26.07.24

CS

TLS : Transport layer security
Cryptography : above transport layer.

IDS ← signature based
application based.

→ 6 trillion dollar economy
→ Dark web – illegal activities, Denial of service (DOS) attack.

Deepfake technology } Generative AI
Audiofake technology }

Crowd strive issue affected banking & airlines.

Internet → Client-server architecture [Web Dev] → Data Centers [Home grown apps ERP] → Cloud Technology

Each new technology brought with it new threats.

→ Generation I Problems → Virus (late 1980s)
→ 1st antivirus solution created.
→ Melisa & ILU virus (started in Vietnam)
→ Gen II (Network) → Firewall developed to identify & block cyberattacks over the Internet; ARPAnet project.

Pakistan virus : created a dent on the hard-drive using disk platter. (physical damage)

ARPA net : conducted a CRC check [no antivirus / other attacks]
to verify data is not
corrupted at the payload level.

Buffer overflow led to DOS attack & no service available.

→ Gen III (Application) → exploitation of vulnerabilities within apps.
yahoo, hotmail, reddit
Still a problem till date
led to adoption of Intrusion Prevention Systems (IPS)

(55 million websites affected by malicious entities)
→ Phishing

Netizens : role / responsibility over the net (cyber space)

→ Gen IV (Payload) : metamorphic malwares
* malware became more targeted & evaded signature-based defenses, anti-bot & sandboxing solutions became necessary.

→ Gen V (Mega) : large scale, multi-vectors

Damaging LLM, data poisoning at the model level.

Almost 2 months to detect malicious code (worms etc.)

Bubble gap : no path to enter the network.

Stuxnet worm ↳ activated every 27th day & increased speed of centrifuge to 600 - 1200 Hz.

Ransomware attack:
Scanning → Penetration → Execution of attack.

HIS: Hospital Information System affected

Cryptocurrency + Cyber Ransomware attaks
Co-ordinated criminal attacks.

Guiding Principle ⟹
① Defence in depth: a single-layer protection is not enough.
Layered approach of security.

Data can't be protected just by encryption.

No way out to bypass defence in Depth.

② No absolute security: no foolproof system

Tradeoff b/w productivity & security.

Future Threat:
supply chain attacks
disinformation campaigns by bots — infected traffic becomes legitimate & legitimate is attacked.
malwares metamorphosis

29.07.24

How to measure security exposure of an organisation?
→ how many times firewall is blocking unwanted exposure (false +ve / true -ve)

Guiding principle : no concept of absolute security.
(i.e. no 100% foolproof)

something that can't be measured → can't be secured.

* Transmission loss in network : in terms of power

Defense in depth : layered approach of security
accepting risk at every level

Port 443 → for https traffic.

Complexity → worst enemy of security

* Which guiding principle is used in a particular scenario?

Threat landscape & Motivation
Attack sophistication v/s Intruder Knowledge

CIA Triad
→ Confidentiality
→ Integrity
→ Availability

→ evolutionary

* Cyber Kill Chain : strategies used by attackers

All 3 aspects of CIA are compromised in this attack.

① Infiltration / Penetration stage → Firewall

Metamorphism : headers change every time
spoofed IP : link containing malicious

② Propagation : Script installed could be virus / worms
could be spreads to all land & sub-land
↓
reach out to no of devices & get data.

③ Attacker is aware of firewall
↓ tries to collect data from all computers to a single device
Aggregation

Attacker will install a Trojan worm & export the data to the computers

④ Exfiltration

Damage control is essential to protect data from exposure

Cyber kill chain. [code from Github – 14 diff. stages]

① Reconnaissance : attacker gathers information without getting notified. (firewall, Co-router or honeypot)

② Weaponisation : Preparing the attack : collecting info. like port no. (technical info)
Eg. creating malware

Attacker finds a pathway
into entering the network infrastructure & designs an attacking weapon. [worm propagate]

can't measure → can't control → can't secure

③ Delivery : delivering attack eg. phishing email,
④ Exploitation : exploiting vulnerabilities to gain access.
⑤ Installation
⑥ C2 – Command & Control
⑦ Actions on objectives

Measurement Issues

→ General metrics : no. of incidents, mean time to detect (MTTD)

event as an incident → compliance as law
Privacy law : GDPR → high degree of prudence

→ Specific Metrics :
Compliance metrics (PCI DSS, HIPAA, GDPR)
network security metrics
application security metrics.

→ Advanced Metrics :
security posture, risk assessment scores,
Threat Intelligence metrics.

→ Key Performance Indicators (KPIs)

Relevant to senior management

Top level management is keen to know about KPIs.
→ security Incident Reduction
→ cost of security incidents
→ User behaviour metrics

CIA Triad: goal is to implement, operate, monitor, review, maintain & improve a documented ISMS.

↓

To measure the properties of information.
→ most scientific way to secure information.

Confidentiality: Info. is intended for a specific purpose & if leaked → Security breach.
controlled by access control
[Top secret document eg. blueprint of missile etc.]
Then confidentiality is compromised.

Integrity: data changed / contaminated even by a single bit.
checked / prevented by Hashing: CRC, checksum.

Availability: website is down for a few hours.

↓

can it be measured as confidentiality? May be.
But mainly availability element.

website is defaced → Integrity.

Develop a redundant web server.
If N/W link is down → can't access web server.
(Imp. people can access info. through access review control)

DMZ: De-militarised zone
↳ attacker can't enter.

02.08.24

INFORMATION SECURITY

① C Confidentiality requirement : Content of the machine should not be accessible by anyone other than sender & receiver. (Info. should not be hampered)

② I Integrity : Receiver should be able to detect that the msg. received is not the original msg.

③ A Authenticity : msg. received from a legitimate sender.

④ Non-Repudiability : Sender can't deny that it had sent a msg. (Achieved using digital signature only)

Above 4 → Cryptography (alone is not sufficient).

↓

Set of algorithms that ensure above 4 properties.

① Availability : DOS attack → Attack on the availability of service.

② Authorization & ③ Access control

go hand-in-hand [can't define one without another]

↓

Purpose is to enable access control

↓

protocol using which you enforce that only legitimate / authorized entities can access.

Cryptography : set of algorithms to ensure security requirements.

Algorithms : SHA, Cipher, MD5, RSA / TLS: Transport layer security
↓
Protocol
[not an algorithm]

Difference between protocol and algorithm
protocol → interaction b/w atleast 2 entities
algorithm → executed in one system → part of protocol

SSL : Secure Socket Layer
↳ TLS is a standard version of SSL.

Cryptographic algorithms categories : (purpose point of view)

① Symmetric v/s Asymmetric. [mathematical point of view]

② → Encryption - Decryption (confidentiality only) (2 diff algo)
→ Hash } (same algo in both S & R)
→ MAC }
→ Digital signature → assures part of doc is not changed (2 diff algo)
↳ (authenticity, integrity & non-repudiation)
↓
can't deny I signed
signed by me
confidentiality is not ensured in digital sign.

Encryption
4 bits → 1 character
Data = 16 bytes
(S) → (R)
encrypted & transmitted

Raw data → Plaintext (128 bits)
↓
to be converted into
Ciphertext.
(128 bits) = 16 bytes.

Decryption : used by receiver to obtain plain text from cipher text.

usually size of plain text = ciphertext.

Encryption algorithm → program/software
Decryption → public Then how is it safe?
∴ Decryption algo. should have a secret piece of info. that is only known by receiver [must not be known to attacker]

This secret info. is key.
↳ sequence of bits.

Key & password are not the same.
Key → can't be random [as it must be remembered by human]
must be a random sequence of bytes

Key → Hash of password.
↳ has to be protected from the storage.
generated by computer algorithms.

Password: generated by human.

Key required for both encryption & decryption.
↓
Why key mandatory for encryption?

Let encryption & decryption keys be denoted by K_1 & K_2

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

encryption → binary function
n-bit binary string [Plaintext]
l-bit binary string [Key]
n-bit output [ciphertext]

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

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

$D(c, k_2) = m$

$k_1 \longleftrightarrow k_2$ (related)

k_1 generated randomly but k_2 must be computed from k_1.

Effect of k_1 is getting cancelled by k_2

$\Rightarrow$ k_2 performs inverse of operation performed by k_1

$\therefore$ k_1 & k_2 are inverses of each other.

Inverse → bijective func.
of func.

Inverse of element → additive / multiplicative.

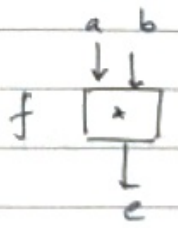
Subtraction: addition of additive inverse
Division: multiplication of multiplicative inverse.

05.08.24

INVERSE

GROUP: $\langle S, * \rangle$

S → set
$*$ → binary operation (is a function)



$a, b \in S$
$f(a,b) = e$
$e \in S$

- closure $f(a,b) = e$
- ~~add~~ associativity $(a*b)*c = a*(b*c)$
- identity $e*a = a*e = a \quad \forall a \in S, e \in S$ → identity element
- inverse $\forall a \in S, b \in S$ s.t. $a*b = e$

all groups are not abelian. If commutative → abelian

additive group

✓ $\langle \mathbb{Z}, + \rangle$ ✓ $\langle \mathbb{R}, + \rangle$ ✓ $\langle \mathbb{Q}, + \rangle$

✓ $\langle \mathbb{Q}, \times \rangle$

✗ $\langle \mathbb{Z}, \times \rangle$ ✓ $\langle \mathbb{R}, \times \rangle$ ← multiplicative grp.

what is a field? ring?

$(+, *)$

set with 2 operators, not both operators if you can define group

Infinite [∵ set is ∞]

$\mathbb{Z}, +, *$ → field ✗

$\langle \mathbb{R}, +, \times \rangle$ → field ✓

$\langle \mathbb{Q}, +, \times \rangle$ — ✓

Ring: same as field but existence of multiplicative inverse is not mandatory

In additive group → only + & −

multiplicative → * & ÷

field → all 4

ring → +, −, * (no ÷)

Exponentiation: repetitive multiplication

[wherever * allowed, it is allowed]

For cryptography → finite group / finite ~~ring~~ field required

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

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

$E(m, k_1) = c \qquad D(c, k_2) = m$

$k_1 = Inv(k_2)$

Decryption : inverse operation of Encryption.

set of all binary strings of fixed length

$n = 8 \rightarrow$ 1 byte

Size $= 2^8 = 256$

8-bit string $\oplus$ 8-bit string $\rightarrow$ 8-bit string.

$\oplus$ = XOR

$\therefore$ Closure ✓

Identity element → 00000000

Inverse of element → same element.

$G = \langle \{0,1\}^n, \oplus \rangle$ (Galoy's Field)

$\therefore$ inverse is element itself $\Rightarrow K_1 = K_2$.

$\therefore$ Inverse of key is the key itself.

SYMMETRIC CRYPTOGRAPHY

Always a combination of 2 algorithms [Sender + Receiver]
[Rarely → same algorithms on both sides]

Scheme : set of algorithms in cryptography.

Hash: Does not require a key

↓ Other 3 classes (E&D, MAC, DS) require key.

Keyless cryptographic algo.

2 classifications

Purpose

- E&D
- MAC
- Hash
- Digital sign.

Method

- Symmetric [MAC]
- Asymmetric [Digital signature]

Hash?

E&D may be symmetric or asymmetric.

secret → symmetric
private → asymmetric.
Algorithms are public → key is secret.

Purpose of Hash: MD5 & SHA

asymmetric cryptography is also known as public cryptography.

K_1 : Public key (not a secret) K_2 : Private key (known to receiver only)

symmetric : only one key (∵ same key for encryp. & decryption)
↓
∴ shared secret key b/w sender & receiver.

Public & private key → inverses of each other
can't we get private key from public key?

$y = f(x)$ $y = K_2$ $x_1 = K_1$
↓
known to all

Why can't attacker calculate x?

$x = f^{-1}(y)$.

① inverse does not exist → [either many-to-1 or onto func.]
② calculation of f^{-1} is hard.

↓↓
can't be possible
∴ Inverse exists

∴ ① can't be possible.

Check for hard & easy:

(2)

Problem solved by algo/ solution / program.

Problem & algorithm are not same.
Easy/Hard → Problem
Time comp → Algorithm

Time Complexity : n : input size (in bits)
bits required to hold the i/p data.

Cryptography : i/p is of variable size
T.C. of encryp. algo : $O(k)$ . . . etc.
k : no. of bits to hold data

Hard problem : can't be solved in polynomial time.

- no algo. exists till now
- no algo. will be discovered in future as well.

$O(1)$, $O(\log k)$, $O(k)$, $O(k^2)$, $O(k^3)$ . . . . ← Polynomial

$O(k!)$, ~~$O(k^k)$~~ ~~$O(2^k)$~~ $O(c^k)$ $O(c^{\log n})$ ← Non-polynomial
(Stirling's approxim.)

- $O(c^k)$: Exponential time algorithm.
- $O(c^{\log n})$: Sub exponential time algorithm

Easy problems :

Hard problem : calculating all subsets, Tower of Hanoi (AI problems) Graph/Map coloring, N queens, 8-puzzle, Travelling Salesman Prob.,

Time Complexity [not problem complexity].
Growth rate of the input size - time curve is very fast (execution time grows at a very fast rate)

Machine will hang in exponential time.

NP problem: P / NP / etc → infinite set
But shown as finite in the venn dia

P: polynomial time solvable.
↳ subset of NP.

Unsolvable problems: more than solvable
(defined by Turing)

If both P_1 & P_2 are hard, then how can you say P_1 is harder than P_2?
⇒ P_2 reduces to P_1.

Problem Reduction: $P_1 \leq P_2$: P_1 reduces to P_2
Using the soln. of P_2, P_1 can be solved.

A2 : solver of P_2
Using A_2, you can design an algo. A1 that can solve P_1.
⇒ we have a black-box access (working not known)
(give i/p → get o/p)
or ORACLE access.

∴ P_1 can be solved.

P_1: find shortest route in an unweighted graph
P_2: " " " " weighted graph.

⇒ P_1 reduces to P_2. [∵ soln. of weighted can be used to solve unweighted]

P_1 is reducible to P_2 in polynomial time.

T.C. of A_2 not known.
Conversion from P_1 to P_2 is polynomial time.

If P_2 is easy then P_1 ?
The no. of times you are calling the func must also be polynomial

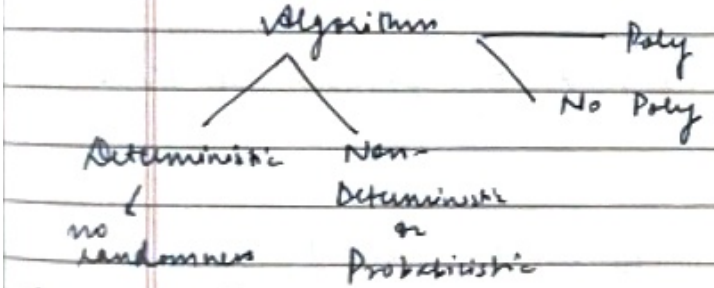
If P_2 is easy then P_1 is easy.

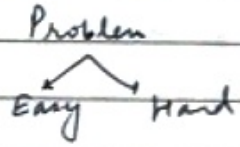
If P_2 is hard? Then P_1 – ?

(?) If P_1 is hard then P_2 is hard [contrapositive]

P_2 is hard as long as $P_1 \overset{\text{in}}{\uparrow}$ hard.

09.08.24





Decision problem → o/1 problem

Eg. prime no. testing
i/p → integer
o/p → yes / no.

Counting problem is decision problem.

Non-deterministic decision algorithm :

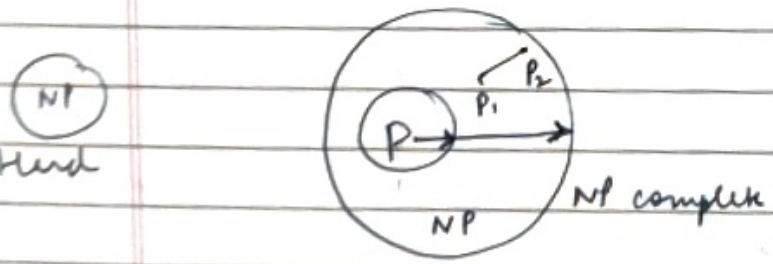
- Las Vegas algorithm
  - either correct ans or failure
  - (yes/no)
- Monte Carlo algorithm
  - Yes / No ans
    - Yes biased
      - If yes → correct
      - no → probability of correctness
    - No biased
      - If yes → probability of correctness
      - No → correct

$y = f(x)$ — x: Private key, y: Public key

y & f : known
Can we find x?

$x = f^{-1}(y)$ → difficult to compute

Any problem which is not solved by polynomial time algorithm is hard.



P_2 harder than P_1
$\Rightarrow P_1$ reduces to P_2

A hard problem which is NP is not hard.

NP → There must exist a polynomial time algo. to check if candidate solution is correct / not

Why all problems are not NP?

NP is hard

But all hard are not NP.

In cryptography, we use NP type problems
(Or maybe NP complete)
(but not NP hard)

Non-deterministic polynomial algo required

Non-negligible probability of success:

↓

not a threshold

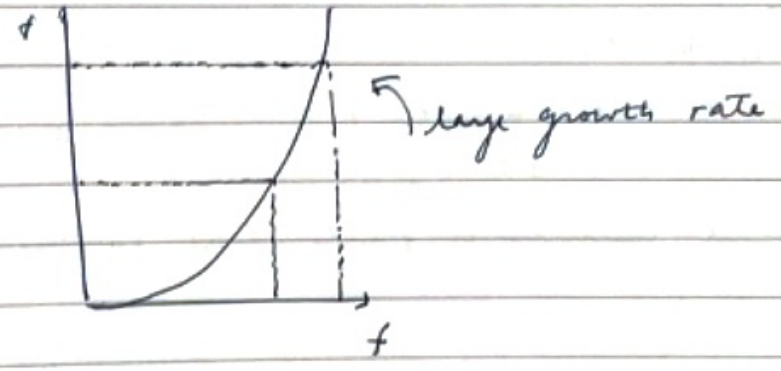
It is the function of input size.

↓

a function which is not negligible

What is negligible function?

If growth rate is very high → not a desirable algorithm.



Security Property of Cryptographic Algorithm

Correctness or completeness

Soundness

Probability of correctness depends on ① Hardness
② Input

N-Queen : hard
But not for = N
Hard for N = 12 / 14 etc.
∴ Hardness depends on input.

---

① One-way function : a function f which is easy but f^{-1} is hard is called one way.

② Trapdoor function : f easy, f^{-1} hard but if some key is known f^{-1} is easy

Integer Factorization Problem:

n: product of 2 prime factors.
$n = p \times q$

For computation problem

Given: n Find / compute: p
s.t. p/n

} Integer factorization Problem.

Constant number of trials done on solution space.

For decision problem:

~~etc~~

Given: n

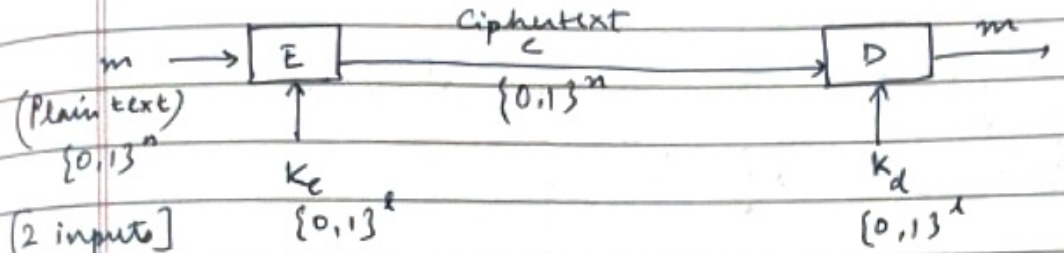
Decide if n is prime

16.08.24

Encryption Decryption

C'
=

Sender (A) — Receiver (B) — either sym or asym



[2 inputs]

K: shared secret key

$c = E(m, k)$

K_e = Inverse (K_d)

Public key B — Private key B

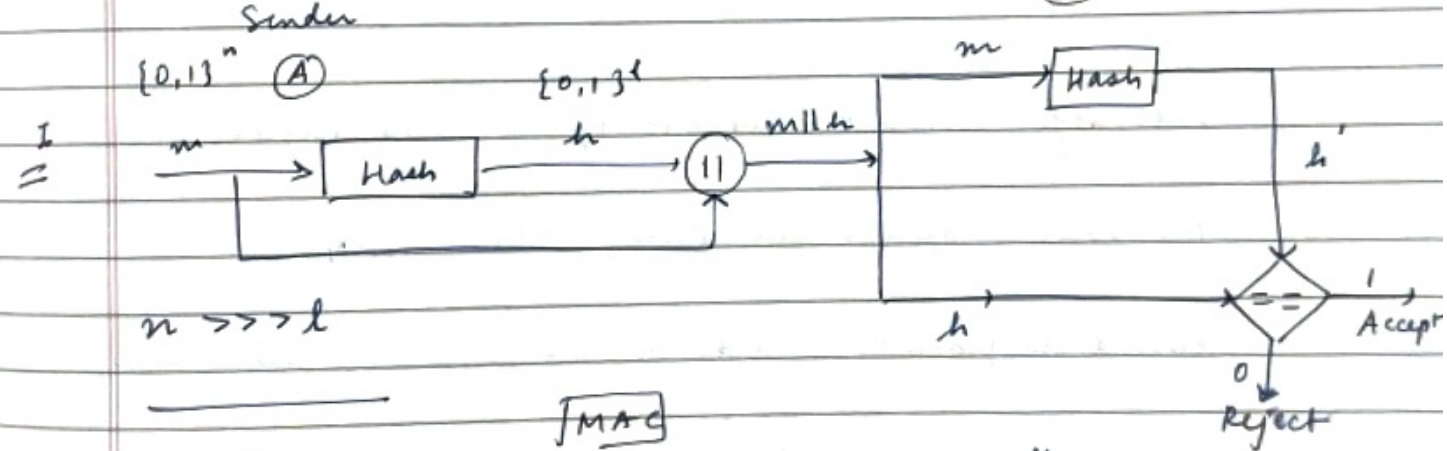
For symmetric encryption, $K_e = K_d$.

$n \approx l$ [approx same]

Hash

Neither sym nor asym

Sender (A) — Receiver (B)



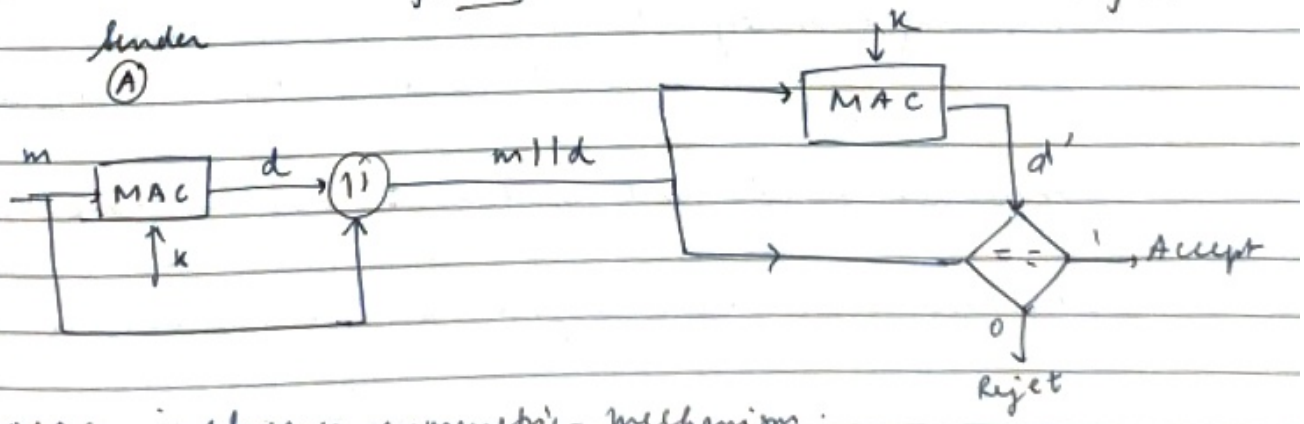
I =

$n >>> l$

MAC

Sender (A)

A-I =



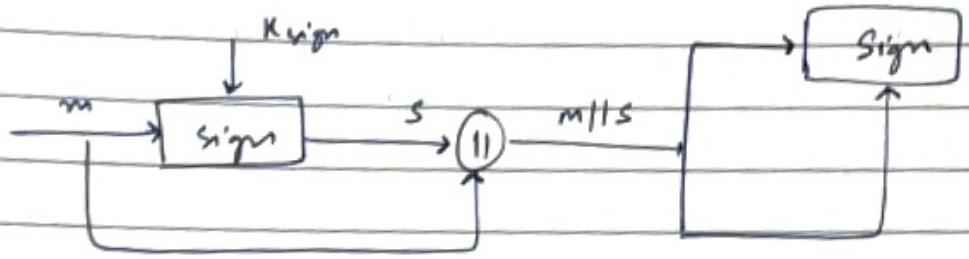
MAC is always symmetric mechanism.

A-1-N

Digital signature : always asymmetric.

Sender

(A)



Integrity without authenticity is meaningless.

OpenSSL : open source library like cryptopp.

Soundness property : attacker can't achieve his goal.

Key confidentiality → key not used in hash.
↳ used in MAC & digital signature

Soundness requirement of hash : does not have key.

Objective of attacker / algorithm? Problems in case of hashing?

↙
Computation or decision problem?

Encryption → Correctness → it should work
→ Soundness → hard problem
↳ decryption

Sender (A) Receiver (B)

encryption key, decryption key

$k_e = (k_d)^{-1}$ $k_e = k_d$

public key sender, private key receiver

Sender (A)

→ confidentiality (k_d cannot be computed)

Encryption

m (plain text) $\{0,1\}^n$ → E → (cipher text) $\{0,1\}^n$ → D → m

k_e $\{0,1\}^d$ k_d $\{0,1\}^d$

$c = E(m,k)$ (cipher encrypt text)

$m = D(c,k)$ (message cannot be computed)

→ integrity

Receiver (B)

Hash

Hash Sender (A)

m $\{0,1\}^n$ → Hash → h $\{0,1\}^l$ → || → m||h

message

$n >> l$

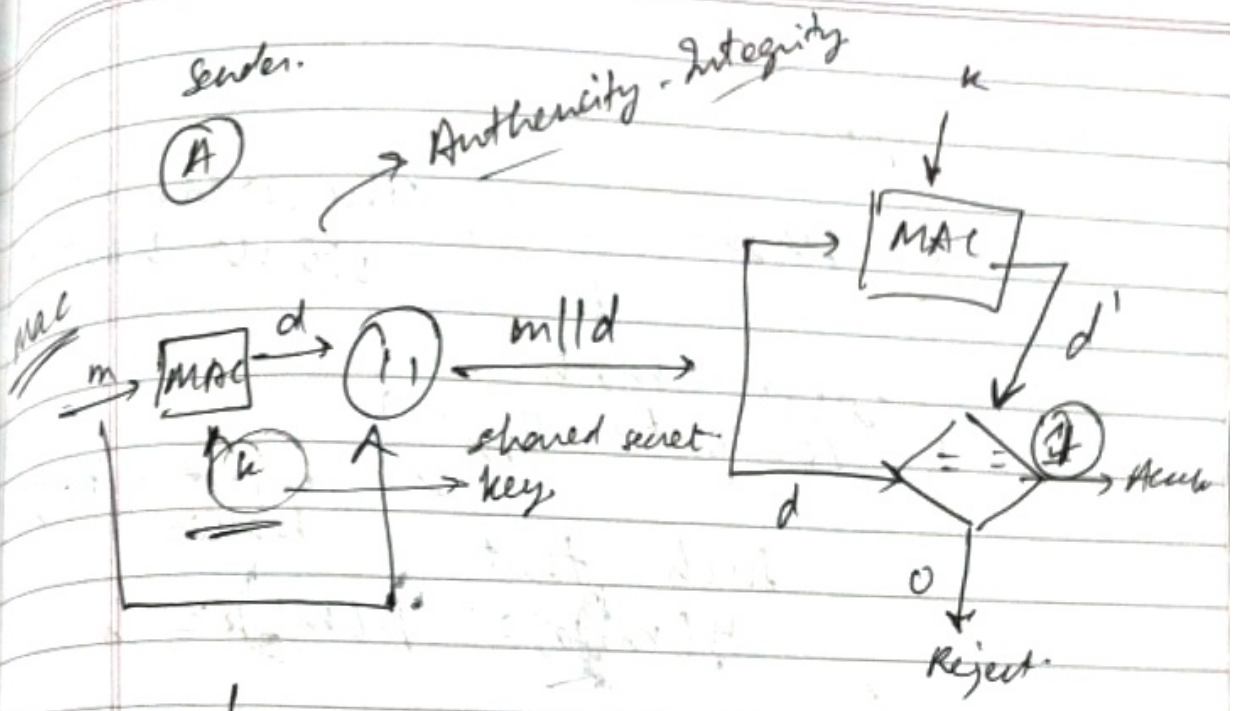
m → Hash → h'

h, h' → == → 1 accept ✓ / 0 Reject ✗

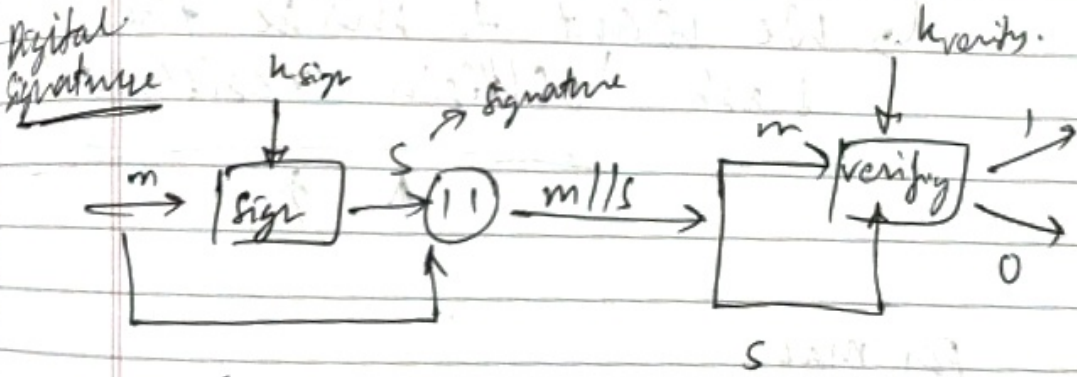
→ keyless encryption

→ similar for MAC

Integrity → modification should be detected



↳ Symmetric Encryption Mechanism



$k_{sign} = (k_{verify})^{-1}$ ↳ inverse.

- Authenticity
- Integrity
- Non repudiability.

private key sender

public key

↑ Always Asymmetric

yes

For Hash

The attacker would want to

given : m, h

compute H(m')

← injected/modified malware

such that

H(m) = H(m') = h

→ We want to make the hash 'collision resistant'

For Mac

Shared secret key is used with hashing.

define input - output relationship.

Hash algo. known
Given: m, h
Compute what? ← Computational Problem
Hard problems?
If yes → sound ✓

m transported over a secure channel. So find m'
s.t. $H(m) = H(m')$
ie digest of new msg. is same as that of m.

Find a malware s.t. hash of malware is same as that of m because hash is a many-to-one function.

$n >>> l$ 2^n can have 2^l hash only.
So repetition possible. → collision

How is the hash secure then? Problem should be hard ie finding m' is difficult due to computational problem.
→ Collision
∴ Hash func. should be designed such that there is collision resistance.
ie finding collision is computationally hard/infeasible.
[but not impossible!]

Diffusion Property: changing a single bit of data will change the hash output.

Normal distribution
2^l → diff. strings possible

Forgot password

Given i/p, o/p should be equally likely from codomain

Hashing : deterministic algorithm.
↳ looks like a random state of bytes but is still deterministic
↓
Not a randomisation algorithm
it is a random string of byte.

3.08.24

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

order $(G) = |S|$

If order $= \infty$ then it is infinite group.

$\langle Z, + \rangle$, $\langle R, + \rangle$, $\langle R, +, \times \rangle$, $\langle Z, \times \rangle$, $\langle R, \times \rangle$, $\langle Z, +, \times \rangle$ } Finite group / fields

$\langle Z_n, \times \rangle$ $\quad G = \langle Z_n, + \rangle$ → valid additive group
↳ mod n

$Z_n = \{0, \ldots, (n-1)\}$

$n = 10$: $3 + 5 = 8$

$3 + 7 = 0$ $\quad (\because 10 \% 10 = 0)$

$3 + 1 = 1$

3 is additive inverse of 7 & vice-versa.

$a \in Z_n$

If b is multiplicative inverse of a

$(a * b) \bmod n = 1$

what is 3^{-1} wrt $n = 10$

$(3 * 7) \bmod 10 = 21 \bmod 10$

$= 1$

$\therefore 3^{-1} = 7$

Is $\langle Z_n, \times \rangle$ a valid group?

Take $a = 2$

a^{-1} : $(2 \times b) \bmod 10 = 1$

$b \notin Z_n$.

∴ Multiplicative inverse of $a = 2$ does not exist (in $a = 4$)

∴ $\langle Z_n, \times \rangle$ not a valid group.

Numbers which are co-prime with n will give inverse. Eg. 1 is co-prime with n always.

∴ $\gcd(1, n) = 1$

$\{1, 3, 7, 9\}$ → co-prime with $n = 10$.

∴ Inverse exists.

Z_n^* : numbers which $\in Z_n$ but are also co-prime with n.

$\langle Z_n^*, \times \rangle$ is a group or not?

Inverse exist ✓

Closure? satisfied.

∴ Z_n^* is a group.

↳ valid multiplicative group.

$Z_p^* = \{1, \ldots, (n-1)\}$ } Public key cryptography

$F = \langle Z_p, +, \times \rangle$

$GF = \langle \{0,1\}^n, \oplus, \otimes \rangle$ → Symmetric Cryptography [much faster than asymmetric crypto]

Operations on large integers → slow process
When done on binary → fast
∴ symmetric ✓ is faster.

Both are equally secure. Security depends on size of key.
Then why use public key cryptography? [if symmetric is much faster]

(PKC)
Public key crypto. takes a larger key size to achieve same level of security as symmetric. This reduces speed of public key crypto.

Distribution of symmetric keys requires asymmetric & symmetric cryptography.

128 bits encrypted using costly public crypto.

To establish a secure communication ~~then~~ through open channel initially → PKC required.
∴ how to send symmetric key through insecure channel?
In PKC, public - private key pair generated.
↓
known to all

PKC essential for initial security establishment.

Hybrid crypto → mixture of PKC & SC.

$F_p = \langle Z_p, +, \times \rangle$

$(a+b) \bmod p$ $(a/b) \bmod p$

$(a * b^{-1}) \bmod p$

↳ inbuilt func. to find inverse

RSA & discrete log problem

- random generator
- modular exponentiation
- multiplicative modular inverse $[b^{-1} \bmod p]$

2nd most hard problem in cryptography. Discrete log Problem

$G = < Z_p^*, \times >$

let $a \in Z_p^*$

$a^i \bmod p = 1$

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

p: any prime no.

modular multiplication:

$a^0 = 1$ $a^1 = 5$

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

$= 3 \times 5 = 15$ $a^4 = 1$

$a^4 = 9$ $a^5 = 1$ $a^6 = 5$ $a^7 = 3$

i = order ord(a) = 5

i.e. $a^i = 1$ for $i = 5$

∴ order(a) = 5

Next is ord(2)? → 10

$p = 7$

$2^i \bmod 7 = 1$

generators / Primitive Roots

↓

generate the entire group.

(a)

Subgroup: subset of elements of parent group such that it satisfies the properties of a group.

Subfield : subset of field that is a field.

Group Extension

Field extension : $\mathbb{R}$ is a field extension over the field of $\mathbb{Q}$ (rational nos)
$\mathbb{C}$ is a field extension over $\mathbb{R}$

[can be understood as Super field]

Cyclic subgroup

Let g : generator for $G = \langle Z_p^*, \times \rangle$

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

β range = Z_p^*
Domain of x

If $x = 7$ gives 1 same as $x = 0$.

$\therefore 7 \equiv 0$

So $x > p$ no use.

mapping from x to β

$g_p^x : Z_p^* \to Z_p^*$ mapping is bijective
$\Rightarrow$ Inverse also exists

Given β, can we find x?

$g^x = \beta$

$x = \log_g \beta$

g, p constants

$g^x \bmod p = \beta$

$\boxed{x = d\log_{g,p} \beta}$ d: discrete

dlog : discrete log

Given β. can you find α?

30.08.24

p g : generator $\quad \langle Z_p^*, x \rangle \longleftarrow \langle Z_p^*, x \bmod p \rangle$

$\alpha \in Z_p^* \qquad \beta = g^\alpha \bmod p$

f ↓ easy ← { Given : g, p, α ; Find : β

But the inverse : given : β, g, p
find : α

$$\boxed{\alpha = \underset{g,p}{d\log}\ \beta}$$

$\alpha = \log_g \beta$

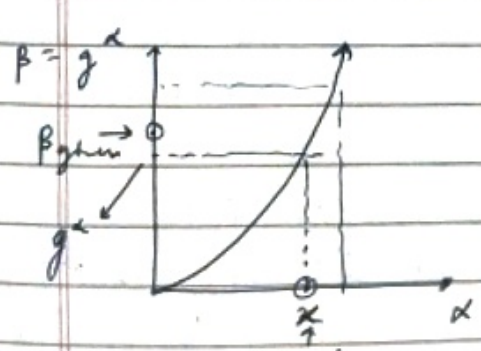
if $g, \beta \to$ integers
$\alpha \to$ may not be integer

↓

α always an integer.

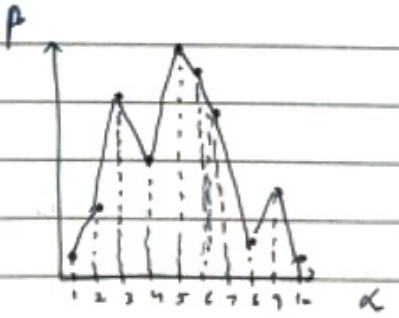
$\alpha = \log_g \beta$: is it polynomial type / easy problem?

If not hard in arithmetic domain, why is it hard in modular arithmetic?



(random (α taken))

expected value is above x
not below x.
∴ search space gets reduced to (a, ∞)



no pattern (neither monotonically increasing nor decreasing)

∴ In case of monotonic graph, binary search can be done ie search space can be reduced.

In 2nd graph : search space = Z_p^*

after 1 check → $Z_p^* - 1$

2nd check → $Z_p^* - 2$

∴ search space is not reduced significantly

∴ Hard problem.

Can't be solved in linear time.

But why can't we apply algo. of arithmetic to modular arithmetic?

No algo. has been found to solve discrete log problem in polynomial time. ∴ Hard.

It is assumed to be hard → Hardness assumption

Integer factorisation problem

Discrete log problem

State the _____ hardness assumption. → first state problem.

State the _____ problem.

It is believed there exists no poly. time algo. to solve this problem.

RSA encryption sys. is secure assuming integer fact. problem is hard. (Starting point is not a fact).

∴ Assumptions have been made to make RSA secure.

Random variable $a \in Z_p^*$ — Z_p^*, multiplicative modulo p.

↓

generator (but not of Z_p^*)

↓

of a ~~set~~ subset of Z_p^*.

When G_2 is subgroup?

$G_1 = \langle S_1, \cdot \rangle$ $G_2 = \langle S_2, \cdot \rangle$ ① $S_2 \subseteq S_1$

② binary operator is same as parent group

③ G_2 is a group.

$G_1 \leftarrow \langle S_1, \times \bmod p \rangle$ $S \subseteq Z_p^*$

$\langle S, \cdot \rangle$ $\cdot \equiv \times \bmod p$.

How to get elements of S? From a.

$S = \{ a^1 \bmod p$

$a^2 \bmod p$

$a^3 \bmod p$

:

upto $|G| = q$ ← order of G.

$a^q \bmod p. \}$

exponents : 1 to q

$b = g^\alpha \bmod p$ α ranges from 1 to q.

$b \in Z_p^*$

Difficulty depends on size of q {not p}

as solution space reduced till q.

Size of sub-group must be a divisor of the size of parent group → Lagrange's Theorem for Subgroup size.

For $p = 11$ $\quad Z_p^*$ size $= 10$

So size of sub-group : divisor of 10
$= 1, 2, 5, 10.$

$|G| = q$: divisor of $p-1$

(p: prime no.)

Size of $Z_p^* = p-1$

q must be a prime number.
Divisor of $(p-1)$ may not be a prime no.

Expectation : q should be prime to get multiplicative inverse of every element ($\because$ co-prime necessary)

Z_n : all elements of Z_n are not co-prime with n
$\therefore$ Multiplicative inverse of every element does not exist.

$\because$ all ~~elements~~ exponents are coming from Z_q, hence to have multiplicative inverse mod q, q must be prime.

Algorithm: Prime & Generator class in cryptopp

Take random no. from Z_p^*
Get order of that no.
Is the order a large prime no.?
If no take another random no.

Prime (p)
Subprime (q)
Generator (g)

Why not generate g from Z_p^* directly?
Why take sub-group?

$\because$ Finding the generator of a group is again a hard problem!

Generating sub-group from a randomly chosen generator is polynomial time.
We are not generating the generator of sub-group.

Using Hardness Assumption of cryptography:
Diffie-Hellman Protocol: 2 party key protocol

→ establishes a shared secret key b/w 2 parties
[for symmetric encryption (∵ it is faster)]

* assumption: channel through which key is shared is trusted for 2 aspects of security:
[Authenticity & Integrity protection.
But it is not trusted for Confidentiality.

trusted in whatsapp, gmail

Had it been confidential, then public-key encryption would not be required, the symmetric key could be ~~was~~ shared directly.

Why authenticity & integrity imp?
→ To get public key: o/w public key will be replaced
→ Man in the middle (Eg. router is attacked)

→ Session hijacking: connection established with an attacker instead of correct server.

Public key certificate attached: ∴ authenticity & integrity assumptions were removed.

Using PKI protocol: public key of CA distributed among all users/parties.

Chatgpt:

Trusted certificate authorities
Import / export certificate
Certificate revokation
Show certificate of website
List of root certificate authorities
How to manually include a certificate in browser?
(green lock, https)

} PKI protocol public key infrastructure (very weak protocol)

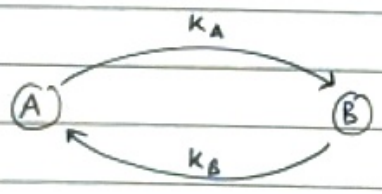
If underlying layer of HTTP is TCP → HTTP
SSL → HTTPS
FTP is SSL → FTPS

SSL: secure socket layer
↳ makes TCP secure.

Confidentiality: symmetric
A, I → " MAC protocols.
↳ symmetric crypto-techniques.

Diffie Hillman: supported by SSL.

31.08.23



1. $a \xleftarrow{R} Z_q^*$ 1. $b \xleftarrow{R} Z_q^*$] Private keys
2. $K_A \leftarrow g^a \bmod p$ 2. $K_B \leftarrow g^b \bmod p$] Public keys

Diffie Hillman public keys are exchanged.

3. $(K_B)^a \bmod p$ 3. $(K_A)^b \bmod p$

Correctness analysis

$(K_B)^a \bmod p = (g^b)^a \bmod p = g^{ab} \bmod p$

$=$

$(K_A)^b \bmod p = (g^a)^b \bmod p = g^{ab} \bmod p$

K_{AB} ↓ can act as a symmetric key for further encryption.

Security Analysis / Soundness

① identify objective of attacker in the protocol.

$g^{aZ} \bmod p$ $g^{bZ} \bmod p$ ← Man-in-the-middle attack.

(?) A Attacker B

Public key certificate : identity public key
(issued for both K_A & K_B)
Without the certificate → it becomes anonymous
Diffi - Hillman (DH).

Fixed DH
Combination of anonymous & fixed DH ← EPHEMERAL DH.
Public RSA key → Certificate issued → digitally signs DH
for both sides public key using RSA private key

(& public key RSA & DH)

no certificate against DH pub. key. ↓ Verify digital signature.

Anonymous is not protected unless channel is secure.
Rest 2 → secure.

Ephemeral : higher cost :: 2-step key generation
2-step verification.

session key established in Fixed DH remains same (not flexible)

no need to exchange keys. Just exchange certificates as they already include keys.

Elliptical cryptography : most advanced form till date.

Interval for which certificate is issued?

Responsibility of browser to avoid insecure website
Authentic browser
Ubuntu : much safer than Windows.
Using authentic browser to download another browser.

① https → SSL underlying layer
② Broken law certificate →

CA revoke the certificates of blacklisted websites.

Why can't public key certificates be changed frequently?

Fixed DH provides one session key for 3-4 months.

Objective of attacker : $g^a \bmod p$, $g^b \bmod p$.
Attacker want to find $g^{ab} \bmod p$.

DH Problem
- Given : g, p, q, $g^a \bmod p$, $g^b \bmod p$
- Find : $g^{ab} \bmod p$

Diffie Hellman problem is different than DH protocol

If DH is not hard, it is not sound.
When will the problem not be hard?

Can't calculate a & b even if g & p given.
↓
Discrete Log Problem
∴ Hard problem.

D-Hellman's security depends on Discrete log hardness assumption.

a from g^a b from g^b
↓
not possible to find.

∴ Attacker can't intrude ← Wrong argument!

inference rule : if - then
$A \rightarrow B \equiv \neg B \rightarrow \neg A$
$\not\equiv \neg A \rightarrow \neg B$

A: discrete log is hard.
B: DH is hard.

B → A

If DH is hard, then certainly discrete log is hard.

B reduces to A

DH reduces to discrete log because solution of discrete log can be used to solve DH.

Security of DH is ensured due to hardness assumption
If DL → True.
DH is secure as long as DL is hard → False
DL is hard as long as DH is secure → True

DH is hard because till date no polynomial time algo. has been found to solve it.

DH hardness assumption : is it same as DL hardness assumption ?

NO!!

Both are separate
Relationship : DH can never be hard if DL is not hard

3 hard problems : Integer factorization
Discrete log
Diffie Hellman.

RSA 3 algorithms:

① KeyGen algorithm: 1. $p \leftarrow$ large prime
2. $q \leftarrow$ large prime
3. $n \leftarrow p \times q$
4. $\phi \leftarrow (p-1) \times (q-1)$
5. $d \xleftarrow{R} \mathbb{Z}^*_\phi$

i.e. d is a random element from $\mathbb{Z}^*_\phi$

1 to $(\phi-1)$ and co-prime with ϕ.

Will terminate in polynomial time.

6. $e \leftarrow d^{-1} \bmod \phi$

[only this step is mod ϕ, other steps are mod n]

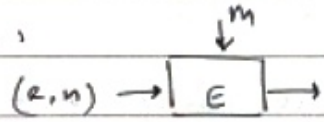
ensures security of RSA instead of n in RSA.

using ϕ → no security in RSA.

(e, n) → Public key

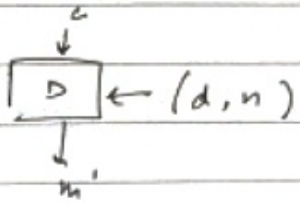
(d, n) → Private key.

② Encryption:



1. $c = m^e \bmod n$

③ Decryption



$m' = c^d \bmod n$

Correctness proof: $m' = c^d \bmod n$

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

$= m^{ed} \bmod n$

$= m^{ed} \bmod n$ $\quad [\because e = d^{-1} \Rightarrow ed = 1]$

$= m$

But $e = d^{-1}$ mod ϕ not n. Here we are doing calc. mod n. So how $ed = 1$?

ϕ : Euler's func.
Due to Euler's Theorem, we can use $ed = 1$

Soundness of RSA:

① Identify attackers problem.
attackers against RSA encryption scheme

→ negation of objective of scheme ⇒ to find m.
Why encryption? to protect m.

Given: ~~(n, e)~~, c $\quad c, e, n$ } RSA problem
Find: m

Is it hard / not?
When is the problem not hard?
When you know factors of ~~m~~ n.
(Integer factorisation)
↓
assumed hard

✓ RSA is not secure until integer fact. is hard

✗ RSA is secure because int. fact is hard.

✓ RSA is not hard if int. fact. is not hard

RSA hardness assumption

∴ 4 hard problems & 4 assumption

① Relationship b/w DH & DL ② RSA & IFact.
one-way dependency

Hash Function:

$H = \{0,1\}^* \rightarrow \{0,1\}^l$

Soundness requirement of hash function?
must be a deterministic algorithm

Purpose:

Given: $m \in \{0,1\}^*$, H
Find: $m' \in \{0,1\}^*$
s.t. $H(m) = H(m')$
} Hash problem

What property will make Hash func. hard?

H is many-to-one function. ∴ Bijection is not true.

Is it NP problem / not?
Can you write non-deterministic polynomial time problem for this?

candidate solution = any binary string.
↓
check in polynomial time ✓ find its hash & match.

Yes. Check search space constant no. of times (c)
if hash obtained matches / not. Monte Carlo / Las Vegas?
$1-(1-(1-\frac{1}{2^{kl}}))$

When the prob. becomes negligible?

Success probability of each iteration = $\frac{1}{2^x}$ (size of output)

Depends on size of output (& not input)

each digest value is equally likely ← assumption.
ie each string comes with same probability
Then only this probs will become hard.

Oracle / black box: i/p → o/p. (no procedure)

Random Oracle Model: not non-deterministic
↳ is a deterministic algo.

$b = g^x \bmod p$.
↳ mathematical relation exists even though
given is so random.

For the same x → same y ≠ deterministic.