- National institute of standards & Technology (NIST)
- Internet Society (ISOC)
- International Telecommunication Union (ITU-T)
- International Organization of Standardization (ISO)

Chapter 2 : Principles and Practice

* Symmetric Encryption: technique for providing confidentiality for transmitted or stored data. aka single key encryption

  Requirements: need strong encryption algorithm, sender and receiver must have obtained copies of secret key securely and must keep them secure.

* Symmetric Encryption can be attacked by cryptanalytic attacks exploits characteristics of algorithm to deduce key or specific plaintext compromise all encryption performed by that key.

  Brute force attack to try all possible combination of keys to decipher text

* DES algorithm the key size is 56-bit which is inadequate.

* Triple DES algorithm works same as DES but uses 2,3 unique keys , 168 key length overcomes vulnerability to brute force, is sluggish in software uses 64 bit block size.

* Security issues:

- applied to a unit of data larger than 64-bit | 128-bit bloc
- ECB (electronic code book) simplest approach to multiple block encryption, each block encrypted using same key regularities in plaintext can be exploited.
- Modes of operation, techniques developed to increase security of symmetric encryption for large sequence to overcome weakness of ECB.

* Block ~~and Stream~~ ciphers - process one input block at a time and produces an output block, can reuse keys

* Stream cipher - processes input elements continously, produces output one element at a time, faster and use less code, encrypts one byte at a time

* Pseudorandom stream - unpredicable without knowledge of input key

* Message authentication protects against active attacks, verifies received message is authentic - (content not altered, authentic source, timely & correct sequence), convential encryption - (sender & receiver share key)

* Message encryption does not provide message authentication

* can have authentication and confidentiality by encrypting message & its authentication tag.

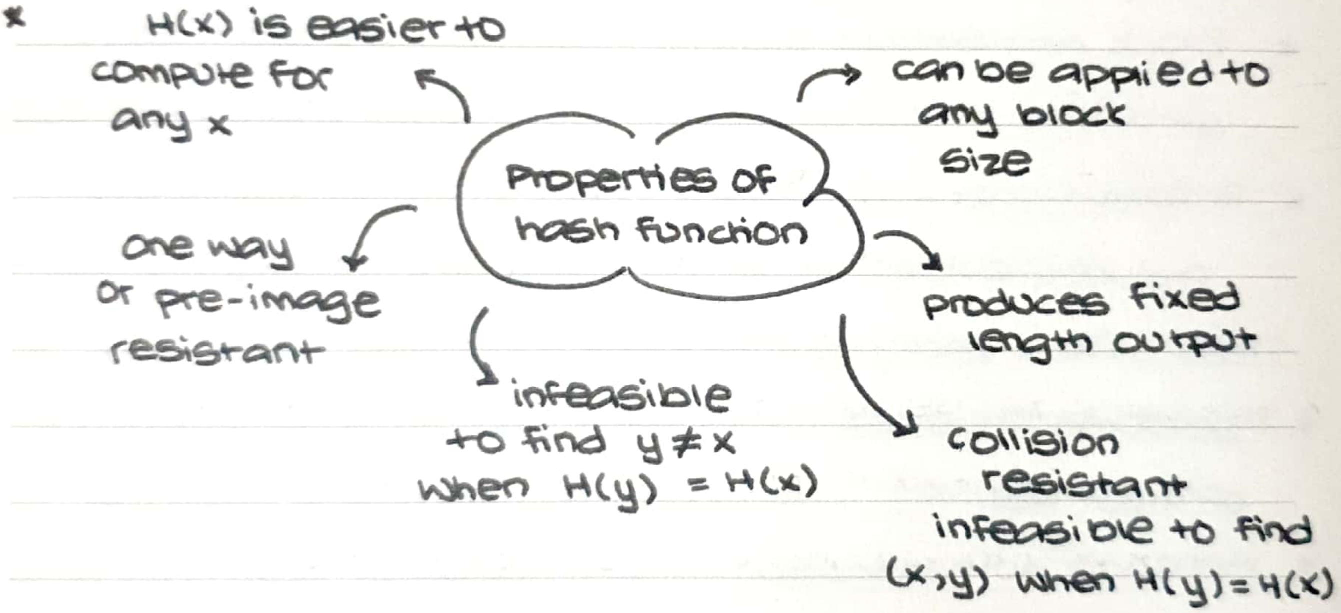
* Message encryption and authentication are seperate functions.

* Message authentication without encryption preferrable

- same message broadcast to multiple destinations
- exchange when one side has heavy load and cannot waste time decrypting
- authentication is an attractive service

* MAC - message authentication code

*



Properties of hash function

H(x) is easier to compute for any x

can be applied to any block size

one way or pre-image resistant

produces fixed length output

infeasible to find $y \neq x$ when $H(y) = H(x)$

collision resistant infeasible to find (x,y) when H(y)=H(x)

* Hash function can be attacked through cryptanalysis or brute force - strength depends upon length of hash code produced.

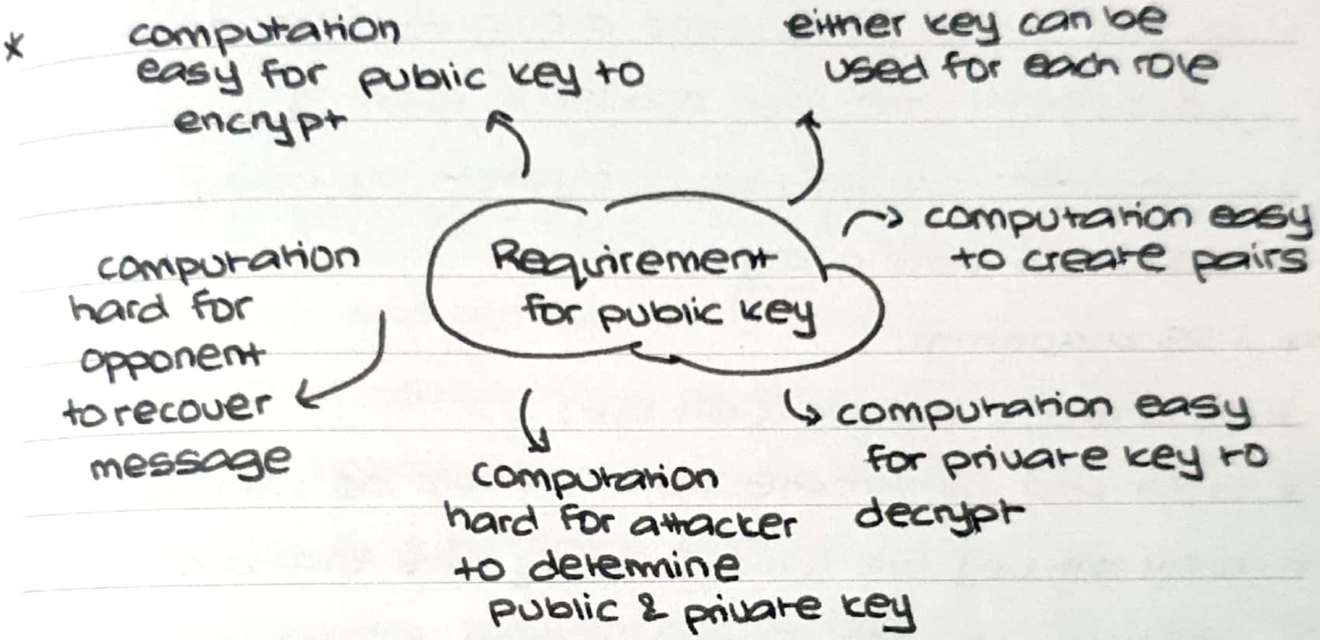
* Additional hash application :

- Passwords: hash of password stored in OS
- Intrusion detection: store H(F) for each file on a system and secure hash values.

* Public key encryption uses two keys public and private, public made public to be used by others.

* user encrypts data using own private key anyone knowing corresponding public key can decrypt data.

*



Requirement for public key

computation easy for public key to encrypt

either key can be used for each role

computation easy to create pairs

computation easy for private key to decrypt

computation hard for attacker to determine public & private key

computation hard for opponent to recover message

* Examples of ~~ass~~ asymmetric encryption algorithms RSA, Diffie-Hellman, DSS and ECC

* Digital signatures are data dependent bit pattern to verify origin authentication, data integrity.

* Digital envelope is created through random symmetric key, receiver's public key ~~and~~ making encrypted key with encrypted message.

* Transmitted data is encrypted often then stored data.

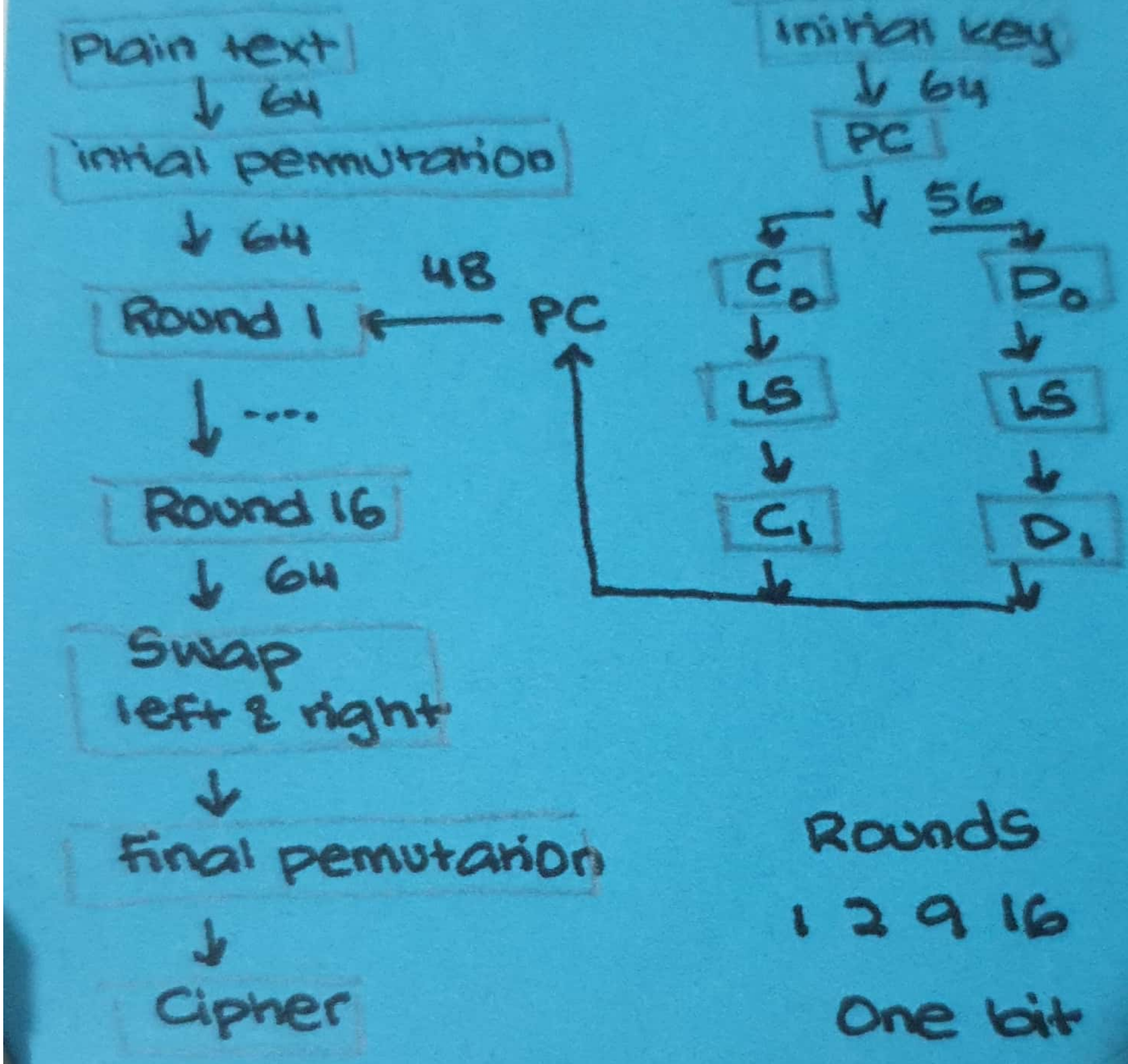
* To encrypt stored data can use commercially available encryption package, back-end appliance, library

based tape encryption and background laptop / PC data encryption.

- * To generate a digital signature we give message and private key to the algorithm.
- * Digital signature is send along with message to authenticate the sender's identity.
- * Signature can only be decrypted by using sender's public key.

=> DES algorithm:

- * is a block cipher (64 bit)
- * input and output are both 64 bit as well as key.
- * in 64 bit key we have 8 parity bits that are removed we get (56 bit) - every 8th bit is discarded. key is then rearranged by permutation
- * Half the key, each half left shifted by 1 bit when round number is 1, 2, 9, 16 otherwise by 2 bit
- * compression permutation is applied to the key to make it 48 bit from 56 bit
- * Plain text (64 bit) is send to initial permutation -> Rounds + keys -> final permutation -> output
- * After round one the previous sub key is input to compute next keys same operations performed.

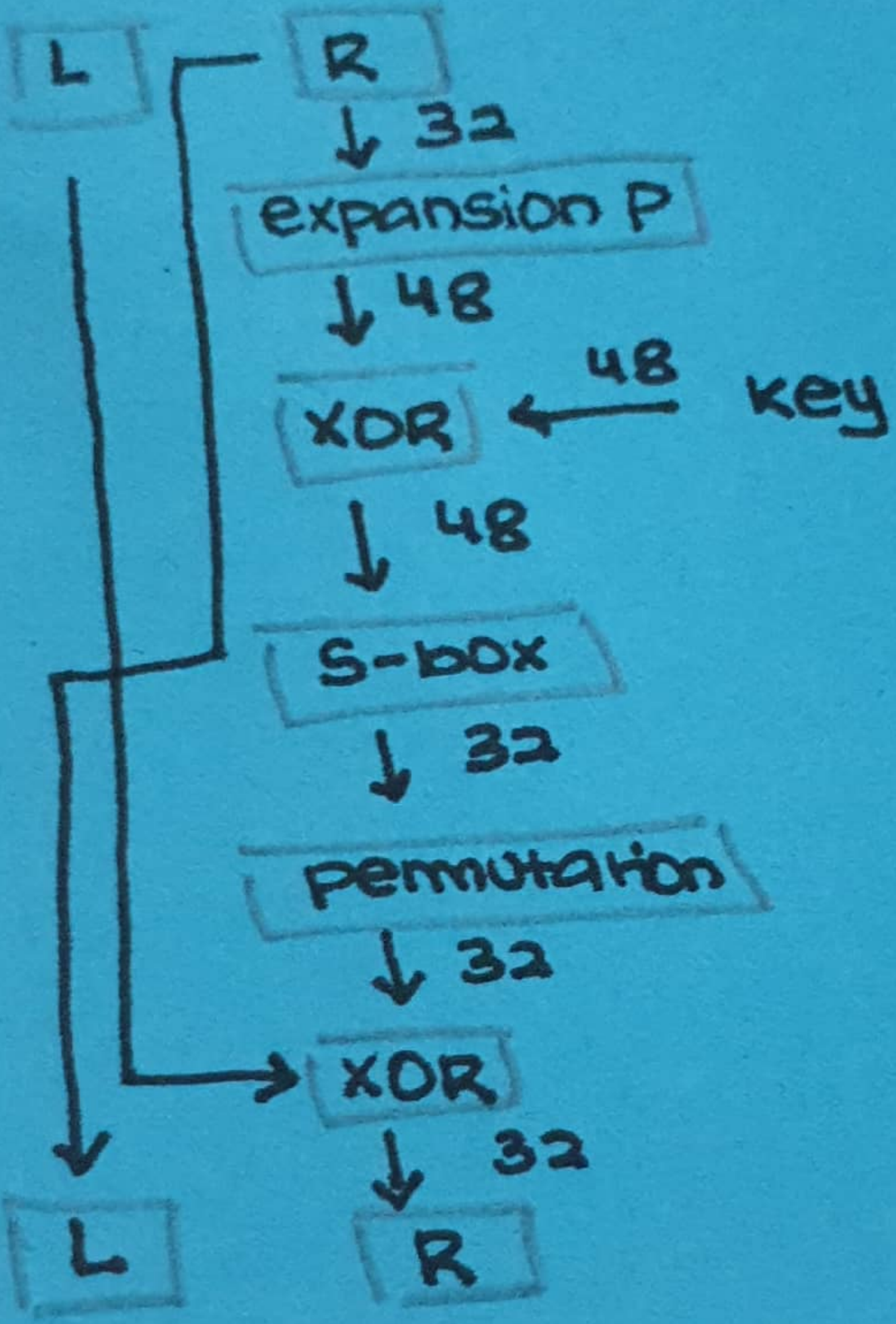


Rounds

1 2 9 16

One bit

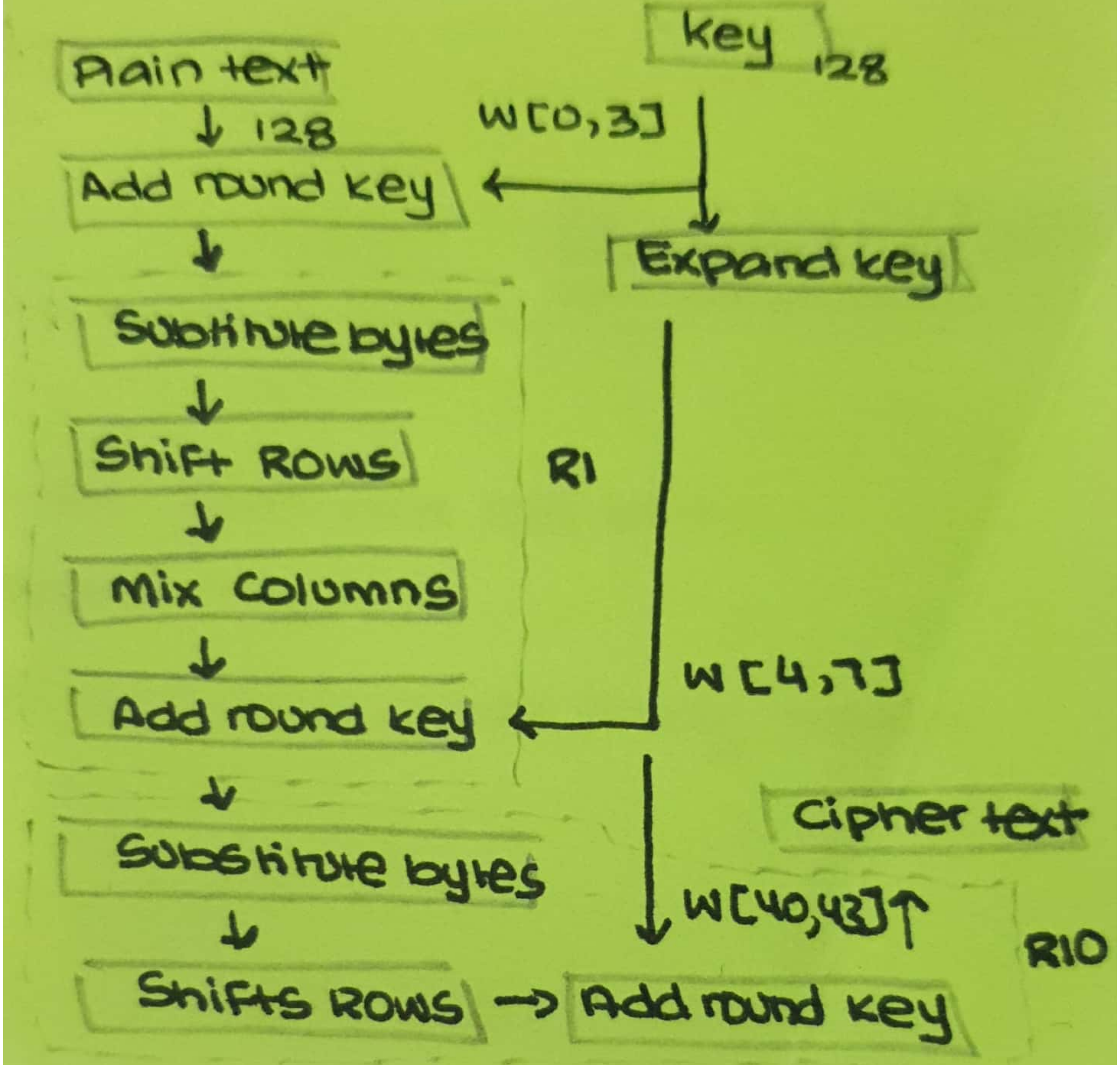
- Final permutation is called inverse initial permutation.
- Inside DES rounds the input 64 bits is broken into half, right side performs expansion permutation to increase to 48 bits to be XORed with 48 bit key
- The output is send to subsitution box to generate 32 bit, we have ● S-boxes, the farthest most bits are row and rest are column ①$\underbrace{0\,0\,0\,0}_{\text{column}}$⓪ ~rows get the value of binary row and column and get the value from the table.
- Output of subsitution box is send to permutation which is XORed from left side to form new right side while left side becomes previous right side
- After round 16 left and right are swapped the output then send to final permutation
- To reduce computational overhead use operations easy to implement.
- Min 12 rounds needed to provide security others are just saftey measures
- Each round is a fiestal cipher
- S-boxes generates confusion
- No swapper in last round
- Decryption of DES uses same algorithm subkeys application reverse initial & final permutation reversed.
- key length makes DES not secure



Working of Rounds in DES

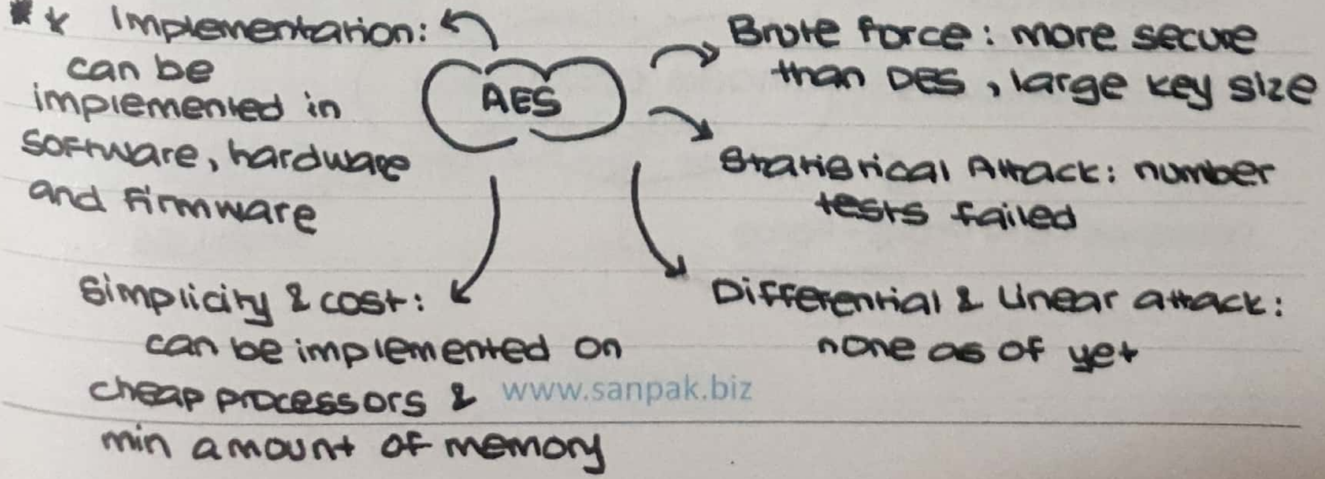
=> AES Encryption :

- Plain text and cipher text both are 128 bits.
- AES has only 10 rounds
- 128 bits is represents in a 4x4 input array where 1 cell is 1 byte.
- 1 word = 32 bits
- Add round key performs simple XOR
- Plain text is send to add round key to be XOR with key (128 bit) represented in an ~~state~~ array.
- key is send to an expand key function that expands key column wise and makes it from 4 words to 44 words.
- use a 4 word key each time for add round key
- State array is used to represent intermediate results between rounds. (4 x 4) $S_{0,1}$ byte word
- Function is Round 1 .. 9 are substitute bytes, shift rows, mix columns, add round key. in Round 10 mix columns does not occur.
- Sub bytes take input state array and S-box (16x16) output is 4x4, each cell in state array is divided into ~~2~~ 4 bits, First 4 indicates row whereas next 4 indicates column of S-box table


Plain text
128
Key
128
W[0,3]
Add round key
Expand key
Subtitute bytes
Shift Rows
R1
Mix Columns
W[4,7]
Add round key
Cipher text
Substitute bytes
W[40,43]
R10
Shifts Rows
Add round key

represent value in S table with 8 bit and replace it in the cell.

- * Shif rows, depends on row number. That ~~bit~~ byte will be shifted circularly to the left
- * Mix columns takes two inputs constant 4x4 input and 4x4 output of Shift rows. One column x constant input = column output
- * Add round key each column is ~~round~~ XORed with respective ~~[illegible]~~ column of key
- * Selected because of its security, cost and implementation.
- * Non-fiestal cipher based on three versions 10, 12 and 14 rounds, with key size 128, 192 and 256 with round key always being 128.
- * Number of rounds = words in key block + 6
- * Sub bytes for encryption site while Invsubbytes for decryption bytes
- * If two bytes have same value transformation is same



* Implementation: can be implemented in Software, hardware and Firmware

AES

Brute force: more secure than DES, large key size

Statistical Attack: number tests failed

Simplicity & cost: can be implemented on cheap processors & min amount of memory

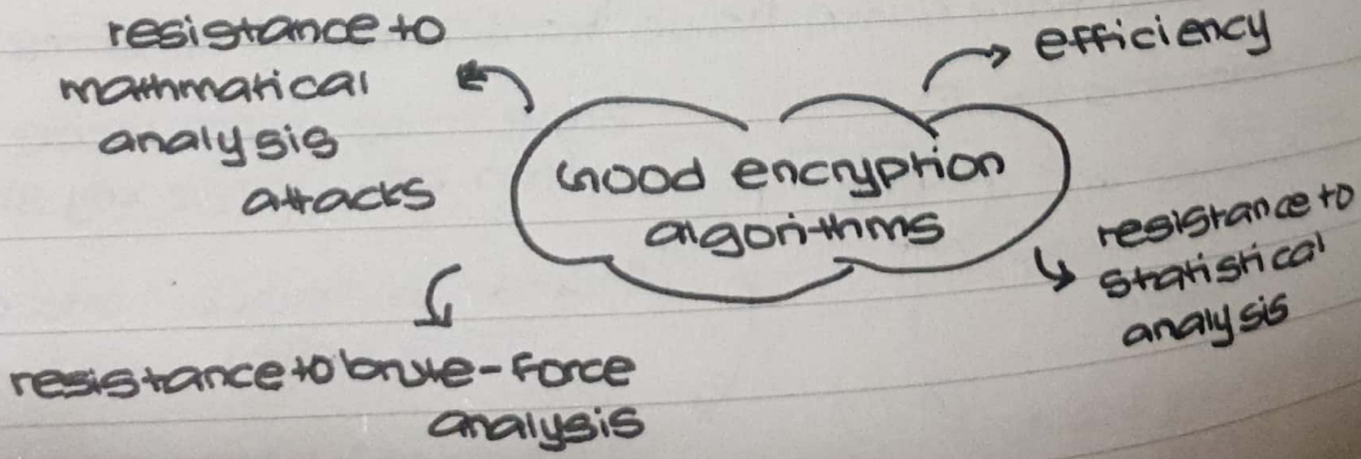
Differential & Linear attack: none as of yet

=> Fiestel Cipher:

* Plaintext Block is divided into two halves
* DES is based on fiestel cipher
* Left performed XOR with output of round function while right is left alone
* round function takes input key and right half
* Every round uses different keys derived from main key
* Before leaving round both sides are swapped L->R and R->L

* One-to-one mapping of characters maintain their character frequency, unreadable at First glance but can be decrypted easily.
* XOR encryption, ~~encrypt~~ XOR text with secret key to get cipher and plain text respectively.
* To keep encryption secure everything except keys are disclosed and they are reusable.

*



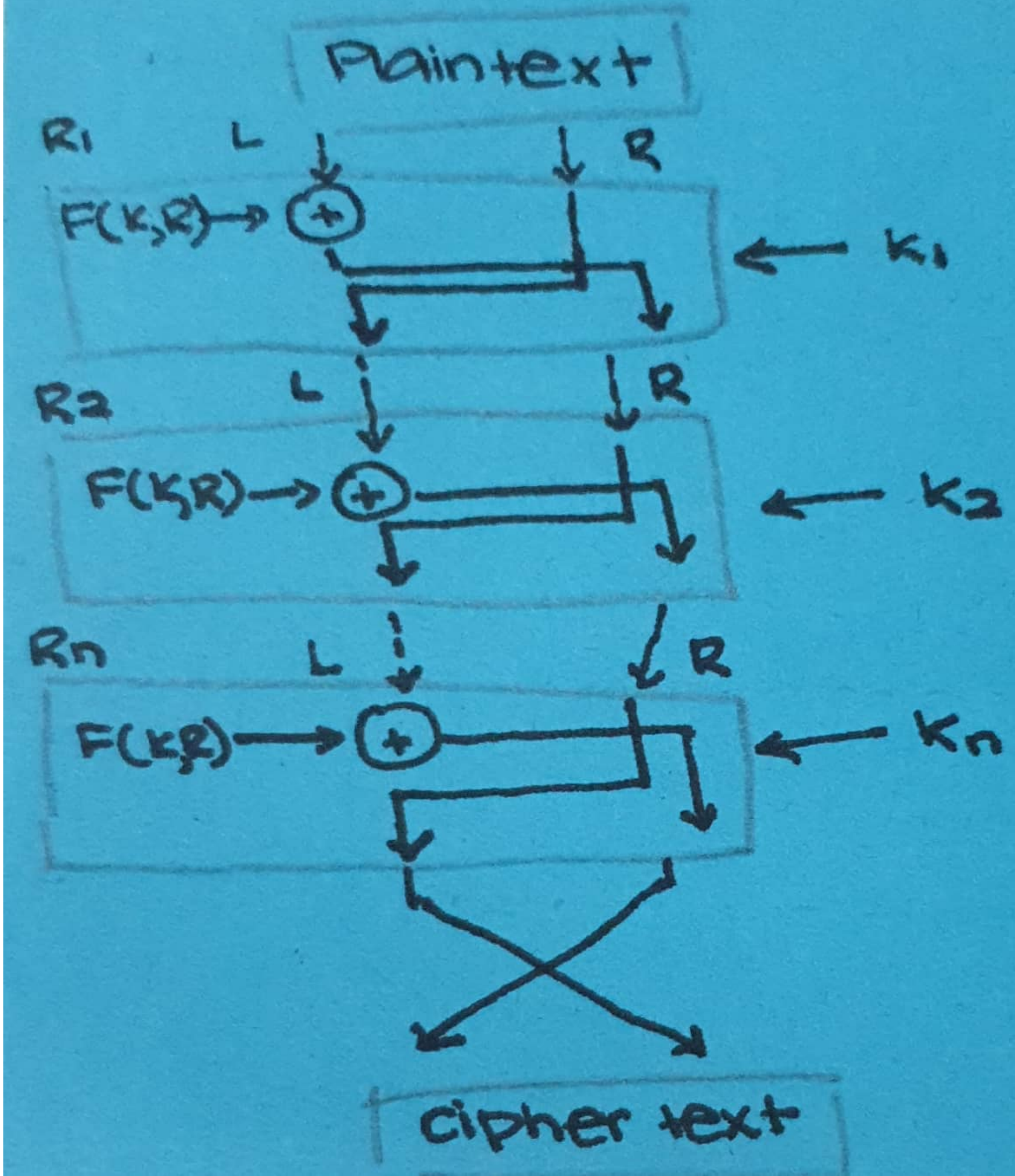
Good encryption algorithms

resistance to mathmatical analysis attacks

efficiency

resistance to statistical analysis

resistance to brute-Force analysis


Plaintext
R1
L
R
F(K,R)
K1
R2
L
R
F(K,R)
K2
Rn
L
R
F(K,R)
Kn
cipher text

- * Efficiency - easy to implement on hardware & software, consume moderate resources, time & space complexity with small constant factor of input size.
- * Statistical Analysis - diffusion by using fixed no. of operations in fixed number of rounds, confusion by using different sub-keys in different rounds.
- * Brute Force Attack - strength depends on length of key and its operations, time complexity of brute force attack used as benchmark for other cryptanalysis attack.
- * chosen plaintext attacks $C = (M \oplus K)$

$$M \oplus C = M \oplus (M \oplus K)$$

$$M \oplus C = K$$

- * Mathmatical Attacks differential cryptanalysis, linear cryptanalysis and algebraic cryptanalysis
- * Cryptanalysis can be performed on DES algorithm by exploiting its S-boxes.
- * 2-DES encrypt each block with different key can be attacked by meet-in-the-middle.

=> modes of cipher

* devised to enchiper text of any size

* Modes

CFB - cipher feedback

ECB - electronic code book

CBC - cipher block chaining

CTR - counter

OFB - output feedback

* ECB

not recommended for more than one block

patterns at block level preserved

block independency can allow exchanging some cipher without knowing the key

can use parallel processing

implemented for database storing

* CBC

patterns at block level not reserved

no parallel processing

plaintext XORed with prev ciphertext before encryption

IV should be known by sender & receiver

cannot use for random-access records

used for authentication

use different IV to avoid text encryption to be same for same text

* CFB

- does not need to wait for large block of data before encryption
- use DES/AES with smaller block size
- does not need padding size of r chosen to fit data unit
- less efficient than ECB/CBC
- stream cipher

applies encryption function for each r.

* OFB

- stream cipher
- each bit of cipher text independent of prev bit
- patterns not preserved

* CTR

- n-bits blocks independent of each other, depends on counter
- for application of fast encryption speed
- no feedback, pseudorandomness in key acheived by counter
- can encrypt random access files
- stream cipher