CS 395T: Topics in Cryptography **Spring 2024**

Exercise Set 6

Due: April 17, 2024 at 11:59pm (Submit on Gradescope) **Instructor:** David Wu

Instructions. You **must** typeset your solution in LaTeX using the provided template:

https://www.cs.utexas.edu/~dwu4/courses/sp24/static/homework.tex

You must submit your problem set via Gradescope (accessible through Canvas).

Collaboration Policy. You may discuss your general approach with other students, but you may not share written documents. You should not search online for solutions to these problems. If you do consult external sources, you must cite them in your submission. You must include the names of all of your collaborators with your submission. Refer to the official course policies for the full details.

Problem 1: Regev Encryption [20 points]. In lecture, we described Regev encryption in the setting where the message is encoded in the *most significant bits* of the ciphertext. Here, we will consider a variant where the message is encoded in the *least significant bits* of the ciphertext. For simplicity, we will just consider the symmetric setting (but everything generalizes to the public-key setting in the manner described in lecture). Let the message space be $\mathbb{Z}_p$, n be the lattice dimension, q be the modulus, and χ be the error distribution. Suppose that $\gcd(p, q) = 1$. Note that p and q need *not* be prime here.

- The secret key is a vector $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$.
- To encrypt a message $\mu \in \mathbb{Z}_p$, sample $\mathbf{a} \xleftarrow{R} \mathbb{Z}_q^n$ and $e \leftarrow \chi$, and output the ciphertext $\mathsf{ct} = (\mathbf{a}, \mathbf{s}^\top \mathbf{a} + pe + \mu)$.

(a) Given a ciphertext ct and the secret key $\mathbf{s}$, describe the decryption algorithm. Prove correctness of the encryption scheme with your choice of decryption algorithm. You may assume that $\Pr[e \leftarrow \chi : |e| < q/(2p) - 1] = 1$.

(b) Show that under the $\mathsf{LWE}_{n,m,q,\chi}$ assumption, the above encryption scheme is CPA-secure against an adversary that makes at most m encryption queries. **Hint:** It suffices to show that the ciphertexts in this scheme are pseudorandom (computationally indistinguishable from a random string). You can use without proof that an encryption scheme with pseudorandom ciphertexts is CPA-secure.

This version of Regev encryption can also be extended to obtain an FHE scheme. One advantage of this construction over the GSW construction described in class is that the ciphertexts are vectors rather than matrices. The challenge problem will walk through how to derive this alternative approach to constructing FHE.

Optional Feedback. Please answer the following *optional* questions to help design future exercise sets. You do not need to answer these questions. However, we do encourage you to provide us feedback on how to improve the course experience.

(a) How long did you spend on this exercise set?

(b) Do you have any feedback for this exercise set?

(c) Do you have any feedback on the course so far?

(d) Are there specific topics that you are interested in seeing in this course?

Challenge Problem: Key Switching in Regev Encryption [Optional]. Let n be the lattice dimension, q be an *odd* modulus, and χ be a B-bounded distribution over $\mathbb{Z}_q$ (where $B = \text{poly}(n, \log q)$). Let $\mathbf{s}^\mathsf{T} = [-\tilde{\mathbf{s}}^\mathsf{T} \mid 1] \in \mathbb{Z}_q^{n+1}$ be a secret key for a Regev encryption scheme. As in the previous problem, we encode the message in the *least* significant bit of the ciphertext. Namely, we say that $\mathbf{c} \in \mathbb{Z}_q^{n+1}$ is an encryption of $\mu \in \{0,1\}$ if $\mathbf{s}^\mathsf{T}\mathbf{c} = \mu + 2e$ for some small e.

(a) Let $\mathbf{t}^\mathsf{T} = [-\tilde{\mathbf{t}}^\mathsf{T} \mid 1] \in \mathbb{Z}_q^{n'+1}$ for some $n' \in \mathbb{N}$. Your goal in this problem is to construct a method that *publicly* translates a ciphertext encrypted under $\mathbf{t}$ to a ciphertext under $\mathbf{s}$. We first define a $\mathsf{GenKeySwitch}$ algorithm:

- $\mathsf{GenKeySwitch}(\mathbf{s}, \mathbf{t}) \to \mathbf{W}$: On input $\mathbf{s} \in \mathbb{Z}_q^{n+1}$ and $\mathbf{t} \in \mathbb{Z}_q^{n'+1}$, the key-switching setup algorithm outputs a key-switching matrix $\mathbf{W} \in \mathbb{Z}_q^{(n+1)\times m'}$ where $m' = (n'+1)\lceil \log q \rceil$.

Suppose $\mathbf{W} \leftarrow \mathsf{GenKeySwitch}(\mathbf{s}, \mathbf{t})$. Then the following properties should hold:

- If $\mathbf{t}^\mathsf{T}\mathbf{c} = \mu + 2e \bmod q$, then $\mathbf{s}^\mathsf{T}\mathbf{W}\mathbf{g}^{-1}(\mathbf{c}) = \mu + 2e' \bmod q$ where $|e'| \le |e| + \text{poly}(n, n', \log q)$.
- If $\tilde{\mathbf{s}} \xleftarrow{\mathsf{R}} \mathbb{Z}_q^n$, then the key-switching matrix $\mathbf{W}$ output by Setup is pseudorandom assuming $\mathsf{LWE}_{n,n',q,\chi}$.

Prove that your scheme satisfies *both* of these properties. Your construction essentially shows how to transform a ciphertext $\mathbf{c} \in \mathbb{Z}_q^{n'+1}$ under any key $\mathbf{t} \in \mathbb{Z}_q^{n'+1}$ to a new ciphertext under $\mathbf{s}$. **Hint:** You may use the fact that $(\mathbf{t}^\mathsf{T} \otimes \mathbf{g}) \cdot \mathbf{g}^{-1}(\mathbf{c}) = \mathbf{t}^\mathsf{T}\mathbf{c}$. (You should convince yourself that this is true).

(b) Suppose you have two Regev ciphertexts $\mathbf{c}_1, \mathbf{c}_2$ encrypting μ_1 and μ_2 under $\mathbf{s} \in \mathbb{Z}_q^{n+1}$ with error magnitude at most e. Using the key-switching procedure defined above, show how to publicly and efficiently compute a Regev encryption $\mathbf{c}_\times$ of the product $\mu_1\mu_2$ under a suitably-chosen target key $\mathbf{t} \in \mathbb{Z}_q^{n+1}$. Note that $\mathbf{s}$ and $\mathbf{t}$ have the *same* dimension. The error in $\mathbf{c}_\times$ should be bounded by $O(e^2) + \text{poly}(n, \log q)$. In this setting, the public key would include a suitably-chosen key-switching matrix. Semantic security of the encryption scheme should still reduce to the LWE assumption. **Hint:** Use the following special case of the mixed product rule for tensor products: for all $\mathbf{u}_1, \mathbf{v}_1, \mathbf{u}_2, \mathbf{v}_2 \in \mathbb{Z}_q^n$, $(\mathbf{u}_1 \otimes \mathbf{u}_2)^\mathsf{T}(\mathbf{v}_1 \otimes \mathbf{v}_2) = (\mathbf{u}_1^\mathsf{T}\mathbf{v}_1)(\mathbf{u}_2^\mathsf{T}\mathbf{v}_2)$.

(c) In *one* sentence, explain how you can extend the above procedure to support any (bounded) number of multiplications. You may make a circular security assumption.

Challenge Problem: ℓ-Succinct SIS and SIS [Optional]. Recall that the ℓ-succinct SIS problem says that SIS is hard with respect to $\mathbf{A} \xleftarrow{\mathsf{R}} \mathbb{Z}_q^{n\times m}$ given a trapdoor for the matrix

$$\mathbf{B}_\ell = \begin{bmatrix} \mathbf{A} & & & \mathbf{W}_1 \\ & \ddots & & \vdots \\ & & \mathbf{A} & \mathbf{W}_\ell \end{bmatrix},$$

where $\mathbf{W}_1, \ldots, \mathbf{W}_\ell \xleftarrow{\mathsf{R}} \mathbb{Z}_q^{n\times t}$. Recall that a trapdoor for $\mathbf{B}_\ell$ is a short matrix $\mathbf{R}$ where $\mathbf{B}_\ell\mathbf{R} = \mathbf{G}$. For the applications to functional commitments (i.e., succinct homomorphic commitments), we set $t = m$ (independent of ℓ). Show that when $t = \Omega(\ell n \log q)$, hardness of vanilla SIS implies the hardness of ℓ-succinct SIS (with comparable parameters). An interesting open problem is to show such a comparable implication when $t \ll \ell$.