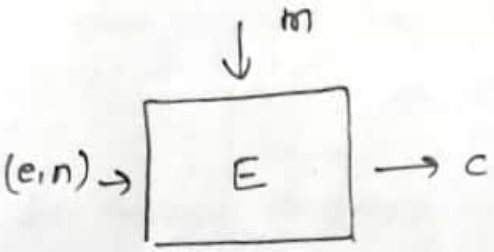
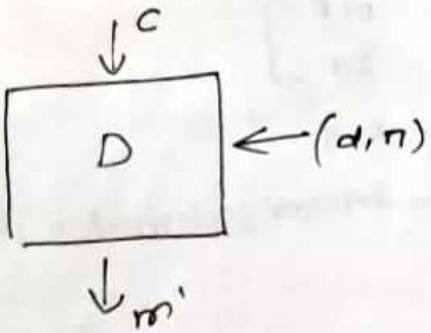
Encryption



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

Decryption



$(m' = c^d \bmod n)$

correctness proof

$$m' = c^d \bmod n$$
$$= (m^e)^d \bmod n$$
$$= (m^{ed}) \bmod n$$
$$= m^1 \bmod n$$
$$= m$$

Benny Madhan

$[e = d^{-1}]$

$ed = 1$

is wrt ϕ

then why we are using wrt n why

Using Euler's theorem we can say

because $\phi = (p-1)(q-1)$

Soundness of RSA

Identify the attacker problem

P_2

→ to get m

algo is also known by default to attacker

given : e, n, c

find : m

RSA problem

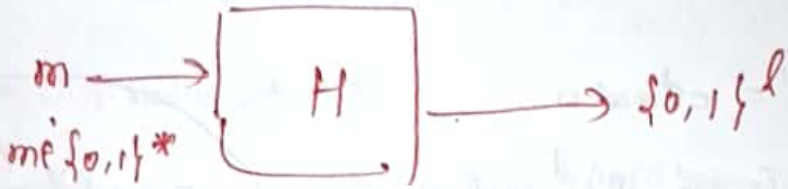
RSA is not secure if Integer factorisation is not hard

$p \rightarrow q \equiv \neg q \rightarrow \neg p$

RSA is hard because Integer factorisation is hard ✗

{ [DLP, OH], [RSA, Integer factorization problem] }

Hash function



it should be deterministic

$$H \equiv \{0,1\}^* \longrightarrow \{0,1\}^l$$

Random Oracle model → deterministic algo

Hash function is modeled as Random Oracle

Diffusion property

for small change in input ⇒ widespread change in the output

(1) Collision Resistent hash function

given $h \in \{0,1\}^l$

find : $m \in \{0,1\}^*$ such that $H(m) = h$

Preimage Resistance → (1)

Second Preimage Resistance → (2)

2 ★

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

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

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

① hash given, find a message for the given hash

② given m and hashfunction, find other m' for which

$$H(m') = H(m)$$

easier to prove

Collision Resistance

③ given : H
find (m, m')
s.t $H(m) = H(m')$

If a hash function is collision Resistance then it follow ① and ②

impossible to prove ②

For symmetrical Hashing → statistical test perform

Desired property Collision Resistance

⚠ If Hash function follows ROM then it must be collision Resistance

How to check ROM following or not

ROM

① function must be deterministic

② independence

③

$$\Delta(K) = |P(S) - 0.5|$$

other parameter only for decision making for binary

- Randomness
- Consistency
- Independence

for single bit with no significant advantage

toss coin without seeing inputs and use memory DP to save answers for each inputs

nist → any correlation b/w I/O or not

$h \leftarrow H(m)$

while ( ) {

$m' \xleftarrow{R} \{0,1\}^n$

if $((H(m')) == h)$

return m'

}

For (i = 1 to c)

↓

Integer factorisation

message size anything but security depends on l

for any message without seeing → $\frac{1}{2^l}$ ** ROM

$\frac{1}{2^l}$ ← probability of success of each iteration

We use statistical testing to prove ROM

If it is ROM, It is a Collision Resistance

Hashing # take if Integrity

Soundness Analysis of RSA Digital Signature

identify attackers objective (polynomial number of pair)

(Public keys) atleast one correct message signature pair

P_1 { given : $e, n, (m_1, s_1), (m_2, s_2) \cdots (m_c, s_c)$; $c = poly(k)$

find : (m', s') st $verify(m', s', e, n) = 1$

Concept of problem Reduction to prove Soundness of Crypto algo

As long as RSA encryption scheme is secure

using soln of P_1 can solve P_2

If $P_1 \in$ Poly then $P_2 \in$ Poly

take contrapositive

P_2 reduces to P_1

08/09/2024

what is the probability that the attacker has selected the same row

$$\epsilon'(K) = \underbrace{\perp}_{c_1+c_2} \quad \epsilon(K)$$

$c_1 + c_2$ → poly

$\epsilon(K)$ → [attacker advantage]

$c_1 \in Poly(K)$

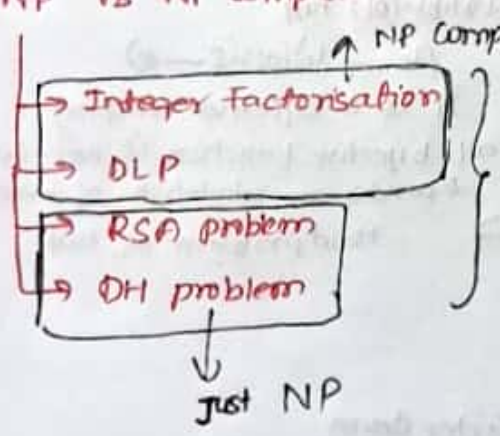
$c_2 \in Poly(K)$

→ $c_1 + c_2 \in Poly(K)$

If $\epsilon(K)$ is not negligible then $\epsilon'(K)$ is also not negligible

which contradicts the global assumption that RSA is hard

NP Vs NP complete



which of the following is NP complete ? and why

Soundness property of all γ

Collision Resistant hash function

Random Oracle

Symmetric Encryption

1) Block cipher
2) Stream cipher

Block cipher → (2) Non Feistel Cipher → # only invertible gates
Eg: AES

depending upon (gates are invertible or not)

Block cipher → (1) Feistel Cipher → DES ← Eg

Feistel Cipher:
- works at binary level only
- weak one
- # if it contains only mixture of invertible & non invertibles

Diffusion, Confusion } property

Diffusion, Confusion:
↓
bits of Key and bits of Cipher text
there should be no statistical relation

Block Cipher:
↓ takes only 128 bits something
need to divide into blocks
encrypt each blocks using AES

XOR → I
left SHIFT
ROTATE → I
COMPRESS
DECOMPRESS
↓
4 bits to 8 bits
8 bits to 4 bits

Gates → may not

Data: K bits → may not be invertible

E: + K bits Key → Cipher K bits

Substitute ⇒ a type of permutation → may (or) may not be invertible

↓
invertible (or) not
(4 — 4) (or) (8 — 8)
if is a bijective mapping
all bijective function is not invertible
depends on calculation of inverse if is Hard problem or not

Stream cipher (Not needed much)

RC4 ← Eg

Key → Random Key Stream Generator →

Can take entire file

Key stream ⊕ Data stream → Cipher stream

XOR

array
Deterministic algo
Cryptographic Random Number Generator

RC4 RC4

How the plain text is obtained from cipher text for Feistel Ciphers