SSL : Secure Socket layer
https : providing security / using encryption.

Foundation for Advancement of Science & Technology

Lecture-2

Risk Estimation

Assets / Vulnerability (How it can be attacked) / Threat / Attack

→ Probabilities of a threat:
prioritizing threats with higher probabilities.

Scarcity & choice → limited resource

- Info-Security: (IS)
→ investment $$, PKR

① WASP: not for profit organizatio

IS → Information System → people/user, Hardware, Software, Network Infrastructure.
(policies) procedure → data

*) Attack surface → # of vulnerabilities in my info-system / how many ways my system can be attacked.

→ Threat Modeling: (Modeling the (possible) threats)

asset → HW, SW, people, procedure

threats

protect

tools → policy, education, training, tech. controls

Microsoft
STRIDE:
S: Spoofing / faking identity
T: Tampering with data
R: Repudiation
I: Info disclosure
D: Denial of Servid
E: Elevation of privilage

Threat Model by Microsoft ← STRIDE → categories of threat

DDOS : Distributed Denial of Service.

→ Repudiation : ability to do an action and later being able to claim that you did not do it.

• D : Denial of Service :-

DNS → Domain Name Service
↓
allows access to a URL.
↓
ping / DNS

root DNS
↓
TLD DNS
↓
authoritative DNS
↓
cache server

→ send many requests so that the system (server) become down in 1 second (DOS)

→ Priority = case of exploitation * impact of exploitation

Lecture - 3

→ Security Design Principles :-

Principle of...

1) least privilege
2) seperation of privilege
3) fail-safe defaults
4) complete mediation
5) economy of mechanisms
6) least common mechanism
7) psychological acceptability

4) → [cache] ---> security vs performance

trade-off (inversely proportional)

(mora) coarse ↖ ↗ fine (baariqi)

"granularity of privileges"

1) provide bare minimum privileges to a program or user to function properly. "temporary elevation" should be relinquished immediately.

2) access should not be granted on basis of a single condition. e.g 2 persons sign on cheque, password / login + OTC (fin. transaction).

3) the default config. of a system should have a conservative approach. e.g default access to an obj is 'none'.

examples → ACL (access control list)

→ firewall rules

4) [user] → [OS] → [object] (instead of 1 time check, OS should always check whenever the obj is accessed)

5) simplicity in design & implementation of security measures

=> a "simple security framework" provides fewer errors; development, testing & verification of security measures is easy; less assumption.

6) in shared systems w/ multiple users, mechanisms allowing resources to be shared by >1 user should be minimized. ("seperated channel" for users, 'seperate of

pg 454, ch. 8, Wirman/Mattord
↳ 6th edition.

Foundation for Advancement of Science & Technology

network resources :) to-do

7) security mechanism should not make the resources difficult to access. • UI should be well-designed & intuitive • ordinary users should be kept in mind when planning.

→ Cryptanalysis : the process of obtainning the plain text msg from a ciphertext msg w/out knowing the keys used to perform encryption

→ Cryptography : the process of making & using codes to secure information.

→ Cryptology : the field of science that encompasses cryptography & cryptanalysis.

→ Types of cryptographic functions :

i) => secret/symmetric key cryptographic functions
• uses 1 key • fast computation

ii) => public/asymmetric key (PKC)
• uses 2 keys • slow computation

iii) => Hash functions
• NO keys • very fast computation.

plain text → encryption algo → cipher text

ciphertext → decryption algo → plain text.

Foundation for Advancement of Science & Technology

i) also called "conventional cryptography". sender & receiver must both know the secret key. uses techniques like 'confusion' & 'diffusion' to encrypt/decrypt data.

Sender → Encrypt → Decrypt → Receiver.

→ Symmetric encryption uses:
- transmitting over a secure channel
- secure storage on insecure media

→ Authentication:
- Strong authentication prove the knowledge of a secret w/out revealing it.

→ Integrity check:
- checksum vs cryptographic checksum
- Message Authorization code (MAC)

→ substitution cipher (monoalphabetic) ↗ caesar's cipher.
↳ using single alphabet to create a mapping.

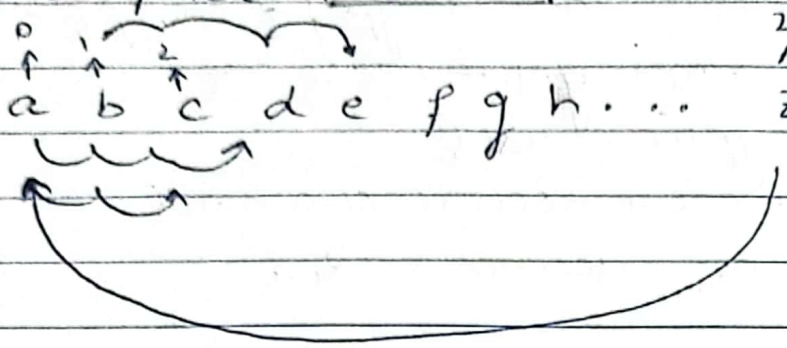
plain text ← jeejanab → "+3" → mhhmdqde → ciphertext

Foundation for Advancement of Science & Technology

Lecture-4

→ type of monoalphabetic

→ caesar's cipher: [+3]



0 ~ 25

z+3 = c

25+3 mod 26 = 2

↳ 26 alphabets

↳ intro to db → plain text

↳ lqwur wr gc (encrypted using caesar's cipher)

Encryption Algo → $C = E(3, p) = (p+3) \bmod 26$

ciphertext character

plain text character

Ex: abdf → plain text

degi → cipher text

Decryption Algo → $D = D(k, C) = (C-k) \bmod 26$

→ Func/

→ monoalphabetic cipher: substitution.

we dont have to follow any specific pattern. Using single alphabet to create a mapping.

Ex:

a → c	q → v	m → r	s → m	y → z
b → j	h → i	n → h	t → u	z → t
c → d	i → c	o → f	u → v	
d → b	j → o	p → g	v → w	
e → k	k → s	q → p	w → x	
f → a	l → n	r → n	x → y	

random mapping stored to encrypt a plain text.

plain text → zebra crossing
encrypted → tkjne dnfmmchq (cipher text)

- Generally, to decrypt this message with brute force, the time taken will be 26!.

- When not using brute force:

single letter words
I, a

frequency of 2-letter ← diagrams
words
an at if
of is to ok

e, t, i, a
12% 8% 7% 9%

We will try/use the letters with higher probabilities to decrypt the message.

→ Vigenére cipher (polyalphabetic)
Vigenére table (in slides)
Ex: keyword: Houghton
column
plain text → michigan technological university
row ← HOUGHTON HOUGHTONHOUGH TONHOUGHTO
cipher text

→ keyword consists of multiple alphabets that is why it is known as polyalphabetic

Foundation for Advancement of Science & Technology

Lecture - 5

Substitution techniques:

- One-time pad

Transposition techniques

- Rail-fence cipher
- Row transposition cipher

→ Rail Fence (with some depth level x) ← no. of rows which will be used

Method: ① Write the plaintext in diagonal form, based on the depth level x

② Read the text in rows, to get the ciphertext.

Ex: Plain text: neso academy is the best

depth: 2 (rows) columns (no. of letters): 20

(white spaces are not considered in this example)

N		S		A		A		E		Y		S		H		B	
	E		O		C		D		M		I		T		E		E

S	
	T

ciphertext: NSAAEYSHBSEOCDMITEET

Ex: depth = 3 (If depth = 3, total no. of letters / 2 = no. of letters in middle row)

ciphertext: TKVMHNYUEYUHAORC (letters = 16)

T				H				E				A			
	K		M		N		U		Y		H				
		V				Y				O		R		C	

T				K				V				M			
	H		N		Y		U		E		Y		U		H
		A								R				C	

Plain text: THANK YOU VERY MUCH

→ Row Transposition cipher:

- Writing plain text into rows & then reading of ciphertext from the columns.
- It works by rearranging the order of characters in the plain text, according to a specific pattern or key, & then writing the resulting cipher text in a grid format.

Ex:

plain text: winter is coming

key: h a c k => 2 3 1 4

2 3 1 4 (smallest in alpha numeric)

↓ columns

4 ~~rows~~

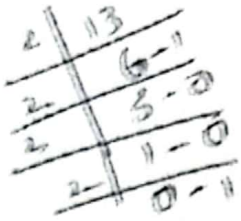
1	2	3	4
W	I	N	T
E	R	-	I
S	-	C	O
M	I	N	G

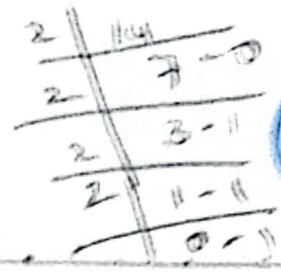
(length of key = no. of columns)

ciphertext: IR-IN-CNWES

MTIOG

(rowcipher.netlify.app) → to convert plain text to cipher text.





Ex: plain text: this life is short
key: quick
4 3 5 1 2 (cols = 5)

q	u	i	c	k
t	h	i	s	-
l	i	f	e	-
i	s	-	s	h
o	r	t	-	-

cipher text: ses_if_t__h_t
lioisr
(no. of char / cols = rows)
↳ for decryption

→ One-Time Pad: key length = plain text length
key is generated randomly
based on vernam cipher

PT [ ] Key [ ]

Take XOR to get CT

Ex: plain text: O A K "14" 0 1 1 1 0
key: S O N "18" 1 0 0 1 0
XOR 1 1 1 0 0
↳ 28-1-26=2
00010
↳ C

C O H

A : 0 0 0 0 0
O : 0 1 1 1 0
0 1 1 1 0 → 14 = O

K 0 1 0 0
↳ 18

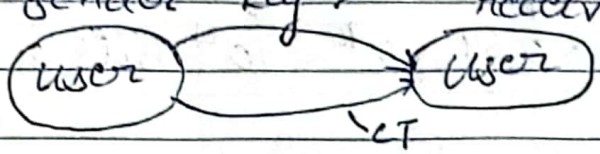
→ perfect secrecy" → CT provides no info of PT except length.

Lecture-6

Traditional Block Cipher Structure

- Stream cipher
- Block cipher — reversible / irreversible

Feistel cipher



*) To solve this, we define an algo which will be used to generate a key

→ The key should be random.

→ Stream cipher:

- One that encrypts a digital data stream one bit or one byte at a time. For example Vigenère cipher, Vernam cipher
- If the cryptographic key stream is random, then this cipher is unbreakable by any means other than acquiring the key stream.

→ Block cipher:

One in which a block of plain text is treated as a whole & used to produce a cipher text block of equal length. Typically, 64-bit or 128-bit.

- Most network based systems. Cryptographic applications make use of block ciphers → key: 11100001

Reversible Mapping

Plain Text	Cipher Text
00	11
01	10
10	00
11	01

e.g. 1-to-1 mapping

4 rows → 2×4 = 8 bits key length

Irreversible Mapping

PT	CT
00	11
01	10
10	01
11	01

PT → CT
(n-bits) (n-bits)

2^n possible different PT blocks

* when n = 4 ↗ no. of combinations.
  → CT length × rows = 4 bits × 16
* when n = 64
  64 bits * $2^{64} = 2^{70} \simeq 10^{21}$ bits

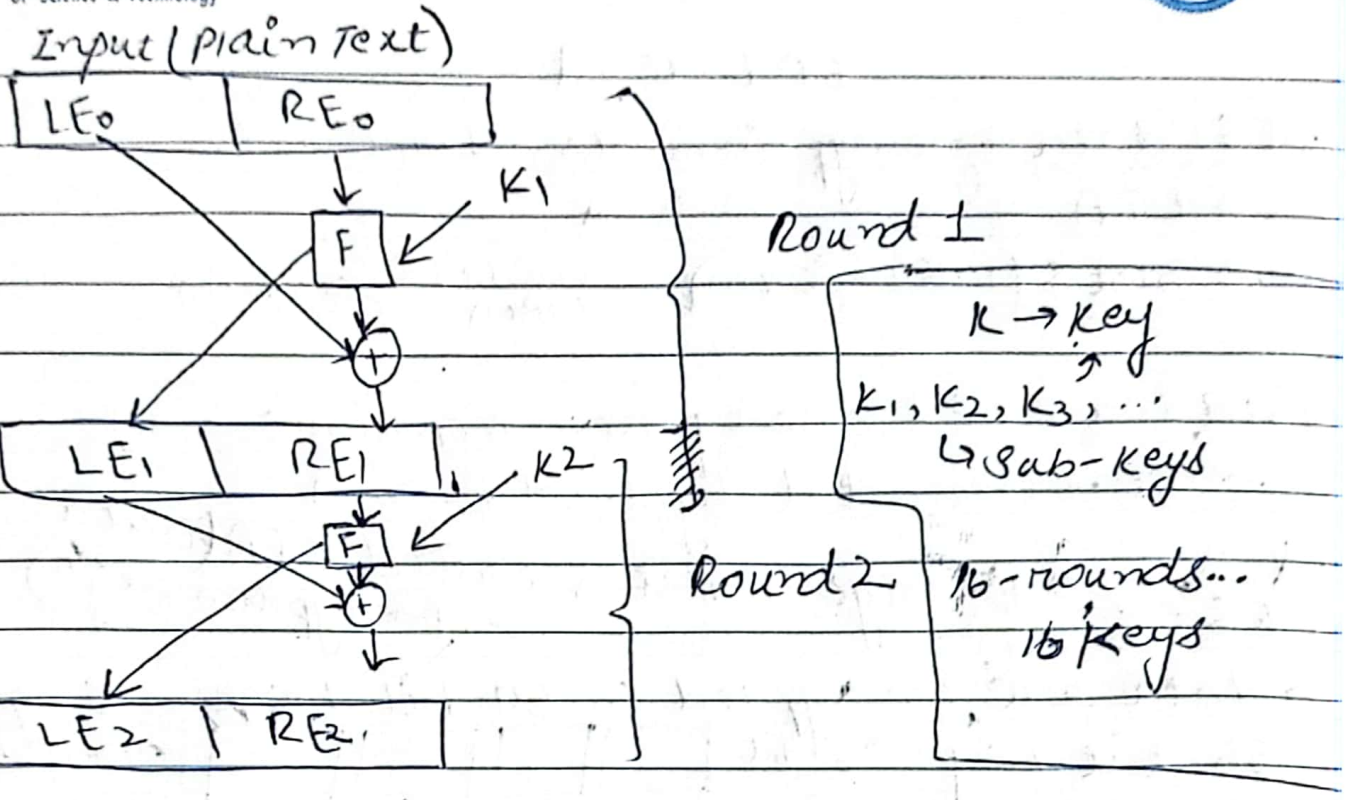
→ Feistel Cipher: / Structure / Network
Step 1: Split plain text into two halfs L_0 & R_0
Step 2: $E = f_s(R_0)$ — f_s → some algo, E → temp variable
Step 3: $L_1 = R_0$ and $R_1 = L_0$ XOR E
Step 4: Concatenate L_1 and R_1 => resulting in our ciphertext

∵ L_1 => first part of cipher text
∵ R_1 => second part of cipher text

Ex: Plain text: 011110 100001
(L_0 R_0)
= ~~101011~~ 010100 (E)

$L_1 = 100001$
$R_1 = 011110$ XOR 010100
$= 001010$
L_1 concatenate R_1 = 100001001010

7edition → Stallings
Crypto.....



The exact realization of a Feistal network depends on the choice of the following parameter of design features.

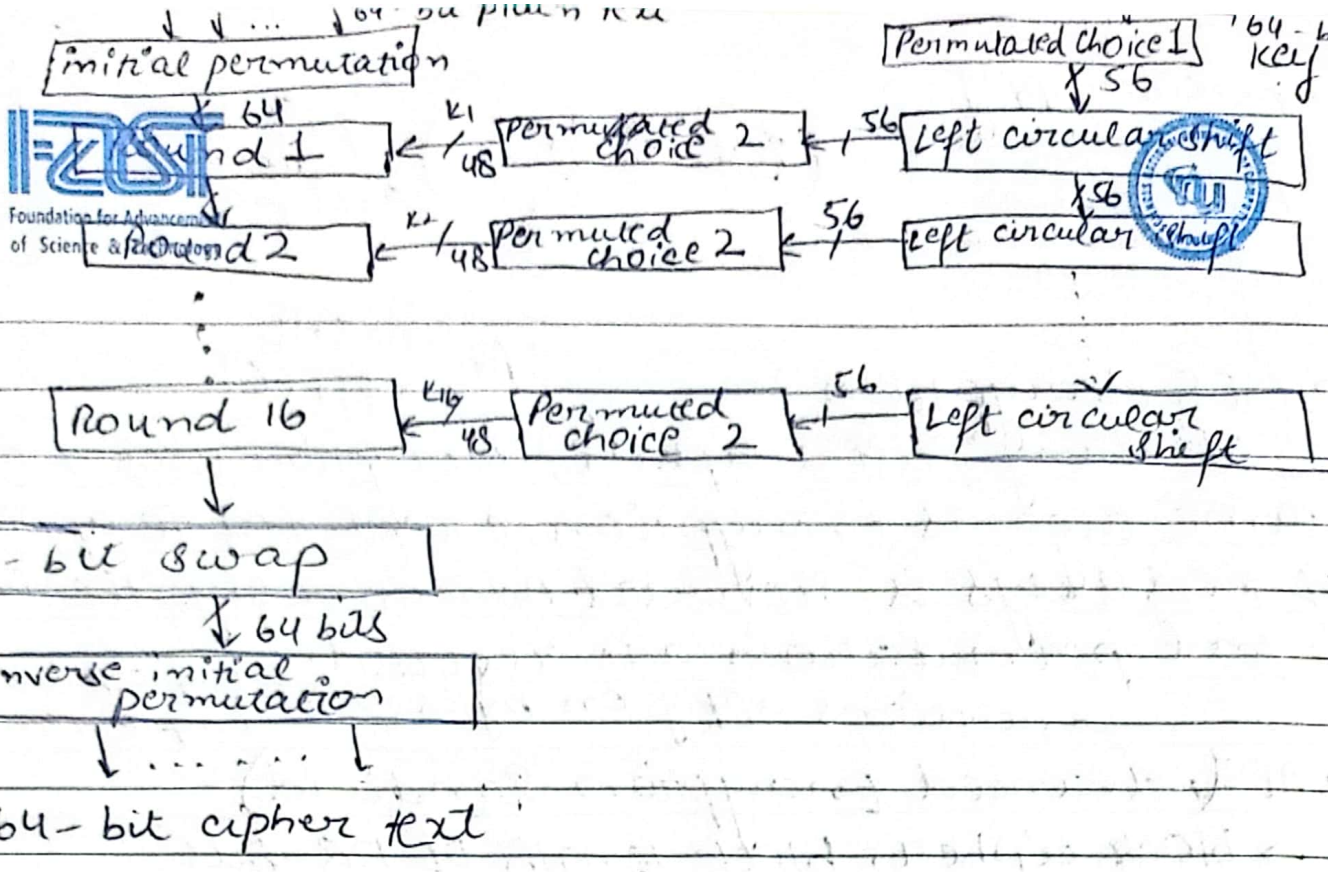
i) Block size
ii) key size
iii) # of rounds
iv) Subkey generation algo
v) Round function F

Lecture - 1

DES: data encryption standard
→ DEA
→ 3DES (ralph-form) LUCIFER (IBM) → Nylon

AES: advanced encryption technique
→ recommended by NIST in 2001.

- DES: → the algo in this is called DEA (Data Encryption algo)
→ data are encrypted in 64-bit blocks using a 56-bit key → 64-bit  every 8th bit
→ the algo transforms 64-bit input in a series of steps into a 64-bit output.
→ the same steps, with the same key, are used to reverse the encryption
~ 1999 ⇒ 3DES introduced by NIST (i.e, repeat DEA 3 times, using 2 or 3 different keys.
~ DES has the exact same structure of a feistal cipher, except for the initial & final permutation
→ key:
Initially, key is passed through a permutation fn. Then, for the each of the 16 rounds, a subkey (k_i) is produced by the combination of a left circular shift & a permutation. In each round, the permutation fn is same.



- DES/DEA (3 phases)

(1) 64-bit PT passes thru an initial permutation (IP) that rearranges the bits to produce the permuted bit

(2) Phase of 16 rounds of the same function, which involves both permutation & substitution functions.
~ the output of the last round (16^{th})... consists of a 64-bits that are a function of the input PT & the key.

(3) Left & right halves of the output are swapped to produce the "preoutput". Then, the preoutput is passed thru a permutation [IP^{-1}] that is the inverse of the initial permutation function, to get 64-bit CT.

function apply on bits

(Table 4.5) → Avg time for key exhaustive search.

→ DES decryption?

AVALANCHE EFFECT

as w/ any feistal cipher, decryption uses the same algo as encryption, except that the application of the subkeys is reversed. Additionally, the initial & final permutation are reversed.

→ functions apply on bytes => quicker software implementation.

- AES (Advanced Encryption Standard)
  - → block cipher, block length of 128-bits
  - → allows for 3 different key lengths: 128, 192, 256 bits.
  - → encryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, & 14 rounds for 256-bit keys
  - → except for the last round, in each case, all other rounds are identical.

Figure (6.1)

DES → chp 4
AES → chp 6

- Each round of processing includes one single byte based substitution step, a row wise permutation step, a column wise mixing step & the addition of the round key.

CH:04

Foundation for Advancement of Science & Technology

in plain text : X → original 64-bit block

∴ IP → initial permutation

Lecture-8

→ DES: (Data Encryption Standard)

DES-key size = 56 bits (64 bits, but 8-bits are used as parity check bits).

plain text

IP(x) = $L_0 R_0$ } - Phase-1 IP: initial permutation

$L_i = R_{i-1}$ } - Phase-2 repeats 16x

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

$y = IP^{-1}(R_{16} L_{16})$ } → Phase-3

↳ cipher text

IP =

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

→ This table specifies the input permutation on a 64-bit block

- The meaning is as follows
  The 1st bit of the output is taken from the 5th bit of the input, the 2nd bit from the 5th bit, & so on, with the last bit of the output taken from 7th bit of the input.
- This info is presented as a table for ease of presentation, it is a vector, not a matrix.

Foundation for Advancement of Science & Technology

Note that, as usual

$R_{16} = L_{15} \oplus f(R_{15}, K_{14})$

$L_{16} = R_{15}$

~ but they are switched in the pre output $\rightarrow (R_{16} L_{16})$

$IP^{-1}(R_{16} L_{16})$

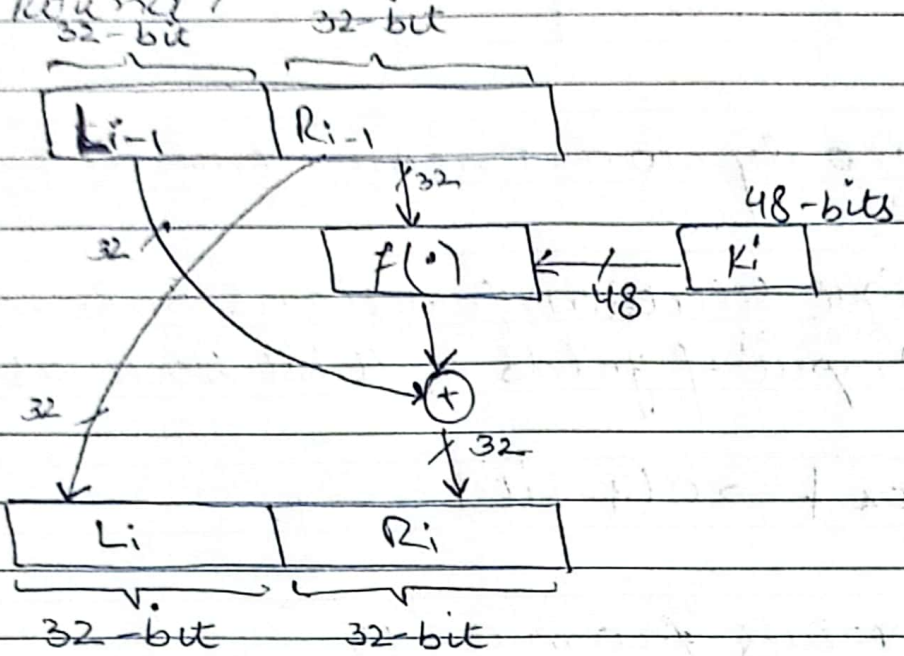
Final Permutation

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

Interpretation (similar to IP)

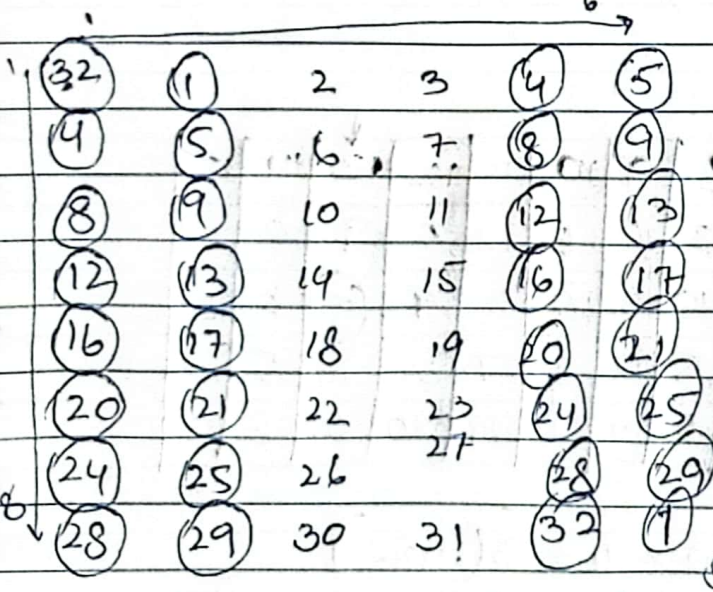
↳ output of final permutation has bit 40 of the preoutput block as it's 1st bit, but 8 as 2nd bit & so on, until bit 25 of the preoutput block is the last bit of the output.

DES Round i

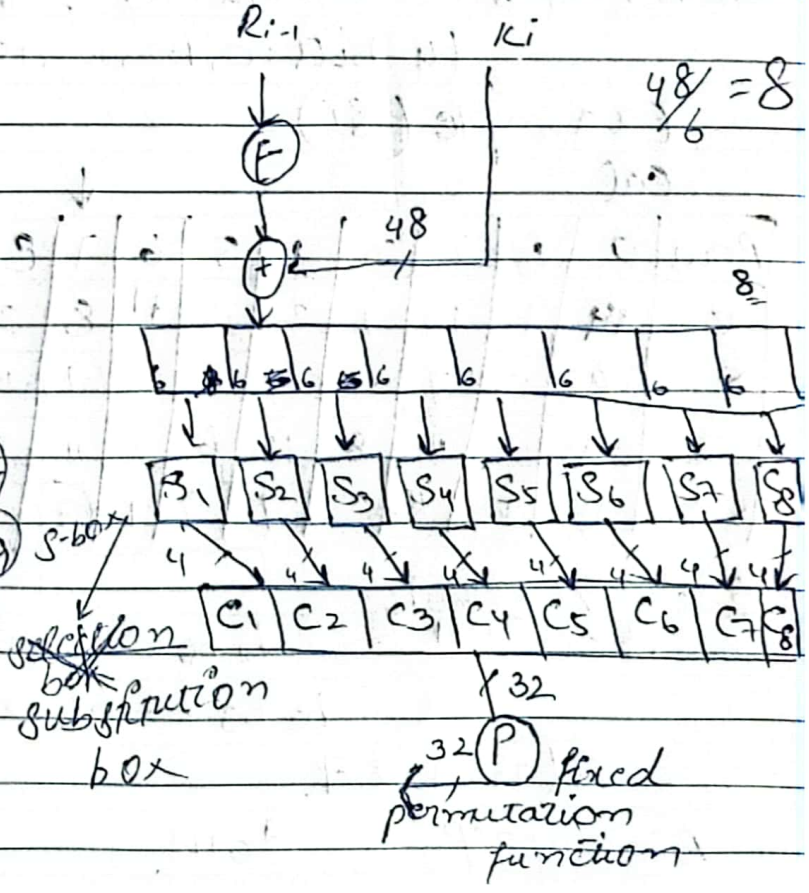


→ DES "f(•)" function.

E is an expansion function which takes a block of 32 bits as input & produces a block of 48-bit bits as output



→ 16-bits appear twice, in the expansion



Foundation for Advancement of Science & Technology

- S-boxes:

S-boxes are the only non-linear elements in DES design.

- each of the unique selection functions $S_1, S_2, \ldots, S_8$ takes a 6-bit block as input & yields a 4-bit block as output

B(6-bit) → [S-box] → C(4-bits)

S = matrix 4×16, values from 0~15

B(6-bit long) = $b_1 b_2 b_3 b_4 b_5 b_6$

$b_1 b_6$ => r = row of the matrix,
(2-bits : 0, 1, 2, 3)

$b_2 b_3 b_4 b_5$ => c = column of the matrix
(4-bits : 0, 1, 2, ..., 15)

Example (S_1)

Row \ Col	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
→ 1	0	15	7	4	14	2	13	1	10	6	12	11	9	(5)	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
→ 3	15	12	8	2	4	9	1	(7)	5	11	3	14	10	0	6	13

(1) $B = 101111$ ($b_1 b_2 b_3 b_4 b_5 b_6$)

$b_1 b_6$ => row = 11 = 3 (row)

$b_2 b_3 b_4 b_5$ => col = 0111 = 7 (row)

$S(r, c) = 7$

Binary rep → 0111

$b_1\ b_2\ b_3\ b_4\ b_5\ b_6$

Ex: $B = 0\ 1\ 1\ \ 0\ 1\ 1$

row = $b_1 b_6$ = 1st row (01)

col = $b_2 b_3 b_4 b_5 = 1101$ (13^{th} col)

$S(r=1, c=13) = 5\ (0101)$

- Finite Field (in AES): GF(P)

  mod p

  ↳ p is a prime

  +, −, ×, /

  ↳ all results should remain in the field

Ex:

	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

$\pm$ mod 3 ∴ inverse

PKC (private key, public key)

(Public key cryptography)

Public key

Lecture-9

RSA:

↳ invented by Ron Rivest, Adi Shamir & Len Adleman at MIT, 1978

- Block size can be variable
- key length can be variable
- Plain text must be smaller than the key length
- CT block will be the length of key.
- product of prime numbers, factoring of result

Applications: secrecy (secret exchange of keys, digital signatures)

→ Euler Totient function (ϕ):

if n is a +ive integer, ϕ (phi) function counts all the +ive integers less than n that are relatively prime to n.

Ex: $\phi(10) = ?$

1 ✓

2

3 ✓

4 1 2×2

5

6 2×3

7 ✓

8 2×2×2

9 ✓

$\phi(10) = 4$

co-prime? / relatively prime?

A & B are co-prime iff gcd(A,B) = 1 (common factor)

⇒ not necessary that the 2 numbers are prime. They have to be relatively prime to each other. e.g. 6 & 11 are co-prime. ↳ (11,1)

(2,3,1)

(6,11) → co-prime

Foundation for Advancement of Science & Technology

Generally, $\phi(p) = p-1$ if p is a prime number.
Ex: $\phi(7) = 7-1 = 6$ (7 = p is prime naturally)
①, ②, ③, ④, ⑤, ⑥

→ Multiplicative Inverse of a number X is Y iff Y multiplied by X yields 1.
So, $X * Y = 1$ → therefore, both X and Y are multiplicative inverses of each other.

- PKC → how does it work?

≈ Plain text M and cipher text C are integers b/w 0 and $(n-1)$ ⇒ n is the product of two prime numbers (p, q)

Ex:
Key 1 {e, n} , Key 2 {d, n}
→ where e and d are 2 secret numbers.

p, q
$n = p * q$

$C = M^e \bmod n$
$M = C^d \bmod n$

→ C & M are integers.. C is cipher text M is plain text.

→ sender does this "encryption"

receiver does this "decryption"

→ RSA key construction:
- select 2 large primes, p, q when $p \neq q$
- $n = p * q$
- Euler's Totient Function $\phi(n) = (p-1)(q-1)$

Ex: $p=2, q=5 \quad n = p \times q = 10$

$\phi(10) = (p-1)(q-1) = (2-1)(5-1) = (1)(4) = 4$

- Select "e" relatively prime to ϕ

$d \times e \bmod \phi = 1$ or

$d \times e \pmod{(p-1)(q-1)} = 1$

PUBLIC KEY => (e, n) > e & d are interchangeable

PRIVATE KEY => (d, n) / $(x^d)^e \bmod n = (x^e)^d \bmod n$

Ex1: $p=5, q=11$

$n = p \times q = 5 \times 11 = 55$ (I advertise $n=55$)

$e = ?$

$\phi(n) = \phi(55) = (p-1)(q-1)$

$= (5-1)(11-1) = 4 \times 10$

$= 40$

e : 3 (assumption => starting with smallest co-prime)

(e, n) → we let people know this

$e = 3$

$n = 55$ > public key

$M = 7$ => original text, plain text

$$55 \overline{)343}\ 6$$

330

13

$C = M^e \bmod n$

$= 7^3 \bmod 55 = 343 \bmod 55 = 13$

$C = 13$

receiver receives $\boxed{e = 13}$

He needs to solve $d \times e = 1 \pmod{\phi(n) = (4)(10)}$

$3d \equiv 1 \pmod{40}$ $\quad \because e = 3$

41
81
121 → woh numbers jin ka mod lene se 1 aaye. koi bhi value select karo jo 3 pe divide ho jaye

$3d = 81$

$d = \frac{81}{3} = 21$

$M = C^d \pmod n$

$M = 13^{27} \bmod 4055 = 77$

↳ $13^2 = 169 \bmod 55 = 4$ (example)

Ex 2:

$p = 23, q = 41$

Suppose $M = 35$

$n = p \times q = 23 \times 41 = 943$

$\phi(943) = (23-1)(41-1) = 22 \times 40 = 880$

Supposing $e = 3$, $n = 943$

$C = M^e \bmod n$

$C = 35^3 \bmod 943 = 42875 \bmod 943$

$C = 440$

$3d = 1 \pmod{880}$ $\quad (d \times e = 1)$

$3d = 1761$

$d = 587$

$M = C^d \pmod n$

$= 440^{587} \bmod 943 = 35$

2	880
2	440
2	220
2	110
5	55
11	11
	1

880 | 1761 (2)
1760
1

Foundation for Advancement of Science & Technology

Assuming $e = 7$

$C = M^e \bmod n$

$C = (35)^7 \pmod{943}$

$C = 545$

$d \times e = 1 \bmod \phi$

$7d = 1 \bmod 880$

$d = 503 \qquad (880 \times 4) + 1 / 7$

$M = C^d \bmod n = 545^{503} \bmod 943$

$M = 35$

Foundation for Advancement of Science & Technology

PKC (public key cryptography)

Lecture - 10

Diffie - Hellman (key exchange) → not used for encryption, used for transfer of keys

- 1st public key algorithm invented
- published in 1976.
- specific method for securely exchanging cryptographic keys over a public channel.
- concept given by Ralph Merkle
- named after Whit-field Diffie & Martin Hellman.
- public key exchange algorithm, neither encryption nor signature.

g mod p

g ~ generator

p ~ prime, used for modular arithmetic

Ex:

p = 11

When is g is primitive root of p, where $1 < g < p$, and p is a prime.

primitive: $g \to p$

$g^1 \pmod p \cdot g^2 \pmod p \dots g^{p-1} \pmod p$

$\{1, 2, \dots, p-1\}$ (10)

↓ should

possible map to all values from ↑

g \ x	1	2	3	4	5	6	7	8	9	10	x
1	1	1	1	1	1	1	1	1	1	1	
2	2	4	8	5	10	9	7	3	6	1	
3											
4											

Example

$2^2 \bmod 11$

$2^3 \bmod 11$

2 is primitive root of 11 as it contains all the possible values from 1 to 10

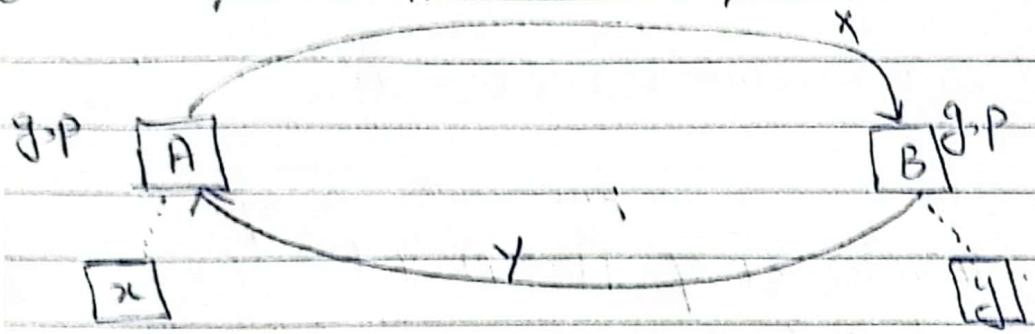
- A prime can have more than 1 primitive roots.

g = generator
p = prime modulus

- will use $g^i \bmod p$
- $1 \le g < p$
- g is primitive root of p

Ex:

1) g, p publicly shared b/w Alice (A) and Bob (B)
2) A computes $x = g^x \bmod p$ { x is a secret key of Alice }
3) B computes $Y = g^y \bmod p$ { y is a secret key of Bob }
4) A and B exchange X and Y
5) A computes $K_{AB} = Y^x \bmod p$
6) B computes $K_{BA} = X^y \bmod p$



$K_{AB} = K_{BA} = g^{xy} \bmod p$

{ x, X } { y, Y }
x, y → secret
X, Y → public

Ex:

p = 11 , g = 2

x → secret key of Alice → x = 8
y → secret key of Bob → y = 4

secret / public

Rough 390621
3904515172

Foundation for Advancement of Science & Technology

Alice {8, 3}
Bob = {4, 5}
private public

① $X = g^x \bmod p = 2^8 \bmod 11 = 3$
② $Y = g^y \bmod p = 2^4 \bmod 11 = 5$

③ $K_{AB} = Y^x \bmod p = 5^8 \bmod 11 = 4$
④ $K_{BA} = X^y \bmod p = 3^4 \bmod 11 = 4$

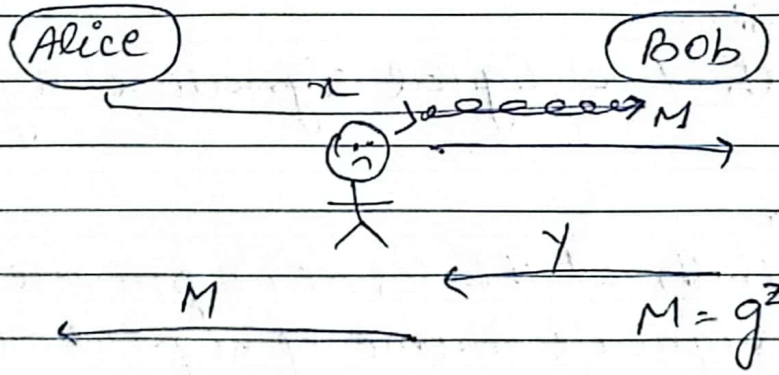
$K_{AB} = K_{BA} = 4$

$K_{AB} = K_{BA} = 2^{8 \times 4} \bmod 11 = 2^{32} \bmod 11 = 4$

→ Problem:

Man in the middle act on DH

$X = g^x \bmod p$ $Y = g^y \bmod p$



$M = g^z \bmod p$

$K_{AM} = M^x \bmod p$ $K_{BM} = M^y \bmod p$

$g^{xz} \bmod p \; != g^{xy} g^{yz} \bmod p$

Ex: $g = 3 \quad p = 353$

Alice → $x = 97$

Bob → $y = 233$

$X = g^x \bmod p = 3^{97} \bmod 353 = 40$ }97, 40{

$Y = g^y \bmod p = 3^{233} \bmod 353 = 248$ }233, 248{

$K_{AB} = Y^x \bmod p = 248^{97} \bmod 353 = 160$

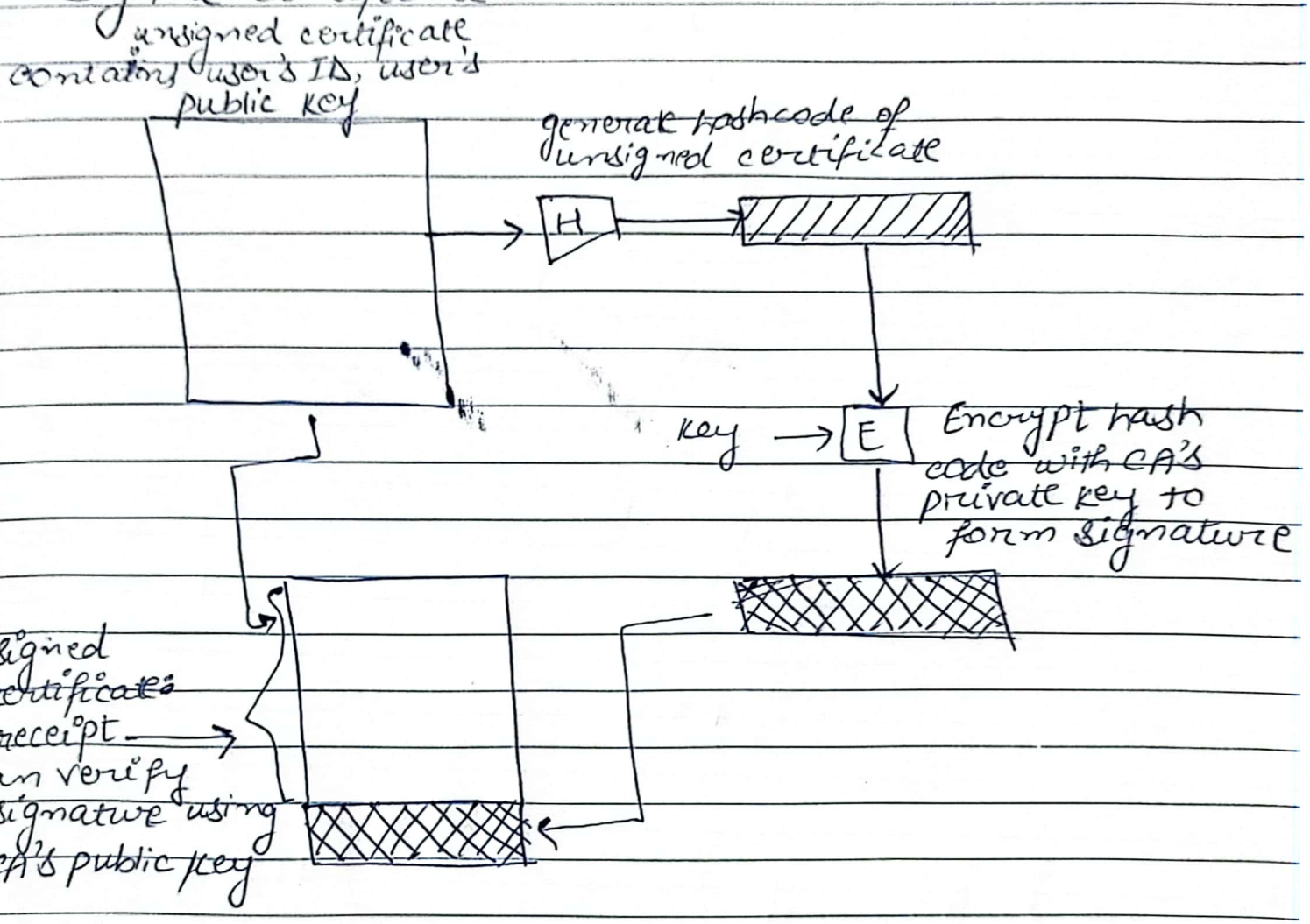
$K_{BA} = X^y \bmod p = 40^{233} \bmod 353 = 160$

$K_{AB} = K_{BA} = 160$

→ Digital Signatures:

- Combines a hash w/ a digital signature algo
- To sign
  - hash the data
  - encrypt the hash w/ the sender's private key
  - send data, signer's name & signature.
- To verify
  - hash the data
  - find the sender's public key
  - decrypt the signature w/ the sender's public key
  - the result of which should match the hash.

→ Digital Certificate:



∵ CA = certificate authority

↳ many vendors

=> Open SSL, Netscape, Entrust, VeriSign, RSA Security.

∵ RA = Registration Authority

• ~~authentication~~ user authentication user

→ banking apps

→ online registration