Page No.
Date

ws

Challenge 1

We use openssl comands on our terminal to obtain key from priv.pem, input from cipher.bin and prints the output after decryption which is our flag.txt

Challenge 2

The name of the issuer can be seen by going to https://www.cse.iitb.ac.in and then going to the section connection security details in safari browser settings. After that tap on show certificate & you view a certificate on which the issuers name appears

Challenge 3

n = order of generator
h_1 & h_2 are the hash of the first 2 elements of the message list that are choose

$s_1 \Rightarrow s_1k = h_1 + \gamma x \pmod{n}$ $\quad s_2k = h_2 + \gamma x \pmod{n}$
[γ will be same]

subtracting $(s_1 - s_2)k = h_1 - h_2 \pmod n$

$$k = (h_1 - h_2) \cdot (s_1 - s_2)^{-1} \pmod n$$

$s_1 k = h_1 + r x \pmod n \rightarrow rx = s_1 k - h_1 \pmod n$

$$x = (s_1 k - h_1) \cdot r^{-1} \pmod n$$

Challenge 4

Variant 1:

If Eve has a signature (R, s) on any message m, it can:

- Recompute r locally by evaluating $H(m \| P_x) \bmod q$
- Compute $h = H(R_x, P_x, m) \bmod q$ like the signer
- $x = (s - r) \cdot h^{-1} \pmod q$

Steps:

- Obtain public key P_x
- Request a single signature on a chosen message m
- Recompute $r = H(m \| P_x) \bmod q$ & $h = H(R_x, P_x, m) \bmod q$
- Compute $x = (s - r) h^{-1} \pmod q$
- Forge signature on challenge message using the recovered x.

Page No.
Date

Variant 2:
We choose messages such that their first halves are identical.
For a 1 byte message m we have $\lfloor |m|/2 \rfloor = 0$ hence $m_{[:0]}$ is the empty string. Therefore for any single byte message the hash inputs become identical, and so the server computes the same γ for each such signed message.

$$s_1 = \gamma + h_1 \cdot x \pmod q \qquad s_2 = \gamma + h_2 \cdot x \pmod q$$

$$s_1 - s_2 = (h_1 - h_2) \cdot x \pmod q$$

$$x = (s_1 - s_2)(h_1 - h_2)^{-1} \pmod q$$

With x recovered we proceed like in variant 1 to compute correct γ for the challenge message & forms a valid signature (R, s) that will pass verification