Advanced Topics in Cryptography - Homework 1.

Review of crypto topics. Here's a quick review of some topics typically taught in a basic crypto class. You can also find them in various textbooks and online lecture notes (including by current teachers). They are also covered in a condensed way in the chapter on cryptography in the complexity textbook by Arora and Barak (see online draft on `http://www.cs.princeton.edu/theory/index.php/Compbook/Draft#crypto`).

Two distributions X, Y ranging over $\{0,1\}^k$ are *ϵ-statistically indistinguishable* if for every function $f : \{0,1\}^k \to \{0,1\}$ it holds that:

$$= |\Pr[f(X) = 1] - \Pr[f(Y) = 1]| < \epsilon \tag{1}$$

we say that they are (ϵ, T)-*computationally indistinguishable* if () holds only with respect to functions T that can be computed by Boolean circuits of size at most T. We say that X, Y are simply computationally indistinguishable / statistically indistinguishable if this holds for functions $\epsilon = \epsilon(k)$, $T = T(k)$ such that ϵ goes to zero and T goes to infinity faster than every polynomial in k. For the next several classes, you can just think of $\epsilon = 2^{-k^{0.001}}$ and $T(k) = 2^{k^{0.001}}$.

An *encryption scheme* is a tuple of three probabilistic polynomial time (p.p.t) algorithms (G, E, D) for key generation, encryption and decryption. In a *private key* scheme the generator G outputs a single key sk used for both encryption and decryption, while in a *public key* scheme the generator outputs a pair of keys (sk, pk) where pk is used for encryption and sk is used for decryption. We require the natural validity condition that for any message m, $D_{sk}(E_{pk}(m)) = m$.

In this exercise we restrict ourselves to encryption whose messages are single bits. A private key encryption scheme is *semantically secure* if it satisfies that the distributions $E_{sk}(0)$ and $E_{sk}(1)$ are computationally indistinguishable. A public key scheme is semantically secure if $(pk, E_{pk}(0))$ and $(pk, E_{pk}(1))$ are computationally indistinguishable, where $(pk, E_{pk}(b))$ denotes the distribution on pairs obtained by running $G(1^k)$ to get (pk, sk) and then running $E_{pk}(b)$. (This is equivalent to the definition given in class - check it!.)

Definition of homomorphic encryption. We say that a quadruple of p.p.t algorithms $(G, E, D, EVAL)$ is a homomorphic encryption scheme with respect to a class of circuits $\mathcal{C}$, if (G, E, D) is semantically secure and in addition for every circuit $C \in \mathcal{C}$ taking t bits as input, the algorithm $EVAL$ satisfies:

Strong homomorphism For every $m_1, \ldots, m_t \in \{0,1\}$ and $c_1, \ldots, c_t$ output by $E_{pk}(m_1), \ldots, E_{pk}(m_t)$ respectively, the distributions $EVAL_{pk}(C, c_1, \ldots, c_t)$ and $E_{pk}(C(m_1, \ldots, m_t)$ are statistically indistinguishable.

Weak homomorphism For every $m_1, \ldots, m_t \in \{0,1\}$ and $c_1, \ldots, c_t$ output by $E_{pk}(m_1), \ldots, E_{pk}(m_t)$ respectively it holds that **(1)** $D_{sk}(EVAL_{pk}(C, c_1, \ldots, c_t)) = C(m_1, \ldots, m_t)$ with probability $1 - negl(k)$ (where $negl(k)$ denotes a function tending to zero faster than any polynomial in k), and **(2)** the output of $EVAL$ on any input is of length at most k^2. (k^2 can be replaced here with any fixed polynomial.)

For private key encryption the definition is the same except that $EVAL$ obviously doesn't get the public key as input (and neither the private key).

Exercise 1. Prove that if a scheme is strongly homomorphic w.r.t. $\mathcal{C}$ then it is also weakly homomorphic. See footnote for hint[1]

The next exercises cover Ron Rothblum's transformation of a private key encryption $(G, E, D, EVAL)$ that is weakly homomorphic w.r.t XOR into a public key encryption (G', E', D'). Recall that the construction is as follows: G' runs G to obtain sk, chooses $r \leftarrow_{\mathrm{R}} \{0,1\}^{\ell}$, and $(c_1, \ldots, c_\ell)$ where $c_i = E_{sk}(r_i)$ for $i = 1..\ell$ and $\ell = k^4$. The public key is $r, c_1, \ldots, c_\ell$. To encrypt the message $m \in \{0,1\}$, the algorithm E' chooses a random $s \in \{0,1\}^{\ell}$ subject to $\sum_i r_i s_i = m \pmod 2$ and outputs $EVAL(XOR, \{c_i\}_{i:s_i=1})$. Decryption uses the same algorithm.

The big question is whether this is semantically secure. This is proven via the following two exercises: (you should verify that you understand why they do indeed suffice!)

Exercise 2. Suppose that there is a polynomial time algorithm A that can distinguish $(r, c_1, \ldots, c_\ell, E'(0))$ from $(r, c_1, \ldots, c_\ell, E'(1))$ with non-negligible success. Then A can also distinguish $(r, \tilde{c}_1, \ldots, \tilde{c}_\ell, E'(0))$ from $(r, \tilde{c}_1, \ldots, \tilde{c}_\ell, E'(1))$ where $\tilde{c}_i$ is obtained by running $E_{sk}(0)$ (instead of $E_{sk}(r_i)$. See footnote for hint[2]

The harder part is the following:

Exercise 3. Let A be any function (possibly not in polynomial time) then

$$\left|\Pr[A(r, \tilde{c}_1, \ldots, \tilde{c}_\ell, E'(0)) = 1] - \Pr[A(r, \tilde{c}_1, \ldots, \tilde{c}_\ell, E'(1)) = 1]\right| < 100 * 2^{-k} \tag{2}$$

We now give some guidance how to solve Exercise 3. It will follow from the stronger statement that (2) holds even after fixing typical values for all randomness used for generating sk, $\tilde{c}_1, \ldots, \tilde{c}_\ell$ and hence the only randomness involved in the probabilities in (2) are r, s. The main claim to prove is the following:

Claim: Let y be a string of size at most k^2, and let S_y denote the set of strings $s \in \{0,1\}^{\ell}$ such that $EVAL(XOR, \{c_i\}_{i:s_i=1}) = y$, and $d = \lfloor \log |S_y| \rfloor$. Then with probability $1 - 2^{-k}$ over the choice of $r \in \{0,1\}^{\ell}$,

$$\frac{1}{2} - 2^{-d/10} \leq \Pr_{s \in S_y}\left[\sum_{i=1}^{\ell} s_i r_i \pmod 2\right] \leq \frac{1}{2} + 2^{-d/10} \tag{3}$$

Exercise 4. Prove that Exercise 3 follows from the claim via the following steps:

1. Prove that with probability at least $1 - 2^{2^{-k}}$ it holds that $d \gg 100k$. See footnote for hint.[3]
2. Now you can argue that with high probability the pair (r, y) where $y = EVAL(XOR, \{c_i\}_{i:s_i=1})$ doesn't reveal any information on the message bit m, and hence even an attacker with unbounded computation time cannot guess m from it.

[1] **Hint:** If two distributions are statistically indistinguishable then the decryption algorithm will output the same answer on them, and also their lengths will be the same, except with negligible error.

[2] **Hint:** This is a fairly standard cryptographic reduction to the semantic security of the scheme (G, E, D). One fact we use is that if an encryption scheme is semantically secure then one also cannot distinguish between two ℓ-tuples of encryptions.

[3] **Hint:** This follows from the fact that there are 2^{ℓ} possible strings s but only 2^{k^2+1} possible outputs of $EVAL$.

Exercise 5. Prove the claim. It is an instance of what is known as the *Leftover Hash Lemma*. In this particular case you can prove it as follows: let $S = S_y$ and define for every $s \in S$ the random variable X_s over the choice of $r \in \{0,1\}^\ell$ where $X_s = \sum_{i=1}^{\ell} s_i r_i \pmod 2$.

1. Prove that $E[X_s] = 1/2$ for very $s \neq 0^n$.
2. Prove that $E[X_s X_t] = E[X_s]E[X_t]$ for every $s \neq t$.
3. Prove that for every n, $\Pr_{r\in\{0,1\}^\ell}[|\sum_{s\in S} X_s - |S|/2| > n\sqrt{|S|}] < 100/n^2$. See footnote for hint[4]
4. Prove the claim.

[4] **Hint:** Compute the variance of $\sum_{s\in S} X_s$ and use the Chebychev Inequality.