UNIT-2

* Block Ciphers and Stream Ciphers:

↳ cryptographic techniques to ensure data confidentiality.

→ Confusion and Diffusion properties:

* Confusion:

→ Given any cipher text; there is no info about plain text, key and algorithm used.

→ Relationship b/w cipher text & key is as complex as possible.

→ Ex: Substitution.

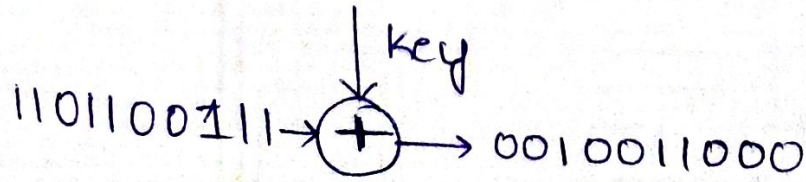
* Diffusion:

→ 1 bit change in plain text; significant effect on cipher text.

→ Ex: transposition.

* Stream Cipher:

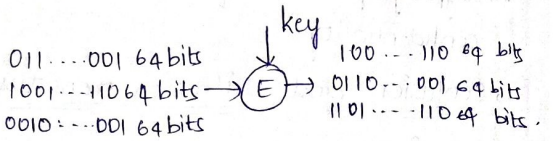
Each plain text digit is encrypted one at a time with the corresponding digit of keystream to give a digit of the cipher text stream.



bit by bit.

* Block Cipher:

A deterministic algorithm operating on a fixed-length group of bits called blocks.



STREAM	BLOCK
① Bit or byte	① 64/128 bits blocks.
② Complex design	② Simple design
③ Confusion	③ Confusion & diffusion
④ Faster.	④ Slower.
⑤ Encryption by Cipher feedback & O/P feedback.	⑤ Encryption by Electronic Code book & Cipher block chaining
⑥ Decryption by XOR	⑥ Decryption by performing reverse operations of encryption
⑦ Ex: Vernam Cipher	⑦ Ex: AES, DES.

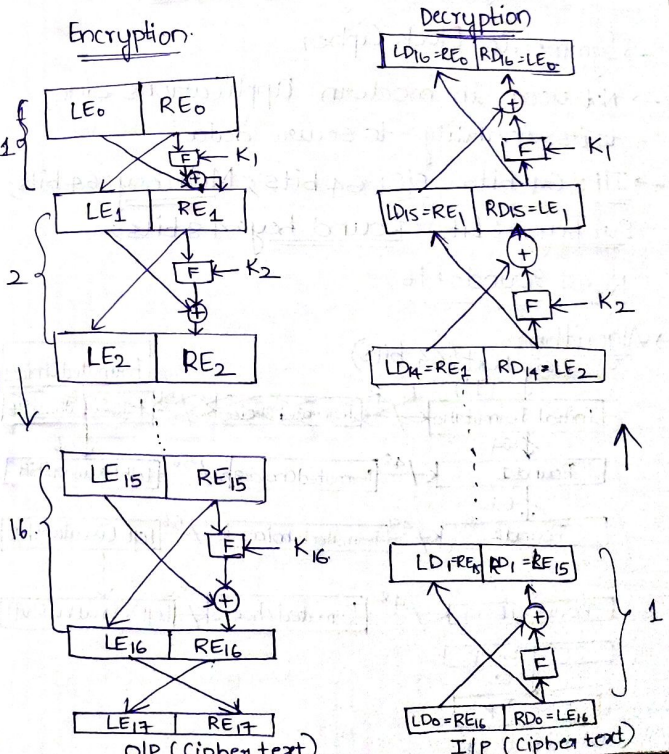
* Traditional Block Cipher Structure:

Common structures:

→ Fiestel Network
→ Substitution-Permutation Network (SPN)

* Fiestel Network:

used in DES; structure divides the block into 2 halves and processes them over multiple rounds with key-dependent functions.



Features:

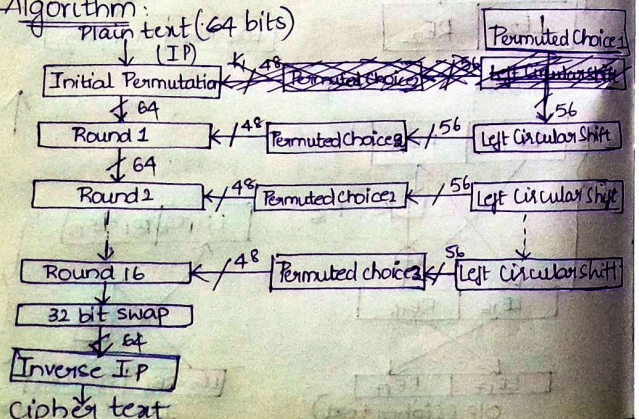
→ Block size & key size
→ No. of rounds
→ Sub key generation algorithm
→ Round fn
} helps to avoid breaking the cipher text

→ Fast encryption / decryption.
→ Ease of analysis.

* Data Encryption Standard (DES):

→ Symmetric Block Cipher
→ Not used in modern applications due to its inability to secure data.
→ IIP: 64 bits ; OP: 64 bits ; Main key: 64 bits.
Sub key: 56 bits ; Round key: 48 bits ;
No. of rounds: 16

→ Algorithm:



* Given 8×8 matrix with 64 bits:
when we perform initial permutation we get a diff. 8×8 matrix,

Initial Permutation:

1							8
							16
							24
17							32
25							40
33							48
41							56
49							64
57							

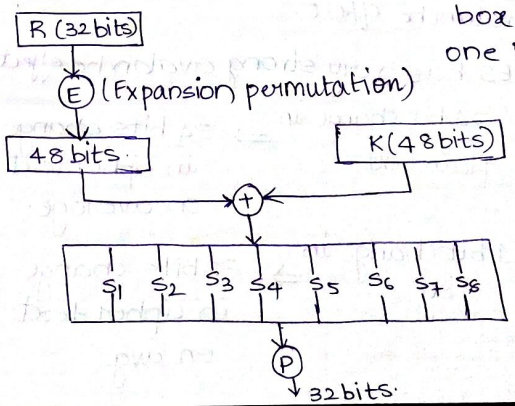
Inverse

40	8	48	16	56	24	64	32
33	1	41	9	49	57		25

→ F function (or) Mangler function:

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

8 Substitution boxes 1 Transposition box in one Round.



→ Key Scheduling:
generates sub keys used in encryption & decryption.

* Key Permutation: 64 bit user key is permuted using a fixed table; which produces 56 bit intermediate key.

* Key Rotation: 56 bit key is split into 28 bit halves; which are then rotated left by one/2 bits

*→ Avalanche effect and Strength of DES:

→ desirable property for all algorithms.

→ If there is one bit change in 64 bit I/P; there is a drastic change in 64 bit cipher text; then it ~~sho~~ said as avalanche effect.

→ DES has very strong avalanche effect.

⇒ 1 bit change in plain text ⟹ 34 bits change in cipher text on average.

⇒ 1 bit change in key ⇒ 35 bits change in cipher text on avg

* Strength of DES:

→ The use of 56 bit keys

→ Nature of DES ⇒ Cryptanalysis, 8 tables/sboxes

→ Timing attacks.

* Advanced Encryption Standard: (AES)

→ widely used symmetric block cipher that operates on 128 bit blocks with key sizes of 128, 192, 256 bits.

→ uses substitution-permutation network

* Structure: Block size: 128 bits
Key size: 128, 192/256 bits
Rounds: 10(128 bits), 12(192), 14(256)

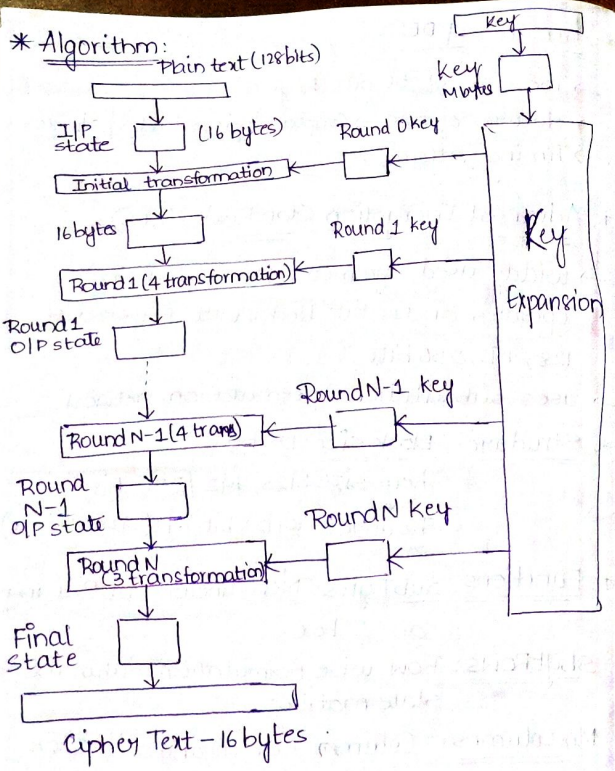
* Functions: SubBytes: Non-linear sub^n using an S box.

ShiftRows: Row wise permutation within the state matrix.

MixColumns: Column wise mixing operation.

* The key expansion algorithm derives all round keys from the original key using byte substitution and shifts.

* Algorithm:

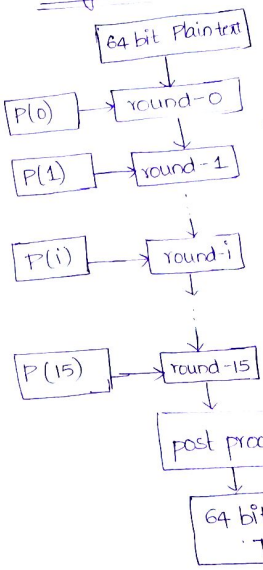


* Blowfish:

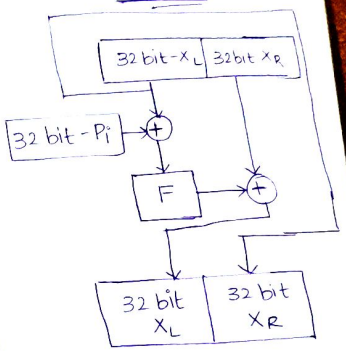
→ Symmetric Block cipher
→ known for its speed and flexibility in hardware and software.
→ 64 bit block size
→ 32 to 448 bit keysize
→ 16 rounds
→ Fiestel Network
→ Complex S-boxes that provide strong confusion

→ highly secure & efficient

Encryption:



Flow diagram each round Ri



* Triple DES: enhancement of DES to use security by applying DES thrice.
Encrypt with key-1; Decrypt with key2; Encrypt with key-3.

* CAST-128:

→ symmetric-key block cipher based on Feistel structure.

→ flexible and efficient.

* International Data Encryption Algorithm (IDEA):

→ symmetric key block cipher

→ resistant to cryptanalysis

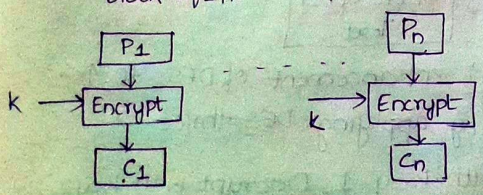
→ operates on 64 bits blocks with a 128 bit key.

→ uses a combination of modular addition, multiplication & XOR to provide security.

* Block Cipher Modes of Operation:

→ Electronic Code Book (ECB):

* easiest block cipher mode of functioning
* easy due to direct encryption of each block of I/P.



→ Cipher Block Chaining:

* previous cipher block is given as I/P to next encryption algorithm after XOR.

→ Cipher Feedback Mode:

* an initial vector IV is used for first encryption & O/P bits are divided as a set of s and b bits.
* LHS 's bits are selected along with plain text bits on which XOR is applied.
* the result is given to shift register.

→ O/P feedback Mode:

* Similar to above; except that it sends the encrypted O/P as feedback instead of the actual cipher which is XOR O/P.
* holds great resistance towards bit transmission errors.

→ Counter Mode:

* A counter - initiated value is encrypted and given as I/P to XOR with plain text which gives cipher text.

* Stream RC4:

→ bit by bit encryption

→ generates a pseudo random keystream from a secret key

→ key size: 40 to 2048 bits.

→ Key Scheduling Algorithm (KSA):
Initializes the state array based on the key.

→ Pseudo-Random Generation Algorithm (PRGA)
Generates key stream

KSA:

```
j = 0
for i = 0 to 255
    j = (j + s[i] + T[i]) mod 256
    swap (s[i], s[j])
```

Encryption = Plain text ⊕ New key.

No. of iterators = size of key.

* Public Key Cryptography:

→ Principles:

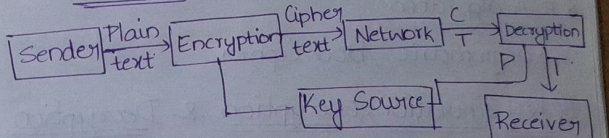
① Asymmetric Keys: → Public key
→ Private key.

② Confidentiality

③ Authentication

④ Non-repudiation.

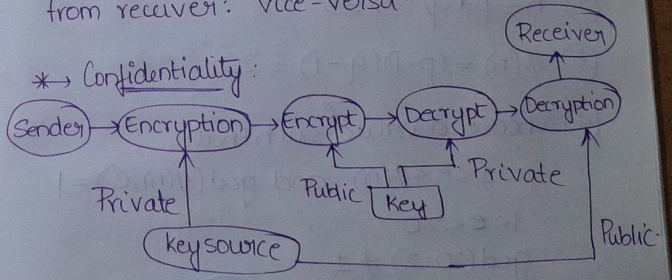
*→ Authentication:



both keys
from sender: sender: private key
receiver: public key.

from receiver: vice-versa.

*→ Confidentiality:



Public Key	Private Key
→ uses public & private key.	→ uses same key at sender & receiver
→ slow	→ fast
→ authentication & confidentiality	→ Confidentiality
→ Private key is secret Public key shared.	→ key is secret.

* RSA algorithm:

→ asymmetric, block cipher

Key Generation; Encryption & Decryption

① Select 2 large prime nos. p, q

Ex: p = 3 q = 11, m = 5

② $n = p * q \Rightarrow 33$

③ $\phi(n) = (p-1)(q-1) = 20$

④ Choose a value 'e' such that

$1 < e < \phi(n)$ and $\gcd(\phi(n), e) = 1$

$1 < e < 20$

$\gcd(20, 2) \neq 1$

$\gcd(20, 3) = 1$ $e = 3$

⑤ Calculate d using $ed \bmod (\phi(n)) = 1$

Private key = $\{d, n\} \Rightarrow \{7, 33\}$

public key = $\{e, n\} \Rightarrow \{3, 33\}$

$3d \bmod (20) = 1$

$3 \times d \,\%\, 20 = 1$

$\therefore d = 7$

⑥ Encryption:

$c = m^e \bmod n$

$= 5^3 \bmod 33 = 26$

⑦ Decryption:

$= c^d \bmod n$

$= 26^7 \bmod 33 = 5$

* → Security issues of RSA:

① Brute force attack.

② Timing attack.

③ Cipher text attack.

* Diffie - Hellman Key Exchange:

→ method for 2 parties to securely establish a shared secret key over a public channel.

→ symmetric encryption.

→ Procedure:

① Agree on public parameters.

p = prime no. q = primitive root.

$p = 23$; $g = 5$ (primitive root modulo 23)

② Private key generation:

Private key 1 ⇒ $a = 6$ ⇒ Party-1

Private key 2 ⇒ $b = 15$ ⇒ Party-2.

③ Public key computation:

$A = g^a \bmod p$; $B = g^b \bmod p$.

$= 5^6 \bmod 23$ $= 5^{15} \bmod 23$

$= 15625 \bmod 23 = 8$ $= 30517578125 \bmod 23 = 19$

④ Exchange public keys:

A → send 8 and B → send 19.

⑤ Compute shared secret key:

$K = B^a \bmod p$ $K = A^b \bmod p$

$= 19^6 \bmod 23$ $= 8^{15} \bmod 23$

$= 2$. $= 2$

Same secret key.

* Man in the Middle Attack (MITM).

→ attacker intercepts the common b/w 2 parties and pretends to each party to other.

① Legitimate key exchange setup:

Ex: Large prime $p = 23$.

base generator $g = 5$.

② Generate their private keys:

A → $a = 6$.

B → $b = 15$.

③ MITM attacker intercepts public keys:

$A = g^a \bmod p = 8$.

$B = g^b \bmod p = 19$.

→ A sends 8 ; but B receives $A' = 9$

→ B sends 19 ; but A " $B' = 4$.

④ Shared secret key computation:

for A: $K_1 = (B')^a \bmod p = 4^6 \bmod 23 = 18$.

for B: $K_2 = (A')^b \bmod p = 9^{15} \bmod 23 = 6$.

⑤ Attacker intercepts and modifies:

→ A encrypts data with $K_1 = 18$
→ B encrypts data with $K_2 = 6$.

* El Gamal Encryption & Decryption:

→ asymmetric encryption.
→ based on hardness of discrete logarithm problem.

① Select large prime $p = 11$
② private key $d = 3$.
③ select second part of encryption key e_1 ⇒ $e_1 = 2$
④ Calculate 3rd part of encryption key e_2.

$e_2 = e_1^d \bmod p$.
$= 2^3 \bmod 11 = 8$ ($e^2 = 8$)

⑤ Public Key $= (e_1, e_2, p) = (2, 8, 11)$
Private key ⇒ $d = 3$

⑥ Encryption: Random no. $R = 4$

* calculate cipher text

$C_1 = e_1^R \bmod p = 2^4 \bmod 11 = 5$

* Calculate $C_2 = (\text{plain text} \times e_2^R) \bmod p$
⇓
7
$= (7 \times 8^4) \bmod 11 = 6$

$(C_1, C_2) = (5, 6)$

⑦ Decryption:

Plain text $= [C_2 \times (C_1^d)^{-1}] \bmod p$
$= (6 \times (5^3)^{-1}) \bmod 11$
$= (6 \times (125)^{-1}) \bmod 11$.
⇓

$(125)^{-1} = 125\,x \bmod 11 = 1$
$x = 3$

Plain text $= (6 \times 3) \bmod 11 = 7$.

* Elliptic Curve Cryptography:

→ form of public key cryptography
→ elliptic curves over finite fields.
→ same level of security as RSA with smaller keysizes
→ efficient

* → Elliptic curve eqn ⇒ $y^2 = x^3 + ax + b$ over a finite field.

* → Rules for addition:

→ 3 points of curve joined by st. line then their sum = 0.

* → Elliptic curves over Z_p (prime):

also called prime curve

$$y^2 \bmod p = (x^3 + ax + b) \bmod p$$

$$P + 0 = P$$

$P = (x, y)$ then $-P = (x, -y)$

$$P + (-P) = 0$$

* → Elliptic curves over $GF(2^m)$:

$$y^2 + xy = x^3 + ax^2 + b.$$

$$P + 0 = P$$

$P = (x, y)$ then $-P = (x, x + y)$

Applications:

↳ mobiles, Bitcoin, SSL/TLS.