previous video).

eg $-10 \bmod 2$

2) congruent modulo

Two integers a and b are said to be congruent modulo n if

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

This is written as:

$a \equiv (b \bmod n)$ or $b \equiv (a \bmod n)$

eg $73 \equiv 4 \pmod{23}$ means $73 \bmod 23 = 4 \bmod 23$

Properties

(i)

(ii)

(iii)

n divides (a-b)

$(a-b)$

$b \equiv a \pmod n$

$\pmod n$

$= c \pmod n$

the keys.

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

This is written as:

$a \equiv (b \bmod n)$ or $b \equiv (a \bmod n)$

eg $73 \equiv 4 \pmod{23}$ means $73 \bmod 23 = 4 \bmod 23$

Properties of congruence

(i) $a \equiv b \pmod n$ if $n \mid (a-b)$ (n divides (a-b))

(ii) $a \equiv b \pmod n$ implies $b \equiv a \pmod n$

(iii) if $a \equiv b \pmod n$ and $b \equiv c \pmod n$ then, $a \equiv c \pmod n$

Set of residues or residue classes modulo n

$$Z_n = \{0, 1, 2, \ldots, (n-1)\}$$

eg $Z_6 = \{0, 1, 2, 3, 4, 5\}$

each integer in Z_n represents a residue class.

[redacted]

mod 2) mod 2

d 2

Euler's Toitient function

→ It is represented using phi as $\phi(n)$ and may also be called Euler's phi function.

→ Euler's toitient fn is defined as the no. of +ve integers less than n that are coprime to n.

$n \geq 1$

$\phi(5) = \{1, 2, 3, 4\}$

$\phi(6) = \{1, 5\}$

no. of elements in these sets is the toitient fn.

Note → Two integers a, b are said to be relatively prime, mutually prime or integer/factor

fn is defined as the no. of +ve integers less than n that are coprime to n.

$n \geq 1$

$\phi(1) = 1$

$\phi(5) = \{1, 2, 3, 4\} = \}4$

$\phi(6) = \{1, 5\} \}2$

no. of elements in these sets is the toitient fn.

① 2 3 4 5
× × × ✓

Note → Two integers a, b are said to be relatively prime, mutually prime or coprime if the only +ve integer/factor that divides both of them is 1.

Now, when $n \rightarrow$ prime $\quad \phi(n) = n - 1$

eg $\phi(5) = 4$ // we have seen the eg. above

$\phi(23) = 23 - 1$

$= 22$

Now, when $n \rightarrow$ prime $\quad \phi(n) = n-1$

eg $\phi(5) = 4$ // we have seen the eg. above

$\phi(23) = 23-1$

$= 22$

Also, $\phi(a*b) = \phi(a) * \phi(b)$ // a and b should be coprime

eg $\phi(35) = \phi(7) * \phi(5)$

$= 6*4 = 24$

eg $\phi(12) = \phi(3) * \phi(4) = 2*2 = 4$

eg $\phi(15) = \phi(3) * \phi(5) = 2*4 = 8$

{1, 2, 4, [redacted], 7, 8, [redacted] 11, 13, 14}

like and share :)

Euler's Theorem

→ also called Fermat - Euler theorem or Euler's totient theorem

(a/c to wikipedia)

Euler's theorem states that if x and n are coprime positive integers, then

$$x^{\phi(n)} \equiv 1 \bmod n$$

where $\phi(n)$ → euler's totient fⁿ.

Note → It is a generalized version of fermat's Theorem.

eg let $x = 11$, $n = 10$ both are coprime

∴ we can represent them as

$x^{\phi(n)} \bmod n = 1 \bmod n$

where $\phi(n)$ → euler's toitient f^n.

Note → It is a generalized version of fermat's Theorem.

eg let $x = 11, n = 10$ both are coprime

∴ we can represent them as

$$11^{\phi(10)} \equiv 1 \bmod 10$$

$$11^4 \equiv 1 \bmod 10$$

$$14641 \equiv 1 \bmod 10$$ which is true

$$\phi(10) = \phi(2) * \phi(5)$$
$$= 1 * 4$$
$$= 4$$

Note → $$x^{\phi(n) \cdot a} \equiv 1 \bmod n$$

$11^8 = 214,358,881$

$\phi(4) = 2$

$n = 4$

mod 4

$11^4 \equiv 1 \bmod 10$

$\boxed{14641 \equiv 1 \bmod 10}$ which is true

$\phi(10) = \phi(2) * \phi(5)$

$= 1 * 4$

$= 4$

$\phi(a*b) = \phi(a) * (\phi(b)$

$\phi(n) = n-1$

Note → $\boxed{x^{\phi(n)\cdot a} \equiv 1 \bmod n}$

$11^8 = 214,358,881$

i.e $11^{4*2} \equiv 1 \bmod 10$

$11^{40} \equiv 1 \bmod 10$

i.e. any multiple of $\phi(n)$ will give the same result.

Solve by Eulers Theorem

Ques $4^{99} \bmod 35$

Eulers Theorem

$$x^{\phi(n)} \equiv 1 \bmod n$$

$x, n \rightarrow$ coprime +ve integer

$x = 4, \quad n = 35$

by eulers theorem,

$$4^{\phi(35)} = 1 \bmod 35$$

$$\phi(35) = \phi(7) \cdot \phi(5)$$
$$= 6 * 4$$
$$\phi(35) = 24$$

$$4^{24} \equiv 1 \bmod 35 \quad \text{---(1)}$$

$$4^{99} \rightarrow 4^{24(4)} \cdot 4^3$$

$$\therefore 4^{99} \bmod 35 = 4^{24 \times 4 + 3} \bmod 35$$
$$= (4^{24})^4 \cdot 4^3 \bmod 35$$
$$= (4^{24})^4 \bmod 35 \cdot 4^3 \bmod 35$$

$$\therefore (a \times b) \bmod n \equiv (a \bmod n)(b \bmod n)$$

$\therefore$ $4^3 \bmod 35$

Fermat's Theorem / Fermat's Little Theorem

→ especial case of eulers theorem

If n is prime and 'x' is a +ve integer not divisible by n then

$$x^{n-1} \equiv 1 \bmod n$$

$\phi(n) = n-1$

n → prime no.

x is not divisible by n

Also, x, n → coprime

eg $x = 3, \quad n = 5$

$3^{5-1} = 3^4 = 81$

$\therefore 81 \equiv 1 \bmod 5$

euler theorem

$$x^{\phi(n)} \equiv 1 \bmod n$$

$$x^{n-1} \equiv 1 \bmod n$$ } fermats theorem

$n \rightarrow$ prime no.
x is not divisible by n

$\phi(n) = n-1$

Also,
$x, n \rightarrow$ coprime

$81/5 = 14.2$

eg $x = 3, \quad n = 5$

$3^{5-1} = 3^4 = 81$

$\therefore 81 \equiv 1 \pmod 5$

euler theorem

$x^{\phi(n)} \equiv 1 \pmod n$

$x^{n-1} \equiv 1 \pmod n$ } fermats theorem

Another form of fermats theorem

$x^n \equiv x \pmod n$ in books $a^p \equiv a \pmod p$

eg $x = 3, \quad n = 5$

$x^n = 3^5 = 2[illegible] \equiv 3 \pmod 5$

$2[illegible] \pmod 5$

inder

$= 1, a_3 = 3$

$7, m_3 = 11$

ime to one

$m_1) = 1$

CHIENESE REMAINDER THEOREM

Chinese Remainder theorem states that there always exists an "x" that satisfies the given congruence.

$x \equiv rem[0] \pmod{num[0]}$

$x \equiv rem[1] \pmod{num[1]}$

. . . .

and (num[0], num[1] ... num[n-1]) all → must be coprime to ~~each other~~ one another.

eg. $x \equiv 1 \bmod 5$

$x \equiv 3 \bmod 7$ → 5 and 7 are coprime

$m_1 = 77$

$b = 55$

$3 = 35$

$(\text{mod } m_1) = 1$

have to find value such that

ch value

$\equiv$ rem[i] (mod num[i])

. . . .

and (num[0], num[1] ... num[n-1]) all → must be coprime to ~~each other~~ one another.

eg. $x \equiv 1 \text{ mod } 5$

$x \equiv 3 \text{ mod } 7$ → 5 and 7 are coprime

we have to find this $x = 31$

eg $x \equiv 2 \text{ mod } 3$

$x \equiv 3 \text{ mod } 4$

$x \equiv 1 \text{ mod } 5$

$gcd(3,4) = gcd(4,5)$

$= gcd(3,5) = 1$

Then only x exists

here $x = 11$

Explain Chinese Remainder Theorem

if $x \equiv a_1 \pmod{m_1}$

$x \equiv a_2 \pmod{m_2}$

$x \equiv a_3 \pmod{m_3}$

(i) $\gcd(m_1, m_2) = \gcd(m_2, m_3) = \gcd(m_3, m_1) = 1$

ie all coprime

(ii) $x = (M_1 X_1 a_1 + M_2 X_2 a_2 + M_3 X_3 a_3 + \cdots + M_n X_n a_n) \bmod M$

$M = m_1 * m_2 * m_3 \cdots m_n$

$M_i = \frac{M}{m_i}$ eg $M_1 = \frac{M}{m_1} = \frac{m_1 m_2 m_3}{m_1} = m_2 m_3$

$\therefore \boxed{M_1 = m_2 m_3}$

Similarly $M_2 = m_1 m_3$

$M_3 = m_1 m_2$

$\frac{77}{5} = 3$

$m_2 \times m_3 \cdots m_n$

$$\boxed{M_i = \frac{M}{m_i}}$$ eg $M_1 = \frac{M}{m_1} = \frac{m_1 m_2 m_3}{m_1} = m_2 m_3$

$$\therefore \boxed{M_1 = m_2 m_3}$$

Similarly → $M_2 = m_1 m_3 = \frac{M}{m_2} = m_1 m_2 m_3 = M$

" $M_3 = m_1 m_2$

To calculate X_i → multiplicative inverse of M_i

$$M_i X_i \equiv 1 \bmod m_i$$

eg $M_1 X_1 \equiv 1 \bmod m_1$

[redacted]/books.

Numerical Chinese Remainder Theorem

eg $x \equiv 1 \pmod 5$
$x \equiv 1 \pmod 7$
$x \equiv 3 \pmod{11}$

$\longrightarrow$ $x \equiv a_i \bmod(m_i)$

So $a_1 = 1, a_2 = 1, a_3 = 3$

$m_1 = 5, m_2 = 7, m_3 = 11$

Solu $\rightarrow$ Since 5, 7 and 11 all are relatively prime to one another. So, we can find x

i.e $\gcd(5,7) = \gcd(7,11) = \gcd(11,5) = 1$

$M = m_1 * m_2 * m_3 = 5 * 7 * 11 = 385$

$\boxed{M = 385}$

$M_1 = \frac{M}{m_1} = m_2 m_3 = 7 * 11 = 77$

$M_2 = m_1 m_3 = 5 * 11 = 55$

$M_3 = m_1 m_2 = 5 * 7 = 35$

$M_1 = 77$
$M_2 = 55$
$M_3 = 35$

$M_1 = \frac{M}{m_1} = m_2 m_3 = 7 * 11 = 77$

$M_2 = m_1 m_3 = 5 * 11 = 55$

$M_3 = m_1 m_2 = 5 * 7 = 35$

$M_1 = 77$
$M_2 = 55$
$M_3 = 35$

Now we will calculate X_i value.

$M_1 X_1 \equiv 1 \pmod{m_1}$ ie $M_1 X_1 \pmod{m_1} = 1$

$77 \cdot X_1 \pmod 5 = 1$

$2 X_1 \pmod 5 = 1$

$\therefore X_1 = 3$

Similarly $M_2 X_2 \equiv 1 \pmod{m_2}$

$55 X_2 \bmod 7 = 1$

$6 X_2 \pmod 7 = 1$

$X_2 = 6$

we have to find a value such that $6 * x_2$ dt value (7 k min multiple + 1) ho

$\therefore$ $\boxed{x_3 = 6}$

Now, $a_1 = a_2 = 1$ $a_3 = 3$

$m_1 = 5$ $m_2 = 7$ $m_3 = 11$

$M_1 = 77$ $M_2 = 55$ $M_3 = 35$ $M = 385$

$x_1 = 3$ $x_2 = 6$ $x_3 = 6$

$x = (M_1 x_1 a_1 + M_2 x_2 a_2 + M_3 x_3 a_3) \bmod M$

$x = (77(3)(1) + 55(6)(1) + 35 \times 6 \times 3) \bmod (385)$

$x = (231 + 330 + 630) \bmod 385$

$x = 1191 \bmod 385$

$\boxed{x = 36}$

If u dont have calculator

$1191 - 385 = 806$

THE EUCLIDEAN / EUCLID'S ALGORITHM

used for determining GCD of two positive integers (generally large nos.)

$$gcd(a,b) = gcd(b, a \bmod b)$$
$$gcd(a,0) = a$$

eg:

gcd(1025, 35) i.e gcd(a, b)

= gcd(b, a % b) = gcd(35, 10)

= gcd(10, 5)

= gcd(5, 0)

$600 = 2^3 \cdot 5^2 \cdot 3^1$

$45 = 2^0 \cdot 5^1 \cdot 3^2$

GCD / HCF → $2^0 \cdot 5^1 \cdot 3^1$

= 15

1025 = 35 * 29 + 10

35 = 10 * 3 + 5

$2^1 \times 5^2$

$\times 5^1$

5^2

5^1

24
25 2
120
48 ×
600

35
×29
1015

$= \gcd(b, a \% b) = \gcd(35, 10)$

$= \gcd(10, 5)$

$= \gcd(5, 0)$

$= 5$ Answer

$45 = 2^0 \cdot 5^1 \cdot 3^2$

GCD / HCF $\rightarrow$ $2^0 \cdot 5^1 \cdot 3^1$

$= 15$

$1025 = 35 * 29 + 10$

$35 = 10 * 3 + 5$

$10 = 5 * 2 + 0$

$\therefore 5 =$ Ans

eg $\gcd(11, 7) \equiv \gcd(a, b)$

$\gcd(b, a \% b) = \gcd(7, 11 \% 7)$

$= \gcd(7, 4)$

$= \gcd(4, 7 \bmod 4) = \gcd(4, 3)$

$= \gcd(3, 4 \% 3) = \gcd(3, 1)$

$= \gcd(1, 3 \% 1)$

$= \gcd(1, 0)$

GCD of 2740 and 1760

q	r_1	r_2	r
1	2740	1760	980
1	1760	980	780
1	980	780	200
3	780	200	180
1	200	180	20
9	180	20	0
	20	0	

→ $r = r1 \% . r2$

1760
×2
3520

we will calculate the remainder (r) as $r1 \% . r2$

Then replace
& r_1 by r_2
r_2 by r

Continue until $r_2 = 0$. At this time we stop and GCD $(a,b) = r_1$ value.

EXTENDED EUCLIDEAN ALGO

Ques find the GCD of (161, 28) and the value of "s" and "t".

Given two integers a and b we often need to find 2 integers, s and t such that

$$s*a + t*b = gcd(a,b)$$

So, extended euclidean algo can calculate gcd(a,b) and the value of s and t.

↓

coeff. of Bezout's identity (in Number theory).

EXTENDED EUCLIDEAN ALGO

Ques find the GCD of (161, 28) and the value of "s" and "t".
a, b

soln

$s = s_1 - q s_2$ $\quad t = t_1 - q t_2$

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
5	161	28	21	1	0	1	0	1	-5
1	28	21	7	0	1	-1	1	-5	6
3	21	7	0	1	-1	4	-5	6	-23
	(7)	0		-1	4		6	-23	
				s			t		

GCD = 7

s = -1

t = 6

-5 - (3(-1))

-5 - 18

=

$(-1) \times 161 + 6 \times 28 = 7$ The answers can be tested because we have

EXAMPLE 2.10 Given $a = 17$ and $b = 0$, find gcd (a, b) and the values of s and t.

SOLUTION We use a table to follow the algorithm.

q	r_1	r_2	r	s_1	s_2	s	t_1	t_2	t
	17	0		**1**	**0**		**0**	**1**	

Note that we need no calculation for q, r, and s. The first value of r_2 meets our termination condition. W
gcd $(17, 0) = 17$, $s = 1$, and $t = 0$. This indicates why we should initialize s_1 to 1 and t_1 to 0. The answer
be tested as shown below:

$$(1 \times 17) + (0 \times 0) = 17$$

EXAMPLE 2.11 Given $a = 0$ and $b = 45$, find gcd (a, b) and ... s and t.

SOLUTION We use a table to follow the algorithm.

q	r_1	r_2	r
0	0	45	

MULTIPLICATIVE INVERSE

in CRYPTOGRAPHY

Ques Find the multiplicative Inverse of 11 in Z_{26}.

Solu→ Before solving check, M.I possible or not

if $\boxed{gcd(x, n) = 1}$ possible

$gcd(11, 26) = 1$ true ∴ possible

$T = t_1 - q\,t_2$

q	r_1	r_2	r	t_1	t_2	t
	26	11				

solve until $r_2 \neq 0$

when $r_2 = 0$ value of t_1 will be M.I.

$\boxed{Z_n}$ → set of residues $Z_n = \{0, 1, 2, \ldots, (n-1)\}$

eg $Z_2 = \{0, 1\}$ $Z_4 = \{0, 1, 2, 3\}$

$Z_5 = \{0, 1, 2, 3, 4\}$

if $\boxed{\gcd(x, n) = 1}$ possible

$\gcd(11, 26) = 1$ true ∴ possible

q	r_1	r_2	r	t_1	t_2	t	$T = t_1 - q t_2$
2	26	11	4	0	1	-2	0-2(1)
2	11	4	3	1	-2	5	
1	4	3	1	-2	5	-7	
3	3	1	0	5	-7	26	5-3(-7)
	1	0		-7	26		

solve until $r_2 \neq 0$

when $r_2 = 0$ value of t_1 will be M.I.

Multiplicative inverse of a number in Cryptography

Ques Find the Multiplicative Inverse of 11 in Z_{26}.

Soln → Before solving check. M.I possible or not

if $\boxed{\gcd(x, n) = 1}$ possible

$\gcd(11, 26) = 1$ true ∴ possible

q	r_1	r_2	r	t_1	t_2	t	$T = t_1 - q t_2$
2	26	11	4	0	1	-2	0-2(1)
2	11	4	3	1	-2	5	
1	4	3	1	-2	5	-7	
3	3	1	0	5	-7	26	5-3(-7)
	1	0		-7	26		

solve until $r_2 \neq 0$

when $r_2 = 0$ value of t_1 will be M.I.

-7
+26
19

$Z_{26} = \{0, 1, 2, \ldots 25\}$

multiplicative Inverse of
11 in $Z_{(26)}$

$$11 * (M^{-1}) \equiv 1 \bmod 26$$

$$11 * 19 \equiv 1 \bmod 26$$

$$\boxed{209 \equiv 1 \bmod 26}$$

Ques Find the multiplicative Inverse of

Sol→ Before solving check M.I possible or not

if $\boxed{\gcd(x, n) = 1}$ possible

$\gcd(11, 26) = 1$ true

q	r_1	r_2	r	t_1	t
2	26	11	4	0	
2	11	4	3	1	
1	4	3	1	-2	
3	3	1	0	5	

AUTHENTICATION FUNCTIONS

Authentication → verifying the users identity

Raman ⟶ John

An authenticator must be there to authenticate the message.

Types of Authentication / Types of fns to produce authentication

(i) Message Encryption (ciphertext act as authenticator)

(ii) MAC (message authentication code)

→ we will have some authentication fn and we apply them on the plaintext along with the key which produces a fixed length code called MAC

Types of Authentication / Types of fns to produce authentication

(i) Message Encryption (ciphertext act as authenticator)

(ii) MAC (message authentication code)

→ we will have some authentication fn and we apply them on the plaintext along with the key which produces a fixed length code called MAC

1MB → 1KB

message

$C(M, K) =$ fixed length code (MAC)

authentication fn, Key

This will act as an authenticator here.

(iii) Hash functions (H)

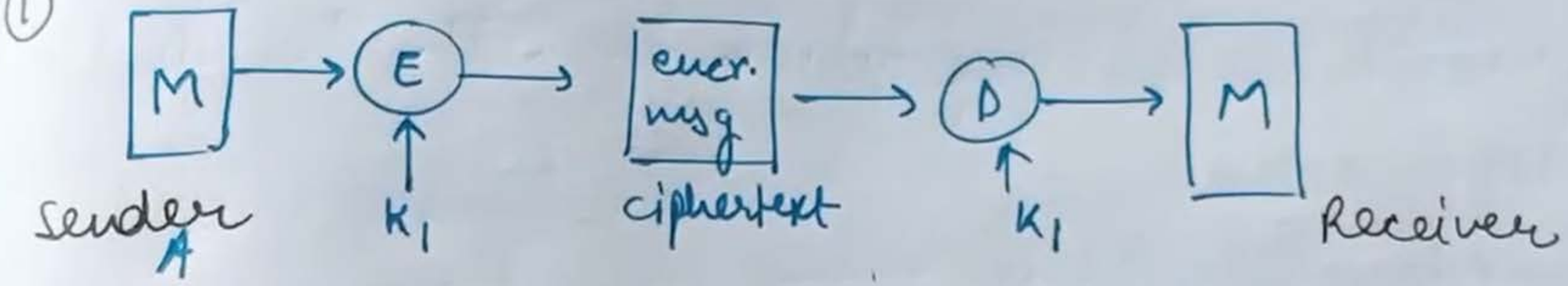
$H(M) =$ fixed length code (Hash code 'h')

Msg

act as an

1. Message encryption → ciphertext is an authentica-tor

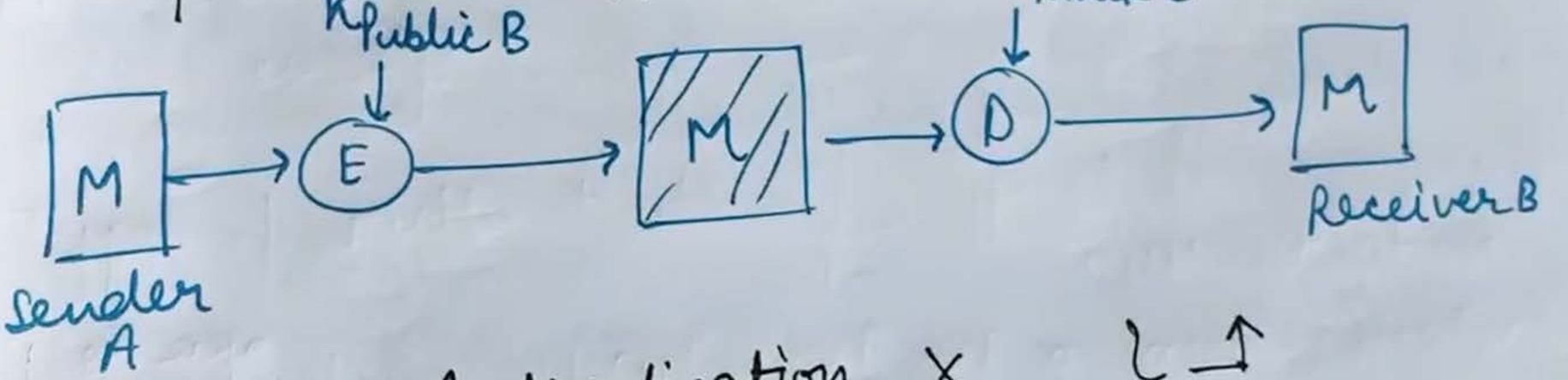
(1)



→ key K_1 shared only b/w sender & Receiver only.

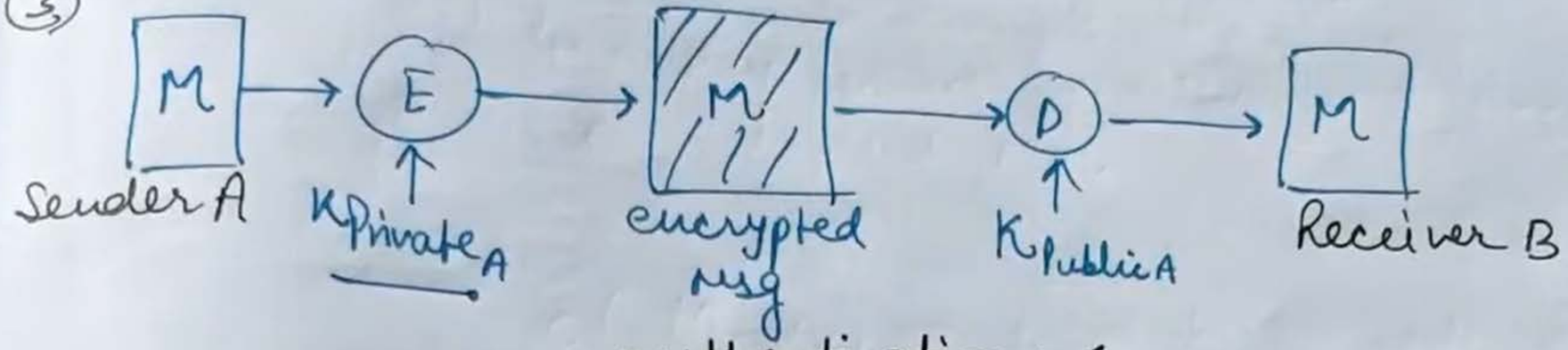
for asymmetric encryption,

(2)



Authentication ✗
confidentiality ✓

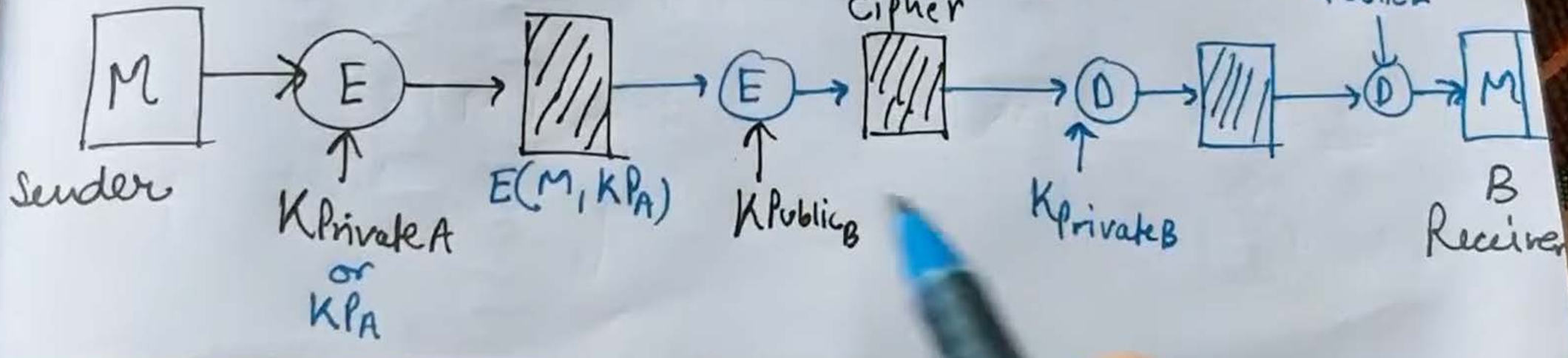
(3)



authentication ✓
confidentiality X

(4)

To get both, use dual encryption & decryption
and also signature



ION

e 5 principles of security.

thenticity of the msg is v.imp.

must be there to authenticate

e authentication

n

tion Code (MAC)

MAC (message authentication code)

→ We will use a secret key to generate a small fixed size block of data called MAC or cryptographic checksum.

→ It is then appended with the message.

→ The communicating parties will share a secret common key.

which will be used to create the MAC

Let A → sender
B → receiver

when A sends a msg to B, it calculates the MAC as a fn of the message and the key.

$$MAC = C(K, M)$$

where

M = input message

be there to authenticate

thentication

ipher text

Code (MAC)

value

small fixed size block of data called MAC or cryptographic checksum.

→ It is then appended with the message.

→ The communicating parties will share a secret common key.

2MB → 2Kb which will be used to create the MAC

Let A → sender
B → receiver

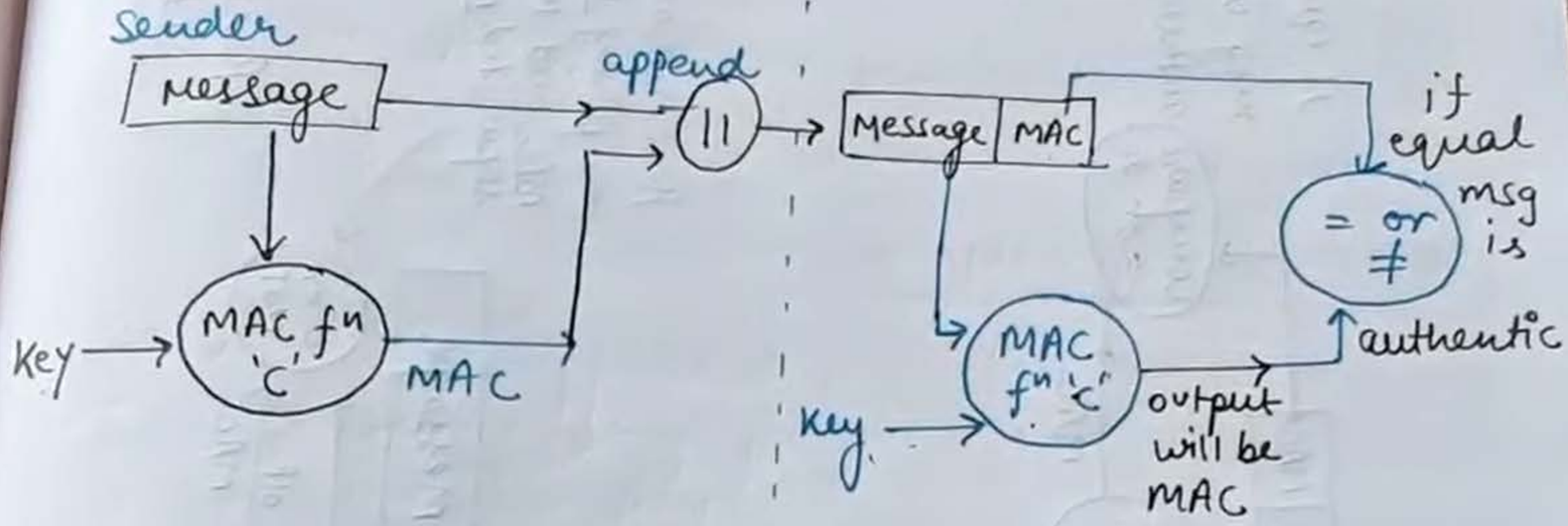
When A sends a msg to B, it calculates the MAC as a fn of the message and the key.

$$MAC = C(K, M)$$

where

M = input message
C = MAC function
K = shared secret key

(i) MAC - for authentication



we separate the msg and the MAC

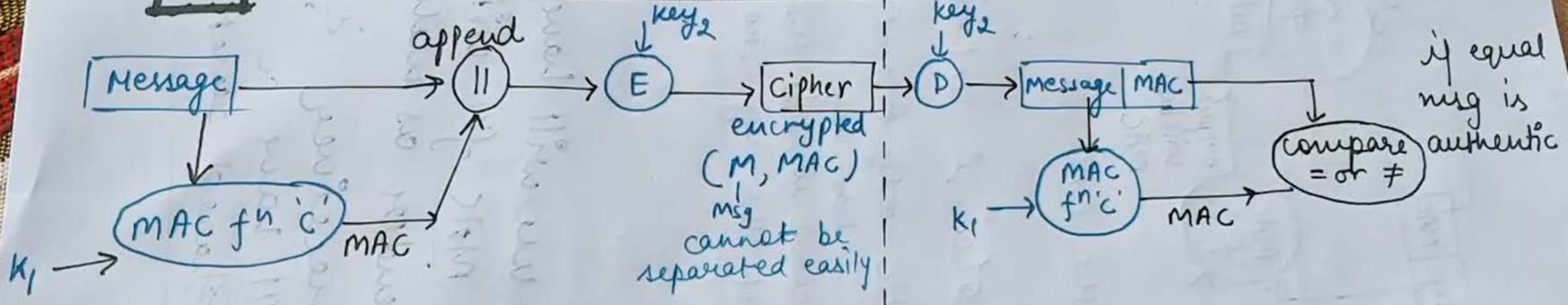
- only authentication is achieved

provides (i) ~~confidentiality~~ No confidentiality - b/c ~~only A and B has key~~

(ii) if 3rd party come in, b/w then he can get the msg ∴ no security

2) MAC - for authentication & confidentiality

(a) authentication tied to plain text

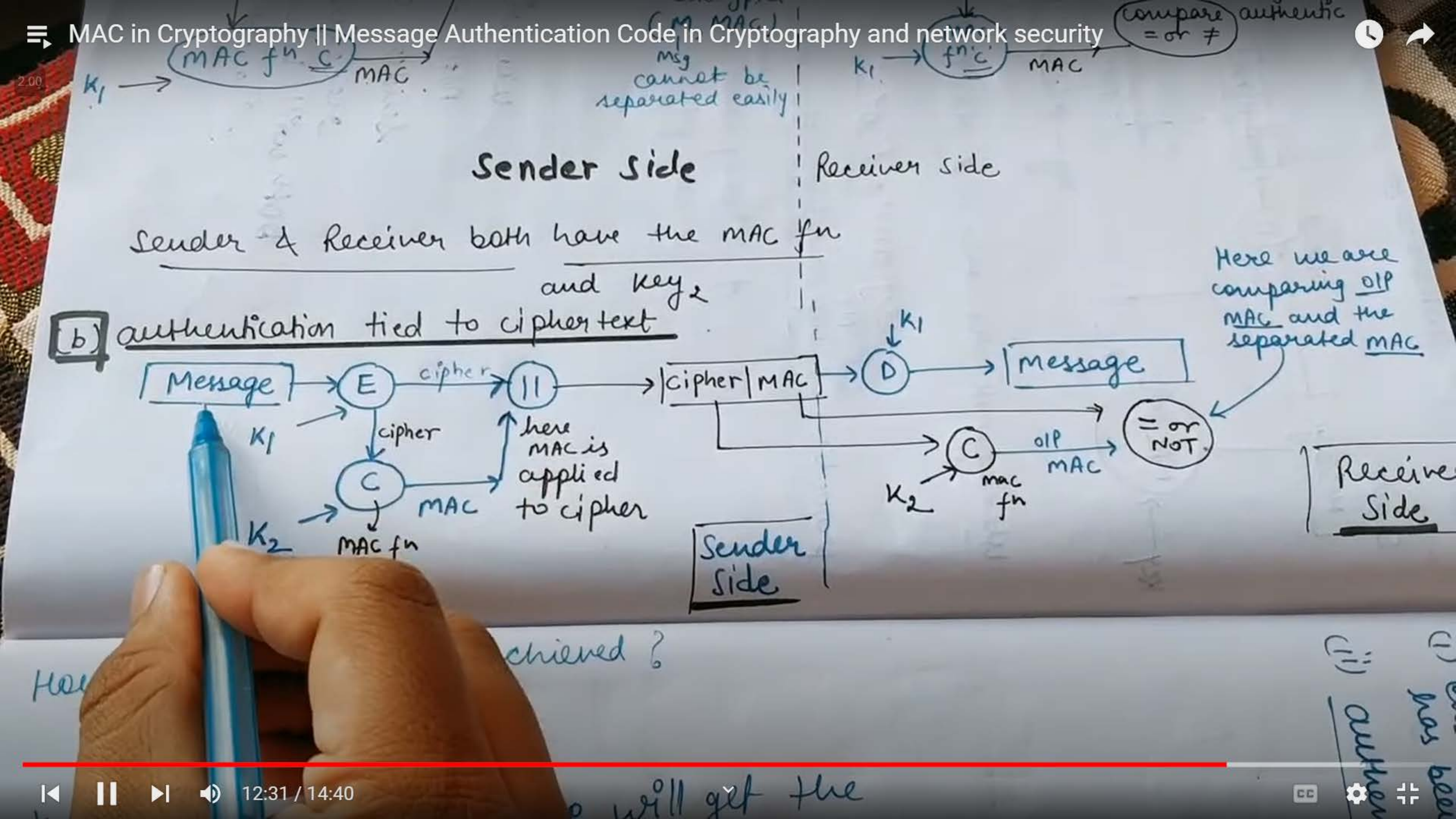


Sender Side | Receiver side

Sender & Receiver both have the MAC fn and key2

Here we are comparing of MAC and th

MAC in Cryptography || Message Authentication Code in Cryptography and network security



12:31 / 14:40

I give you
Rs

B

ember

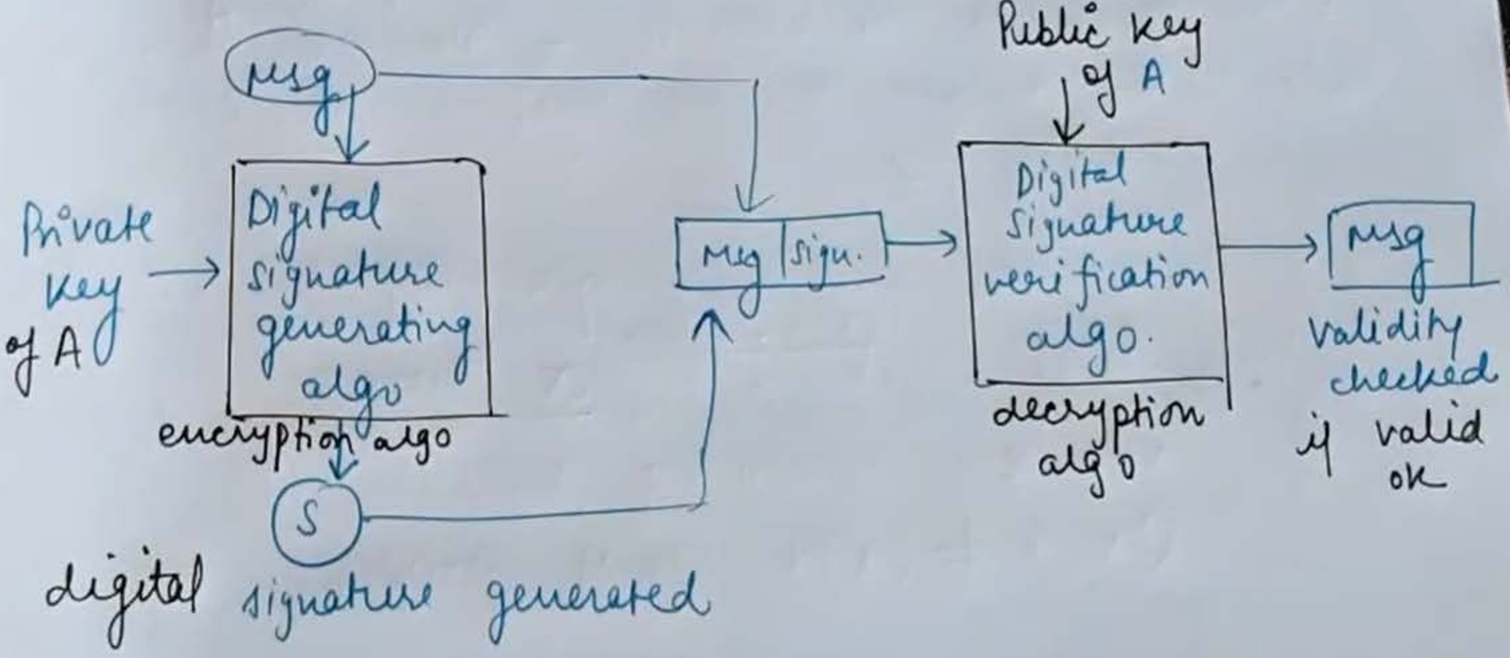
Digital signature

→ vimp role in e-commerce, online transaction, etc.

→ based on asymmetric key cryptography
encryption → private key
decryption → public key

→ used for msg authentication & non repudiation & msg integrity

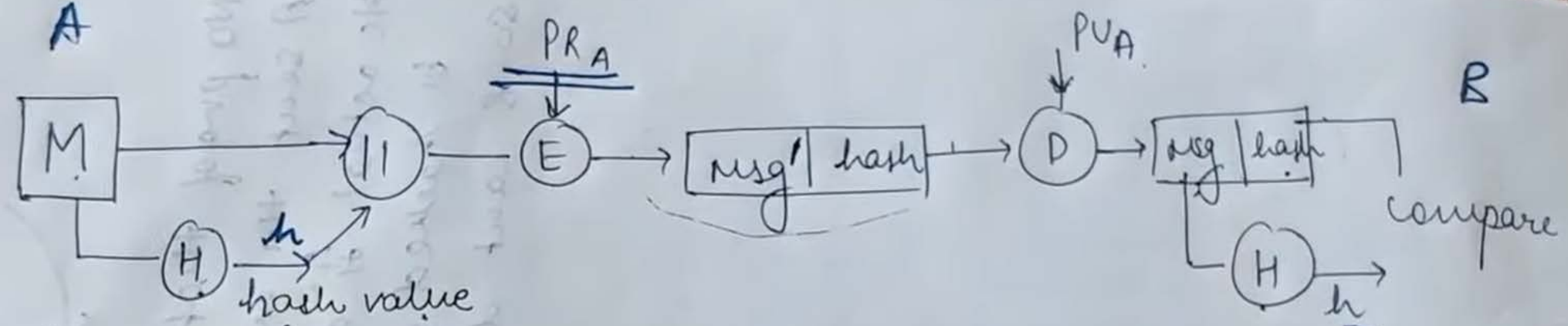
→ not used for confidentiality



code

re.

encryption → private key
decryption → public key
→ used for msg authentication & non repudiation & msg integrity
→ not used for confidentiality

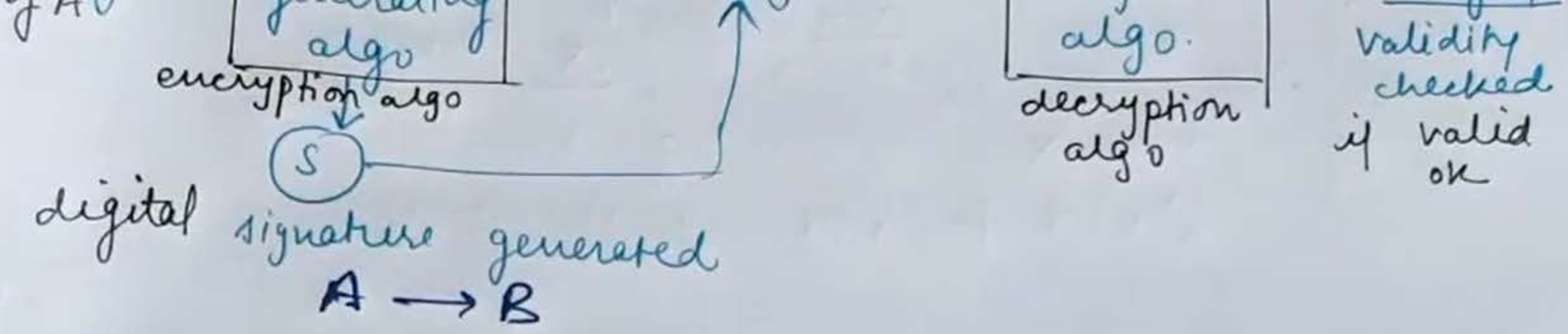


general concept of digital signature

H → hash fn
h → hash code
|| → append

Note → The signature must use some info. unique to the sender to prevent both forgery

signature



digital signature generated

A → B

→ also provides msg integrity

b/c if msg changed then at receiver side, we will not get the exact msg.

↓

achieved using Hashing concept using msg digest / hash values

sign
Public
decrypt
h'
compare
same valid else not

Note → When we sign a document digitally,
= we send the signature as a separate document.

Sender sends 2 docs → msg & signature.

Non repudiation
achieved by using a trusted 3rd party

pg 351
pg 352
confidentiality

Digital Signature

→ signature must use some info unique to the sender, to prevent forgery & denial.
→ It must be easy to produce digital signatures.
→ " " " " to verify & recognize " " .
→ we need (i) key generation algo → to generate private key
(ii) Signing Algo i/p → M and Private key, o/p → Digital sign
(iii) verifying algo → using public key & sign.

3.4 Confidentiality

A digital signature does not provide confidential communication. If confidentiality is required, the message and the signature must be encrypted using either a secret-key or public-key cryptosystem. Figure 13.5 shows how this extra level can be added to a simple digital signature scheme.

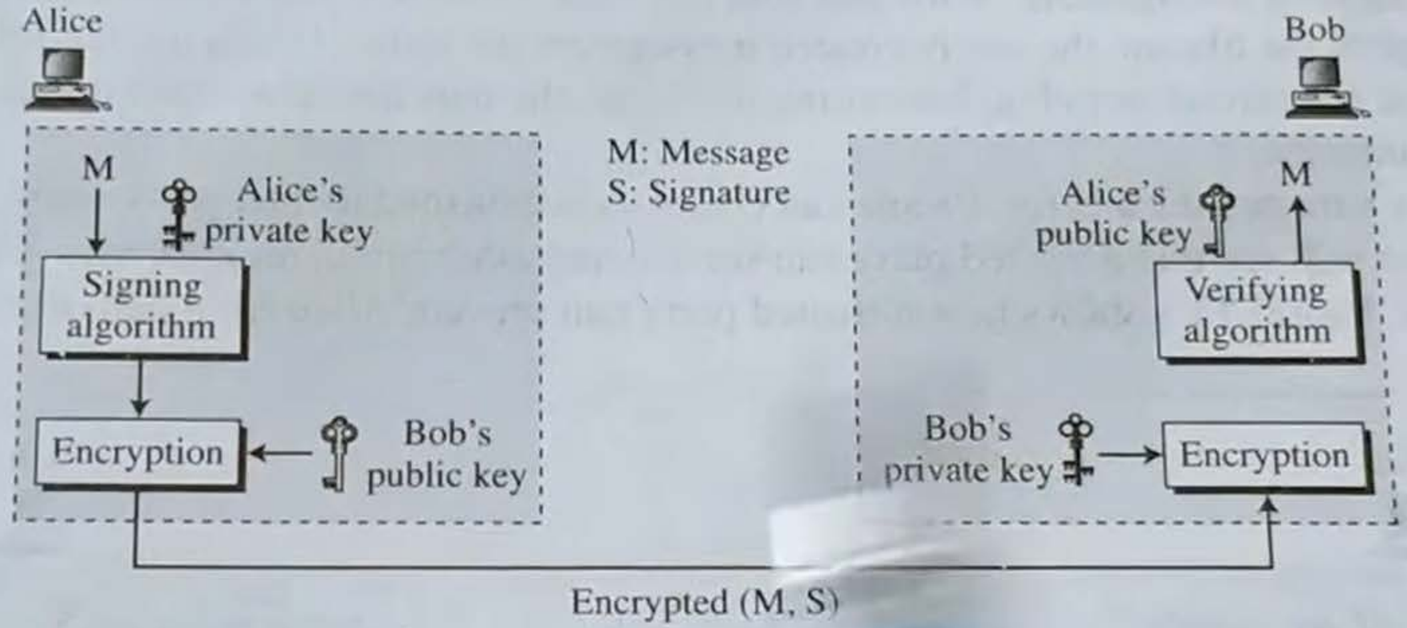


Fig. 13.5 *Adding confidentiality to a digital signature scheme*

We have shown asymmetric-key encryption/decryption to emphasize the keys used at each end. Encryption/decryption can also be done with

(iii) # HASH FUNCTIONS

→ ° Similar to MAC (msg. authentication code)
BUT it doesnot use a key.

→ ° Takes in variable size message and produce a fixed output.
↓
called Hash Code | Hash value | or Message Digest

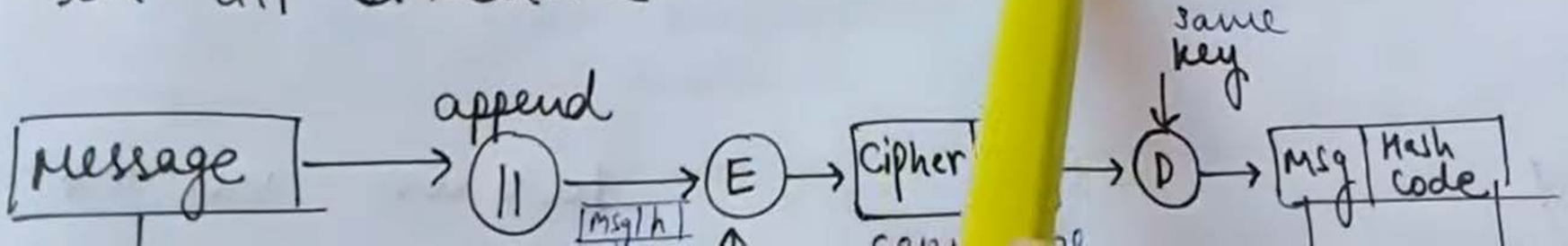
→ The only i/p is the message.
→ A hash value 'h' is generated by a fn H

H(M) = fixed length code 'h'
↑ variable length msg ↑ hash code

They are also called Compression Functions.

There are dif methods to provide authentication in dif situations

(i)

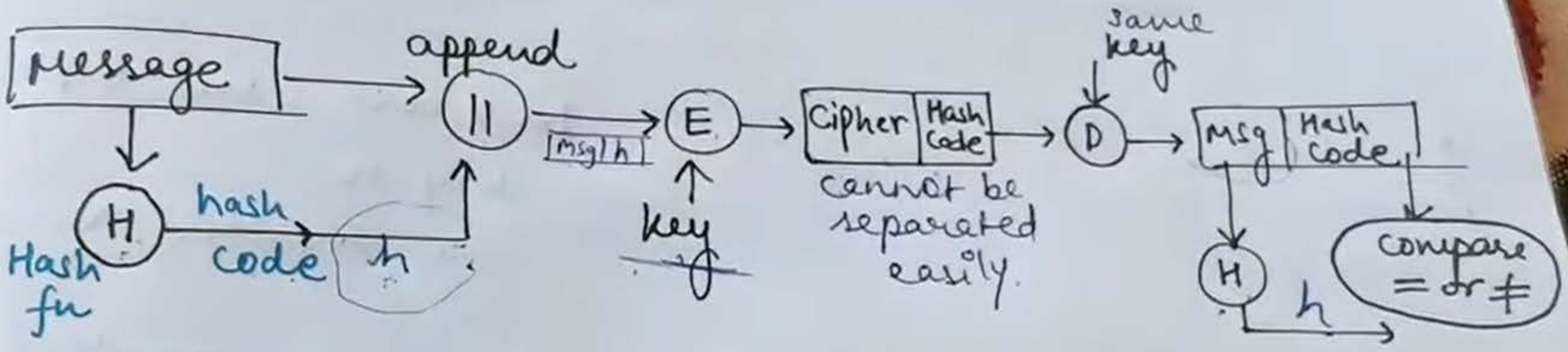


variable length msg → hashcode

These are also called Compression Functions.

There are dif methods to provide authentication in dif situations

(i)

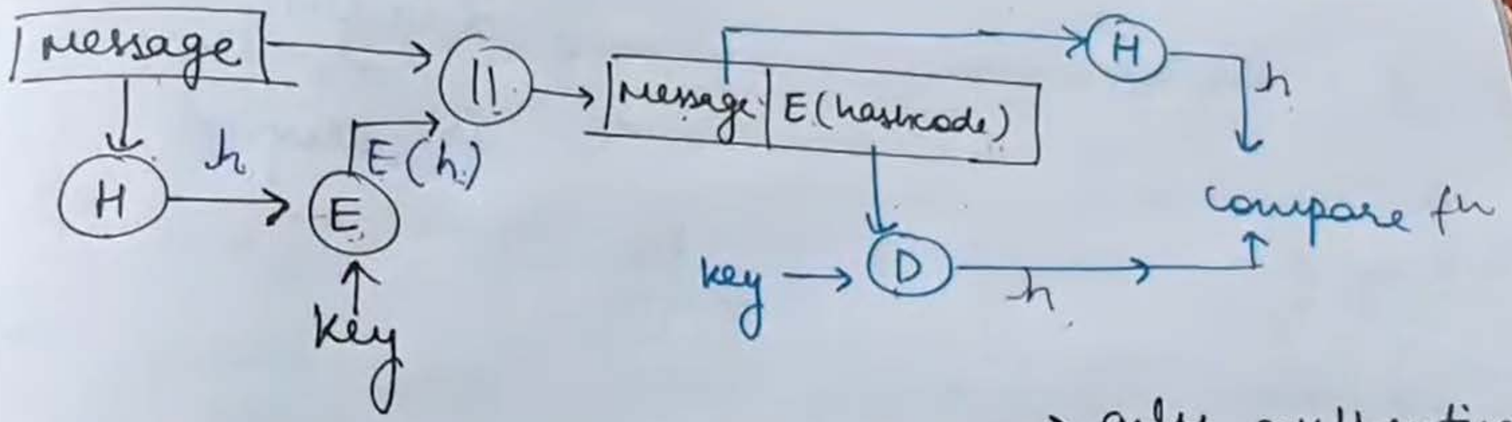


authentication + confidentiality maintained b/c msg was encrypted before sending

if both hashcodes equal in the end

b/c only A & B share the secret key, the msg must have come from A & has not been altered.

method2 –



→ only authentication
→ No confidentiality

only Hash code is encrypted, using symmetric encryption.

If you need only authentication & no confidentiality & your msg is not private messages, the processing time will be less b/c we are not encrypting the message. We are only encrypting the hash code

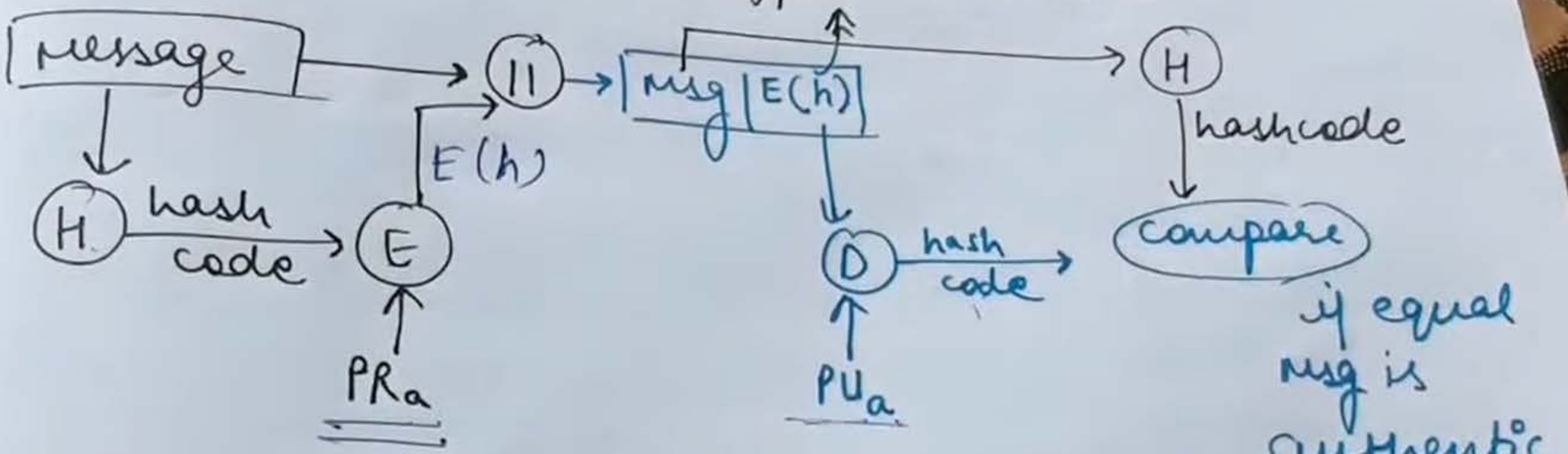
so we can use it b/c

encrypted hashcode

method3

blc are not encrypting the message. we are only encrypting the hash code

method3

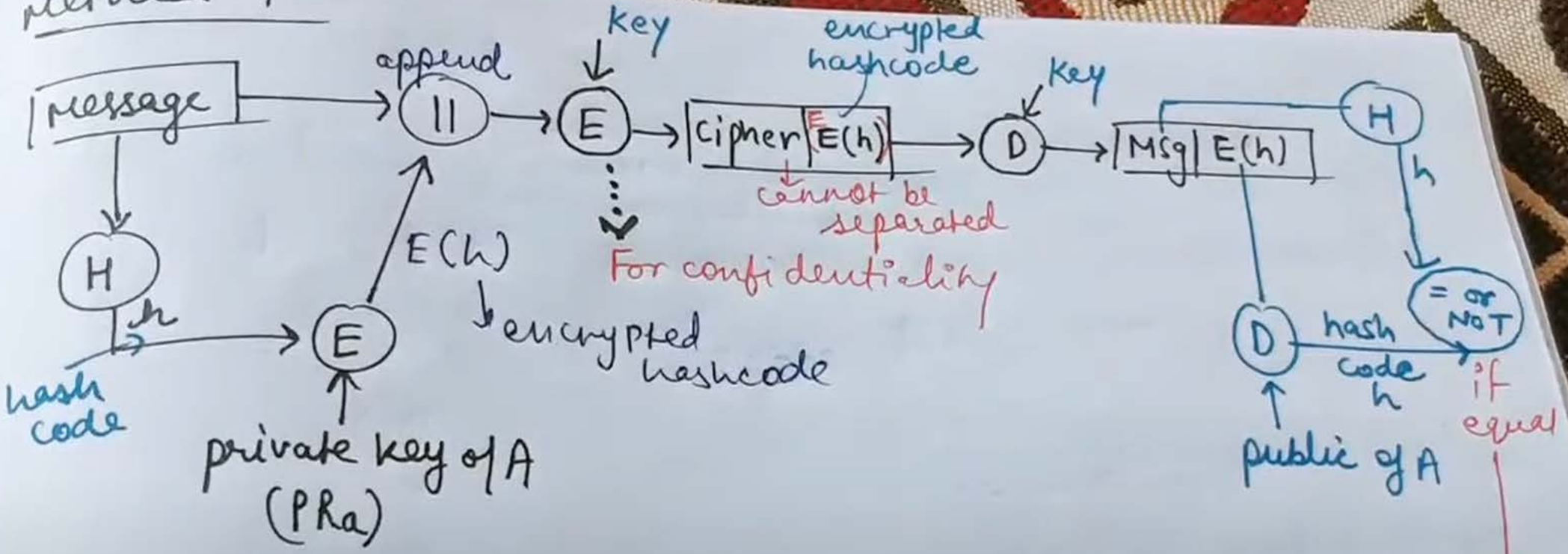


→ no confidentiality
→ only authentication

→ processing time will be less as the msg is not encrypted.

(Same concept as above but using Asymmetric key crypto)

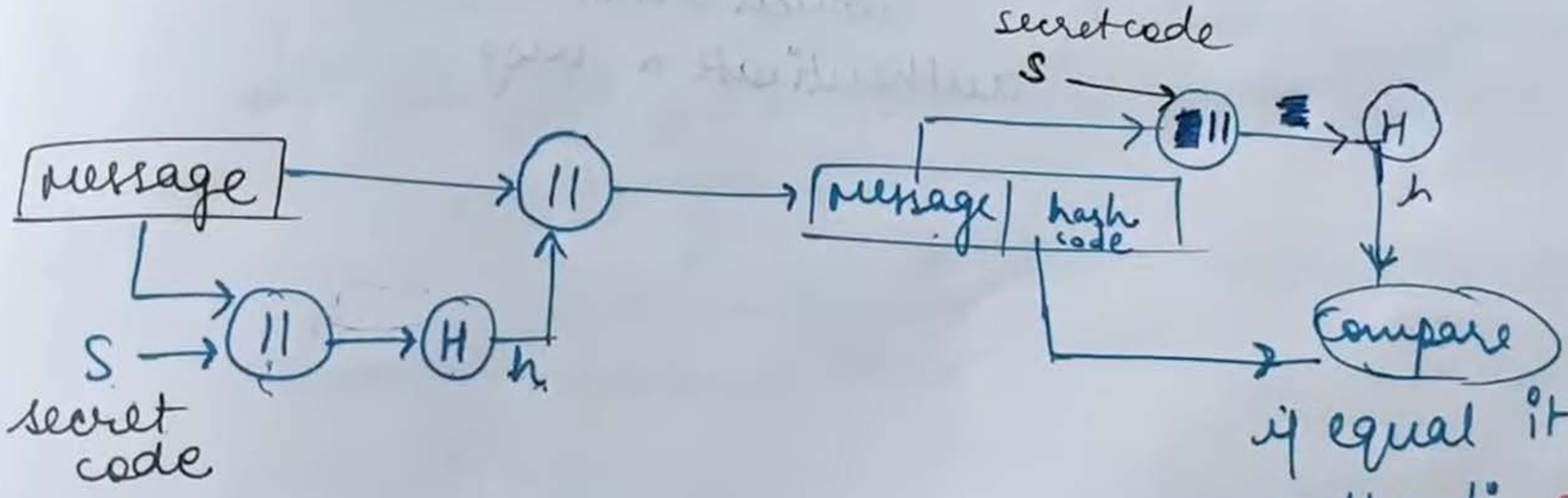
method-4



used when we need confidentiality + authentication
using Symmetric key

using symmetric key authentication

Method - 5 Sender & Receiver will have a secret code 's' and it must be kept secret



if equal it is authentic else not

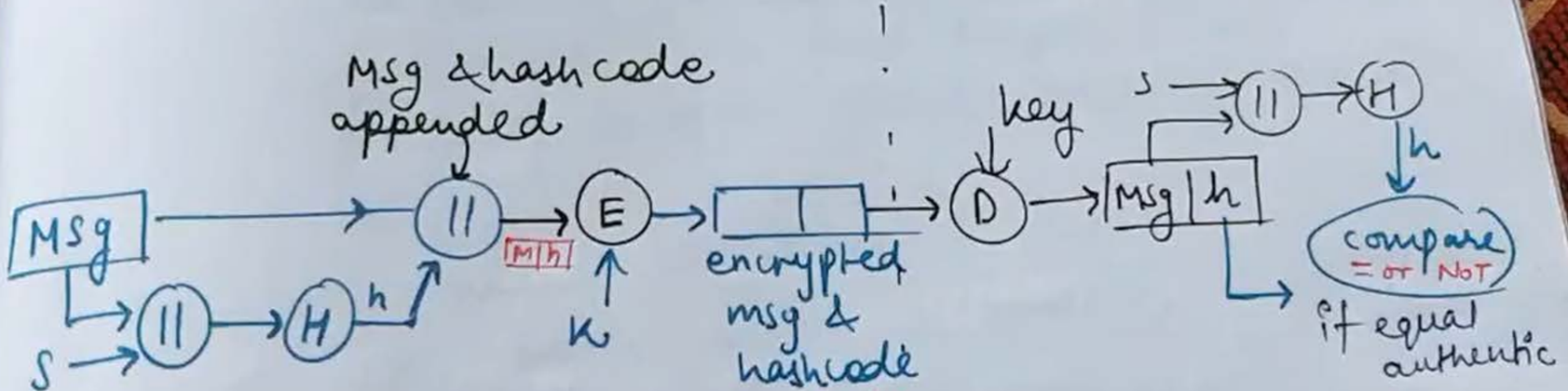
No confidentiality (we get in method 6)
only authentication

Method - 6

confidentiality can be added to the prev. approach by encrypting the entire msg.

Take the msg & append with the secret code 's'. Then apply Hash fn. It gives 'h'. Now append 'h' & msg. Now encrypt using key 'K'. We get encrypted(msg+h)

Now, it will be sent to receiver side



cannot be broken down easily by attacker

receiver side we will use decryption algo &

Added + Updated → 26 Apr 2020

STEGANOGRAPHY

Basic idea → information hide / covered writing.

It is the practice of concealing messages / file / image (i.e, any type of information) within another file, message or image/video.

Note → Later, we will extract it at its destination.

It is derived from Greek words steganos meaning covered or concealed. & graphia which means writing.

IMP

(*) steganography is different from cryptography but, using both together can improve security of the protected data/info. and prevent the detection of the secret communication.

In cryptography, we make the data unreadable (by encryption)

In steganography, we are hiding the existence of data.

STEGANOGRAPHY

hiding the existence of data.

Various forms of steganography are:

1) Text Steganography
2) Audio "
3) Video "
4) Images " (hiding the data in the img. file)

∴ in short steganography can be used to hide/conceal any type of digital content (including text, img, video, audio).

Elliptic Curve Cryptography

ECC

→ It is asymmetric / public key cryptosystem.

→ It provides equal security with smaller key size (eg: as compared to RSA) as compared to non ECC algos.

ie small key size and high security

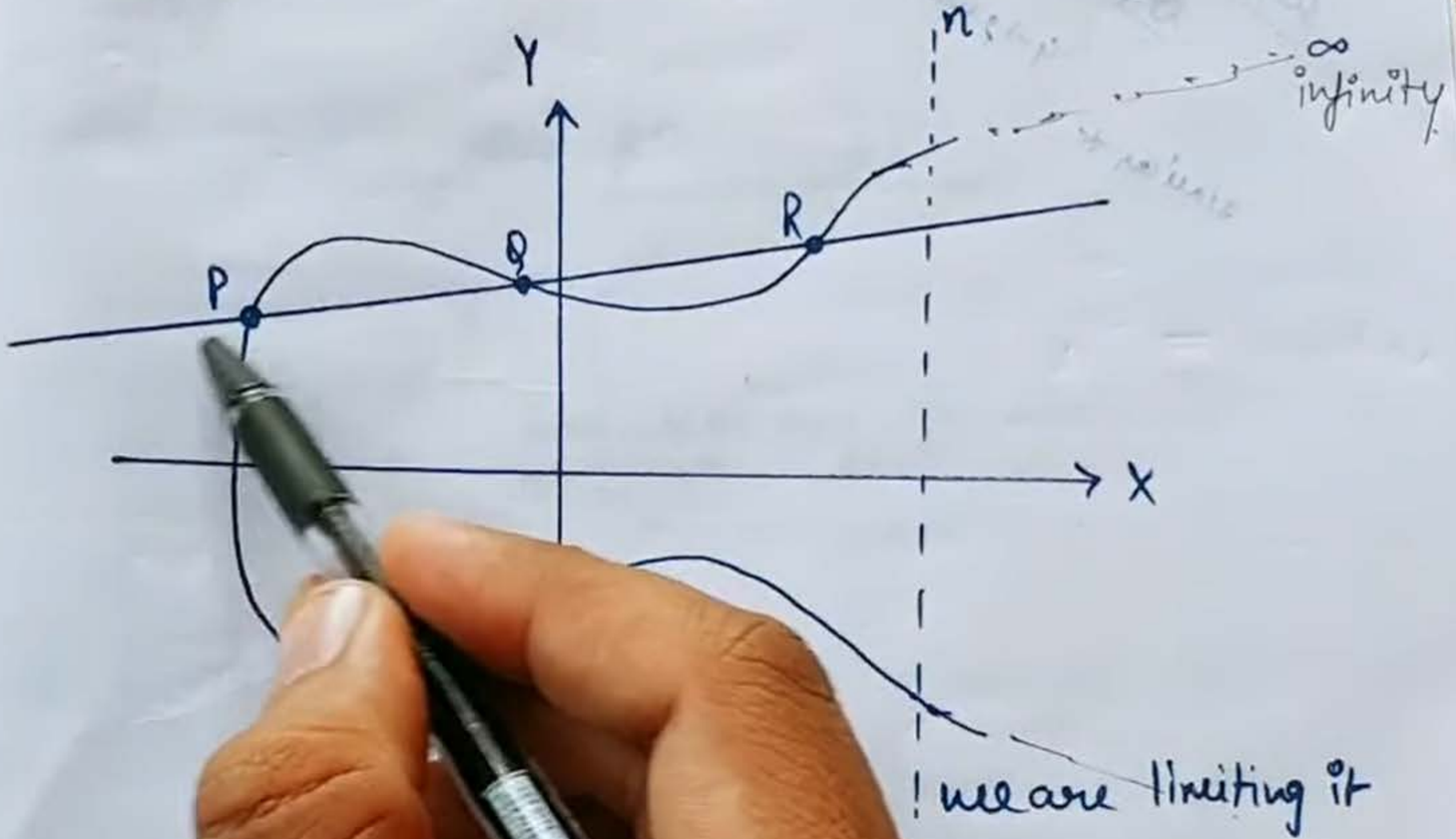
→ It makes use of Elliptic curves.

→ Elliptic curves are defined by some mathematical functions - cubic fns.

eg $y^2 = x^3 + ax + b$ // equation of degree 3

⇒ Elliptic curves are defined by some mathematical functions- cubic fns.

eg $\boxed{y^2 = x^3 + ax + b}$ // equation of degree 3



! we are limiting it

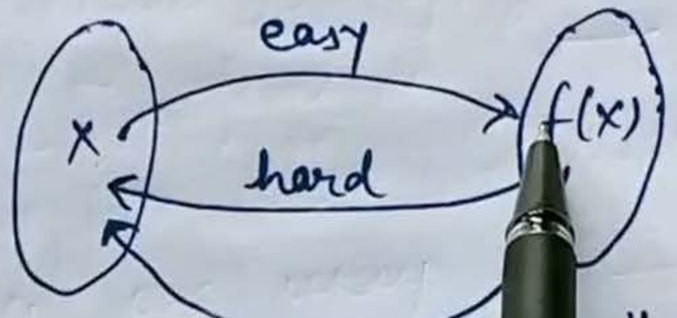
→ symmetric

→ If we d

touch a max of

3 poin

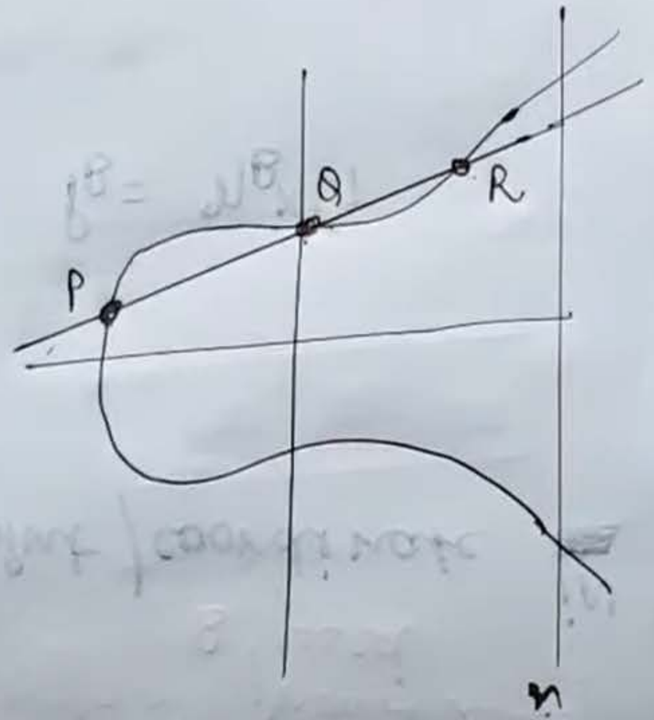
A Trapdoor function is a fn that is easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information, called the trapdoor.



"t" → trapdoor value.

A ⟶ B.

Refer wikipedia for details.



Let $E_p(a,b)$ be the elliptic curve.

Consider the equation $Q = kP$

where $Q, P \rightarrow$ points on curve and $k < n$

(If k and P → given, it should be easy to find Q

but if we know Q and P, it should be extremely difficult to find k.)

This is called the discrete logarithm problem for elliptic curves

ie

It is a one way fn → Trap door fn

ie A → B is easy

but doing B → A

is difficult.

Algo is somewhat similar to Def... the

ECC - ALGORITHM

Ecc - Key Exchange

Global Public Elements

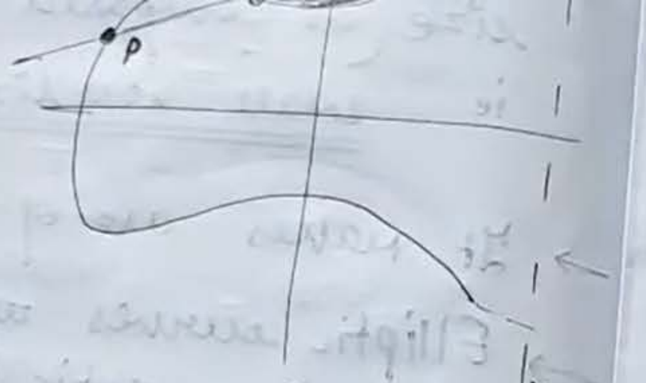
$E_q(a,b)$: elliptic curve with parameters a, b and q

q → prime no. or an integer of the form 2^m.

G : Point on the curve/elliptic curve whose order is large value of n.

User key generation

...vate key n_A $\quad n_A < n$



: elliptic curve with parameters a, b and q

↓

prime no. or an integer of the form 2^m.

G: Point on the curve/elliptic curve whose order is large value of n.

User A key generation

Select private key n_A $\quad n_A < n$

calculate public key P_A $\quad P_A = n_A \times G$

User B key generation

select private k n_B $\quad n_B < n$

calculate public P_B $\quad P_B = n_B \times G$

Calculation of secret

$K = n$

Calculation o

point will be

$C_m = \{ kG, P_m + kP_B \}$ → for encryption public key of B used

This point will be sent to the receiver

DECRYPTION

for decryption, multiply 1st point in the pair with receiver's secret key

ie $kG * n_B$ // for decryption private key of B used

Then subtract it from 2nd point/coordinate in the pair

i.e $P_m + kP_B - (kG * n_B)$

but we know $P_B = n_B \times G$

So $= P_m + kP_B - kP_B$

$= P_m$ (original point).

→ So receiver gets same point

A → B