Satyam Kumar Modi, 2019CS50448
Rupanshu Shah, 2019CS10395

Question 1

0.1 Shamir's Trick

We will use the Shamir's trick which states that given n, e, f, w, y as input, where n is a positive integer, e and f are relatively prime and w and y are elements of Z_n^*, that satisfy $w^e = y^f$ and outputs $x \in Z_n^*$ such that $x^e = y$.

Proof: Since, e and f are co-prime, $gcd(e, f) = 1$, we can compute s, t such that $es + ft = 1$. Compute $x = y^s.w^t$, $x^e = y^{se}.w^{te} = y^{se}.y^{ft} = y$

0.2 CRHF construction

$N = pq$ and e is a random prime in $Z_\phi(N)$ that is coprime to $\phi(N)$. The key is (N, e) and a random integer $z \leftarrow Z_N^*$. The hash function is defined as $H_{N,e,z} : Z_N^* \times Z_e \rightarrow Z_N^*$ where $H_{N,e,z}(x, y) = x^e.z^y(modN)$.

0.3 Proof that is a CRHF

Assume that above defined hash function is not a secure CRHF. Ths means there exists a ppt adversary $\mathcal{A}$ such that given (N, e, z), it is able to produce a collision (x_1, y_1) and (x_2, y_2) such that ${x_1}^e.z^{y_1} \pmod N = {x_2}^e.z^{y_2} \pmod N$. We present a ppt reduction $\mathcal{B}$ such that it can break the RSA.

The reduction $\mathcal{B}$ does the following:

- Challenger of RSA sends (N, e, y) to the reduction $\mathcal{B}$.
- Reduction $\mathcal{B}$ sends (N, e, y) to the adversary $\mathcal{A}$.
- Now, adversary $\mathcal{A}$ produces two collisions (x_1, y_1) and (x_2, y_2) to the reduction $\mathcal{B}$ such that ${x_1}^e.y^{y_1} \pmod N = {x_2}^e.y^{y_2} \pmod N \implies (x_1.x_2^{-1})^e \pmod N = y^{y_2-y_1} \pmod N$. (The inverse for x_2^e exists as $x_2 \in Z_N^*$ which means $gcd(x_2, N) = 1$, which implies x_2^e must also be co-prime to N. The inverse of x_2^e can be calculated from the Euclid's division Lemma. Same argument works for the inverse of y^{y_1}).
- Reduction $\mathcal{B}$ uses Shamir's Trick to get x such that $x^e = y$ in the following way:
  - Here, $f = (y_2 - y_1)$, $(y_2 - y_1)$ must be co-prime to e as $y_1, y_2 \in Z_e$.
  - Here $w = x_1.x_2^{-1}$ as $x_1, x_2 \in Z_n^*$ and $y = x^e \pmod N$, both w and y are in Z_n^*. (The inverse of x can be efficiently computed as $gcd(x, e) = 1$ and using Euclid's division algorithm, we can get (a, b) such that $x.a + e.b = 1$, here a mod e is the inverse of x in Z_{ϕ_N}).
  - Since, the reduction $\mathcal{B}$ knows $n = N, e, w, f, y$, it uses Shamir's trick and gets x.
- Reduction $\mathcal{B}$ sends x to the adversary.

Proof or correctness follows from the proof of correctness of Shamir's trick.

Question 2

We are given $\mathcal{E} = (KeyGen, Enc, Dec)$ CCA-secure encryption scheme where $\mathcal{E}$ encrypts $n - bit$ messages. We construct encryption scheme $\mathcal{E}' = (KeyGen', Enc', Dec')$ such that $\mathcal{E}'$ encrypts $n - bit$ messages. $\mathcal{E}'$ is no-pre CCA secure but not CCA secure. Ciphertext of $\mathcal{E}'$ is twice the size of ciphertext of original encryption scheme $\mathcal{E}$.

$KeyGen'$:
$(pk, sk) \leftarrow KeyGen$
k_1 : PRF key
$pk' = pk; sk' = (sk, k_1)$
return (pk', sk')

$Enc'(m, pk)$:
$\alpha = Enc(m, pk)$
return $\alpha||\alpha$

$Dec'(ct_1||ct_2)$:
$m^* = F(sk, k_1)$ where F is a secure PRF that maps $n - bit$ strings to $n/2 - bit$ strings
If $(ct_1 == ct_2)$:
then return $Dec(ct_1, sk)$
Else
{
$\beta_1 = Dec(ct_1, sk)$
$\beta_2 = Dec(ct_2, sk)$
If $\beta_1 = m^*||m^*$
then return $m^*||m^*$
Else return $0^{n/2}||m^*$
}

$\mathcal{E}'$ is CCA-no-pre secure:

Proof:
Proof is through a sequence of Games. **Game 0** is the original CCA-no-pre security game.
Game 1 is same as **Game 0** with PRF F being replaced by totally random function F'

Claim 1: **Game 0** is indistinguishable from **Game 1**

Proof (Claim 1):
If there exists a p.p.t $\mathcal{A}$ that can distinguish between **Game 0** and **Game 1** with non-negligible advantage then there exists a reduction $\mathcal{B}$ that breaks PRF security of F.

$\mathcal{B}$ chooses $(pk, sk) \leftarrow KeyGen$.
$\mathcal{B}$ obtains a $n/2 - bit$ string m^* from PRF Challenger.
For $\mathcal{A}$'s decryption queries(post-challenge), $\mathcal{B}$ responds according to Decryption Protocol of $\mathcal{E}'$.
For Challenge messages, $\mathcal{B}$ always returns encryption of m_0.
$\mathcal{A}$ finally sends its guess b' corresponding to the **Game b**. $\mathcal{B}$ forwards b' to PRF Challenger.
Probability of $\mathcal{B}$ winning PRF security game = Probability of $\mathcal{A}$ winning this distinguish game.
Thus, if there exists a p.p.t $\mathcal{A}$ that can distinguish between **Game 0** and **Game 1** with non-negligible advantage then there exists a reduction $\mathcal{B}$ that breaks PRF security of F.
Hence proved.

Claim 2: If there is a p.p.t. adversary $\mathcal{A}$ that can win **Game 1** then there exists a p.p.t. reduction $\mathcal{B}$

that breaks CCA-no-pre security (and consequently CCA security) of $\mathcal{E}$

Step 1: Challenger sends pk to $\mathcal{B}$. $\mathcal{B}$ forwards pk to $\mathcal{A}$
Challenger also chooses random bit b
$\mathcal{B}$ chooses a random $n/2 - bit$ string m^*. Choosing m^* randomly is like computing totally random function F' on the fly and assigning $F'(sk) = m^*$

Step 2: $\mathcal{A}$ sends m_0, m_1 challenge messages to $\mathcal{B}$.
$\mathcal{B}$ forwards these challenge messages to Challenger.
Challenger sends Challenge ciphertext ct^*(Encrypton of m_b) to $\mathcal{B}$.
$\mathcal{B}$ sends $ct^*||ct^*$ to $\mathcal{A}$

Step 3: (polynomially many Post-challenge decryption queries)
$\mathcal{A}$ sends $(ct_1||ct_2)_i$ to $\mathcal{B}$.
$\mathcal{B}$ checks if $ct_1 == ct_2$. If yes, then $\mathcal{B}$ forwards ct_1 to Challenger, gets reply m_i from Challenger. $\mathcal{B}$ sends m_i to $\mathcal{A}$
If $ct_1 \neq ct_2$, then $\mathcal{B}$ sends $(ct_1)_i$ to Challenger, receives $(\beta_1)_i$ from Challenger.
Similarly,$\mathcal{B}$ sends $(ct_2)_i$ to Challenger, receives $(\beta_2)_i$ from Challenger.
$\mathcal{B}$ checks if $(\beta_1)_i == m^*||m^*$. If yes, then $\mathcal{B}$ returns $m^*||m^*$ to $\mathcal{A}$. Else it sends $0^{n/2}||m^*$ to $\mathcal{A}$

Step 4: $\mathcal{A}$ sends guess b' to $\mathcal{B}$.
$\mathcal{B}$ forwards b' to Challenger.

Probability of $\mathcal{B}$ winning (against CCA-no-pre for $\mathcal{E}$) = Probability of $\mathcal{A}$ winning in above construction.
Therefore, if $\mathcal{A}$ breaks CCA-no-pre security of $\mathcal{E}'$ then $\mathcal{B}$ breaks CCA-no-pre security of $\mathcal{E}$.
Hence proved.

<u>$\mathcal{E}'$ is not CCA-secure:</u>

Proof:
p.p.t. adversary $\mathcal{B}$ wins as follows:

- $\mathcal{B}$ chooses two $n - bit$ random strings m_0, m_1 such that first $n/2$ bits of m_0 are not same as last $n/2$ bits of m_0.
- $\mathcal{B}$ computes $ct_0 = Enc(m_0, pk)$ and $ct_1 = Enc(m_1, pk)$.
- $\mathcal{B}$ sends one Decryption query $ct_0||ct_1$
- $\mathcal{B}$ gets back $m' = 0^{n/2}||m^*$.
- $\mathcal{B}$ sets $m_3 = m^*||m^*$ and m_4 a random string such that $m_3 \neq m_4$
- $\mathcal{B}$ sends m_3, m_4 as challenge messages to the challenger
- $\mathcal{B}$ receives Challenge ciphertext $(ct^*||ct^*)$
- $\mathcal{B}$ chooses a random $n - bit$ string m_5 and encrypts it using pk, such that $ct_5 = Enc(m_5, pk)$ and $ct_5 \neq ct^*$.
- $\mathcal{B}$ sends $ct^*||ct_5$ as a decryption query. If $\mathcal{B}$ receives $m^*||m^*$ as the decryption then $\mathcal{B}$ guesses $b' = 0$(i.e., $ct^*||ct^*$ is Encryption of m_3).
  Else $\mathcal{B}$ receives $0^{n/2}||m^*$ as the decryption in which case it guesses $b' = 1$(i.e., $ct^*||ct^*$ is Encryption of m_4).
- $\mathcal{B}$ wins with probability 1. $\blacksquare$

Question 3

Let $\mathcal{S} = KeyGen, Sign, Verify$ be a secure Digital signature scheme. that signs $n - bit$ messages. Construct $\mathcal{S}'$ from $\mathcal{S} = KeyGen', Sign', Verify'$ that signs $n - bit$ messages as follows:

$KeyGen' = Keygen$

$Sign'(m, sk)$:
Choose $n - bit$ random string r
Return $\sigma = (r,\ Sign(r, sk),\ Sign(m \oplus r))$

$Verify'(m, \sigma, vk)$:
$\sigma = (\sigma_1, \sigma_2, \sigma_3)$
Check if $Verify(r, \sigma_2, vk)$ and $Verify(m \oplus \sigma_1, \sigma_3, vk)$ both return 1.
If yes, then return 1, else return 0.

This scheme $\mathcal{S}'$ is not a secure digital scheme, although the attack is not very trivial.

Attack: Choose two random $n - bit$ strings m_0, m_1.
Ask for signature of m_0, obtain $\sigma_0 = (\sigma_{01}, \sigma_{02}, \sigma_{03})$
Ask for signature of m_1, obtain $\sigma_1 = (\sigma_{11}, \sigma_{12}, \sigma_{13})$
Now, $m^* = m_1 \oplus \sigma_{11} \oplus \sigma_{01}$ is a message such that $m^* \oplus \sigma_{01} = m_1 \oplus \sigma_{11}$.
Send $m^*, \sigma^* = (\sigma_{01}, \sigma_{02}, \sigma_{13})$ as the forgery.
$Verify'(m^*, \sigma^*) = 1$. Thus, $\mathcal{S}'$ is not secure digital signature scheme. $\blacksquare$