UNIT-II

Symmetric Key Ciphers:-

Block Cipher principles:-

DES -

AES -

Blowfish -

RC5 -

IDEA -

Block cipher operation -

Stream Ciphers

RC4.

Asymmetric key Ciphers:-

principles of public key cryptosystems -

RSA algorithm -

Elgamal cryptography

Diffie-Hellman key Exchange

Knapsack Algorithm.

---

* Block Cipher principles:-

→ A block cipher is a symmetric-key encryption algorithm that processes data in fixed-size blocks using a secret key.

→ It transforms a plaintext block into a cipher text.

→ Block ciphers are built in the Feistel cipher structure.

→ Block cipher has a specific number of rounds and keys for generating cipher-text.

Some of these principles are:

1) Number of Rounds:- The number of Rounds is regularly considered in design criteria.

→ 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.

→ If the function is very much complicated to understand, more time hacker will take to decode the data.

→ you have to take non-linear functions, because linear functions are easy.

2) Confusion and Diffusion:- The cipher should provide

3) Key Schedule Algorithm:- It should be designed carefully to ensure that the keys used for encryption are independent & unpredictable.

→ The key schedule should also resist attacks that exploit weak or key-dependent properties of the cipher.

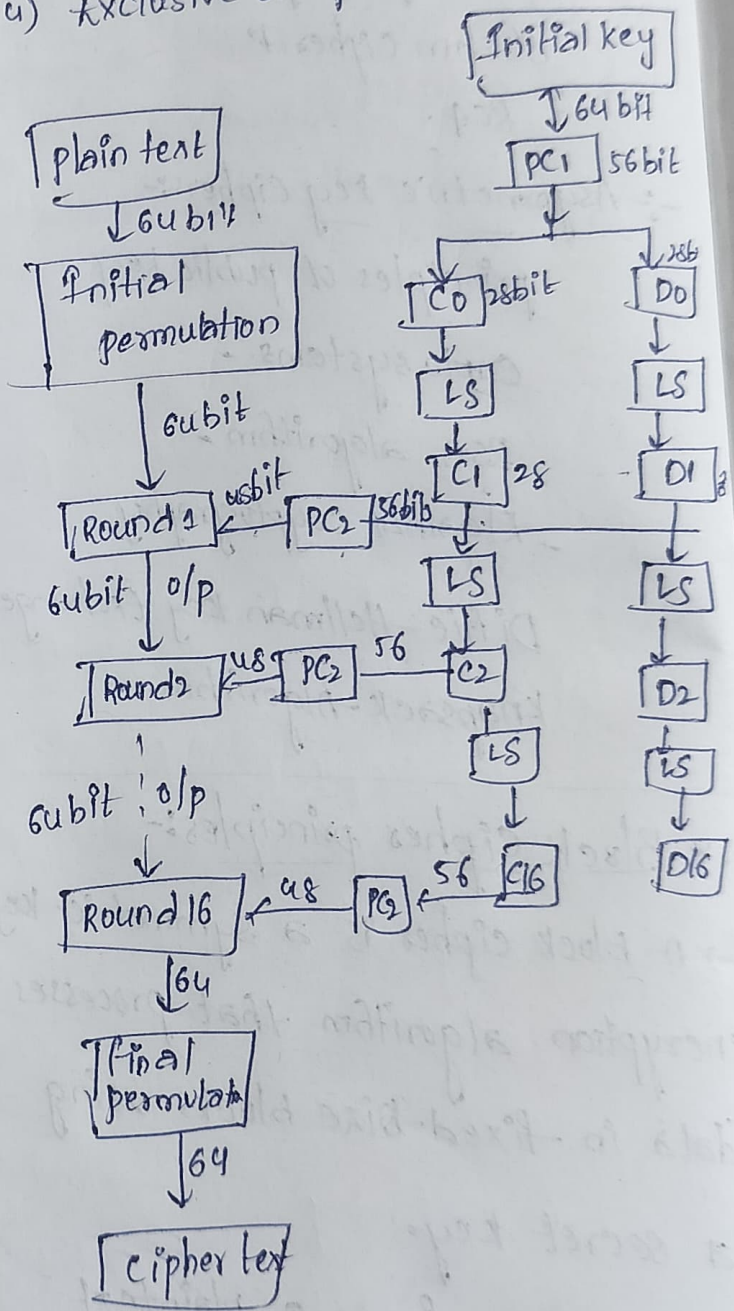
DES [~~Data Standard~~ Data Encryption Standard]:

- It is a block cipher algorithm
- It converts the plain text into cipher text

* It has 16 rounds
* The text size of plain text & cipher text is 64 bits
* The key size is 48 bits. The remaining 16 bits are removed.
* 8 bits are removed for parity and 8 bits for rearrangement.

In each round (4) steps are performed

1) Dividing bits into two parts-32 bits for each
2) Bit shuffling
3) Non linear substitutions
4) Exclusive OR operations



-> In PC1, initially there are 64 bits 8 bits are removed from every parity 8th position

64 = (8×8)

-64-8 = 56

* Then apply left circular shift after dividing 56 bits into 2 parts C0 and D0, each having 28 bits

-> D1 and C1 are obtained as result.

Left Circular shift:-

* Move the bits based on Round number

* for Rounds 1, 2, 9, 16 - 1 bit shift

other Rounds - 2 bit shift.

* Here in PC2, C1 and D1 are combined to form 56 bits again, permutted choice 2 is applied.

* 56 bits are rearranged, permutted and in that 48 bits are selected.

Round 1 - 48 bits are the key

At last cipher text is 64 bit

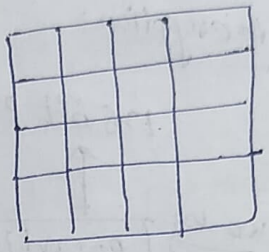
AES Algorithm

-> Advanced Encryption standard

-> It is a block cipher algorithm

-> It has i/p array, state array and a key array.

1) Input Array



each cell = 1 byte / 8 bit

Total = 16 cells

16 × 8 = 128 bits

= 4 words (32 each)

in i/p array PT is represented.

2) State Array

-> It is used to store intermediate states within the rounds

	$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
1 byte / 1 word	$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
	$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
	$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

Total 4 words

3) Key Array:-

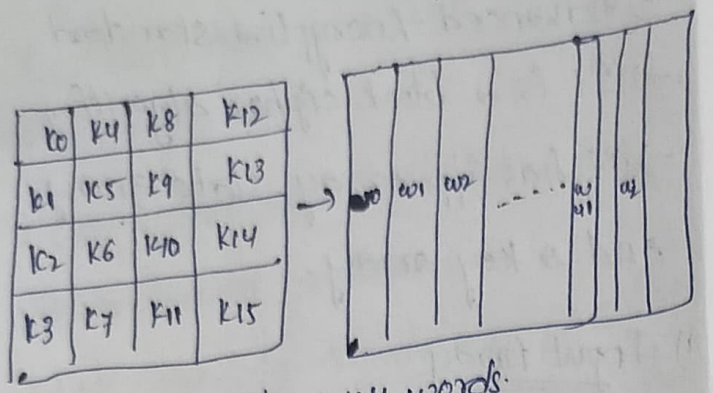
Actually 4 words expanded into 44 words

Each round =

-> In AES there are 10 rounds

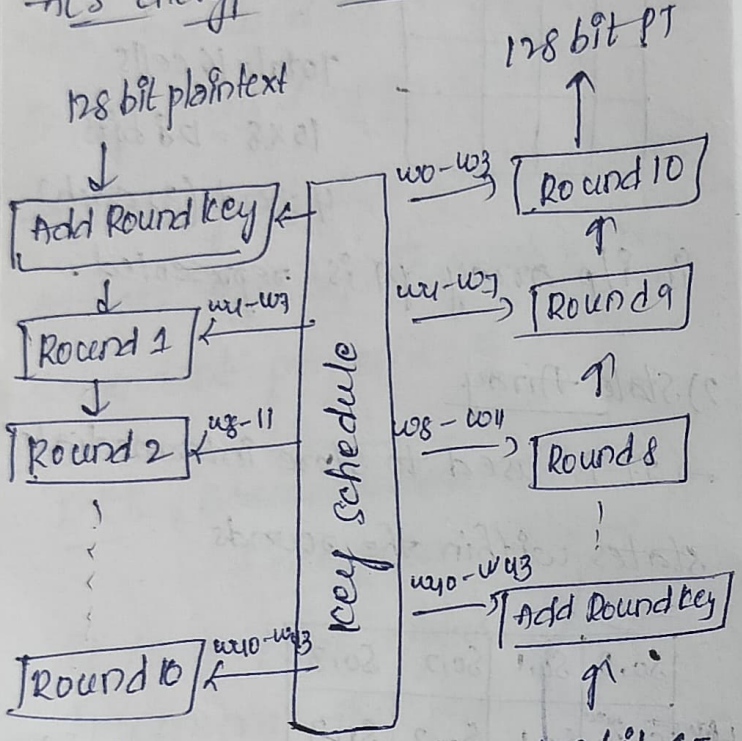
Each round = 4 words

∴ 10 rounds × 4 words = 40+4
= 44 words.



4 words → 44 words.

AES encryption & decryption.



* The encryption & Decryption starts from Add Round key.
* Each process starts from Round 1 and ends with Round 10.

In each round we have 4 steps
1) substitute Bytes
2) shift Rows (LCS)
3) mix columns - Not in Round (10)
4) Add Round key
↓
XOR operation between PT & key.

* 128 bit plain text is sending into the Add Round key along with the words w0, w1, w2, & w3
* for each and every round we have four words.
* Total 44 words along with add round key.
* Then we will get 128 bit cipher text.
* This is the process of Encryption
* This process will continue until the 128 bit plain text is obtained.

* Blow fish:

* It is a block cipher algorithm
* It is a symmetric key cryptography
* The input size is 64 bits
* The key size is variable length key i.e from 32 to 448 bits
* It is more secure

properties

* It is very fast
* It takes less memory
* It is simple to understand

Blow fish Algorithm has 2 parts

1) key generation
2) Data encryption

* key generation:

1) keys are stored in an array

$K_1, K_2, K_3 \dots K_n \quad [1 \le n \le 14]$

* Length of each block is 32 bits

(32 × 14 = 448 bits)

2) Initialize an array (P)

$P_1, P_2, P_3 \dots P_{18}$

length of each word is 32 bits

3) Initialize s-boxes (4)
   └→ substitution

$S_1 \Rightarrow S_0, S_1 \dots S_{255}$

$S_2 \Rightarrow S_0, S_1 \dots S_{255}$

$S_3 \Rightarrow$ "

$S_4 \Rightarrow$ "

4) Initialize each element of parray and S-boxes with heradecimal values

5) XOR operations are performed

$P_1 = P_1 \text{ XOR } K_1$

$P_2 = P_2 \text{ XOR } K_2$

⋮

$P_{14} = P_{14} \text{ XOR } K_{14}$

$P_{15} = P_{15} \text{ XOR } K_1$ (because only 14 keys)

⋮

$P_{18} = P_{18} \text{ XOR } K_4$

6) Take 64 bit PT (Initially all bits are '0')
   subkey is generated.

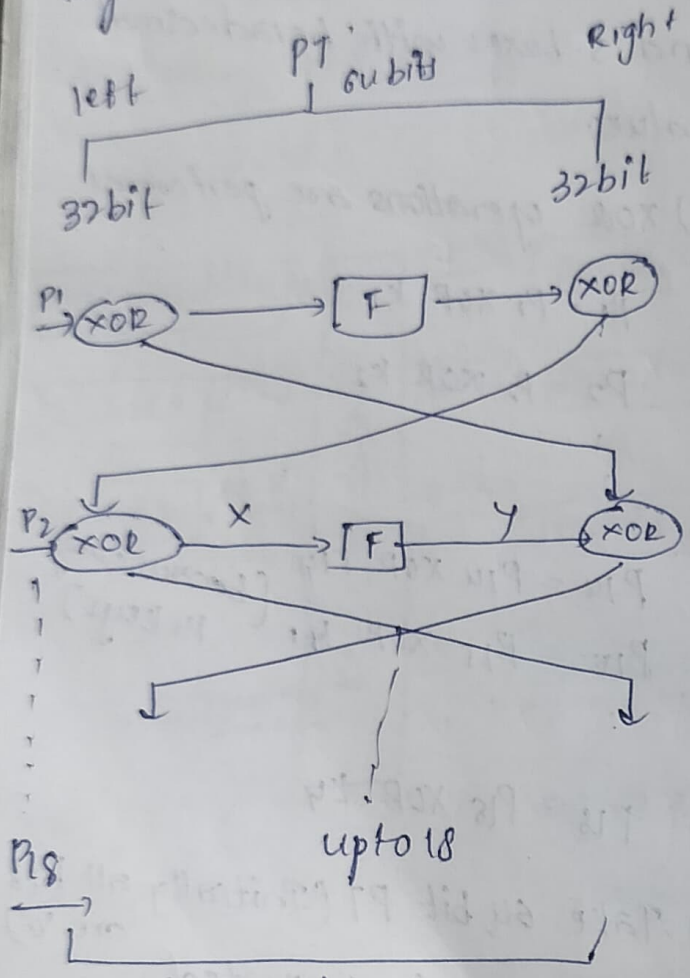
(2) Data Encryption

* Divide the plain text into two parts.
* Then perform XOR operation with P_1 with left 32 bit

we will get output 'x'

* This output will sent into a function and do the function we will get output 'y'.

• The output 'y' and 32 bit of Right side will do the XOR operation

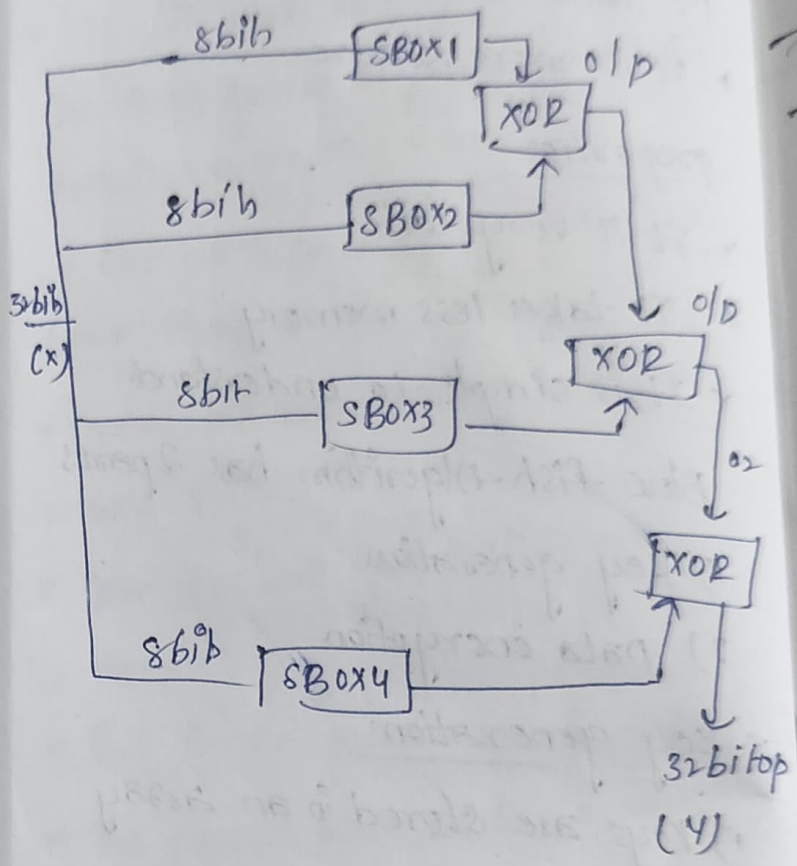


* The output of XOR operation of left side will send to right side vice-versa

* This process continue upto P18

* After, we have to merge the 32 bit of left side & right side to get cipher text of 64 bits

* The diagrametic representation of function is



RC5 Algorithm:

* RC stands for Rivest cipher
* It is designed by Ronald Rivest
* RC5 is a block cipher technique
* It is a symmetric block cipher that means same secret key is used for both enc & dec

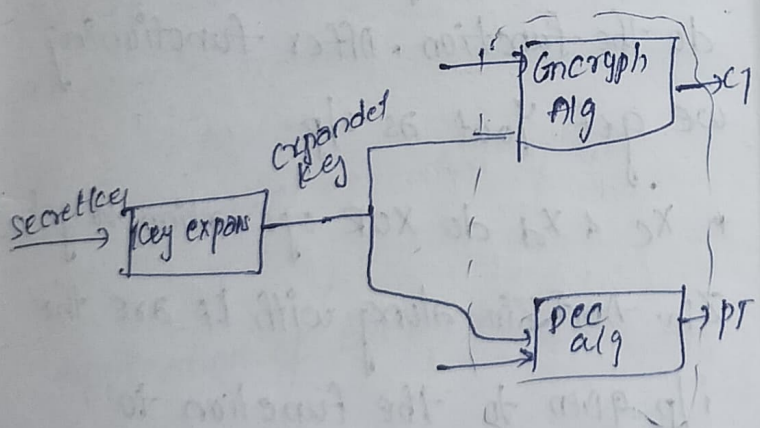
RC5 having three parameter

1) w - word size [16, 32, 64]
   encrypts two word blocks [2w]
2) r - No. of rounds [0, 1 ... 255]
3) b - No. of bytes [0, 1 ... 255]

→ It's similar to DES
→ Based on choosing 'w' remaining things will be based
→ More rounds provides an increased level of security.

RC5 algorithm having three ~~algorithms~~ components

1) key expansion algorithm
2) Encryption algorithm
3) Decryption "



Advantages:

1) High level of security
2) fast encryption & decryption
3) flexible key length.

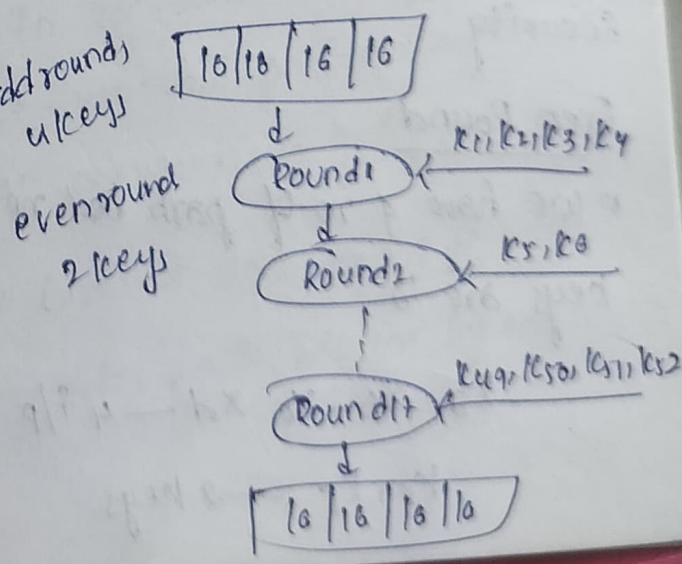
Dis:

1) vulnerable to side-channel attacks
2) Limited adoption
3) patent issues.

IDEA [International data Encryption Algorithm].

→ It is a Block cipher Algorithm
→ Symmetric key cryptography
→ Frestel cipher → 4 blocks

i/p size = 64 bits — 16, 16, 16, 16
key size = 128 bits — into 52 subkeys
No. of rounds = 17.
plain text = 64



Odd Rounds:

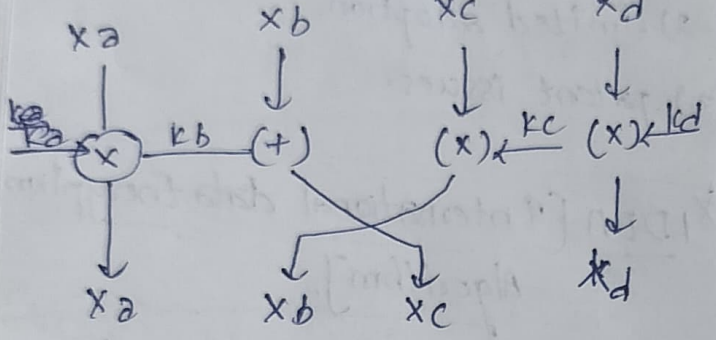
Input is divided in four parts

keys are '4'

16 16 16 16

X_a X_b X_c X_d → i/p

K_a K_b K_c K_d → keys



* X_a & K_a are the i/p and new X_a is generated as o/p.

* X_d and K_d are the i/p & new X_d is generated as o/p

* But in case of X_b & X_c the o/p's of X_b, K_b and X_c, K_c are swapped each other in order to ensure security.

Even Round:

* we have 4 no. of parts but no. of keys are '2'

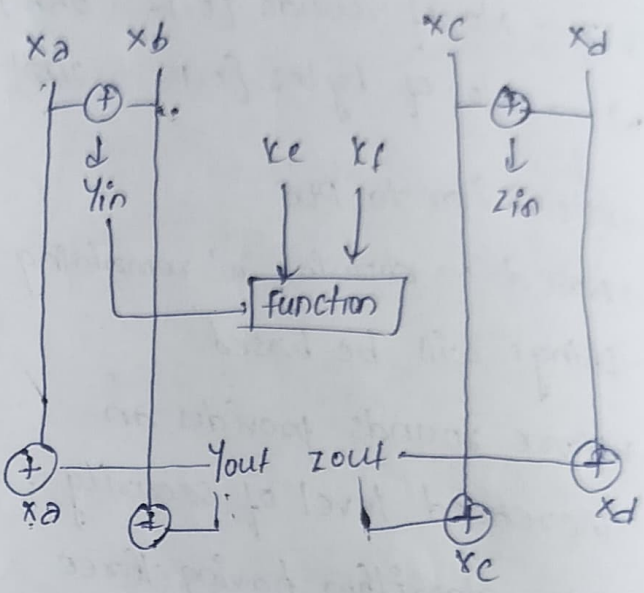
X_a, X_b, X_c X_d — 4 i/p

K_e, K_f — 2 keys

$Y_{in} = X_a \oplus X_b$

$Z_{in} = X_c \oplus X_d$

Now i/p = 2 & key = 2



$X_a = X_a \oplus Y_{out}$

$X_b = X_b \oplus Y_{out}$

$X_c = X_c \oplus Z_{out}$

$X_d = X_d \oplus Z_{out}$

* X_a & X_b do XOR operation to get Y_{in} and along K_e are the i/ps to do the function. After functioning we get Y_{out} as o/p.

* X_c & X_d do XOR operation to get Z_{in}. And Z_{in} along with K_f are the i/p given to the function to generate the Z_{out} as o/p

* X_b and Yout will do XOR operation to get X_b as o/p
* X_c & Zout will do XOR operation to get X_c as o/p
* X_d and Zout will do XOR operation to get X_d as o/p

* Block cipher modes of operation:

* A block cipher is an encryption method that encrypts data in fixed size blocks

① ECB: properties

1) Block size equal to 64 bit
2) key is same in everywhere
3) size of PT & CT should be same.

2) CBC: Adv good authentication
3) CFB: can handle data streams
4) counter mode.

Adv
parallel execution of enc

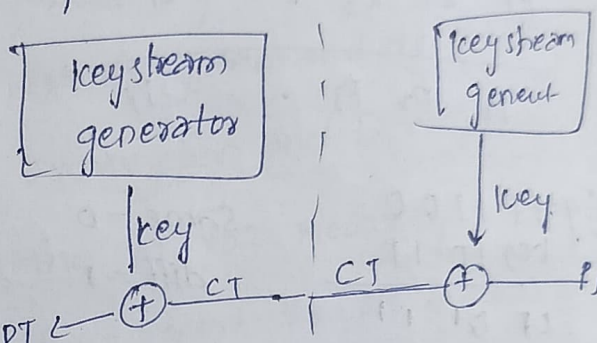
Dis
Application of Block cipher

1) Data Enc
2) File & Disk Encryption
3) Virtual private Networks

* Stream Cipher

→ A stream cipher is a type of symmetric encryption that encrypts data one bit or byte at a time by combining it with keystream generated from a shared secret key

* plain text is divided into number of streams



* keys are generated from keystream generator
* Bitwise XOR operation is performed b/w the key & PT. As a result we get CT.
* In dec, with the help of key & CT, we will perform a bitwise XOR operation to get PT
* Bitwise XOR means it will consider each bit one by one.

Explanation

m_1 m_2 m_3 ---- m_i → PT

k_1 k_2 k_3 ---- k_i → key

c_1 c_2 c_3 ---- c_i → CT

(Ciphertext ⊕ key → PT) decryption

c_1 c_2 c_3 ---- c_i

k_1 k_2 k_3 ---- k_i

p_1 p_2 p_3 ---- p_i

Eg: PT 1100 same = 0

key 1011 diff = 1

CT 0111

RC4

→ stream cipher algorithm

Procedure

1) uses an array (s) - state vector of length 256 bits (0-255)

2) It has a key encoded with ASCII

3) It has a key array of length 256 (0-255)

RC4 algorithm has three steps

1) key scheduling

2) key stream generation

3) Encryption & decryption

① key scheduling

* No. of iterations is equal to size of s-array.

initialize j = 0

for i = 0 to 255 do (state vector, key array)

$j = [j + s(i) + T(i)] \bmod 256$

Swap (s[i], s[j]) ;

Eg: s-array [0 1 2 3 4 5 6 7] (s[0] 0 1 2 3)

key-array [1 2 3 6]

PT = [1 2 2 2]

Initialize T-array with key

S & T should be same.

T = [1 2 3 6 1 2 3 6] (0) (1) (2) ...

i) j = 0

for i = 0 to 7

$j = [0 + 0 + 1] \bmod 8$

$1 \bmod 8 = 1$

Swap s(0) & s(1)

s = [1 0 2 3 4 5 6 7]

ii) for i=0 to 7

$j = [1 + 0 + 2) \bmod 8$

$= 3 \bmod 8 = j$

$= 3$

swap S(1) & S(3) (0) & (1)

$= [1\ 3\ 2\ 0\ 4\ 5\ 6\ 7)$

iii) for i=2.

~~for 1~~.

$j = [3 + 2 + 3]$

$j = 8$

$= 8 \bmod 8 = 1$

j=0. swap S(2) & S(0)

$S = [2\ 3\ 1\ 0\ 4\ 5\ 6\ 7]$

iv) for i=3

$j = [4 + 3 + 6]$

$= 13 \bmod 8 = 5$

swap S(3) & S(5)

~~Stream generation:~~

No. of iterations is equal to size of key (4)

i, j = 0.

while (true)

i = (i+1) mod 256;

j = (j + S[i]) mod 256;

swap (S[i], S[j]);

t = (S[i] + S[j]) mod 256

K = S[t];

for 1st iteration we get K(0)

for 2nd K(1)

K(n)

=) New key is obtained & used for encryption and decryption.

Encryption and Decryption:-

Enc PT XOR New key.

first convert the PT & new key into binary

PT = 1 2 2 2

PT: 0001 0010 0010 0010

key: ( ) ( ) ( ) ( )

CT =

decryption - CT XOR New key

*Principles of public key

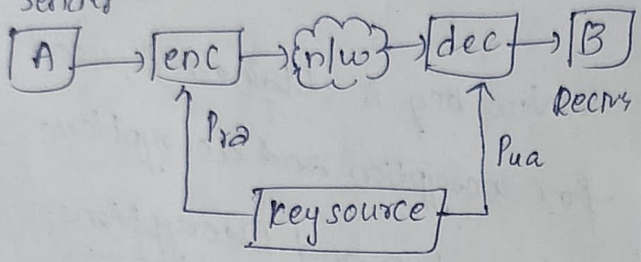
for Asymmetric key cryptography we have different key for enc & dec

These are two principals

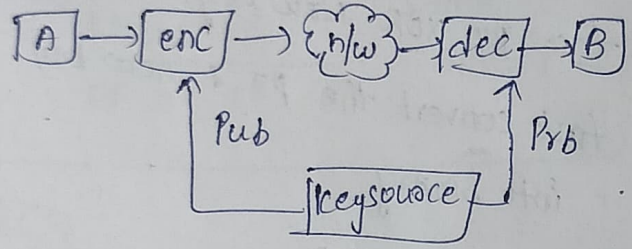
1) Authentication

2) Confidentiality

① Sender



②



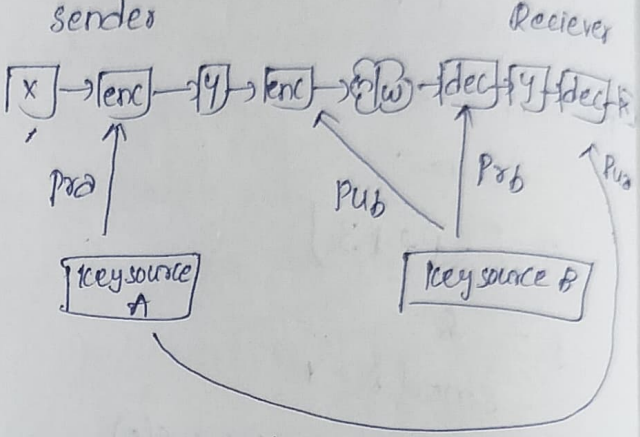
* Here the sender sends the PT & it will encrypted with the help of key which is named as private, taken from key source.

* Then the encrypted message will enter into a n/w and come out of this n/w then to decrypt the msg we are using public key from the key source

* At last the cipher text is converted into plain text which is recieved by reciever

* This is Authentication

2) Confidentiality



In confidentiality we have diff key sources (A & B)

* The keysource A will generate the public A & private A keys which is provided to x (sender)

* The keysource B will generate the private B & public B which is provided to reciever

* The private B key of keysource B provided to reciever & public B key of keysource B is provided to sender

* Crypto Systems:

An asymmetric key cryptosystems uses two different keys for security operations

- public key → shared openly for enc
- private key → kept secret for dec

Asymmetric crypto systems are used for

- Confidentiality
- Authentication
- Digital Signature
- key exchange

Components:

1) key Generation Algorithm
   - Generate pair of keys
   - keys are mathematically related

2) Encryption algorithm
   - Uses the reciever's public key to convert PT → CT

3) Decryption alg
   - Uses the private key to recover PT from CT
   - Ensures confidentiality

4) Digital Signature Algorithm:
   - Sender signs a message using their private key.
   - Anyone can verify using public key.

Examples of Assymetric Cryptosystems:

1) RSA:
   - Used for encryption & digital signatu

2) Diffie-Hellman
   - key exchange protocol
   - Allows two parties to generate a shared secret.

3) ElGamal:
   - Used for enc & digital signatures.

Characteristics:

- Two key mechanism
- High Security
- Slower than symmetric.

Useful for

- Secure key exchange
- Secure email

* RSA algorithm

→ It is one of the public key ENC algorithm used for both enc & dec of data.

→ In this the P.T & C.T, key values are represented in numbers, integers for some 'n' value and ranges from 0, n-1

→ By Ran Rivest Shamir Adlemn

→ It is a block cipher method

ENC $C = m^e \bmod n$

DEC $m = c^d \bmod n$

$m \rightarrow$ PT

$C \rightarrow$ CT

$n \rightarrow$ integer

public key $\{e, n\}$

public key $\{d, n\}$

Steps:

1) select p, q values where p, q are prime numbers

2) Calculate 'n' $n = p \times q$

3) calculate $\phi(n)$ Euler's totient function

$\phi(n) = (p-1)(q-1)$

4) select 'e' value where 'e' is relatively prime to $\phi(n)$

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

5) calculate $d = de = \bmod \phi(n) = 1$

6) ENC $\boxed{C = m^e \bmod n}$

7) DEC $\boxed{m = c^d \bmod n}$

Eg 1) $p = 3$ $q = 5$

2) $n = p \times q$

$n = 3 \times 5$

$n = 15$

3) $\phi(n) = (p-1)(q-1)$

$= (3-1)(5-1)$

$= 2 \times 4 = 8$

4) $\gcd(\phi(n), e) = 1$

$\phi(n) = 8 = 1, 2, 4, 8$

8 → 1, 2, 4, 8 → 1, 2, 4, 8 ∴ $\boxed{e = 3}$

5) $d \times e \bmod \phi(n) = 1$

$d \times 3 \bmod 8 = 1$

$\therefore 1 \times 3 \bmod 8 = 3 \bmod 8 = 3$

$2 \times 3 \bmod 8 = 6 \bmod 8 = 6$

$3 \times 3 \bmod 8 = 9 \bmod 8 = 1$

ENC $C = 5^3 \bmod 15$

$C = 125 \bmod 15$

$\boxed{C = 5}$

DEC $m = c^d \bmod n$

$= 5^3 \bmod 15$

$= 125 \bmod 15$ $\boxed{m = 5}$

* Elgamal Cryptography

→ Assymetric key cryptography.

1) select large prime number (p) = 11

2) select a dec. key also called as private key

d = 3

3) select second part of encryption key (e1) = 2

e1 = 2

4) ~~Select~~ third part of enc key calculate

$e_2 = e_1^d \bmod P$

$= (2)^3 \bmod 11$

$= 8 \bmod = 8$

e2 = 8

5) public key = (e1, e2, p) & private key = d

pub = (2, 8, 11)

ENC:

1) select random Integer (R)

R = 4

2) calculate $C_1 = e_1^r \bmod P$

$= 2^4 \bmod 11 = 16 \bmod 11$

$= 5$

C1 = 5

calculate $C_2 = (PT \times e_2^R) \bmod p$

$= (7 \times 8^4) \bmod 11$

$= 28672 \bmod 11 = 6$

C2 = 6

CT = (C1, C2) = (5, 6)

DEC:

1) $PT = \left[c_2 \times (c_1)^p\right)^{-1}] \bmod P$

$= \left(6 \times (5)^3\right)^{-1}] \bmod 11$

$= 6 (5^3)^{-1} \bmod 11$

$= 6 (125)^{-1} \bmod 11$

$= 125 \times x \bmod 11 = 1$

If x = 3 $= 125 \times 3 \bmod 11 = 375 \bmod 11$
$= 1$

x = 3

$6 \times 3 \bmod 11 = 18 \bmod 11 = 7$.

PT = 7

∴ Correct =

Diffie-Hellman key Exchange:

→ It is a asymmetric key

→ It is used to exchange keys b/w sender and Reciever

* It just exchange the key, doesn't perform any enc/dec algorithm.

1) consider prime number 'q'
   calculate 'α' 'α' is relatively prime to 'q'

2) user 'A' generate key
   selects a private key 'X_A'
   → calculate public key

$$Y_A = \alpha^{X_A} \bmod q$$

3) user 'B' generate key
   selects a private key 'X_B'
   calculate public key

$$Y_B = \alpha^{X_B} \bmod q$$

4) Exchange key values

A B

Y_A Y_B

Y_B → Y_A

5) user A generate Secret key

$$K_A = Y_B^{X_A} \bmod q$$

6) user B generate Secret key

$$K_B = Y_A^{X_B} \bmod q$$

A	B
q, α, X_A	q, α, X_B
Y_A	Y_B
K_A	K_B

Ex: $q = 11$, $\alpha = ?$, $X_A = 5$, $X_B = 7$ find

Y_A, Y_B, K_A, K_B

$\alpha = 1 \Rightarrow (\alpha^i \bmod q) = i = 1$ to 10, $\alpha = \alpha$

$1^1 \bmod 11 = 1$

$1^2 \bmod 11 = 1$

$1^3 \bmod 11 = 1$

$\alpha \neq$ as it generate same values i.e. 1

$\alpha = 2$

$2^1 \bmod 11 = 2$

$2^2 \bmod 11 = 4$

$2^3 \bmod 11 = 8$

$2^4 \bmod 11 = 5$

$2^5 \bmod 11 = 10$

$\therefore \alpha = 2$

Step 2: User A

Private key $X_A = 5$

$Y_A = \alpha^{X_A} \bmod q$

$Y_A = 2^5 \bmod 11$

$\boxed{Y_A = 10}$

Step 3: User B

Private key $X_B = 7$

$Y_B = \alpha^{X_B} \bmod q$

$Y_B = 2^7 \bmod 11$

$\boxed{Y_B = 7}$

Step 4

A	B
$X_A = 5$	$X_B = 7$
$Y_A = 10$	$Y_B = 7$
$\alpha = 2$	$\alpha = 2$
$q = 11$	$q = 11$
$Y_B = 7$	$Y_A = 10$

Step 5: User A generate secret key

$K_A = Y_B^{X_A} \bmod q$

$K_A = 7^5 \bmod 11$

Step 6:

$K_B = Y_A^{X_B} \bmod q$

$= 10^7 \bmod 11$

$K_B =$

---

* Knapsack Algorithm:-

* It was proposed by Hellman

* It is a Asymetric key cryptography

Ex: (weights) = (1, 6, 8, 15 & 24)

-) In general knapsack, we select weights to achieve a sum

Sum = 30

||

1, 6, 8, and 15

PT = 1 0 0 1 1

1 6 8 15 24

1 0 0 15 24 = 40

1 1 0 1 0

1 6 8 15 24

1 + 6 + 15 = 22

CT = plain text * corresponding weight

CT = 40 22.

Key Generation

1) public key (Hard knapsack)

2) private key (Easy knapsack)