TD 9: IND-CCA Security - Signature (corrected version)

Exercise 1.
Recall the ElGamal public key encryption scheme from the lecture.

- $\mathsf{KeyGen}(1^\lambda)$: Choose a group G with generator g and order $p = O(2^\lambda)$. Sample $x \hookleftarrow U(\mathbb{Z}_p)$ and return:
$$\mathsf{pk} := (G, g, p, g^x) \text{ and } \mathsf{sk} := x.$$
- $\mathsf{Enc}(\mathsf{pk}, m \in G)$: Sample $r \hookleftarrow U(\mathbb{Z}_p)$ and output $(c_1, c_2) = (g^r, (g^x)^r \cdot m)$.
- $\mathsf{Dec}(\mathsf{sk}, c_1, c_2)$: output $m = c_2 \cdot c_1^{-\mathsf{sk}}$.

1. Show that for any $m, m' \in G$, and $(c_1, c_2) := \mathsf{Enc}(\mathsf{pk}, m)$ and $(c_1', c_2') := \mathsf{Enc}(\mathsf{pk}, m')$, it holds that $(c_1 \cdot c_1', c_2 \cdot c_2')$ is a valid ciphertext for $m \cdot m'$. We say that the scheme is homomorphic for multiplication.

   ☞ we have $c_2 \cdot c_2' = (c_1 \cdot c_1')^x \cdot m \cdot m'$. This is a valid encryption for mm', i.e. the decryption algorithm called on this ciphertext returns $m \cdot m'$.

2. Provide a modification of the scheme such that it is now *additively* homomorphic instead of multiplicatively. *Hint: you may want to choose $\mathcal{M} = \{m \in \mathbb{Z}_p, |m| \leq \mathsf{poly}(\lambda)\}$ as your message space.*

   ☞ Instead of encrypting an element of G, we choose to encrypt an element of $\mathcal{M}$. We keep the keygen and change the encryption scheme: $\mathsf{Enc}'(\mathsf{pk}, m) = \mathsf{Enc}(\mathsf{pk}, g^m)$. However, to decrypt, we need to do more: $\mathsf{Dec}'(\mathsf{sk}, c = \mathsf{Enc}'(\mathsf{pk}, m))$ recovers g^m with $\mathsf{Dec}(\mathsf{sk}, c)$. With our choice of message space, it is possible to brute force in polynomial time the Discrete Logarithm Problem, and recover m from g^m.

   With the same trick as in the previous question, we get that our scheme is homomorphic, but this time additively: we get an encryption of $m + m'$, which is still polynomial in λ.

3. Show that the (genuine) ElGamal encryption scheme is not IND-CCA2 secure.

   ☞ Let $m_0 \neq m_1 \in G$. Let $c_1', c_2' = \mathsf{Enc}(\mathsf{pk}, g)$. When given c_1, c_2 encrypting either m_0 or m_1, it is still possible to query the decryption oracle for $(c_1 c_1', c_2 c_2')$ which returns either $m_0 g$ or $m_1 g$, which are different. It is then possible to win the IND-CCA2 security game with probability 1.

   Remark: No homomorphic encryption scheme can be IND-CCA2 secure.

Exercise 2.
We are looking here at different modifications of the Fujisaki-Okamoto (FO) transform that fail at providing CCA2 security. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public-key encryption scheme assumed to be IND-CPA secure with message space $\{0,1\}^{k+\ell}$. We recall the FO transform, where H is a hash function that is modeled as a RO.

$\mathsf{KeyGen}(1^\lambda)$**:** Sample and return $(pk, sk) \leftarrow \mathsf{Gen}(1^\lambda)$.

$\mathsf{Enc}'(pk, m \in \{0,1\}^k)$**:** Sample $r \hookleftarrow U(\{0,1\}^\ell)$ and return $c = \mathsf{Enc}(pk, m||r; H(m||r))$, where $H(m||r)$ is the randomness used by the algorithm.

$\mathsf{Dec}'(sk, c)$**:** Compute $m||r \leftarrow \mathsf{Dec}(sk, c)$ and return m if $c = \mathsf{Enc}(pk, m||r; H(m||r))$. Otherwise, return $\perp$.

1. What happens if $\ell = O(\log(\lambda))$?

   ☞ One can try to guess r: after a challenge phase where the adversary gets $c^\star$ which is an encryption of either m_0 or m_1, sample $r \hookleftarrow U(\{0,1\}^\ell)$ and compute $c_0 = \mathsf{Enc}(pk, m_0||r, H(m_0||r))$ and $c_1 = \mathsf{Enc}(pk, m_1||r, H(m_1||r))$. If there exists $b \in \{0,1\}$ such that $c_b = c^\star$, return it. Otherwise return a uniform bit.

   The advantage of this adversary is equal to the probability of guessing the right r which happens with probability $O(1/\lambda)$, which is non-negligible.

2. Show that there exists an IND-CPA secure encryption scheme such that if we always return m in the decryption algorithm, without checking the consistency of the randomness used in the ecnryption, then its FO transform is not IND-CCA2 secure.

☞ We can for instance use the ElGamal encryption scheme. In this case, when we get $c^* = (g^{H(m_b||r)}, pk^{H(m_b||r)} \cdot (m_b||r))$ we can query the decryption algorithm for $g \cdot g^{H(m_b||r)}, pk \cdot pk^{H(m_b||r)} \cdot (m_b||r)$, which will return m_b, letting us win the game with probability 1.

Exercise 3.
In this exercise we show a scheme that can be proven secure in the random oracle model, but is insecure when the random oracle model is instantiated with SHA-3 (or any fixed (unkeyed) hash function $H : \{0,1\}^* \to \{0,1\}^n$). Let Π be a signature scheme that is euCMA-secure in the standard model.
Let $y \in \{0,1\}^n$ and define the following signature scheme Π_y. The signing and verifying keys are obtained by running $\Pi.\mathsf{Gen}(1^\lambda)$. Signature of a message m is computed out as follows: if $H(0) = y$ then output the secret key, if $H(0) \neq y$ then return a signature computed using Π.Sign. To verify a message, if $y = H(0)$ then accept any signature for any message and otherwise, verify it using Π.Verify.

1. Prove that for any value y, the scheme Π_y is euCMA-secure in the random oracle model.

☞ In the ROM, we can reduce the security of Π_y from the security of Π, as the event $y = H(0)$ happens with negligible probability ($< 2^{-\lambda}$).

Let us assume that there exists an adversary $\mathcal{A}$ that breaks the euCMA security of Π_y in the ROM. We build the following reduction $\mathcal{B}_y$ that on input a verification key vk does the following. It queries $H(0)$. If $H(0) = y$, it aborts. Otherwise, it forwards vk to $\mathcal{A}$ and uses its own signing oracle to sign the messages queried by $\mathcal{A}$. When $\mathcal{A}$ outputs a forgery, it forwards it. We then have:

$$\mathsf{Adv}(\mathcal{B}) = \Pr(\mathcal{A} \text{ wins} \wedge H(0) \neq y).$$

Moreover, it holds that

$$\mathsf{Adv}(\mathcal{A}) \leq \Pr(\mathcal{A} \text{ wins} \wedge H(0) \neq y) + \frac{1}{2^n}.$$

Then $\mathsf{Adv}(\mathcal{B}) \geq \mathsf{Adv}(\mathcal{A}) - 1/2^n$, which is non-negligible.

2. Show that there exists a particular y for which Π_y is insecure when the hash function is not modeled as a random oracle anymore.

☞ Let H be fixed. We look at $\Pi_{H(0)}$. This signature scheme always output its secret key as signature and moreover it accepts any signature for any message. It is then not euCMA-secure.