6.5630 Advanced Topics in Cryptography

Problem Set 1

Due: October 7, 2024

1 Minkowski is Tight [15 points]

Minkowski's theorem is tight for general lattices. In particular, there is a family of lattices $\{\mathcal{L}_n\}_{n\in\mathbb{N}}$ where $\mathcal{L}_n$ lives in n dimensions, and

$$\lambda_1(\mathcal{L}_n) \geq c \cdot \sqrt{n} \cdot \det(\mathcal{L}_n)^{1/n}$$

where c is a universal constant independent of n. Show that such a family of lattices exists (your proof doesn't have to construct this family, you merely have to show existence). *Hint:* Try the SIS lattice. That is, pick a random $\mathbf{A} \in \mathbb{Z}_q^{n\times m}$ and look at the lattice $\Lambda^\perp(\mathbf{A}) := \{\mathbf{x} : \mathbf{A}\mathbf{x} = 0 \pmod q\}$. You can use the following fact: if $\mathbf{A}$ has rank n over $\mathbb{Z}_q$, then the determinant of $\Lambda^\perp(\mathbf{A})$ is q^n.

Optional. Same problem except show an explicit construction of such a family of lattices $\{\mathcal{L}_n\}_{n\in\mathbb{N}}$.

2 (Our Analysis of) LLL is Tight [15 points]

(For a refresher on the LLL algorithm, look at the notes for Lecture 2.)

Let $\delta = 3/4$. Find a δ-LLL reduced basis $\mathbf{b}_1, \ldots, \mathbf{b}_n \in \mathbb{R}^n$ such that $\mathbf{b}_1$ is longer than the shortest vector by a factor of $\Theta(2^{n/2})$. In other words, our analysis of the LLL algorithm using LLL reduced bases is tight.

3 Cheap Gaussian Sampling? [15 points]

Consider the following algorithm for sampling from the zero-centered discrete Gaussian distribution $D_{\mathcal{L},s}$. Assume we have a good basis $\mathbf{B}$ of $\mathcal{L}$. The algorithm samples a point from the continuous Gaussian distribution $\rho_s(x)/s^n$, rounds it to a nearby lattice point (say, using Babai's rounding algorithm), and outputs the result.

Show that the output of this algorithm is statistically quite far from $D_{\mathcal{L},s}$, even for radii s that are polynomially bigger than the length of the given basis. *Hint:* Try $\mathbb{Z}$.

4 Spooky Encryption (20 points)

Fully homomorphic encryption tells us how to transform an encrypted input $c \in \mathsf{Enc}_{\mathsf{pk}}(x)$ into an encrypted output $c' \in \mathsf{Enc}_{\mathsf{pk}}(f(x))$ for any polynomial-time computable function f. Suppose you are now given

$$c_1 \in \mathsf{Enc}_{\mathsf{pk}_1}(x_1) \text{ and } c_2 \in \mathsf{Enc}_{\mathsf{pk}_2}(x_2)$$

for *independent* public keys pk_1 and pk_2 and some x_1, x_2.

- Could one now compute an encryption of $f(x_1, x_2)$ under either pk_1 or pk_2? Show a function f such that being able to do so for any x_1, x_2 will violate the IND-CPA security of the encryption scheme.
- Starting with the GSW FHE scheme we saw in class, construct an FHE scheme where one can produce two ciphertexts c_1' and c_2' such that

$$\mathsf{Dec}_{\mathsf{sk}_1}(c_1') \oplus \mathsf{Dec}_{\mathsf{sk}_2}(c_2') = f(x_1, x_2)$$

5 If Pigs Fly...? [20 points]

The goal of private information retrieval (PIR) is for a client to obtain the i'th bit of a database $D \in \{0,1\}^N$ from server, without the server learning anything about i.

Definition 1 (Private Information Retrieval). *A PIR scheme consists of four (potentially randomized) algorithms* Prep, Query, Resp, *and* Dec, *which are run in the following order:*

1. ***Prep.*** *The server preprocesses the dataset by computing* $\tilde{D} \leftarrow \mathsf{Prep}(D)$. *(This step is done once and for all by the server.)*
2. ***Query.*** *To query index* $i \in [N]$, *the client computes* $(q, s) \leftarrow \mathsf{Query}(i, 1^\lambda)$ *and sends* q *to the server. (*s *is some private state that the client keeps to herself.)*
3. ***Respond.*** *The server sends* $a \leftarrow \mathsf{Resp}(q, \tilde{D})$ *back to the client. (Here,* Resp *has random access to its inputs, so it can potentially run in sublinear time)*
4. ***Decode.*** *Client computes* $b \leftarrow \mathsf{Dec}(a, s)$.

We require that the scheme has the following two properties:

- ***Correct:*** *In the notation above, for all* i, *we have* $b = D_i$ *with probability one.*
- ***Secure:*** *for all* $i, i' \in [N]$, *the distributions* $\mathsf{Query}(i, 1^\lambda)$ *and* $\mathsf{Query}(i', 1^\lambda)$ *are computationally indistinguishable.*

We ask you to prove the following:

1. Show that if no preprocessing is done (i.e., $\tilde{D} = D$) by a PIR scheme, then $\mathsf{Resp}(q, \tilde{D})$ needs to run in $\Omega(N)$ time.
2. Assuming a "strongly preprocessable" (defined below) homomorphic encryption scheme exists, construct a PIR scheme where Query, Resp, and Dec run in $\mathrm{poly}(\log N)$ time and Prep runs in $\mathrm{poly}(N)$ time.

Definition 2 (Strongly Preprocessable Homomorphic Encryption Scheme). *A fully homomorphic encryption scheme is strongly preprocessable if there are deterministic algorithms* Process *and* Eval *such that both of the following hold:*

- *given a circuit* $C : \{0,1\}^n \to \{0,1\}$ *of size* s *(which can be much larger than* n*),* $\mathsf{Process}(C)$ *runs in* $\mathrm{poly}(s, n)$*-time and outputs a string* $\tilde{C}$, *and*
- *if* ct *is an encryption of* $x \in \{0,1\}^n$, *then* $\mathsf{Eval}(ct, \tilde{C})$ *runs in* $\mathrm{poly}(n)$*-time (note this is independent of* s*!!) and outputs an encryption of* $C(x)$. *Here, Eval is given random access to* $\tilde{C}$.