Tutorial 6: Hash functions

Exercise 1. *Pedersen's hash function*

Pedersen's hash function is as follows:

- Given a security parameter n, algorithm $\texttt{Gen}$ samples (G, g, q) where $G = \langle g \rangle$ is a cyclic group of cardinality q, a prime number. It then sets $g_1 = g$ and samples g_i uniformly in G for all $i \in \{2, \ldots, k\}$, where $k \geq 2$ is some parameter. Finally, it returns $(G, q, g_1, \ldots, g_k)$.
- The hash of message $M = (M_1, \ldots, M_k) \in (\mathbb{Z}/q\mathbb{Z})^k$ is $H(M) = \prod_{i=1}^k g_i^{M_i} \in G$.

1. Assume for this question that G is a subgroup of prime order q of $(\mathbb{Z}/p\mathbb{Z})^\times$, where $p = 2q + 1$ is prime. What is the compression factor in terms of k and p?

2. Assume for this question that $k = 2$. Show that Pedersen's hash function is collision-resistant, under the assumption that the Discrete Logarithm Problem (DLP) is hard for G.

3. Same question as the previous one, with $k \geq 2$ arbitrary.

Exercise 2. *CCA security*

1. We define the scheme "Encrypt and tag" by: for a message m, independent keys k and k', a CPA-secure encryption Enc and a secure MAC $Sign$, let $c = Enc(k, m)$ and $t = Sign(k', m)$, return (c, t). Is this scheme CCA-secure ?

Exercise 3. *Authenticated encryption*

Consider the following construction of symmetric encryption.

Gen(1^λ)**:** Choose a random key $K_1 \leftarrow U(\{0,1\}^\lambda)$ for an IND-CPA secure symmetric encryption scheme $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$. Choose a random key $K_0 \leftarrow U(\{0,1\}^\lambda)$ for a MAC $\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Verify})$. The secret key is $K = (K_0, K_1)$

Enc(K, M)**:** To encrypt M, do the following.

1. Compute $c = \mathsf{Enc}'(K_1, M)$.
2. Compute $t = \Pi.\mathsf{Mac}(K_0, c)$.

Return $C = (t, c)$.

Dec(K, C)**:** Return $\perp$ if $\Pi.\mathsf{Verify}(K_0, c, t) = 0$. Otherwise, return $M = \mathsf{Dec}'(K_1, c)$.

Recall that the MAC is said to be unforgeable if, in the security game, the adversary succeeds if it manages to create a valid pair (m, t) where t is a valid signature for m and m has never been queried before. The MAC is said to be **strongly** unforgeable if we replace in the previous definition "m has never been queried" by "(m, t) has never been sent by the challenger".

1. Show that the scheme may not be IND-CCA secure if the MAC Π is unforgeable (but not strongly) under chosen-message attacks.

2. Prove that the scheme is IND-CCA secure assuming that: (i) $(\mathsf{Gen}', \mathsf{Enc}', \mathsf{Dec}')$ is IND-CPA-secure; (ii) Π is stronly unforgeable under chosen-message attacks.

   Hint : you may want to introduce ValidQuery*, the event that the attacker $\mathcal{A}$ against the CCA security of the scheme makes a decryption query on (c, t) which was not previously obtained by the encryption oracle but such that t is a valid signature of c.*

Exercise 4. *Repetition*

1. Let (Gen, H_1) and (Gen', H_2) be collision-resistant hash functions such that $H_1 : \{0,1\}^n \to \{0,1\}^m$ and $H_2 : \{0,1\}^m \to \{0,1\}^\ell$ (with $n > m > \ell$). Is $(Gen, \widehat{H})$ defined by $\widehat{H}^{(s_1,s_2)} =_{def} H_2^{s_2}(H_1^{s_1}(x))$ necessarily collision-resistant?

Exercise 5. *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 6. *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 7. *One-time to Many-Times*

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

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

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

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?