Page No.:
Date: / /

assumption for ML algo

↳ know that ∃ rel^n b/w x, and y

but rel^n 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 model.

⇒ jab mathematical fn hai x and y ke beech mein but graphs will be look like random.

↳ then any ML algo won't give any benefit.

→ can't predict 'y' with 'x'.

WJMK

Page No.:
Date:

→ is it NP or not?
↳ not (Yes) How?
can't write non-deterministic poly time algo to solve this.

→ essential property of NP problem?
→ definition of NP problem?
↳ candidate sol^n must be checked in poly time.

to give this sol^n, what to do??

what is sol^n domain here?

→ randomly taken from solution space
↳ binary string set.

so candidate sol^n
↳ any binary string.

→ as candidate sol^n 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 s f^n → depends on size of output, not input.

assumption :-
each digest value is equally likely.
→ means every string is coming with equal probability
↳ only then this problem is Hard!!

such f^ns are
↓
Random oracle f^n

Work ☺

name is : RSA problem.

→ Identify attacker's problem :-
attacker is against encryption scheme
→ attacker want to know (m).
↳ this prblm is hard or not ?

* factors of (n) agar pta, can solve in polynomial time
↳ as will repeat RSA algo steps again.
→ have to perform integer factorization. (IF)

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

RSA assumption is difference assumptions.
(RSA A) assumption.
but If AP related to RSA A assumption
DL A ——— assumption. DH A assumptions

Hash function :-

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

H

$\{0,1\}^l$

SHA → secure hash algorithms

↳ soundness req. of hash fn ??
↳ must be deterministic algo.

with given → find another →
$m \in \{0,1\}^*$, H
$m' \in \{0,1\}^*$.
s.t. $H(m) = H(m')$
↳ what can ensure, this prblm is hard.

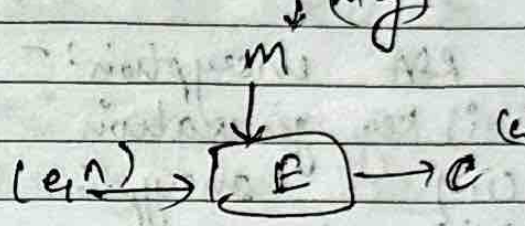
↙ must be deterministic
↳ same input must give same output

WORK ON MM.

Page No.:
Date: / /

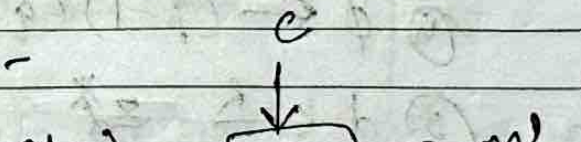
* (1) inverse calculation is also polynomial time.
(2) gcd " " euclid algo.

ENCRYPTION:-



$c = m^e \bmod(n)$

DECRYPTION:-



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

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

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

as $ed = 1$

e, d are inverse of each other.

→ how??

as, $e\phi$ is d^{-1} wrt mod ϕ, but here is mod (n).

$\phi \rightarrow$ euler's fn
↳ bcoz of euler's thm

$m' = m^1 \bmod(n)$

$m' = m$

↓

Correctness proof

this can be written

(3) Soundness of RSA:-

Page No.:
Date: / /

they are related as:-

⇓

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

① 3 algo in RSA encryption:-

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

(*) keyGen algo:-

generate 2 large primes ⇒ ① p ← large prime
② q ← " "

③ n ← (p * q)

④ ϕ ← (p-1)(q-1)

⑤ d ← $\xleftarrow{R}$ Z_ϕ^* (d is random element of Z_ϕ^*)

(d coming from random set → Z_ϕ^*).

btw 1 & (ϕ-1), but must be coprime with ϕ.

definitely mil jaayega in polynomial time ☺

⑥ e ← d^{-1} mod ϕ

wrt ϕ, multiplicative inverse of d.

Isiliye yeh mod ϕ se hua, baaki sare (mod(n)) se honge.

(e, n) → public key
(d, n) → private key

can generate with of help of crypto++

work 30 min.

So DHP's hardness depends on DLP.

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

↓

This argument is wrong.

→ Inference rule ⇒ if → then rule

→ as ⇒

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

Here ⇒

A: DLP is hard

B: DHP is hard.

DHP reduces to DLP.

↳ show

↳ this

↳ VICE-VERSA also true?

→ Security of DH ~~depends on~~ is ensured due to hardness of DL?

↳ True

↳ DH is secure bcoz DLP is hard → False.

↳ ——— as long as ——— → False.

→ DL is hard ——— DH is secure → 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 → both are assumptions, but ...

they are related as:-

⇓

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

① 3 algo in RSA encryption:-
(i) key generation
(ii) encryption
(iii) decryption

(*) keyGen algo:-

generate 2 large primes ⇒ ① p ← large prime
② q ← " "
③ n ← (p * q)
④ ϕ ← (p-1)(q-1)
⑤ d ←$ Z_ϕ^* (d is random element of Z_ϕ^*)

(d coming from random set → Z_ϕ^*).

↳ b/w 1 & (ϕ-1), but must be coprime with ϕ.
definitely mil jaayega in polynomial time ☺

⑥ e ← d^{-1} mod ϕ

↳ w.r.t ϕ, multiplicative inverse of d.

↳ sirf yeh mod ϕ ka hunga, baaki saare (mod(n)) ka hoga.

(e, n) → public key
(d, n) → private key

can generate with of help of cryptoPP

work 3 Sem.

So DHP's hardness depends on DLP.

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

↓

This argument is wrong.

→ Inference rule → if - then rule

→ as →

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

Here →

A: DLP is hard

B: DHP is hard.

DHP reduces to DLP ✓

↳ show

↳ (H/w 22)

↳ VICE-VERSA also true?

22

→ Security of DH ~~depends on~~ is ensured due to hardness of DL?

↳ True

↳ DH is secure bcoz DLP is hard → false.

↳ _____ as long as _____ → false.

→ DL is hard _____ DH is secure → 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 → both are assumptions, but ...

WJMK :)

Page No.:
Date: / /

(4) RSA

↳ SSL have all these 4.

→ OS first must be authentic
with OS → such web browser code ki have
↳ must be authentic

→ If clock time change kro → may be browser mna dete → as ge certificate validity gone.

→ Soundness :- attacker also want to calculate $(g^{ab}) \bmod(p)$ ↳ Diffie Hellmann Problem.

Diffie Hellman's protocol attacks

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

↓ problem of attacker against Diffie Hellmann protocol.

Now ques^n comes whether this problm is Hard or not?
↳ if hard, protocol is sound o/w 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 Discrete log problem (DLP) is hard
so diffe Hellman protocol (DHP) is hard.

Page No.:
Date: / /

(3) $(k_B)^a \bmod(p)$ ⇌ (3) $(k_A)^b \bmod p$

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

→ so $(k_A)^b \bmod(p) = (k_B)^a \bmod(p) = k_{ssh}$ (say)

↑ behaves like symmetric key.

(*) soundness ??

→ firstly identify objective of attacker in this protocol.

↓ want to calculate k_{ssh}.

assumption req :-

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

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

(2) anonymous diffie Hellmann

↳ if without certificate, doing exchange.

→ Combo of both :- (*)

(3) → Ephemered Diffie Hellmann

↳ cost higher as verification bhut vaar!

WORK SHEET.

Page No.:
Date: / /

thus ⟹

$g^{\alpha}: Z_p^* \to Z_p^*$ ⟶ BIJECTIVE FUNCTION !!

$(g^{\alpha} \bmod (p))$ ← modular exponentiation is easy !!!

↓ have library functions to solve it in polynomial time.

↓ inverse exists !!!

$g^{\alpha} \bmod p = \beta$

$\alpha = \log_{g,p} \beta$

g, p → are constants

Stands for "discrete" log.

base 'g' wrt modulo 'p'

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

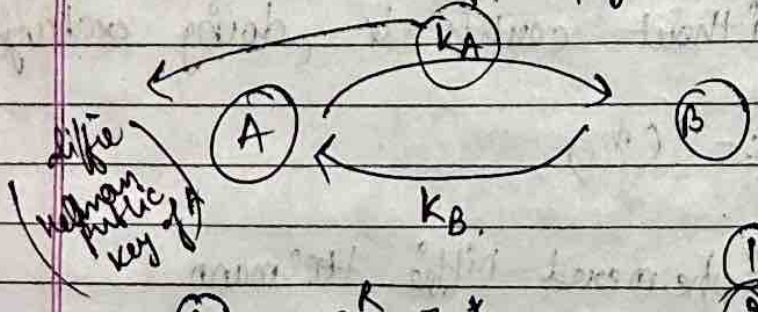
31/Aug/2024 Saturday

LEC-8 :-

(1) fixed.

Diffie-Hellman protocol :-

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



Diffie Hellman public key of A

(1) $a \xleftarrow{R} Z_q^*$

(2) $k_A \leftarrow g^a \bmod p$

(1) $b \xleftarrow{R} Z_q^*$

(2) $k_B \leftarrow g^b \bmod p$

a, b → private keys of A, B.

Page No.:
Date: / /

Ideal, subfield etc.

{ super group ko "group extension" bolte hain.
field — "field" ——————.

⇒ 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 $\langle Z_p^*, * \rangle$

let $(g^\alpha) \bmod (p) = (\beta)$

$\beta \in Z_p^*$
as $\beta \neq 0$
$\beta \in \{1, \ldots, p-1\}$.

ultimately
it
repeats cyclically.

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

so, take $\alpha \in Z_p^*$ also.

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

but this is not order of (β) → but some permutation of Z_p^*.

WORK & IMP.

Page No.:
Date: / /

thus ⟹

g^{α}_{p} : $Z_p^* \rightarrow Z_p^*$ ⟶ BIJECTIVE FUNCTION !!

$(g^{\alpha} \bmod (p)$ ← ∴ inverse exists !!!

modular exponentiation is easy !!!

↓ have library functions to solve it in polynomial time.

$g^{\alpha} \bmod p = \beta$

$\alpha = \log_{g,p} \beta$

g, p → are constants

Stands for "discrete" log.

base 'g' wrt modulo 'p'

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

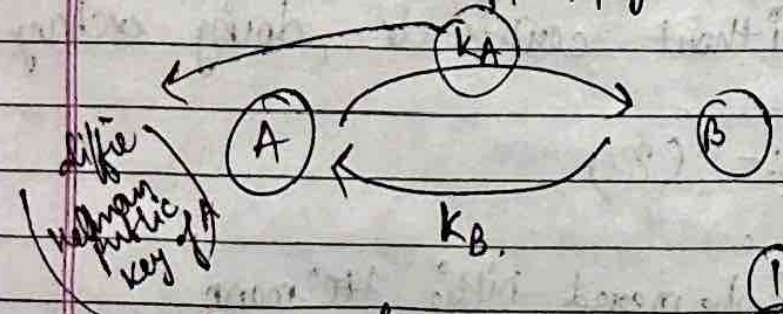
31/Aug/2024 Saturday

lec-8 :-

(1) Fixed.

Diffie-Hellman protocol :-

p, q, g → global parameters known to all (even, may be to attacker also)



diffie hellman public key of A

(1) $a \xleftarrow{R} Z_q^*$

(2) $K_A \leftarrow g^a \bmod p$

(1) $b \xleftarrow{R} Z_q^*$

(2) $K_B \leftarrow g^b \bmod p$

a, b → private keys of A, B.

Page No.:
Date: / /

like subfield etc.

{ super group ko "group extension" bolte hain.
field — "field" —— .

⇒ 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 $\langle Z_p^*, * \rangle$

let $(g^\alpha) \bmod (p) = \beta$

$\beta \in Z_p^*$
as $\beta \neq 0$
$\beta \in \{1, \ldots, p-1\}$

ultimately
it
repeats cyclically.

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

so, take $\alpha \in Z_p^*$ also.

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

but this is not order of β → but some permutation of Z_p^*.

Page No.:
Date: / /

thus ⟹

$g^x \bmod(p)$ ← g^x_p : $Z_p^* \to Z_p^*$ ⟹ BIJECTIVE FUNCTION !!

⇓ inverse exists !!!!

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

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

$\alpha = \log_{g,p} \beta$

g, p → are constants

Stands for "discrete" log

base 'g' wrt modulo 'p'

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

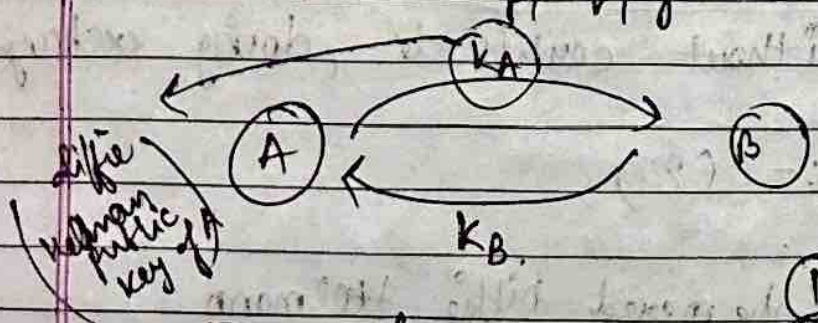
31/Aug/2024 Saturday

LEC-8 :-

① Fixed.

Diffie-Hellman protocol :-

p, q, g → global parameters known to all (even, may be to attacker also)



① $a \xleftarrow{R} Z_q^*$

② $k_A \leftarrow g^a \bmod p$

① $b \xleftarrow{R} Z_q^*$

② $k_B \leftarrow g^b \bmod p$

a, b → private keys of A, B.

WINK :)

Page No.:
Date: / /

Ussy subfield etc.

{ super group ko "group extension" bolte hain.
field — "field" ——————.

⇒ 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 $\langle Z_p^*, * \rangle$

let $(g^\alpha) \bmod (p) = (\beta)$

$\beta \in Z_p^*$

as $\beta \neq 0$

$\beta \in \{1, \ldots, p-1\}$.

ultimately it repeats cyclically.

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

so, take $\alpha \in Z_p^*$ also.

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

but this is not order of (β) → but some permutation of Z_p^*.

WORK SHEET.

Page No.:
Date: / /

this smallest 'i' → is 'order' of a.
In Z_p^*, we'll have only one i s.t. $a^i \bmod(p) == 1$.

$G = \langle Z_p^*, * \rangle$, $p = 11$

↳ order of $G = |G| = 10$.

as mod(11) ke rahe

as $G = \{1, 2, \ldots, 10\}$

① let $a = 2$, $p = 11$

$i = 0 \rightarrow 1$
$1 \rightarrow 2$
$2 \rightarrow 4$
$3 \rightarrow 8$
$4 \rightarrow 5$
$5 \rightarrow 10$
$6 \rightarrow 9$
$7 \rightarrow 7$
$8 \rightarrow 3$
$9 \rightarrow 6$
$10 \rightarrow 1$

order of 2 = 10.

subset, 2 is generating is → {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} = Z_p^*.

* Primitive root ?

* what is Generator ?

↳ element which generate entire group like '2' here is a generator.

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

known as "SUBGROUP"

subset of main group which in itself is a group.

Work is worship.

Page No.:
Date: / /

(*) Non-repudability can be achieved only by Digital Signature → public key cryptography, can't be achieved by symmetric key cryptography.

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

2nd Hard Problem of Public key Cryptography :-

Discrete Log Problem

Want smallest i s.t. $(a^i) \bmod(p) = 1$,

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

$a^3 = a^2 \times a$
$= 3 \times 5$
$= 15$
↳ mod 11
$= 4$

let, $p = 11$, $a = 5$

then :- $(a^i) \bmod(p)$

for i ⇒
0 → 1
1 → 5
2 → 3
3 → 4
4 → 9
5 → 1
6 → 5
7 → 3
8 → 4

whatever element u take inverse this way !!

OBSERVATION :-

(1) so, $a^i \bmod(p) \in \{1, 5, 3, 4, 9\}$

(2) we'll have 'ꝏ' i's s.t. $a^i \bmod(p) == 1$. But want smallest i.

→ And we are looking in Z_p^*. → so that $i = 0$ answer na ho.

100MK ≡ 5M.

$Z_p = \{0, 1, \ldots, p-1\}$

$Z_p^* = \{1, \ldots, p-1\}$

① $\langle Z_p, +, * \rangle$ is a finite field
↳ with identity of one operator, we did not require the inverse of another operator.

$Z_p, + \rightarrow$ group ✓

$Z_p, \times \rightarrow$ group ✗

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

③ symmetric cryptography is faster (so much → almost 10 times) than public key cryptography but both are equally secure.
↳ why is public key cryptography used then?

⇓

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

↓

that's why speed ↓.

Hybrid CRYPTO-GRAPHY

costly! ↳ but only once!!

to encrypt symmetric key !!

used to encrypt data!

To establish secure communication b/w sender & receiver through open/insecure channel

RSA → 128 bits of data can encrypt by this costly algo.

↓

to that symmetric key encrypt ho kar jaaye through open channel, and public key a jaiya, to jiske paas private key hogi, bas wo decrypt kr paayega → so for initial secure establishment → public key is required.

WORK ☺ ☺

Page No.:
Date: / /

23/Aug/Friday

Lec-7 :-

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

order(G) = $|S|$ = cardinality of S

let ⇒

$G = \langle Z_n, \oplus \rangle$, $Z_n = \{0, 1, \ldots, n-1\}$

↳ this is mod(n).

↳ 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 $\langle Z_n, * \rangle$ is also group.

↳ No!

let n = 10.

2 ka inverse does not exist.

↳ Numbers which are co-prime with 'n', sirf unka inverse milega. bs!, baakiyon ka nahi!

Z_n^* is that subset.

↳ $\in Z_n$, but are also co-prime with n.

↳ satisfies closure property ✓

associativity ✓

Identity exist ✓

Inverse exist ✓

so, $(Z_n^*, *)$ is a valid group. (MG)

(multiplicative group)

WOMK 3 ♡♡.

Page No.:
Date: / /

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

↳ h → predetermined, 1-8 bit value.

and it is possible as it is many-to one function.

↳ as (n >>> l) (:)

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

↳ so want this problem of finding such m' to be a hard problem.

→ so want to design hash f^n s.t. above pblm hard ho.

↳ 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.

↳ haan chahe it generates random string of bytes (essa kyu hai), but for same m, H(m) must be same.

WORK ☺

→ Why attacker modifies file, can modify hash only ☺, tab tak toh detect bhi na ho paata ki attack huya → but we assume data and hash ikattha chlta hai.

Authenticity w/o Integrity has no meaning (???)

→ SHA → secured version (??)

① Encryption properties:- (or Requirements)

1) Correctness:- ensuring that Encryption would work correctly when done.

2) Soundness :- ??

→ In encryption - Decryption:-

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

↳ Hard Problem.

① Soundness req. of hash:-

If msg changes, hash must also change.

Integrity → if msg is modified; it must be detected.

→ In hashing, prblm of attacker is computational problem.

Given: m, h, Hash algorithm also known

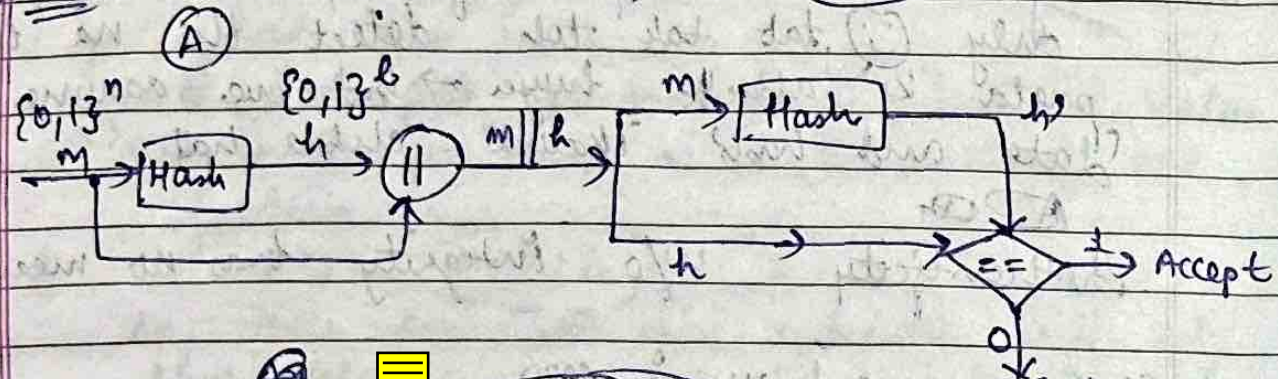
Compute: ? → Prblm of attacker.

↳ m' (as m is passed over insecure channel)

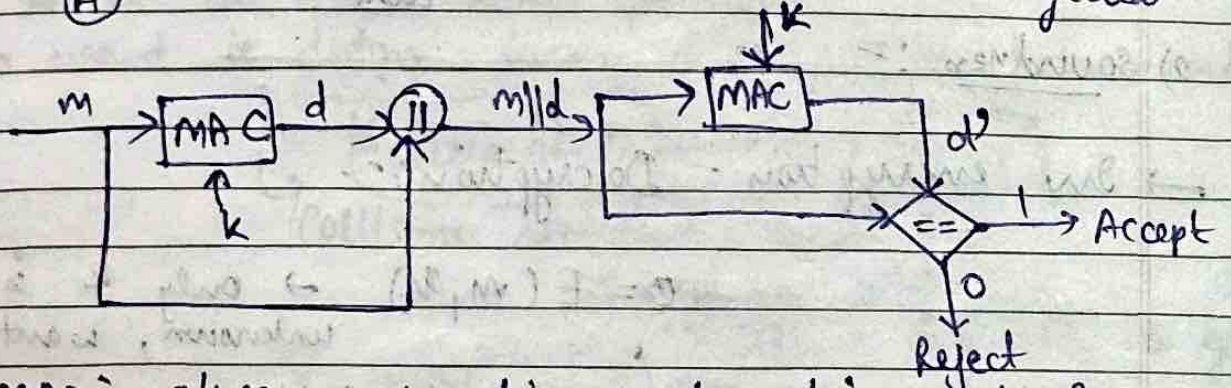
s.t. $H(m) = H(m') = h$

Page No.:
Date: / /

① Hash:- sender. $(n >>> l)$ — Integrity

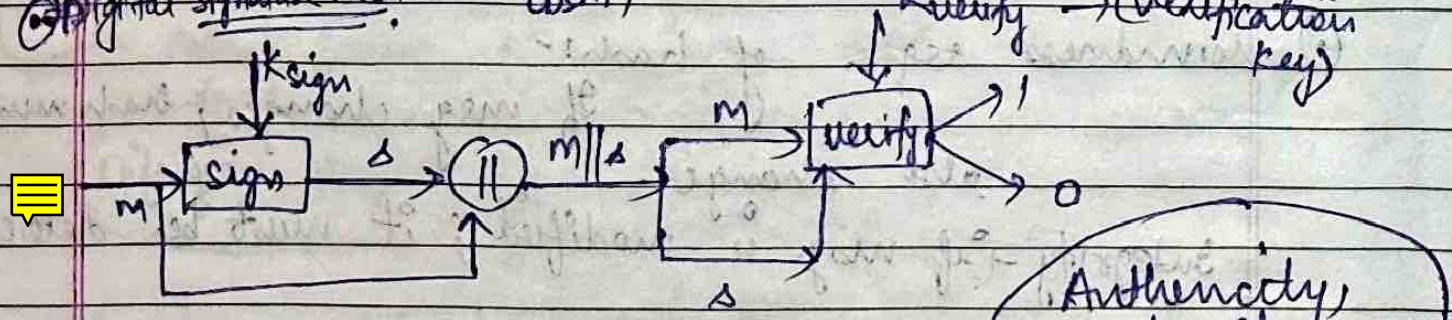


② MAC :- sender — Authenticity & Integrity



MAC is always symmetric cryptographic mechanism.

③ Digital Signature Mechanism (DSM)



Authenticity, Integrity, non-repudability

↳ How? (H/w).

K_{sign} = Inverse (K_{verify})

K_{verify} = Public key A

K_{sign} = Private key A

So, DSM is always asymmetric cryptographic mechanism.

to ensure in poly time, have to take constant numbers & WJMK ∈ 50 / 50 ... ∈ ...

Page No.:
Date: / /

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

⇒ Both functions are used in cryptography, (NP sol^n) → from sol^n space → eka random no. utha ske the → check kiya sol^n H(first) = H(that) ... not → kiya ... vaale constant

② **Integer Factorization Problem** :-

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

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

given :- n
find / compute : p
such that : $p | n$

} → integer factorization cryptography problem ↓ If known 'n' is exactly product of 2 prime numbers.

Q. If decision problem :-
given :- n
decide if :- n is prime

ans Hard / Easy ???

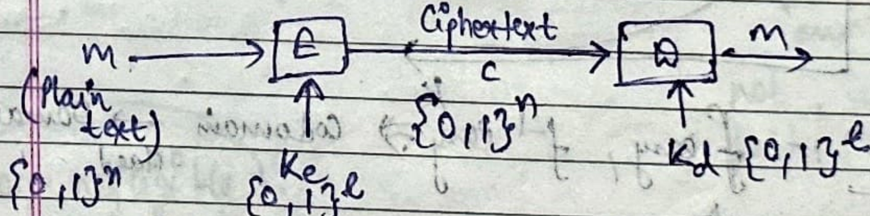
16 Aug 2024
Friday

lec-6 :- (Block Diagrams :-)

Encryption-decryption :-

sender (A) — Reciever (B)

K_e = Inverse (K_d)



n, l are almost same ???

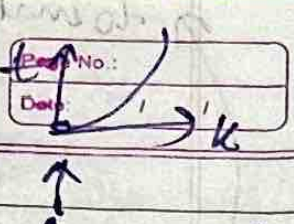
Confidentiality

$C = E(m, K)$ — only unknown is K

for symmetric encryption :- $K_e = K_d$

K_e :- public key$_B$ [redacted]
K_d : Private key$_B$ — K : Shared secret key

work in progress.



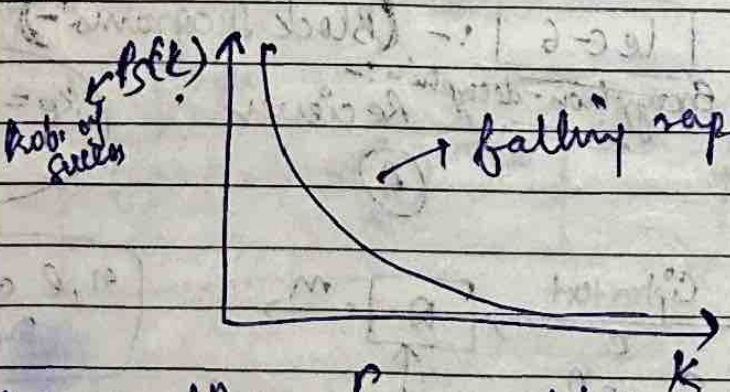
① If growth rate of f_n is very high (exponential), then f^n is non-negligible (????)
→ attacker is an algo
problem for attacker → opposite wale ka security protocol etc. to find private key etc.

we want problem of attacker should be Hard (for our safety) but want hashing, mapping, encryption etc to be hard (o/w maybe problem, ab cryptographic algo must be easy)

security properties
Correctness (completeness) — Soundness

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

N Queen Hard only if N is large, not for N = 2, 3 etc.

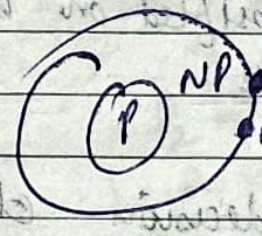


② One-way f" :- f easy, f^{-1} easy ⇒ Codomain → Domain Hard
Domain → Codomain easy

③ Trap door f" :- sat can enter easily, and only if you want then sat can come out easily, o/w hard.

work ☺

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



P1 ; P1 ; P2 mein reduce ho jaayega.

→ Hard prblm which is NP ☺ may not be NP-Hard".

→ any hard problem ∈ NP or NP-Hard or NP-complete.

↓ deterministic.

All have no prblm with poly. TC sol"/algo.

Q. why not all hard problem are NP??

→ In CS, we need NP type of problems.

NP → have not deterministic Poly TC, (but must have non-" " " algo to solve!!

with non-negligible probability of success

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

↓ ans may be correct/not, may return ans/failure.

non-negligible???

↳ It is not threshhold, it is function of input size (k) (no. of bits).

→ function, which is not negligible → non-negligible fn.!

eg. Quick sort with random pivot element / Heuristic AI algo → are non-deterministic.

~~Algo~~ → Problem.

can be classified on basis of purpose

- Optimization → e.g. getting shortest distance
- counting
- search
- decision ek.

Easiest problem

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

↳ (0-1) problem.

also known as

P1 is reducable to P2 ⇒ if P2 is easy then P1 is easy.

↳ P1 is not more harder than P2.

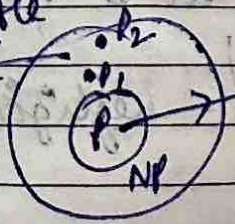
Nondeterministic decision algo

- LV → Las Vegas
- MC → Monte Carlo

always gives answer, may / may not be correct.

- Yes biased
- No biased

P2 reducable to P1.



→ as more (fat), hardness ↑s.

Page No.:
Date: / /

WORK ☺

⇒ If P_2 is easy, then P_1 is easy.
⇒ But if P_2 is hard ⇏ P_1 is hard.
⇒ P_1 hard ⇒ P_2 hard.

(P_2 hard as long as P_1 is hard.)

→ If P_1 reduces to P_2. (???)

WORK ☺

9/8/24 Sunday

Lec-5 :-

Problem
- Easy
- Hard

Algorithm
- Deterministic
- Non-deterministic / Probabilistic

(1) nature of algo is deterministic

i) means same input se same output milega.

ii) ~~They case there is no source of~~ randomness in algo/prblm (???) → (like generating random no
same input
↳ diff output possible
↳ Non-deterministic)

iii) Flow chart bnta hai algo ka, toh same input ke liye, path from start to end is constant/same everytime, so can determine path everytime, but in non-deterministic can not judge/guess ~~the~~ path as it is not fixed.

→ In A-I, mostly we use Deterministic algo.

eg. Quick sort with random pivot everytime / Heuristic AI algo → are non-deterministic.

Algo → Problem.

can be classified on basis of purpose

- Optimization → e.g. getting shortest distance
- counting
- search
- decision ok.
  - easiest problem
  - as any input but output is either "Yes"/1 or "No"/0.
  - (0-1) problem.
  - also known as

P1 is reducable to P2 ⇒ if P2 is easy then P1 is easy.

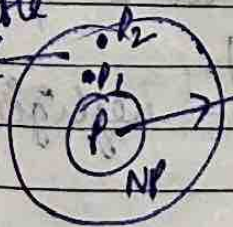
↳ P1 is not more harder than P2.

Nondeterministic decision algo

- LV → Las Vegas
- MC → Monte Carlo
  - always gives answer, may/may not be correct.
  - Yes biased
  - No biased

as more fail, hardness ↑s

P2 reducable to P1.



WORK ☺

⇒ If P_2 is easy, then P_1 is easy.
⇒ But if P_2 is hard ⇏ P_1 is hard.
⇒ P_1 hard ⇒ P_2 hard.
⇓
(P_2 hard as long as P_1 is hard.)

→ If P_1 reduces to P_2.
(???)

WORK ☺

9/8/24 Friday

Lec-5 :-

Problem
- Easy
- Hard

Algorithm
- Deterministic
  - ① nature of algo is deterministic
    - i) means same input se same output milega.
    - ii) ~~They case the source of randomness in algo/prob~~ There is no (???)
    - iii) Flow chart bata hai algo ka, toh same input ke liye, path from start to end is constant/same everytime, so can determine path everytime, but in non-deterministic can not judge/guess the path as it is not fixed.
- Non-deterministic / Probabilistic

→ (like generating random no
↳ same input
↳ diff output possible
↳ Non-deterministic)

→ 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 than, P_1 reduces to P_2,
↳ what do you mean by this??

⇓

($P_1 \leq P_2$) ??

using sol^n 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 reducable to P_2 in polynomial time

↓

means?

like unweighted ko pehle weighted mein karenge, then f^n of weighted wale ka use karenge, then again kyi vaar kuch karenge.

↳ Joh f^n ka use karne se pehle kuch kiya, then f^n call ke baad kuch kiya → in dono mein jo time lga ⇒ poly. must.
and f^n call kiya → may be ek se jyada vaar ke she hawn → must call poly no. of times (P???)

Page No.:
Date: / /

Work & Stuff

in arr[ ] → n = no. of elements in array
↳ as assuming every element is taking constant space, so we ignore that "1".

But here input size → not constant.
↳ It's not like, 2 line ka code hai toh easy !!

* what is Easy and what is hard problems ???
↳ problem which cannot be solved in polynomial time complexity.
↳ soln exists but does not have (TC = polynomial). (may)
↳ which can be solved in poly. time.

O(1)
O(log k)
O(k)
O(k²)
etc.

built in future it may have !

$2^{\log k}$ → subexponential (> poly, < exp)
2^k → exponential
↳ 8 puzzle
↳ N queen
↳ graph coloring problem

Q. What is NP, NP Hard, NP complete problem ?
↳ all are 'os', unbounded sets !

→ aage se jitna dur jaayoge, difficulty level utna badhega.

Work ☺

Page No.:
Date: / /

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 than, (P_1 reduces to P_2)

↓ ↳ what do you mean by this??

($P_1 \leq P_2$) ??

using soln 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)

(*) P_1 is reducable to P_2 in polynomial time

↓ means ?

like unweighted to pehle weighted mein karenge ; then fn of weighted wale ka use karenge, then again kyi waar kuch karenge.

↳ Toh fn ko use karne se pehle kuch kiya, then fn call ke baad kuch kiya → inn dono mein jo time lga ⇒ poly. must.

and fn calls kiya → may be ek se jyada baar ke sthe hoon → must call poly no. of times (????)

WOMK 2/34 ♡

Page No.:
Date: / /

in arr[ ] → n = no. of elements in array
↳ as assuming every element is taking constant space, so we take ignore that "

But here input size → not constant.
↳ It's not like, 2 line ka code hai toh easy !!

* what is Easy and what is hard problems ???

hard ↳ problem which can not be solved in polynomial time complexity.
↳ soln exists but does not have TC = polynomial. → (may) → but in future it may have!

Easy → which can be solved in poly. time.

- O(1)
- O(log k)
- O(k)
- O(k^2)
- etc.

$2^{\log k}$ → subexponential (> poly, < exp)

2^k → exponential
↳ 8 puzzle
↳ N queen
↳ graph coloring problem

Q. What is NP, NP Hard, NP complete problem ?
↳ all are ∞, unbounded sets!

→ centre se jitna dur jaayoge, difficulty level utna bdhega.

Work ☺

Page No.:
Date: / /

Public key → key which is not supposed to be secret to somebody.

Symmetric key → shared secret key (???)
↳ where no body can be sender, only one who knows your secret key can be sender.

But wait....
Private key = inverse (public key)
↳ toh Private kaise nahi ho?
where is secrecy of private key, then?
→ Algos are public always.
↳ to generate keys.
↳ to get k_2 from k_1.

→ 2 options:- → (1) Inverse does not exist.
(2) You can't calculate Inverse!!
↳ It's (hard!)
↳ It exist, but difficult.

→ Not possible!
→ as one-one hoga pkka!
↳ ek private key ek public key se hi map hogi.
→ onto bhi hogi pkka!
So, Inverse Exists.

what do you mean by it?

→ Hard can be problem, not algo.
→ TC " " of algo, not of problem.
↳ N → size of input (in no. of bits)
↳ bits req. to hold input data.

WORK 3

encry → purpose only 'confidentiality'

Digital Signature
↳ ensures authentication, non-repudiability, integrity (NI)

⇒ 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 inverse(key) = key, o/w
Assymetric key cryptography / assymetric cryptography.
↳ so (hash) → is neither symmetric nor assymetric,

Digital sig → always Assymetric
MAC → " Symmetric
encry-decry → can be " as well as Assymetric

eg: MD5, SHA are examples of 'hash' algo.

Assymetric key cryptography → also known as
Public " ".
↳ as, K_2 should be secret to Receiver,
but kya K_1 will be " " 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.

eg: $(\mathbb{Z}, +, *)$ is a ring, but not a field.
↳ multiplicative identity exist,
but " inverse does not "

In ring, can perform +, −, *, not ÷ / as multiplicative inverse does not exist.

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

→ 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 inverse of no. is no. itself.
→ Let set → $\{0,1\}^n$, binary operator → ⊕.
↳ binary strings of fixed length.
↳ Is it a group ?!
↳ Yes!!
identity element = 0000...0
↳ n length
inverse (element) = element itself.

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

WORK is worship. we know inverse of

Page No.
Date:

(6) What is inverse of an element?

⇒ 3

↳ multiplicative inverse → (1/3)

↳ additive inverse → (-3)

5/8/24 Monday

Lec-4

WORK is worship ♡♡.

Group : < set , binary operator (*) >

S ↓ a b ↓ ↓ (2 input)

a, b ∈ S.

f(a,b) = c

(1 output always as it is fn)

Group must satisfies :-

(1) closure i.e. c ∈ S.

(2) Associativity (3) Identity (4) Inverse.

∃ e ∈ S s.t. a * e = e * a = a ∀ a ∈ S.

∃ b ∈ S s.t. b * a = a * b = e ∀ a ∈ S.

→ If group satisfies commutative property, then it is (a * b = b * a)

abelian group.

→ field :- → 2 operator → usually +, *.

↳ group wrt both operators

→ Ring :- more than group, less than field.

↳ multiplicative inverse is missing in the ring.

+, * → fundamental operators.

÷, / → are not fundamental operators, as ÷ is nothing but * with multiplicative inverse

WJMK 30 Oct.

Encryption key (K_1) | Decryption (K_2)

(*) key is req. for both encryption & Decryption.

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

to ensure that only 'B' ki 'key' hi use decrypt kr paaye!

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

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

- E → (Encryption f^n)
- $\{0,1\}^n$ → plain text
- $\{0,1\}^l$ → key
- $\{0,1\}^n$ → cipher text

↳ binary f^n (2 input f^n)

↳ 1st input: n bit binary string

2nd ———— l ————

output : n ————

$E(m, k_1) = c$ $m \in \{0,1\}^n$

$D(c, k_2) = m$ $k_1 \in \{0,1\}^l$

k_2 → receiver's private key $c \in \{0,1\}^n$

k_1 → somehow related to k_2.

↳ so this is not random string of l bits.

k_2 can generate randomly, but k_1 have to compute from k_2

OR

k_1 can generate randomly, but then k_2 have to compute from k_1.

so ∃ relationship → 'inverse' of each other, as ek ka effect on plain text, dusre ne kaatna hai, means usko firse plain-text bnaana 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.

(*) Are these algorithms (Encryption / decryption) are secret or Public?
↳ Public, obviously !!!
So attacker can also use them, and cipher text channel se leke plain text nikaal skta hai
⇒ this defn is not complete.
→ decryption must req. some additional info (Secret piece of info) to convert cipher text to plain text.
↳ must not be known to attacker!
known as 'key'
↳ also string of bits
↳ this must be random sequence of bytes
↳ thus human can't remember such sequence of almost 16 bytes.
→ key is hash of Password.
↳ Password ⟶ different from
↳ seq. of characters
↳ jo binary form mein store hota hai in memory.
↳ must be human readable and remembrable
↳ key created/generated by computer with some algorithms.
↳ We create/generate.

(algo, protocol → they may/may not standard) work ☺.

ASCII → standard, Not algo, not protocol.

Cryptography :-

↳ algo to ensure :- C-I-A-NR.

↳ eg: SHA, cipher, MD5, RSA, TLS

↳ not an algo, it is standard of protocol (SSL)

• Diff b/w algo & protocol.

algo → can be executed in one system. (part of protocol)

protocol → can't be executed on one system, atleast 2 entities are there, 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 Assymmetric (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

eg. msg → string of bits (streams)

→ can't send data in raw form of text

↳ plain text

↳ have to convert it, transform it

↳ known as "Cipher text".

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

(5) what is 5th most important req.

3) Non-Repudiability (NR)
4 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 khraab kde koi jisse msg bhejna tha !!
DOS.

→ Information service is one of the services.
→ not a req. in information security.

(6) "Authorization" and "access control"
↳ these are two very related requirements. Can't explain one w/o another.
eg. ID card milna, jisse can access.

→ Purpose of authorization is to enable access control.
↓
protocol by which we ensure that only legitimate entities / authorised entities can access that thing.

(7) "Availability, authorization, access control"
↓
cryptography has important role in ensuring them, but not sufficient to ensure these three.

WOMK 3 OM.

Book:- Frozen

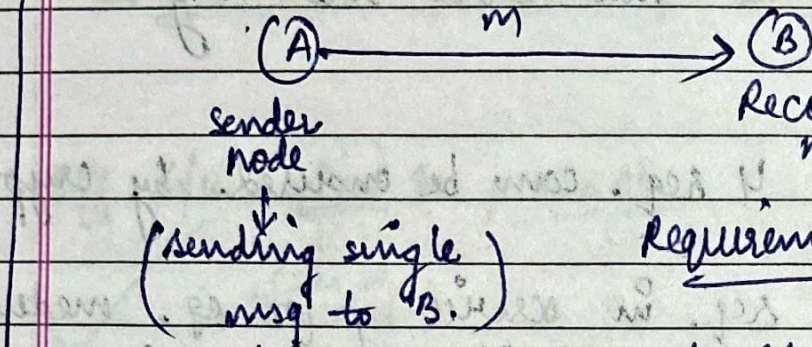
Page No.:
Date: / /

2/8/24
Friday

Lec-3

Cyber Security :-

Information Security :-



Requirements :-

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

1) Confidentiality

msg should not be altered / tempered / modified

2) Integrity

But this is not actual meaning of integrity

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

Integrity

↳ actual meaning

↳ agar attacker ne modify kr diya, toh B detect kr ske ki modified hai msg.

3) Availability

4) Authenticity (A)

↳ How B will know msg 'A' ne bheja, this is authencity of msg.

C-I-A triad :-

↳ Fundamental requirements of security