Homework 3 for 02231, 2025 (10 points)

Due 1.12.2025, **08:00**

Notation. When reducing modulo p the remainder will be within the set $\{-(p-1)/2, \ldots, (p-1)/2\}$. So for example, $8 \mod 5 = -2$ while $10 \mod 7 = 3$. For $x \in \mathbb{R}$ (i.e. a real number x) we let $\lfloor x \rceil$ be the function that rounds x to the nearest integer y. That is, for any integer $y \in \mathbb{Z}$ we round any $y - 0.5 \leq x < y + 0.5$ to y. For example, $\lfloor 1 \rceil = 1$, $\lfloor 0.4 \rceil = 0$, $\lfloor 0.5 \rceil = 1$ and $\lfloor 2.99999 \rceil = 3$.

Exercise 3.1. (A simple CCA attack on Regev's encryption scheme - 3 points)

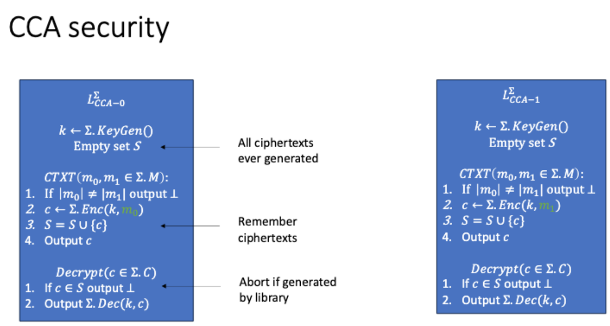

Figure 1: The libraries for IND-CCA.

In this exercise, we consider the security of Regev's public-key encryption scheme under *chosen-ciphertext attacks* (CCA). Recall that it consists of the following 3 algorithms:

KEYGEN() Samples $\mathbf{A} \leftarrow \mathbb{Z}_p^{\ell \times n}$, $\mathbf{s} \leftarrow \mathbb{Z}_p^n$ and $\mathbf{e} \leftarrow q_\sigma$ where q_σ is the error distribution. Then compute $\mathbf{b} = \mathbf{A}\mathbf{s} + \mathbf{e} \mod p$ and output $(sk, pk) = (\mathbf{s}, (\mathbf{A}, \mathbf{b}))$.

ENC(pk, m) on input $m \in \{0, 1\}$ samples $\mathbf{r} \in \{0, 1\}^\ell$, computes $\mathbf{c}_0 = \mathbf{r}^\top \mathbf{A} \mod p$ and $c_1 = \mathbf{r}^\top \mathbf{b} + m \cdot (p-1)/2 \mod p$ and output $c = (\mathbf{c}_0, c_1)$.

DEC(sk, c) computes

$$m' = \left\lfloor \frac{2(c_1 - \mathbf{c}_0^\top \mathbf{s} \mod p)}{p} \right\rceil$$

and outputs m'.

Assume that an adversary has access to a *decryption oracle* that on input any ciphertext $c' = (\mathbf{c}_0', c_1')$ returns the decrypted message m', except that the oracle refuses to decrypt one specific challenge ciphertext $c^* = (\mathbf{c}_0^*, c_1^*)$ that was output by CTXT.

1. Show that for any bit $b \in \{0,1\}$, the ciphertext

$$c' = (\mathbf{c}_0^*,\ c_1^* + \frac{p-1}{2} \bmod p)$$

   decrypts to $1 - m_b$, where m_b is the message encrypted in c^*.

2. Use this observation to describe a chosen-ciphertext attack, i.e., an elgorithm that can distinguish L_{CCA-0} and L_{CCA-1} for Regev's public-key encryption scheme.

Exercise 3.2. (One-time signatures - 7 points)

Consider the Lamport one-time signature scheme (OTS) for messages of length n bits, using a hash function H with λ-bit output. Let the scheme be denoted by $\Sigma_{\text{Lam}} = (\Sigma_{\text{Lam}}.\mathsf{KeyGen}, \Sigma_{\text{Lam}}.\mathsf{Sign}, \Sigma_{\text{Lam}}.\mathsf{Ver})$:

- $\Sigma_{\text{Lam}}.\mathsf{KeyGen}(1^\lambda)$: sample $2n$ random bit strings $x_{0,1}, x_{1,1}, \ldots, x_{0,n}, x_{1,n} \in \{0,1\}^\lambda$ and set $y_{b,i} = H(x_{b,i})$ for $i \in [n], b \in \{0,1\}$. Set $vk = (y_{0,1}, y_{1,1}, \ldots, y_{1,n})$ and $sk = (x_{0,1}, x_{1,1}, \ldots, x_{1,n})$.

- $\Sigma_{\text{Lam}}.\mathsf{Sign}(sk, m)$: Let $m = (m_1 \ldots m_n)$ with $m_i \in \{0,1\}$. Output the signature $\sigma = (x_{m_1,1}, \ldots, x_{m_n,n})$.

- $\Sigma_{\text{Lam}}.\mathsf{Ver}(vk, m, \sigma)$: Parse $vk = (y_{0,1}, y_{1,1}, \ldots, y_{1,n})$ and $\sigma = (\sigma_1, \ldots, \sigma_n)$. Output 1 if and only if $y_{m_i,i} = H(\sigma_i)$ for all $i \in [n]$.

1. Show that Σ_{Lam} is correct.

2. In the lecture, trapdoor one-way functions were briefly mentioned. A one-way function is a similar concept, just without a trapdoor:

   Definition 1 (One-wayness)**.** *Let $H : \{0,1\}^* \to \{0,1\}^\lambda$ be a hash function, and let $m \in \mathbb{N}$ be a length bound. We define the libraries*

$$\mathcal{L}_{H,m}^{\mathsf{ow}\text{-}real} \quad \textit{and} \quad \mathcal{L}_{H,m}^{\mathsf{ow}\text{-}ideal}$$

   as follows:

$\mathcal{L}_{H,m}^{\textsf{ow-}\textit{real}}$

$Y \leftarrow \emptyset$

Oracle $\mathsf{Challenge}()$*:*
  Sample $x \leftarrow \{0,1\}^m$
  $y \leftarrow H(x)$
  $Y \leftarrow Y \cup \{y\}$
  return y

Oracle $\mathsf{Check}(x')$*:*
  return $\left[H(x') \in Y\right]$

$\mathcal{L}_{H,m}^{\textsf{ow-}\textit{ideal}}$

Oracle $\mathsf{Challenge}()$*:*
  Sample $x \leftarrow \{0,1\}^m$
  $y \leftarrow H(x)$
  return y

Oracle $\mathsf{Check}(x')$*:*
  return 0

We say that H is a one-way hash function for length m *if* $\mathcal{L}_{H,m}^{\textsf{ow-}\textit{real}} \approx \mathcal{L}_{H,m}^{\textsf{ow-}\textit{ideal}}$.

Show that Σ_{Lam} is secure. To do that, show that from any adversary that can distinguish $L_{\textit{sig-real}}^{\Sigma_{\text{Lam}}}$ and $L_{\textit{sig-fake}}^{\Sigma_{\text{Lam}}}$, you can build an adversary distinguishing $\mathcal{L}_{H,m}^{\textsf{ow-real}}$ and $\mathcal{L}_{H,m}^{\textsf{ow-ideal}}$ for $m \geq \lambda$.

Hint: Simulate $L_{\textit{sig-real}}^{\Sigma_{\text{Lam}}}$ and use the $\mathsf{Challenge}()$ function of $\mathcal{L}_{H,m}^{\textsf{ow-real}}$ or $\mathcal{L}_{H,m}^{\textsf{ow-ideal}}$ for key generation.

3. Σ_{Lam} is not very efficient. What is the combined size (in bits) of one public key and one signature[1] when $n = 256$ and $\lambda = 128$? Compare this with the size of one public key and one signature for RSA-FDH as introduced in the lecture, with $2^{4095} < N < 2^{4096}$ and fixed public exponent e that does not need to be included in the public key.

4. Consider a variant $\Pi = (\Pi.\mathsf{KeyGen}, \Pi.\mathsf{Sign}, \Pi.\mathsf{Ver})$ using hash chains for a small message space $\mathcal{M} = \{0,1,2,3,4,5,6,7\}$ and the same H:

   - $\Pi.\mathsf{KeyGen}(1^\lambda)$: sample $x \in \{0,1\}^\lambda$, set $y = H^8(x) = H(H(H(H(H(H(H(H(x))))))))$, and output $(sk, vk) = (x, y)$.
   - $\Pi.\mathsf{Sign}(sk, m)$: output $\sigma = H^m(sk)$.
   - $\Pi.\mathsf{Ver}(vk, m, \sigma)$: accept if and only if $vk = H^{8-m}(\sigma)$.

   Show that this OTS is correct.

5. Show that Π is not secure by constructing an adversary that can distinguish $L_{\textit{sig-real}}$ and $L_{\textit{sig-fake}}$ using at most one query to $\textit{GetSig}$.

[1] In many settings, the combined size of public key and signature is a good measure for the bandwidth consumed by the digital signature scheme.

6. Construct a variant of Π (the *Winternitz OTS*) as follows:

   - $\Pi_W.\mathsf{KeyGen}$: generate two key pairs $(sk_i, vk_i) \leftarrow \Pi.\mathsf{KeyGen}$ for $i \in \{0,1\}$, and set $sk = (sk_0, sk_1)$, $vk = (vk_0, vk_1)$.
   - $\Pi_W.\mathsf{Sign}(sk, m)$: output $\sigma = (\sigma_0, \sigma_1)$ where $\sigma_0 \leftarrow \Pi.\mathsf{Sign}(sk_0, m)$ and $\sigma_1 \leftarrow \Pi.\mathsf{Sign}(sk_1, 7 - m)$.
   - $\Pi_W.\mathsf{Ver}(vk, m, \sigma)$: accept if $\Pi.\mathsf{Ver}(vk_0, m, \sigma_0) = 1$ and $\Pi.\mathsf{Ver}(vk_1, 7-m, \sigma_1) = 1$.

   Explain why the adversary from the previous sub-problem can no longer forge a signature in Π_W, even if it queries the signing oracle once.

7. Compare the combined size (in bits) of one public key and one signature for the Winternitz OTS defined here and for Σ_{Lam} with $n = 3$ (i.e. for the same message space size).

What you should do

- Enroll into one of the homework submission groups. You are encouraged to work in groups of (up to) 3, so the groups have capacity 3.
- Write the solutions to the exercises in one document.
- Upload your document on Learn.
- You may work in groups of at most three students.
- The format of your document should be PDF, together with a ZIP file containing any program code that you created as part of the exercises.
- For any program code, please also describe your solution in the pdf so that it can be understood without looking at all the details of your code.
- The PDF document should contain your group number as part of the title. Each submitted code file should contain your group number in the beginning as a comment.