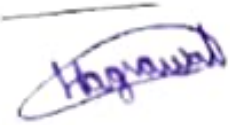
Declaration and statement of authorship

I, bearing Registration Number 106118036, agree and acknowledge that:

1. The assessment was answered by me as per the instructions applicable to each assessment, and that I have not resorted to any unfair means to deliberately improve my performance.

2. I have neither impersonated anyone, nor have I been impersonated by any person for the purpose of assessments.

Signature of the Student : 

Full Name : Harshit Agrawal

Roll No. : 106118036

Sub Code : CSPE 18

Mobile No. : 7016004637

106118036
Adv. Crypto Endsem

Date
Page

1.)

a) Plaintext space $X = \{a, b, c\}$
Ciphertext space $Y = \{1, 2, 3, 4\}$
Key Space $K = \{K_1, K_2\}$

Encryption fn : $e_K =$

	a	b	c
K_1	1	2	3
K_2	2	3	4

Let 'P' be a random discrete variable over X and 'K' be random discrete variable over set K.

So, $Pr[P=a] = 1/4$, $Pr[P=b] = 1/4$, $Pr(P=c) = 1/2$

$Pr(K=K_1) = 1/4$, $Pr[K=K_2] = 3/4$.

Now, Let 'C' be a discrete random variable over set 'Y'.

$$\therefore Pr[C=y] = \sum_K Pr(k)\, Pr(d_k(y))$$

So, $Pr[C=1] = Pr(K_1) \cdot Pr(a)$

$= \frac{1}{4} * \frac{1}{4} = \frac{1}{16} \rightarrow (i)$

$Pr[C=2] = Pr(K_1) \cdot Pr(b) + Pr(K_2) \cdot Pr(a)$

$= \frac{1}{4} * \frac{1}{4} + \frac{3}{4} \cdot \frac{1}{4} = \frac{1}{4} \rightarrow (ii)$

$$P_x[C=3] = P_x(k_1) \cdot P_x(c) + P_x(k_2) \cdot P_x(b)$$
$$= \frac{1}{4} \cdot \frac{2}{4} + \frac{3}{4} \cdot \frac{1}{4}$$
$$= \frac{5}{16} \rightarrow (iii)$$

$$P_x[C=4] = P_x(k_2) \cdot P_x(c)$$
$$= \frac{3}{4} \cdot \frac{1}{2} = \frac{3}{8} = \frac{6}{16} \rightarrow (iv)$$

Now, for perfect secrecy,

~~$P_x[C=y | P=x] = P_x[C=y]$~~

$$P_x[P=x | C=y] = P_x[P=x]$$

Now

$$P_x[x|y] = \frac{P_x[x] * P_x[y|x]}{P_x[y]}$$

Now,

$$P_x[y|x] = \sum_{\{k : d_k(y)=x\}} P_x[k]$$

So, $P_x[1|a] = P[k_1] = \frac{1}{4}$

$P_x[1|b] = 0$

$P_x[1|c] = 0$

$P_x[2|a] = P_x[k_2] = \frac{3}{4}$

$P_x[2|b] = P_x[k_1] = \frac{1}{4}$

$P_x[2|c] = 0$

$P_X[3|a] = 0$

$P_X[3|b] = \frac{3}{4}$

$P_X[3|c] = \frac{1}{4}$

Computing A Posteriori Probabilities, we get

$$P_X[a|1] = \frac{P_X[a] * P_X[1|a]}{P_X[1]} = \frac{1/4 * 1/4}{1/16} = 1$$

Here itself we can see that

$$\boxed{P_X[a|1] \neq P_X[a]}$$

Hence, we can say that it is

NOT Perfectly Secure

9.17 b) Defining the actions and knowledge of the parties for all 3 messages.

1.) a) If receive 'x' as message 1, B can choose a 'y' and compute $k = x^y \bmod p$ as the key.

b) Compute $Y = g^y \bmod p$ & using it as signature (Y, x) and encrypts it using k.

c) B has no idea message was received from A.

2.) a) Receive (Y, c) as second message
b) A computes $k \cdot y^n \bmod p$.
c) Now since 'k' is publickey, A can use it to decrypt 'c' getting S.
d) Thus now we verify S is B's signature on (Y, x).

Thus, finally A concludes that sender of Message 2 knows.

→ k; encryption was done using this
→ B's signing key, since we got signature S.
→ X & y discrete log x (A decrypted c using $k = Y^n$, but anybody could've computed k as x^y).

→ (Y, X) signature, since sender signs it, it includes X which A had chosen before sending first message.

Now, A knows message 2 sender is B. A & B share 'k'.

3) Now on receiving (') as third message, B decrypts it and verifies plaintext was sent by A on (X, Y). As the sender of message 3 is deciphered as A, & again A & B share K.

1.> c) Multiplicative inverse of (7^4) wrt to prime polynomial x^3+5.

i) $5x^2 + 2x + 3$

Using Extended Euclidean algorithm, we get (mod 7)

x^3+5		1	0
$5x^2+2x+3$	$3x+3$	0	1
$6x+3$	$2x+4$	1	$4x+4$
5		$5x+3$	$6x^2+4x+1$

$$
\begin{array}{r|l}
 & 3x+3 \\
\hline
5x^2+2x+3 & x^3+5 \\
 & \underline{x^3+6x^2+9x} \\
 & \quad x^2+5x+5 \\
 & \quad \underline{x^2+6x+9} \\
 & \qquad 6x+3
\end{array}
$$

$$
\begin{array}{r|l}
 & 2x+4 \\
\hline
6x+3 & 5x^2+2x+3 \\
 & \underline{5x^2+6x} \\
 & \quad 3x+3 \\
 & \quad \underline{3x+5} \\
 & \qquad 5
\end{array}
$$

So, Multiplicative inverse does not exist as $\gcd(x^3+5, 5x^2+2x+3) = 5 \neq 1$

1.) d) Construct 1 out of 2 OT protocol using ECC.

A-

Sender

Receiver

$b_0 \rightarrow$ private key

$P_{B_0} = b_0 G$

$P_{B_1} \rightarrow$ Chooses Randomly

P_{B_0}, P_{B_1}

$a \rightarrow$ private key

$m \rightarrow$ message Plaintext

Encryption

$P_{m_0} (x_{m_0}, y_{m_0})$

$P_{m_1} (x_{m_1}, y_{m_1})$

$k \rightarrow$ random integer $(1 < k < p-1)$

$P_{C_0} = [(kG), (P_{m_0} + kP_{B_0})]$

$P_{C_1} = [(kG), (P_{m_1} + kP_{B_1})]$

Decryption

$C_1 b = kGb_0 = kP_{B_0}$

$P_{m_0} + kP_{B_0} - kP_{B_0} \rightarrow P_{m_0}$

$P_{m_1} + kP_{B_1} - kP_{B_0} \rightarrow$ Random value

∴ We can see that the sender doesn't get the knowledge about the message chosen by receiver and receiver doesn't know about the other message.

So, now if Receiver doesn't choose P_B randomly then this protocol fails. So, we accommodate this by,

We choose a B initially such that $B = b_0 b_1$, so, receiver computes $P_{B_0} = b_0 G$
and $b_1 = B / b_0$

Now, computing P_{B_1} for this is a hard problem.

∴ We can say that the protocol works.

2) a) Given $u = g^a$, $v = g^b$, $w = g^{ab}$

Prover | Verifier

Random d $v' = g^d$

$w' = g^{acd}$ ⟶

⟵ Generates random c

Calculates $e = d + bc$ ⟶

Now, Verifier checks

$g^1 = v' v^c$ & $u^e = w' w^c$

only possible if prover knew a, b or both.

* Completeness:

If prover is honest, then

$g^c = g^{d+bc}$

& $v' v^c = g^d (g^b)^c = g^{d+bc}$

$u^e = (g^a)^e = g^{acd+bca}$

$w' w^e = g^{acd} (g^{ab})^e = g^{acd+bca}$

So, $Pr[x \in L ; \text{Checks passes}) = 1$

* Soundness:-

If relationship b/w u, v & w is incorrect, we can define 'a' using discrete log

$u = g^a * b$ using dis crete log

$V = g^b$

but $w \neq g^{ab}$ otherwise relationship would be correct.

Finally 'e' sent from Prover is either d+bc or not d+bc. So,

$u^e \neq w$ so, second check failed

If $e \neq d+bc$, then

$V' V^c = g^d (g^b)^c = g^{d+bc}$

but $g^e \neq g^{d+bc}$ so first check fail.

$P[x \notin L, \text{check pass}] \leq S$

∴ Perfect Zero knowledge.

→ Simulator generates random e and picks random c.

→ Simulator creates V', w', so that $d = e - bc$.

2.) b) Pedersen commitment $(g^a h^{r_1} \quad g^b h^{r_2})$

$h = g^x$

So, we choose a random 'k' $g^a g^{x r_1}, g^{a+x r_1}$

Send $(g^a h^{r_1})^k * g^{-k(a+x r_1)}$

This is 1

Now, when we sent,

$(g^b h^{r_2})^k \; g^{-k(a+x r_2)}$

Not result is 1

$a_1 = g^a h^{r_1}$

$u_2 = g^b h^{r_2}$ →

P ← k V

$u_1^k, ka + k r_1 r_2$ →

$u_2^k, ka + k r_1 r_2$

Verify $u_1^k \cdot g^{-(ka + k r_1 r_2)} = 1$

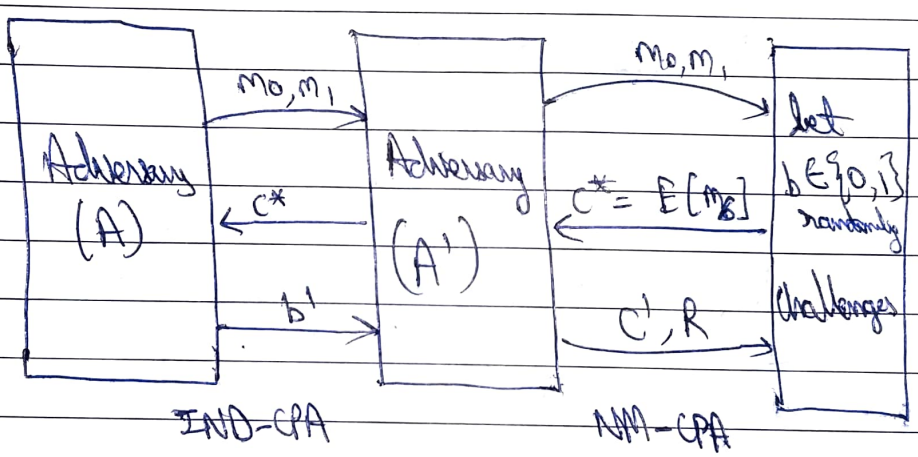
$u_2^k \cdot g^{-(ka + k r_1 r_2)} = c \neq 1$

3.)

a) Prove that NM-CPA $\rightleftharpoons$ IND-CPA

Let us assume that we have an adversary A which challenges/attacks under the IND-CPA (Chosen Plaintext Attack) (Indistinguishability)

Let there also be an adversary A' trying to attack the challenges using NM-CPA (Non-malleable)



$c' = E[m_b', r]$

$R = \begin{cases} d(c') = r \cdot d(c^*) \\ \text{relation between } c^* \text{ \& } c' \Rightarrow \text{Malleability} \end{cases}$

→ Now, if the adversary A can break the IND-CPA by finding the correct b' always, then A' can always get the correct relation R.

Hence, we proved IND-CPA insecure → NM-CPA insecure

So, $\neg$(IND-CPA) $\rightarrow$ $\neg$(NM-CPA)

$\therefore$ NM-CPA $\rightarrow$ IND-CPA

For the other way round, we have no proof but it is always valid, therefore IND-CPA $\not\rightarrow$ NM-CPA.

This is because if NM-CPA can be broken, the IND-CPA need not be broken.

3.) b) Bitcoin forking is the splits that happen in the transaction chain based on different user opinions about transaction history.

→ These splits create new versions of Bitcoin currency and are a natural result of the structure of blockchain system, which operates without a central authority. These forks allow for different buying opportunities for the cryptocurrency.

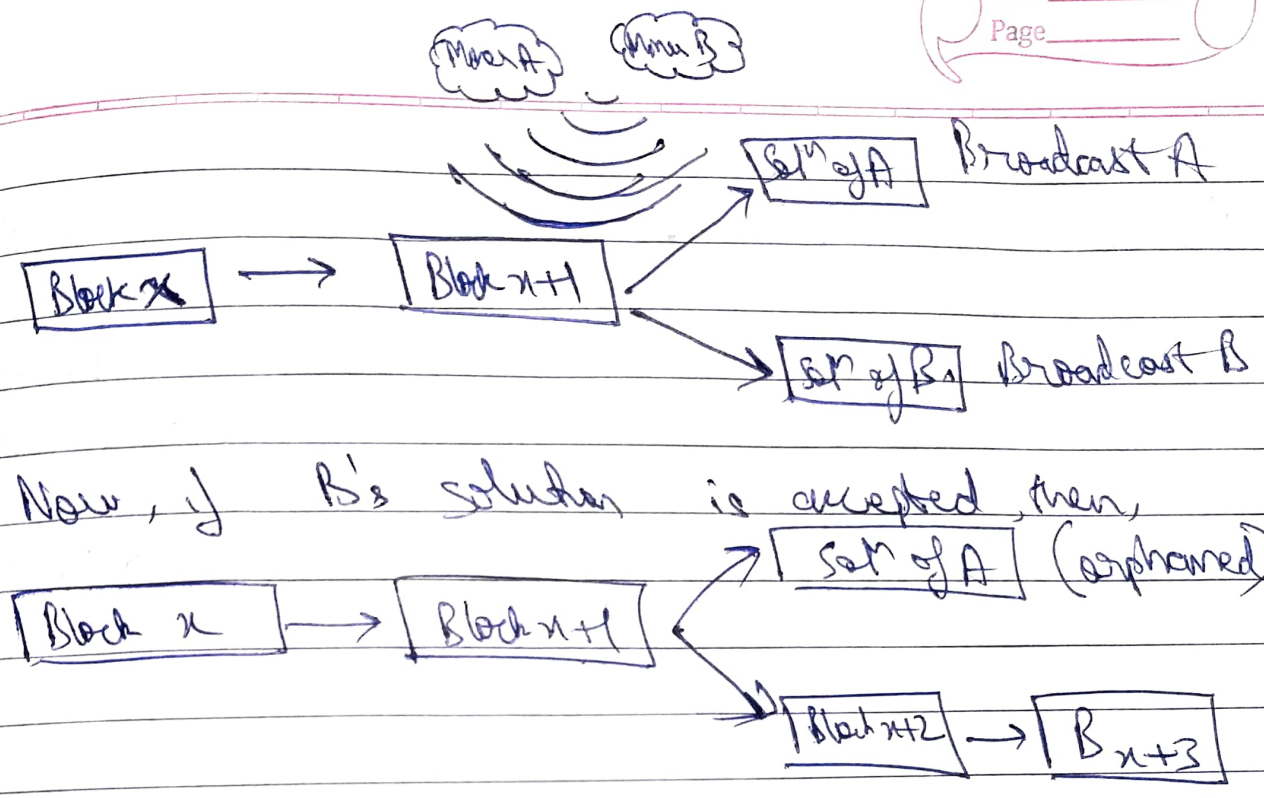
→ A forking tends to happen when there are two or more miners that find a block at the same time.

→ In such situations both miners will mine the same block and will have to BROADCAST their solution onto the Bitcoin network.

→ The various nodes on the Bitcoin network will accept the first broadcast and reject the other solutions.

→ Once the fork branch is chosen, the upcoming mined blocks will be appending to the chosen block and rest of the fork branches are left. These are called as Orphan Blocks

→ This situation is very rare and occurs in a 1:60 block frequency.



Now, if B's solution is accepted, then,

3.)

c) Secondary Pre-Image Resistance :-

→ It is the property of a hash function that it is computationally infeasible to find another input m_2 that has the same output as given input m_1,

i.e., for given m_1, it is hard to find m_2 such that

$$H(m_1) = H(m_2)$$

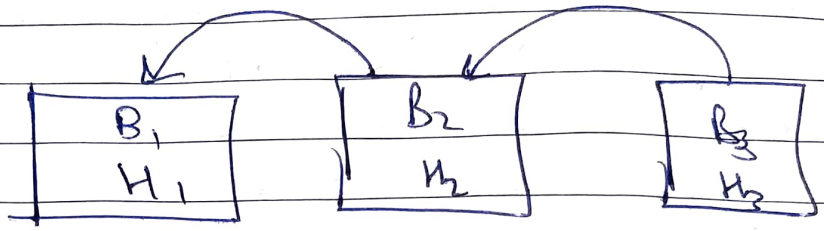
where $H(x)$ is the hash function.

* Security in Blockchain Technology :-

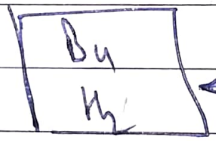
In block chain each block consists of the the hash of the previous block it is linked to, in the block header.

→ If the used hash function is not secondary pre-image resistant, then there may be another block which can have the same hash of a block in the main blockchain. So, hackers can easily fork the blockchain if the used hash function is not secondary pre-image resistant.

Example:- Let us assume there are 3 blocks in the blockchain B_1, B_2, B_3 and their hashes are H_1, H_2, H_3 respectively.



Now, if used hash function H is not the secondary pre-image resistant, then it is easy to find another block with the same hash H_2.



← Same as B_2.

So, now, B_3 won't know which block (B_2 or B_4) is its predecessor as there are 2 blocks with same hash H_2.

So, if used Hash function is secondary pre-image resistant, then it is hard to fork the blockchain thus making it Secure.

3) To prove Shamir Secret sharing is perfect security.

→ To achieve perfect security, the adversary should gain no additional advantage by knowing $t-1$ shares instead of 't' shares.

i.e.,

$$P_r\left[\text{secret} = S \mid \underset{\text{anything}}{\text{before knowing}}\right] = P_r\left[\text{secret} = S \mid \underset{\text{shares}}{\text{after knowing } t-1}\right]$$

R.H.S :-

$$P_r\left[\text{secret} = S \mid \text{knows } t-1 \text{ shares}\right]$$

$$= \frac{P_r\left[\text{knowing } t-1 \text{ shares} \mid \text{secret}=S\right] * P\left[\text{secret}=S\right]}{P_r\left[\text{knowing } t-1 \text{ shares}\right]}$$

(∵ Bayes Theorem).

= Since the shares are uniformly distributed randomly, the P_r [knowing $t-1$ shares | secret=S]

$$= P_r\left[\text{knowing } t-1 \text{ shares}\right] = \left[\frac{1}{p}\right]^t$$

$$\therefore P_r\left[\text{secret} = S \mid \text{knowing } t-1 \text{ shares}\right] = P\left[\text{secret} = S\right]$$

Hence Proved

3.) c) Given : $P = (0, 2)$

$E(\mathbb{F}) \rightarrow y^2 = x^3 + x + 4$

$GF(23)$

Solution: differentiating, we get

$$2y \frac{dy}{dx} = 3x^2 + 1$$

$$\therefore \frac{dy}{dx}(d) = \frac{3x^2 + 1}{2y}$$

Now, if $y \neq 0$, since non infinity points sum

$\therefore 0 = x^3 - m^2 x^2$

$x_3 = m^2 - 2x_1$ and $y_3 = m(x_1 - x_3) - y$

Substituting $(0, 2)$

$$P_3 = (0, 2) + (0, 2) + (0, 2)$$

For $2P \Rightarrow \lambda \Rightarrow \frac{3x^2 + 1}{2y}\Big|_{0,2} \Rightarrow \frac{1}{4} \bmod 23$

$= 6$

$x_3 = 36 - (0+0) \bmod 23 = 13$

And, $-y_3 = 80 \bmod 23$

$\therefore y_3 = (-11) \bmod 23$

$\Rightarrow y_3 = 12$

So, $2P = (13, 12)$

Now, $(13, 12) \oplus (0, 2)$

$$\lambda = \frac{12-2}{12-0} = \frac{10}{13} \bmod 23$$

$$= (10 * (13^{-1} \bmod 23)) \bmod 23$$

$$= 22$$

Now,

$$x_3 = 22^2 - (13+0) \bmod 23$$

$$= 471 \bmod 23$$

$$x_3 = 11$$

Now,

$$-y_3 = (22 \cdot 11 + 12) \bmod 23$$

$$-y_3 = 14$$

$$\therefore y_3 = (-14) \bmod 23 = 9$$

$$\therefore 3P = (11, 9)$$

Answer