

assumption for ML algo

- ↳ know that f is b/w x , and y
but x nahi pta
- ↳ and that we are trying
to predict with help of ML algo.

Random Oracle model

- ↳ not a non-deterministic algo
- ↳ It's deterministic ~~algo~~ model.
→ just mathematical fn has x and y ke
bech mein but graphs will be
look like random.
- ↳ then any ML algo won't ~~be~~ give
any benefit.
- can't predict ' y ' with ' x '.

WJMK 3
30
2018

→ is it NP or not?

↳ ~~NP~~ Yes How?

→ can't write non-deterministic

poly time algo to solve this.

→ essential property of NP problem?

→ definition of NP problem?

↳ candidate soln must be

checked in poly time.

to give this soln, what to do??

what is soln domain here?

→ randomly taken from solution space

↳ binary string set.

so candidate soln

↳ any binary string.

→ as candidate soln can be checked in poly time.

→ what property to preserve so that problem is hard?

success probability of each iteration =

$$\frac{1}{2^k}$$

probability of soln → depends on size of output, not input.

assumption:-

→ each digit value is equally likely.

→ means every string is coming with equal probability

↳ only then this problem is

Hard!!

random oracle for



Name is RSA problem.

→ Identify attacker's problem:-

attacker is against encryption scheme

→ So attacker want to know m .

↳ this problem is hard or not?

* factors of n again PFA , can solve in polynomial time

↳ as will repeat RSA algo

3 steps again.

→ have to perform integer factorization. (If)

* RSA is not secured until integer "n" is hard → True.

* RSA is secured bcoz → False.

Left assumption is difference assumption.

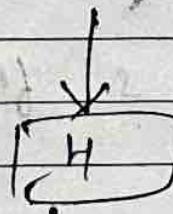
(RSA A) assumption.

but if AP related to RSA A assumption

or A assumption → OH A assumption

$$m \in \{0, 1\}^*$$

Hash function:-



RSA → secure hash algorithm

↳ soundness req. of

hash fn ??

$$\{0, 1\}^*$$

It must be deterministic algo.

With given $m \in \{0, 1\}^*$, H.

find another $m' \in \{0, 1\}^*$.

$$\text{s.t. } H(m) = H(m')$$

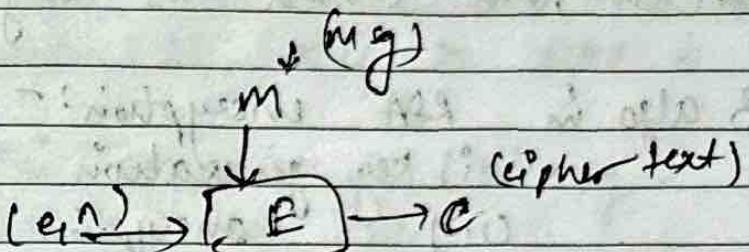
} what can ensure this problem is hard.

It must be deterministic

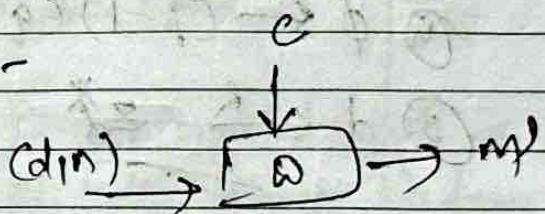
same input must give same output

- * (1) inverse calculation is also polynomial time.
 (2) gcd " " euclid algo.

ENCRYPTION:-



DECRYPTION:-



$$m' = (m^e)^d \pmod{n}$$

$$= m^{ed} \pmod{n}$$

$$m' = m^1 \pmod{n}$$

$$m' = m$$

Correctness proof

this can be written

e, d are inverse of each other.

how??

as, e^{-1} is d^{-1}

as well mod p,
but here is mod n.

$\phi \rightarrow$ extends of
because of extends them

(C) Soundness of RSA:-

-they are related as :-



OH can never be hard as long as DL is not hard.

① 3 algo in RSA encryption:-

- (i) key generation
- (ii) encryption
- (iii) decryption

② keyGen algo :-

generate 2 large prime numbers

$$\textcircled{1} \quad p \leftarrow \text{large prime}$$

$$\textcircled{2} \quad q \leftarrow \text{large prime}$$

$$\textcircled{3} \quad n \leftarrow (p * q)$$

$$\textcircled{4} \quad \phi \leftarrow (p-1)(q-1)$$

③ $d \leftarrow \text{random element of } Z_{\phi}^*$ (d is random element of Z_{ϕ}^*)

(d coming from random set $\rightarrow Z_{\phi}^*$).

b/w $1 \leq d \leq \phi - 1$, but must be coprime with ϕ .
definitely will always be in polynomial time.

④ $e \leftarrow d^{-1} \pmod{\phi}$

wrt ϕ , multiplicative inverse of d .

sif yeh mod ϕ se huya,
baaki sare (\pmod{n})
& hoga.

$(e, n) \rightarrow$ public key

$(d, n) \rightarrow$ private key

so DHP's hardness depends on DLP,

→ but don't write this argument → as ~~DLP~~ DLP hard, can't calculate ' a ', so DHP is hard.

This argument is wrong.

→ inference rule \rightarrow if then sub

→ add

$A \rightarrow B$ is equivalent to $\neg B \rightarrow \neg A$,
but not equivalent to $\neg A \rightarrow \neg B$.

(Here) \rightarrow

A : DLP is hard

B : DHP is hard,

DHP reduces to DLP.

C show

↳ if

↳ vice versa
also true?

→ security of DH is ensured due to

security of DLP.

↳ True

→ DH is secure bcoz DLP is hard \rightarrow false.

→ _____ as long as _____ \rightarrow False.

→ DL is hard \rightarrow DH is secure \rightarrow True.

DH is hard bcoz till date no other algo has been found which can solve it in polynomial time.

→ can't say it will remain hard always.

DH hardness assumption & DL hardness assumption. \rightarrow both are assumptions, but ...

they are related as :-

DH can never be hard as long as DL is not hard.

Q) 3 algo in RSA encryption:-

- (i) key generation
- (ii) encryption
- (iii) decryption

* keyGen algo :-

generate 2 large prime numbers $p \leftarrow$ large prime

$q \leftarrow$ " "

$$③ n \leftarrow (p * q)$$

$$④ \phi \leftarrow (p-1)(q-1)$$

⑤ $d \leftarrow R \subset Z_{\phi}^*$ (d is random element of Z_{ϕ}^*)

(d coming from random set $\rightarrow Z_{\phi}^*$).

blw: $1 \leq d < \phi$, but must be coprime with ϕ .
definitely not payga in polynomial time.

$$⑥ e \leftarrow d^{-1} \text{ mod } \phi$$

wrt ϕ , multiplicative inverse of d .

if yeh mod ϕ se huya,
baaki sare $(\text{mod}(n))$

se hoga.

$(e, n) \rightarrow$ public key

$(d, n) \rightarrow$ private key

so DHP's hardness depends on DLP.

→ But don't write this argument → as ~~DLP~~ DLP hard, can't calculate 'a', so DHP is hard.

This argument is wrong.

→ Inference rule \rightarrow if then sub
→ as \rightarrow

$A \rightarrow B$ is equivalent to $\neg B \rightarrow \neg A$,
but not equivalent to $\neg A \rightarrow \neg B$.

Here \rightarrow

A: DLP is hard

B: DHP is hard,

DHP reduces to DLP ✓

↳ show \rightarrow (f/w ??)

↳ CONVERSE VERSA
also true?

→ security of DH \rightarrow depends on hardness of DL?

↳ True

→ DH is secure bcoz DLP is hard \rightarrow false.

→ _____ as long as _____ \rightarrow False.

→ DL is hard \rightarrow DH is secure \rightarrow True.

DH is hard bcoz till date no other algo has been found which can solve it in polynomial time.

→ can't say it will remain hard always.

DH hardness assumption & DL hardness assumption. \rightarrow both are assumptions, but

WJMK
50
= = = = =

(1) RSA

→ OS first must be authentic
 with OS → such web browser aad. to have
b must be authentic.

② → If clock time change too → may be
 browser naa chle → as per certificate validity
 gone.

→ soundness:- diffie Hellman's protocol attack

attacker also want to calculate
 $(g^{ab}) \bmod p$

knows,

g, p, q ,
 $g^a \bmod p, g^b \bmod p$.

Diffie Hellmann
problem.

problem of
 attacker against
 Diffie Hellmann
 protocol.

Now ques" comes whether this problem is Hard or
 not?

If hard, ~~diffie~~ protocol is sound or not.

→ just need a, b from $g^a \bmod p$ and $g^b \bmod p$.

→ Discrete log problem → Hard problem.

→ can't calculate a, b .

so If discrete log problem (DLP) is hard

so diffie Hellmann protocol (DHP) is
 hard.

$$\textcircled{3} \quad (k_B)^a \bmod p \quad \longleftrightarrow \quad \textcircled{3} \quad (k_A)^b \bmod p$$

$$\begin{cases} (k_B)^a \bmod p = (g^b)^a \bmod p = g^{ab} \bmod p \\ (k_A)^b \bmod p = (g^a)^b \bmod p = g^{ab} \bmod p \end{cases}$$

$$\therefore (k_A)^b \bmod p = (k_B)^a \bmod p = \text{kash (say)}$$

behaves like

symmetric key.

soundness ??

→ Firstly identify objective of attacker in this protocol.
 ↓
 want to calculate kash.

assumption seq:-

exchanging public keys (k_A, k_B) via channel → channel won't change.

→ A, B won't be able to detect (with help of packets) that hm ek dusre se interact nahi kar rahi.

③ anonymous diffie Hellmann

↳ without certificate, doing exchange.

→ Combo of both:- (??)

③ → Ephemeral diffie Hellmann

↳ cost higher as

certification Ghut vaar!

WORK

thus

$$(g^a \bmod p)$$

modular exponentiation is easy!!!
 ↓ have library to solve it.
 functions to
 in polynomial time.

$g, p \rightarrow$ are constants

$$z_p^* \rightarrow z_p^* \rightarrow \text{BIJECTIVE}$$

FUNCTION!!

↑ inverse exists !!!

$$g^x \bmod p = \beta$$

$$\Rightarrow \text{discrete log } (\text{DLog } \beta)$$

$$x = \text{Dlog}_{g,p} \beta$$

stands for
"discrete" log

base 'g'
 wst
 modulo 'p'

given β , can calculate x easily or not???

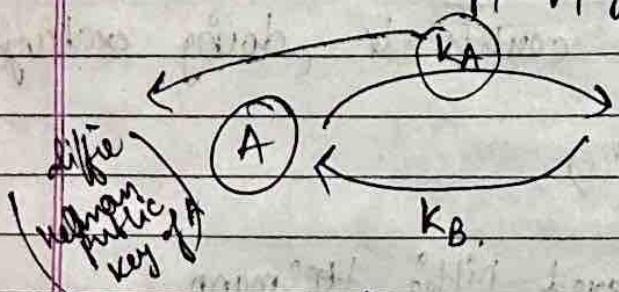
31 Aug 2024
 saturday

(lec-8) :-

1) fixed.

Diffie - Hellman protocol:-

$p, q, g \rightarrow$ global parameters known to all (even, may be to attacker also).



$$1) a \leftarrow K_B$$

$$2) K_{AB} \leftarrow g^a \bmod p$$

$$1) b \leftarrow K_A$$

$$2) K_{AB} \leftarrow g^b \bmod p$$

$a, b \rightarrow$ private keys of A, B.

lily subfield etc.

{ Super group \rightarrow "group extension" folle haan-
field \rightarrow field }

\Rightarrow field of \mathbb{C} is field extension over field of \mathbb{R} .

$$\mathbb{R} * \mathbb{R} \in \mathbb{R}$$

$$\mathbb{R} * \mathbb{C} \in \mathbb{C}$$

$$\mathbb{C} * \mathbb{C} \in \mathbb{C}$$

(*) any element can generate subgroup.

(*) Cyclic Subgroup?

let g is generator of $\langle \mathbb{Z}_p^*, * \rangle$

$$\text{let } (g^\alpha) \text{ mod } (p) = \beta$$

$$\beta \in \mathbb{Z}_p^*$$

$$\text{as } \beta \neq 0$$

$$\alpha \beta \in \{1, \dots, p-1\}$$

ultimately

it repeats cyclically.

so, so not benefit of taking α till ∞ , as it will just repeat the things.

so, take $\alpha \in \mathbb{Z}_p^*$ also.

$$\alpha \in \{1, \dots, p-1\}$$

but this is not order of $\beta \rightarrow$ but some permutation of \mathbb{Z}_p^* .

thus

$$g^d \bmod p \leftarrow Z_p^* \rightarrow Z_p^* \rightarrow \text{BIJECTIVE FUNCTION!}$$

so inverse exists !!!

 $(g^d \bmod p)$

modular
exponentiation is
easy!!!
have library to solve it.
functions to
in polynomial time.

$$g^d \bmod p = \beta$$

 $\Rightarrow \text{DLog}(\beta)$

$$\alpha = \text{DLog}_{g, p} \beta$$

stands for
"discrete" log

base 'g'
wrt
modulo 'p'

 $g, p \rightarrow$ are constantsgiven β , can calculate ' α ' easily or not???31/ Aug/2024
Saturday

(lec-8) :-

① fixed.

Diffie-Hellman protocol:-

$p, q, g \rightarrow$ global parameters
known to all
(even, may be to
attacker also).

Diffie
Hellman
public
key

A

KA

B

KB

$$① a \leftarrow Z_q^*$$

$$② K_A \leftarrow g^a \bmod p$$

$$① b \leftarrow Z_q^*$$

$$② K_B \leftarrow g^b \bmod p$$

 $a, b \rightarrow$ private keys of A | B.

liley subfield etc.

Super group $\xrightarrow{\text{field}}$ "group extension" folle hain.
 $\xrightarrow{\text{field}}$ "field"

\Rightarrow field of \mathbb{C} is field extension over field of \mathbb{R} .

$$\mathbb{R} * \mathbb{R} \in \mathbb{R}$$

$$\mathbb{R} * \mathbb{C} \in \mathbb{C}$$

$$\mathbb{C} * \mathbb{C} \in \mathbb{C}$$

(*) any element can generate subgroup.

(**) Cyclic Subgroup?

let g is generator of $\langle \mathbb{Z}_p^*, * \rangle$

$$\text{let } (g^\alpha) \text{ mod. } (p) = \beta \in \mathbb{Z}_p^*$$

as $\beta \neq 0$

$$\beta \in \{1, \dots, p-1\}$$

ultimately it repeats cyclically.

so not benefit of taking α till ∞ , as it will just repeat the things.

so, take $\alpha \in \mathbb{Z}_p^*$ also.

$$\alpha \in \{1, \dots, p-1\},$$

but this is not order of $\beta \rightarrow$ but some permutation of \mathbb{Z}_p^* .

this

$$(g^x \bmod p)$$

modular
exponentiation is
easy!!!

↓ have library to solve it
functions to
for polynomial

$g, p \rightarrow$ are constants

$$z_p^* \rightarrow z_p^{*x}$$

BIDJECTIVE
FUNCTION!!

so inverse exists !!!

$$g^x \bmod p = \beta$$

$$\Rightarrow \text{dlog } (\beta)$$

$$x = \text{dlog}_{g, p} \beta$$

stands for
"discrete" log.

base 'g'
wst
modulo 'p'

given β , can calculate ' x ' easily or not???

31/ Aug/ 2024
Saturday

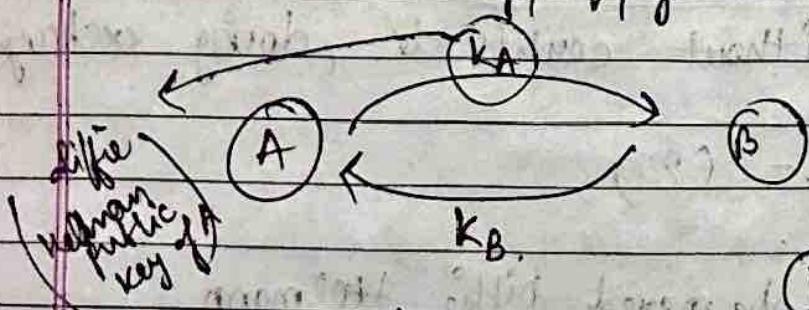
lec-8:-

(1) fixed.

(2) Diffie - Hellman protocol:-

$p, q, g \rightarrow$ global parameters

known to all
(even, may be to
attacker also).



$$① a \leftarrow z_q^*$$

$$② KA \leftarrow g^a \bmod p$$

$$① b \leftarrow z_q^*$$

$$② KB \leftarrow g^b \bmod p$$

$a, b \rightarrow$ private keys of A, B.

WJM

Ugly subfield etc.

[super group to "group extension" better name
 field — field — field —]

\Rightarrow field of C is field extension over field of R .

$$R \neq R \in R$$

$$R \neq C \in C$$

$$C \neq C \in C$$

(*) any element can generate subgroup.

(**) Cyclic Subgroup?

let g is generator of $\langle \mathbb{Z}_p^*, * \rangle$

$$\text{let } (g^\alpha) \bmod(p) = \beta \in \mathbb{Z}_p^*$$

as $\beta \neq 0$

$$\alpha \beta \in \{1, \dots, p-1\}$$

unfortunately

it repeats cyclically,

\therefore go not benefit of taking α till ∞ , as it will just repeat the things.

so, take $\alpha \in \mathbb{Z}_p^*$ also.

$$\alpha \in \{1, \dots, p-1\}$$

but this is not order of $\beta \rightarrow$ but some permutation of \mathbb{Z}_p^* .

this smallest ' i ' \rightarrow is 'order' of a .
 In \mathbb{Z}_p^* , we'll have only one i s.t. $a^i \text{ mod}(p) = 1$.
 $G = \langle \mathbb{Z}_p^*, * \rangle, p=11$

① let $a = 2, p=11$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 2 \\ 4 \\ 8 \\ 5 \\ 10 \\ 9 \\ 7 \\ 3 \\ 6 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 4 \\ 8 \\ 3 \\ 5 \\ 10 \\ 9 \\ 7 \\ 6 \\ 2 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 8 \\ 3 \\ 5 \\ 10 \\ 9 \\ 7 \\ 6 \\ 2 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 6 \\ 5 \\ 10 \\ 9 \\ 7 \\ 3 \\ 2 \\ 8 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 5 \\ 10 \\ 9 \\ 7 \\ 3 \\ 2 \\ 8 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 10 \\ 9 \\ 7 \\ 3 \\ 2 \\ 8 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 9 \\ 7 \\ 3 \\ 2 \\ 8 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 7 \\ 3 \\ 2 \\ 8 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 3 \\ 2 \\ 8 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 2 \\ 8 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 8 \\ 1 \end{matrix}$$

$$\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \\ 10 \end{matrix} \rightarrow \begin{matrix} 1 \end{matrix}$$

order of $2 = 10$.

subset, 2 in generating is $\rightarrow \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

$$\mathbb{Z}_p^*$$

* Primitive root ?

* what is Generator?

\hookrightarrow element which generate entire group
 like, 2 here is a generator.

known as
 "SUBGROUP"

so we generated
 a small group with
 same operator
 as that of
 parent group

subset of main
 group which in itself is a
 group.

* Non-repudability can be achieved only by Digital signature \rightarrow public key cryptography, can't be achieved by symmetric key cryptography.

* Elliptic Curve Cryptography. (Not in course)
↳ Nowadays, mostly it is used.

Hard Problem of Public key Cryptography:-

Discrete Log Problem

$$\text{Want } i \text{ s.t. } (a^i) \bmod(p) = t,$$

$$\text{Let, } p = 11, a = 5$$

$$\text{then } (a^i) \bmod(11)$$

$$\text{for } i \Rightarrow 0 \rightarrow 1$$

$$1 \rightarrow 5$$

$$2 \rightarrow 3$$

$$3 \rightarrow 4$$

$$4 \rightarrow 9$$

$$5 \rightarrow 1$$

$$6 \rightarrow 5$$

$$7 \rightarrow 3$$

$$8 \rightarrow 4$$

$$a^4 = a^3 \times a$$

$$a^3 = a^2 \times a \\ = 3 \times 5$$

$$= 15$$

$$\equiv 4 \pmod{11}$$

OBSERVATION:-

$$① \text{ So, } (a^i) \bmod(11) \in \{1, 5, 3, 4, 9\}$$

② We'll have ' ∞ '. i.e. s.t. $(a^i) \bmod(11) = 1$. But want smallest i .

\rightarrow And we are looking in $\mathbb{Z}_{p^k}^*$. \rightarrow so that $i = 0$ answer no. no.

WEEK 3
MATERIAL

$$\mathbb{Z}_p = \{0, 1, \dots, p-1\}$$

$$\mathbb{Z}_p^* = \{1, \dots, p-1\}$$

① $\langle \mathbb{Z}_p, +, * \rangle$ is a finite field.

↳ with identity of one operator, we did not require the inverse of another operator.

$\mathbb{Z}_p, + \rightarrow$ group ✓

$\mathbb{Z}_p, * \rightarrow$ group ✗

② $\langle \{0, 1\}^n, +, * \rangle \rightarrow$ also a field.

so much → (almost 10-times)

③ Symmetric cryptography is faster than public key cryptography but both are equally secure.

↳ why is public key cryptography used then?



fact it takes little bit large key size to ensure security as much as of symmetric.

↓ that's why speed is.

costly!
but only once!!

to encrypt symmetric key !!

used to encrypt data!

RSA \rightarrow 128 bits of data can encrypt 1 by this costly algo.

so that symmetric key encrypt

no key paaya through open channel, and public key a tariha, so jiske paas private key hogi, kisi no decrypt ke paayega \rightarrow so for mutual secure establishment \rightarrow public key is required.

To establish secure communication between receiver and sender through open channel

WEEK 3

23 Aug | Friday

lec-7 :-

$$G = \langle S, \cdot \rangle$$

$\text{order}(G) = |S| = \text{cardinality of } S$

let \Rightarrow

$$G = \langle \mathbb{Z}_n, + \rangle, \quad \mathbb{Z}_n = \{0, 1, \dots, n-1\}$$

\hookrightarrow this is $\text{mod}(n)$

\hookrightarrow is it group or not??

Closure property ✓

Associativity ✓

Additive inverse of 3 is 7.

Additive inverse of a is $n-a$.

so valid group.

Similarly $\langle \mathbb{Z}_n^*, * \rangle$ is also group.

\hookrightarrow No!

$$\text{let } n=10$$

2 ka inverse does not exist.

Numbers which are co-prime with 'n',

- \hookrightarrow if unk inverse milega. bs!; baakiyon ka nahi!

$\hookrightarrow \mathbb{Z}_n^*$ is that subset.

$\hookrightarrow \in \mathbb{Z}_n$, but are also co-prime with n.

\hookrightarrow satisfies closure property ✓

associativity ✓

Identity exist ✓

Inverse exist ✓

so, $(\mathbb{Z}_n^*, *)$ is a valid group, (MG)

(multiplicative group)

to find another msg s.t. both have same hash value.

↳ $h \rightarrow \text{fixed}$ predetermined, 1-8 bit value.
and it is possible as it is many-to one function.

↳ as $n \gg l$.

so, there does not exist only one such m' ,
but saare honge s.t. $H(m) = H(m')$,
then how can say 'Hash' is secure ??,

↳ we want this problem of finding such m' to be a hard problem.
also want to design hash f^n s.t. above probm hard to.

↳ collision resistance property

Collision resistant hash fn

↳ collision exist but
computationally infeasible to get,
not impossible ***.

MAC

MAC takes 'K' as input while calculating hash.

Diffusion property ?

→ hashing is deterministic algorithm.

↳ as for same ' m ',
 $H(m)$ should be same.

→ Not a randomistic algo.

↳ hash chache it generates random
string of bytes (esa lgta hai), but for
same m , $H(m)$ must be same.

→ why attacker modifies file, can modify hash
only if file detection fails na to
paata ki attack hoga → but we assume
file and hash ikathaa chitta hai.

Ans

Authenticity w/o Integrity has no meaning ???

→ SHA → secured version ???

① Encryption properties? (or Requirements)
ensuring that

i) Correctness :- Encryption would work correctly when done.

ii) Soundness :- ??

→ Inv. encryption - Decryption :-

$C = E(m, k)$ → only k is unknown, want k !

↳ Hard problem.

② Soundness req. of hash :-

If msg changes, hash must also change.

Integrity :- If msg is modified; it must be detected.

→ In hashing, problem of attacker is computational problem.

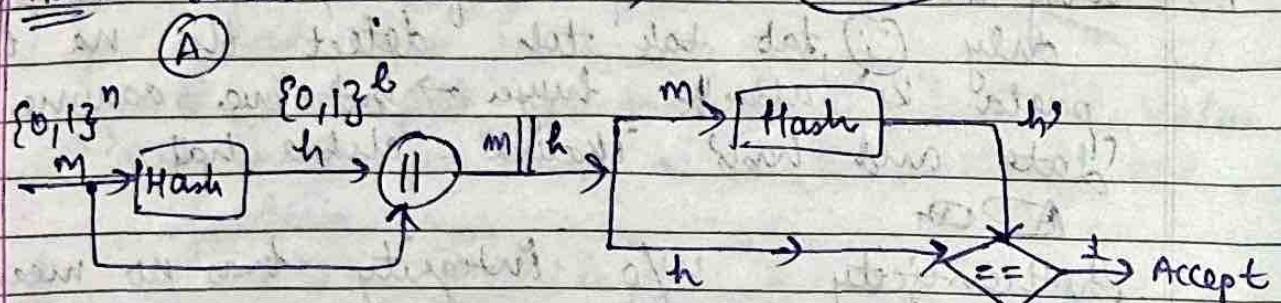
Given: m , h , Hash algorithm also known

Compute: ? → Problem of attacker.

→ m' (as m is passed over insecure channel)
s.t. $H(m) = H(m') = h$

① Hash :- sender . . . ($n > > l$)

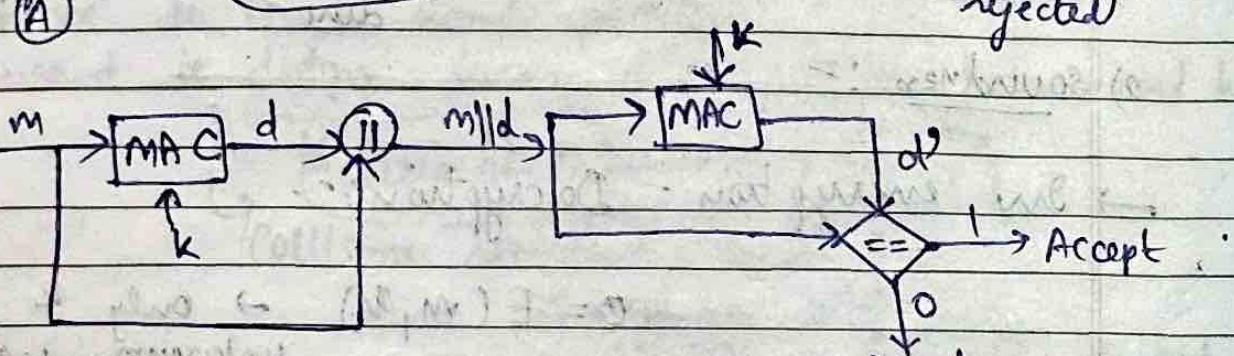
Integrity



② MAC :-
sender

Authenticity &
Integrity

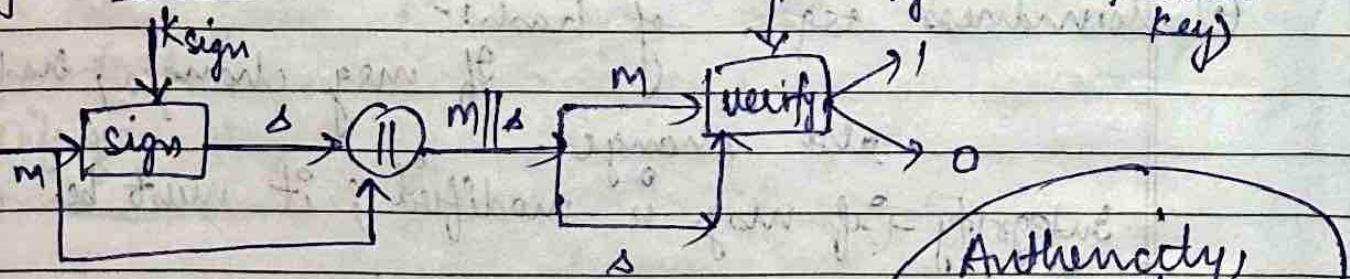
(Permission is
rejected)



MAC is always symmetric cryptographic mechanisms.

③ Digital Signature Mechanism (DSM)

Kuverify → (Verification key)



k_{sign} = Inverse ($Kuverify$)

$Kuverify$ = Public key A

k_{sign} = Private key A

Authenticity,
Integrity,
non-repudiability

How?
(H/W).

So, DSM is always asymmetric cryptographic mechanism.

to ensure in poly time, have to take constant numbers &
 $W\leq \frac{c}{n^{\alpha}}$ Page No.:
Date:

① f easy, f^{-1} hard, but if some key
 is known, f^{-1} easy.

② Both functions are used in cryptography,

(NP solvable) from \mathbb{Z}^m space \rightarrow like random

③ Integer Factorization Problem :- no. what is the

\Rightarrow check by all \mathbb{Z}^{1^n}

$H(\text{hard}) = H(\text{hard})$

or not \Rightarrow

or not viya

Chances constant

$$36 = 1 \times 2^2 \times 3^2.$$

Q:- If computation problem :- In Exam :- Write

given :- n

find/ compute :- p

such that : $p | n$

integer factorization
cryptography
problem

Q:- If decision problem :-

given :- n

decide if n is prime

If known

n is

exactly

product

of prime

numbers.

16 Aug 2024

Friday

[lec-6] :- (Block Diagrams:-)

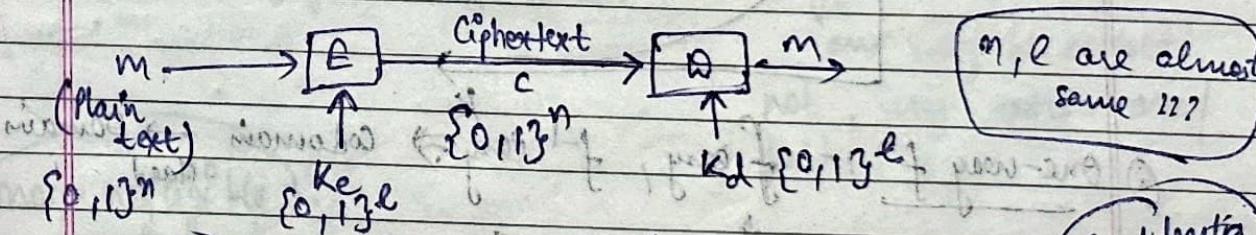
Encryption-decryption :-

Sender \xrightarrow{E} Receiver

$K_e = \text{Inverse}$
 (K_d)

(A)

(B)



m, l are almost same ??

for symmetric encryption :- Confidentiality

$K_e = K_d$

$C = E(M, K)$

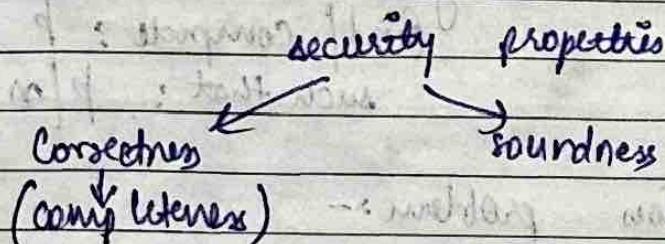
only unknown K

K_e = public key

K_d : private key

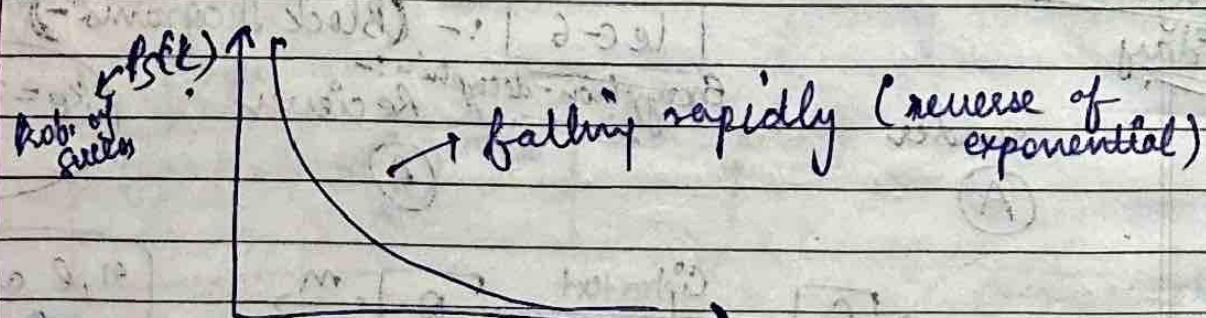
K : shared secret key

- ① If growth rate of f^n is very high (exponential),
 then f^n is non-negligible ($\approx \infty$)
 ↗ power for attacker → opposite wall ka security protector etc.
 → attacker is an algo to find private key etc.
- we want problem of attacker should be hard (for our safety) but want hashing, mapping, encryption etc to be hard (of w)
 maybe problem, so cryptographic algo must be easy)



So that there should not exist deterministic poly. time complexity algo to solve its problem.

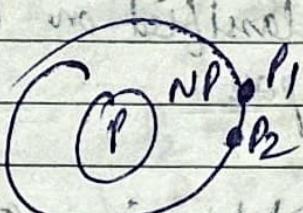
N Queen Hard only if $n \approx \infty$, not for $n = 2, 3$ etc.



- ② One-way f^n i- f easy, f^{-1} easy \Rightarrow codomain \rightarrow domains Hard
- Domain $\xrightarrow{\text{easy}} \text{Codomain}$

- ③ Trap door f^n i- rat can enter easily, and only if you want them rat can come out. \rightarrow easily, of w hard.

any NP problem can be reduced to NP complete (???)
 one "complete" problem can be converted to
 any other NP "problem".



$P \leq_p P_1$ & $P_1 \leq_p P_2$ then reduce to
 $P \leq_p P_2$

→ Hard problem which is NP \Rightarrow may not be NP

→ any hard problem \in NP or NP-Hard or NP-
Complete.

all have no problem with "poly. TC sol" algo.

Q. why not all hard problem are ~~NP~~ NP??

→ In CS we use NP type of problems.

NP → have not deterministic Poly TC,

but must have non-deterministic algo to solve it.

NP-Hard → This may not have non-deterministic poly TC
 algo to solve.

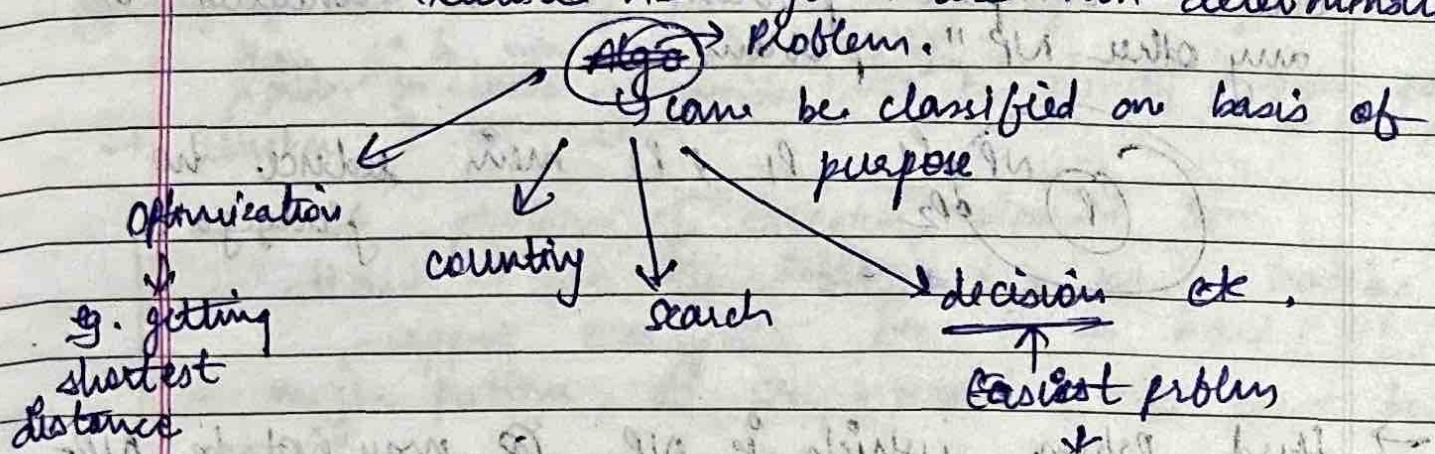
ans may be correct

not, may return ans/
 failure.

non-negligible

get B not threshold, it is function
 of input size (k) (no. of bits)
 → function is not negligible \rightarrow non-negligible
 fn.!

e.g. quick sort with random pivot everytime,
Heuristic AI algo. \rightarrow are non-deterministic.



as any input but output is either "Yes"/1 or "No"/0.

\hookrightarrow (0-1) problem.

also known as

P_1 is reducible to $P_2 \Rightarrow$ if P_2 is easy then P_1 is easy.

$\hookrightarrow P_1$ is not more harder than P_2 .

Non-deterministic decision algo

LV
 \downarrow
Las Vegas

MC

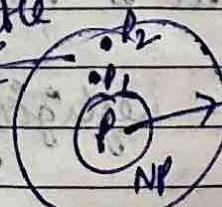
Monte Carlo

It always gives answer, may
may not be correct.

Yes

No biased.

P_2
reducible
to
 P_1



biased

\rightarrow as more fair, hardness P_2

WDMK
SUNDAY

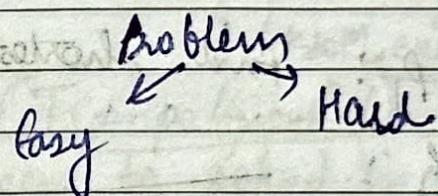
- ⇒ If P_2 is easy, then P_1 is easy.
- ⇒ But if P_2 is hard $\Rightarrow P_1$ is hard. if P_1 reduces to B .
- ⇒ P_1 hard $\Rightarrow B$ hard. (???)
- \downarrow
 $(P_2 \text{ hard as long as})$
 $(P_1 \text{ is hard.})$

9/8/24

sunday

WDMK
SUNDAY

lec-5 :-



Algorithm

Deterministic Non-deterministic / Probabilistic

① nature of algo is deterministic

i) means same input \rightarrow same output

ii) They ~~use~~ ^{there is no} source of randomness in algo/problm \rightarrow like generating random no.

iii) Flow chart nota hdi algo ka, so same input \rightarrow same output

path from start \rightarrow end

is constant / same everytime, so

can determine path everytime, but

in non-deterministic we can not judge /

we guess the path as it is not fixed

\rightarrow in AI mostly we use Deterministic algo.

↳ Non-deterministic

↳ diff output possible

e.g. quick sort with random pivot everywhere,
Hanoi AI algo. \rightarrow are non-deterministic.

algo

Problem.

↳ can be classified on basis of purpose

optimization

counting

search

decision pk.

e.g. getting
shortest
distance

Easiest problem

as any input but output is

either "Yes"/1 or "No"/0.

\hookrightarrow (0-1) problem.
also known as

P_1 is reducible to $P_2 \Rightarrow$ if P_2 is easy, then P_1 is easy.

$\hookrightarrow P_1$ is not more harder than P_2 .

Non-deterministic decision algo

LV

Las Vegas

MC

Monte Carlo

↳ always gives answer, may
may not be correct.

Yes

No biased.

P_2
reducible

P_1
to

P_2 is
reducible
to P_1 .

P_1

\rightarrow as more fair, hardness P_2

NP

WDMK

⇒ If P_2 is easy, then P_1 is easy.

⇒ But if P_2 is hard $\Rightarrow P_1$ is hard.] → If P_1 reduces to P_2 .

⇒ P_1 hard $\Rightarrow P_2$ hard.

↓

(P_2 hard as long as
 P_1 is hard)



~~9/8/24~~
Friday

WDMK

Lec-5 :-

Problem

Easy

Hard

Algorithm

Deterministic

Non-deterministic / Probabilistic

① nature of algo is deterministici) means same input \rightarrow sameoutput will beii) They use there is no sources of \rightarrow like generating

randomness in algo/problm(???)

random nos

iii) Flow chart nota hui algo ka,

same input

toh same input ke liye,

diff output

path from start to end

possible

iv) constant / same everytime, so

non-deterministic

can determine path everytime, but

in non-deterministic can not judge /

guess the path as it is not fixed.

→ In AI, mostly we use deterministic algo.

Q. What do you mean by P_2 is more harder than P_1 or P_1 is not as hard as P_2 , both are hard. How you are comparing hardness???

A. It means that, P_1 reduces to P_2 .

↓
↳ what do you mean by
 $(P_1 \leq P_2)$??

using soln of P_2 , P_1 can be solved.

Example:-

P_1 : find shortest route in unweighted graph

P_2 : _____ unweighted graph.

then P_2 reduces to P_1 .

(Polynomial time reduction)

(*) P_1 is reducible to P_2 in polynomial time

means?

like unweighted to fully weighted
main keege; they fr. of (2) fully weighted
wale ka use keege; then again kya kya
kuch keege.

Joh jn ko use kene se phle kuch kiya, then
jn call ke baad kuch kiya \rightarrow in
dono mein jo time lgा poly. must -
(and jn call kiya \rightarrow may be ek se jyada
vara ke she hours \rightarrow must call poly no.
of times (P???)

WORK

in arr[]. $\rightarrow n = \text{no. of elements in array}$
 ↳ as assuming every element
 is taking constant space, so we take
 ignore that.

But here input size \rightarrow not constant.
 It's not like, & line ka code hai toh easy!!

* what is easy and what is hard problem ???

↳ problem which
 can not be solved
 in polynomial
 time complexity.
 \rightarrow PPT exists but does not

have. ($Tc = \text{polynomial}$). (may)

↳ which can be solved in poly. time.

$O(1)$

$O(\log k)$

$O(k)$

$O(k^2)$

etc.

but in future
 it may have!

$\alpha^{\log k} \rightarrow$ subexponential ($>$ poly, $<$ exp)

$\alpha^k \rightarrow$ exponential

↳ puzzle

↳ N queen

↳ graph coloring problem

Q. What is NP, NP Hard, NP complete problem?

↳ all are '∞', unbounded

→ centre se jitan aur jaayoge,
 difficulty level extra bhega.
 sets!

Q. What do you mean by P_2 is more harder than P_1 or P_1 is not as hard as P_2 . Both are hard. How you are comparing hardness???

A. It means that, P_1 reduces to P_2 .

↓ what do you mean by this??,
 $(P_1 \leq P_2)$??

using sol' of P_2 , P_1 can be solved.

Example:-

P_1 : find shortest route in weighted graph

P_2 :  unweighted graph.

then P_2 reduces to P_1 . (Polynomial time reduction)

K P_1 is reducible to P_2 in polynomial time

means?

like unweighted to fully weighted
 mein kaise; then fr' of ~~fully~~ weighted
 wale ke use kaise; then again kya naar
 kuch kaise.

Jab jn ko use kese se phle kuch kya, then
 jn call ke taad kuch kya → inn
 dono mein jo time lgा poly. mud -
 and jn call kya → may be ek se jyada
 var ki the hain → must call poly no.
 of times (????)

WORK
STUDY

Time complexity $\rightarrow n = \text{no. of elements in array}$

↳ as assuming every element

is taking constant space, so we take
ignore that

But here input size \rightarrow not constant.

It's not like, a line ka code hai toh easy!!

* what is easy and what is hard problem???

↳ problem which
can not be solved
in polynomial
time complexity.

↳ exists but does not
have. ($TC = \text{polynomial}$), (may)

↳ which can be solved in poly. time.

$O(1)$

$O(\log k)$

$O(k)$

$O(k^2)$

etc.

but in future
it may have!

$\alpha^{\log k} \rightarrow$ subexponential ($> \text{poly}, < \text{exp}$)

$\alpha^k \rightarrow$ exponential

↳ puzzle

↳ n queen

↳ graph coloring problems

Q. What is NP, NP Hard, NP complete problem?

↳ all are 'co', unbounded

→ centre se either 'dun jaafoge,
difficulty level, itna bdaega'.

→ sets!

Public key \rightarrow key which is not supposed to be secret to somebody.

Symmetric key \rightarrow shared secret key (?)

\hookrightarrow where no body can be sender, only one who knows your secret key can be sender.

But wait...

Private key = inverse (public key)

\hookrightarrow tan private kaise milo wo?

where is secrecy of private key, then?

\rightarrow Algos are public always.

\hookrightarrow to generate keys

\hookrightarrow to get k_2 from k_1 .

\rightarrow 2 options:- (1) Inverse does not exist.

(2) You can't calculate inverse!

\hookrightarrow it's hard!

\hookrightarrow it exists, but difficult.

\rightarrow Not possible!

\rightarrow as one-one hogi pkka!

\hookrightarrow ek private key ek public key se
li' map hogi,

\rightarrow onto bhi hogi pkka!

So, Inverse exists.

& what do you mean by it?

\rightarrow Hard can be problem, not algo.

\rightarrow TC " " of algo, not of problem.

\hookrightarrow size of input (in no. of bits)

\hookrightarrow bits req. to hold input data.

Digital Sig nature

↳ ^{ensures} authentication, non-repudiability, integrity.

⇒ Hash ↳ Receiver and Sender both uses same hash algorithms.

↳ does not require key, while all other classes req. key.

↳ so called keyless class.

⇒ symmetric key cryptography / symmetric cryptography →

• if $\text{inverse}(\text{key}) = \text{key}$, then

Asymmetric key cryptography / asymmetric cryptography.

↳ so Hash → is neither symmetric nor asymmetric.

Digital sig → always Asymmetric

MAC → " Symmetric

Encry-Decry → can be as well as Asymmetric

e.g. MD5, SHA are examples of 'hash' algo.

Asymmetric key cryptography → also known as Public ".

k_1, k_2 should be secret to Receiver,

but k_1, k_2 will be "the sender or someone else? → No!

↳ as sender can be anybody,
so k_1 should be announced to all, so k_1 is public key and k_2 is private key.

e.g. $(\Sigma, +, *)$ is a ring, but not a field.

• \Rightarrow multiplicative identity exist,
but " " inverse does not "

In ring, can perform $+, -, *,$ not \div , as
multiplicative inverse does not exist.

* Group, ring, fields are 'as' as sets are 'as'.

→ Why we need closure property?

↳ as we need a fixed domain!!

Q. But if domain is very big, ∞ , then what is use of
closure property?

→ So in Cryptography, we need finite group/
finite field.

Decryption key must only be known to receiver secretly.

$k_2 = \text{inverse}(k_1)$ ↗ Encryption key

→ want group where ~~all~~ inverse of no. is no. itself.

→ Let set $\rightarrow \{0, 1\}^n$, binary operator $\rightarrow \oplus$,

↳ 4 binary strings of fixed length.
Is it a group? "

Identity element = $\underbrace{0000\dots 0}_{\text{on length}}$

Inverse (element) = element itself,

Cryptography

↳ set of algos to secure the fundamental
req. of information security. (4 req.)

Q) what is inverse of an element?

\Rightarrow 3

↳ multiplicative inverse $\rightarrow (1/3)$

↳ additive inverse $\rightarrow (-3)$

5/8/24

Monday

Lec-4

WORK
STATION

Group: $\langle S, \text{ binary operator } (*) \rangle$

$a, b \in S$.

$$f(a, b) = c$$

Group must satisfies:-

(1) closure i.e. $c \in S$.

(2) associativity : (3) identity (4) inverse.

Fees sit. $a * e = e * a = a \forall a \in S$.

$$\cancel{f(S \times S)} \rightarrow a * b = b * a$$

\rightarrow if group satisfies commutative property, then it is

abelian group.

\rightarrow field: - \rightarrow 2 operators \rightarrow usually $+$, $*$

↳ group w.r.t both operators

$+, *$ \rightarrow fundamental operators.

$\div, \frac{1}{\cdot}$ are not fundamental operators,

$\frac{1}{\cdot}$ is nothing but $*$ with multiplicative inverse

\rightarrow Ring: - more than group, less than field.

↳ multiplicative inverse is missing in the ring.

* key is req. for both encryption & decryption.

why for this, 'key' is mandatory & thus req.

to ensure that only 'B1' ke 'key' hi use decrypt ke page!

→ so need to supply some secret which relates with secret on other side.

$$E : \{0,1\}^n \times \{0,1\}^l \rightarrow \{0,1\}^n$$

↑ ↗ plain text ↗ key ↗ cipher text
 (Encryption function)

↳ binary fⁿ (or input fⁿ)

↳ 1st input: n bit binary string
2nd l

output: n

$$E(m, k_1) = c$$

$$m \in \{0,1\}^n$$

$$D(c, k_2) = m$$

$$k_1 \in \{0,1\}^l$$

k₂ → receiver's private key

$$c \in \{0,1\}^n$$

(k₁) → somehow related to k₂.

so this is not random string of l bits,
k₂ can generate randomly, but k₁ have to compute from k₂

OR

k₁ can generate randomly, but then k₂ have to compute from k₁.

so 3 relationship → 'Inverse' of each other,
as ek ka effect on plain text, dusse ne
kaatna hai, means usko firse plain text
baana hai!

Plain Text → Cipher Text

algo used is "Encryption".

Now, need plain text back from cipher text → for that algo used is called "Decryption".

Ciphertext → Plain Text

Decryption

Usually size of cipher and plain text is same.

- Q) Are these algorithms (Encryption / decryption) are secret or public?
- A) Public, obviously !!.

So attacker can also use them, and cipher text channel is like plain text.

This defn is not complete.

→ Decryption must req. some additional info

(Secret piece of info) to convert cipher text

to plain text. (It must not be known to attacker).

Known as 'key'

different from Password.

also long

of bits

this must be random sequence of bytes

thus human can't remember such a sequence of almost 16 bytes.

→ Key is hash of Password.

must be
human
readable
and
rememberable

↳ seq. of
characters

↳ go binary form
↳ mem. store hots in memory.

↳ created/generated by computer via some algorithm.

↳ some

(algo / protocol, they may have standard) work ~~c/o.~~ ASCII → standard, not algo, not protocol.

Cryptography :-

algo to ensure :-

→ I-A-NR.

eg: SHA, cipher, MD5, RSA, TLS

• Diff b/w algo & protocol.

can be
executed in
one system:

(part
of
protocol)

→ can't be executed on

one system,

at least 2 entities
charlie, executes on top of
computer n/w systems.

Inside TLS, SSL, we use cryptographic algo.

→ Cryptographic algo can be classified in diff-2 ways:-

(1) symmetric v/s asymmetric (from mathematical pt of view)

(2) from purpose pt of view, can classify as:-

Encryption,
decryption
to ensure
"Confidentiality"

Hashing
MAC
media access control

Digital
signature

msg → stream of bits

→ can't send data in raw form of text

→ plain text

↳ have to convert it into,

transformed text

transform it

↳ known as "Cipher text".

$$WOMK = \frac{2}{3} MCW$$

→ Can't ensure channel
to physically safe
from Date: attacker.

- ① what is 5th most important req.

3) Non-Repudability (NR)

↳ sender should not be able to deny that he has send the msg.

C-I-A-NR

↳ these 4 req. can be ensured by cryptography.

Availability → is req. in service, for eg. medium threat kde koi jisse msg bhejna tha !!
DOS.

→ Information service is one of the services.

→ not a req. in information security.

- ① Authorization and "access control"

↳ these are two very related requirements, cannot explain w/o another.

e.g. ID card milna, jisse can access.

→ purpose of authorization is to enable access control.

→ without defining an protocol by which we ensure that only legitimate entities / authorized entities can access that thing.

- ① Availability, authorization, access control

cryptography has important role in ensuring them, but not sufficient to ensure these three.

2/8/24

Friday

(lec-3)

Cyber Security :-

Book:-

Frozen

Information Security :-(A)
sender
node

m

(B)

Receiver
node(sending single
msg to B.)Requirements :-

→ apart from A & B, no one should know the content of m.

1) Confidentiality

① msg should not be altered / tampered / modified

2) Integrity

② But this is not actual meaning of integrity

↳ practically can't avoid attacker from modifying the content.

Integrity → detection of modification.

↳ actual meaning

↳ agar attacker ne modify ki diya, toh

B detect ki ske, ki modified hai msg.

③ Availability

④ Authenticity (A)

↳ how B will know msg 'A' ne kya, this is authenticity of msg.

C-I - A tried aq.

3) Fundamental requirements of security