MTAT.07.003 Cryptology II
Spring 2008 / Homework 4

1. Recall that the message space of the ElGamal cryptosystem is a (t, ε_1)-DDH group $\mathbb{G}$. The latter is rather limiting, since normally one needs to encrypt n-bit messages and not the group elements. The simplified Elgamal cryptosystem is defined as follows:

   - Gen returns $\mathsf{sk} = x$ and $\mathsf{pk} = y = g^x$ for $x \xleftarrow{u} \mathbb{Z}_{|\mathbb{G}|}$;
   - $\mathsf{Enc}_{\mathsf{pk}}(m) = (g^k, h(y^k) \oplus m)$;
   - $\mathsf{Dec}_{\mathsf{sk}}(c_1, c_2) = c_2 \oplus h(c_1^x)$;

   where $h : \mathbb{G} \to \{0,1\}^\ell$ is a almost regular hash function. That is, the distribution $h(y)$ for $y \xleftarrow{u} \mathbb{G}$ is statistically ε_2-close to the uniform distribution over $\{0,1\}^n$. Prove that the simplified ElGamal cryptosystem is also IND-CPA secure and give the corresponding security bounds.

   Hint: Mofify the security proof for the ElGamal cryptosystem to accomodate the change. Where do you need almost regularity?

   ($\star$) In practice, it is difficult if not impossible to define almost regular hash function $h : \mathbb{G} \to \{0,1\}^n$. Relax the security requirements even further so that the corresponding construction is also practical.

2. Let $(\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ be a public key cryptosystem and $\mathsf{Gen}^\circ, \mathsf{Enc}^\circ, \mathsf{Dec}^\circ)$ a symmetric key cryptosystem. Then we can define a hybrid cryptosystem.

   - **Key generation.** Run the key generation algorithm Gen and output the corresponding secret and public key pair $(\mathsf{sk}, \mathsf{pk})$.
   - **Encryption.** Given a message m, generate a session key $\mathsf{sk}^\circ \leftarrow \mathsf{Gen}^\circ$ and output a pair $c_1 \leftarrow \mathsf{Enc}_{\mathsf{pk}}(\mathsf{sk}_\circ)$ and $c_2 \leftarrow \mathsf{Enc}^\circ_{\mathsf{sk}^\circ}(m)$.
   - **Decryption.** To decrypt a ciphertext (c_1, c_2), first reconstruct the session key $\mathsf{sk}^\circ \leftarrow \mathsf{Dec}_{\mathsf{sk}}(c_1)$ and then recover $m \leftarrow \mathsf{Dec}^\circ_{\mathsf{sk}^\circ}(c_2)$.

   Prove the following facts about the hybrid encryption scheme.

   (a) Hybrid encryption scheme is functional.

   (b) If both cryptosystems are IND-CPA secure, then the hybrid encryption scheme is IND-CPA secure. Derive corresponding security guarantees.

   (c) If both cryptosystems are IND-CCA1 secure then the hybrid encryption scheme is IND-CC1 secure. Derive corresponding security guarantees. What about IND-CCA2 security?

   (d) Can one represent the ElGamal and the Goldwasser-Micali cryptosystems as hybrid encryption schemes or not?

3. A cryptosystem is homomorphic if there exists an efficient multiplication operation defined over the ciphertext space $\mathcal{C}$ such that for any valid encryption $c_1 \leftarrow \mathsf{Enc}_{\mathsf{pk}}(m_1)$ the distribution $c_1 \cdot \mathsf{Enc}_{\mathsf{pk}}(m_2)$ coincides with the distribution $\mathsf{Enc}_{\mathsf{pk}}(m_1 \otimes m_2)$, where $\otimes$ is a binary operation defined over the message space $\mathcal{M}$. Show that

   (a) the RSA cryptosystem is multiplicatively homomorphic;

   (b) the ElGamal cryptosystem is multiplicatively homomorphic;

   (c) the Goldwasser-Micali cryptosystem is XOR homomorphic;

4. Prove the following claims about public key cryptosystems

   (a) A homomorphic cryptosystem cannot be non-malleable.

   (b) NM-CPA security implies IND-CPA security.

   (c) NM-CCA1 security implies IND-CCA1 security.

   (d) NM-CCA2 security implies IND-CCA2 security.

($\star$) Show as many separations among the security properties of cryptosystem as you can. For example, show that there are IND-CPA secure cryptosystems that are not IND-CCA1 secure.

5. We can convert a pseudorandom permutation into a pseudorandom generator by using it in the counter mode and output $f(0)\|f(1)\|\ldots\|f(n)$. Alternatively, we can use the following iterative scheme

$$c_1 \leftarrow f(0), c_2 \leftarrow f(c_1), \ldots, c_n \leftarrow f(c_{n-1}) \enspace ,$$

   where $c_1, \ldots, c_n$ is the corresponding output. Compare the corresponding security guarantees. Which of them is better?

6. Feistel cipher $\textsc{Feistel}_{f_1,\ldots,f_k} : \{0,1\}^{2n} \rightarrow \{0,1\}^{2n}$ is a classical block cipher construction that consists of many rounds. In the beginning of the first round, the input x is split into two halves such that $L_0\|R_0 = x$. Next, each round uses a random function $f_i \leftarrow \mathcal{F}_{\mathrm{all}}$ to update both halves:

$$L_{i+1} \leftarrow R_i \qquad \text{and} \qquad R_{i+1} \leftarrow L_i \oplus f_i(R_i) \enspace .$$

   The output of the Feistel cipher $\textsc{Feistel}_{f_1,\ldots,f_k}(L_0\|R_0) = L_k\|R_k$.

   (a) Show that the Feistel cipher is indeed a permutation.

   (b) Show that the two-round Feistel cipher $\textsc{Feistel}_{f_1,f_2}(L_0\|R_0)$ where $f_1, f_2 \leftarrow \mathcal{F}_{\mathrm{all}}$ is not a pseudorandom permutation. Give a corresponding distinguisher that uses two encryption queries.

   (c) Show the three-round Feistel cipher $\textsc{Feistel}_{f_1,f_2,f_3}(L_0\|R_0)$ where $f_1, f_2, f_3 \leftarrow \mathcal{F}_{\mathrm{all}}$ is a pseudorandom permutation. For the proof, note that the output of the three round Feistel cipher can be replaced with uniform distribution if f_2 and f_3 are always evaluated at distinct inputs. Estimate the probability that the ith encryption query creates the corresponding input collision for f_2. Estimate the probability that the ith encryption query creates an input collision for f_3.

($\star$) Show that the tree-round Feistel cipher $\textsc{Feistel}_{f_1,f_2,f_3}(L_0\|R_0)$ is not pseudorandom if the adversary can also make decryption queries.

($\star$) Show that the four-round Feistel cipher $\textsc{Feistel}_{f_1,f_2,f_3,f_4}(L_0\|R_0)$ where $f_1, f_2, f_3, f_4 \leftarrow \mathcal{F}_{\text{all}}$ is indistinguishable from $\mathcal{F}_{\text{prm}}$ even if the adversary can make also decryption calls.

($\star$) The counter mode converts any pseudorandom function into a pseudorandom generator. Give a converse construction that converts any pseudorandom generator into a pseudorandom function. Give the corresponding security proof together with precise security guarantees.

Hint: Use a stretching function $f : \{0,1\}^n \to \{0,1\}^{2n}$ to fill a complete binary tree with n-bit values.