Homework 4

Instructor: Chris Peikert
Student: YOUR NAME HERE

This homework is due by the **start of class on November 23** via the course Canvas page. Start early!

Instructions. Solutions must be typeset in LaTeX (a template for this homework is available on the course web page). Your work will be graded on *correctness*, *clarity*, and *conciseness*. You should only submit work that you believe to be correct; if you cannot solve a problem completely, you will get significantly more partial credit if you clearly identify the gap(s) in your solution. It is good practice to start any long solution with an informal (but accurate) "proof summary" that describes the main idea.

You may collaborate with others on this problem set and consult external sources. However, you must ***write your own solutions*** and ***list your collaborators/sources*** for each problem.

1. Let $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ be a parity-check matrix specifying the q-ary lattice $\mathcal{L}^\perp(\mathbf{A}) = \{\mathbf{z} \in \mathbb{Z}^m : \mathbf{A}\mathbf{z} = \mathbf{0} \in \mathbb{Z}_q^n\}$. If you wish, throughout this problem you may assume that q is prime (though this is not necessary hypothesis for any of the conclusions).
   (a) Describe an efficient algorithm that finds an invertible n-by-n submatrix of $\mathbf{A}$ which is invertible over $\mathbb{Z}_q$, if one exists. (For uniformly random $\mathbf{A}$ and typically used m, it can be shown that such a submatrix exists with high probability.) Also argue that this invertible submatrix can be moved to the first n columns of $\mathbf{A}$, without essentially changing the lattice.
   (b) Prove that the invertible submatrix can be replaced by the identity matrix $\mathbf{I}_n$, possibly changing the rest of $\mathbf{A}$ as well, without changing the lattice.
   (c) Using the previous parts, describe how to efficiently compute a basis of $\mathcal{L}^\perp(\mathbf{A})$.
   Hint: if $\mathbf{A} = [\mathbf{I}_n \mid \bar{\mathbf{A}}]$, then the n columns of $\begin{bmatrix} q\mathbf{I}_n \\ \mathbf{0} \end{bmatrix}$ are vectors in $\mathcal{L}^\perp(\mathbf{A})$. Find $m - n$ more columns, and prove that all m columns together form a basis $\mathbf{B}$ of $\mathcal{L}^\perp(\mathbf{A})$, i.e., $\mathbf{B} \cdot \mathbb{Z}^m = \mathcal{L}^\perp(\mathbf{A})$.
   (d) Recall that the SIS problem is to find a short nonzero solution to $\mathbf{A}\mathbf{z} = \mathbf{0}$ for uniformly random $\mathbf{A}$. Using the previous parts, prove that the following problem is at leas as hard as SIS: given uniformly random $\mathbf{A}'$, find a short nonzero solution to $\mathbf{A}'\mathbf{z} = \mathbf{e} \bmod q$, where $\mathbf{e} \in \mathbb{Z}^n$ is *any* short vector of the attacker's choice. *Hint:* the number of columns need not be the same in $\mathbf{A}$ and $\mathbf{A}'$.

2. Assume that decision-LWE is hard for some parameters n, q and error distribution χ. That is, pairs of the form
$$(\mathbf{a}_i, b_i = \langle \mathbf{s}, \mathbf{a}_i \rangle + e_i),$$
where $\mathbf{s} \leftarrow \mathbb{Z}_q^n$ is chosen once and for all and all the $\mathbf{a}_i \leftarrow \mathbb{Z}_q^n$ and $e_i \leftarrow \chi$ are independent, are pseudorandom (i.e., indistinguishable from uniformly random).
   Prove that the "multi-secret" form of decision-LWE is also hard, i.e., for any polynomially bounded t, tuples of the following form are pseudorandom:
$$(\mathbf{a}_i, b_{i,1} = \langle \mathbf{s}_1, \mathbf{a} \rangle + e_{i,1}, \ldots, b_{i,t} = \langle \mathbf{s}_t, \mathbf{a} \rangle + e_{i,t}) \in \mathbb{Z}_q^n \times \mathbb{Z}_q^t,$$
where each $\mathbf{s}_j \leftarrow \mathbb{Z}_q^n$ is chosen once and for all, and the $\mathbf{a}_i \leftarrow \mathbb{Z}_q^n$ and $e_{i,j} \leftarrow \chi$ are independent.
   Hint: use a "hybrid" argument: define a sequence of $t + 1$ distributions, where the first is the multi-secret LWE distribution and the last is uniformly random. Show that distinguishing any pair of adjacent distributions in the sequence is as hard as decision-LWE.

3. The public-key cryptosystems described in class encrypts a single message bit using a ciphertext of $n \log q$ bits or more, and a public key of $n^2 \log q$ bits or more. This is quite a large overhead! Extend at least one of those systems to securely encrypt (say) n message bits while preserving the ciphertext and public key sizes, up to small constant factors. *Hint:* use the result from the previous question.