Homework 4

1 Problem 1 (40 points)

Sometimes, it is useful to consider variants of the LWE problem. Here, we will explore some of these variants.

(a) Consider the variant of LWE where the distribution on the error terms is uniform in $0, 1$. That is, you are given $A, s^T A + e$ where A is a random $n \times m$ matrix over $\mathbb{Z}_q$, s is a random vector in $\mathbb{Z}_q^n$, e is a random vector in $\{0,1\}^m$. The goal is to find s, or at least distinguish from a random A, u.

Show that the search version of this variant of LWE is *insecure*, provided m is sufficiently large. That is, that it is possible to find s

Here are the suggested steps to achieve this goal:

i. Let A, u be a sample. We know that $u = s^T A + e$ for unknown vectors s, e. Show how to use the fact that e is binary to remove the e variables, obtaining quadratic equations in the s variables.

ii. Unfortunately, solving quadratic equations is hard in general. However, here we can use the following trick. In each quadratic equation from (i), expand the equation into a sum of monomials in the $s_{i,j}$ variables. For each quadratic monomial $s_i s_j$, replace $s_i s_j$ with a new variable $t_{i,j}$.

Suppose we forget that $t_{i,j}$ is supposed to equal $s_i s_j$, and instead let it be a free variable. Explain how this new system of equations over the $s_i, t_{i,j}$ variables can be solved.

iii. The solutions obtained above may not correspond to an actual solution to the equations from part (i) (namely, for a given solution, $t_{i,j}$ may not equal $s_i s_j$). Explain why, with enough equations, we nonetheless expect to be able to find a solution such that $t_{i,j} = s_i s_j$, which *does* correspond to a solution to the equations from (i).

This part is allowed to be heuristic, you do not need to formally prove that the solution is unique. However, you should derive an estimate of the number of equations needed, and justify this estimate.

Hint: How many equations should be necessary for the solution to be unique?

(b) Let p, q be integers, and q exponentially larger than p (meaning that q/p is exponential in some security parameter). For simplicity, assume p divides q. For an integer $x \in \mathbb{Z}_q$, let $\lfloor x \rceil_p$ denote the process of rounding x to the nearest multiple of p, and then dividing by p. For for example, if $p = 5$,$q = 1000$, and $x = 63$, $\lfloor x \rceil_p = 13$ (since the nearest multiple of 5 is $65 = 13 \times 5$)

Consider this variant of LWE: choose a random matrix $A \in \mathbb{Z}_q^{n \times m}$, a random vector $s \in \mathbb{Z}_q^n$, and output $A, \lfloor s^T A \rceil_p$.

Show that the search and decision version of this problem are *secure*, provided that q is sufficiently larger than p, assuming the standard LWE assumption holds for an appropriate choice of parameters

(c) Consider yet another variant of LWE, where now the secret s is also sampled from a discrete Gaussian with the same width σ as the error term.

Show that this version of LWE is *secure*, provided that the standard version of LWE is secure for the same choice of n and σ. You may, however, assume standard LWE is hard for a slightly larger choice of m

Hint: Given a sample (A, u) from standard LWE. Let A_0, u_0 be the first n rows of A and u, respectively, and A_1, u_1 be the remaining $m - n$ rows. Suppose the first n columns of A are linearly independent, so that A_0 is full rank. Explain how to modify A_1, u_1 into a new sample A_1', u_1' (using A_0, u_0) such that (A_1', u_1') is an LWE sample where the secret is just the error vector in A_0, u_0 (which is Gaussian, as desired). Explain how to handle random A, where the first n columns of A may not be linearly independent.

2 Problem 2 (25 points)

Suppose you are given an algorithm C that, on input a random matrix $A \in \mathbb{Z}_q^{n \times m}$, finds a binary SIS solution. Namely, C outputs an $x \in \{0,1\}^m$ such that $A \cdot x = 0 \bmod q$. You may assume the algorithm succeeds with overwhelming probability.

Show how to use C to find binary SIS solutions, but for random $B \in \mathbb{Z}_{q^2}^{n \times m^2}$. Your algorithm will run C several times.

Hint: Use C to find many solutions $x_i \in \{0,1\}^{m^2}$ to $B \cdot x_i = 0 \bmod q$. Any linear combination x of the x_i will also satisfy $B \cdot x = 0 \bmod q$. Use C once more to find one such solution x that is still in $\{0,1\}^{m^2}$, but such that $B \cdot x = 0 \bmod q^2$. For this to work, your x_i will need a particular structure

3 Problem 3 (20 points)

Recall the quantum money scheme shown in class. Consider the case of a single banknote. The secret serial number for the note consists of two bit strings $b, c \in \{0,1\}^\lambda$. The banknote consists of λ qubits, where the ith qubit is in the state $|\psi_{b_i,c_i}\rangle$, where:

- $|\psi_{0,c}\rangle = |c\rangle$
- $|\psi_{1,c}\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}(-1)^c|1\rangle$

Suppose the bank also offers a verification oracle for the banknote. That is, the bank, on input a quantum money state $|\phi_1\rangle \dots |\phi_\lambda\rangle$, measures $|\phi_i\rangle$ in the basis B_{b_i}, where $B_b = \{|\psi_{b,0}\rangle, |\psi_{b,1}\rangle\}$. It obtains the output c'_i. If c', the bitstring consisting of all c_i, is identical to c, then the bank accepts. Otherwise, it rejects. Additionally, in either case, it returns whatever is left of the banknote after the measurement.

(a) Show that given such an oracle, and a single valid banknote, it is possible in polynomial time to forge new banknotes in polynomial time (with high probability).

(b) Suggest a fix to the attack in part (a). You do not need to prove the security of your fix, but must provide an informal argument why it blocks the attack from (a)

4 Problem 4 (25 points)

Let $f : [N] \to \{0,1\}$ be a function. Recall that Grover's algorithm lets you find a random solution x to $f(x) = 1$, making only $O(\sqrt{N/r})$ calls to f, where r is the number of solutions. The number of solutions r need not be known.

Suppose now your goal is to find *all* possible solutions. The naive approach is to run Grover's algorithm as above roughly until you hit every solution. This becomes an instance of the coupon collectors problem. This approach will require running Grover as above approximately $r \log r$ times, giving a total time of $O(\sqrt{(Nr)} \log r)$ time.

Show how to remove the extra $\log r$ factor, obtaining an algorithm that runs in time $O(\sqrt{Nr})$, and still finds all solutions. You can assume r is known.

Hint: You will need to run Grover's algorithm on functions other than f.

5 Problem 5 (40 points)

Recall the security definition for a PRF. When we switch to considering quantum adversaries, typically the only thing that would change is that we allow the adversary to have a quantum computer. However, the queries the adversary makes are still classical. Call a PRF secure in this way a post-quantum PRF. It turns out that our construction and analysis of PRFs from one-way functions from the beginning of the course works also for building post-quantum PRFs. Thus we can construct post-quantum PRFs from one-way functions, provided the one-way functions are secure against quantum computers.

A stronger notion of security, however, considers an adversary that can query the PRF on a quantum superposition of inputs. That is, the adversary submits a state $\sum_{x,y} \alpha_{x,y}|x, z\rangle$, and in response gets the state $\sum_{x,y} \alpha_{x,y}|x, y \oplus H(x)\rangle$, where $H(x)$ is either the PRF or a random function. The adversary's task is still to distinguish the two cases. Call such PRFs fully-quantum PRFs. (Note that the PRF itself is still classical, just that it is being evaluated on superpositions of inputs).

(a) Explain why the proof of security for constructing PRFs from PRGs that we saw in class breaks down when trying to prove that the construction is a fully-quantum PRF.

(b) Given a post-quantum PRF $\texttt{PRF}$, devise a new PRF $\texttt{PRF}'$ that is (1) post-quantum secure, but (2) is not fully-quantum secure.