Massachusetts Institute of Technology
6.5610: Applied Cryptography — March 16, 2024
Henry Corrigan-Gibbs and Yael Tauman Kalai — **Due:** March 1, 2024 at 4:59pm

---

Problem Set 2

Please submit your problem set, in PDF format, on Gradescope. *Each problem should be in a separate page.*

You are to work on this problem set in groups. For problem sets 1, 2, and 3, we will randomly assign the groups for the problem set. After problem set 3, you are to work on the following problem sets with groups of your choosing of size three or four. If you need help finding a group, try posting on Piazza. See the course website for our policy on collaboration. Each group member must independently write up and submit their own solutions.

Homework must be typeset in LaTeX and submitted electronically! Each problem answer must be provided as a separate page. Mark the top of each page with your group member names, the course number (6.5610), the problem set number and question, and the date. We have provided a template for LaTeX on the course website (see the *Psets* tab at the top of the page).

Problem 2-1. [Extra Credit] Seminar attendance

Nick Sullivan is coming to speak at our security seminar. He co-chairs the IRTF Crypto Forum Research Group, which standardizes many of the crypto algorithms we use on the Internet. Until recently, he was also the head of research at CloudFlare, where he was one of the industry leaders in deploying new and experimental crypto protocols at scale.

We will give **extra credit** on this problem set if you attend.

Details are:

- Time: Thursday, February 29 at noon
- Place: 32-D463 (Star)
- Also important: There will be free food.
- Even more important, please register at this URL so that we get enough free food: `https://forms.gle/XbNcqjZohvh898677`

If you have a timing conflict at this time, we cannot give make-up extra credit but there will be many future extra credit opportunities!

There are other seminar talks as well and many are related to the content that we will cover in this class! See `https://securityseminar.csail.mit.edu/` for details.

Problem 2-2. Linearly Homomorphic Encryption and PIR Recall Regev's secret-key encryption scheme from lecture, where the secret key is $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$,

$$\mathsf{Enc}(\mathbf{s}, b \in \{0,1\}) = (\mathbf{a},\ \mathbf{a}^\top \mathbf{s} + e + b \cdot \lfloor q/2 \rfloor),$$

where $\mathbf{a} \xleftarrow{R} \mathbb{Z}_q^n$ and $e \xleftarrow{R} \chi \in \mathbb{Z}_q$, for some error distribution χ over $\mathbb{Z}_q$. Suppose χ is a uniform distribution over the interval $[-B, B] \subseteq \mathbb{Z}_q$ for some B. Moreover, decryption is:

$$\mathsf{Dec}(\mathbf{s}, (\mathbf{a}, c)) = 0 \begin{cases} 0 & \text{if } |c - \mathbf{a}^\top \cdot \mathbf{s}| < q/4 \\ 1 & \text{otherwise} \end{cases}.$$

(a) What restriction is needed on B to ensure decryption correctness?

Solution: $B < \frac{q}{4}$.

(b) Recall that this encryption scheme is additively homomorphic. That is,

$$\begin{aligned}\mathsf{Enc}(\mu) + \mathsf{Enc}(\mu') &= (\mathbf{a}, \mathbf{a}^\top \mathbf{s} + e + \lfloor q/2 \rfloor \mu) + (\mathbf{a}', \mathbf{a}'^\top \mathbf{s} + e' + \lfloor q/2 \rfloor \mu') \\ &= (\mathbf{a} + \mathbf{a}', (\mathbf{a} + \mathbf{a}')^\top \mathbf{s} + (e + e') + \lfloor q/2 \rfloor (\mu + \mu')) \\ &= \mathsf{Enc}(\mu + \mu')\end{aligned}$$

As we compute more additions of encryptions, note that the error grows.

Give possible parameter values for B and q in terms of the security parameter λ so that the scheme supports poly(λ) many additions without breaking decryption correctness.

Solution: Every time we add ciphertexts, the error grows by $\leq 2B$. So for poly(λ) many additions, the error is $\leq \text{poly}(\lambda) \cdot 2B$. This must be less than $q/4$, so q must be super-polynomial in λ (e.g. $\lambda^{\log \lambda}$) and B must be polynomial in λ.

(c) Consider the following variant of Regev's symmetric encryption scheme, where the message space is $\{0, 1, 2\}$: As before the secret key is $\mathbf{s} \xleftarrow{R} \mathbb{Z}_q^n$ and the new encryption algorithm is defined by

$$\mathsf{Enc}(\mathbf{s}, m) = (\mathbf{a}, \mathbf{a}^\top \mathbf{s} + e + m \lfloor q/3 \rfloor)$$

where $\mathbf{a} \xleftarrow{R} \mathbb{Z}_q^n$ and $e \leftarrow \chi$.

Give a decryption algorithm that will ensure correctness of this scheme, assuming the error is in the interval $[-B, B]$ and B is small (say smaller than $\frac{q}{10}$).

How can you use this encryption scheme to construct a PIR scheme (for a sufficiently small B)?

Solution: Decryption(a, c): output

- 0 if $|c - a^T \cdot s| < b$
- 1 if $|c - a^T \cdot s - \lfloor q/3 \rfloor| < b$
- 2 if $|c - a^T \cdot s - 2\lfloor q/3 \rfloor| < b$

where $q/10 \leq b$. Looser bounds are also okay, as long as the $\pm q/10$ intervals around $0, \lfloor q/3 \rfloor, 2\lfloor q/3 \rfloor$ are covered and the bounds do not overlap.

To construct a PIR scheme, can use the same scheme as described here: https://65610.csail.mit.edu/2024/lec/l05-pir.pdf in the "A classic PIR scheme: Square-root PIR" section but modified to $\mathbb{Z}_3$.

Problem 2-3. Negligible Functions and CPA Security Let $\mu : \mathbb{N} \to \mathbb{R}$ be a negligible function, and let p be a polynomial such that $p(k) \geq 0$ for all $k > 0$. State whether the following functions are negligible with a 1-3 sentence explanation.

(a) $\mu(k) \cdot p(k)$

Solution: Yes. Suppose for contradiction that for all k_0, there exists $k > k_0$ such that $\mu(k)p(k) > 1/p'(k)$ for some polynomial p', then that would imply $\mu(k) > \frac{1}{p'(k)p(k)}$, which is a contradiction because $p'(k)p(k)$ is also a polynomial.

(b) $\mu(k)^{1/p(k)}$

Solution: No. We know that the function 2^{-k} is negligible. But $(2^{-k})^{1/k} = 2^{-1}$ is not.

(c) $\mu(k)^{1/c}$ for some constant $c > 0$

Solution: Yes. Suppose for contradiction that for all k_0, there exists $k > k_0$ such that $\mu(k)^{1/c} > 1/p'(k)$ for some polynomial p', then that would imply $\mu(k) > 1/p'(k)^c$, which is a contradiction because $p'(k)^c$ is also a polynomial.

(d) $\mu_1(k) + \mu_2(k)$ for negligible functions $\mu_1, \mu_2 : \mathbb{N} \to \mathbb{R}$

Solution: Yes. We know that there exists k'_0 such that for all $k > k'_0$, $\mu_1(k) < 1/p(k)$ for all polynomials p. Now suppose for contradiction that for all k_0, there exists $k > k_0$ such that $\mu_1(k) + \mu_2(k) > 1/p'(k)$ for some polynomial p'. Then that means for all $k_0 > k'_0$, there exists $k > k_0$ such that $1/(2p'(k)) + \mu_2(k) > \mu_1(k) + \mu_2(k) > 1/p'(k)$ and $\mu_2(k) > 1/p'(k) - 1/(2p'(k)) = 1/(2p'(k))$, which is a contradiction.

Let $F : K \times \{0,1\}^n \to \{0,1\}^n$ be a PRF. State whether the following encryption schemes are CPA secure with a 1-2 sentence explanation.

(e) $\text{Enc}(k, m) \stackrel{\text{def}}{=} m \oplus F(k, 0^n)$
$\text{Dec}(k, c) \stackrel{\text{def}}{=} c \oplus F(k, 0^n)$

Solution: No because the scheme is not randomized, same input will always map to same output.

(f) $\text{Enc}(k, (m_1, m_2)) \stackrel{\text{def}}{=} (r, m_1 \oplus F(k, r), m_2 \oplus F(k, r))$
$\text{Dec}(k, (r, c_1, c_2)) \stackrel{\text{def}}{=} (c_1 \oplus F(k, r), c_2 \oplus F(k, r))$

Solution: No because the adversary can calculate $c_1 \oplus c_2$ to get $m_1 \oplus m_2$.

Recall our construction of a CPA secure scheme using a PRF.

(g) We know that this construction provides secrecy, but show that it does not provide message integrity. More specifically, show that if an adversary is given the ciphertext $c = \text{Enc}(k, m)$, then the adversary can construct a new ciphertext c' for the message $m \oplus m_2$ for any m_2 without knowing the original message m.

Solution: We know $c = (r, F(k, r) \oplus m)$. We can simply set $c' = (r, F(k, r) \oplus m \oplus m_2)$

Problem 2-4. Implementing Regev

In this problem, you will work on Regev's public key encryption scheme. Specifically, we will use the following version:

- Private key: $\mathbf{s} \stackrel{R}{\leftarrow} \mathbb{Z}_q^n$
- Public key: $\mathbf{A} \stackrel{R}{\leftarrow} \mathbb{Z}_q^{m \times n}$, $\mathbf{u} = \mathbf{As} + \mathbf{e} \in \mathbb{Z}_q^m$ where $\mathbf{e} \stackrel{R}{\leftarrow} [-B, B]^m \subseteq \mathbb{Z}_q^m$ for some bound B.
- Encryption:

$$\mathsf{Enc}(b) = \mathbf{r}^T \cdot \begin{bmatrix} \mathbf{A} & \mathbf{u} \end{bmatrix} + b \cdot \left(0, 0, \cdots, 0, \left\lfloor \frac{q}{2} \right\rfloor\right)$$

where b is a message bit and $\mathbf{r} \stackrel{R}{\leftarrow} \{0,1\}^m \subseteq \mathbb{Z}_q^m$.

You can find a Python file on Piazza:

- `encrypt.py`: This file contains the template file for you to fill in the encryption function. The public key and the parameters can also be found here. It is recommended to not touch anything other than the `enc` function.

(a) It is clear that if the error vector is too big, the decryption scheme might fail. But what happens if the error vector is really small? If every entry is in the range $[-B, B]$, an attacker can bruteforce the error vector and recover the secret key in time $O(B^n)$ (not $O(B^m)$ because we only need n equations to find the secret key). It turns out that there's a much stronger attack based on lattice reduction algorithms. We won't describe the details here, but the attack finds a candidate $(\mathbf{s}', \mathbf{e}')$ such that $\mathbf{u} = \mathbf{A} \cdot \mathbf{s}' + \mathbf{e}'$ and every entry of $\mathbf{e}'$ is bounded by T:

$$|\mathbf{e}'_i| \leq T = 2^{(n+m)/4} \cdot (q^{m-n} \cdot B^{n+1})^{1/(n+m+1)} \tag{1}$$

The hope is that $\mathbf{e} = \mathbf{e}'$ and $\mathbf{s} = \mathbf{s}'$. To verify this, we ask the question "How many valid error vectors are there in the range $[-T, T]^m$?" If there's only one such vector, then we are 100% sure that $\mathbf{e} = \mathbf{e}'$; otherwise, $\mathbf{e}'$ might just be another solution.

Since there are q^n possible error vectors (every $\mathbf{s}$ is mapped to an error vector). The probability that a random vector of length m is a valid error vector is $q^n/q^m = q^{n-m}$. Therefore, the expected value of the number of valid vectors is

$$(2T)^m \cdot q^{(n-m)},$$

where T is defined as in Eq. (1). If this quantity is at most one, we are confident that there's only one valid small error vector in the range $[-T, T]^m$ and thus $\mathbf{e} = \mathbf{e}'$ and $\mathbf{s} = \mathbf{s}'$.

Using the parameters in `encrypt.py`, find the minimum positive integer B such that

$$(2T)^m \cdot q^{(n-m)} > 1$$

Bonus: the public key is generated with a smaller error vector, can you recover the secret key? Come to Thursday's office hour and discuss with the TA!

Solution: $B = 748$. Any number in $[745, 751]$ should be considered correct.

(b) Implement the `enc()` in `encrypt.py` and submit the Python file (i.e. filename ends with `.py`) to the Gradescope. **Make sure that r is not deterministic**.