about the background of mathematics of asymmetric key cryptography and understand and analyze asymmetric key encryption algorithms and digital signatures.
CO4	Analyze the concept of message integrity and the algorithms for checking the integrity of data.
CO5	To understand various protocols for network security to protect against the threats in the networks.

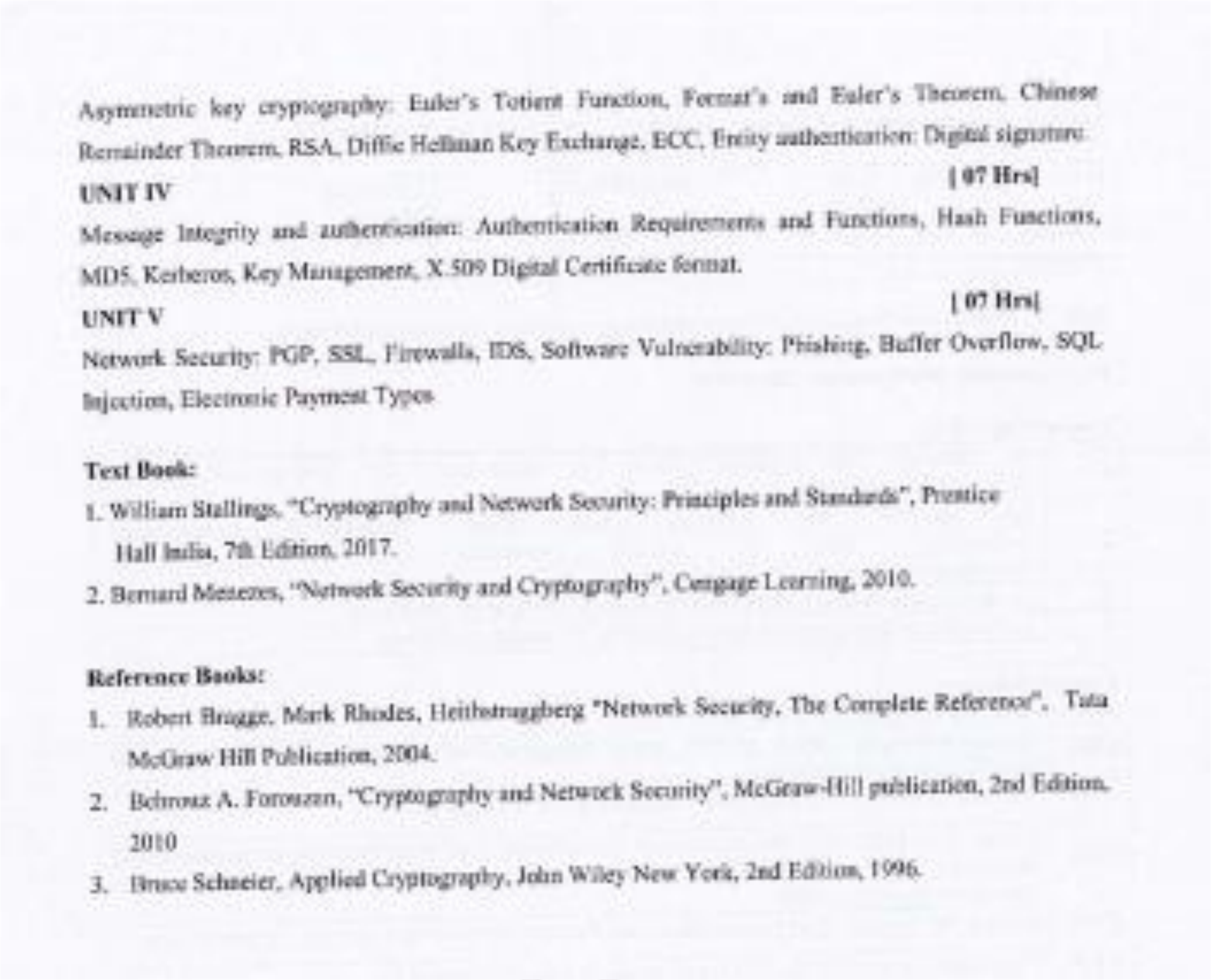
UNIT-I **[08 Hrs]**

Introduction, Model for network security. Mathematics of cryptography: modular arithmetic, Euclidean and extended Euclidean algorithm. Classical encryption techniques: substitution techniques-Caesar cipher, Vigenere's ciphers, Playfair ciphers and transposition techniques.

UNIT-II **[07 Hrs]**

Symmetric key cryptography: Block Cipher Principles, Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), RC4, Key Distribution.

UNIT III **[07 Hrs]**



Asymmetric key cryptography: Euler's Totient Function, Fermat's and Euler's Theorem, Chinese Remainder Theorem, RSA, Diffie Hellman Key Exchange, ECC, Entity authentication: Digital signature

[07 Hrs]

UNIT IV

Message Integrity and authentication: Authentication Requirements and Functions, Hash Functions, MD5, Kerberos, Key Management, X.509 Digital Certificate format.

[07 Hrs]

UNIT V

Network Security: PGP, SSL, Firewalls, IDS, Software Vulnerability: Phishing, Buffer Overflow, SQL Injection, Electronic Payment Types

Text Book:

1. William Stallings, "Cryptography and Network Security: Principles and Standards", Prentice Hall India, 7th Edition, 2017.
2. Bernard Menezes, "Network Security and Cryptography", Cengage Learning, 2010.

Reference Books:

1. Robert Bragge, Mark Rhodes, Heithstraggberg "Network Security, The Complete Reference", Tata McGraw Hill Publication, 2004.
2. Behrouz A. Forouzan, "Cryptography and Network Security", McGraw-Hill publication, 2nd Edition, 2010
3. Bruce Schneier, Applied Cryptography, John Wiley New York, 2nd Edition, 1996.

UNIT II

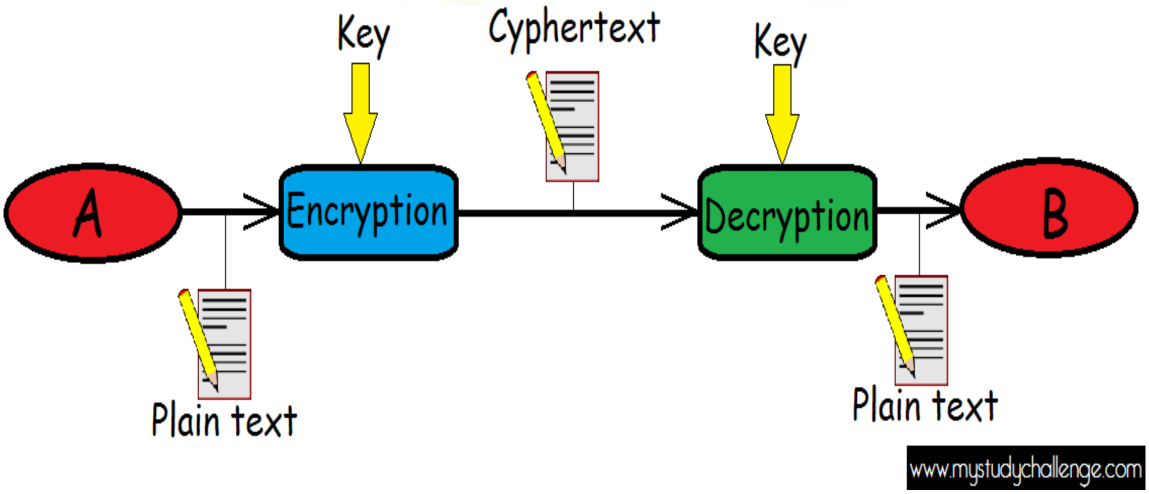
Symmetric Key Cryptography: Block Cipher Principles, Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), RC4, Key Distribution.

Symmetric Key Cryptography

Cryptography-

Cryptographic systems are characterized along three independent dimensions:

1. The type of operations used for transforming plaintext to ciphertext. All encryption algorithms are based on two general principles: substitution, in which each element in the plaintext (bit, letter, group of bits or letters) is mapped into another element, and transposition, in which elements in the plaintext are rearranged. The fundamental requirement is that no information be lost (that is, that all operations are reversible). Most systems, referred to as product systems, involve multiple stages of substitutions and transpositions.
2. The number of keys used. If both sender and receiver use the same key, the system is referred to as symmetric, single-key, secret-key, or conventional encryption. If the sender and receiver use different keys, the system is referred to as asymmetric, two-key, or public-key encryption.
3. The way in which the plaintext is processed. A block cipher processes the input one block of elements at a time, producing an output block for each input block. A stream cipher processes the input elements continuously, producing output one element at a time, as it goes along.



Symmetric Key Algorithms:

Symmetric and public key algorithms Encryption/Decryption methods fall into two categories.

Symmetric Key :

In symmetric key algorithms, the encryption and decryption keys are known both to sender and receiver. The encryption key is shared and the decryption key is easily calculated from it. In many cases, the encryption and decryption keys are the same.

Public Key :

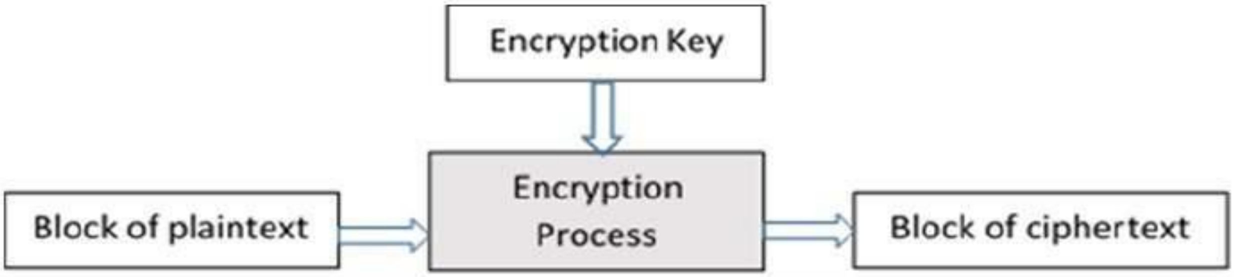
In public key cryptography, encryption key is made public, but it is computationally infeasible to find the decryption key without the information known to the receiver.

Block Cipher Principles-

In cryptography, a block cipher is a deterministic algorithm operating on fixed-length groups of bits, called a block, with an unvarying transformation that is specified by a symmetric key. Block ciphers operate as important elementary components in the design of many cryptographic protocols, and are widely used to implement encryption of bulk data.

A block cipher consists of two paired algorithms, one for encryption, E, and the other for decryption, D. Both algorithms accept two inputs: an input block of size n bits and a key of size k bits; and both yield an n-bit output block. The decryption algorithm D is defined to be the inverse function of encryption, i.e., $D = E^{-1}$. More formally, a block cipher is specified by an encryption function.

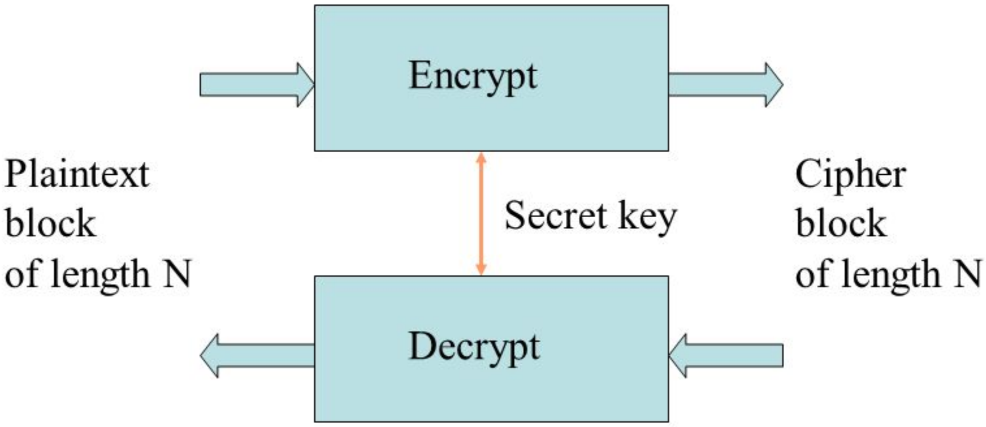
The basic scheme of a block cipher is depicted as follows –



A block cipher takes a block of plaintext bits and generates a block of ciphertext bits, generally of same size. The size of block is fixed in the given scheme. The choice of block size does not directly affect to the strength of encryption scheme. The strength of cipher depends up on the key length which takes as input a key K of bit length k, called the key size, and a bit string P of length n, called the block size, and returns a string C of n bits. P is called the plaintext, and C is termed the ciphertext. For each K, the function $E_K(P)$ is required to be an invertible mapping on $\{0,1\}^n$. The inverse for E is defined as a function taking a key K and a ciphertext C to return a plaintext value P, such that For example, a block cipher encryption algorithm might take a 128-bit block of plaintext as input, and output a corresponding 128-bit block of ciphertext. The exact transformation is controlled using a second input – the secret key. Decryption is similar: the decryption algorithm takes, in this example, a 128-bit block of ciphertext together with the secret key, and yields the original 128-bit block of plain text. For each key K, E_K is a permutation (a bijective mapping) over the set of input blocks. Each key selects one permutation from the set of possible permutations.

Block Cipher Schemes:-

There is a vast number of block ciphers schemes that are in use. Many of them are publically known. Most popular and prominent block ciphers are listed below.



- **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 ciphers 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.
- **IDEA** − It is a sufficiently strong block cipher with a block size of 64 and a key size of 128 bits. A number of applications use IDEA encryption, including early versions of

Pretty Good Privacy (PGP) protocol. The use of IDEA scheme has a restricted adoption due to patent issues.

- **Twofish** − This scheme of block cipher uses block size of 128 bits and a key of variable length. It was one of the AES finalists. It is based on the earlier block cipher Blowfish with a block size of 64 bits.
- **Serpent** − A block cipher with a block size of 128 bits and key lengths of 128, 192, or 256 bits, which was also an AES competition finalist. It is a slower but has more secure design than other block cipher.

Block ciphers are built in the Feistel cipher structure. Block cipher has a specific number of rounds and keys for generating ciphertext. Block cipher is a type of encryption algorithm that processes fixed-size blocks of data, usually 64 or 128 bits, to produce ciphertext. The design of a block cipher involves several important principles to ensure the security and efficiency of the algorithm. Some of these principles are:

1. **Number of Rounds:** The number of Rounds is regularly considered in design criteria, it just reflects the number of rounds to be suitable for an algorithm to make it more complex, in DES we have 16 rounds ensuring it to be more secure while in AES we have 10 rounds which makes it more secure.

2. **Design of function F:** The core part of the Feistel Block cipher structure is the Round Function. The complexity of cryptanalysis can be derived from the Round function i.e. the increasing level of complexity for the round function would be greatly contributing to an increase in complexity. To increase the complexity of the round function, the avalanche effect is also included in the round function, as the change of a single bit in plain text would produce a mischievous output due to the presence of avalanche effect.

3. **Confusion and Diffusion:** The cipher should provide confusion and diffusion to make it difficult for an attacker to determine the relationship between the plaintext and ciphertext. Confusion means that the ciphertext should be a complex function of the key and plaintext, making it difficult to guess the key. Diffusion means that a small change in the plaintext should cause a significant change in the ciphertext, which makes it difficult to analyze the encryption pattern.

4. **Key Size:** The key size should be large enough to prevent brute-force attacks. A larger key size means that there are more possible keys, making it harder for an attacker to guess the correct one. A key size of 128 bits is considered to be secure for most applications.

5. **Key Schedule:** The key schedule should be designed carefully to ensure that the keys used for encryption are independent and unpredictable. The key schedule should also resist attacks that exploit weak keys or key-dependent properties of the cipher.

6. **Block Size:** The block size should be large enough to prevent attacks that exploit statistical patterns in the plaintext. A block size of 128 bits is generally considered to be secure for most applications.

7. **Non-linearity:** The S-box used in the cipher should be non-linear to provide confusion. A linear S-box is vulnerable to attacks that exploit the linear properties of the cipher.

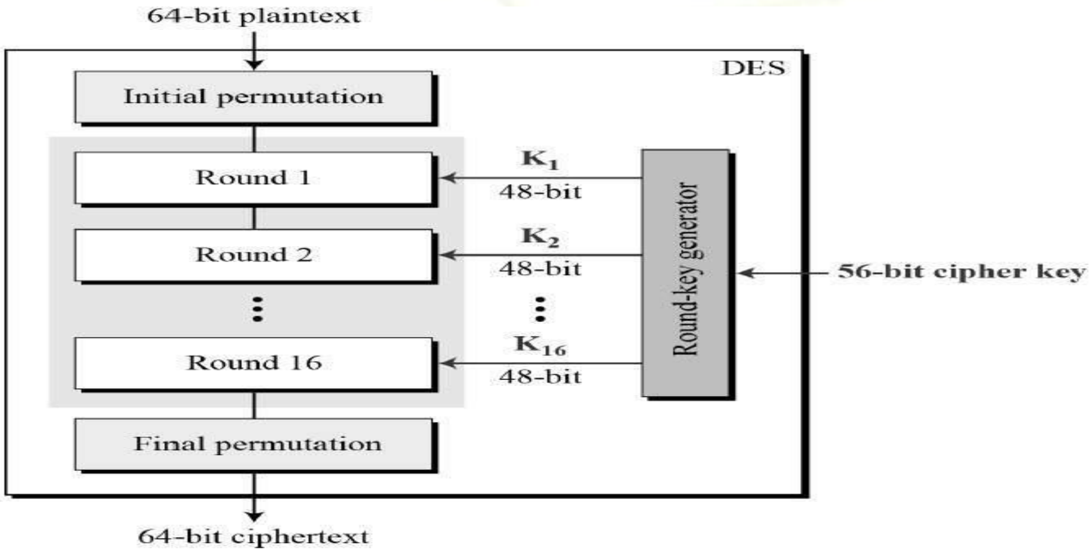
8. **Avalanche Effect:** The cipher should exhibit the avalanche effect, which means that a small change in the plaintext or key should cause a significant change in the ciphertext. This ensures that any change in the input results in a complete change in the output.

9. **Security Analysis:** The cipher should be analyzed for its security against various attacks such as differential cryptanalysis, linear cryptanalysis, and brute-force attacks. The cipher should also be tested for its resistance to implementation attacks, such as side-channel attacks.

Overall, a good block cipher design should be resistant to various attacks, efficient, and easy to implement.

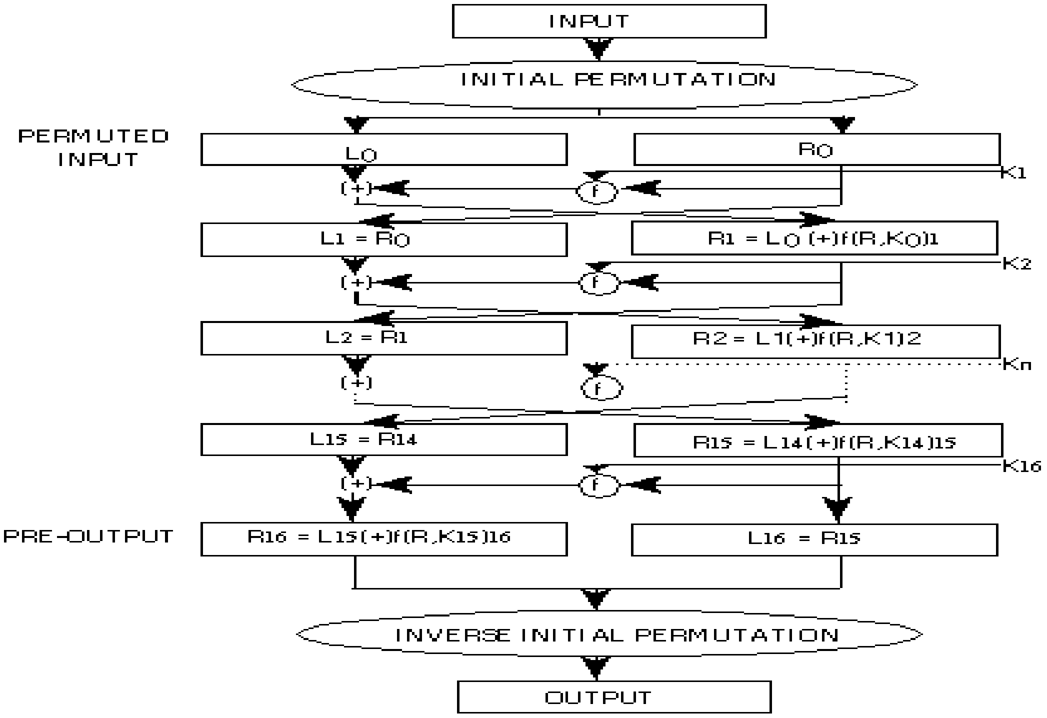
Data Encryption Standard (DES)-

The Data Encryption Standard (DES) is an outdated symmetric-key method of data encryption. DES works by using the same key to encrypt and decrypt a message, so both the sender and the receiver must know and use the same private key. Once the go-to, symmetric-key algorithm for the encryption of electronic data, DES has been superseded by the more secure Advanced Encryption Standard (AES) algorithm. The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST). DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only). General Structure of DES is depicted in the following illustration −



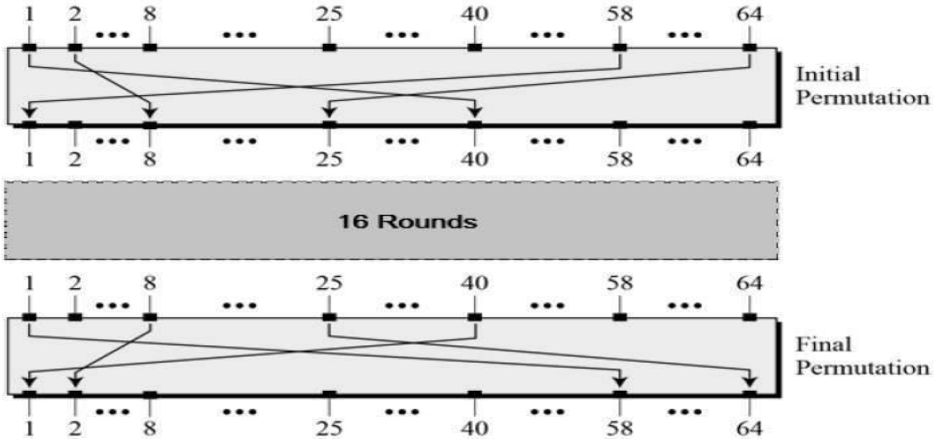
Since DES is based on the Feistel Cipher, all that is required to specify DES is −

- Round function
- Key schedule
- Any additional processing − Initial and final permutation



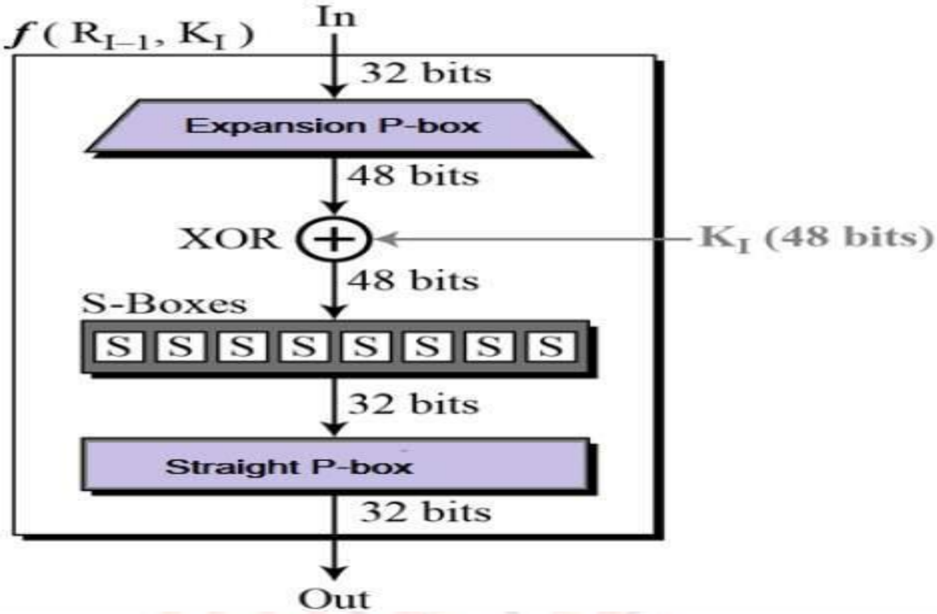
Initial and Final Permutations

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

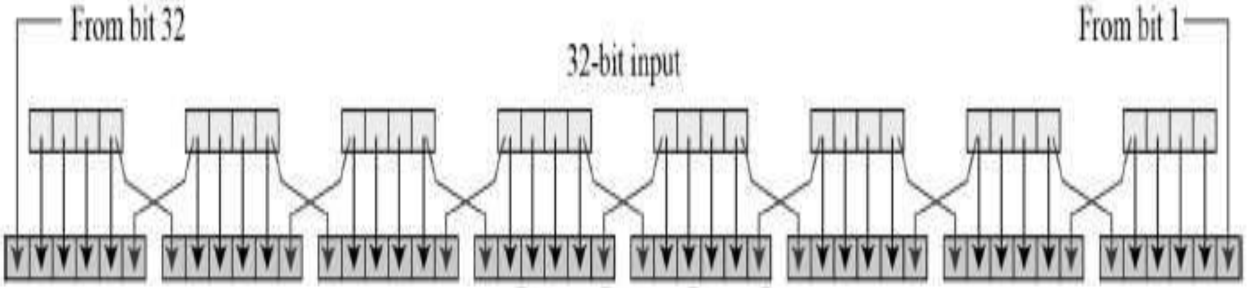


Round Function

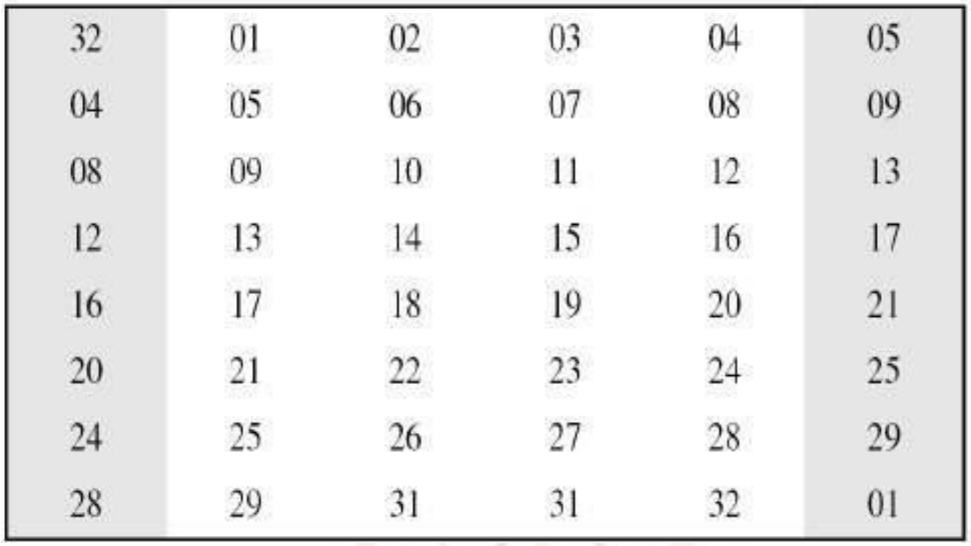
The heart of this cipher is the DES function, *f*. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



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 –

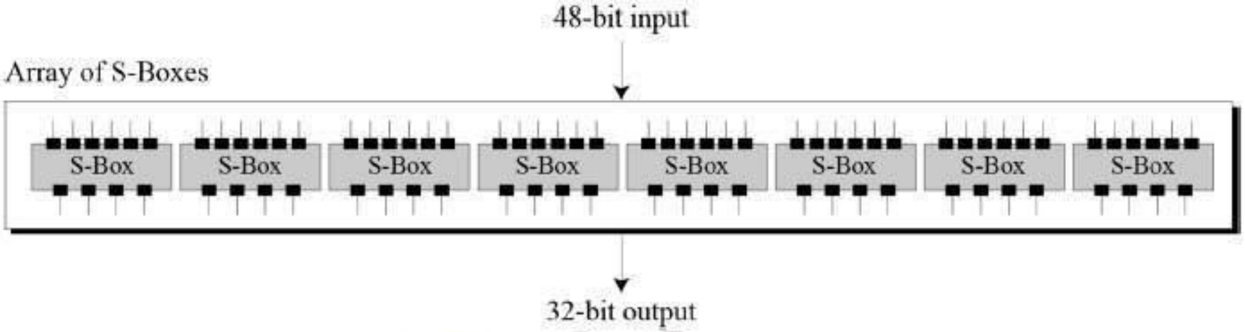


The graphically depicted permutation logic is generally described as table in DES specification illustrated as shown –

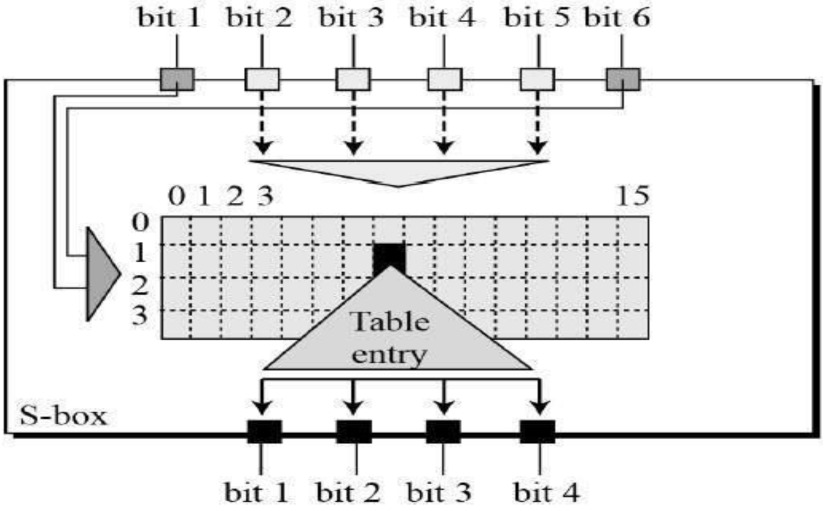


32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

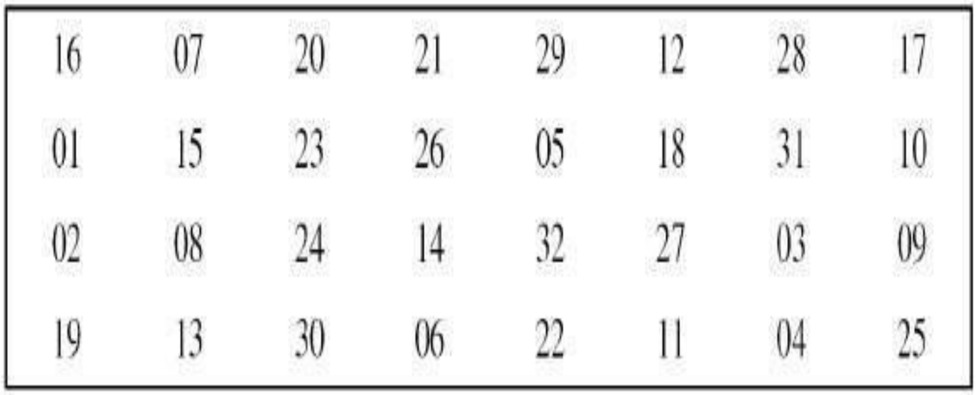
- **XOR (Whitener).** − After the expansion permutation, DES does XOR operation on the expanded right section and the round key. The round key is used only in this operation.
- **Substitution Boxes.** − The S-boxes carry out the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output. Refer the following illustration –



The S-box rule is illustrated below –



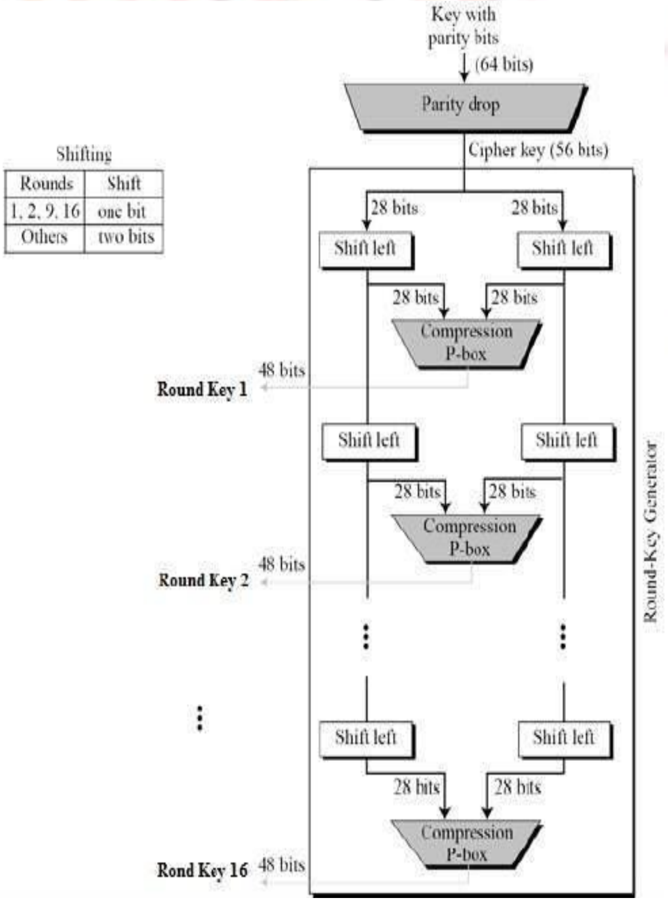
- There are a total of eight S-box tables. The output of all eight s-boxes is then combined in to 32 bit section.
- **Straight Permutation** − The 32 bit output of S-boxes is then subjected to the straight permutation with rule shown in the following illustration:



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

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 –



The logic for Parity drop, shifting, and Compression P-box is given in the DES description.

DES Analysis

The DES satisfies both the desired properties of block cipher. These two properties make cipher very strong.

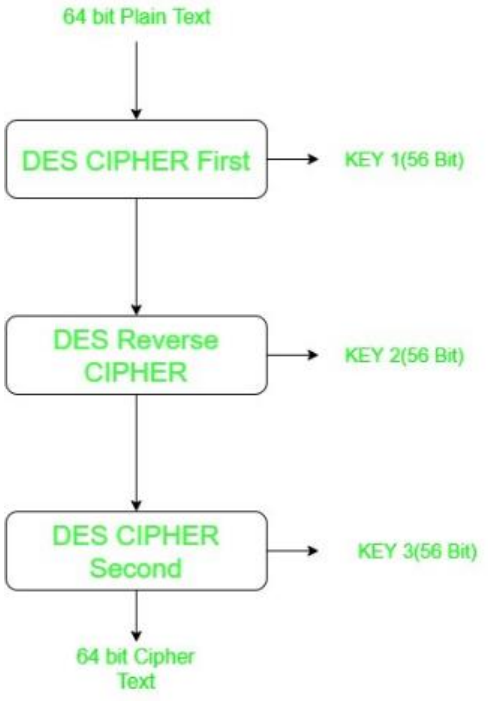
- **Avalanche effect −** A small change in plaintext results in the very great change in the cipher text.

- **Completeness −** Each bit of cipher text depends on many bits of plaintext.

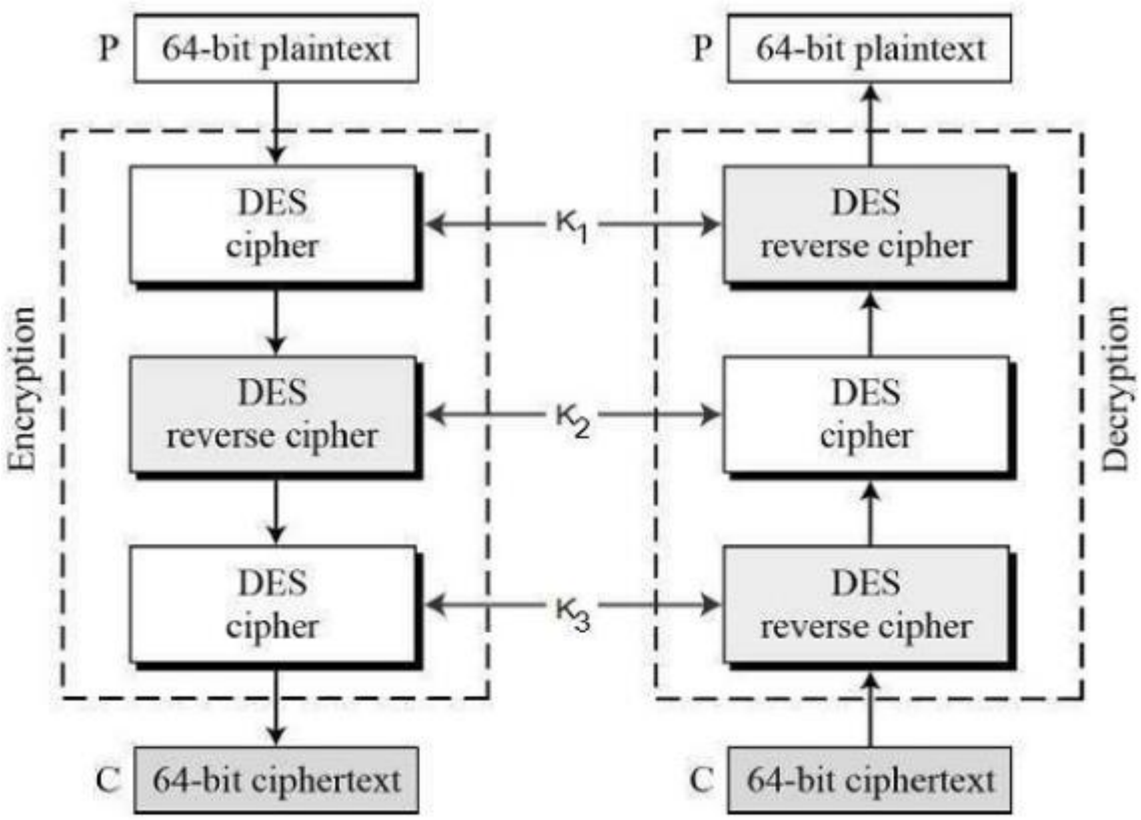
During the last few years, cryptanalysis have found some weaknesses in DES when key selected are weak keys. These keys shall be avoided. DES has proved to be a very well designed block cipher. There have been no significant cryptanalytic attacks on DES other than exhaustive key search.

Triple DES-

Triple DES is a encryption technique which uses three instance of DES on same plain text. It uses there different types of key choosing technique in first all used keys are different and in second two keys are same and one is different and in third all keys are same. Triple DES is also vulnerable to meet-in-the middle attack because of which it give total security level of 2^112 instead of using 168 bit of key. The block collision attack can also be done because of short block size and using same key to encrypt large size of text. It is also vulnerable to sweet32 attack.



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 3×56 = 168 bits. The encryption scheme is illustrated as follows –



The encryption-decryption process is as follows −

- 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. 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. The Triple DES scheme that uses three different keys offers a 100-bit security level which is considered acceptable until the year 2030.

Key Features of 3DES-

- Block Cipher Encryption: 3DES is a block cipher encryption algorithm that operates on 64-bit blocks of plaintext at a time.

- Symmetric Key Encryption: 3DES uses a symmetric key encryption system, meaning that the same key is used for both encryption and decryption.

- Triple Layer Encryption: 3DES uses three different keys to encrypt the plaintext three times, hence the name Triple DES.

- Variable Key Size: 3DES supports variable key sizes, ranging from 128 to 192 bits, offering enhanced security compared to DES.

Encryption Process-

The encryption process of 3DES involves the following steps:

1. Key Generation: Three unique keys are generated using a key derivation algorithm.

2. Initial Permutation: The 64-bit plaintext is subjected to an initial permutation.

3. Three Rounds of Encryption: The plaintext is encrypted three times, each time using a different key, to create three layers of encryption.

4. Final Permutation: After the three rounds of encryption, a final permutation is applied to the output to produce the ciphertext.

Decryption Process-

The decryption process of 3DES is simply the reverse of the encryption process, with the ciphertext being fed into the algorithm and the steps being performed in reverse order, using the three keys in reverse order.

Advantages of 3DES-

- Enhanced Security: The triple-layered encryption technique of 3DES provides enhanced security compared to DES.

- Widely Used: 3DES is a widely used encryption algorithm, and is included in many encryption standards and protocols.

- Compatible: 3DES is backward compatible with DES, which means that it can be used in legacy systems that still use DES.

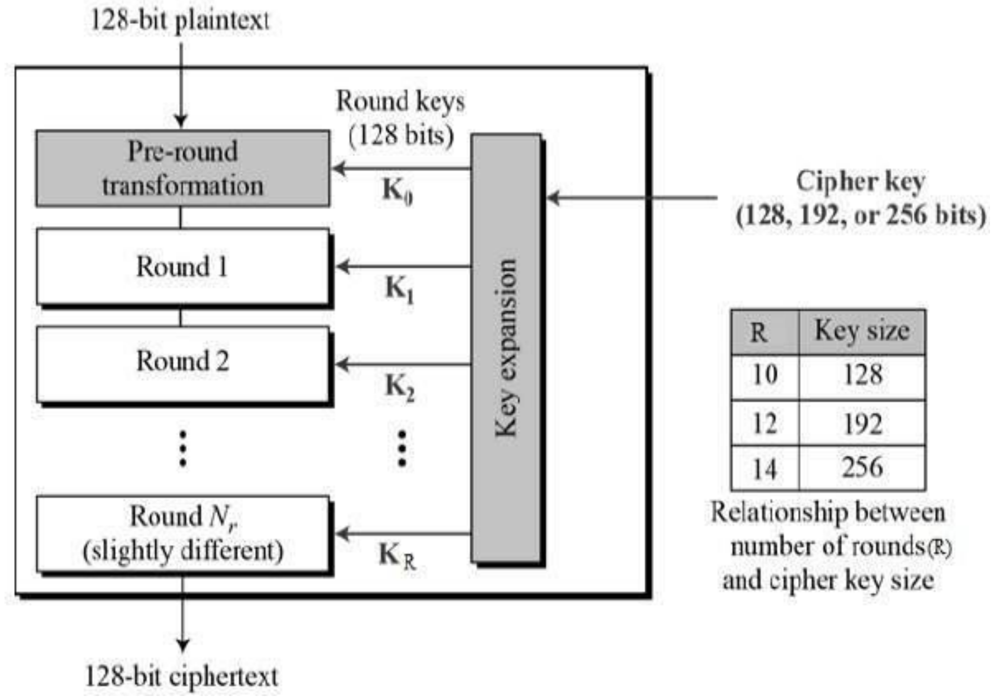
- Customizable Key Sizes: 3DES supports variable key sizes, which makes it more adaptable to different security needs.

Applications of 3DES-

- Financial Transactions: 3DES is used to secure financial transactions, such as online banking, credit card processing, and electronic fund transfers.
- VPNs: 3DES is used to secure virtual private networks (VPNs) to provide secure communication between remote locations.
- Healthcare Systems: 3DES is used to secure patient information in healthcare systems, such as electronic health records and medical imaging systems.
- Government Communications: 3DES is used to secure government communications, such as military communications and secure data transfers.

Advanced Encryption Standard (AES) –

The Advanced Encryption Standard, or AES, is a symmetric block cipher chosen by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data.



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

AES features:

- The selection process for this new symmetric key algorithm was fully open to public scrutiny and comment; this ensured a thorough, transparent analysis of the designs submitted.
- NIST specified the new advanced encryption standard algorithm must be a block cipher capable of handling 128 bit blocks, using keys sized at 128, 192, and 256 bits; other criteria for being chosen as the next advanced encryption standard algorithm included:
  - Security: Competing algorithms were to be judged on their ability to resist attack, as compared to other submitted ciphers, though security strength was to be considered the most important factor in the competition.
  - Cost: Intended to be released under a global, nonexclusive and royalty-free basis, the candidate algorithms were to be evaluated on computational and memory efficiency.
  - Implementation: Algorithm and implementation characteristics to be evaluated included the flexibility of the algorithm; suitability of the algorithm to be implemented in hardware or software; and overall, relative simplicity of implementation.

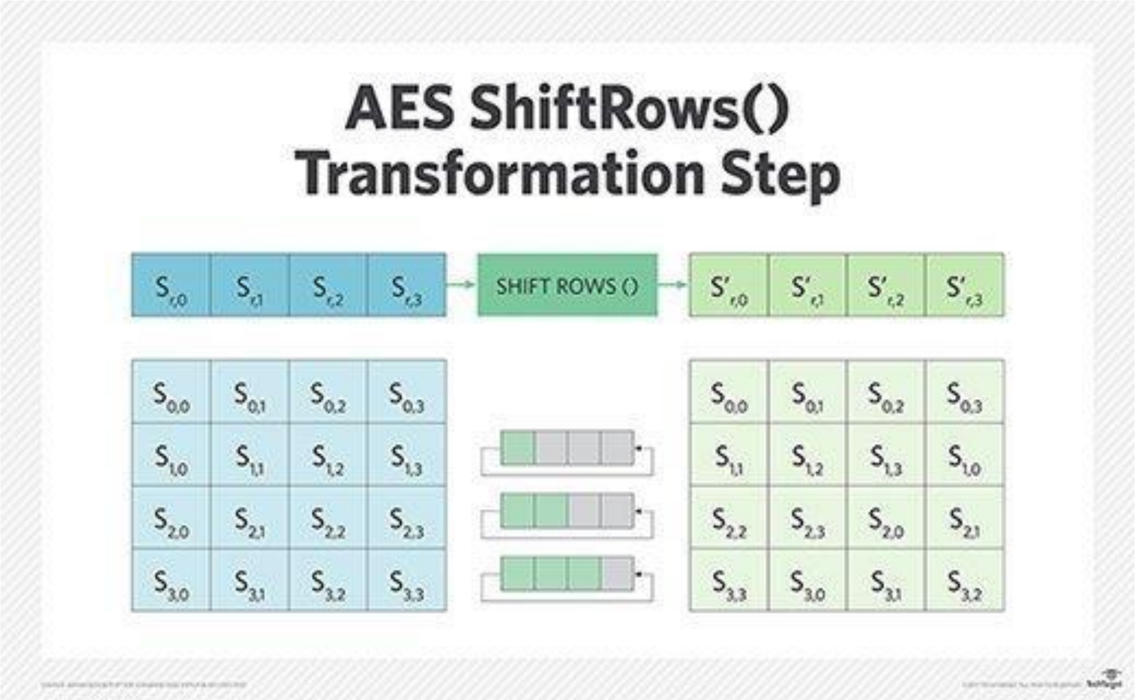
How AES encryption works:

AES comprises three block ciphers: AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128-, 192- and 256-bits, respectively. The Rijndael cipher was designed to accept additional block sizes and key lengths, but for AES, those functions were not adopted.

Symmetric (also known as secret-key) ciphers use the same key for encrypting and decrypting, so the sender and the receiver must both know -- and use -- the same secret key. All key lengths are deemed sufficient to protect classified information up to the "Secret" level with "Top Secret" information requiring either 192- or 256-bit key lengths. There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys -- a round consists of several processing steps that include substitution, transposition and mixing of the input plaintext and transform it into the final output of cipher text.

The AES encryption algorithm defines a number of transformations that are to be performed on data stored in an array. The first step of the cipher is to put the data into an array; after which the cipher transformations are repeated over a number of encryption rounds. The number of rounds is determined by the key length, with 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys.

The first transformation in the AES encryption cipher is substitution of data using a substitution table; the second transformation shifts data rows, the third mixes columns. The last transformation is a simple exclusive or (XOR) operation performed on each column using a different part of the encryption key -- longer keys need more rounds to complete. AES encryption transforms array data by shuffling rows and columns, and substitutions based on the encryption key.



$S_{r,0}$	$S_{r,1}$	$S_{r,2}$	$S_{r,3}$		$S'_{r,0}$	$S'_{r,1}$	$S'_{r,2}$	$S'_{r,3}$

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$	$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,0}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$	$S_{2,2}$	$S_{2,3}$	$S_{2,0}$	$S_{2,1}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$	$S_{3,3}$	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$

Attacks on AES encryption:

Research into attacks on AES encryption has continued since the standard was finalized in 2000. Various researchers have published attacks against reduced-round versions of the Advanced Encryption Standard.

In 2005, cryptographer Daniel J. Bernstein published a paper, "Cache-timing attacks on AES," in which he demonstrated a timing attack on AES capable of achieving a "complete AES key recovery from known-plaintext timings of a network server on another computer."

A research paper published in 2011, titled "Biclique Cryptanalysis of the Full AES," by researchers Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger, demonstrated that by using a technique called a biclique attack, they could recover AES keys faster than a brute-force attack by a factor of between three and five, depending on the cipher version. However, even this attack does not threaten the practical use of AES due to its high-computational complexity.

AES has proven to be a reliable cipher, and the only practical successful attacks against AES have leveraged side-channel attacks on weaknesses found in the implementation or key management of specific AES-based encryption products.

Side-channel attacks exploit flaws in the way a cipher has been implemented rather than brute force or theoretical weaknesses in a cipher. The Browser Exploit Against SSL/TLS (BEAST) browser exploit against the TLS v1.0 protocol is a good example; TLS can use AES to encrypt data, but due to the information that TLS exposes, attackers managed to predict the initialization vector block used at the start of the encryption process.

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 time 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.

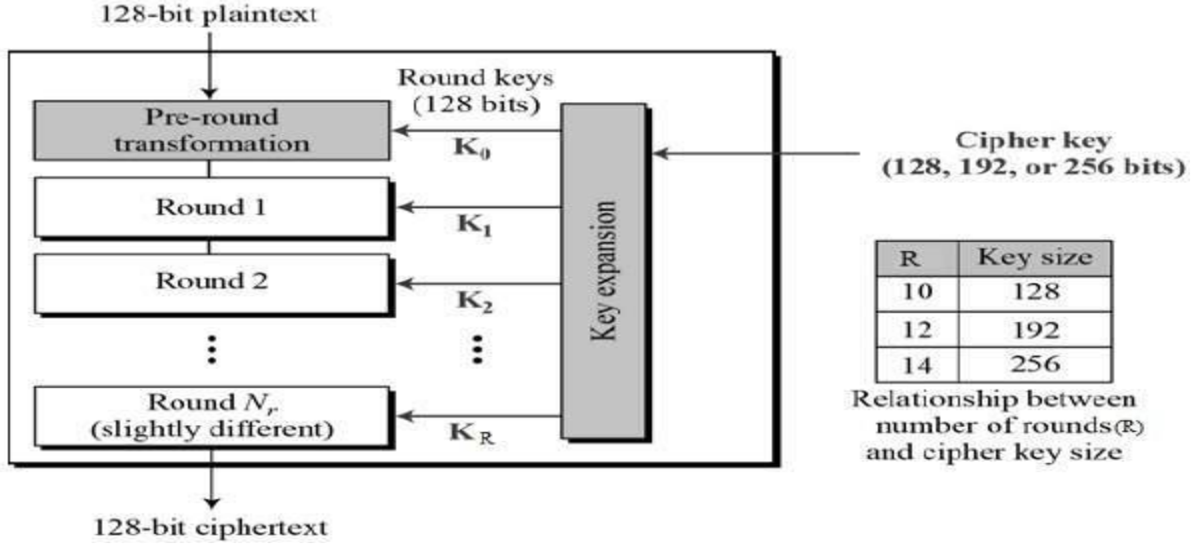
The features of AES are as follows −

- 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

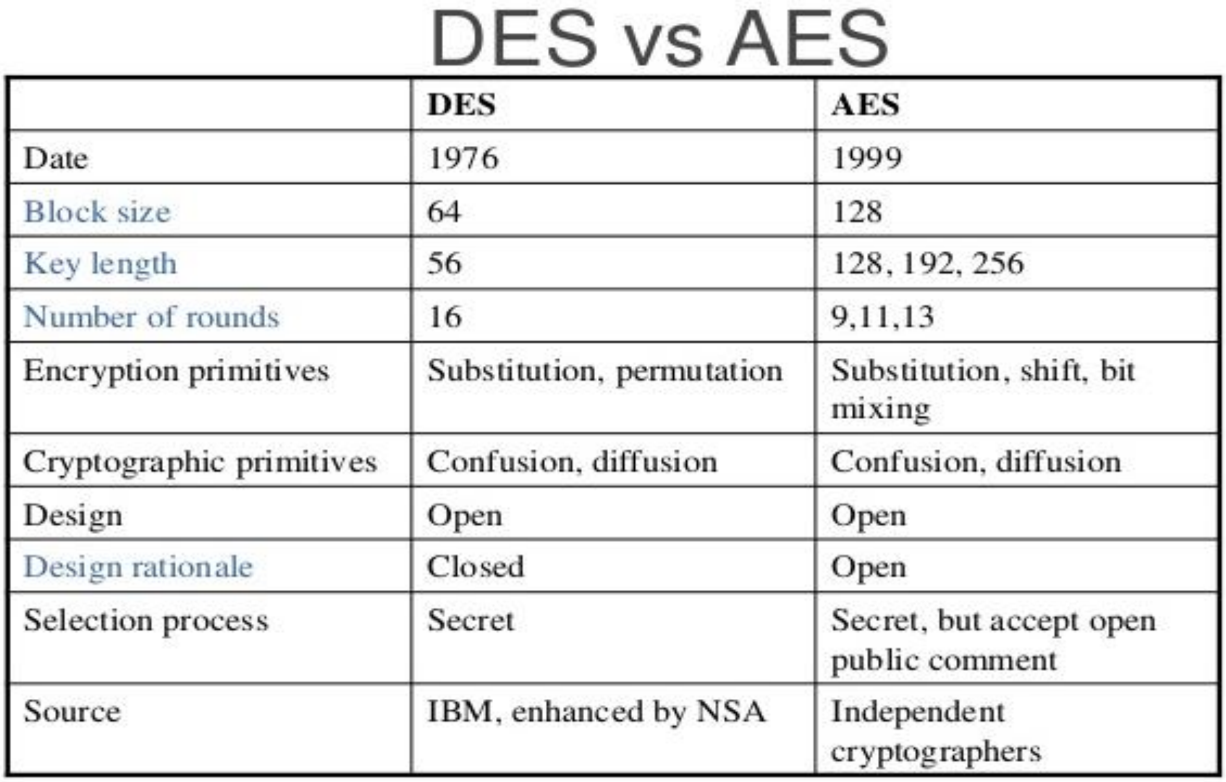
Operation of AES:

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. The schematic of AES structure is given in the following illustration –



R	Key size
10	128
12	192
14	256



DES vs AES

	DES	AES
Date	1976	1999
Block size	64	128
Key length	56	128, 192, 256
Number of rounds	16	9,11,13
Encryption primitives	Substitution, permutation	Substitution, shift, bit mixing
Cryptographic primitives	Confusion, diffusion	Confusion, diffusion
Design	Open	Open
Design rationale	Closed	Open
Selection process	Secret	Secret, but accept open public comment
Source	IBM, enhanced by NSA	Independent cryptographers

Rivest Cipher 4 (RC4) –

RC4 (also known as Rivest Cipher 4) is a form of stream cipher. It encrypts messages one byte at a time via an algorithm. Plenty of stream ciphers exist, but RC4 is among the most popular. It's simple to apply, and it works quickly, even on very large pieces of data. Rivest Cipher 4, or RC4, is a stream cipher created in 1987. A stream cipher is a type of cipher that operates on data a byte at a time to encrypt that data. RC4 is one of the most commonly used stream ciphers, having been used in Secure Socket Layer (SSL)/ Transport Layer Security (TLS) protocols, IEEE 802.11 wireless LAN standard, and the Wi-Fi Security Protocol WEP (Wireless Equivalent Protocol). RC4 owes its popularity, relating to stream ciphers, to its ease of use and performance speed. Now, significant flaws mean RC4 is not used nearly as often as before.

RC4 is a stream cipher and variable-length key algorithm. This algorithm encrypts one byte at a time (or larger units at a time). A key input is a pseudorandom bit generator that produces a stream 8-bit number that is unpredictable without knowledge of input key, The output of the generator is called key-stream, is combined one byte at a time with the plaintext stream cipher using X-OR operation.

Example:
RC4 Encryption

10011000 ? 01010000 = 11001000

RC4 Decryption

11001000 ? 01010000 = 10011000

Key-Generation Algorithm – A variable-length key from 1 to 256 bytes is used to initialize a 256-byte state vector S, with elements S[0] to S[255]. For encryption and decryption, a byte k is generated from S by selecting one of the 255 entries in a systematic fashion, then the entries in S are permuted again.

Key-Scheduling Algorithm: Initialization: The entries of S are set equal to the values from 0 to 255 in ascending order, a temporary vector T, is created. If the length of the key k is 256 bytes, then k is assigned to T. Otherwise, for a key with length(k-len) bytes, the first k-len elements of T as copied from K, and then K is repeated as many times as necessary to fill T.

Features of the RC4 encryption algorithm:

1. Symmetric key algorithm: RC4 is a symmetric key encryption algorithm, which means that the same key is used for encryption and decryption.
2. Stream cipher algorithm: RC4 is a stream cipher algorithm, which means that it encrypts and decrypts data one byte at a time. It generates a key stream of pseudorandom bits that are XORed with the plaintext to produce the ciphertext.
3. Variable key size: RC4 supports variable key sizes, from 40 bits to 2048 bits, making it flexible for different security requirements.
4. Fast and efficient: RC4 is a fast and efficient encryption algorithm that is suitable for low-power devices and applications that require high-speed data transmission.
5. Widely used: RC4 has been widely used in various applications, including wireless networks, secure sockets layer (SSL), virtual private networks (VPN), and file encryption.
6. Vulnerabilities: RC4 has several vulnerabilities, including a bias in the first few bytes of the keystream, which can be exploited to recover the key. As a result, RC4 is no longer recommended for use in new applications.

How secure is RC4?

RC4 was initially used in many applications, like SSL/TLS and WEP, until severe vulnerabilities were found in RC4 in 2003 and 2013. As RC4 was used in WEP, attackers had a chance to practice cracking it as often as they wished. With this practice, a flaw was found in RC4 where the encryption key used by RC4 could be cracked in less than a minute. RC4 keys can come in sizes of 64 or 128-bits, and the 128-bit key is able to be obtained in seconds. At the time, WEP was the only security protocol used for Wi-Fi, so the next phase, Wi-Fi Protected Access (WPA), had to be rushed for use.

Another vulnerability was discovered in RC4 in 2013 while it was being used as a workaround for a cipher block chaining issue that was discovered in 2011. Cipher block chaining is an operational mode used by block ciphers, which RC4 did not use. A group of security researchers found a way around RC4, with only a slight increase in processing power necessary in the previous RC4 attack. Due to these vulnerabilities, and other smaller ones found later, RC4 is no longer a cipher that is recommended to be used.

Working of RC4-

RC4 creates a pseudo-random bit stream (a keystream). These, like any other stream cipher, can be used for encryption by utilizing bit-wise exclusive or to combine it with the plaintext. The same procedure is used for decryption (since exclusive-OR is a symmetric operation).

The cipher uses a secret internal state that is divided into two sections to generate the keystream-

- Each of the 256 available bytes is permuted.
- Two index pointers (8 bits each).

The key-scheduling algorithm is known to initialize the permutation using a variable-length key, typically between 40 and 256 bits (KSA). A pseudo-random generating technique then generates the stream of bits.

For encryption −

- The user enters the Plaintext and a secret key.
- For the secret key entered, the encryption engine creates the keystream using the KSA and PRGA algorithms.
- Plaintext is XORed with the generated keystream. Because RC4 is a stream cipher, byte-by-byte XORing is used to generate the encrypted text.
- This encrypted text is now sent in encrypted form to the intended recipient.

For Decryption −

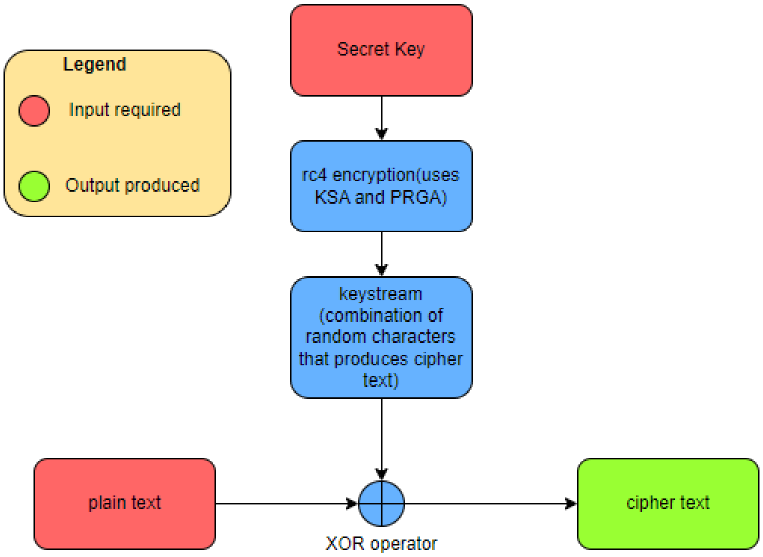
- The same byte-wise X-OR technique is used on the ciphertext to decrypt it.

Usage of RC4-

Over the years, RC4 has grown in popularity and has become a standard in commercial applications. It has a reputation for being a simple, quick, and inexpensive encryption technology.

The key benefits of RC4 are its ease of implementation and use, as well as its speed of operation and deployment. It enables efficient and quick processing of large data streams. In terms of memory usage, RC4 stream ciphers are also efficient. However, due to proof of flaws and cyberattacks in recent years, there have been calls to stop using RC4 encryption algorithms. Other drawbacks were identified, such as the inability to operate with small data streams and the need for additional investigation prior to implementing new systems.

The Internet Engineering Task Force (IETF) banned the usage of RC4 in TLS protocols in 2015. Because of threat vulnerabilities, Microsoft and Mozilla have also issued recommendations to limit the use of RC4. There are many RC4 based ecosystems such as WEP, WPA, BitTorrent protocol encryption, Microsoft Point-to-Point Encryption, etc. RC4A is a more powerful variation of RC4. RC4A+ is a modified version of RC4 with a more complex 3-phase key schedule that is 1.7 times longer than the basic RC4.


Legend
Input required
Output produced
Secret Key
rc4 encryption(uses KSA and PRGA)
keystream (combination of random characters that produces cipher text)
plain text
XOR operator
cipher text

Implementation of rc4 encryption algorithm

Advantages:

1. **Fast and efficient:** RC4 is a very fast and efficient encryption algorithm, which makes it suitable for use in applications where speed and efficiency are critical.
2. **Simple to implement:** RC4 is a relatively simple algorithm to implement, which means that it can be easily implemented in software or hardware.
3. **Variable key size:** RC4 supports variable key sizes, which makes it flexible and adaptable for different security requirements.
4. **Widely used:** RC4 has been widely used in various applications, including wireless networks, secure sockets layer (SSL), virtual private networks (VPN), and file encryption.

Disadvantages:

1. **Vulnerabilities:** RC4 has several known vulnerabilities that make it unsuitable for new applications. For example, there is a bias in the first few bytes of the keystream, which can be exploited to recover the key.
2. **Security weaknesses:** RC4 has some inherent weaknesses in its design, which make it less secure than other encryption algorithms, such as AES or ChaCha20.
3. **Limited key length:** The maximum key length for RC4 is 2048 bits, which may not be sufficient for some applications that require stronger encryption.

4. **Not recommended for new applications:** Due to its vulnerabilities and weaknesses, RC4 is no longer recommended for use in new applications. Other more secure stream cipher algorithms, such as AES-CTR or ChaCha20, should be used instead.

Key Distribution-

In public key cryptography, the key distribution of public keys is done through public key servers. When a person creates a key-pair, they keep one key private and the other, known as the public-key, is uploaded to a server where it can be accessed by anyone to send the user a private, encrypted, message. The public key can be distributed in four ways:

1. Public Announcement: Here the public key is broadcasted to everyone. The major weakness of this method is a forgery. Anyone can create a key claiming to be someone else and broadcast it. Until forgery is discovered can masquerade as claimed user.

2. Publicly Available Directory: In this type, the public key is stored in a public directory. Directories are trusted here, with properties like Participant Registration, access and allow to modify values at any time, contains entries like {name, public-key}. Directories can be accessed electronically still vulnerable to forgery or tampering.

3. Public Key Authority: It is similar to the directory but, improves security by tightening control over the distribution of keys from the directory. It requires users to know the public key for the directory. Whenever the keys are needed, real-time access to the directory is made by the user to obtain any desired public key securely.

4. Public Certification: This time authority provides a certificate (which binds an identity to the public key) to allow key exchange without real-time access to the public authority each time. The certificate is accompanied by some other info such as period of validity, rights of use, etc. All of this content is signed by the private key of the certificate authority and it can be verified by anyone possessing the authority’s public key. First sender and receiver both request CA for a certificate which contains a public key and other information and then they can exchange these certificates and can start communication.