Tutorial 7: Public key encryption

Exercise 1. *HMAC*

Before HMAC was invented, it was quite common to define a MAC by $\mathrm{Mac}_k(m) = H^s(k \parallel m)$ where H is a collision-resistant hash function. Show that this MAC is not unforgeable when H is constructed via the Merkle-Damgård transform.

Exercise 2. *SIS*

Definition 1 (Learning with Errors)**.** *Let $\ell < k \in \mathbb{N}$, $n < m \in \mathbb{N}$, $q = 2^k$, $B = 2^\ell$, $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{m\times n})$. The Learning with Errors (*LWE*) distribution is defined as follows: $D_{\mathrm{LWE},\mathbf{A}} = (\mathbf{A}, \mathbf{A}\cdot\mathbf{s} + \mathbf{e} \bmod q)$ for $\mathbf{s} \hookleftarrow U(\mathbb{Z}_q^n)$ and $\mathbf{e} \hookleftarrow U\left(\left[-\frac{B}{2}, \frac{B}{2}\right]^m \cap \mathbb{Z}^m\right)$.*

The $LWE_\mathbf{A}$ *assumption* states that, given suitable parameters k, ℓ, m, n, it is computationally hard to distinguish $D_{\mathrm{LWE},\mathbf{A}}$ from the distribution $(\mathbf{A}, U(\mathbb{Z}_q^m))$.

Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{m\times n}$ with $m > n \lg q$, let us define the following hash function:

$$\begin{array}{rccc} H_\mathbf{A}: & \{0,1\}^m & \to & \{0,1\}^n \\ & \mathbf{x} & \mapsto & \mathbf{x}^T \cdot \mathbf{A} \bmod q. \end{array}$$

1. Why finding a sufficiently "short" non-zero vector $\mathbf{z}$ such that $\mathbf{z}^T \cdot \mathbf{A} = \mathbf{0}$ is enough to distinguish $D_{\mathrm{LWE},\mathbf{A}}$ from the distribution $(\mathbf{A}, U(\mathbb{Z}_q^m))$? Define "short".
2. Show that $H_\mathbf{A}$ is *collision-resistant* under the $LWE_\mathbf{A}$ assumption.
3. Is it still a secure hash function if we let $H_\mathbf{A} : \mathbf{x} \in \{0,1\}^m \mapsto \mathbf{x}^T \cdot \mathbf{A} \in \mathbb{Z}^n$? (without the reduction modulo q).

Exercise 3. *One-time to Many-Times*

Let us define the following experiments for $b \in \{0,1\}$, and $Q = \mathsf{poly}(\lambda)$.

$$\mathbf{Exp}_b^{\text{many-CPA}}$$

$$\begin{array}{lcr} \mathcal{A} & & \mathcal{C} \\ \hline & & (pk, sk) \leftarrow \mathsf{Keygen}(1^\lambda) \\ & \xleftarrow{pk} & \\ \text{Choose } \left(m_0^{(i)}, m_1^{(i)}\right)_{i=1}^Q & & \\ & \xrightarrow{(m_0^{(i)}, m_1^{(i)})_{i=1}^Q} & \\ & & \left(c_i^\star = \mathsf{Enc}_{pk}\left(m_b^{(i)}\right)\right)_{i=1}^Q \\ & \xleftarrow{(c_i^\star)_{i=1}^Q} & \\ \text{Output } b' \in \{0,1\} & & \end{array}$$

The advantage of $\mathcal{A}$ in the many-time CPA game is defined as

$$\mathrm{Advt}^{\text{many-CPA}}(\mathcal{A}) = \left| \Pr_{(pk,sk)}[\mathcal{A} \to 1 \mid \mathbf{Exp}_1^{\text{many-CPA}}] - \Pr_{(pk,sk)}[\mathcal{A} \to 1 \mid \mathbf{Exp}_0^{\text{many-CPA}}] \right|$$

1. Recall the definition of CPA-security that was given during the course. What is the difference?

2. Show that this two definitions are equivalent.

3. Do we have a similar equivalence in the secret-key setting?

Exercise 4. *Variants of LWE*

We define a variant of the LWE problem with multiple secrets as follows.

Definition 2 (Multiple-secrets-LWE distribution). *Let $\ell < k \in \mathbb{N}$, $n < m \in \mathbb{N}$, $q = 2^k$, $B = 2^\ell$, $t = poly(m)$ be some integer, and $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$. The multiple-secrets-LWE distribution is defined as follows: $D_{ms\text{LWE},A} = (A, A \cdot S + E \bmod q)$ for $S \hookleftarrow U(\mathbb{Z}_q^{n\times t})$ and $E \hookleftarrow U\left(\left[-\frac{B}{2}, \frac{B}{2} - 1\right]^{m\times t} \cap \mathbb{Z}^{m\times t}\right)$.*

Note. The secret is now a matrix instead of a vector. Each column of this matrix can be seen as a secret for the LWE distribution.

1. Show that if the LWE assumption holds, then the multiple-secrets-LWE distribution is computationally indistinguishable from the uniform distribution $U(\mathbb{Z}_q^{m\times n} \times \mathbb{Z}_q^{m\times t})$.
   Hint: you may want to use a hybrid argument.

   We study another variant of the LWE problem, where the matrix A is chosen uniformly among the matrices with coefficients in $\{0, 1\}$ instead of with coefficients in $\mathbb{Z}_q$. We want to show that this variant of LWE is also secure, as long as the LWE assumption holds.

   Definition 3 (Binary-matrix-LWE). *Let $\ell < k \in \mathbb{N}$, $n < m \in \mathbb{N}$, $q = 2^k$, $B = 2^\ell$, $A \hookleftarrow U(\{0, 1\}^{m\times n})$. The binary-matrix-LWE distribution is defined as follows: $D_{bm\text{LWE},A} = (A, A \cdot s + e \bmod q)$ for $s \hookleftarrow U(\mathbb{Z}_q^n)$ and $e \hookleftarrow U\left(\left[-\frac{B}{2}, \frac{B}{2} - 1\right]^{m} \cap \mathbb{Z}^{m}\right)$.*

   We write binary-matrix-LWE$_{n,m,\ell,k}$ when the parameters needs to be specified.

2. Show that there exist a matrix $G \in \mathbb{Z}_q^{nk\times n}$ such that for any matrix $A \in \mathbb{Z}_q^{m\times n}$, there exist a binary matrix $A_{bin} \in \{0, 1\}^{m\times nk}$ such that $A = A_{bin}G$.

3. Show that if A is sampled uniformly in $\mathbb{Z}_q^{m\times n}$, then A_{bin} is uniform in $\{0, 1\}^{m\times nk}$.

4. Let $s \in \mathbb{Z}_q^n$ be sampled uniformly. Is $G \cdot s$ still a uniform vector in $\mathbb{Z}_q^{nk}$? Is it computationally indistinguishable from a uniform vector?

5. Let $A \hookleftarrow U(\mathbb{Z}_q^{m\times n})$ and e be some error sampled as in the LWE distribution. Let s be any vector (not necessarily uniform) and let u be either $As + e$ or some uniform vector in $\mathbb{Z}_q^m$. Show that given (A, u) you can construct (A, u') such that u' is either uniform in $\mathbb{Z}_q^m$ or is of the form $As' + e$ for s' uniform in $\mathbb{Z}_q^n$.

6. Show that if the LWE$_{n,m,\ell,k}$ problem holds, then the binary-matrix-LWE$_{kn,m,\ell,k}$ distribution is indistinguishable from uniform.

7. Is the LWE problem still hard when both A and s are binary?

Exercise 5. *Pollard-rho*

Let $\mathbb{G}$ be a cyclic group generated by g, of (known) prime order q, and let h be an element of $\mathbb{G}$. Let $F : \mathbb{G} \to \mathbb{Z}_q$ be a nonzero function, and let us define the function $H : \mathbb{G} \to \mathbb{G}$ by $H(\alpha) = \alpha \cdot h \cdot g^{F(\alpha)}$. We consider the following algorithm (called *Pollard ρ Algorithm*).

Pollard ρ Algorithm

Input: $h, g \in \mathbb{G}$

Output: $x \in \{0, \ldots, q-1\}$ such that $h = g^x$ or FAIL.

1. $i \leftarrow 1$
2. $x \leftarrow 0$, $\alpha \leftarrow h$
3. $y \leftarrow F(\alpha)$; $\beta \leftarrow H(\alpha)$
4. **while** $\alpha \neq \beta$ do
5. $x \leftarrow x + F(\alpha) \bmod q; \alpha \leftarrow H(\alpha)$
6. $y \leftarrow y + F(\beta) \bmod q; \beta \leftarrow H(\beta)$
7. $y \leftarrow y + F(\beta) \bmod q; \beta \leftarrow H(\beta)$
8. $i \leftarrow i + 1$
9. **end while**
10. **if** $i < q$ **then**
11.   **return** $(x - y)/i \bmod q$
12. **else**
13.   **return** FAIL
14. **end if**

To study this algorithm, we define the sequence (γ_i) by $\gamma_1 = h$ and $\gamma_{i+1} = H(\gamma_i)$ for $i \geqslant 1$.

1. Show that in the **while** loop from lines 4 to 9 of the algorithm, we have $\alpha = \gamma_i = g^x h^i$ and $\beta = \gamma_{2i} = g^y h^{2i}$.

2. Show that if this loop finishes with $i < q$, then the algorithm returns the discrete logarithm of h in basis g.

3. Let j be the smallest integer such that $\gamma_j = \gamma_k$ for $k < j$. Show that $j \leqslant q + 1$ and that the loop ends with $i < j$.

4. Show that if F is a random function, then the average execution time of the algorithm is in $O(q^{1/2})$ multiplications in $\mathbb{G}$.