Problem 3-1 (Lossy Encryption) 23 (+10) Points

Let \((KG, E, D)\) be a public key encryption scheme. In this problem we define a new property for PKE schemes that we call “lossy encryption”. We say that a scheme \((KG, E, D)\) is lossy if there exists an algorithm \(LossyKG(1^k)\) which generates a “lossy” public key \(PK'\) such that the following two properties are satisfied:

1. A lossy public key is computationally indistinguishable from a public key generated by \(KG\): \(PK \approx PK'\). More formally, for any PPT adversary \(A\) it holds:
   \[
   |\Pr[A(PK) = 1 | (PK, SK) \leftarrow KG(1^k)] \triangleright \Pr[A(PK') = 1 | PK' \leftarrow LossyKG(1^k)]| \leq \text{negl}(k)
   \]

2. For any lossy public key \(PK' \leftarrow LossyKG(1^k)\), encrypting any message using \(PK'\) produces ciphertexts that have identical distribution. Namely, for any \(PK' \leftarrow LossyKG(1^k)\), and any pair of messages \(m_0, m_1 \in \mathcal{M}\), we have \((PK', E(PK', m_0)) \equiv (PK', E(PK', m_1))\).

   Intuitively, notice that this second property is telling that encrypting using the lossy public key completely loses information about the original plaintext, and thus it is not possible to decrypt.

Part I.

(a) (5 points) In this part you are to prove that if an encryption scheme is lossy according to the definition provided above, then the scheme is also CPA-secure.

   (Hint: Using hybrid arguments you can define an intermediate distribution in which you replace the real public key with a lossy one. What is the advantage of the adversary in this experiment?)

   Solution: **************** INSERT PROBLEM 1a SOLUTION HERE ****************

Part II.

Consider the following scheme as a potential candidate for being a lossy public key encryption. \(KG(1^k)\) chooses a random \(k\)-bit large safe prime \(p\) (i.e., \(p = 2q + 1\) for a large prime \(q\)) and chooses two random generators \(g_0, g_1\) of \(G = QR_p\) (recall that \(QR_p\) is the subgroup of quadratic residues in...
Next, it chooses two random (but distinct) values $x_0, x_1 \in \mathbb{Z}_q$, computes $h_0 = g_0^{x_0}$, $h_1 = g_1^{x_1}$, and outputs $PK = (p, g_0, g_1, h_0, h_1)$ and $SK = (x_0, x_1)$.

To encrypt a 1-bit message $m \in \{0, 1\}$, $E(PK, m)$ proceeds as follows: choose a random $r \in \mathbb{Z}_q$ and output $C = (g_r^m, h_r^m)$.

(b) (3 points) First, show that any user who knows the secret key $SK$ can decrypt, i.e., describe a decryption algorithm $D(SK, C)$ which uses $SK$ to recover $m$ from $C$.

Solution: ****************** INSERT PROBLEM 1b SOLUTION HERE ******************

(c) (10 points) Second, prove that the scheme described above (together with the decryption algorithm that you obtained from part (b)) is a lossy public key encryption based on the DDH assumption. Namely, first describe a lossy key generation algorithm $LossyKG(1^k)$ and then show that it satisfies both properties (1) and (2).

(Hint: Write $g_1 = g_0^\alpha$ for some $\alpha \in \mathbb{Z}_q$, and notice that $PK = (g_0, g_1, h_0, h_1)$ (as generated by $KG$) is a “random” non-DDH tuple $(g_0, g_0^\alpha, g_0^{x_0}, g_0^{\alpha x_1})$ where $h_1 = g_0^{\alpha x_1}$ is completely independent from the prior three elements. What could thus be a natural candidate for a lossy public key?)

Solution: ****************** INSERT PROBLEM 1c SOLUTION HERE ******************

(d) (5 points) By combining part (b) and (c) you showed that the scheme described above is CPA-secure. Although the lossy property may be nice and useful in some contexts, this is not necessary to prove that the scheme is CPA-secure. In fact, in this part you are to prove directly that this scheme is CPA-secure under the DDH assumption; namely,

$$(g_0, g_1, h_0, h_1, g_0^r, h_0^r) \approx (g_0, g_1, h_0, h_1, g_1^r, h_1^r)$$

(Hint: On each side define an intermediate distribution using DDH assumption, and then argue the intermediate distributions are identical.)

Solution: ****************** INSERT PROBLEM 1d SOLUTION HERE ******************

Part III (Extra credit)

(e) (Extra credit: 10 points) Consider the following variant of the Goldwasser-Micali encryption scheme. The public key is $PK = (n, y)$ where $n = pq$ is the product of two large primes $p$ and $q$, and $y$ is a random quadratic non-residue $y \in J_n \setminus QR_n$ with Jacobi symbol 1. The secret key is the factorization of $n$, $SK = (p, q)$.

To encrypt a 1-bit message $m \in \{0, 1\}$, $E(PK, m)$ samples a random $r \in \mathbb{Z}_n^*$ and outputs $c = y^{mn}r^2 \mod n$. Basically, observe that if $m = 0$, then $c$ is a random quadratic residue; otherwise, if $m = 1$, then $c$ is a random quadratic non-residue.

PS3-2
First, describe a decryption algorithm for this scheme.

Second, prove that this scheme is a lossy encryption scheme based on the quadratic residuosity assumption (see previous homework), i.e., describe a lossy key generation algorithm which satisfies properties (1) and (2).

(Hint: How can you change the distribution of the public key in order to loose information about \(m\)?)

Solution: ****************** INSERT PROBLEM 1e SOLUTION HERE ******************

Problem 3-2 (Encryption Candidates) 15 Points

Assume that you have a one-bit message \(m \in \{0,1\}\) that you want to encrypt in a probabilistic way. For each of the following methods, (1) describe the decryption algorithm and determine if it is efficient (i.e., eventually determine what is the secret key); (2) determine if the encryption of 0 is indistinguishable from the encryption of 1; and (3) conclude whether the method really defines a polynomially indistinguishable encryption scheme. Be sure to justify your answers.

(a) (2 points) Let \(PK = (p, g)\), where \(p\) is a random \(k\)-bit large prime and \(g\) is a generator \(\mathbb{Z}_p^*\). Choose at random \(x \in \mathbb{Z}_{p-1}\) such that \(\text{LSB}(x) = m\) (\(\text{LSB}(x)\) is the least significant bit of \(x\)). The ciphertext is \(y = g^x \mod p\).

Solution: ****************** INSERT PROBLEM 2a SOLUTION HERE ******************

(b) (3 points) The same as above, except that you use \(x\) such that \(\text{MSB}(x) = m\) (\(\text{MSB}(x)\) is the modified most significant bit of \(x\) which is 1 only if \(x \geq (p-1)/2\)?

Solution: ****************** INSERT PROBLEM 2b SOLUTION HERE ******************

(c) (4 points) Let \(PK = (n, e)\), where \(n\) is a random RSA modulus and \(e\) is a random exponent in \(\mathbb{Z}_\phi(n)^*\). Choose at random \(x \in \mathbb{Z}_n^*\) such that \(\text{LSB}(x) = m\). The ciphertext is \(y = x^e \mod n\).

Solution: ****************** INSERT PROBLEM 2c SOLUTION HERE ******************

(d) (6 points) Let \(PK = (p, g, h)\), where \(p\) is a random \(k\)-bit large safe prime (i.e., \(p = 2q + 1\) for a large prime \(q\)), \(g\) is a generator of \(G = QR_p\) (the subgroup of quadratic residues in \(\mathbb{Z}_p^*\)), and \(h = g^x \mod p\), for \(x\) chosen at random in \(\mathbb{Z}_q\). To encrypt a message \(m \in \{0,1\}\), compute the ciphertext \(C = (g^r \mod p, h^{r+ms} \mod p)\) choosing \(r, s \in \mathbb{Z}_q\) uniformly at random.

Solution: ****************** INSERT PROBLEM 2d SOLUTION HERE ******************
Problem 3-3 (Insecurities of Plain RSA)  12 Points

In this problem you will see some dangers of a deterministic encryption like RSA. It turns out that none of these problems exists if one uses instead a probabilistic CPA-secure encryption. Outside this problem, try to think why this is the case.

(a) (4 points) Assume that Alice and Bob belong to the same organization, and that they have RSA keys with the same public modulus \( n \), but with different public exponents \( e_A, e_B \) respectively, where \( e_A, e_B \) are relatively prime. Assume further that the usual RSA encryption is used to send the same message \( m \) to both Alice and Bob.

Prove that an eavesdropper who knows \( n, e_A, e_B \) and sees \( c_A = m^{e_A} \mod n, c_B = m^{e_B} \mod n \), can reconstruct the message \( m \).

(Hint: Recall that if \( a, b \) are integers such that \( \gcd(a, b) = 1 \) then the extended Euclid’s algorithm can be used to compute integers \( \alpha, \beta \) for which \( \alpha a + \beta b = 1 \).)

Solution: **************** INSERT PROBLEM 3a SOLUTION HERE ************

(b) (4 points) Assume that Alice, Bob and Carol use three different and relatively prime public RSA moduli \( n_A, n_B, n_C \) respectively, but all with the same public exponent \( e = 3 \) (in particular, 3 is relatively prime to \( \varphi(n_A), \varphi(n_B), \varphi(n_C) \)). Again, assume that the usual RSA encryption is used to send the same message \( m \) to Alice, Bob and Carol.

Prove that an eavesdropper who knows \( n_A, n_B, n_C \) and sees \( c_A = m^3 \mod n_A, c_B = m^3 \mod n_B, c_C = m^3 \mod n_C \), can reconstruct the message \( m \).

(Hint: Apply Chinese remainder theorem. Also, think is the RSA problem is easy over the integers.)

Solution: **************** INSERT PROBLEM 3b SOLUTION HERE ************

(c) (4 points) Assume that Alice uses RSA with modulus \( n \) and public exponent 3. Assume Bob sends Alice encryptions of \( m, m + 1 \) and \( m + 2 \): \( c_0 = m^3 \mod n, c_1 = (m + 1)^3 \mod n, c_2 = (m + 2)^3 \mod n \). Show that the eavesdropper, who knows \( c_0, c_1, c_2 \) and the fact that 3 consecutive messages were encrypted (but does not know \( m \)), can recover \( m \).

(Hint: Write a system of 3 cubic equations with unknowns \( m^3 \mod n, m^2 \mod n \) and \( m \mod n \) and try to solve for \( m \).)

Solution: **************** INSERT PROBLEM 3c SOLUTION HERE ************

Problem 3-4 (Encrypting the Password)  11 Points

Consider the following simplified authentication application, which is the basis of many real programs like SSH. The user \( U \) has a secret password \( pw \) which he shares with the server \( S \). For
simplicity, assume $pw$ is chosen at random from some known dictionary $D$. To authenticate, the user enters some candidate password $pw'$ and the server lets the user in if and only if $pw' = pw$. The problem is that the user and the server are located at remote places, connected only by a public link, so that the user cannot send his password $pw$ in the clear. Luckily, the user sits at the remote terminal $T$, which knows the encryption public key $PK$ of the server $S$. Therefore, when the user enters $pw'$ to the terminal, the terminal computes $c \leftarrow E_{PK}(pw')$, and send $c$ over the public channel. The server then decrypts $c$, gets $pw'$, and checks that $pw' = pw$.

Consider now the adversary $A$ who observes $c$ (equal to $E_{PK}(pw)$), and later comes to the remote terminal himself trying to guess $pw$. In this problem you will argue that if $E$ is CPA-secure, then the attacker has only negligible probability in winning this game. Specifically, he can win this game with probability at most $\epsilon' \leq \epsilon + \frac{1}{|D|}$, where $\epsilon$ is the probability of breaking the CPA-security of $E$. You will show the bound using both definitions of semantic security and indistinguishability. By comparing your proofs, you will see that the semantic security view saves you the entire argument needed in part (b)!

(a) (3 points) Show that $\epsilon' \leq \epsilon + \frac{1}{|D|}$ using the definition of semantic security (which we know is equivalent to indistinguishability). The correct answer should follow very easily from the definition of semantic security.

(Hint: Think what is the chance the simulator can win the game?)

**Solution:** ****************** INSERT PROBLEM 4a SOLUTION HERE ******************

(b) (4 points) Now use the actual definition of CPA-security as given in class (see Problem 7). First consider a hybrid experiment, where the attacker sees $c \leftarrow E_{PK}(0)$ instead of the “real” $E_{PK}(pw)$. Formally argue that the adversary’s chance of winning this new game cannot decrease by more than $\epsilon$.

**Solution:** ****************** INSERT PROBLEM 4b SOLUTION HERE ******************

(c) (4 points) Now consider the modified hybrid game in part (b). Argue that the adversary’s chance of winning this game is at most $1/|D|$. Conclude from this and part (b) the desired bound on $\epsilon'$.

**Solution:** ****************** INSERT PROBLEM 4c SOLUTION HERE ******************