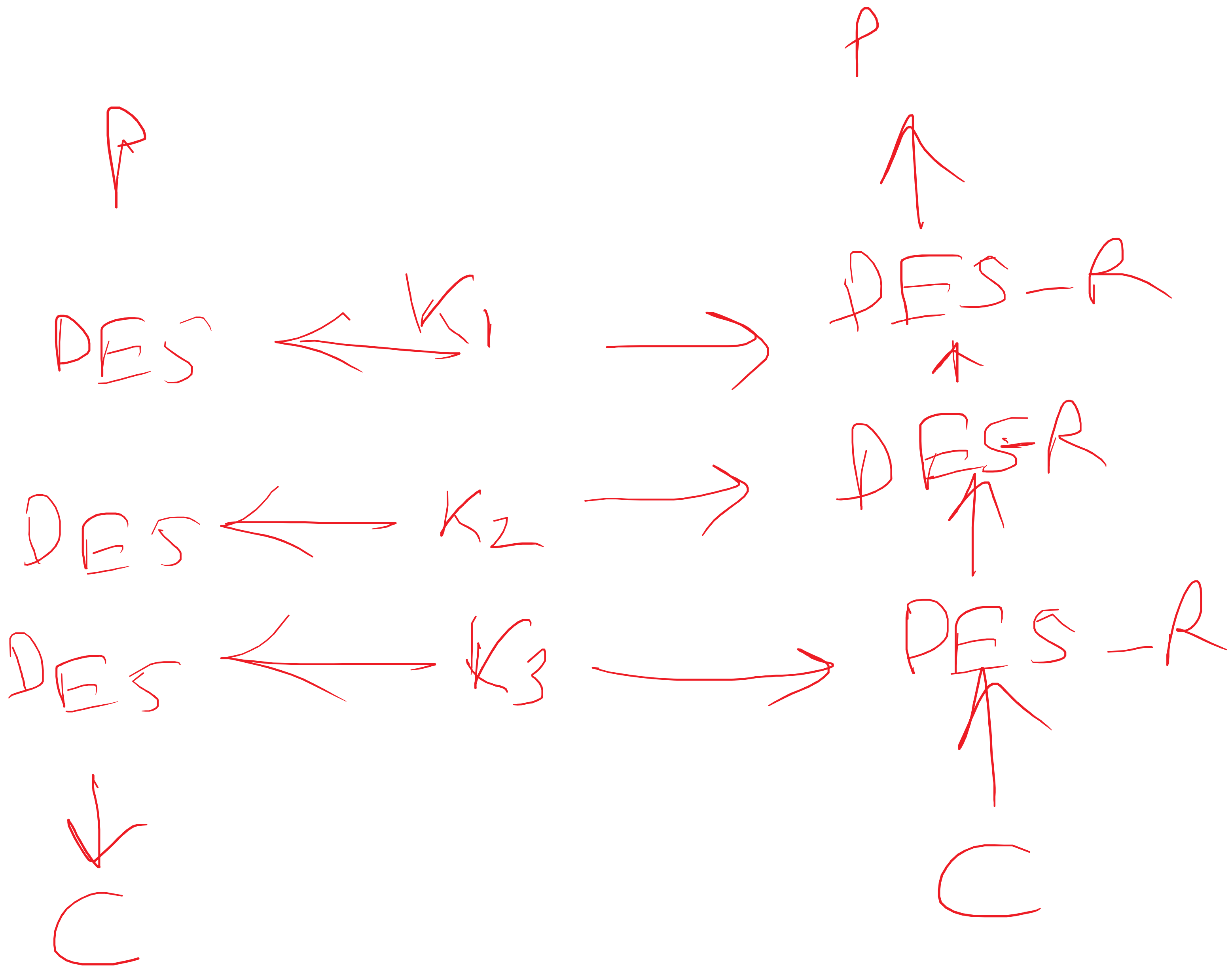
3



Blowfish: 32 bit Processor

1993

- fast ciphering than DES

variable key size

- compact, have a variable key size
  less memory up to 448 bits

- suitable for application where key does not change frequently

- Summetric: same key is used both side

- 16 rounds

- Block cipher: 64 bit plaintext

- fiestal structure (16 Rounds)

· Secure : Variable Key Length
(32 bit to 448 bit)

Suppose key size = 448 (bit)

All Possible key = 2^{448}

P-Array : P_1 - P_{18} each value is 32 bit long.
(subKey)

Initially Initialize with fixed strings.

S-Box : four S-Box are used here.

each S-Box have 256 entry.

↓

(each entry is 32 bit)

S_1-Box : S_0 S_1 ------ S_{255}

S_2-Box: S_0 S_1 ------ S_{255}

S_3-Box: S_0 S_1 ----- S_{255}

S_4-Box: S_8 S_1 ---- S_{255}

each 32 bit value

↓

there 32 bit is written in Hexa code

↓

Exa: "243Fab18"

Diagram of S-Box (value)

P-Array & S-Box values are 32 bit long and represented in Hexa code using 8 Hexa digit.

8 symbol of Hexa code

① Initialization of P-Array & S-Boxes:

P-Array ↓ $P_1 - P_{18}$ ↓ each 32

S-Boxes ↓ $S_0 - S_{255}$ ↓ each 32

both are initialized with a fixed string and string is Hexadecimal digit of π(3.14----)

$\pi = 3.2\,4\,3FGA\,88\underset{||}{}\,85A\,3\,0\,8D\,3\underset{||}{1}\,31\,98A$ ----

$$\pi = 3.\underbrace{243F6A88}_{P_1}\underbrace{85A308D3}_{P_2}\underbrace{13198A}_{P_3}\;----$$

all P-Array values are initialized with π,

after that next π values are used to initialize the all S-Box in same manner.

$$\underset{\text{Block (32bit)}}{\text{Total}} = \underset{\displaystyle\downarrow}{18}\,(\text{P-arry}) + \underset{\displaystyle\downarrow}{4 \times 256}\,(\text{S-Box}) = 1042 \text{ Block}$$

$$\underset{\text{(Hexa digit)}}{\text{Total}} = 18 \times 8 + 4 \times 256 \times 8 = 1042 \times 8 = 8336$$

$$\rightarrow (32, 64, ----\, 320 \,--\, 448)$$

→ (32,

② subkey generation: $K = (448 \text{ bit Long}) = 32 \times 14$

$K_1 --- K_{14}$

$P_1 = P_1 \oplus K_1$ (first 32 bit)

$P_2 = P_2 \oplus K_2$ (next 32 bit)

⋮

$P_{14} = P_{14} \oplus K_{14}$ (Last 32 bit)

$P_{15} = P_{15} \oplus K_1$ (first 32 bit)

⋮

$P_{18} = P_{18} \oplus K_4$ (fourth 32 bit)

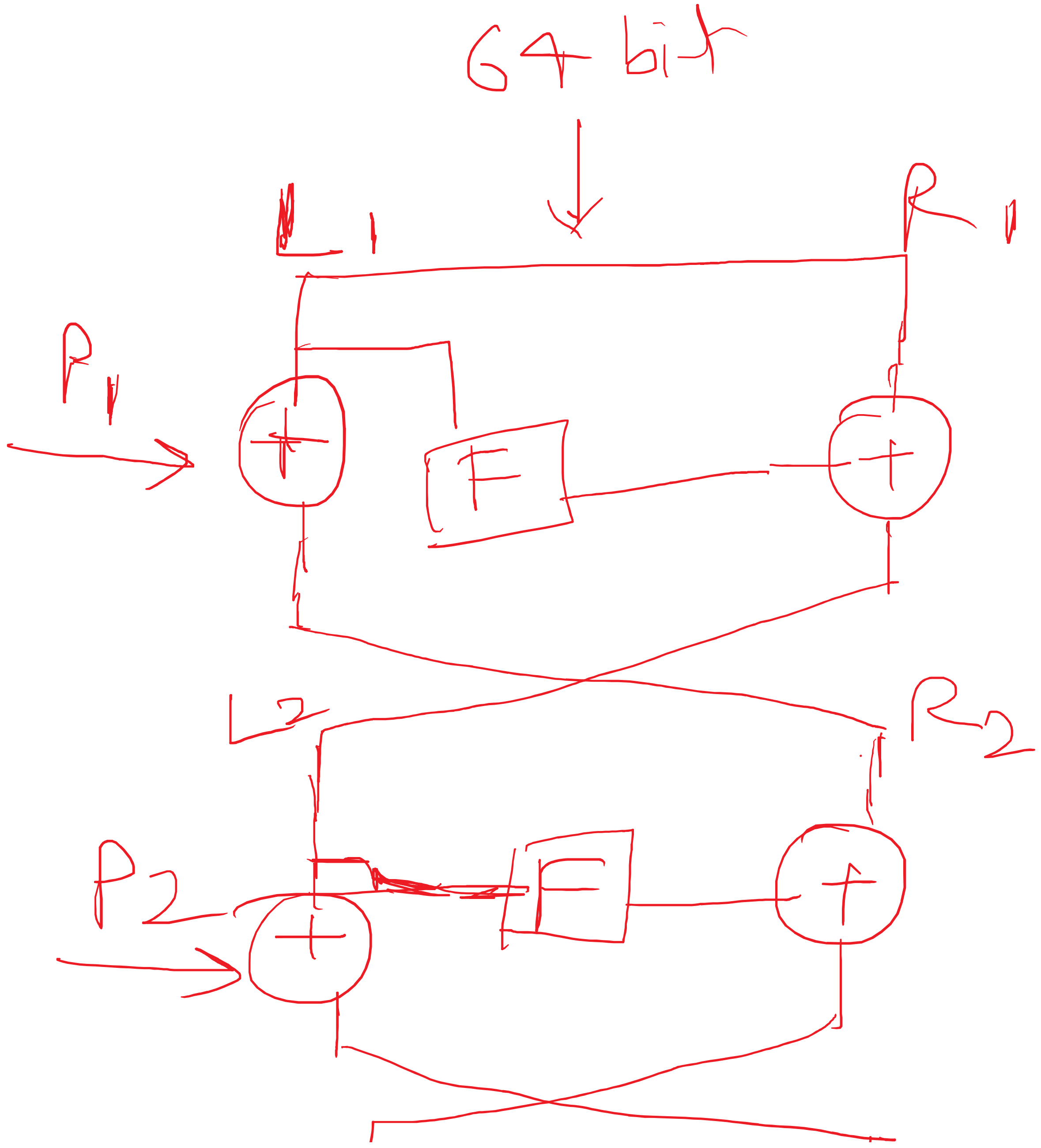
↑

suppose K size is 320 bit then

We use K_1 to K_{10} then K_1 to K_8

③ Encryption Algorithm: plaintext(64), key

(i) Divide plaintext into two Block L & R of equal size.
(each is 32 bit)



(ii) for $i=1$ to 16

$L_i = L_i \oplus P_i$

$R_i = F(L_i) \oplus R_i$

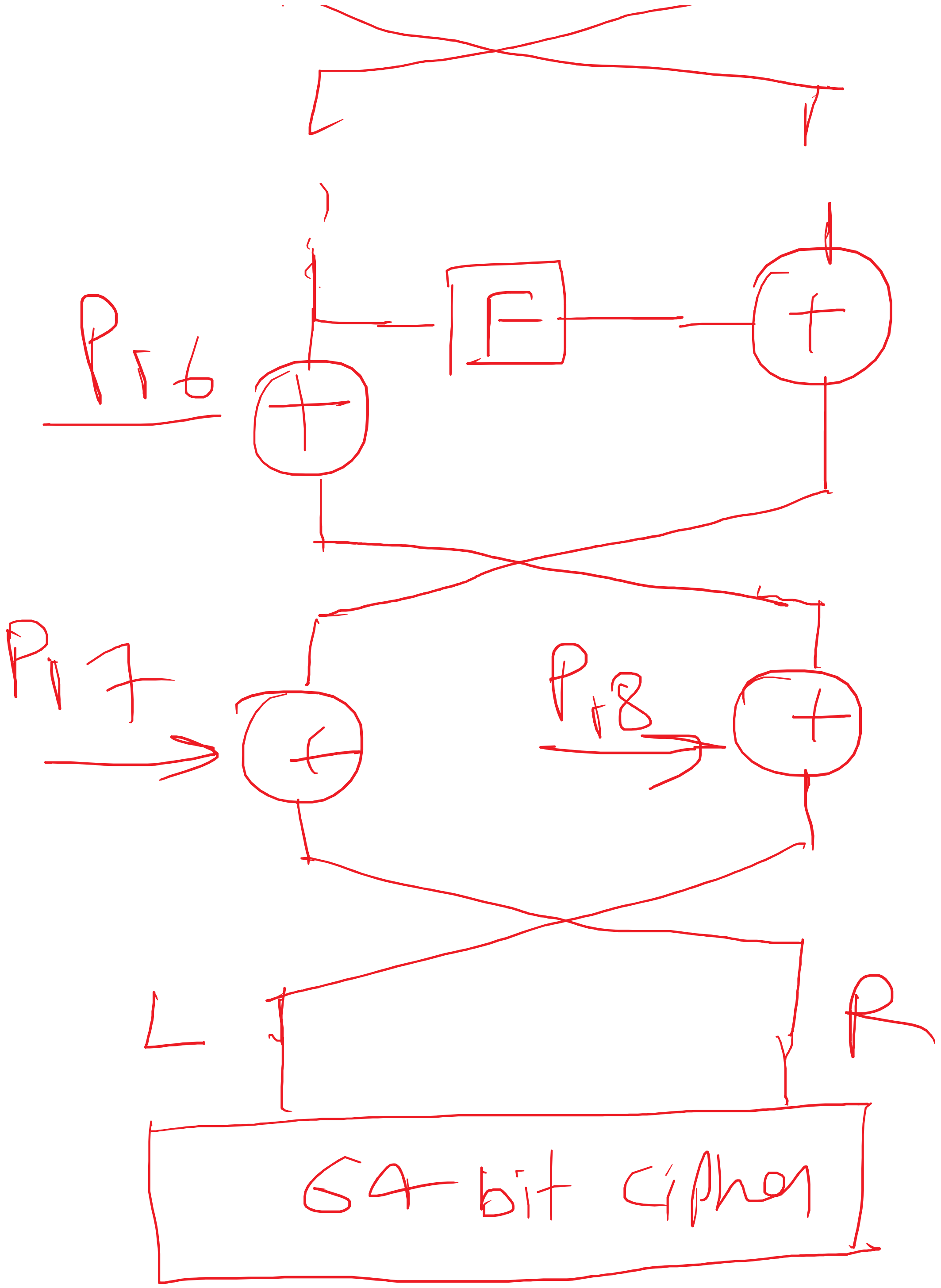
Swap(L_i, R_i)

(iii) undo last swap:

$R = R \oplus P_{17}$

$L = L \oplus P_{18}$

(v) concatenate L & R to get 64 bit ciphertext.

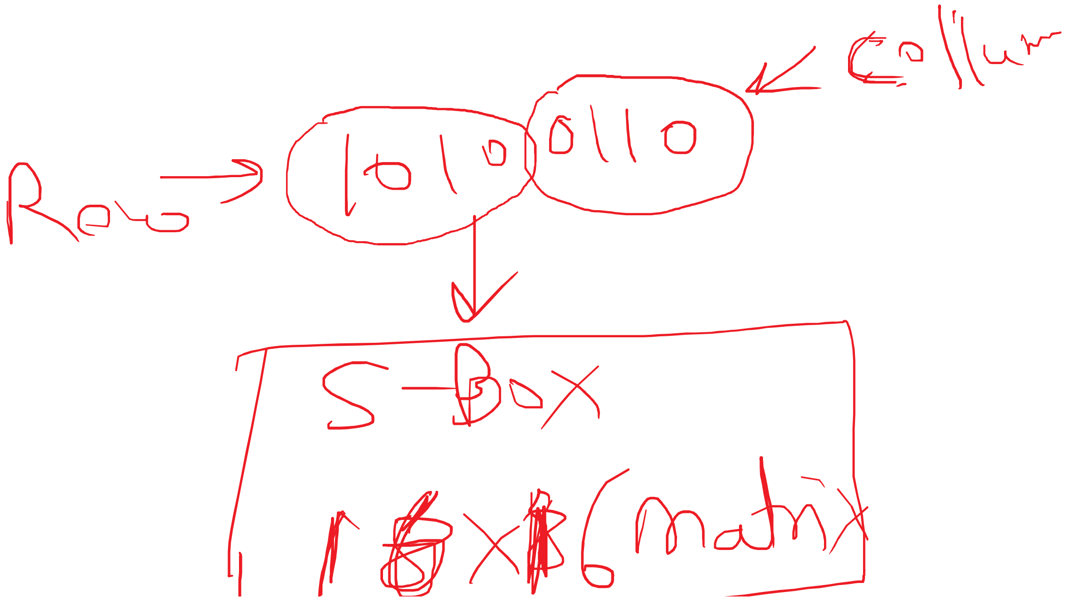


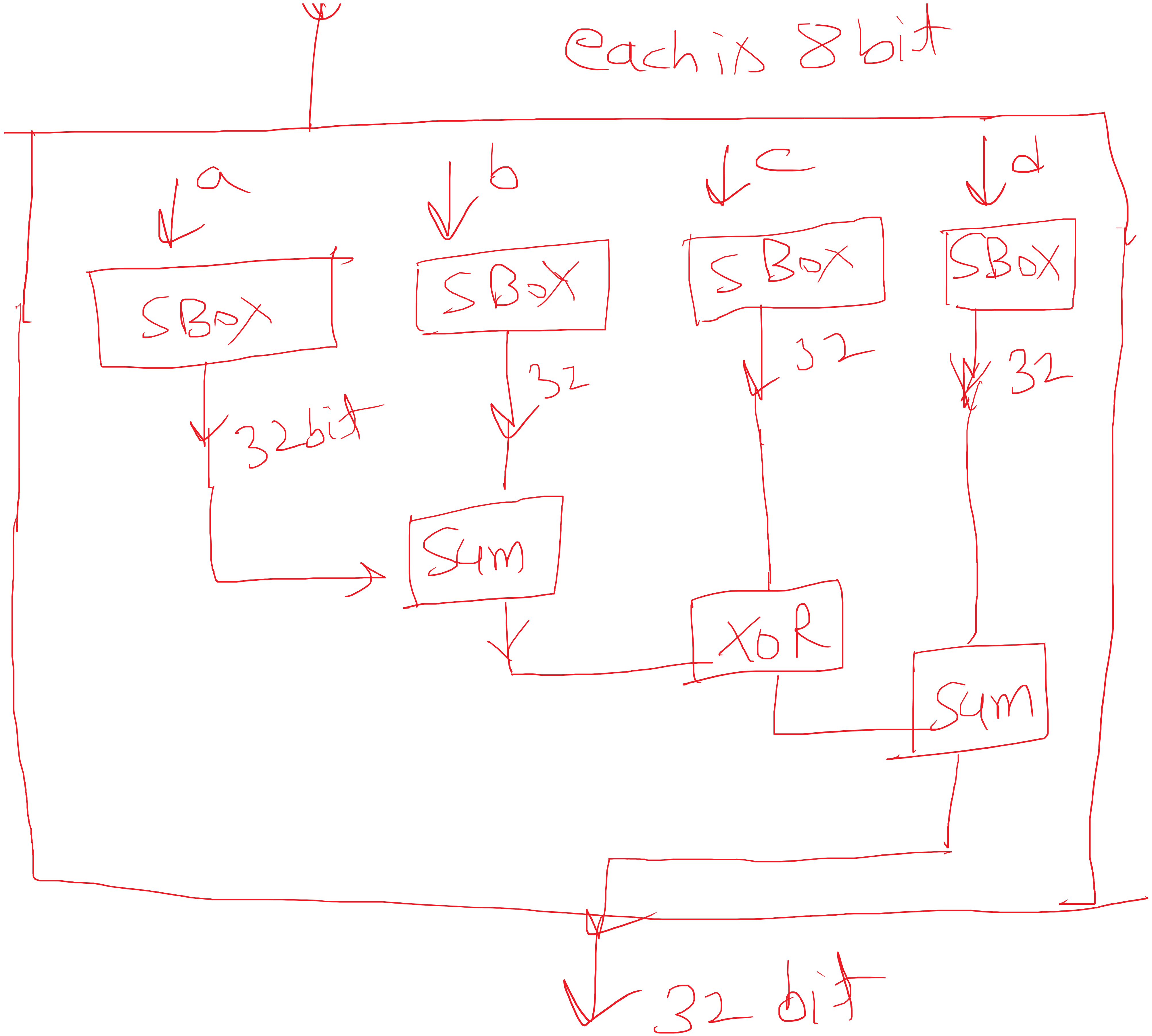
F = Function(Li)

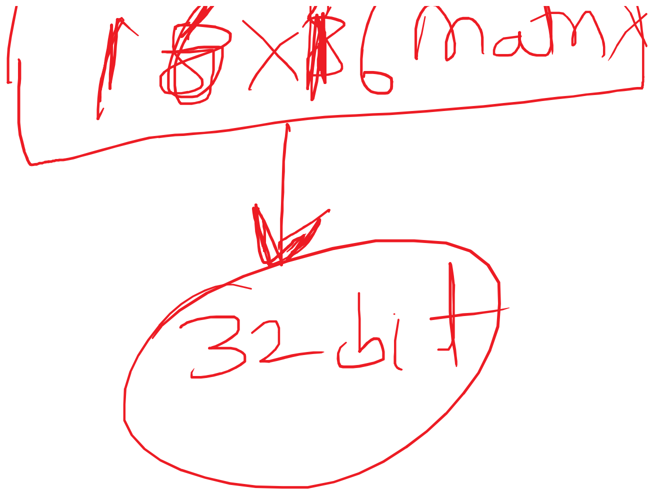
↑ 32 bit

↓ 32 bit (a, b, c, d)

each is 8 bit







$$F(XL) = ((S_1(a) + S_2(b)) \oplus S_3(c)) + S_4(d)$$

$f(X_L) = ((S_1(a) + S_2(b))$

④ Key generation Process:

(1) Initialize P-Array & S-Boxes using digit of π.

② update P-Array with given Key as described in step ② ----

$P_1 = P_1 \text{ XOR } K_1$

$P_2 = P_2 \text{ XOR } K_2$

$\vdots$

$P_{18} =$

③ An all zero string is encrypted with Blowfish Algo.

③ An all zero string is encrypted with blowfish

with subkey P_1 --- P_{18}.

④ P_1 & P_2 are Replaced by 64 bit output cipher of Step ③.

⑤ 64 bit cipher of step ③ is encrypted with updated subkey to replace P_3 & P_4 with cypher text going to be generated.

⑥ This Process is continue to replace all the P-array and all S-Boxes value in ordere

- This complex key generation Algo implies that for faster operation, the subkey should be pre computed and stored in cache for faster encryption.