PAGE No.
DATE / /

A

* RC4 Algorithm.

→ RC4 is a stream cipher and variable - length Key algorithm.

→ Encrypts one byte at a time

→ The Key input is a pseudorandom bit generator that produces a stream 8-bit number that is unpredictable.

① Key - Generation Algorithm :-

a) A variable - length Key from 1 to 256 bytes is used to initialise a 256 bytes states vector S with elements S[0] to S[256].

b) A byte K is generated from S by selecting one of the 256 entries in a systematic fashion.

② Key - Scheduling Algorithm :- The entries of S are set equal to the values from 0-255 in ascending order, a temporary vector T is created. If length of HOLK Key K is 256 bytes, the K is assigned to T.

PAGE No.
DATE / /

Main Formula.

for i = 0 to 255
    j = (j + S[i] + T[i]) % 256.
    Swap(S[i], S[j]);

Here S is the state vector. T is the temporary vector and j is element which will be swapped.

After the S-array is completely changed then the following algorithm is used.

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]

Then continue with encryption using the new K array and S vector.

DATE / / /

* A5/1 Stream Cipher.

→ A5/1 is a stream cipher used to provide over the air comm. privacy in GSM cellular telephone standards.

→ Uses three LFSRs (Linear-feedback shift register).

→ A register is clocked if its clocking bit agrees with the clocking bit of atleast one of the other two register.

→ LFSR (1) → 19-bits Feedback Polynomial.
$x^{19} + x^{18} + x^{17} + x^{14} + 1$.

clocking bit → 8
Tapped bits → 13, 16, 17, 18

→ LFSR (2) → 22 bits Feedback Polynomial
$x^{22} + x^{21} + 1$

clocking bit → 10.
Tapped bits → 20, 21.

→ LFSR (3) → 23 bits Feedback polynomial
$x^{23} + x^{22} + x^{21} + x^8 + 1$

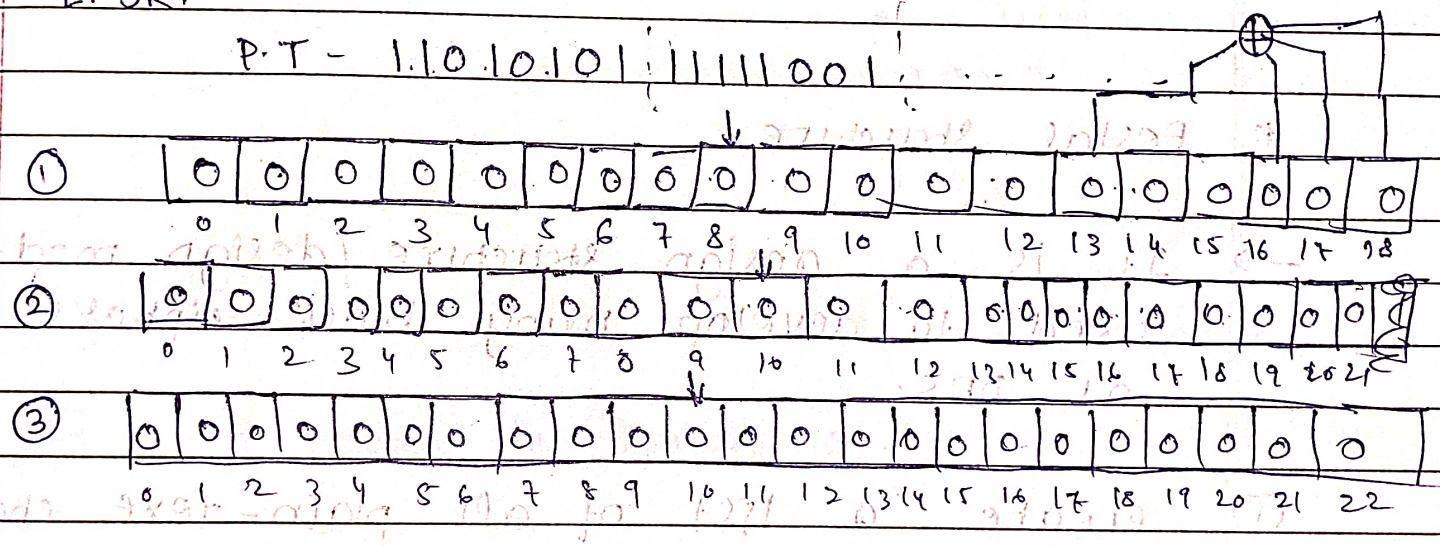
clocking bit = 8
Tapped bit = 7, 20, 21, 22

PAGE No.
DATE / /

→ The Keystream generator produces an infinite sequence of pseudorandom bits that are XORed with the plain-text.

→ Keystream generator is initialised with a secret key and a public frame number.

→ The Keystream generator produces a keystream bit by clocking each LFSR in turn and XORing the selected outbits, from each LFSR.



Majority = maj(A[8], B[10], C[10])

Key bit (Every iteration) = A[18] ⊕ B[21] ⊕ C[22]

* Block ciphers.

→ A block ciphers takes a block of plaintext bits and generates a block of cipher text bits generally of same size.

→ Avoid very small block size. Taking smaller size will reduce the possible block permutatio

→ Multiple of 8-bit is the perfect size.

* Festal Structure.

→ It is a design structure / design model used to develop many block ciphers such as DES.

① Create a list of all plain-text characters

② Convert the plain text to Ascii and then 8-bit binary format.

③ Divide the binary Plain Text String into two halves, Ll and Rl.

④ Generate a random binary keys (K1 and K2) of length equal to half the length of plain text for two rounds.

PAGE No.
DATE / /

F1 = XOR (R1, K1)

F2 = XOR (R2, K2)

R2 = XOR (F1, L1)
L2 = R1

R3 = XOR (F2, L2)
L3 = R2.

* AES (Advanced Encryption Standard)

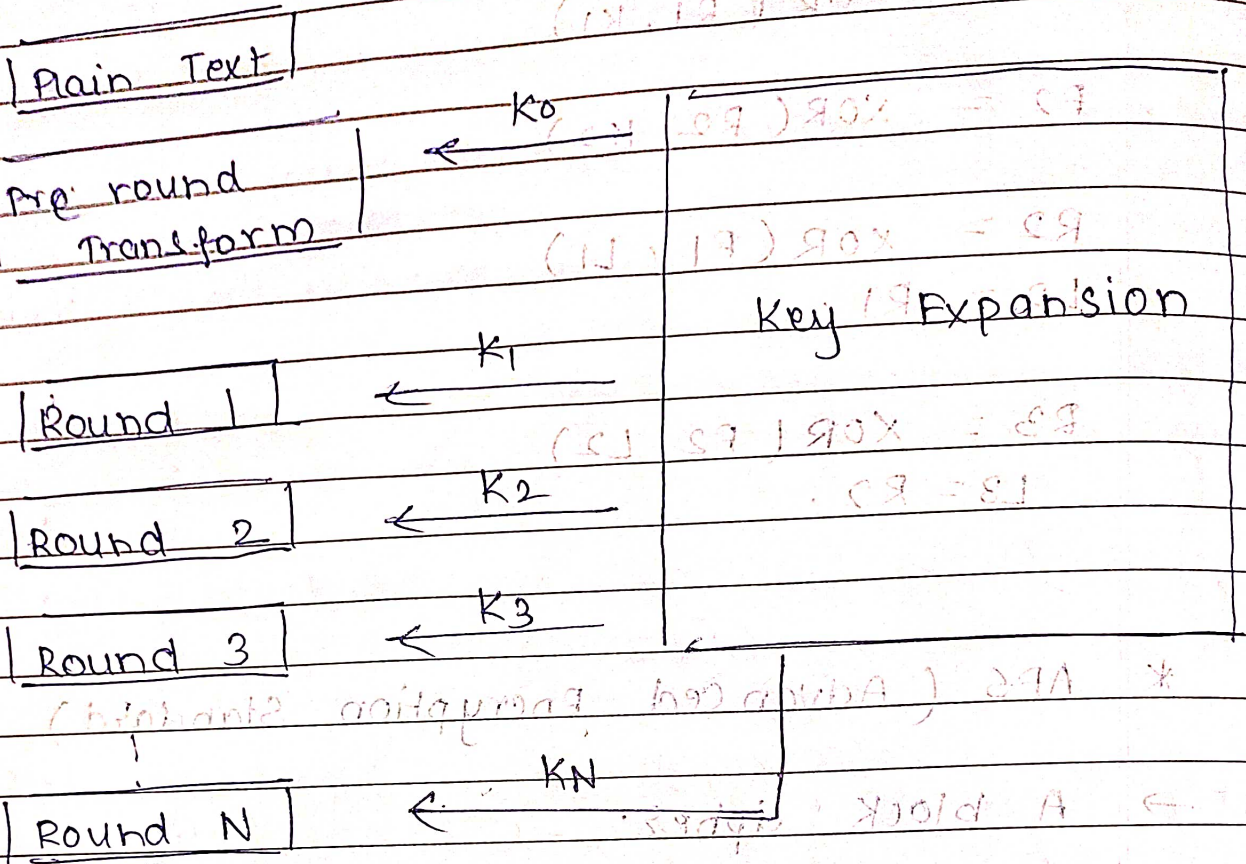
→ A block cipher.

→ The key-size can be 128/192/256 bits

→ Encrypts data in blocks of 128 bits.

No. of rounds depend on the length of key.

① 128 bit key → 10 rounds
② 192 bit key → 12 rounds
③ 256 bit key → 14 rounds.



Encryption:

AES considers 16 byte.

b0	b4	b8	b12
b1	b5	b9	b13
b2	b6	b10	b14
b3	b7	b11	b15

PAGE No.
DATE / /

Each round comprises of 4 steps

① SubBytes :- Each byte is substituted with another byte. Performed using a lookup table also called the S-Box.

② ShiftRows :- Each row is shifted a particular no. of times.

③ MixColomns :- Each column is multiplied with a specific matrix. (Matrix Multiplication)

④ Add Round Key :- Resultant output of previous stages is XORed with corresponding round key.

Decryption :-

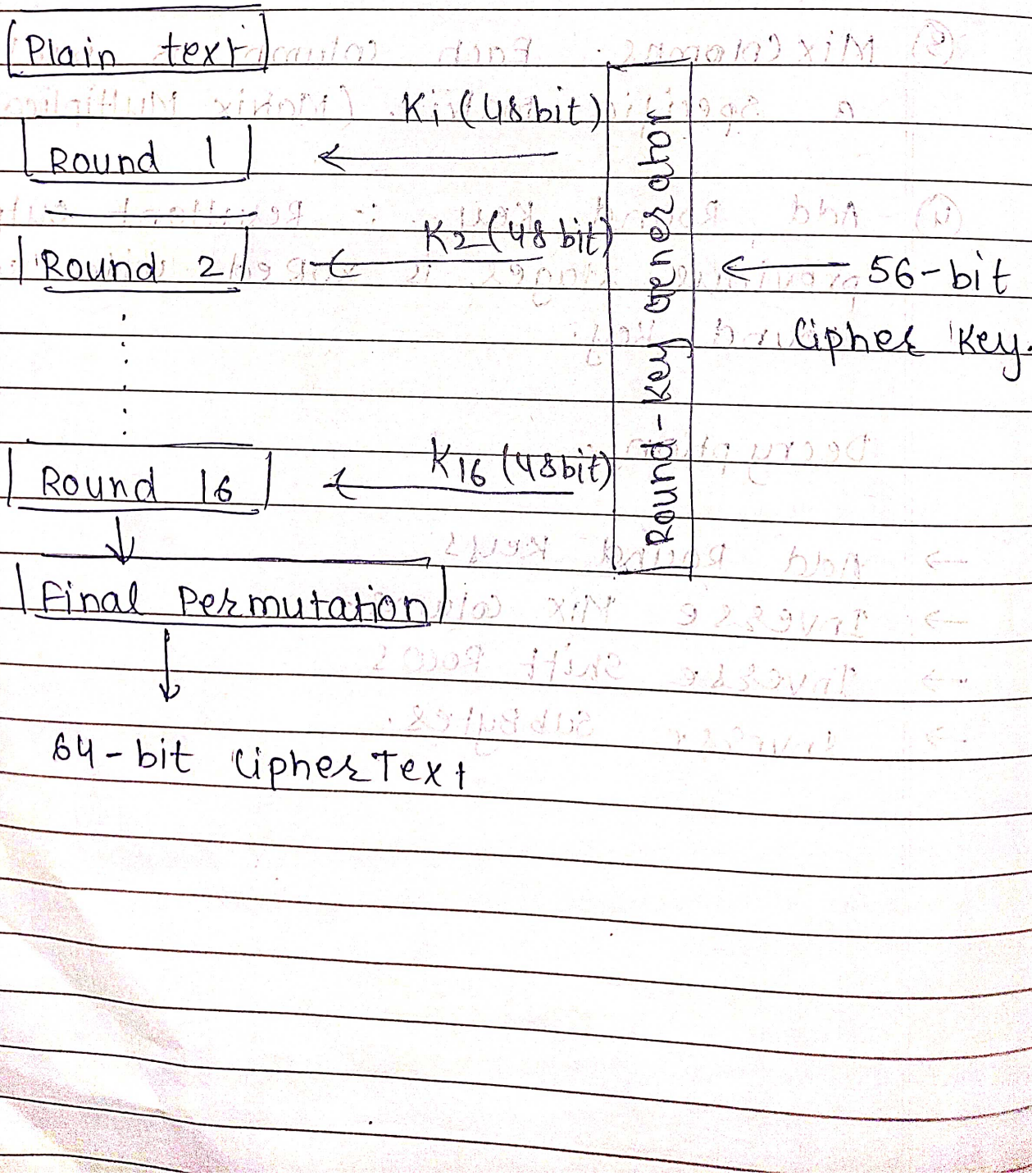
→ Add Round Keys
→ Inverse Mix Colums
→ Inverse Shift Rows
→ Inverse SubBytes.

* DES (Data Encryption Standard.)

→ Uses 16 round Feistel structure

→ Block size of 64 bit.

→ Effective Key length of 56 bits.

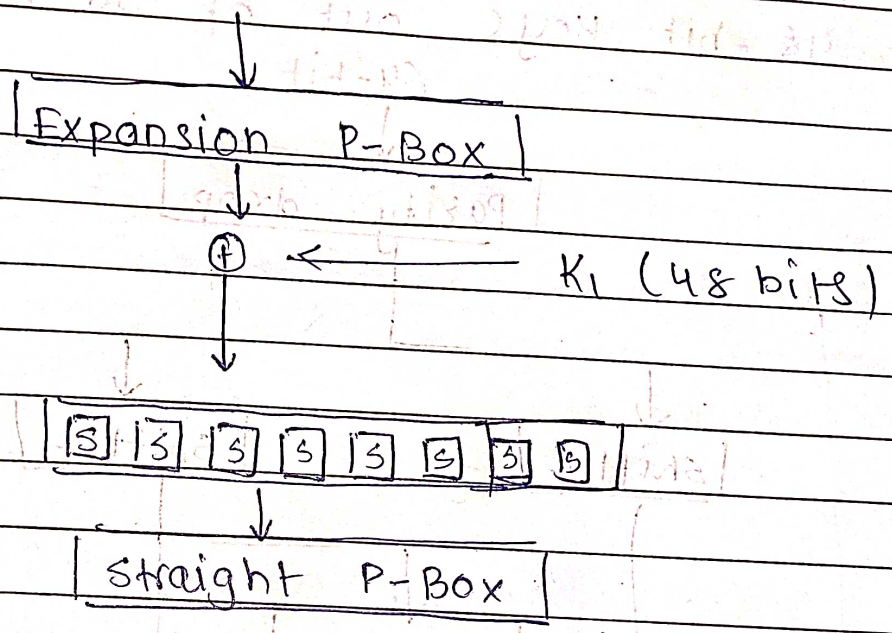


Hello.

PAGE No.
DATE / /

① Round function.

→ The heart of this cipher is the DES function. The DES function applies a 48-bit key to the right-most 32-bits and produces a 32-bit output.



→ Expansion Permutation Box. : Expands the current 32-bit input to 48-bit input.

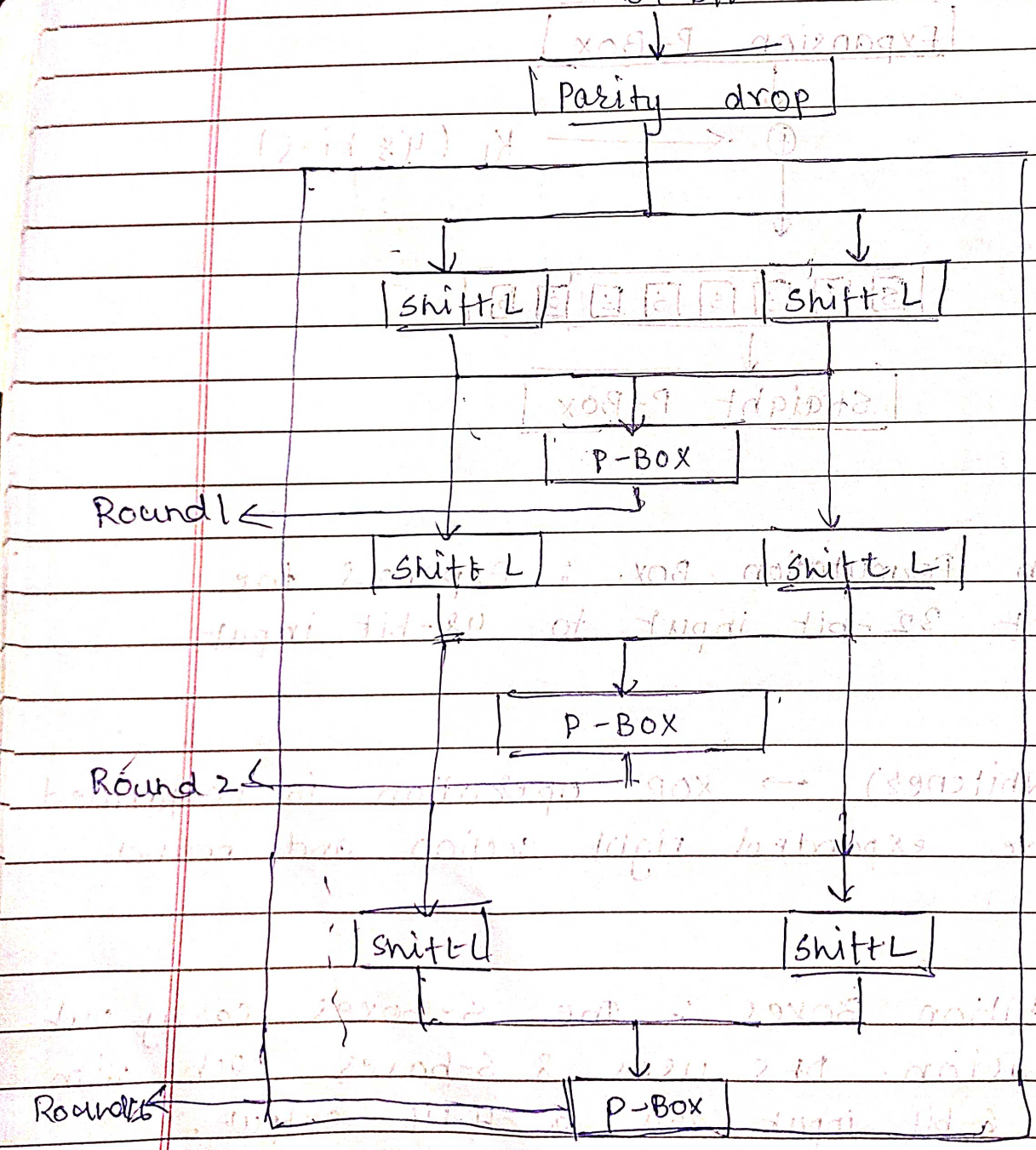
→ XOR (Whitener) → XOR operation is performed on the expanded right section and round key.

→ Substitution Boxes :- The S-boxes carry out confusion. DES uses 8 S-boxes each with a 6-bit input and a 4-bit output.

→ Straight Permutation.

② Key Generation.

→ The round key generator generates 16-48-bit keys out of 56-bit cipher keys



PAGE No.
DATE / /

③ DES Analysis

a) Avalanche Effect : A small change in plaintext result in the very great change in the cipher text.

b) Completeness :- Each bit of ciphertext depends on many bits of plain text.

X X

*

* Shared Key-Gen using Diffie-Hellman Key exchange protocol.

→ Used to establish a shared secret that can be used for secret communication while exchanging data over a public network using the elliptic curve

Person 1	Person 2
Public Keys - P, G	Public Keys - P, G
Private Key - a	Private Key - b
Key generated - $x = G^a \bmod p$	Key generated - $y = G^b \bmod P$

DATE / /

Key recieved = y

Generated secret Key = $K_a = y^a \bmod p$

Key recieved = x

Generated secret Key = $K_b = x^b \bmod p$

Now $K_a = K_b$

Steps.

① Person 1 and person 2 get public numbers $p = 23$, $G = 9$.

② Person 1 selected a private key $a = 4$ and person 2 selected private key $b = 3$

③ Person 1 $= 9^4 \bmod 23 = 6$.

Person 2 $= 9^3 \bmod 23 = 16$

Person 1 recieves $y = 16$

Person 2 recieves $x = 6$.

Person 1 : $K_a = y^a \bmod p$

$= 16^4 \bmod 23 = 9$

Person 2 : $K_b = 6^3 \bmod 23 = 9$.

PAGE No.
DATE / /

Part 208

* Asymmetric Key Cryptography

→ Needs two keys. Public Key used to encrypt (or code) a message and anyone can use it.

→ The other key allows you to decode/decrypt the message and is kept private

* Advantage :- No need for secure Key exchange b/w Alice and Bob.

→ Example - A client sends its public key to the server and requests for some data.

→ The server encrypts the data using client's public key and sends the encrypted data.

→ Client receives his data and decrypts it.

* Applications

① SSL Protocol
② SSH protocol.
③ Digital Signatures.

PAGE No.
DATE / /

* Digital Signature:

→ Result of encrypting the Hash of the data to be exchanged.

→ A mathematical scheme for verifying the authenticity of digital messages and documents.

→ A valid digital signature gives a recieptent reason to believe that the message was created by a known sender, that sender cannot deny having sent that message

→ A hash of a document is created using a mathematical algorithm. This hash is specific to this particular document; even the slightest change would result in a different hash.

→ Encrypted using signer's private key, the encrypted hash and the signer's public key are combined into a digital signature which is appended to the document.

PAGE No.
DATE / /

* Properties of a Good Cryptographic Function.

① Deterministic: No matter how many times you passe a particular input 'm' through a hash function you will always get the same result h(m)

② Compression: For any size input 'm', output length of h(m) should be small.

③ Efficiency/quick computation: For any 'm' hash function should be capable of returning the hash h(m) of an input quickly.

④ Pre-Image Resistance: Given h(m), there is no way to find an 'm'. It should be difficult to invert the hashes.

⑤ Avalanche Effect: Even if you make a small change in your input m the changes in hash output will be huge.

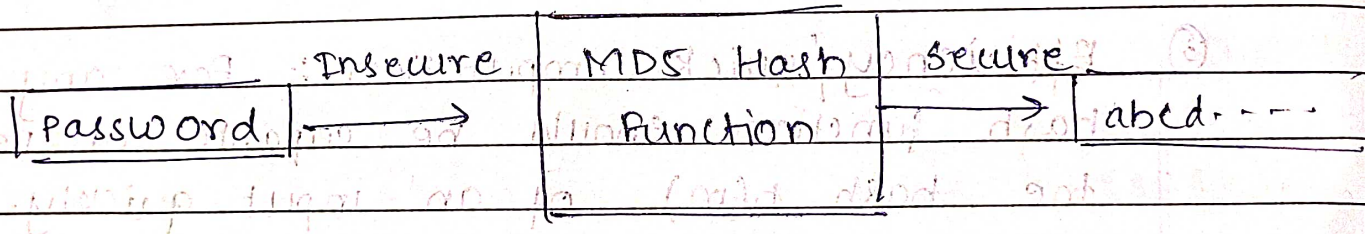
⑥ Collision Resistance: Given two different where h(m) and h(n) are their respective hashes, it is infeasible for h(m) to be equal to h(n). It is computationally infeasible to find two values that hash the same thing.

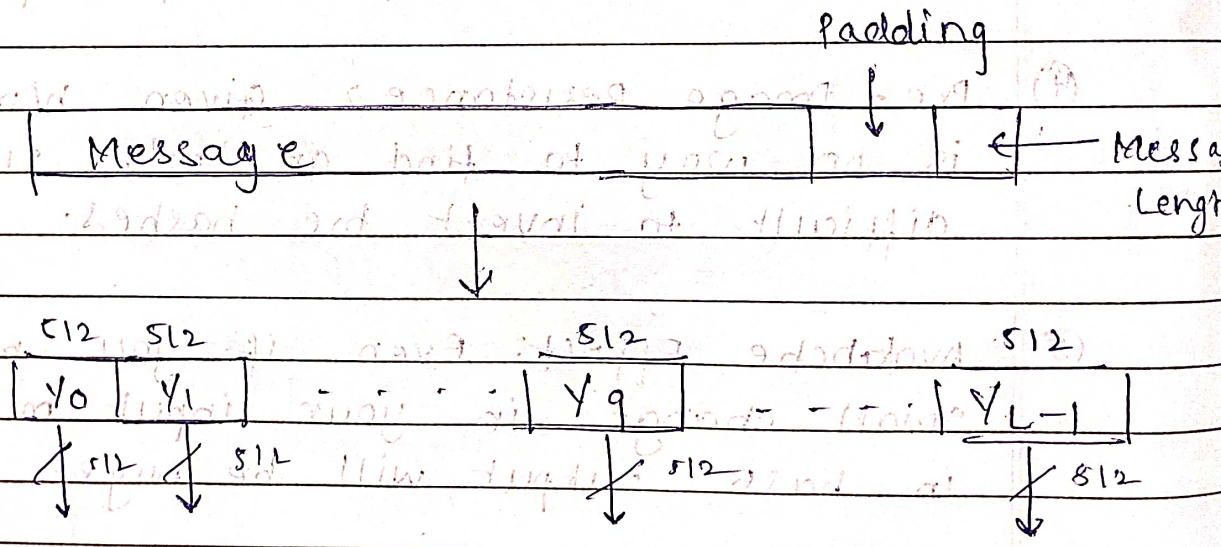
* MD5 - (Message - Digest 5.)

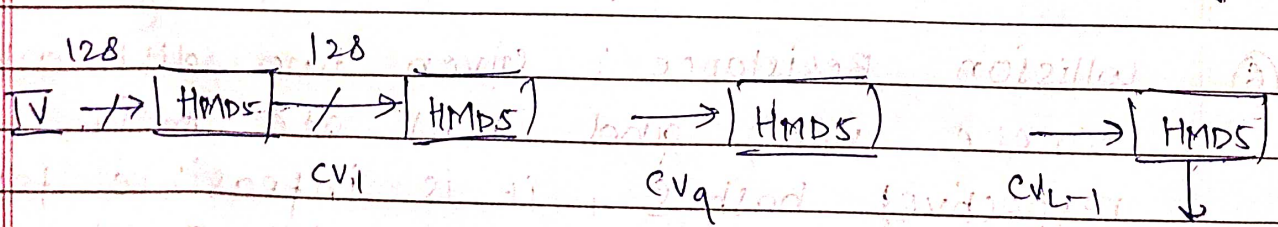
→ Processes input data in 512 - bit blocks.

→ MD5 generated message digest of 128 bits

→ Used as an encryption or fingerprint function for a file.







PAGE No.
DATE / /

① Append Padding Bits : Bits are padded to make length of orignal message equal to 64-bit less than exact multiple of 512.

→ Padding is single 1 followed by 0s.

→ Padding is always added when length of orignal message is 64 bit less than exact multiple of 512.

② Append Length. : calculate orignal length of the message excluding padding bits and add it to the end of the message as a 64-bit value.

③ Form Input Blocks. : Divide input message into blocks of 512 bit each. Divide into 16 sub-blocks.

→ Each sub-block contains 32-bits.

④ Initialize chaining variables.

Word A	01	23	45	67
Word B	89	AB	CD	EF
Word C	FE	DC	BA	98
Word D	76	54	32	10

PAGE NO.
DATE / /

⑤ Process message in 16-word blocks.

$b = b + ((a + AUX\ Fun(B,C,D) + M[i] + T[K]) <<< s)$

(1 . . . . 16) $F(B,C,D) = (B \& C) || (\sim B \&$

(17 . . . 32) $G(B,C,D) = (B \& D) || (C \&$

(33 . . . . 48) $H(B,C,D) = B ? C ? D$

(49 . . . . 64) $I(B,C,D) = C ? (B || \sim D)$

→ A added to 32-bit output

→ M[i] is added to output

→ T[K] is added to output.

→ Circular left shifts by s-bits

→ Variable b added to output

→ Resultant output becomes input for next round.

→ T[K] is an array of constant derived from sin value containing 64 elements

PAGE No.
DATE / /

where each element is 32-bit.

* SHA-1

→ Processes input data in 512-bit blocks (16 32-bit words) and generates a 160-bit (5 words of 32-bit hash value called message-digest.

→ Considered insecure since 2005.

Steps.

① Append Padding Bits.

→ Same as MD5.

② Append Length

→ Same as MD5.

③ Forming Input Blocks

→ Same as MD5.

④ Initialising chaining variables

→ Same as MD5.

(5) consists of 4 rounds with each rounds consisting of 20 steps (Total 80)

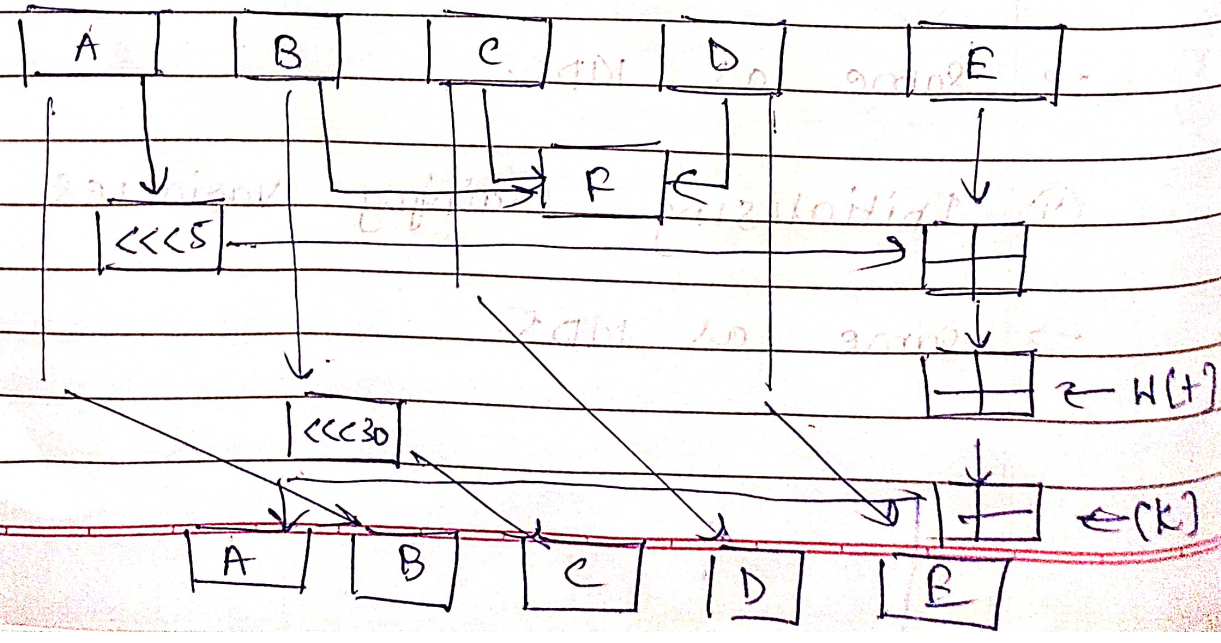
abcde = e + (Process P + $S^5(a)$ + W[t] + K[t])

Each round takes 512 bit block, registers a, b, c, d, e and a constant K[t]
t = 0 - 79 and 1-19 for each round.

W[t] = S' (W[t-16] XOR W[t-14] XOR W[t-8] XOR W[t-3])

S' denotes circular left shift by 1

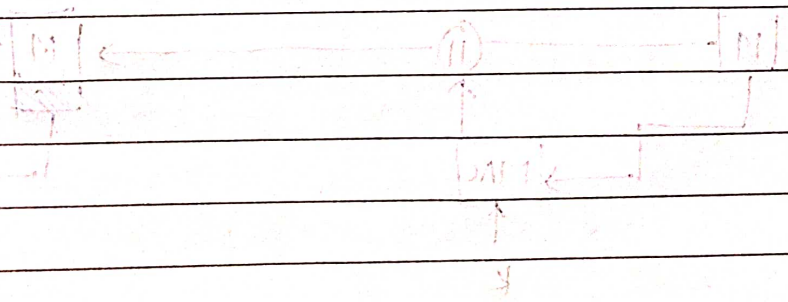
Round		Function
1 (1-20)	⟶	(b & c) || (~b & d)
2 (21-40)	⟶	b XOR c XOR d
3 (41-60)	⟶	(b & c) or (b & d) || (c &
4 (61-80)	⟶	(b XOR c XOR d)



PAGE No.
DATE / /

Difference in MD5 and

	MD5	SHA-1
Message Digest	128	160
Operations to break from MD.	2^{128}	2^{160}
If 2 msg MD are same	2^{60}	2^{80}
Speed	Faster	slower
Software	Simple	Simple.
Attacks	Attempts to compromise some extent	No claim.



DATE / /

* MAC (Message Authentication Code)

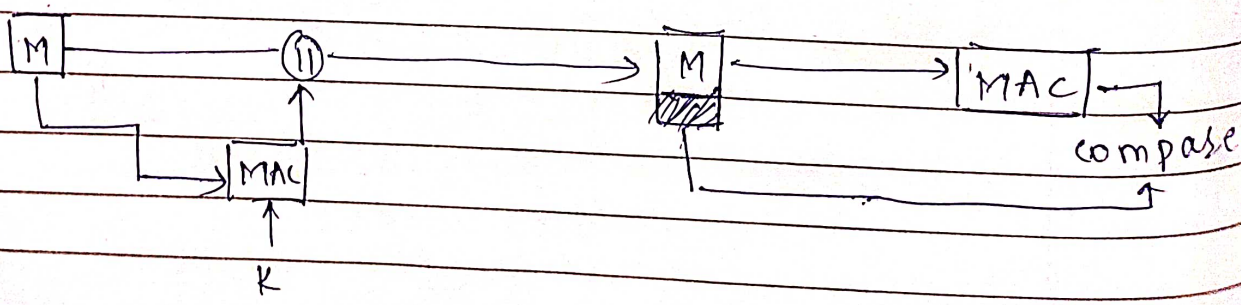
→ MACs allow the message and the digest to be sent over insecure channels.

→ Involves shared symmetric key b/w Alice and Bob.

→ Also known as cryptographic checksum. involving the use of secret key to generate small fixed-size block of data.

→ When A has a message to send to B. It calculates the MAC as a function of the message MAC = C(M, K) which is appended to the orignal messag

→ Reciever also generates the code and compares it with what he/she recieved thus ensuring the originality of the msg.



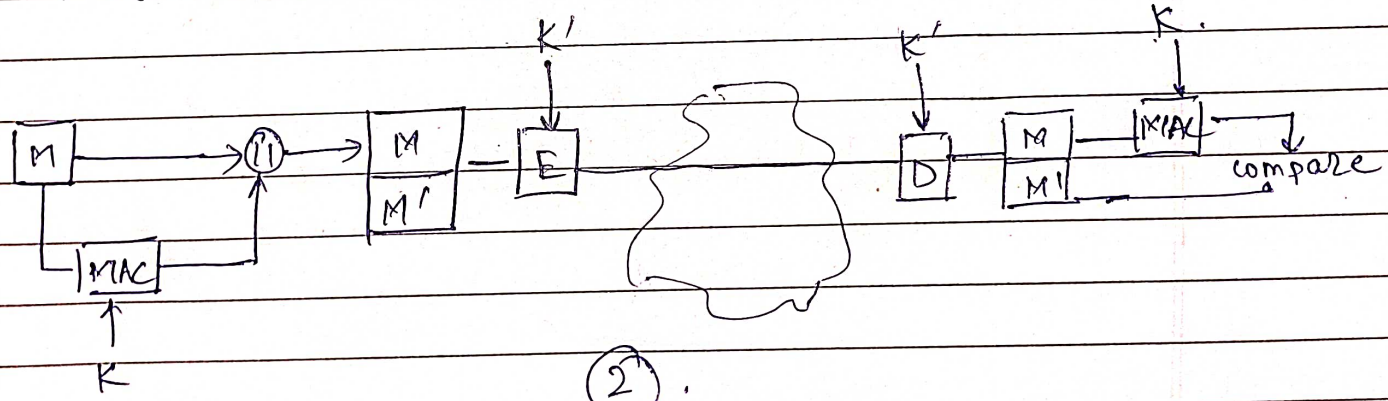
PAGE No.
DATE / /

* Types of MAC.

① MAC without encryption: This model can provide authentication but not confidentiality so anyone can see the message.

② Internal Server Error Code: Sender encrypts the content before sending it through network for confidentiality. Thus provides both.

③ External Error Code: First apply MAC on the encrypted message 'c' and compare it with received MAC value on the recieves's side and then decrypt 'c' if they both are same. else we simply discard the content recieved.



(2).