Rabin Cryptosystem → not deterministic.

RSA → e and d are fixed.

$e = 2$, $d = 1/2$

RSA

$C = P^e \bmod n = P^2 \bmod n$

$P = C^d \bmod n = C^{1/2} \bmod n$

Decryption creates 4 equally probable PTs.

RABIN

p and q ⇒ large primes.

$4k+3$ & $p \neq q$.

↑ for different values of k

$p \& q$ ⇒ complement to $4k+3$.

$3 \equiv 3 \bmod 4$

$4\times1+3 = 3 \bmod 4$

$4\times2+3 = 3 \bmod 4$

$4\times3+3 \equiv 3 \bmod 4$.

if $x = 3$, & $y = 4$

$ky + x \equiv x$

$\Rightarrow (ky + x) \bmod y \equiv x \pmod y$

Q. Given $p = 7$, $q = 11$ $C = 28$

What is the PT P?

$n = p \times q = 77$

$a_1 = +\left(c^{(p+1)/4}\right) \bmod p$

$a_2 = -\left(c^{(p+1)/4}\right) \bmod p$

$b_1 = +\left(c^{(q+1)/4}\right) \bmod q$

$b_2 = -\left(c^{(q+1)/4}\right) \bmod q$

CRT is invoked 4 times

$p_1 \leftarrow CRT(a_1, b_1, p, q)$

$p_2 \leftarrow CRT(a_1, b_2, p, q)$

$p_3 \leftarrow CRT(a_2, b_1, p, q)$

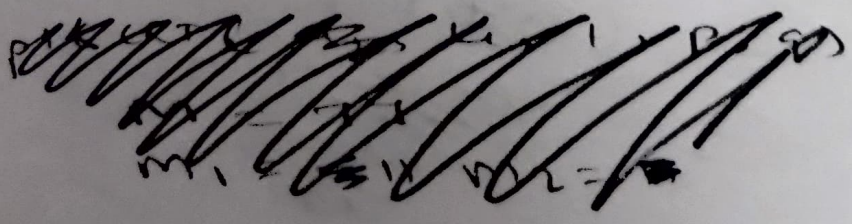
$p_4 \leftarrow CRT(a_2, b_2, p, q)$

$a_1 = +23^2 \bmod 7 = 4 \bmod 7$

$a_2 = -4 \bmod 7 = 3 - 4 \bmod 7$

$b_1 = 23^3 \bmod 11 = 1 \bmod 11$

$b_2 = -23^3 \bmod 11 = 10 - 1 \bmod 11$



P1

$a_1 = 4 \bmod 7$

$b_1 = 1 \bmod 11$

$m = 77$

$m_1 = \frac{77}{7} = 11$ $\quad m_2 = 7$

$m_1^{-1} \bmod 7 = 11^{-1} \bmod 7$

$= 11^{\phi(7)-1} \bmod 7$

$= 11^5 \bmod 7$

$= 2 \bmod 7$

$m_2^{-1} \bmod 11 = 7^{-1} \bmod 11$

$= 7^{11-2} \bmod 11$

$= 7^9 \bmod 11$

$= 8 \bmod 11$

$m_1^{-1} = 2, \quad m_2^{-1} = 9$

$p_1 = (4 \times 11 \times 2 + 1 \times 7 \times 8) \ldots$

$= (88 + 56) \bmod 77$

$= (144) \bmod 77$

$= 67$

P2

$a_1 = 4 \bmod 7$

$b_2 = -1 \bmod 11 = 10 \bmod 11$

$m = 77$

$m_1 = 11 \quad m_2 = 7$

$m_1^{-1} = 2 \quad m_2^{-1} = 8$

$x = (4 \times 11 \times 2 + 10 \times 7 \times 8) \bmod 77$

$= (88 + 560) \bmod 77$

$= 648 \bmod 77$

$= 32 \bmod 77$

$= 32$

77
× 8
616

648
- 616
32

P3

$a_2 = -4 \bmod 7 = 3 \bmod 7$

$b_1 = 1 \bmod 11$

$m = 77$

$m_1 = 11 \quad m_2 = 7$

$m_1^{-1} = 2 \quad m_2^{-1} = 8$

$x = (3 \times 11 \times 2 + 1 \times 7 \times 8) \bmod 77$

$= (66 + 56) \bmod 77$

$= 45$

$P4 = -67 \bmod 77 = 10.$

$p_1 \leftarrow CRT(a_1, b_1, p, q)$

$p_1 \leftarrow 67$

$p_2 \leftarrow CRT(a_1, b_2, p, q)$

$p_2 \leftarrow 32$

$p_3 \leftarrow CRT(a_2, b_1, p, q)$

$p_3 \leftarrow -32 \bmod 77 = 45$

$p_4 \leftarrow CRT(a_2 b_2, p, q)$

$p_4 \leftarrow -67 \bmod 77 = 10$

$p \rightarrow$	p_1	p_2	p_3	p_4
	67	32	45	10
$p^2 \rightarrow$	4489	1024	2025	1000
$p^2 \bmod 77 \rightarrow$	23	23	23	23

↑ 4 possible values

45
45
225
1804
2025
168
345
312
33

Q. given $p = 23$, $q = 7$ & $P = 24$

what is actual msg?

$n = p \times q = 23 \times 7 = 161$

$C = P^2 \bmod 161 = 576 \bmod 161$

$= \underline{\underline{93}}$.

$a_1 = +(93^6 \bmod 23)$

$a_2 = -(93^6 \bmod 23)$

$b_1 = +(93^2 \bmod 7)$

$b_2 = -(93^2 \bmod 7)$

$a_1 = 1 \bmod 23 = 1$

$a_2 = -1 \bmod 23 = 22$

$b_1 = 4 \bmod 23 = 4$

$b_2 = -4 \bmod 23 = 19$

P1

$m = 161$

$m_1 = 7 \quad m_2 = 23$

$m_1^{-1} = 7^{-1} \bmod 23 = 7^{21} \bmod 23$

$= 3^{10} \times 7 \bmod 23$

$= 4^3 \times 21 \bmod 23$

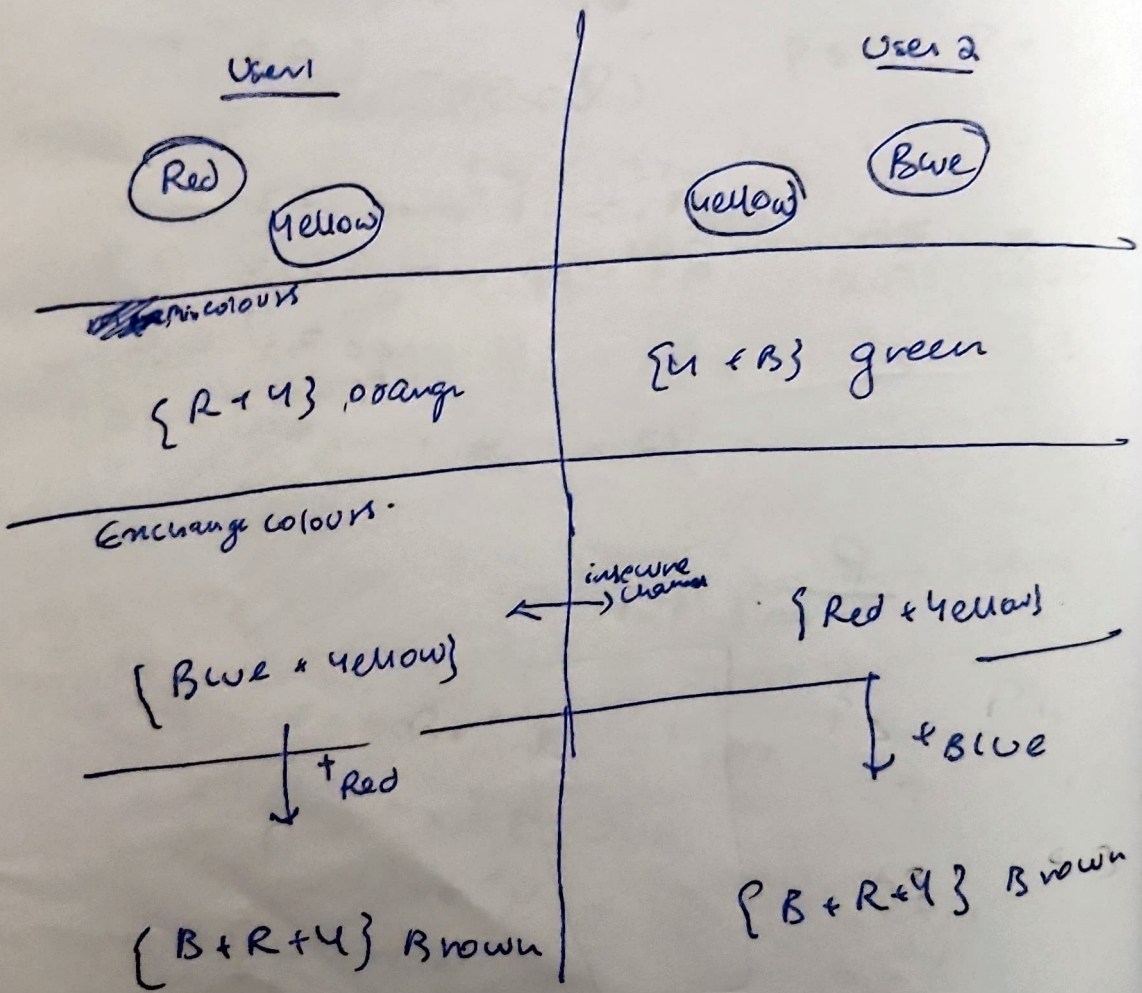
$= 6 \bmod 23$

Diffie Hellman Key Exchange (DHKE)

- Algorithm to do secret exchange b/w two users.
- It is not an encryption algo
- use asymmetric encryption to exchange keys. (for exchanging keys).

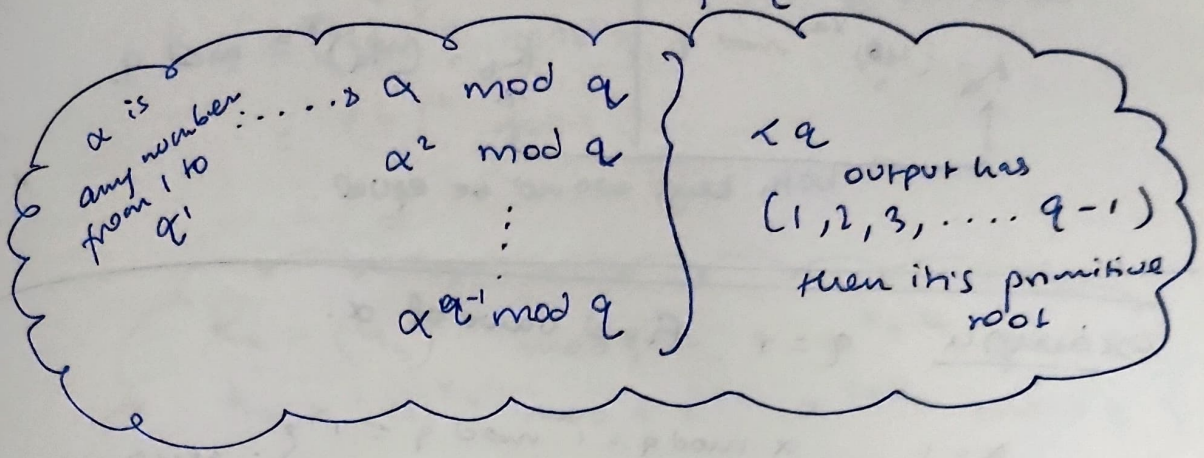
- Used as a precursor to asymmetric encryption methods for encryption.

One way function : easy to compute f(x), difficult to find $f^{-1}(x)$.



Write the steps in key exchange

I. Choose q and α
   - a) q is a prime no.
   - b) select α as a primitive root of q.



II. Deriving the key pair

user 1	user 2
• Assume private key X_a where $X_a < q$	• Assume private key = X_b where $X_b < q$.
• Public key (Y_a) becomes $Y_a = \alpha^{X_a} \bmod q$	• Pubic key (Y_b) becomes $Y_b = \alpha^{X_b} \bmod q$
• key pair (X_a, Y_a)	Key pair (X_b, Y_b)

III Key generation

user 1	user 2
parameters: x_a, y_b, q	parameters: y_b, y_a, q
Secret key generation: $k_a = (y_b)^{x_a} \bmod q$	Secret key generation: $k_b = (y_a)^{x_b} \bmod q$

both keys would be equal.

---

example $q = 7$. find value of α.

$\alpha = 1$ ✗
$\alpha \bmod q = 1 \bmod 7 = 1$
$\alpha^2 \bmod q = 1^2 \bmod 7 = 1$ } : always 1.

$\alpha = 2$ ✗
$2 \bmod 7 = 2$
$2^2 \bmod 7 = 4$
$2^3 \bmod 7 = 1$
$2^4 \bmod 7 = 2$ } repeat.

$\alpha = 3$ ✓
$3 \bmod 7 = 3$
$3^2 \bmod 7 = 2$
$3^3 \bmod 7 = 6$
$3^4 \bmod 7 = 4$
$3^5 \bmod 7 = 5$
$3^6 \bmod 7 = 1$ } all values are there. $\alpha = 3$ is primitive root

~~$\alpha = 4$~~ ✗
$4 \bmod 7 = 4$
$4^2 \bmod 7 = 2$
$4^3 \bmod 7 = 1$
$4^4 \bmod 7 = 4$ } repeat

~~$5 \bmod 7 =$~~
~~$5^2 \bmod 7 = 4$~~

example: for above q, α assume $x_A = 3$, $x_B = 4$, find y_a, y_b, k_a, k_b.

$$y_a = (\alpha)^{x_a} \bmod q = (3^3) \bmod 7 = 6$$
$$y_b = (\alpha)^{x_b} \bmod q = (3^4) \bmod 7 = 4$$

$k_a = (y_b)^{x_a} \bmod q$

$= (4)^3 \bmod 7$

$= 1$

$k_b = (y_a)^{x_b} \bmod q$

$= (6)^4 \bmod 7$

$= 1$

$k_a = (y_b)^{x_a} \bmod q = \alpha^{x_b x_a} \bmod q$

$k_b = (y_a)^{x_b} \bmod q = \alpha^{x_a x_b} \bmod q.$

both are equal.

Applications of DHKE

SSH,

SSL/TLS

ElGamal Cryptosystem:

It uses primitive root.

A number 'g' is a primitive root modulo n if every number a coprime to n is congruent to the power of g modulo n.

i.e, a is coprime to n,
then there is some integer k for which

$$g^k = a \bmod n.$$

this g is called primitive root mod n.

g is a primitive root mod n iff g is a generator of the multiplicative group of integers mod n.

(Eg) consider no. 3.
This is ~~not~~ primitive root mod 7.

$3^1 = 3 \bmod 7$
$3^2 = 2 \bmod 7$
$3^3 = 6 \bmod 7$
$3^4 = 4 \bmod 7$
$3^5 = 5 \bmod 7$
$3^6 = 1 \bmod 7$
$3^7 = 3 \bmod 7.$

(Eg) Let n = 14
Find out Z_n^* & primitive roots mod n.

$Z_{14}^* = \{1, 3, 5, 9, 11, 13\}$

$1^1 = 1 \bmod 14$
$1^2 = 1 \bmod 14$

primitive:
$3^1 = 3 \bmod 14$
$3^2 = 9 \bmod 14$
$3^3 = 13 \bmod 14$
$3^4 = 11 \bmod 14$
$3^5 = 5 \bmod 14$
$3^6 = 1 \bmod 14$

$5^1 = 5 \bmod 14$
$5^2 = 11 \bmod 14$
$5^3 = 13 \bmod 14$
$5^4 = 9 \bmod 14$
$5^5 = 3 \bmod 14$
$5^6 = 1 \bmod 14$
$5^7 = 5 \bmod 14$

primitive

$9^1 = 9 \bmod 14$
$9^2 = 11 \bmod 14$
$9^3 = 1 \bmod 14$
$9^4 = 9 \bmod 14$

$\times$

$11^1 = 11 \bmod 14$
$11^2 = 9 \bmod 14$
$11^3 = 1 \bmod 14$
$11^4 = 11 \bmod 14$

$\times$

121
-112
169

$13^1 = 13 \bmod 14$
$13^2 = 1 \bmod 14$
$13^3 = 13 \bmod 14$

$\times$

primitive roots are 3, 5

(Eg) Let $n = 15$

Find $\mathbb{Z}_n^*$ and primitive roots mod 15

$\mathbb{Z}_{15}^* = \{1, 2, 4, 7, 8, 11, 13, 14\}$

$1^1 = 1 \bmod 15$
$1^2 = 1 \bmod 15$

$2^1 = 2 \bmod 15$
$2^2 = 4 \bmod 15$
$2^3 = 8 \bmod 15$
$2^4 = 1 \bmod 15$
$2^5 = 2 \bmod 15$

$\times$

$4^1 = 4 \bmod 15$
$4^2 = 1 \bmod 15$
$4^3 = 4 \bmod 15$

$\times$

$8^1 = 8 \bmod 15$

x	x^1	x^2	x^3	x^4	x^5	x^6
1	1	1				
2	2	4	8	1	2	
4	4	1	4			
7	7	4	13	1	7	
8	8	4	2	1	8	
11	11	1	11			
13	13	4	7	1	13	
14	14	1	14			

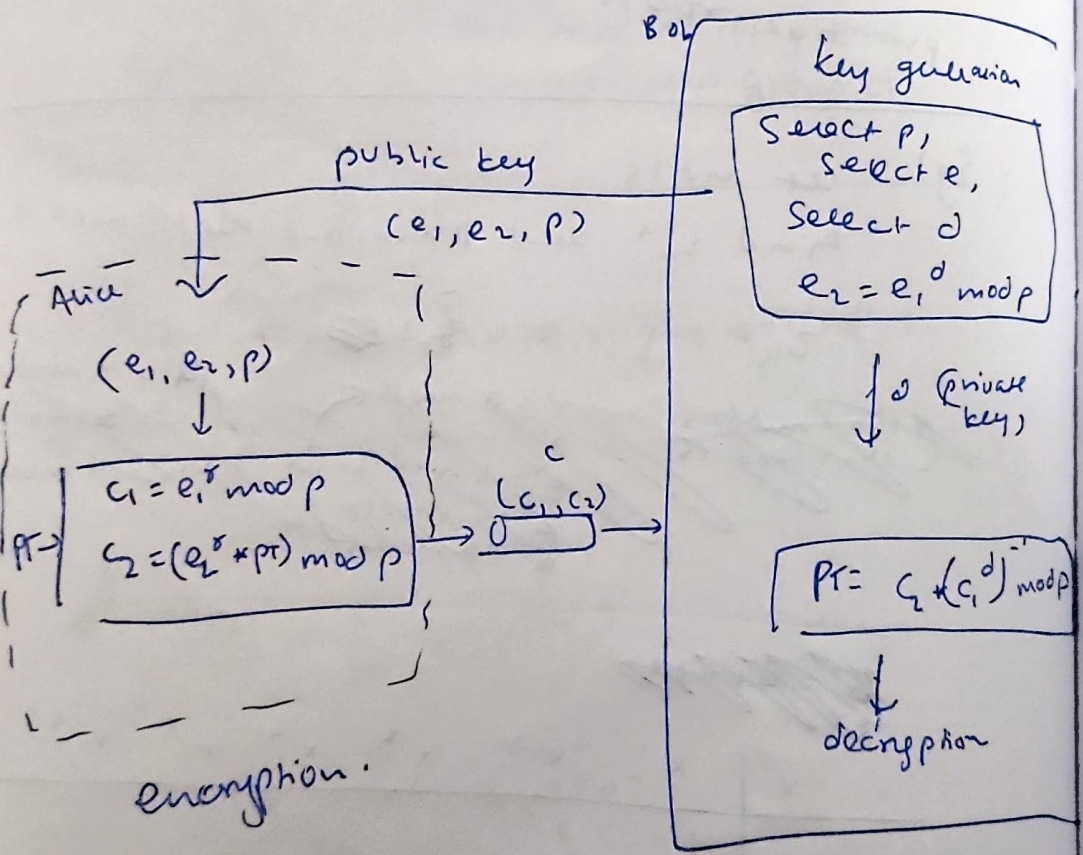
no primitive roots

Elgamal CS

given p, (a very large prime)
and e_1 (a primitive root in $G = \langle Z_p^*, * \rangle$
and r is an integer

compute $e_2 = e_1^r \bmod p$ → use fast exponentiation algo.

But given e_1, e_2, p it is infeasible to compute r which is $\log_{e_1} e_2 \bmod p$ which is discrete logarithm problem.



encryption

$$c_1 = e_1^r \bmod p$$

$$c_2 = (e_2^r * p) \bmod p$$

decryption

$$pT \leftarrow c_2 * (c_1^d)^{-1} \bmod p$$

$$\leftarrow (e_2^r * pT) * (e_1^{rd})^{-1} \bmod p$$

$$\leftarrow e_1^{rd} \; pT * (e_1^{rd}) \bmod p$$

$$\leftarrow pT \bmod p \Rightarrow \text{True}.$$

Q. $p = 11, e_1 = 2, d = 3, r = 4$

① what is public key

$$e_2 = e_1^d \bmod p = 2^3 \bmod 11 = 8$$

$$(e_1, e_2, p) \Rightarrow (2, 8, 11)$$

public key.

② Encrypt and decrypt $pT = 7$

encrypt

$$c_1 = e_1^r \bmod p = 2^4 \bmod 11 = 5$$

$$c_2 = e_2^r * pT \bmod p = 8^4 * 7 \bmod 11$$

$$= 6 \bmod 11.$$

$$= 6$$

decryption

$$pT = c_2 * (c_1^d)^{-1} \bmod 11$$

$$= 6 * (5^3)^{-1} \bmod 11$$

$$= 6 * 4^{-1} \bmod 11$$

$$= 6 * 3 \bmod 11 = 7 \bmod 11$$

Q. $p = 17$, $e_1 = 6$, $d = 5$, $r = 10$

key gen:

$e_1 = 6$, $p = 17$

$e_2 = e_1^d \bmod p = 6^5 \bmod 17$

$= 7 \bmod 17$

$(6, 7, 17)$

encrypt PT = 13

$C_1 = e_1^r \bmod p = 6^{10} \bmod 17 = 15$

$C_2 = e_2^r \wedge PT \bmod p = 7^{10} \times 13 \bmod 17$

$= 9$

$(15, 9)$

decrypt

$PT = C_2 (C_1^d)^{-1} \bmod p$

$= 9 (15^5)^{-1} \bmod 17$

$= 9 \times 2^{-1} \bmod 17$

$= 9 \times 9 \bmod 17$

$= 13$

---

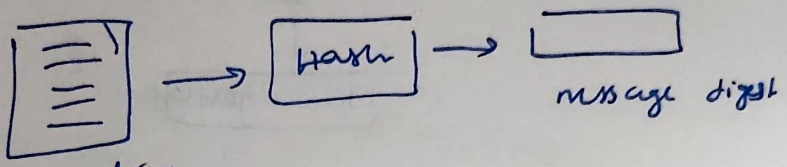
randomise r, so chosen PT attack cant be executed.

Message integrity:

occasions where we check message integrity and how we can preserve.

fingerprint / signature (physical integrity).
to put at the bottom of document.

message and message digest:



hash: for accessing fast (like maps, hash tables, etc.)

for checking integrity, compare between current and previous digest then if it is same then we can say its proper.

problem: no collision
(two messages have same cached.)

Cryptographic hash fn

should satisfy

▶ preimage resistance.

preimage attack: Given $y = h(m)$, Find m' such that $y = h(m')$

no two values should have same hash value.

▶ second preimage resistance

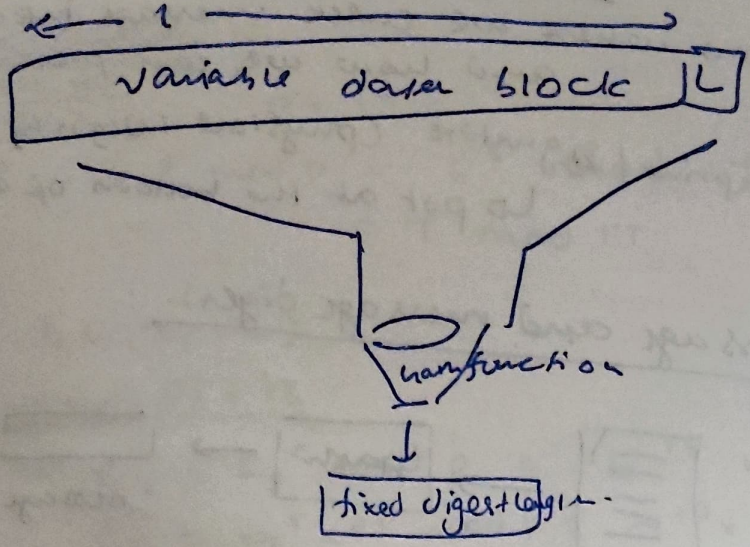
second preimage attack: Given $m, h(m)$, Find $m' \neq m$ such that $h(m) = h(m')$

if they find, attacker replaces m with m'

▶ Collision resistance

Given none, Find $m' \neq m$ such that $h(m') = h(m)$

Cryptographic hash fn



any change in bit in data would give different hash to the data.

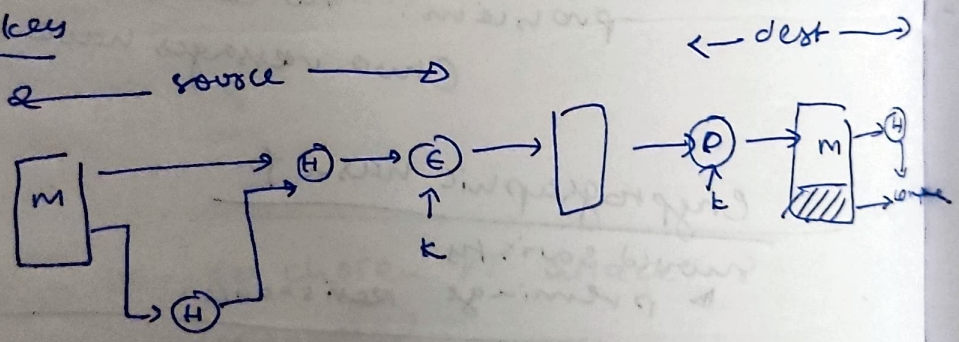
↓

integrity of message.

but do not authenticate sender.

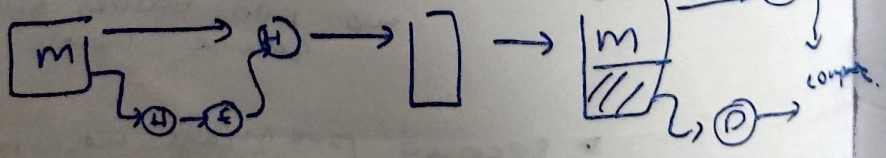
MAC → message authentication + message integrity.

Symmetric key



message encrypted.

Costlier since lesser chance of attacker knows.



message unencrypted.

keyed hash

message encrypted

message unencrypted.

Applications of hash fns

- data integrity
- password storage
  - protect the password
    store as (uid, H(Pass))
  - check if hash of password matches
    ↓
    intruder can only see H(password) so wont know password.

---

other Data Integrity check

- use checksum (CRC)
  ↓
  provides assurance to user about integrity.

Limitations:

- do not provide assurance about the originality.

Diagram.

Hash function requirements

- variable input size
- fixed o/p size
- efficiency
- pseudorandomness
- preimage resistance
- second " "
- collision resistance.

popular hash functions:

- MD
- SHA
- RIPEMD
- Whirlpool.

Message digest (MD)

↳ compare checksum
↳ MD5 was widely used.
↳ found collision in MD5.

SHA → (eg SHA0, SHA1, SHA2, SHA3)

SHA1
↳ 160 bit hash value
↳ used in SSL (secure socket layer)
↳ for TLS in HTTPS.

SHA2
↳ more secure (based on Merkle Dagmard).

(check SHA3, SHA512 and diags in PPT)

RIPEMD (160bit)

- secure replacement for 128 bits.
- bitcoin uses SHA512 and RIPEMD160.

Whirlpool

- derived from modified version of AES.
- 512 bit function.
- operating on function < 512 bits.

attacks on hash function:

- brute force/cryptanalysis
- preimage attack
- collision attack.