SP-23

(1) → (a) Explain the Confidentiality, Integrity & Availability with suitable example.

Confidentiality: keeping information secret and only accessible to those who have Permission.

→ Think of your email account. you have a Password to ensure that only you can access your emails. This Protects your Messages from being read by others.

Intrigrity: Making sure information is accurated & has not been altered or tampered with.

→ Imagine you are writing a report for school. you save it on your Computer. The next day, you open a file, and it looks the same as when you saved it. This means the intrigrity of your report is maintained because no one changed it.

Availability: Ensure that information & resources are accessible when needed.

→ you need to use an online learning platform to study for your exams. Availability means that the platform is up & running whenever you need to use it. so you can access your study materials without any issues.

⇒ Example → online Banking.

Confidentiality → your bank Account information is Protected by a Password & encryption so that only you can see your balance & transactions.

Integrity :→ The bank ensure that your transaction's history is accurate & hasn't been Altered. If you deposited $150, it shows up as $150, not some other amount.

Availability → you can log into your online banking account at any time to check your balance, transfer money, or pay bills. The service is available whenever you need it.

#4 Explain the Access Control, selective field data Integrity & Non-repudiation.

⇒ Access Controls like having a Password on your phone. only ya & people you (who know the Password) can access your phone. So, if your friend doesn't know the Password, they can't open your ~~Password~~ phone & see your messages or photos. Access Control works the same way for Computer systems & data. It ensures that only authorized people can access specific information.
only can you unlock your phone, preventing others from accessing your personal information.

Selective Field Data Integrity:

It ensures that certain parts of your data remain accurate & unchanged unless authorized. Think of it like a book where each chapter must stay correct. If you're writing a school project and your teacher wants to ensure that only the introduction & conclusion can be edited, while the middle part stays the same, that's selective field data integrity.

→ only your teacher can write on change the grades, ensuring their accuracy.

Non-repudiation:

Non-repudiation is a way to guarantee that someone cannot deny the authenticity of their actions or commitments. It's like sending a package with a tracking number & requiring a signature upon delivery. Once the package is delivered & signed for, the person who received it can't deny they got it because there is proof (the signature)

→ The signature proves the package was received, so, the receiver can't claim they didn't get it.

#4 Describe some security mechanism that are include in ITU X.800 recommendation.

The ITU X.800 recommendation Provides guidelines to protect information system.

Encipherment: is locking your data in a safe so that only someone with the correct key can open it. It involves changing your data into a secret code so that if anyone tries to read it without the key, they can't understand it.
→ Turning "HELLO" into "KHOOR" using a secret code. only those who know the code can read the message.

Digital Signature: is like signing a document with your unique signature. It Proves that the message or document really came from you & haven't been changed.
→ Adding a digital signature to an email to Prove it was sent by you & haven't been tampered with.

Access Control: is like having a list of who is allowed to enter your house. Only people on the list can come in. This ensures that only authorized users can access certain information or resources.
→ Using an ID Card to enter an office building, allowing only employees inside.

Data Integrity: is like making sure a recipe hasn't been changed. It ensures that the information sent or stored hasn't been altered or corrupted.

→ A small piece of data added to a file to make sure it hasn't been changed. If the file is altered, the checksum won't match.

Authentical Exchanges: It is like showing your ID Card to Prove who you are. It involves verifying the identity of a user or system before allowing access.

→ Entering a username & password to log into a Computer system.

Traffic Padding: is like adding extra, meaningless messages to hide the real ones. It prevents eavesdroppers from knowing when real messages are being sent & their size.

→ Sending extra messages along with real ones to confuse anyone trying to spy on the Communication.

Routing Control: is like choosing a safe path to walk home to avoid dangerous areas. It involves directing data along secure & reliable paths to protect it from being intercepted.

→ Choosing a secure route for data to travel on the internet to avoid hackers.

(5) ## Explain how Passwords are stored nowdays. Give Examples of tradeoffs between usability & security considering "Authentication".

① Hashing Passwords:

(i) When you create a password for an account, the actual password is not stored!
(ii) It turned into a random string of characters using a process called "Hashing"
(iii) Hashing takes your password & transform it into a fixed-length string looks nothing like the orginal password.
→ If your password "mypassword" it might be hashed into something like "5f4dcc3b5aa765d61deb2f99".

② Salt:

This ensure that even if two people have the same password, their hashes will be different.
→ If your password "mypassword" & salt is "1234", the combined string "mypassword1234" is hashed, resulting in a unique hash.

③ Storing the Hash:

The system stores the salted hash in its database.
→ When you log in, your password is combined with the salt & hashed again. If the result matches the stored hash, you are authenticated.

Different:

① Security:

→ Strong Password (P@ssw0rd!123) are hard for hackers to guess.
→ more secure because they are harder to crack.
→ Adds an extra layer of security.
→ After entering your Password, you receive a text message with a code to enter.
→ Harder for hacker because they need both your phone.
→ your workplace might make you change your Password every few months.
→ Instead of remembering 10 different Passwords, you only remember one master Password for the manager.

④ Usability:

→ Harder to remember.
→ Imagine your email Password "P@ssw0rd!123" instead of "Password". The 1st one is safer, but it's also easier to forgot.
→ Takes more time. If you're in a hurry, having to enter a code from your phone can be annoying.
→ Easier to use because you don't have to remember all your Passwords. If the ~~first~~ Password manager gets hacked, all your stored Passwords could be exposed.

⑧ "Software & system security is all about managing risk"

→ Managing Risk in Security:

Software & system security is all about managing risk.
→ Find the dangers like hackers or viruses.
→ Figure out how bad these dangers could be like stealing money or data.
→ Take steps to reduce the danger like using strong passwords or firewalls.

ATM fraud → Imagine you own some ATMs & you want to know how much money you might lose each year because of fraud.

Annualized loss Expectancy → ALE helps you estimate how much money you could lose in a year due to fraud.

The formula is:

ALE = Single loss Expectancy (SLE) × Annual Rate of Occurance (ARO)

① Small ATM Fraud:
You loss $1000 each time.
It happens 10 times a year.

② Large ATM fraud:
. You loss $10,000 each time.
. It happens 2 times a year.

Calculates ALE.

① For all small ATM fraud:
$ALE_{small} = \$1000 \times 10 = \$10,000$
You expect to lose $10,000 per year from small fraud.

② For Large ATM fraud:
$ALE_{large} = \$15000 \times 2 = \$20000$
You expect to lose $20000 per year from large fraud.

③ Total ALE for both types fra
Total ALE = $10000 + $20000
= $30000.
You expect to lose $30000 per year from all ATM fraud

(5) Explain the Components of a symmetric cipher Model

A symmetric cipher model is a way of securing data by using the same key for both encryption & decryption.

(i) Plaintext: The original, readable message or data that you want to protect. Think of a text message you want to send to a friend.

(ii) Encryption Algorithm: A set of rules of a process that transforms the plaintext into something unreadable. Imagine you have a secret code where each letter is replaced by a different letter or symbol. This code represents the encryption algorithm.

(iii) Secret key: A special piece of information that controls the encryption & decryption process. Both the sender & receiver need to have the same key. If you and your friend agree to use the same secret code to write messages, that code is your secret code.

(iv) Ciphertext: The result after the plaintext has been encrypted. This is the unreadable version of the original message.
After applying your secret code, your message is now a jumble of letters & symbols that only you & your friend can understand.

Decryption Algorithm:

The reverse of the encryption algorithm. It takes the ciphertext & transforms it back into the original plaintext, using the same secret key.
Using the secret code to convert the scrambled message back into the original readable message.

Key Distribution: The method by which the secret key is shared between the sender & the receiver securely.
Imagine you & your friend meet in person to exchange the secret code so no one else can get it.

Analyze the differents between cryptanalysis of Transposition cipher & Caesar ciphr.

Cryptanalysis is the study of breaking ciphers or encrypted messages.

Transposition Cipher:

This cipher mixes up the letters in a message. The letters are same, but they are in a different order.
→ Imagine you have a scrambled word like "LOSHL" instead of "HELLO". To break the code, you try to rearrange the letters back to their original order.
→ you can also look at how letters are grouped together to guess the original message.

Caesar Cipher:

The cipher shifts each letter by a certain number.

for example,

If you shift by 3, 'A' becomes 'D', 'B' becomes 'E', & so on.

→ Since there are only 25 possible shifts, you can try each one until the message makes sense.

→ Some letters appear more often than others. ('E')

by looking at which letters appears the most, you can guess the shift.

Complexity:

Transposition → This one is tougher because there are many ways to mix up the letters. The more mixed up the letters are, the harder it is to figure out the original message.

Caesar: This one is simpler because there are only a few ways to shift the letters. you can try to all possible shifts quickly, so it doesn't take long to break the code.

(*) (i) Given initial permutation "86574231" find the inverse initial Permutation, show with the a bit string example.

(ii) Given a 64 bit key, how would you derive the round key in DES for the 3 round?

(i) Inverse initial Permutation,

InitialPermutation P as "86574231", the goal is to find the inverse initial Permutation P^{-1}.

The bit position 8 goes to position 1.

" " " 6 " " " 2

" " " 5 " " " 3 & so on

If P[1] = 8, then $P^{-1}[8] = 1$.

If P[2] = 6, then $P^{-1}[6] = 2$.

So, the inverse Permutation would be, 87632541.

bit string Example, bit string consider "10110011"

Position 1 taken bit position 8 → 1.

" 2 " " " 6 → 0

" 3 " " " 5 → 0

" 4 " " " 7 → 1

" 5 " " " 4 → 0

" 6 " " " 2 → 0

" 7 " " " 3 → 1

" 8 " " " 1 → 1.

So, the result after Permutation is "10011011".

(11) DES → Data encryption standard is like a recipe for scrambling information so that it's hard to understand unless you know how to unscramble it.

round key → Imagine you have a lock that changes every time you turn it. Each time you turn the lock, you use a slightly different key. In DES, the main key is used to create several smaller keys, one for each round of scrambling.

64 bit key →

1010101011101100001001000110000010011100111011
0110011001100110 1.

Permuted choice 1 (PC-1)

Imagine you are organizing your key by cutting off a few bits & rearranging the rest.
This gives you a shorter key, 56 bits long.
Two halves C0 & D0.

Shifting the key → To make the key for the 2nd round you shift or rotate these halves to the left.
For the 3rd round, you shift both halves by 2 positions.
This means the bits in each half move over by two spots.
→ After shifting you combine C3 & D3 & the rearrange them again to make a 48-bit key.
This 48-bit key is your round key for the 3rd round.

Explain the CIA triad with necessary examples. what it is & its significance?

The CIA is a simple way to understand how to keep information. The CIA Triad is a basic model in cybersecurity that helps us understand how to protect information.

(i) Confidentiality:

Confidentiality is about keeping information secret. only the people who are supposed to see or use the information should have access to it.

→ Imagine you have a diary where write your personal thoughts. you wouldn't want just anyone to read it. so, you might put a lock on it, or keep it in a safe place. This ensures that only you can read what's inside.

→ Passwords protect your email account, ensuring that only you read your messages.

→ Encryption is like a lock on your digital data. If someone steals your files, they can't read them without the key to unlock the encryption.

(ii) Integrity:

Integrity is about keeping information accurate & unaltered. The information should be the same as it was originally, without anyone tampering with it.

Imagine you're baking a cake & with a recipe. If someone changes the recipe while you're not looking, the cake won't turn out right. The recipe's integrity has been compromised.

→ when you send a message online, you want to make sure it arrives, just as you sent it. If someone changes the message in transit, the integrity of the message is lost.

→ checksums or hashes are like digital fingerprints that ensures files haven't been tampered with. If the fingerprint doesn't match, you know somethings wrong.

Availability: Availibility means that information & resources are accessible when needed. If you need certain data on systems, they should be available to you without delay.

→ Imagine you have an important exam tomorrow, and you need to study from your textbook. But, when you go to ~~the~~ get ~~back~~ the book, you find out it's locked in a room, & you ~~don't~~ have the key. You can't study, which means the book isn't available to you when you need it.

→ website that go down due to too much traffic or a ~~cyclic~~ cyberattack are examples of availibility issues. If you can't access your bank account online when you need to transfer money, that's a problem with availability.

Why do we need both End to end & link encryption? Example & explain.

⇒ End-to-End Encryption & Link Encryption are two methods used to Protect data as it travels over a network. Each method secures data in different ways, & sometimes we need to both of ensure the highest level of security.

Let's breakdown →

End-to-end Encryption protects data from the moment it leaves the sender until it reaches the receiver. No one in between, not even the service Provider, can read or alter the data.

When you send a message using an app like whatsApp, end-to-end encryption ensures that only you & the Person you're messaging can read the message. Even the Company running the app can't see it.

⇒ Link Encryption secures data as it travels between two points, like from your Computer to your internet service Provide & from your Isp to another network. However, the data is decrypted at each point & then re-encrypted before it moves to the next point.

Imagine sending a letter through a series of mail carriers. At each stop, the mailman opens the letter, checks it, then puts it back in a new envelope before sending it to

to next mailman. while this keeps the letter safe from being tampered with during each leg of the journey; the letter is briefly visible to each mailman along the way.

Why we need both?

→ End-to-End encryption is excellent for ensuring that only the sender & receiver can read the message, but it doesn't protect the data from being seen or intercepted during its journey.

→ Link encryption protects data each step of the journey, but since the data is decrypted and re-encrypted at each point, someone at one of those points could potentially see or alter the data.

⇒ Combining both methods means your data is protected from start to finish, both during transit & at its destination. This makes it much harder for hackers or unauthorized users to access, read or tamper with your information.

Find the GCD of (450, 120) with Euclidean algorithm.

$450 = 2 \times 3^2 \times 5^2$

$120 = 2^3 \times 3 \times 5$

Common prime factors 2, 3, 5.

lowest power $2^1 = 2$

" " $3^1 = 3$

" " $5^1 = 5$.

Common factors GCD $= 2^1 \times 3^1 \times 5^1 = 30$.

30, the GCD of 450 & 120 is 30.

Is Z_5 as Galois field? If yes, why is it? Show the additive inverse & multiplicative inverse for modulo 5 at. arithmetic.

→ Z usually refers to the set of all integers, both positive & negative $[-3, -2, -1, 0, 1, 2, 3, \ldots]$.

A galosis field is a field with a finite number of elements; It must satisfy these two conditions,

i) The field has a finite number of elements.

ii) It must support addition, substraction, multiplication, division with every operation having an inverse.

⇒ Z by itself is NOT a Galois field because Z is infinite.

Z_p, where p is prime number is a Galois field. Z_p refers to the integers module p, meaning the set $\{0, 1, 2, \ldots p-1\}$ and with arithmetic operations done modulo p.

Additive Inverse:

- The additive inverse of a number a in Z_p is number b such that $a+b \equiv 0 \pmod{p}$.

Ex → Z_5 (modulo 5)

The additive inverse of 3 is 2 because $3+2=5$

Multiplicative Inverse:

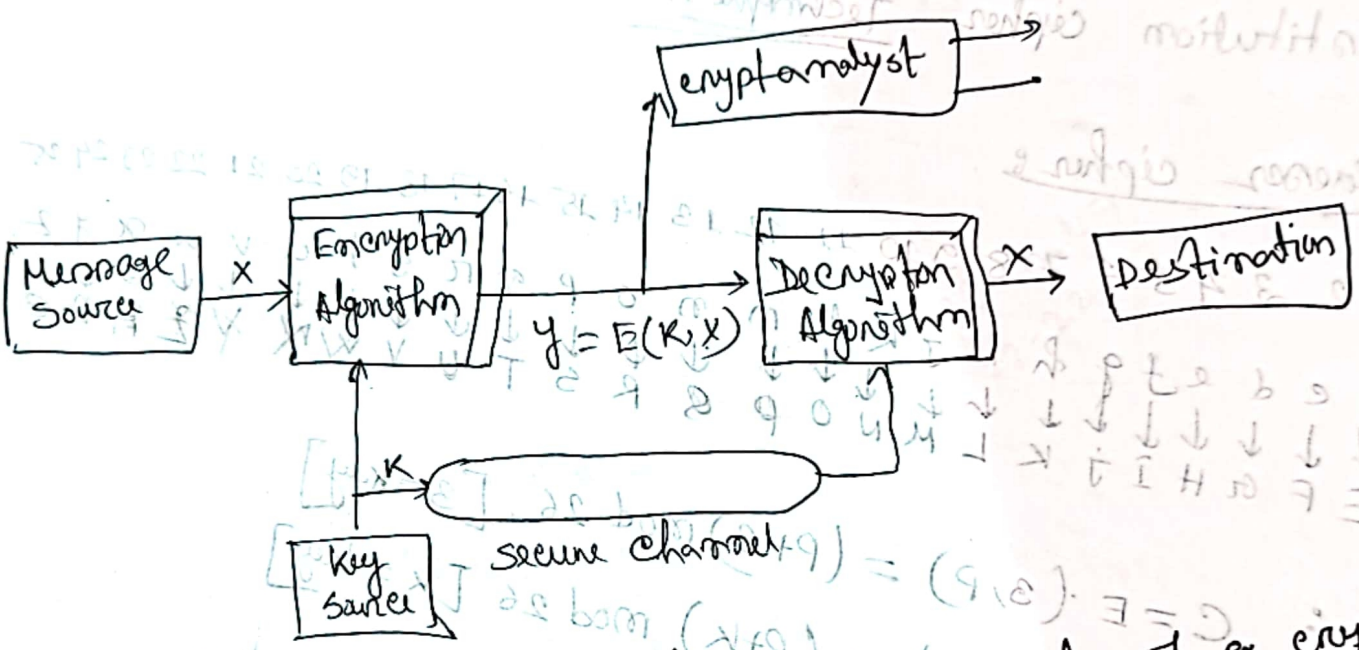
- The multiplicative inverse of a number a in Z_p is a number b such that $a \times b \equiv 1 \pmod{p}$

Ex → $Z_5 \pmod 5$

The multiplicative inverse of 3 is 2 because

$3 \times 2 = 6 \equiv 1 \pmod 5$.

What is the role of cryptanalyst in symmetric key cryptographic model?



In symmetric key cryptographic model, the role of a cryptanalyst is to analyze encrypted messages & attempt to decrypt them without knowledge of the secret key.

The cryptanalyst's goal is to break the confidentiality of the encrypted message, allowing them to read the contents of the message.

12, 17, 18, 19, 20, 22

For a shift cipher find the cipher text for plain text letter x with a shift value of K equals to 3 show modular operations in detail for both encryption & decryption

Substitution cipher Technique:

① Caeser cipher:

	0	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
plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
cipher	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Cipher: $C = E(3,P) = (P+3) \bmod 26$. [3 = key]

$C = E(K,P) = (P+K) \bmod 26$ [K = key]

Plain: $D(K,C) = (C-K) \bmod 26$. [E ⇒ Encryption

$D(3,C) = (C-3) \bmod 26$. D ⇒ Decryption]

Plain Text:

$C = E(3,P) = (w+3) \bmod 26$
$= (22+3) \bmod 26$
$= 25$
$= z$

Again,
$(a+3) \bmod 26$
$= (0+3) \bmod 26$
$= 3 \bmod 26$.
$= 3$.
$= D$.

[**Mod:** $25 \div 26$
$= 0.9 \ldots$
ans 0.9 - point 0.9 arrest value minus 225
$= 0.9 - 0 = 0.9$. multiply with 26
$= 0.9 \times 26 = 25$.]

Again,

$= (m+3) \bmod 26$

$= (12+3) \bmod 26$

$= 15 \bmod 26$

$= 15$

$= P$

Again,

$= (i+3) \bmod 26$

$= (8+3) \bmod 26$

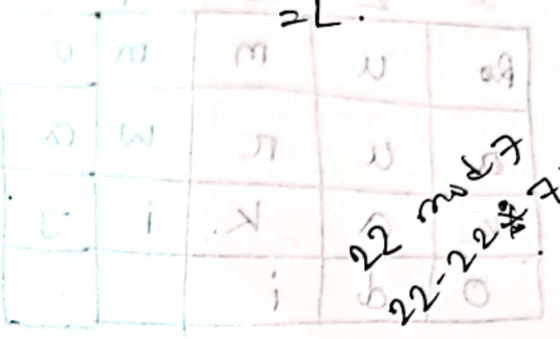
$= 11 \bmod 26$

$= 11$

$= L$.

Cipher Text: wami

ZDPL



22 mod 7

22-22*7+7

Decrypted:

$D = (c-3) \bmod 26$

$= (z-3) \bmod 26$

$= (25-3) \bmod 26$

$= 22 \bmod 26$

$= 22$

$= w$.

Plain text: wami

w a m i

01110111 01100001 01101101 01101001

⊕

Key -> ~~01101110~~ 01101110 01101110 01101110 01101110

encryp -> 000 11001 00001111 00000011 00000111

⊕ 011 01110 01101110 01101110 01101110

Decrypted: 01110111 01100001 01101101 01101001

w a m i

Transposition cipher:

① Column Transposition Technique:

⇒ Plain text: Rumma aur waka ki Jodi

	1	2	3	4	5
row →	R	u	m	m	a
	a	u	r	w	a
	k	a	k	i	J
	o	d	i		

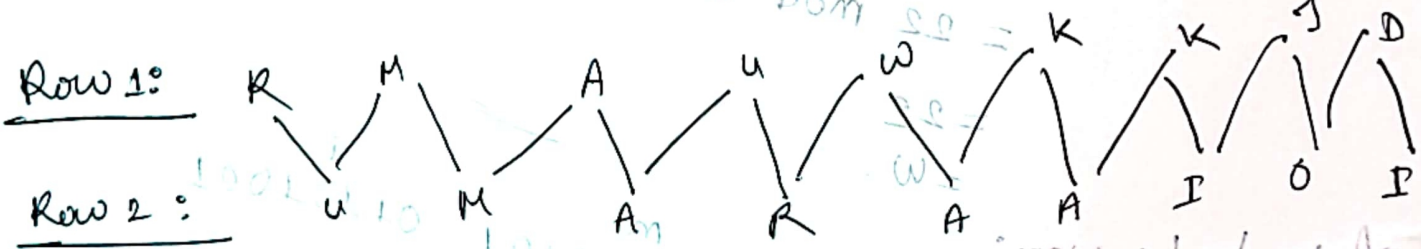
↓ col (important).

key: 3 1 4 2 5.

cipher: MRKIRAKOMWIUUADAAJ.

② Keyless Transposition technique: Rail Fence.

Plain text: Rumma aur waka ki Jodi.



Cipher TEXT: RMAUWKKJDUMARAAIOI.

Alice need to send message "Enemy attack tonight" to bob. Alice & bob have agreed to divide the next into groups of five characters. Encrypt the message using a tabular transposition cipher with encryption keyword 2,5,3,1,4. find out the cipher text, decryption number key. Then decrypt the message so that bob can see that message.

Plain text: Enemy Attack tonight.

Tabular manner of this message.

1	2	3	4	5
E	n	e	m	y
-	a	t	t	a
c	k	-	t	o
n	i	g	n	t

Given key = 2, 5, 3, 1, 4.

Arranging the data according to given encryption key:

2	5	3	1	4
n	y	e	E	m
a	a	t	-	t
k	o	-	c	t
i	t	g	n	h

cipher text: E-CNNAKIET-GMTTTHYAOT.

Decrypt key: 4 1 3 5 2

Rearrange the columns to get the original plain Text.

4	1	3	5	2
E	n	e	m	y
-	a	t	t	a
c	k	-	t	o
m	i	g	h	t

Plain text: Enemy attack tonight

(iii) Show the matrix representation of the transposition cipher encryption key with the key (3, 2, 6, 1, 5, 4). Find the matrix representation of the decryption key.

decryption key is 4 2 1 6 5 3 → Transposition cipher

It is a technique which is used to encrypt the data according to given column key.

message: This is an example

Encryption key → 3, 2, 6, 1, 5, 4

1	2	3	4	5	6
T	h	i	s	-	i
s	-	a	n	-	e
x	a	m	p	l	e

arrange the data according to given encryption key = 3, 2, 6, 1, 5, 4.

Decryption:

Arrange the given ciphertext into table form key = 6.

so, take 6 Columns

no. of rows 18/6 = 3, whereas 18 is total length of message.

⇒ arrange text in up to down manner.

1	2	3	4	5	6
i	h	i	t	-	s
a	-	e	s	-	m
m	a	e	x	l	p

Rearrange the Columns to get original:

4	2	1	6	5	3
t	h	i	s	-	i
s	-	a	m	-	e
x	a	m	p	l	e

So, Decrypt key is 4 2 1 6 5 3.

Encrypt the message "this is an exercise" using Caesar ciphar or additive ciphar with key 20.

Ignore the space between words. Decrypt the message to get the orginal plaintext.

Ex: "this is an exercise"

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}	x_{22}	y_{23}
y_{24}	z_{25}										

key = 20.

Encryption:

t = (19+20) %. 26
= 13 = n.

h (7+20) %. 26 = 1 = b

i = (8+20) %. 26 = 02 = c

s = (18+20) %. 26 = 12 = m

a = (0+20) %. 26 = 20 = u

n = (13+20) %. 26 = 7 = h

e = (4+20) %. 26 = 24 = y

x = (23+20) %. 26 = 17 = r

e = (4+20) %. 26 = 24 = y

r = (17+20) %. 26 = 11 = l

c = (2+20) %. 26 = 22 = w

i = (8+20) %. 26 = 2 = e

s = (18+20) %. 26 = 12 = m

e = (4+20) %. 26 = 24 = y.

ciphertext:

nbemcuhynyleocmy.

Decryption:

$m = (13-20) \% 26 = 19 = t$

$b = (1-20) \% 26 = 7 = h$

$c = (2-20) \% 26 = 8 = i$

$m = (12-20) \% 26 = 18 = s$

$c = (2-20) \% 26 = 8 = i$

$m = (12-20) \% 26 = 18 = s$

$a = (20-22) \% 26 = 0 = a$

$h = (7-20) \% 26 = 13 = m$

$y = (24-20) \% 26 = 4 = e$

$n = (17-20) \% 26 = 23 = x$

$y = (24-20) \% 26 = 4 = e$

$l = (11-20) \% 26 = 26 = z.$

Final Security

(#) Explain a situation where Diffie-Helman exchange might be used. How can recipient verify that a message came from you?

The main purpose of the Diffie-Helman key exchange is to securely developed shared secrets that can be used to derive keys. These keys can then be used with symmetric-key algorithms to transmit information in a protected manner. symmetric algorithms tend to be used to encrypt the bulk of the data because they are more efficient than public key Algorith

The diffe-Hellman key exchange can be used to established Public and Privates keys.

26 mod 11
26/11
a b c

Here, module (P) = 17 & base (g) = 4.

Alice & bobs secreat number a = 3 & b = 6.

$A = g^a \bmod p$.

$\Rightarrow A = 4^3 \bmod 17$ [64-64/17×17

$\Rightarrow A = 64 \bmod 17$

$\therefore A = 13$

when, we do same for bob,

$B = 4^6 \bmod 17$

$\Rightarrow B = 4096 \bmod 17$

$\therefore B = 16.$

Alice then sends the results to Bob. when bob sends to Alice.

$S = B^a \bmod P$. Alice Computes

$= 16^3 \bmod 17$

$= 4096 \bmod 17$

$= 16$.

$\therefore S = A^b \bmod P$. Bob Computes

$= 13^6 \bmod 17$

$= 4826809 \bmod 17$

$= 16$.

both Parties ended up the same results for S, this is the shared secret, which only Alice and bob know. They can ~~see~~ use this to set up a key for symmetric encryption, allowing them to safely send information between themselves in a way that can only they can Access.

Diffie-Hellman is secure because it is extremely hard to figure out the shared secret g^{ab} just from knowing g, g^a and g^b. This relies on a math Problem that's simple one way but almost impossible to reverse, keeping the exchange safe even if someone intercepts Parts of it.

$$(g^a \bmod P)^b \bmod P = (g^b \bmod P)^a \bmod P.$$

23 → 2ण এর মধ্যে এক কোনো নিয়ে আসতে হবে.
Limitations ৩টা.

Diffi-Hellman Key Exchange:

(I) key exchange → A method for securly exchanging cryptographic keys between two Parties.

(II) No Authentication → it doesn't verify identities - making it vulnerable to main-in-the middle attacks.

(III) man-in the middle attacks → Attackers can intercept Communications & impersonate the Parties involved.

(IV) Need for Autentications → To Prevent these attacks, Diffie-Hellman is often combined with methods like digital Certificates & RSA to verify identities.

(II) Example → Suppose → RSA Algorithm:

p = 3.

q = 11.

So, $n = p \times q$.
$= 3 \times 11$.
$= 33$.

and, $z = (p-1) \times (q-1)$
$= 2 \times 10$
$= 20$.

Here, $d = 7$.

$7 \mid 20 \mid 2$
14
$6 \mid 7 \mid 1$
6
$1 \mid 6 \mid 6$
6
0

Again, $e = ((2 \times i) + 1)/d$
$= ((20 \times 1) + 1)/7$
$= 21/7 = 3$.

Public key $(e,n) = (3,33)$

Private key $(d,n) = (7,33)$.

Encryption:

W's, encrypt E, where numerical value is 5.

So, $c = 5^3 \pmod{33} \Rightarrow \boxed{c = p^e \pmod{n}}$

$= 125 \bmod 33$

$= 26.$

Decryption:

$p = c^d \pmod{n}$

$= 26^7 \pmod{33}$

$= 5$

Euler's theorem:

If x & n are coprime then,

$\Rightarrow x^{\phi(n)} = 1 \bmod n$; representation of Euler's theorem.

Example $\rightarrow$ $x = 4, n = 165$

$\gcd(4,165) = 1.$

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

$\Rightarrow (4)^{\phi(165)} = 1 \bmod 165$

Here, $\phi(165) = \phi(15) * \phi(11)$

$= \phi(3) * \phi(5) * \phi(11)$

$= 2 \times 4 \times 10 = 80.$

$\therefore (4)^{80} = 1 \bmod 165$

-21-25)

Here, $\phi(13) = 13-1$ [where 13 is a prime number]

$= 12.$

again, $\phi(14) = \{1, 3, 5, 9, 11, 13\}$

$= 6.$

Digital Signature :

⇒ It plays on very important role in e-commerce, online transaction etc.

⇒ based on asymmetric key cryptography:

→ encryption → Private key.

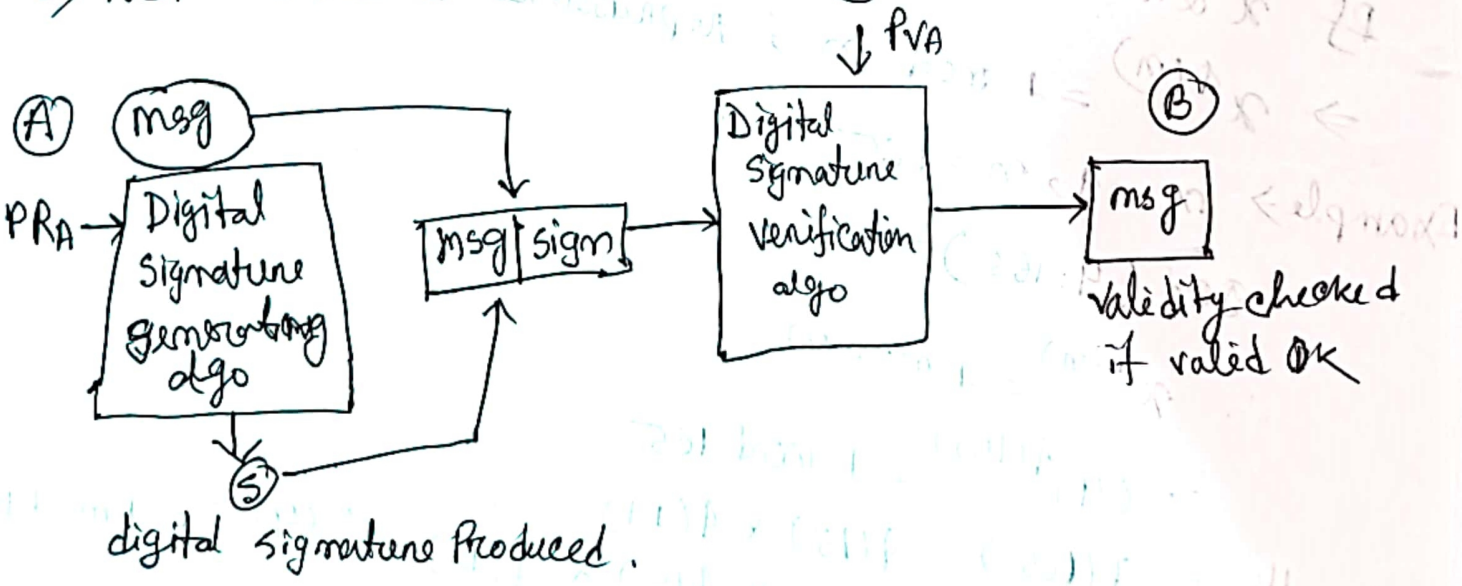
→ decryption → Public key.

⇒ Used for message authentication & non-repudiation & message integrity.

This is particular Person

If someone send something, he can't deny that

⇒ Not used for confidentiality.



Here,

M = Msg, PR_A = Private key of A.

PV_A = Public key of A.

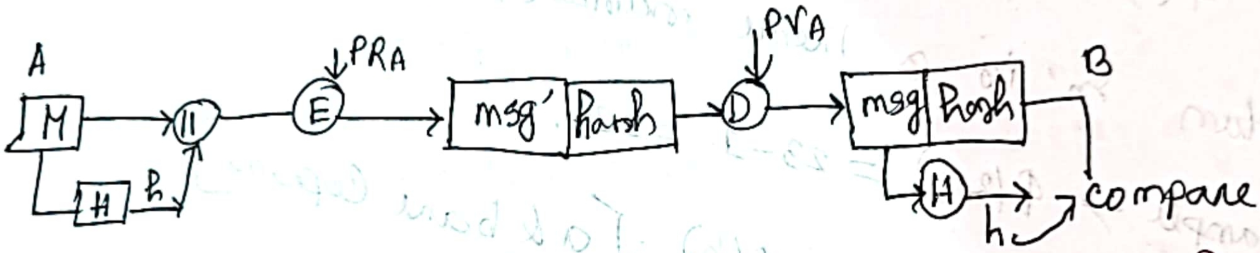
H = hash function,

h = hash value.

$||$ ⇒ append.

E → Encryption

D → Decryption.



A msg send করবে, তারপর একটি Hash function নেবে, তারপর একটি hash value generate হবে, M & H কে append করবে, PR_A দিয়ে encrypted হবে; ... তারপর msg এর hash value এর সাথে নতুন generate করা hash value compare করব যদি তারা same হয়ে থাকে data ঠিক আছে;

Note: The signature must be use some info, unique to the sender to prevent both forgery & denial.

(*) When we sign a document digitally, we send the signature as a separate sender sends 2 docs (msg & signature)

(#) Data encrypt করার জন্য → symmetric এর মাধ্যমে

key " " " → asymmetric " ".

Euler's Totient function :

⇒ $\phi(n)$ for $[n \geq 1]$ is defined as the number of the integers less than 'n' that are Coprime to n.

Example:

$\phi(5) = \{1,2,3,4,5\} = 4$ (number of integers less than n & co-primes)

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

⇒ when 'n' is a Prime number, $\phi(n) = n-1$.

Example → $\phi(23) = 23-1 = 22$.

⇒ $\phi(a*b) = \phi(a) * \phi(b)$ · [a & b are Coprime]

Example:

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

$= \phi(7) * \phi(5)$ [Coprime, gcd (7,5) = 1]

$= 6 * 4$ [n-1]

$= 24$.

Prime factorization এর ক্ষেত্রে,

* $\phi(n) = n * (1 - 1/P_1)$

* $\phi(n) = \phi(a) * \phi(b) = n * (1 - \frac{1}{a}) * (1 - \frac{1}{b})$

Explain symmetric secret key distribution between two Parties ensuring Confidentiality & authentication.

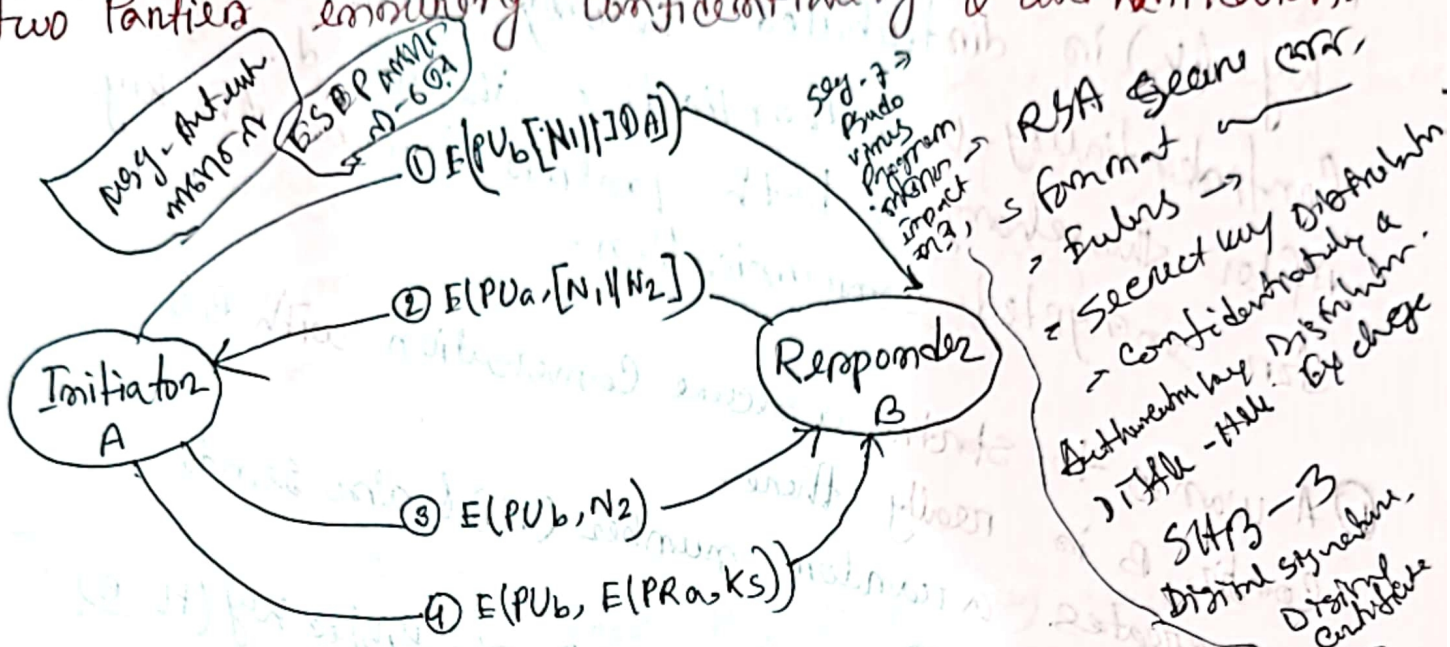


Figure: Public key Distribution of secret keys

Initiator A is Person, and Responder B is Person. The goal is to make sure only A and B know the secret key by the end of the Process. They also want to ~~know~~ be sure they're really ~~want~~ talking to each other, not an imposter.

Confidentiality → By using each other's Public keys, A & B ensure that only the intended recipient can decrypt & read each message.

Authentication: The use of nonces (N1, N2) & both Parties Public / Private keys verifies that each Party is who they claim to be.

This method is ensures that the shared secret key (K) is distributed securely, with both Confidentiality & authentication in mind. After this exchange, both parties can use the key for encrypted communication.

(I) A wants to start a secure conversation with B & Confirm B is really there.

- A creates a random number (N_1) & also sends their ID (ID-A).
- A encrypts this message with B's Public key (PU-B). B can open it. message from A to B,
- B, Here's a random number (N_2) & my ID (ID-A). but only he can read this because it is encrypted with his Public key (PU-B).

(II) B wants to Confirm to A that they received the message & show they are real.

- B Decrypts the message using their Private key, so they see N_1 & ID-A.
- B then creates their own random number (N_2) & sends both N_1 & N_2 back to A.
- This message is encrypted with 'A' Public key (PU-A) so, A can only read it.

message from B to A, the random number (N_1) & my own new random number (N_2), now only he can read it, it is encrypted with his Public key (PU-A).

(iii) A Now Confirms that they're really talking to B.

- A Decrypts the message their Private key & sees N_2.
- To show B that they got this message, A sends N_2 back to B, encrypted with B's Public key (PU. B)
- Message from A to B. I got random number (N_2) & only he can read this it+s encrypted with his Public key (PU. B).

(iv) A Sends the secreat key to B.

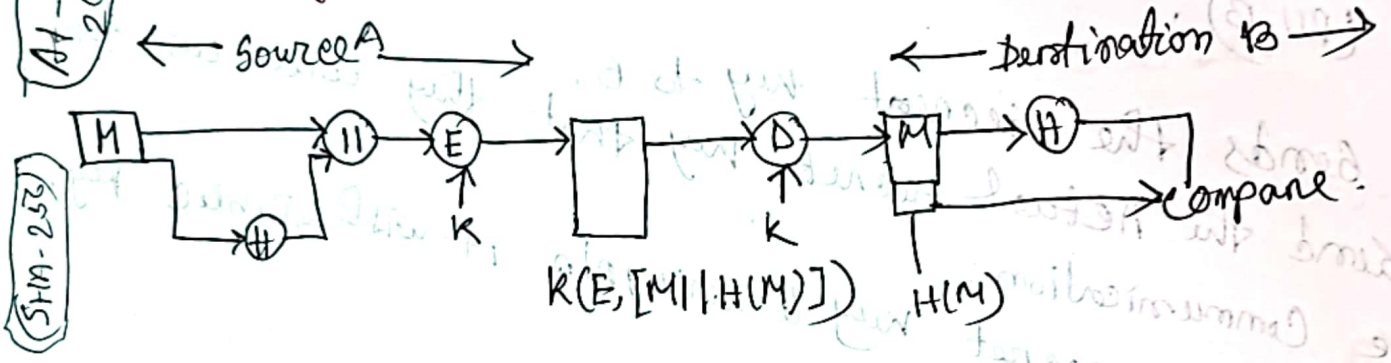
A send the Actual secret key that they will use for Private Communication.

- A Creates a secret key & encrypts it with B's Public key. so, B open it.
- A also uses A's own private key to sign it, so B knows it really Came from A.

Message from A to B, Secret key (K). only he can open this message with his Public key (PU. B).

Alice is sending Message to Bob. The message is "I had my breakfast this morning. Eve wants to modify this message " I did not have breakfast this morning. and wants to send this modified message to bob. How Can Eve be prevented from modifying this message with the help of Hash function?

A4-22 2(a)



To ensure that Eve can't modify Alice's message without being detected, we can use a hash function for integrity checking.

- Alice takes her original message: I had my breakfast this morning, and uses a cryptographic hash function (SHA-256) to create a unique "hash" for it. This Hash, H_1, is a fixed-length code that represents her exact message. Even a tiny ~~change~~ change in the message would create a completely different hash.
- Alice sends bobs two things:
  - → The original message, "I had my breakfast this morning.
  - → The hash of the message (H_1), a unique code created by SHA-256.

when bob receives the message, he calculates his own hash (H_2) by running Alice's message through the same SHA-256 hash function.

- If H_1 & H_2 match, bob can be confident that the message was not tampered with during transmission. If H_1 & H_2 don't match, then bob knows that the message was changed, possibly by someone like eve.

Works:

hash function create a unique code for a specific message — creates a completely different hash. so, if eve tries to change the message to 'I did not have breakfast this morning', the hash no longer match.

- only Alice & bob have the same original hash. so, even if Eve intercepts the message, she cannot alter it without detection.

Elaborate the CIA traid in the perspective of Computer security. Propose a countermeasure to Prevent Hash table attacks in storing Passwords. (Password are not Necessarily unique).

The CIA traid standards for Confidentiality, Integrity, Availability, which are the three main goals of Computer security.

(i) Confidentiality: It means keeping data Private so that only the right people can Access it. when you send a message online, encryption Can Scramble the msg- so no one can read it if they intercept it.

(ii) Integrity: It makes sure that data is correct and hasn't been changed by someone who is not allowed it. If you send a file, a hash can be used to check that the file wasn't changed during.

(iii) Availability: Means that people who are allowed to access data or services can do so whenever they need. Having backups and Protecting against attacks like a Dos ensures that systems stay up and running.

The CIA traid ensures that Computer systems are secure, data stays safe and systems are reliable for users.

Countermeasure to Prevent Hash table Attacks

Password are often stored in database using method called hashing, which Passwords into a scramble string. Hackers can try to guess Passwords using precomputed hash tables.

Even if two people use the same Password, their hashes will be different because of the unique salts.

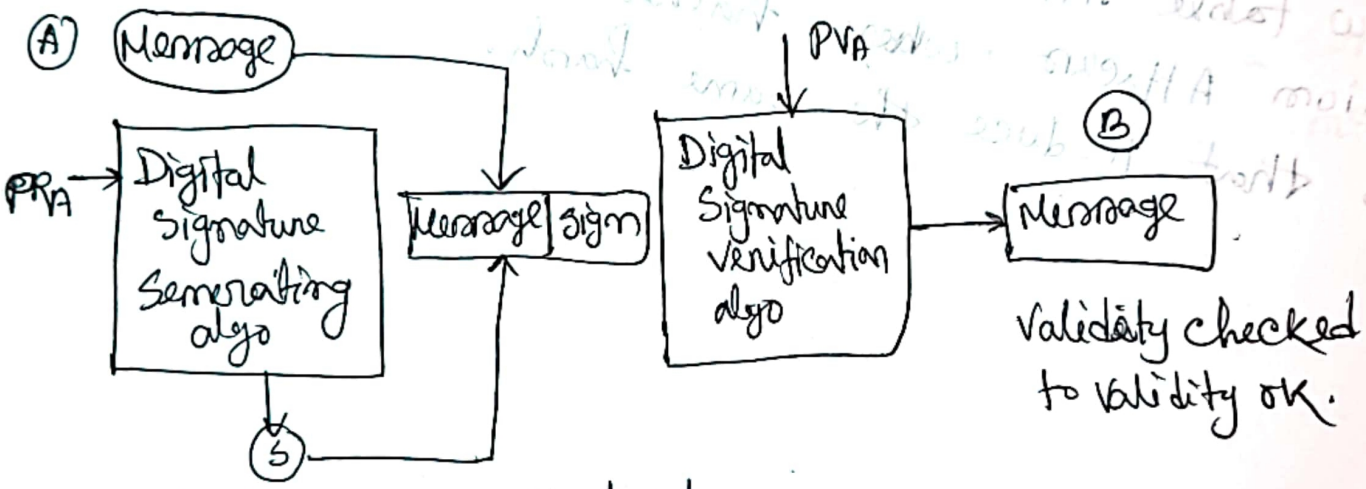
- Rainbow table Attacks, which rely on precomputed hashes.
- Collision Attacks, where hackers try to find two inputs that produce the same hash.

Digital Signature Verification →

It Plays on very important role in e-commerce, online transactions. Based on asymmetric key cryptography.

encryption → Private key.

decryption → Public key.



Digital signature verification is an important use of PGP (Pretty Good Privacy) that helps verify the identity of someone sending an email. If a Journalist wants to confirm that the person messaging them is genuine, they can use a digital signature along with PGP.

Digital signature verification works by using the senders Private key and a special method to create a unique summary of the message called Hash function. Which represents the whole message, is then encrypted with the senders private key to form the digital signature.

The sender sends the message along with the digital signature. The recipient uses the sender's Public key to decrypt the signature and compares the hash with the message. If they match, it confirms the message is authentic, unchanged, and from the claimed sender.

If the hashes don't match, it means the message was altered, the sender's identity is false, or the signature was forged.

Co-Prime হলো — যেগুলোর 1 ছাড়া অন্য কোনো উৎপাদক সাধারণ নেই,
[8 → 1, 2, 4, 8
15 → 1, 3, 5, 15]
1 সাধারণ, তাই Co-Prime.

6 & 9 →
6 → 1, 2, 3, 6.
9 → 1, 3, 9
তাই, Co-Prime নয়.

#4. Explain the use of Euler's Totient function in cryptography. Find $\phi(13)$ & $\phi(14)$.

Prime number হলে → $\phi(n) = (n-1)$

Composite হলে $(14 = 2 \times 7)$ $= n \cdot \left(1 - \frac{1}{P_1}\right) \cdot \left(1 - \frac{1}{P_2}\right)$.

Eulers function $\phi(n)$ is important for RSA encryption. It helps generate the Public and Private keys for secure Communication.

RSA

(i) Two Prime numbers P & q chosen. $n = P \cdot q$ is for encryption. $\phi(n) = (P-1)(q-1)$ is calculated. and Public key $\gcd((e, \phi(n))) = 1$.

(ii) Public & Private key. The Private d is Computed using $d \cdot e \equiv 1 (\bmod \phi(n))$. e encryption & d is decryption.

(iii) Prime check: $\phi(n)$ helps verify the Property of n is cryptographic systems.

→ $\varphi(13)$ and $\varphi(14)$.

Euler's Totient function $\varphi(n)$:

Prime → $\varphi(n) = n-1$.

Composite → If $n = p_1^{e1} \cdot p_2^{e2} \cdots$,

$$\varphi(n) = n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right)$$

Qs $= 13$, $\varphi(13)$ = Prime number so,

$$\varphi(13) = n - 1 = 13 - 1 = 12$$

Qs $= 14$, $\varphi(14)$ = Composite.

$$= 2 \times 7$$

$$\varphi(14) = n \cdot \left(1 - \frac{1}{p_1}\right) \cdot \left(1 - \frac{1}{p_2}\right)$$

$$= 14 \cdot \left(1 - \frac{1}{2}\right) \cdot \left(1 - \frac{1}{7}\right)$$

$$= 14 \left(\frac{1}{2}\right) \times \left(\frac{6}{7}\right)$$

$$= \frac{2}{14} \times \frac{3}{7}$$

$$= 6.$$

Ans $\varphi(13) = 12$

$\varphi(14) = 6$.

A good hash function has the property that the results of applying the function to a large set of inputs will produce outputs that are evenly distributed and apparently

A function takes some input data and turns into a fixed-size output, called a hash or digit. A good hash function has these two key properties.

① Even Distribution:
when you give a hash function a large set of inputs, it should produce outputs that are spread evenly across the possible - range of values. This prevents too many inputs from mapping to the same hash value and ensures better performance.

② Randomness:
The output should look random. Even a tiny change in the input (changing one character) should create completely different hash value. This is called avalanche effect and makes it hard to guess output.

These properties make hash function useful for securing data, checking file integrity and detecting unauthorized changes.

How hash function help in Intrusion Detection.

① Checking file Intrigrity → A trusted value of important files is stored when the system is safe. If hash value is changed, it means the file was modified - this could be a sign attack.

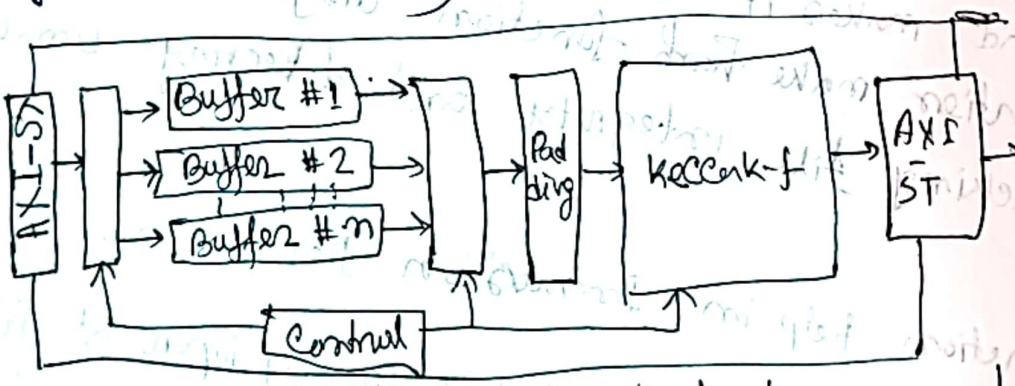
② Detecting Network Tampering:

Hash functions can verify that data sent over a network has not been changed. If a hacker tries to change the data in transit, the hash of the received data won't match the original hash, showing that something was wrong.

③ finding malware:

Every file has a unique hash value. If a files has matches the hash of a known malware file in a database. It's a flagged as a dangerous. This helps quickly identify malicious files.

④ Explain the Design and steps of secure Hashing Algorithm: 3 (SHA-3)



SHA-3 is a special formula used to take any data and turn it into a fixed-sized output. It's like taking a block of clay and shaping it into a unique stamp. No matter how big the data is, the output is always the same size, like 256 bits or 512 bits. It helps checking

if the data has been changed or damaged.

→ SHA-3 is built differently from older hashing methods.

(I) Absorbing the input.

(II) Squeezing out the hash. (II)

Steps:

(I) SHA-3 uses a state which is big block of 1600 bits.

(II) To process the input, SHA-3 first adds bits to it. This is called Padding. Padding makes sures the input fits perfectly into blocks that can be processed.

(III) After Padding, the input is split into chunks or blocks. The size of each blocks depends on a part of SHA-3 called the rate (r).

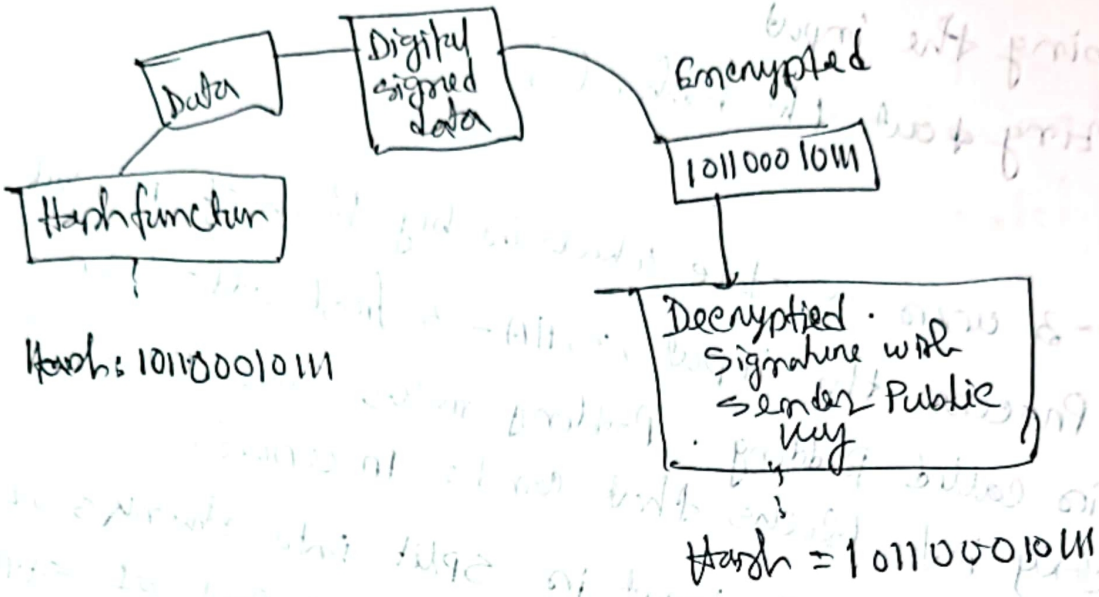
(IV) Each block of the input is XORed with the 1st r bits of the state. After mixing a block, the entire state is updated using a process called Keccak-f Permutation.

(V) Keccak-f Permutation is the brain of SHA-3 of 5 steps. (I) Theta (II) Rho, (III) Pi (IV) Chi (V) Iota

(VI) Once All input blocks are Proceeds, SHA-3 starts giving out the hash bits.

If more output bits are needed, the keechak-f Permutation is applied again, more bits are squeezed out until the desired length is reached.

(5) Explain origin, integrity and non repudiation are ensured in Digital signature.



Digital signatures are like electronic fingerprints. They prove who sent a message, that it wasn't changed and that the sender cannot deny sending it.

(i) Ensure origin:

The sender signs the message using their Private key. Only the sender has this Private key, so anyone verifying the signature knows it could only come from them.

→ This message is hased, The hash is encrypted with the sender's Private key to create the digital signature.

(ii) Ensuring Integrity:

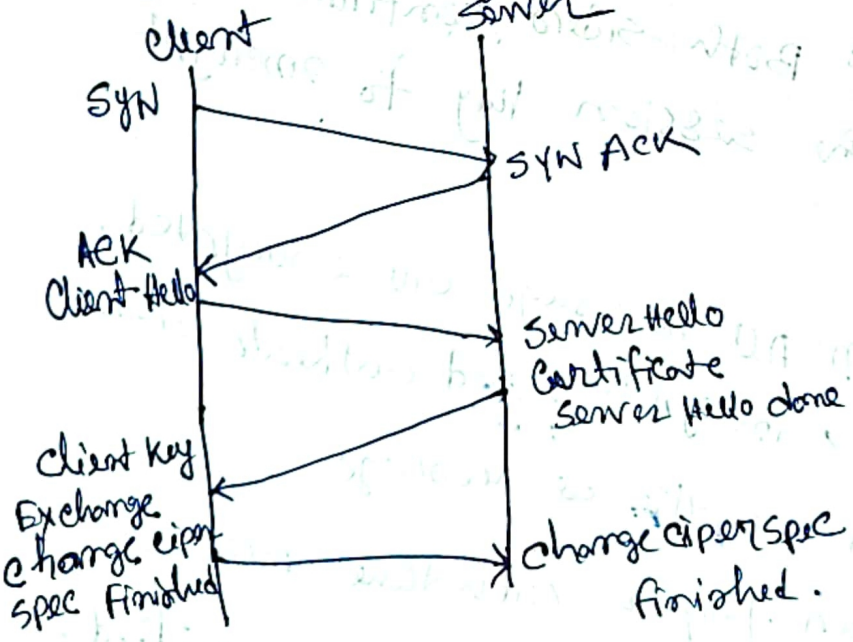
When the receiver gets the message, they verify it using the senders Public key.

The receiver decrypts the digital signature using the senders public key to get the hash. Hashes received message and compares it to the decrypted hash. If the hashes match, the message was not tampered with.

(iii) Ensuring Non-Repudiation:

The sender cannot deny sending the message because the digital signature is unique to their private key. Since only the sender has the private key, no one else could have created the signature.

(5) How does transport Layer security (TLS) start a secured communication involving two parties



TLS secure Communications

① **Handshake Begins:** The client sends supported encryption methods and a random number.

② **Server Responds:** The server agrees on an encryption method, sends its certificate and another random number.

③ **Certificate check:** The client verifies the servers Certificate. If valid, it Proceeds, if not, it warns the user.

④ **key exchange:** The client and server use the Public key to surely create a shared session key.

⑤ **Session key ready:** Both sides Confirm secure setup and use the session key to encrypt Communication.

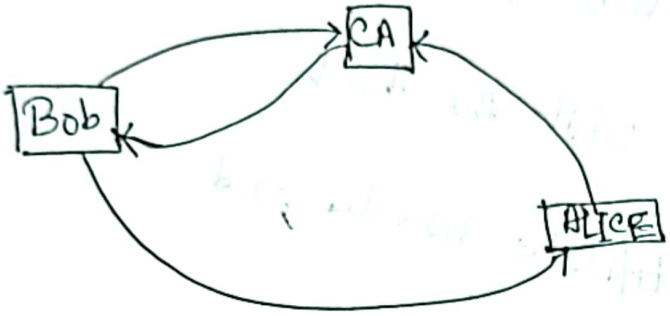
⑥ **Secure Communication:** All message are encrypted, ensuring Privacy, integrity, and authentication.

Privacy → No one Can see the message.

Integrity → No One Can tamper with the message.

Authentication → The servers identity is verified.

(8) Why are bob's ID Information, Certificate authority information and Bob's Public key are included in the digital Certificate issued to Bob?
why those are hashed and how can you be sure that a Particular certificate authority provided the Certificate?



Information is included — in Bob's Digital Certificate:

(I) Bob's ID Information → This identifies the Certificate owner (Bob). It typically includes details Bob's name, email address, and organization. It allows others to verify that the Certificate belongs to Bob.

(II) Certificate authority Information → Identifies the trusted organizations that issued the certificate.
The CA is a trusted third Party that guarantees the certificate is valid and links bob's identity with the Public key.

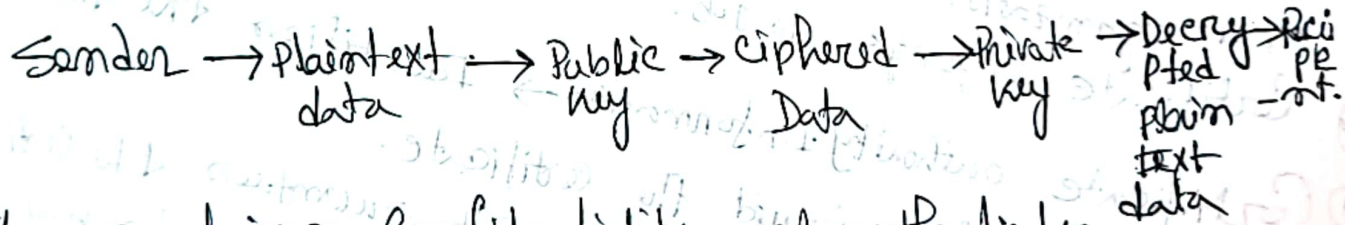
(III) Bob's Public key → This is the key used by others to encrypt data that only Bob can decrypt with his Private key.
It always secure Communication and ensures the authenticity of Bob's Message.

Hashing: A hash function Converts the certificate data into a fixed length string. This Process ensures data intrigrity. Hashing ensures the Certificate stays unchanged. If the certificate data is altered, the hash will be different

How to verified the Certificate Authority:

- Trusted CAs have Public keys stored in system.
- The Public key is used to decrypt the CAs digital signature, on the certificate.
- The system creates a hash ~~function~~ of the Certificate data.
- It compares this hash with the hash from CA signature.
- If they match, the certificate is valid and issued by the CA.

⑤ Demonstrate how Confidentiality and authentication are achieved in S/MIME.



S/MIME achieves Confidentiality and authentication through the use of encryption and digital signatures.

① Confidentiality in S/MIME

Confidentiality ensures that only the intended receipient can read the message.

→ The sender encrypts the email Content using the receipent's Public key.
Only the receipent, who possesses the matching Private key, can decrypt and read the message.

Process → The sender retrieves the receipents Public key, and encrypts the message using this Public key. The encrypted message is sent to the receipient. The recipient uses their Private key to decrypted the message.

(ii) Authentication in S/MIME

The Sender signs the message with their Private key. The recipient can verify the sender's identity using sender Public key. The signature also ensures the intregity of the message.

Process ⇒ The sender creates a hash of the message, encrypts it with their Private key and ~~will~~ send ~~with~~ it with the message. The recipient use the sender Public key to decrypt the signature and compare hashes to verify authenticity and integrity.

⑧ Explain IP traffic Processing with IPSEC for outbound Packets.

when sending data using IPsec

① Security policy check →

Each outbound Packet is checked against SPD. The SPD determines whether to apply IPsec, bypass it, or discard the Packet based on destination address and Port.

② Security Association Selection:

If the SPD indicates IPsec is required, the system looks up the security SA in the SAD.

The SA Contains the rules for encryption and authentication.

③ IPsec Processing:

Based on the SA, the Packet undergoes IPsec Processing.

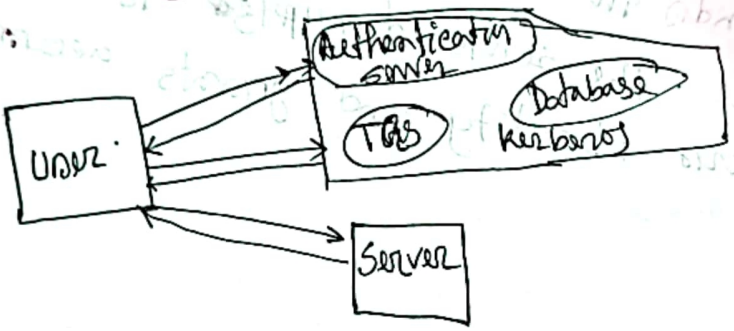
→ The data & the full Packet is wrapped in an IPsec header.
→ Ensures the data Confidential, so only the intended recipient Can read it.
→ Adds a digital signature to Confirm the senders identity & Integrity.

(iv) Packet forwarding:

After Processing, the secure Packet is sent to its destination. For transport mode, only the data is encrypted. For tunnel mode, the entire Packet is encrypted.

(v) Router: If the sender is behind a secure router or firewall, it Performs IPsec operations like filtering and encapsulating Packets before sending them.

(8) Show the steps of Kerberos Protocol with the aid of Necessary diagrams:



① User login: The user enters their username and Password on their Computer. The Computer sends the username to the Authentication Server (AS) to request.

② AS Issues a Ticket-Granting Ticket:

The AS checks the username and verifies it with its database. If valid, the AS creates a TGT and sends it to the user. The TGT is encrypted with the users Password.

③ Request for Service Access:

The users Computers sends the TGT to the TGS to request access to the specific service. The TGT is used to Prove the users identity to the TGS.

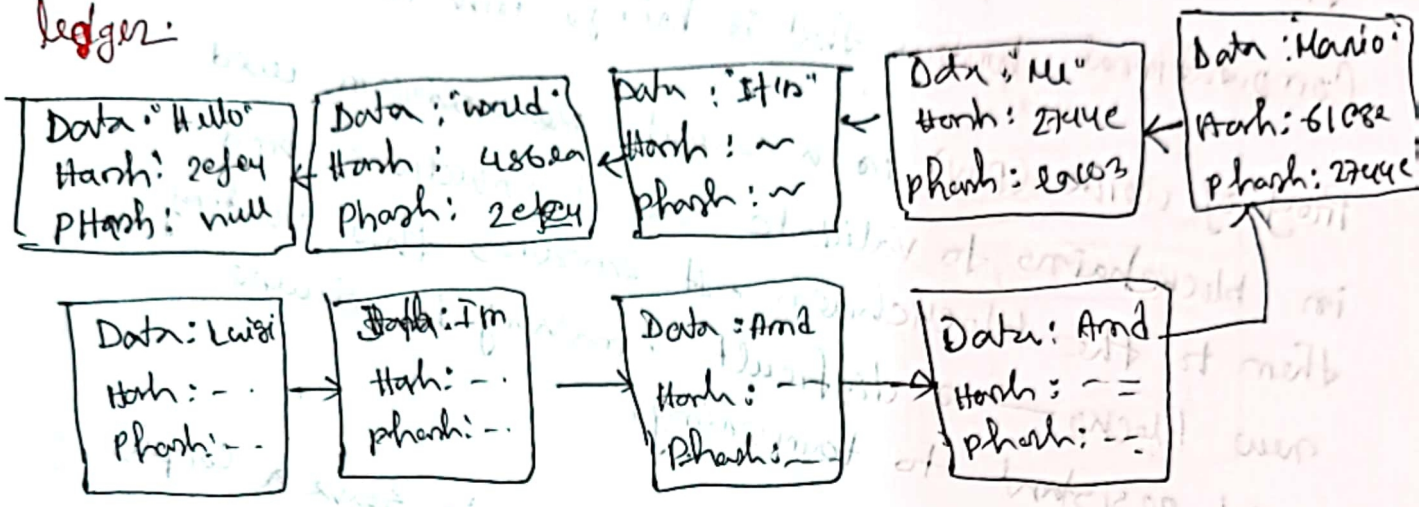
④ TGS issues a Service Ticket:

The TGS verifies the TGT.
If valid, it creates a service Ticket for the requested service and sends it to the user.
The service Ticket is encrypted with the services secret key.

⑤ Accessing the service:

The Computer sends the service Ticket to the requested service. The service decrypts to verify the users identity and grants access.

(8) Show with example and explain the difficulty of inserting a fake block of transaction into a blockchain ledger.



Data: Browser
Hash: ac6b1
phash: 2744e

SHA256("Browser") ⇒
ac6b1 dd10 ---

SHA256("Browser" + ac6b1dd10) ⇒
4cb01 ---

Blockchain is a distributed, secure ledger that stores transactions in linked blocks. Altering or inserting a fake block is extremely difficult.

→ In a block chain, each block stores transaction data and a unique hash, linking it to the previous block. If someone tries to insert a fake block by changing a transaction, the hash of that block changes, breaking the chain.

→ To make the fake block valid, the attacker must recalculate the hashes for all subsequent blocks, which requires solving complex puzzles. Even if this is done, the attacker needs control over 51% of the networks computational power to get the fake chain accepted. These machanisms make it nearly impossible to alter the blockchain.

⑧ Explain the Proof of work (PoW) mechanism in the context of blockchain and comment on the Computational burden that it brings with it.

Proof of work (PoW) is a security mechanism used in blockchains to validate new transactions and them to the blockchain. It ensures that creating new blocks — is difficult; making it secure and resistant to tampering.

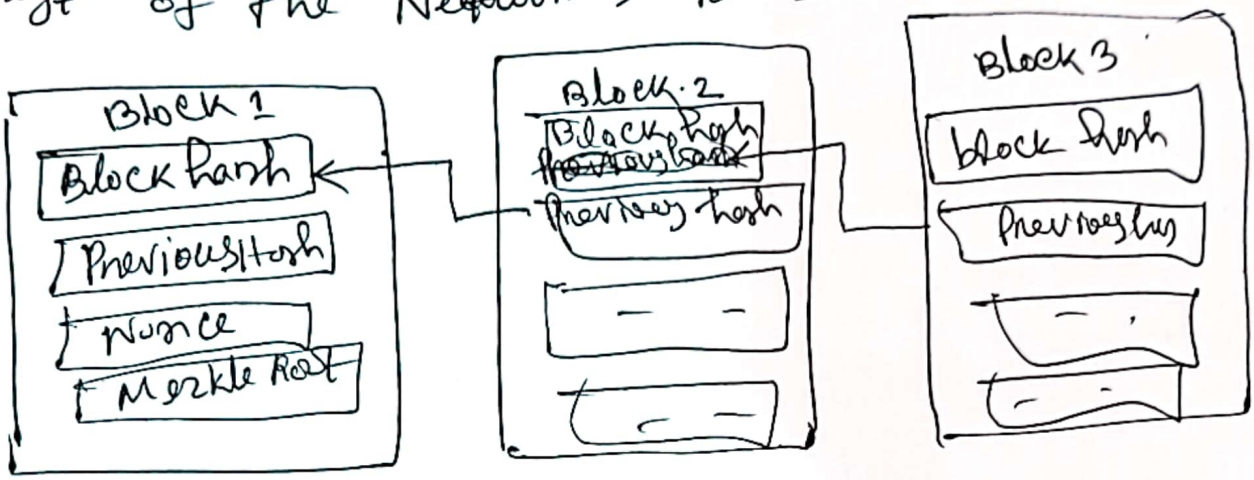
PoW works:

In Proof-of-works, miners Complete to solve a complex Mathmatical ~~Solve~~ Puzzle by finding a hash value that meets specific conditions (starts with zero). They repeatedly try different numbers in the blocks data until they find the correct hash, Proving they did the work. The solution then sent to other nodes for verification; and once most nodes agree, the block is added to the blockchain.

Computational burden:

Proof of work requires massive computing power, as miners must make billions of guesses to find the correct hash, consuming significant electricity. The Process is slow and expensive, needing specialized hardware like ASICs. The Computational burden ensures security, making it almost

impossible for attackers to alter the blockchain without redoing all the work and controlling most of the Networks Power.



Proof of work (PoW)