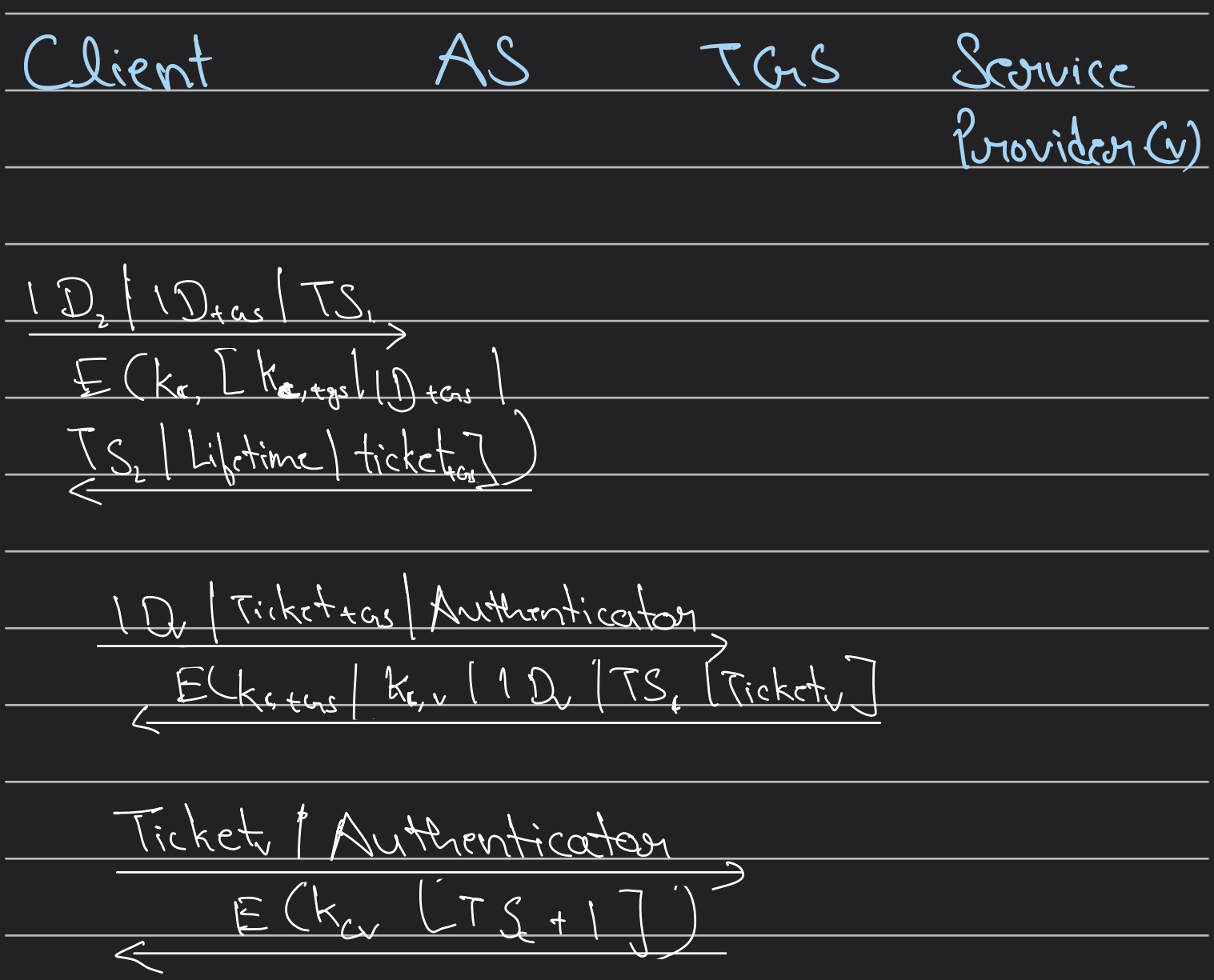

Cryptography
& Network Security
End Sem



→ random bits
→ Server. ω MAC
→ Client. ω MAC
→ Server write key

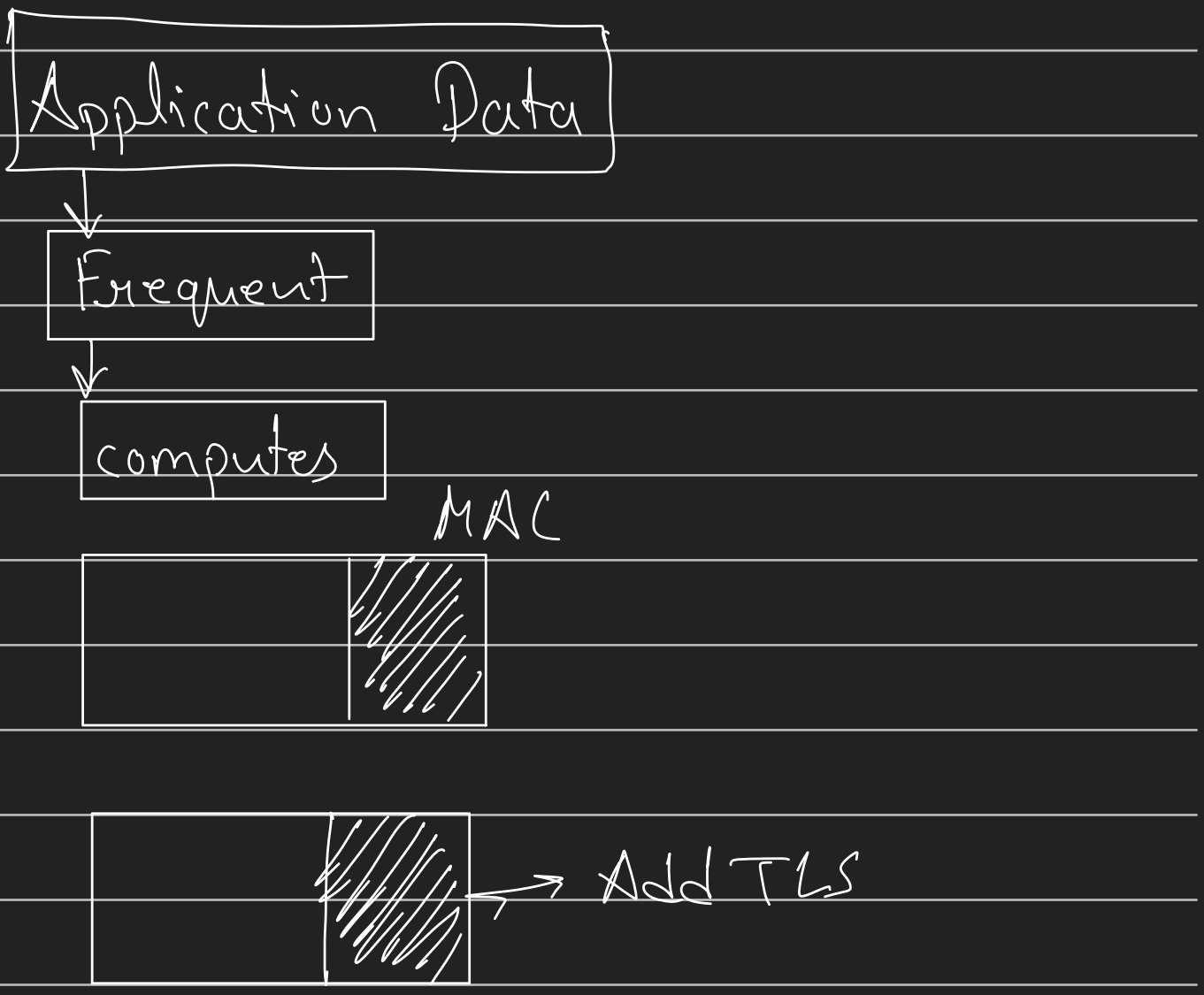
→ connection
→ Session
↳ session id

→ Client. ω. key
→ certificate

→ IV
→ compression

→ Sequence No., $< 2^{64}$

→ Cipher spec (MAC)
→ Master secret
→ Is resumable



TLS Header

Content type (8 bits)
Version (8 bits) 1.2-7
Minor version (8 bits) → 1

Compressed Length (16 bits)

Symmetric Cryptography → for faster encryption

public key " → for digital signature & key exchange

Dual signatures → SET's unique feature.

Unit-3

Number Theory →

Algebraic Structures

If there exists a system such that it consists of a non-empty set and one or more operations on that set then that system is called an algebraic system/structure.

Closure property →

If $a * b \in A$, where a & b are the elements of A.

kisi bhi set be *(sitara wala operation lagane ke baad agar answer set ke hi kisi element me se ho toh woh closure property hai

Associative →

$(a * b) * c = a * (b * c)$

Commutative →

$a * b = b * a$

Prime Nos -

→ Many encryption algorithms are based on prime nos.

→ Very fast to multiply two large prime nos.

→ Extremely computer intensive to do the reverse.

→ Factoring very large no. is very hard i.e. computers take a long time.

Modular Arithmetic →

→ System of arithmetic for integers
→ Wrap around after reaching a certain value called modulus.
→ Central mathematical concept in cryptography.

Properties →

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a+b) \bmod n$

2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a-b) \bmod n$

3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

Properties of Modular Arithmetic

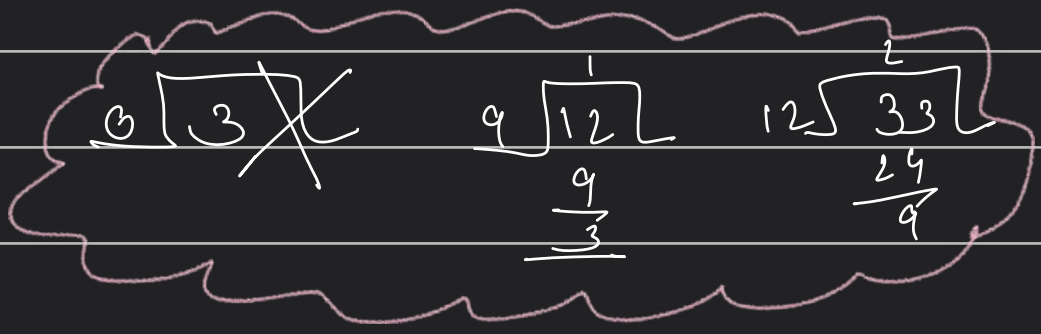
Property	Expression
Commutative Laws	$(a + b) \bmod n = (b + a) \bmod n$ $(a \times b) \bmod n = (b \times a) \bmod n$
Associative Laws	$[(a + b) + c] \bmod n = [a + (b + c)] \bmod n$ $[(a \times b) \times c] \bmod n = [a \times (b \times c)] \bmod n$
Distributive Laws	$[a \times (b + c)] \bmod n = [(a \times b) + (a \times c)] \bmod n$
Identities	$(0 + a) \bmod n = a \bmod n$ $(1 \times a) \bmod n = a \bmod n$
Additive Inverse	For each $a \in Z_n$, there exists a '-a' such that $a + (-a) \equiv 0 \bmod n$

Euclidean Algorithm →

Used to compute GCD

Find GCD of (33, 12)

Q	A	B	R
2	33	12	9
1	12	9	3
3	9	3	0
X	3	0	X



Find the GCD of (4864, 3458)

Q	A	B	R
1	4864	3458	1406
2	3458	1406	646
2	1406	646	114
5	646	114	76
1	114	76	38
2	76	38	0
X	38	0	X

38 is the GCD

Relatively co-prime →
If the GCD is only 1

Euler's Totient Function

Find $\phi(5)$

Numbers less than 5 are 1, 2, 3 & 4

GCD	Relatively Prime?
$GCD(1,5)=1$	1 ✓ (Yes)
$GCD(2,5)=1$	1, 2 ✓ (Yes)
$GCD(3,5)=1$	1, 2, 3 ✓ (Yes)
$GCD(4,5)=1$	1, 2, 3, 4 ✓ (Yes)

$$\boxed{\phi(5)=4}$$

Euler's Totient Function

	Criteria of 'n'	Formula
$\Phi(n)$	'n' is prime.	$\Phi(n) = (n-1)$
	$n = p \times q$. 'p' and 'q' are primes.	$\Phi(n) = (p-1) \times (q-1)$
	$n = a \times b$. Either 'a' or 'b' is composite. Both 'a' and 'b' are composite.	$\Phi(n) = n \times \left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \ldots$ where $p_1, p_2, \ldots$ are distinct primes.

Find $\phi(5)$

Here $n = 5$

'n' is a prime no.

$\phi(n) = (n-1)$

$\phi(5) = (5-1)$

$\phi(5) = 4$

∴ There 4 no. that are lesser than 5 and relatively prime to 5.

Find $\phi(35)$

$n = 35$

It is a product of 2 prime nos.

7×5

$p = 5$ & $q = 7$

$$\phi(n) = (p-1) \times (q-1)$$
$$= (5-1) \times (7-1)$$
$$= 4 \times 6 \Rightarrow \boxed{24}$$

∴ There are 24 nos. lesser than 35 & relatively prime to 35.

Find $\phi(1000)$

$n = 1000 = 2^3 \times 5^3$

(2) & (5)

$$\phi(n) = n \times \left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \cdots$$

$$= 1000 \times \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{5}\right)$$

$$= \overset{200}{\cancel{1000}} \times \left(\frac{1}{2}\right)\left(\frac{4^2}{5}\right)$$

$$= \boxed{400}$$

Q. Find $\phi(7000)$

$n = 7000 = 2^3 \times 5^3 \times 7^1$

$$\phi(n) = n \times \left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\left(1 - \frac{1}{p_3}\right) \cdots$$

$$= 7000 \times \left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{5}\right)\left(1 - \frac{1}{7}\right)$$

$$= \overset{1400\,200}{\cancel{7000}} \times \left(\frac{1}{2}\right)\left(\frac{4^2}{5}\right)\left(\frac{6}{7}\right) = \boxed{2400}$$

Fermat's Little theorem →

If 'p' is a prime no. & 'a' is a +ve integer not divisible by 'p' then

$a^{p-1} \equiv 1 \pmod{p}$

Example 1: Does Fermat's theorem hold true for p=5 and a=2?

$p = 5$ & $a = 2$

$a^{p-1} \equiv 1 \pmod{p}$

$2^{5-1} \equiv 1 \pmod{5}$

$2^4 \equiv 1 \pmod{5}$

$16 \equiv 1 \pmod{5}$

$\therefore$ True

$p = 13$ & $a = 11$

$a^{p-1} \equiv 1 \pmod{p}$

$11^{12} \equiv 1 \pmod{13}$

$-2^{12} \equiv 1 \pmod{13} \Rightarrow -2^{4\times3} \equiv 1 \pmod{13}$

$3^3 \equiv 1 \pmod{13} \Rightarrow 27 \equiv 1 \pmod{13}$ $\therefore$ true

Modular Exponentiation

$23^3 \bmod 30$

$-7^3 \bmod 30$

$\Rightarrow -7^2 \times -7^1 \bmod 30$

$\Rightarrow 49 \times -7 \bmod 30$

$\Rightarrow -133 \bmod 30$

$\Rightarrow -13 \bmod 30$

$\Rightarrow \boxed{17 \bmod 30}$

Q $31^{500} \bmod 30$

$\Rightarrow 1^{500} \bmod 30$

$\Rightarrow \boxed{1 \bmod 30}$

Q $242^{329} \bmod 243$

$\Rightarrow -1^{329} \bmod 243$

$\Rightarrow -1 \bmod 243$

$\Rightarrow 242$

Q. $88^7 \bmod 187$

$88^1 \bmod 187 = 88$

$88^2 \bmod 187 = 88^1 \times 88^1 \bmod 187$

$= 88 \times 88$

$= 7744 \bmod 187 = 77$

$88^4 \bmod 187 = 88^2 \times 88^2 \bmod 187$

$= 77 \times 77$

$= 5929 \bmod 187 = 132$

$88^7 \bmod 187 = 88^4 \times 88^2 \times 88^1 \bmod 187$

$= (132 \times 77 \times 88) \bmod 187$

$= 894432 \bmod 187$

$= 11$

Q. $29^5 \bmod 100$

$29^1 \bmod 100 = 29$

$29^2 \bmod 100 = 29^1 \times 29 \bmod 100$

$= 841 \bmod 100 \Rightarrow 41$

$29^4 \bmod 100 = 29^2 \times 29^2 \bmod 100$

$= 41 \times 41 \bmod 100$

$= 81$

$29^5 \bmod 100 = 29^4 \times 29^1 \bmod 100$

$= (81 \times 29) \bmod 100$

$= \boxed{49}$ Ans.

Euler's Theorem

For every +ve integer 'a' & 'n', which are said to be relatively prime, then

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

Q. Prove that Euler's theorem theorem hold true for $a=3$ & $n=10$

$a=3$, $n=10$

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

$3^{\phi(10)} \equiv 1 \bmod 10$

for $\phi 10$

$n=10 \Rightarrow 2 \times 5$

$p=2$, $q=5$

$\phi 10 = (p-1) \times (q-1)$

$= 1 \times 4$

$= \boxed{4}$

$3^4 \equiv 1 \bmod 10$

$\boxed{81 \equiv 1 \bmod 10}$ ∴ True Euler's Theorem hold true

Q. $a = 2, \quad n = 10$

$$a^{\phi(n)} \equiv 1 \bmod n$$
$$2^{\phi(10)} \equiv 1 \bmod 10$$
$$2^4 \equiv 1 \bmod 10$$
$$16 \equiv 1 \bmod 10$$

∴ Euler's theorem does not hold true.

Q. $a = 10$ & $n = 11$

$$a^{\phi(n)} \equiv 1 \bmod n$$
$$10^{\phi(11)} \equiv 1 \bmod 11$$
$$10^{10} \equiv 1 \bmod 11$$
$$-1^{10} \equiv 1 \bmod 11$$
$$\boxed{1 \equiv 1 \bmod 11}$$ ∴ Yes, true

Multiplicative Inverse →

$$A \times A^{-1} \equiv 1 \bmod n$$

$2x \; ? \equiv 1 \bmod 11$

$2 \times 6 = 1 \bmod 11$

Extended Euclidean Algorithm →

Q MI of 3 mod 5

$T = T_1 - T_2 \times Q$

$T_1 = 0, \; T_2 = 1$

Q	A	B	R	T_1	T_2	T
1	5	3	2	0	1	-1
1	3	2	1	1	-1	2
2	2	1	0	-1	2	-5
X	1	0	X	2	-5	X

Q: MI of 11 mod 13

Q	A	B	R	T_1	T_2	T
1	13	11	2	0	1	-1
5	11	2	1	1	-1	6
2	2	1	0	-1	6	-13
X	1	0	X	6	-13	X

$$11 \times 6 = 1 \bmod 13$$

Primality Testing →

$$a^p - a$$

Check for 5 →

$a^5 - a$

$1^5 - 1 = 0$, $2^5 - 2 = 30$

$3^5 - 3 = 240$, $4^5 - 4 = 1020$

all are div. by 5 so it is prime.

RSA Algorithm →

→ It was invented by 3 MIT cryptographers Ron Rivest, Adi Shamir & Leonard Adleman.
→ It was published in 1977.
→ It is a widely used public-key cryptosystem for secure data transmission.
→ It is an asymmetric cryptographic algorithm. (2) keys i.e. public & private key are used here for encryption & decryption.
→ Many protocols rely on RSA for encryption & digital signature functions.
→ These include secure shell or SSH, OpenPGP, SSL/TLS.
→ used in software programs, such as browsers to establish a secure connection over an insecure network like the internet.
→ RSA signature verification is one of the most commonly performed operations in network connected systems.

RSA is a Block cipher.

1. Key Generation

(i) Select 2 large prime nos. 'p' and 'q'.
(ii) calculate $n = p \times q$
(iii) calculate $\phi(n)$
(iv) choose value of e
$1 < e < \phi(n)$ and $\gcd(\phi(n), e) = 1$
(v) calculate
$d \equiv e^{-1} \bmod \phi(n)$
i.e. $ed \equiv 1 \bmod \phi(n)$

(vi) public key = $\{e, n\}$
(vii) private key = $\{d, n\}$

2. Encryption

$$C = M^e \bmod n$$

3. Decryption

$$M = C^d \bmod n$$

If the value of m is not given then assume it

Q. $p = 3, q = 11$ $\{M = 31\}$

(i) $p = 3, q = 11$

(ii) calculating n

$n = p \times q \Rightarrow \boxed{33}$

(iii) calculating ϕn

$\phi n = (p-1) \times (q-1)$

$= 2 \times 10$

$\boxed{\phi n = 20}$

(iv) let $\boxed{e = 7}$

as $\boxed{1 < 7 < 20}$

$\boxed{\gcd(7, 20) = 1}$

(v) calculating d

$d \equiv e^{-1} \bmod \phi(n)$

$d \times e \equiv 1 \bmod \phi(n)$

$(d \times 7) \bmod 20 = 1$

$\boxed{(3 \times 7) \bmod 20 = 1}$

$\boxed{d = 3}$

EED bhi use kr skte.

(vi) public key = (7, 33) [e, n]

(vii) private key = (3, 33) [d, n]

2. Encryption

$C = M^e \bmod n$

$C = 31^7 \bmod 33$

$C = 31^1 \bmod 33 = -2$

$= 31^2 \bmod 33 = 31^1 \times 31^1 \bmod 33$

$= (-2 \times -2) \bmod 33$

$= 4 \bmod 33$

$= -29$

$= 31^4 \bmod 33 = 31^2 \times 31^2 \bmod 33$

$= (-29) \times (-29) \bmod 33$

$= 841 \bmod 33$

$= 16$

$C = 31^7 \bmod 33 = (31^4 \times 31^2 \times 31^1) \bmod 33$

$= 928 \bmod 33$

$= \boxed{4}$

3. Decryption

$$M = C^d \bmod n$$

$$M = 4^3 \bmod 33$$

$$M = 64 \bmod 33$$

$$\boxed{M = 31}$$

Advantages of RSA →

- Secure data Encryption & Decryption.
- Digital Signatures for authenticity & integrity
- Public key infrastructure & digital certificates.
- User authentication (SSH, smart cards)
- Email security (PGP, S-secure, MIME - Multi Purpose Internet Mail Extension.
- Secure Web Browsing.
- Cloud & Network data protection.

Disadvantages of RSA -

- Very Slow key Generation.
- Slow signing and decryption, which are slightly tricky to implement security.
- key is vulnerable to various attacks if poorly implemented.

Diffie-Hellman key Exchange

→ A mathematical problem, foundation for many cryptographic protocals.

→ one of the greatest inventions in cybersecurity.

→ proposed by whitfield Diffie & Martin Hellman in 1976.

→ It offers a powerful solution to secure key exchange, which has always been challenging thus ensuring confidentiality & integrity in information.

→ based on the principle of modular exponentiation & discrete logarithms.

Algorithm →

(i) consider a prime no. 'q'

(ii) select $\boxed{\alpha}$ which is a primitive root of q and $\boxed{\alpha < q}$

primitive root →

Is 3 is a primitive root of 7?

$3^1 \bmod 7 = 3$

$3^2 \bmod 7 = 2$

$3^3 \bmod 7 = 6$

$3^4 \bmod 7 = 4$

$3^5 \bmod 7 = 5$

$3^6 \bmod 7 = 1$

all values should be distinct & must not repeat

result → {1, 2, 3, 4, 5, 6}

$\boxed{[1, 2, 3 \ldots\ldots\ldots p-1]}$

X → private key of users

Y → public key of users

(iii) assume X_A (private key) and $X_A < q$

calculate $\boxed{Y_A = \alpha^{X_A} \bmod q}$

(iv) assume X_B (private of B)

$X_B < q$

calculate

$$Y_B = \alpha^{X_B} \bmod q$$

(v) Calculate secret key

Both the sender & receiver will use public key.

$$K_1 = (Y_B)^{X_A} \bmod q \qquad K_2 = (Y_A)^{X_B} \bmod q$$

→ public keys ←

known to all

α & q are global public elements

i.e. they are known to everyone

Q. $q = 7$

given $q = 7$
let $\alpha = 5$ {primitive root}

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

(iii) Let $X_A = 3$ { $X_A < q$ }

Calculating

$$Y_A = \alpha^{X_A} \bmod q$$
$$= 5^3 \bmod 7$$
$$= 6$$

{ Key Gen. of person A }

(iv) Key Generation of person B

Let $X_B = 4$

$$Y_B = \alpha^{X_B} \bmod q$$
$$= 5^4 \bmod 7 \Rightarrow 2$$

(v) Calculating Secret key

$K_1 = Y_B^{X_A} \bmod q$

$K_1 = 2^6 \bmod 7$

$= 64 \bmod 7 \Rightarrow \boxed{K_1 = 1}$

$K_2 = Y_A^{X_B} \bmod q$

$= 6^4 \bmod 7$

$= 6^2 \bmod 7 = 36 \bmod 7 = 1$

$= (6^2 \times 6^2) \bmod 7$

$= 1 \times 1 \bmod 7 = 1 \bmod 7 = 1$

$\boxed{K_2 = 1}$

As $K_1 = K_2$

∴ keys are successfully exchanged.

Applications of Diffie-Hellman →

→ SSL/TLS uses DHKE to establish a secure channel b/w client & server. This allows client & server to exchange encrypted messages over an insecure network.

→ SSH uses this to establish a secure channel b/w client & server. Allowing users to securely log in to a remote server & execute commands, and perform other tasks.

→ Many VPN protocols such as IPSec and OpenVPN uses DHKE.

→ SFTP (Secure File Transfer Protocol) also uses DHKE to securely transfer files b/w two systems over an insecure network.

Elliptic Curve Cryptography →

→ Elliptic Curve Cryptography is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller and more efficient cryptographic keys.

→ An alternative to RSA

→ often used for digital signatures in cryptocurrencies such as Bitcoin and Epithereum & one way encryption of emails, data & software.

→ Asymmetric cryptosystem.

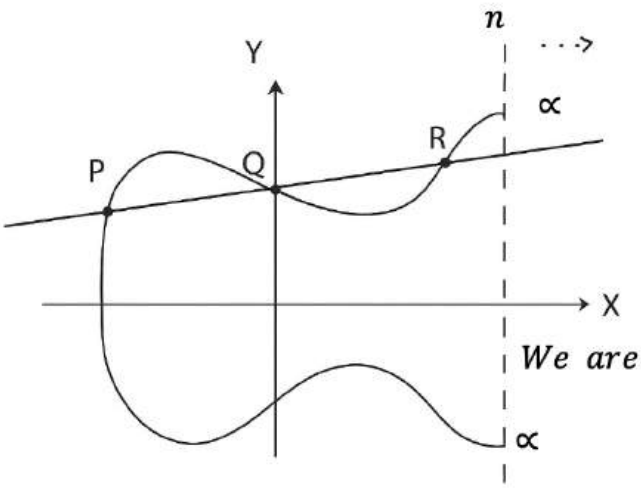
→ provides equal security with smaller key size.

→ Elliptic curves are defined using some mathematical cubic functions.

eg→ $y^2 = x^3 + ax + b$

// equation of degree 3

Algorithm →



We are limiting here by value of n

- symmetric to x-axis
- If we draw a line, it will touch a max of 3 points

Trapdoor Function →

A function that is easy to compute in one direction, but difficult to compute in the opposite direction.

for eg. hum A → B pe toh chale gye lekin B → A pe jaana is tough

potty kr skte but wapas nhi le skte.

easy if given a trapdoor value

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

Consider the equation $\boxed{Q = kP}$

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

If k and P are given → it would be easy to find Q.

but if Q & P are given → it would be very dibficult to find k.

↳ Discrete logarithm problem for elliptic curve.

ECC - Algorithm →

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 A key Generation →

select private key n_A $\quad n_A < n$

calculate public key P_A →

$$P_A = n_A \times G$$

User B key Generation →

select private key n_B $\boxed{n_B < n}$

Calculate public key P_B →

$$\boxed{P_B = n_B \times G}$$

Calculation of secret key by user A →

$$\boxed{K_A = k = n_A \times P_B}$$

Calculation of secret key by user B →

$$\boxed{k = n_B \times P_A}$$

ECC Encryption →

→ Let the message be M.

→ First encode this message M into a point on elliptic curve.

Let this point be $\boxed{P_m}$

↓

Now, this point is encrypted.

For encryption choose a random +ve integer k.

The Cipher point will be →

$$C_m = \{ kG, P_m + kP_B \}$$

This point will be sent to the receiver.

ECC Decryption →

For decryption, multiply 1st point in the pair with receiver's secret key. i.e.,

$$kG \times n_B$$

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

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

we know $P_B = n_B \times G$

So,

$= P_m + kP_B - kP_B$

$= \boxed{P_m}$ (original point)

Parameters	ECC	RSA
Working algorithm	ECC is a cryptography technique that works just on a mathematical model of elliptic curves.	RSA cryptography algorithm is primarily based on the prime factorization approach.
Bandwidth savings	ECC gives significant bandwidth savings over RSA.	RSA provides much lesser bandwidth saving than ECC.
Encryption process	The encryption process takes less time in ECC.	The encryption process takes more time in RSA.
Decryption process	The decryption process takes more time.	Decryption is faster than ECC.
Security	ECC is much safer than RSA and is currently in the process of adapting.	RSA is heading toward the end of its tenure.

Security(in Bits)	RSA key length required	ECC key length required
80	1024	160-223
112	2048	224-255
128	3072	256-383
192	7680	384-511
256	15360	512+

Unit - 4

Hash Functions →

→ Mathematical algorithms that transform input data into a fixed-length sequence of characters, refered to as hash value.

→ Hash functions are intended to be fast, deterministic and one-way, meaning that even a minor change in input yields a very different hash.

→ Takes in variable size message and produce a fixed output called Hash code / hash value / message digest.

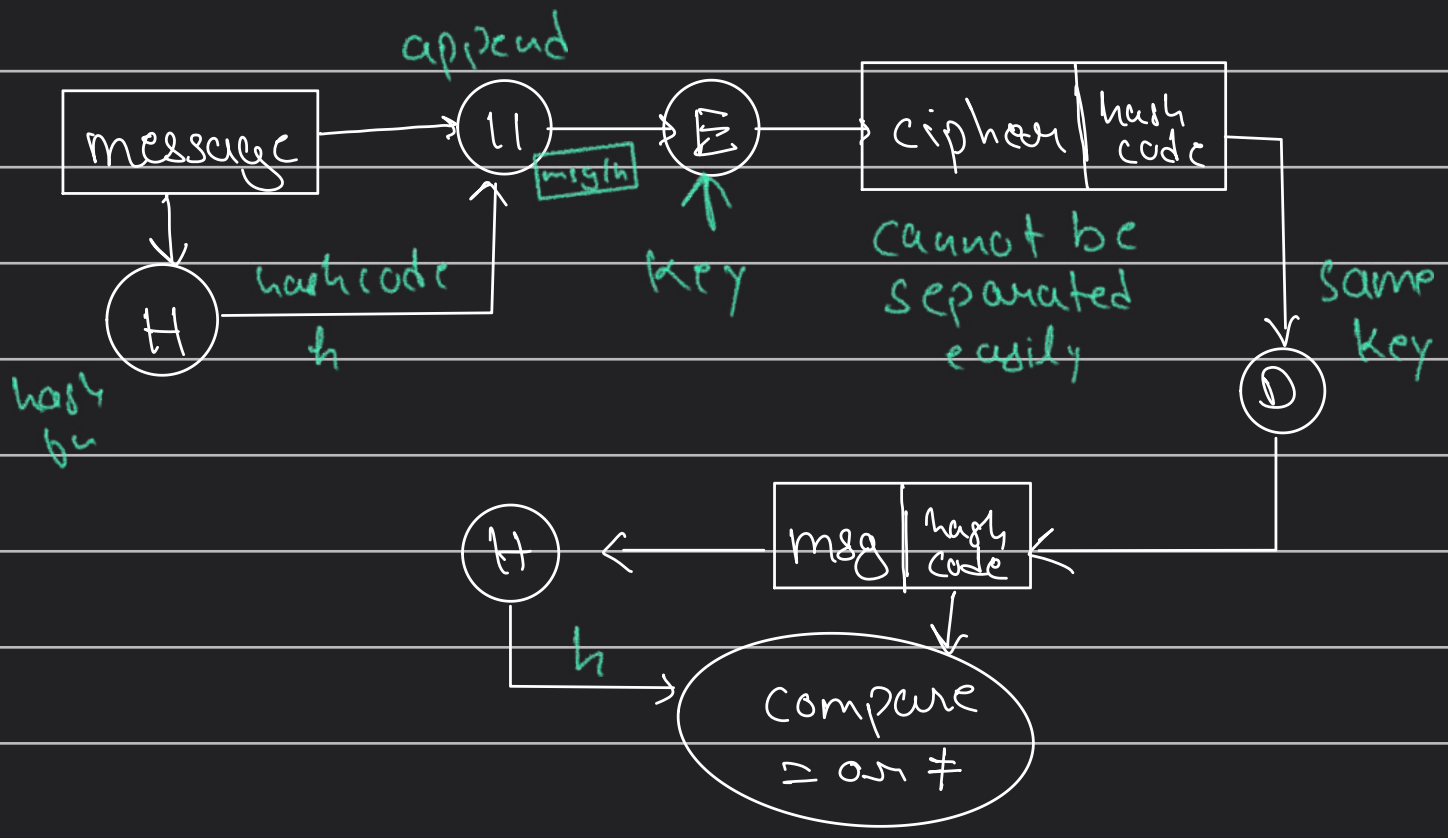
$$H(M) = \text{fixed length code } 'h'$$

M ← variable length msg

h ← hash code

There are diff. methods to provide authentication in diff. situations.

(i)



Properties of Cryptographic Hash Fns →

→ Deterministic - same input always generates the same exact hash function.

→ fast computation

→ Pre-image Resistance - It is computationally infeasible to reverse-engineer the original data from its hash value.

→ Collision Resistance - no two inputs can produce identical hash values.

→ Avalanche effect - Even a tiny change in the input can cause an unpredictible change in hash output.

Application →

- Digital Signatures
- password storage
- Blockchain
- Aws S3 bucket
- hashing passwords.

DE-5 →

1. Append padding bits

$512 \times n - 64$

↓

multiple

2. Append length bits

$512 \times n - 64 + 64$

3. Initializing MD buffer

32 bits	A =	0	1	2	3	4	5	6	7
32 bits	B =	8	9	a	b	c	d	e	f
32 bits	C =	f	e	d	c	b	a	9	8
32 bits	D =	7	6	5	4	3	2	1	0

$F(B, C, D) = (B \wedge C) \vee (\neg B \wedge D)$

$G(B, C, D) = (B \wedge D) \vee (C \wedge \neg D)$

$H(B, C, D) = B \oplus C \oplus D$

$I(B, C, D) = C \oplus (B \vee \neg D)$

4.) Process each 512 bit block one after another & output msg digest.

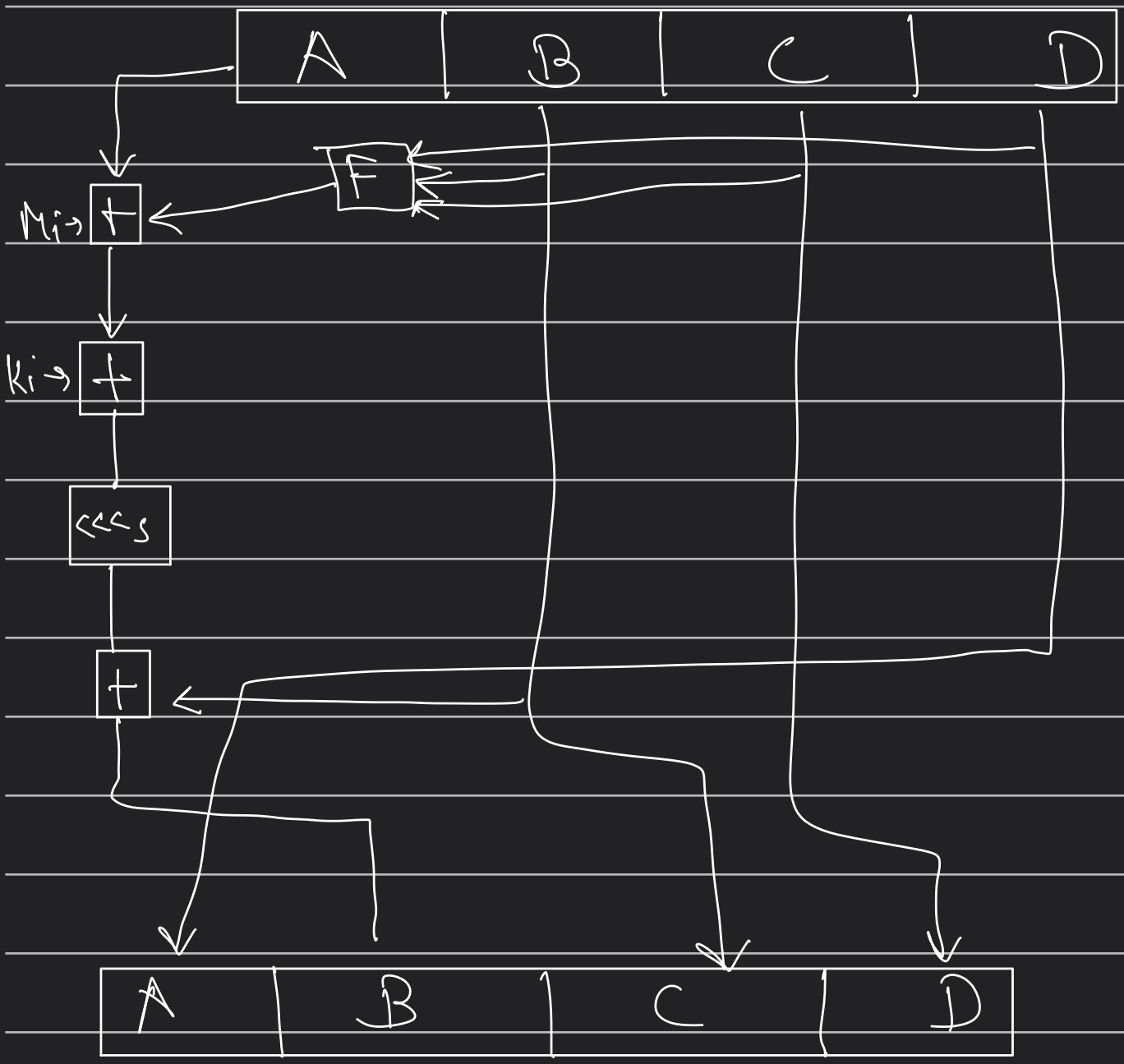
64 { 16 - F, 16 - G, 16 - H, 16 - I } 4 rounds, 16 operations.

$\boxplus$ - add mod 2^{32}

M_i - 32 bit msg

k_i - 32 bit const.

<<< s - left shift by 's'



end of 'I', O/p msg digest => 128 bit

Disadvantages →

- Generates the same hash function for diff. inputs.
- provides poor security over SHA-1
- SHA256 is used instead of MD-5 as it is considered insecure.
- neither symmetric nor assymetric algorithm.

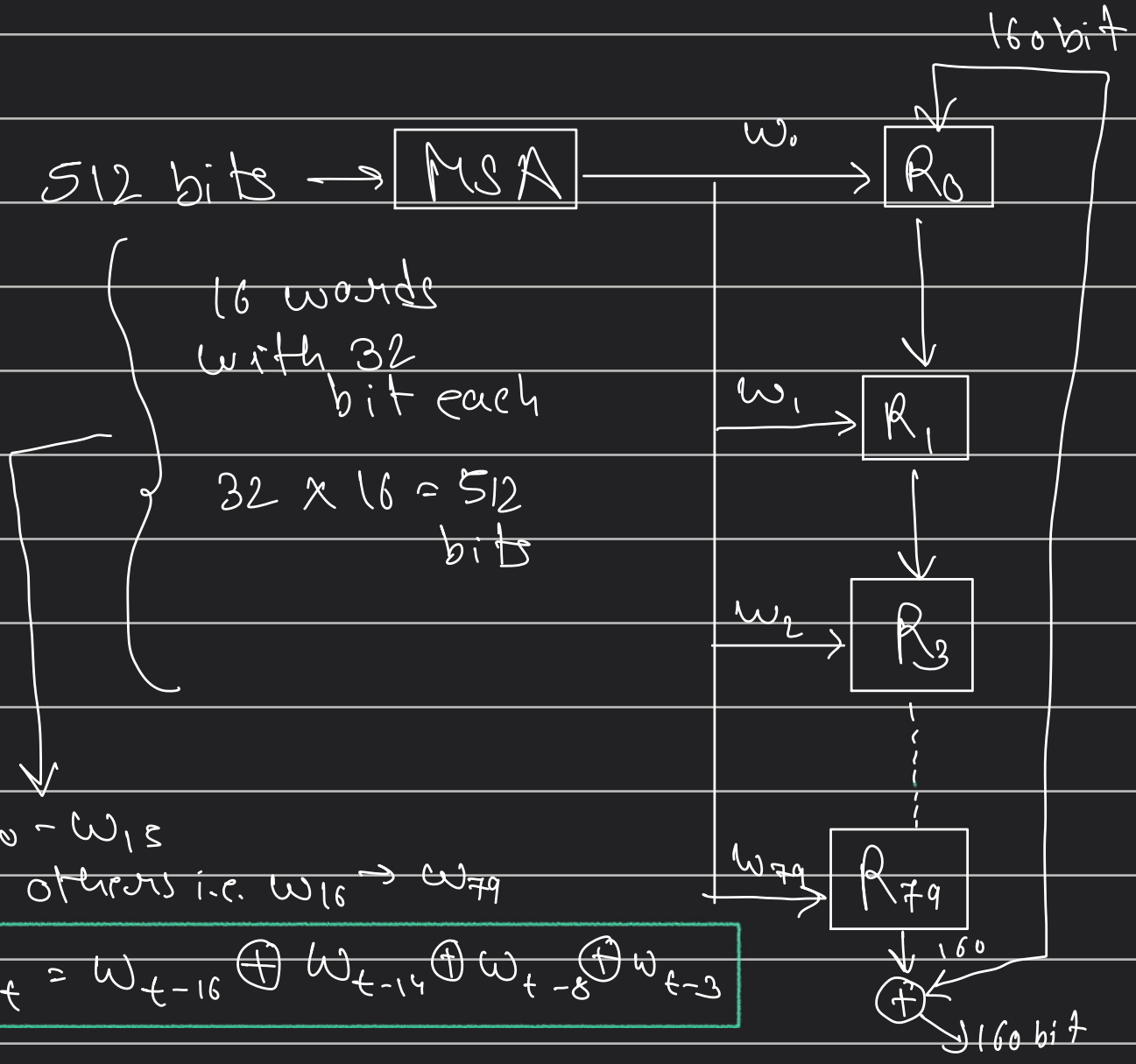
SHA-1

① message M divided into multiple of 512 bits = $M(m_1, m_2, m_3 \ldots\ldots m_n)$

② i.e. 448 + 64 bits = 512 bits

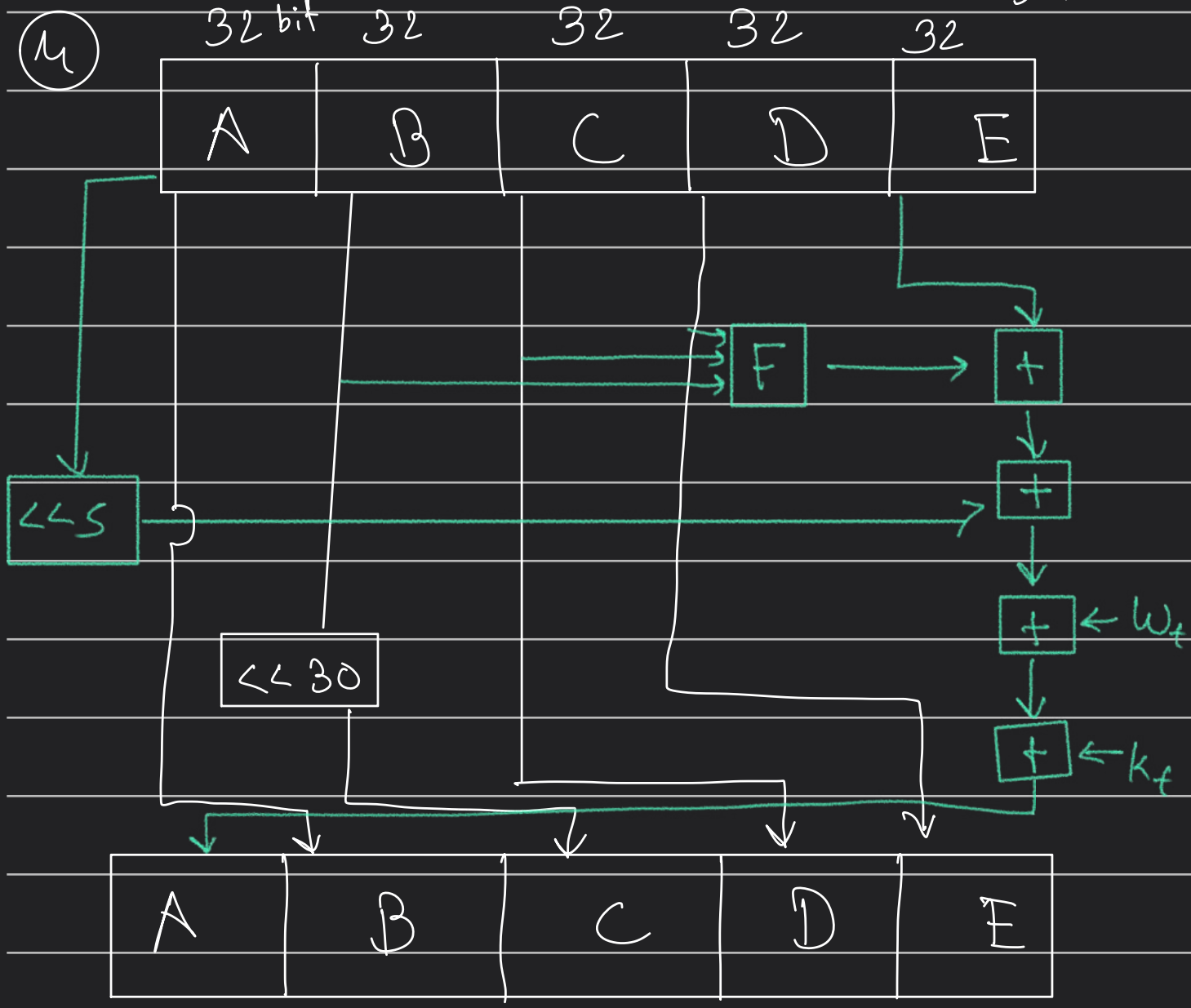
- 448 → msg padding
- 64 bits → append length

③



32 x 5 = 160

④



⑤ Initial A, B, C, D, E values

A = 0 1 2 3 4 5 6 7

B = 8 9 A B C D E F

C = F E D C B A 9 8

D = 7 6 5 4 3 2 1 0

E = C 3 D 2 E 1 F A

K_{1t} = 5A827999 — $B \cdot C + \bar{B} D$

K_{2t} = 6ED9EBA1 — $B \oplus C \oplus D$

K_{3t} = 8F1BBCDC — $BC + BD + CD$

K_{4t} = CA62C1D6 — $B \oplus (C \oplus D)$

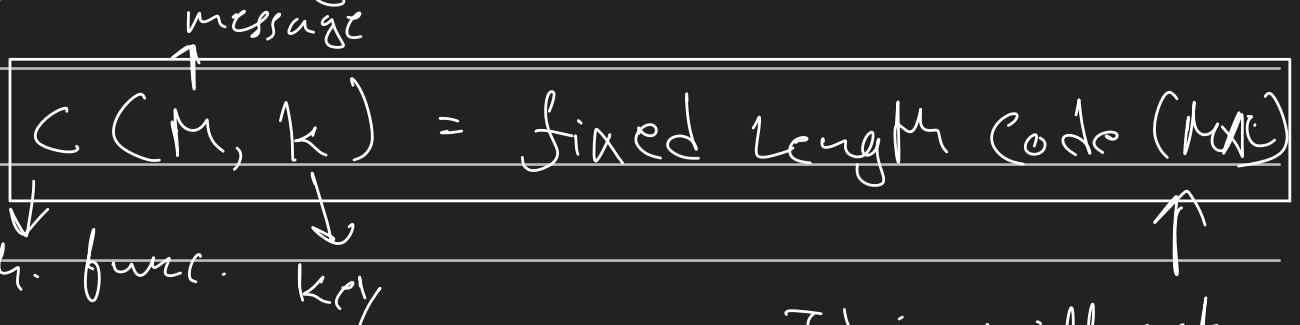
Authentication Functions →

Message Encryption

→ ciphen text act as authentication.

MAC (Message Authentication code)

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



Hash Functions →

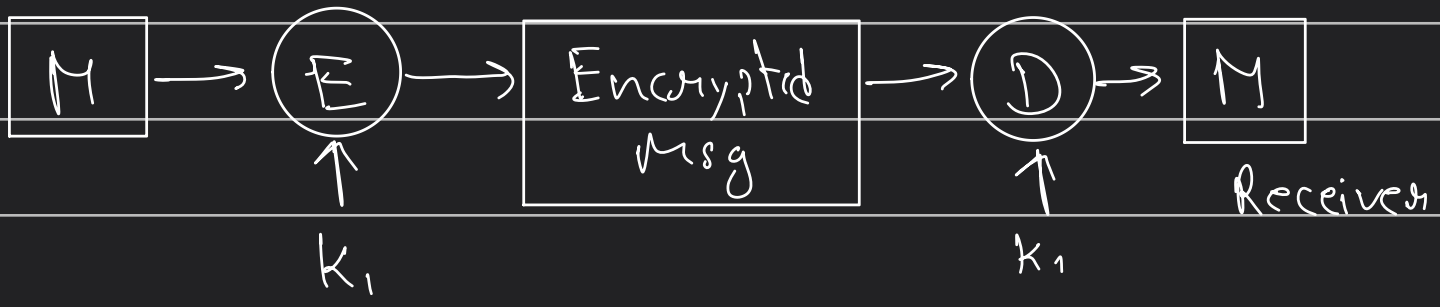
$H(M)$ = fixed Length code (Hash code 'h')

↳ msg

↓

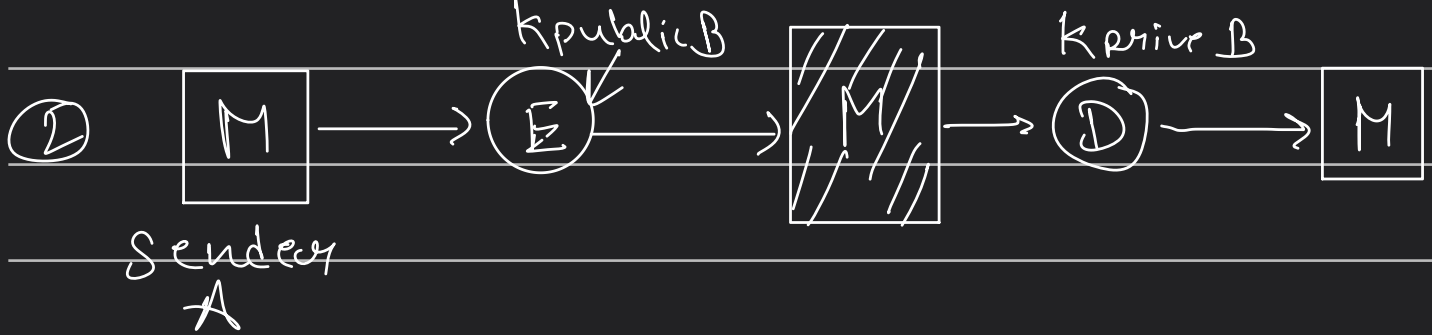
independent key.

Message Encryption



→ key k, shared only b/w sender & Receiver only.

For asymmetric encryption



HMAC →

- A cryptographic technique that ensures data integrity & authenticity using hash function & secret key.

Working →

K - shared key b/w sender & Receiver

↓

K^+ - pad 0's on left side & K until length becomes 'b' bits.

→ where P.T block size is = b bits

ipad = 00110110 .... upto b/8 bits.

opad = 01011100 .... upto b/8 bits.

Calculating S bits →

$K^+ \oplus$ ipad

b bits ← ↓ → b bits

S_i (b bits)

↓ ←b→←b→←b→←b→

S_i	Y_0	Y_1	.....	Y_{L-1}

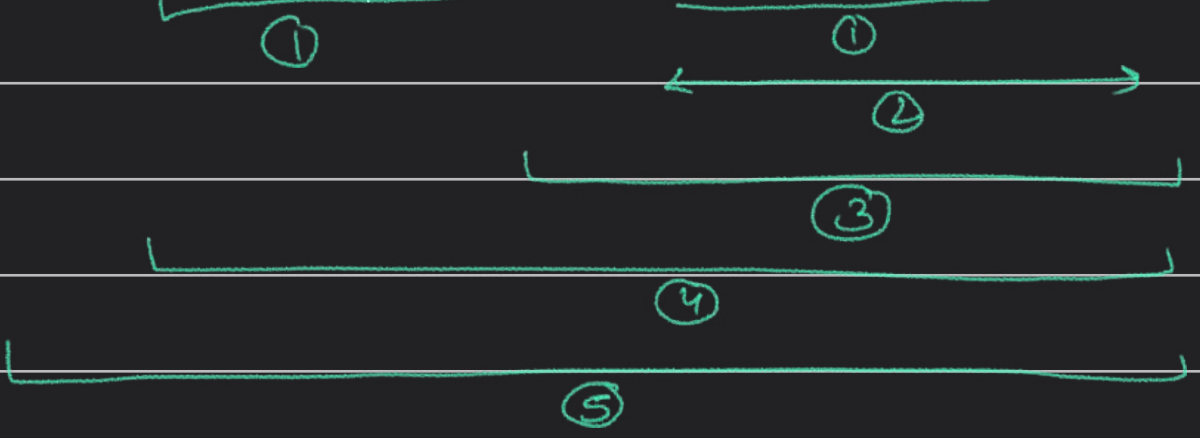
L → No. of P.T. blocks

Y_0 → plain text

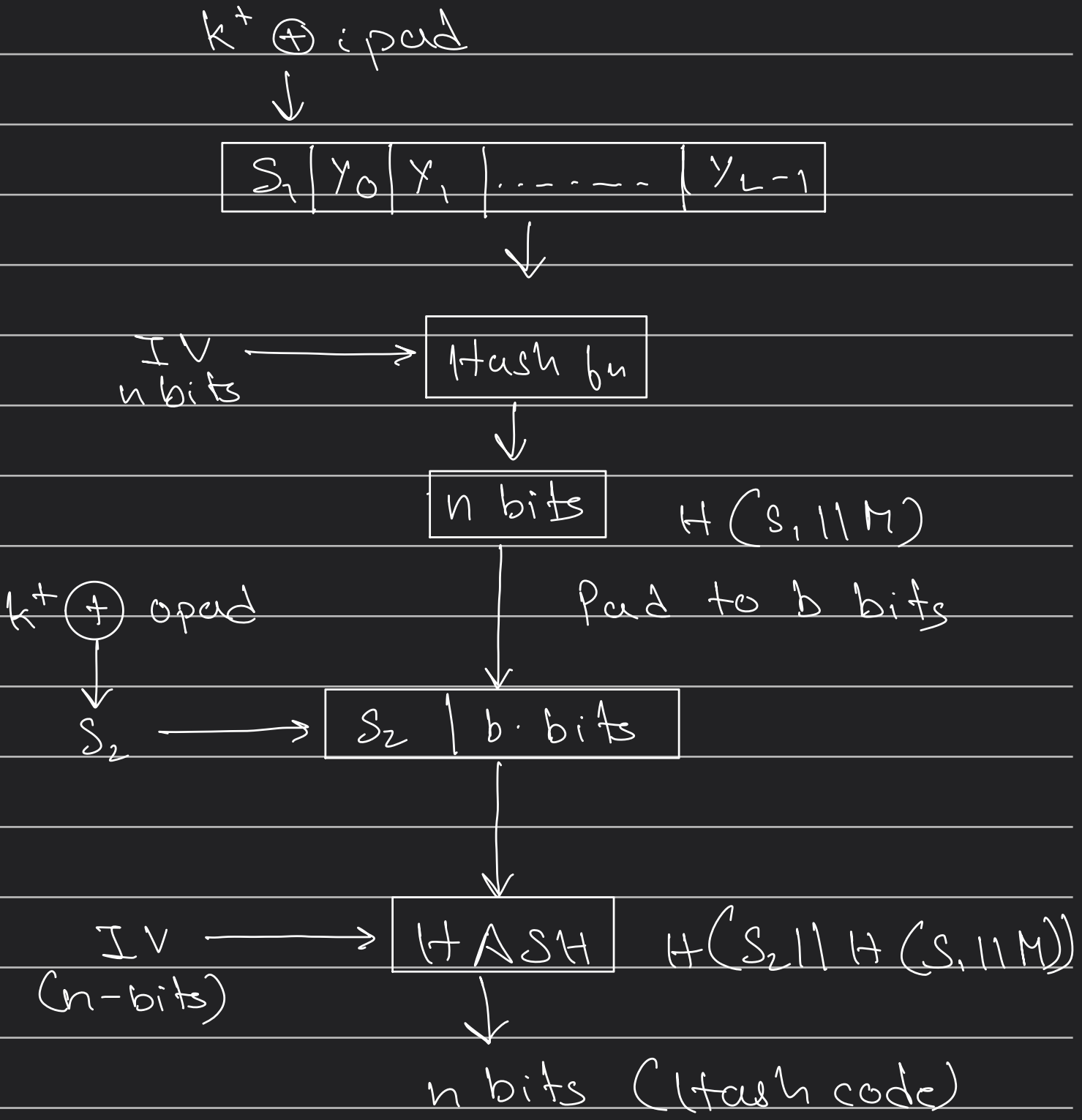
b → size of P.T. blocks

HMAC(k, M) =

$H((k' \oplus opad) \| H((k' \oplus ipad) \| M))$



Block Diagram →



K → secret key b/w S & R
M → MSG to authenticator
H → Hash fn (SHA-256)
K' → key to resize to blocksize
|| → concatenation
ipad → Inner Padding
= 0 x 36 (Repeated to Blocksize)

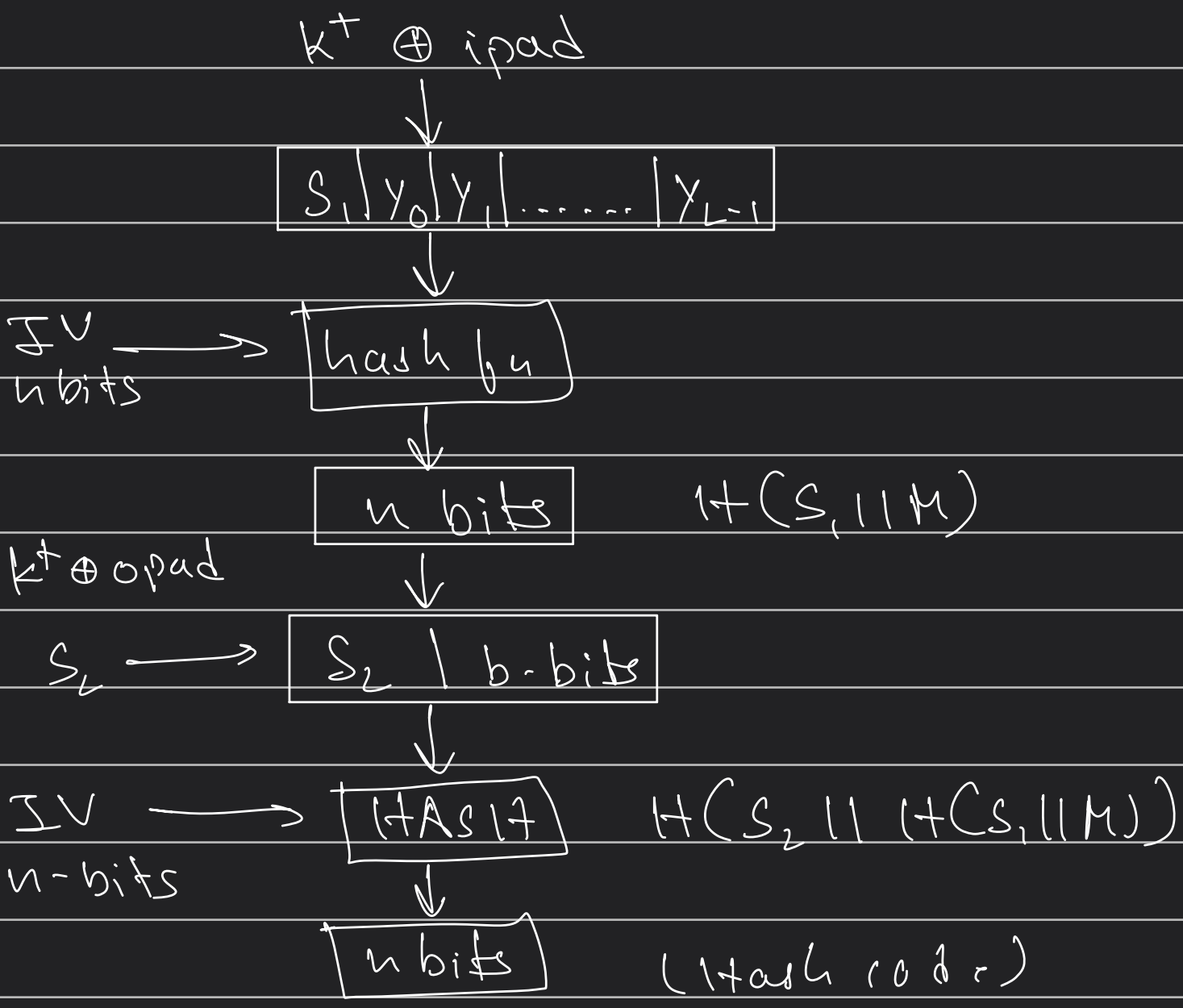
Opad → outer padding
= 0x5C (Repeated blocksize)

Advantages →

→ Ideal for high-performance systems like routers due to the use of hash functions.
→ Digital signatures are larger than HMACs, yet HMAC's provide higher security.
→ HMAC's are used in administrations where public key systems are prohibited

Applications →

→ Verification of email address
→ Authentication of form data sent to client browser.
→ in password reset service
→ used in IoT due to less cost.



X.509 Authentication Service →

→ Digital certificate accepted internationally
→ Does not generate any key but provides a way to access public keys.
→ has 3 versions

---

① Versions - 1, 2, 3
② Serial no. - S.No of certificator
③ Signature algo identifier
algo used by issuer
RSA etc.
④ Issuer name - Org. name
⑤ Validity name
from which date to end date
⑥ Subject name
to whom certificate is given
⑦ Public key info

V1

⑧ Issue unique id
⑨ Subject unique id

V_2

⑩ Extensions

V_3

① Version - 1, 2, 3
② Serial no. - S.no. of certification
③ signature algo identifier.
algo used by issuer
RSA
④ Name of issuer - org" name
⑤ Validity name
start date - end date
⑥ Subject name
to whom is cert. given
⑦ Public key info

⑧ Issue unique ID
⑨ Subject unique ID

⑩ optional

Kerberos →

- centralized authentication server whose fn. is to authenticate user to client & vice-versa

Main components →

1) Authentication server → performs initial authentication & ticket for TGS
2) Database → authentication server verifies the access rights of users in the database.
3) TGS → issues ticket for the server.

working →

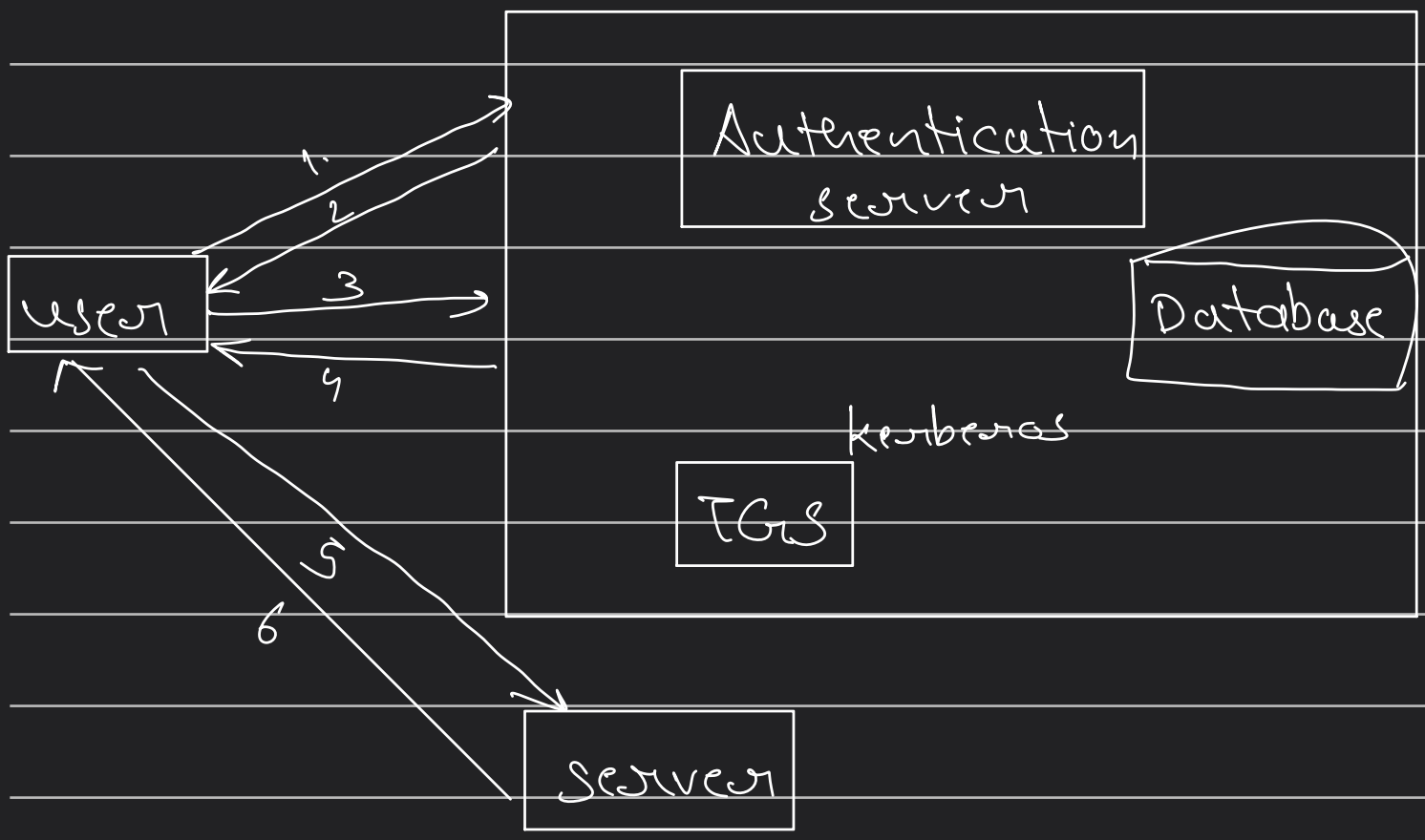
1) User login & request service on the host user requests for TGS.
2) authentication server verifies access rights of user in the database & then gives ticket & session key.
3) decryption of msg is done using password then send the ticket to TGS.

Ticket contains authenticators like user name.

4) TGS decrypts the ticket send by the user & authenticator verifies the req. then creates a ticket for requesting services

5) user sends ticket & authenticator to server.

6) server verifies ticket & authenticator then generates access to service. After this user can access services.



Limitations →

- Each network service must be modified individually.
- does not work in timeshare environment.
- all pass encrypted with single key.

Applications →

1. User Authentication
2. SSO (Single Sign On)
3. Mutual Authentication
4. Authorization
5. Network security.

DES →

Input → 64 bits

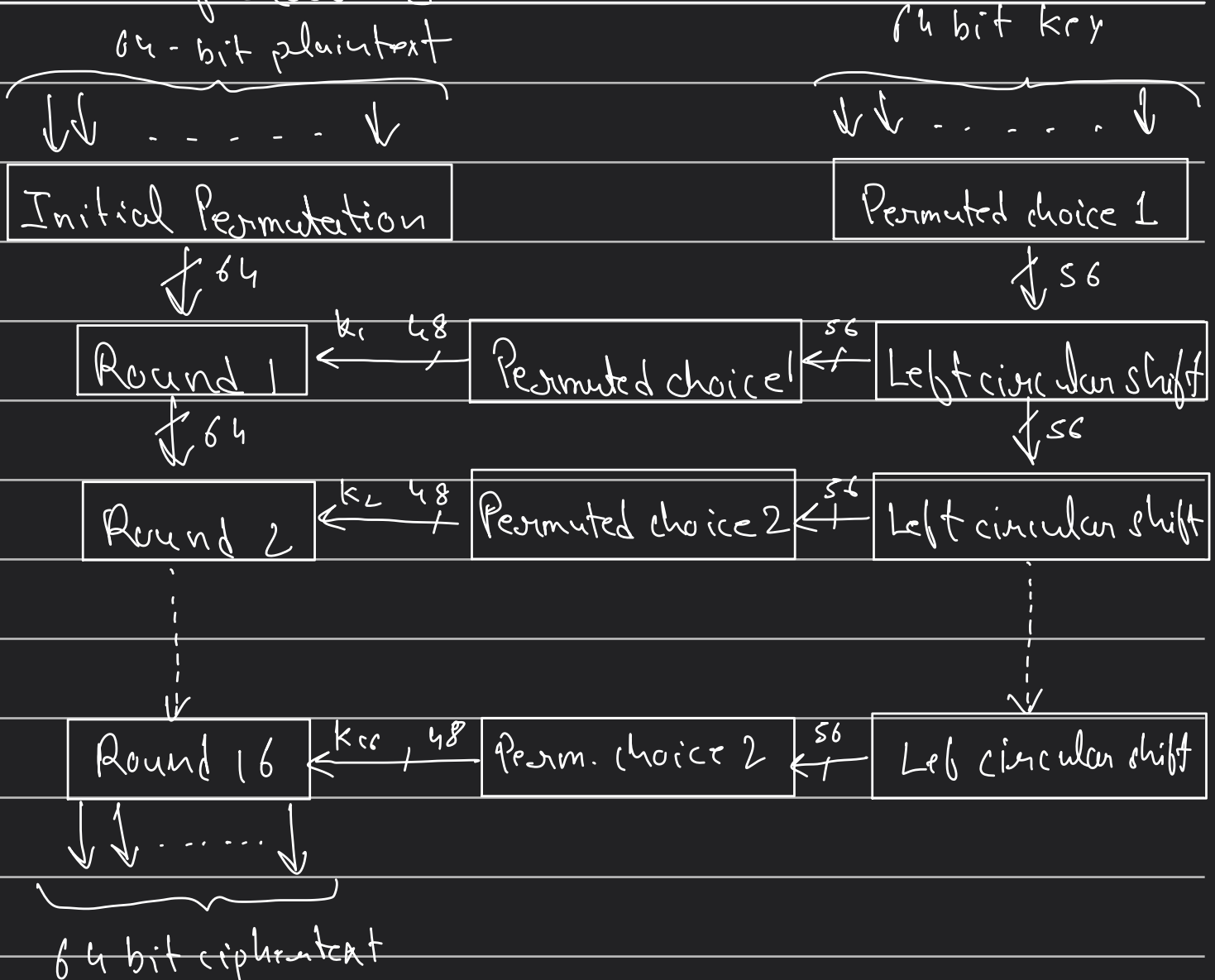
Output → 64 bits

Main key → 64 bits

Sub key → 56 bits

Round key → 48 bits

No. of rounds → 16



Round - 1

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

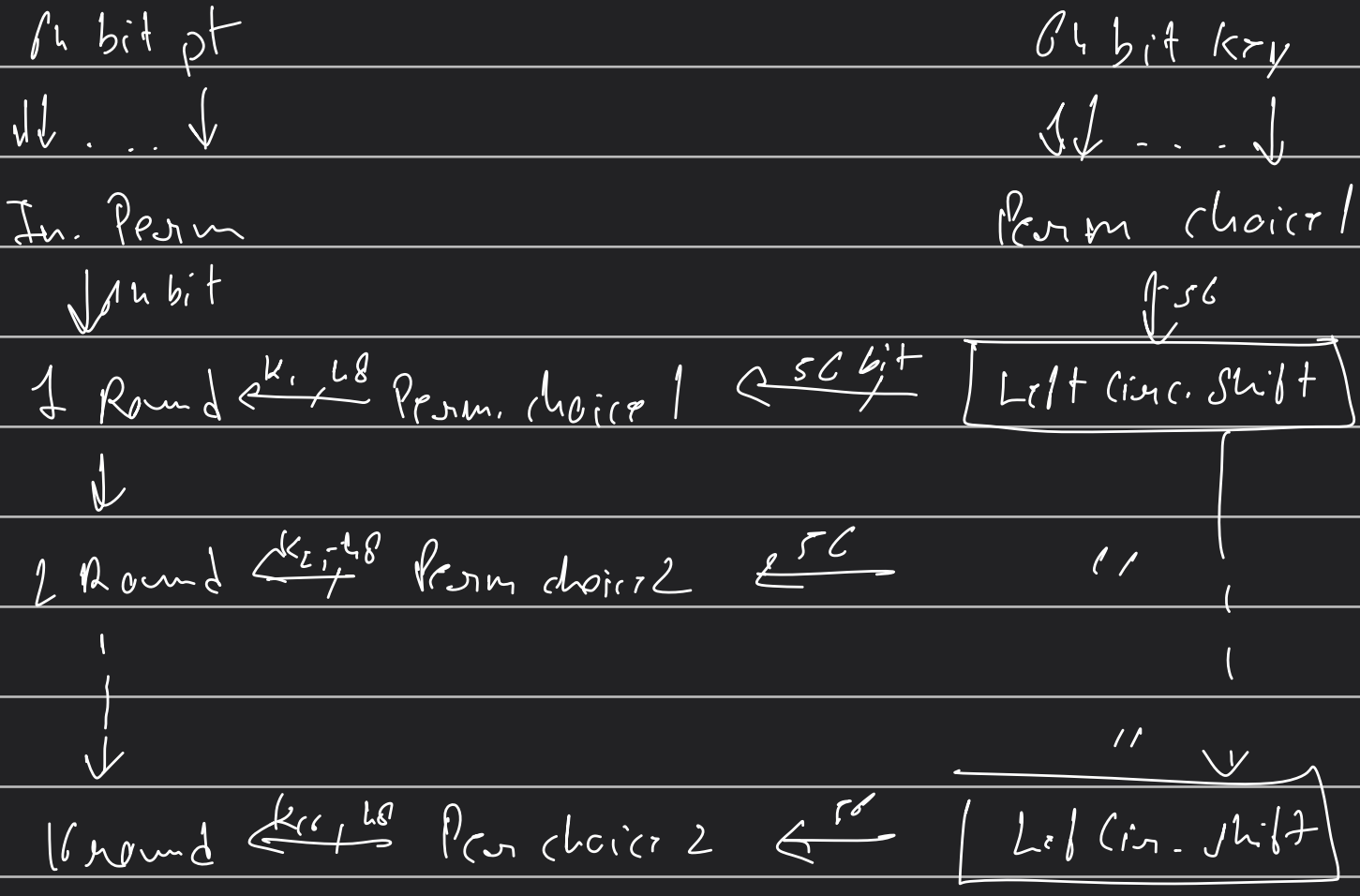
DES

(i) block cipher

(ii) symmetric cipher (same key for encr & decr.)

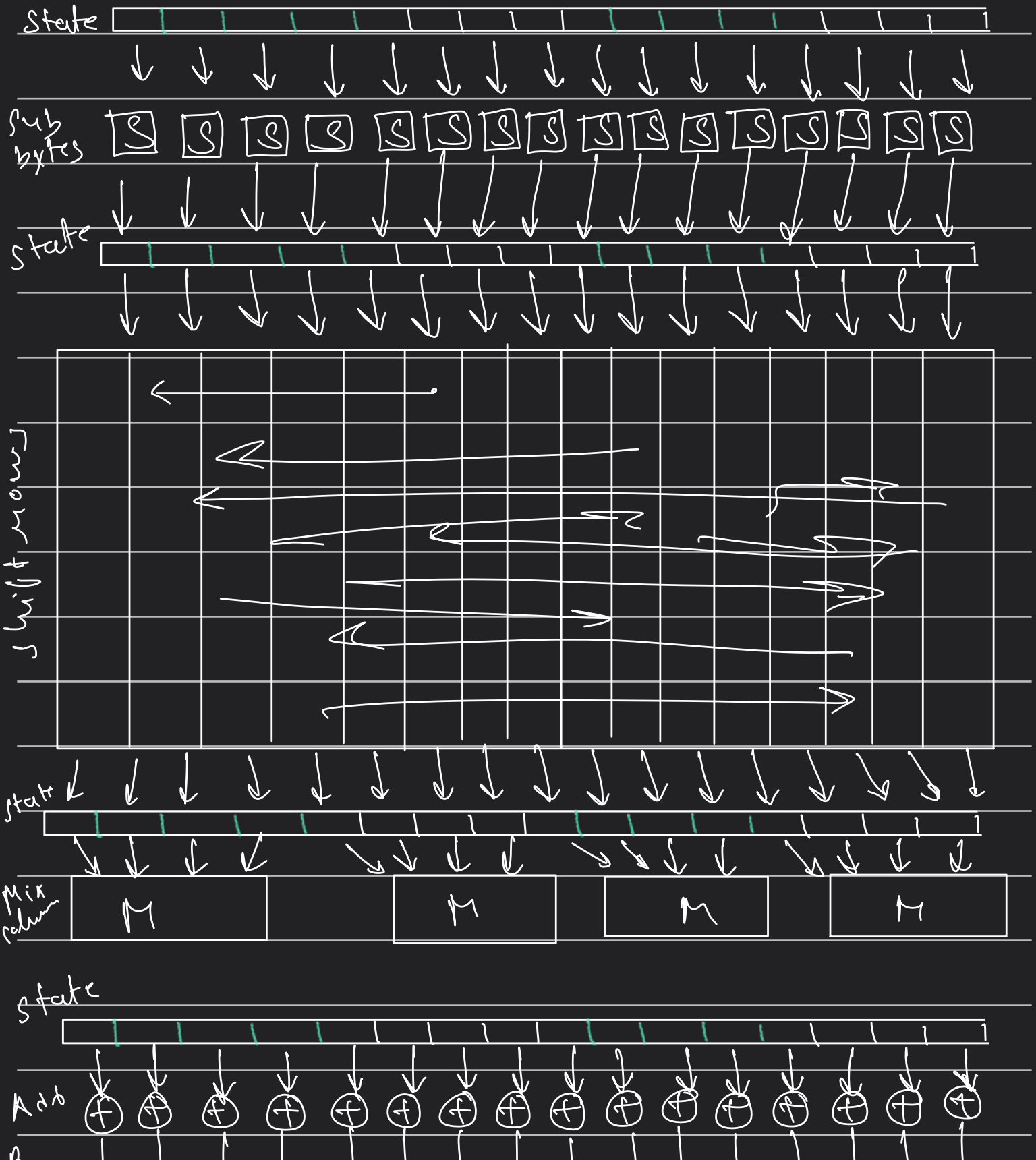
(iii) 64 bit plain text block.

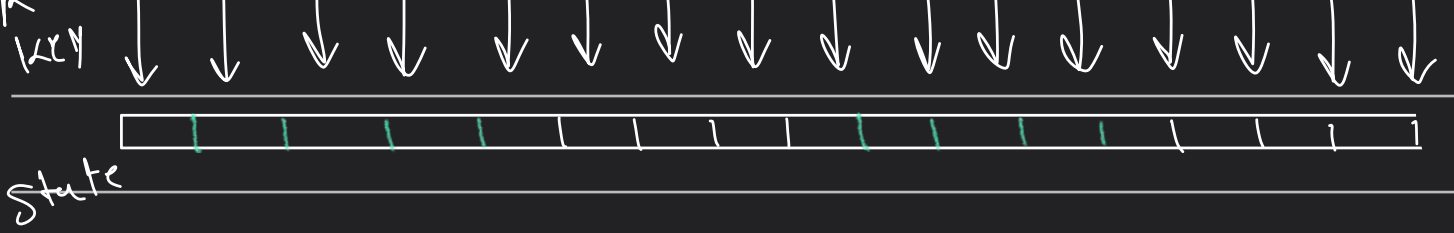
(iv) 16 rounds, each round is a feistal round



AES →

- Symmetric key Block Cipher
- fixed block size 16 bytes → 128 bits





State → Sub bytes → State → Shift rows → State

↓

State ← Add Round key ← State ← Mix columns

RSA →

$p = 13, q = 7$

$n = p \times q$

$= 13 \times 7 \Rightarrow 91$ $\boxed{n = 91}$

calculating ϕn →

$\phi n = 12 \times 6$

$= 72$

$e = 7$

$d = e^{-1} \bmod \phi(n)$

$d \times e = 1 \bmod 72$

$d \times 7 = 1 \bmod 72$

$(d \times 7) \bmod 72 = 1$

Q	A	B	R	T_1	T_2	T
10	72	7	2	0	1	-10
3	7	2	1	1	-10	31
2	2	1	0	-10	31	-72
X	1	0	X	31	-72	X

$d = 31$

$pk = [e, n]$

$= [7, 91]$

$priv.\ k = [d, n]$

$= [31, 91]$

$C = M^e \bmod n$

$= 3^7 \bmod 91 \Rightarrow 3^1 \bmod 91 \Rightarrow 3$

$= 3^1 \times 3^1 \bmod 91 = 9 \bmod 91 = 9$

$= 3^2 \times 3^2 \bmod 91 = 9 \times 9 \bmod 91 = 81$

$= 3^4 \times 3^2 \times 3^1 \bmod 91$

$= 81 \times 9 \times 3 \bmod 91$

$= \boxed{3}$

$q = 7$ $\quad \boxed{\alpha = 3}$

$X_A = 2$

$X_B = 5$

$Y_A = \alpha^{X_A} \bmod q$

$Y_A = 3^2 \bmod 7$

$\boxed{Y_A = 2}$

$Y_B = \alpha^{X_B} \bmod q$

$= 3^5 \bmod 7$

$= 243 \bmod 7$

$\boxed{Y_B = 5}$

$$k_1 = (Y_B)^{X_A} \bmod q$$
$$= 5^2 \bmod 7$$
$$= 25 \bmod 7$$
$$= 4$$

$$k_2 = (Y_A)^{X_B} \bmod q$$
$$= 2^5 \bmod 7$$
$$= 32 \bmod 7$$
$$= 4$$

$\boxed{k_1 = k_2}$ $\therefore$ key exchanged

$q = 17$

$\alpha = 5$

$X_A = 4, \quad X_B = 6$

$Y_A = \alpha^{X_A} \bmod q$

$= 5^4 \bmod 17$

$= 25 \bmod 7 = 7$

$= 7 \times 7 \bmod 17$

$= 49 \bmod 17$

$= 15$

$Y_B = \alpha^{X_B} \bmod q$

$= 5^6 \bmod q$

$= 5^4 \times 5^2 \bmod 17$

$= 15 \times 7 \bmod 17$

$= 105 \bmod 17$

$= 3$

$K_1 = Y_B^{X_A} \bmod q$

$= 3^4 \bmod 17$

$= 81 \bmod 17$

$= 13$

$K_2 = Y_A^{X_B} \bmod q$

$= 15^6 \bmod q$

$= 4 \times 4 \times 4 \bmod 17$

$= 64 - 51$

$= 13$

13 they exchanged

Caesar cipher

$(P+k) \bmod 26$

key = 3

AKAI

0 10 0 8

$(0+3) \bmod 26$

$3 \bmod 26$

3 13 3 11

D N D L

A → 0
B - 1
C - 2
D - 3
E - 4
F - 5
G - 6
H - 7
I - 8
J - 9
K - 10
L - 11
M - 12
N - 13
O - 14
P - 15
Q - 16
R - 17
S - 18
T - 19
U - 20
V - 21
W - 22
X - 23
Y - 24
Z - 25

Play Fair Cipher

key MONARCHY

plaintext → AKAI

M	O	N	A	R
C	H	Y	b	d
e	f	g	i/j	k
L	p	q	s	t
u	v	w	x	z

A K → R U

A I → B S

R J B S

HILL Cipher

plaintext → UPES

$$\begin{bmatrix} H & I \\ L & L \end{bmatrix}$$

$$\begin{bmatrix} 7 & 8 \\ 11 & 11 \end{bmatrix} \quad \begin{bmatrix} U \\ P \end{bmatrix} \quad \begin{bmatrix} E \\ S \end{bmatrix}$$

$$\begin{bmatrix} 7 & 8 \\ 11 & 11 \end{bmatrix} \quad \begin{bmatrix} 20 \\ 15 \end{bmatrix}$$

Vigenere

key → US

LAPTOP

USUSUS

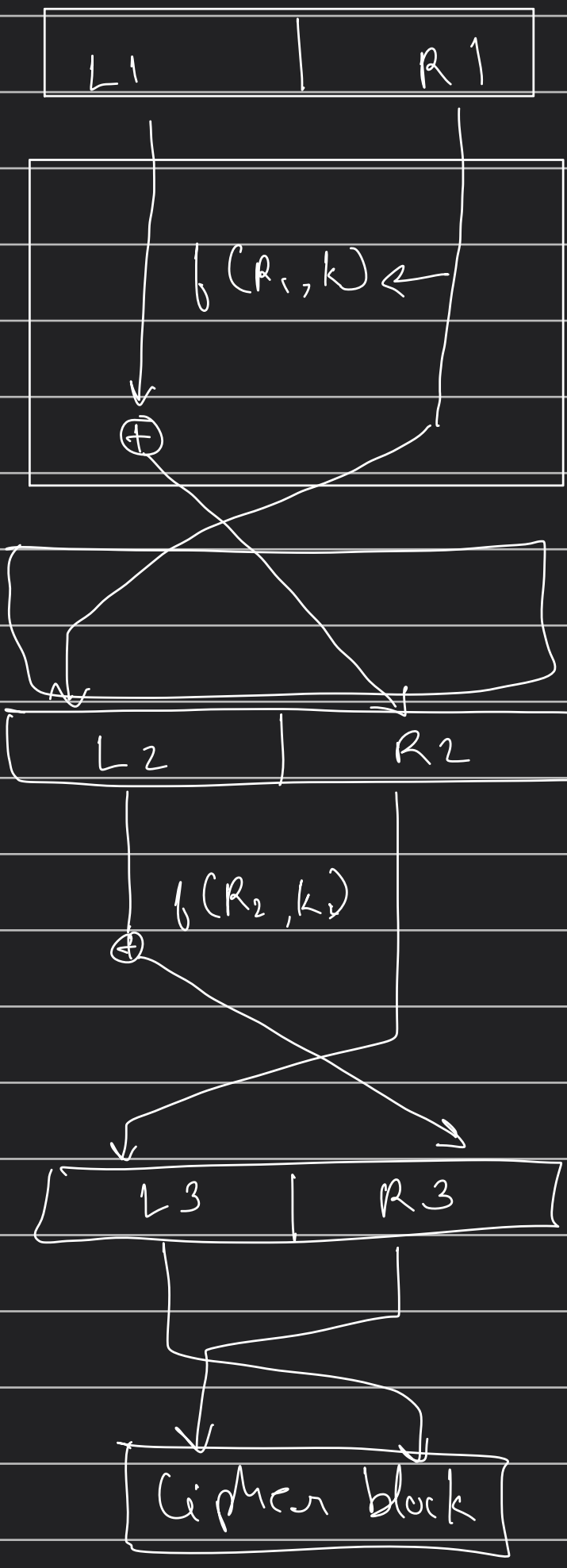
11 0 15 19 14 25

20 18 20 18 20 18

mod 26

31 18 35 37 34 33

Ay


L1
R1
f(R1, k)
L2
R2
f(R2, k2)
L3
R3
Cipher block