TD 10: Digital Signatures (corrected version)

Exercise 1.
In this exercise we show a scheme that can be proven secure in the random oracle model, but is insecure when the random oracle model is instantiated with SHA-3 (or any fixed (unkeyed) hash function $H : \{0,1\}^* \to \{0,1\}^n$). Let Π be a signature scheme that is euCMA-secure in the standard model.
Let $y \in \{0,1\}^n$ and define the following signature scheme Π_y. The signing and verifying keys are obtained by running $\Pi.\mathsf{Gen}(1^\lambda)$. Signature of a message m is computed out as follows: if $H(0) = y$ then output the secret key, if $H(0) \neq y$ then return a signature computed using $\Pi.\mathsf{Sign}$. To verify a message, if $y = H(0)$ then accept any signature for any message and otherwise, verify it using $\Pi.\mathsf{Verify}$.

1. Prove that for any value y, the scheme Π_y is euCMA-secure in the random oracle model.

   ☞ In the ROM, we can reduce the security of Π_y from the security of Π, as the event $y = H(0)$ happens with negligible probability ($< 2^{-\lambda}$).

   Let us assume that there exists an adversary $\mathcal{A}$ that breaks the euCMA security of Π_y in the ROM. We build the following reduction $\mathcal{B}_y$ that on input a verification key vk does the following. It queries $H(0)$. If $H(0) = y$, it aborts. Otherwise, it forwards vk to $\mathcal{A}$ and uses its own signing oracle to sign the messages queried by $\mathcal{A}$. When $\mathcal{A}$ outputs a forgery, it forwards it. We then have:

   $$\mathsf{Adv}(\mathcal{B}) = \Pr(\mathcal{A} \text{ wins} \wedge H(0) \neq y).$$

   Moreover, it holds that

   $$\mathsf{Adv}(\mathcal{A}) \leq \Pr(\mathcal{A} \text{ wins} \wedge H(0) \neq y) + \frac{1}{2^n}.$$

   Then $\mathsf{Adv}(\mathcal{B}) \geq \mathsf{Adv}(\mathcal{A}) - 1/2^n$, which is non-negligible.

2. Show that there exists a particular y for which Π_y is insecure when the hash function is not modeled as a random oracle anymore.

   ☞ Let H be fixed. We look at $\Pi_{H(0)}$. This signature scheme always output its secret key as signature and moreover it accepts any signature for any message. It is then not euCMA-secure.

Exercise 2.
Define a lattice-based Schnorr-like signature scheme as follows:

$\mathsf{Gen}(1^\lambda)$**:** Sample $\mathbf{A} \hookleftarrow U(\mathbb{Z}_q^{n\times n})$, $\mathbf{S}, \mathbf{E} \hookleftarrow U((-B,B]^{n\times n})$. Return $vk = (\mathbf{A}, \mathbf{T} = \mathbf{AS} + \mathbf{E}, H)$ with H a hash function modeled as a random oracle with values in $\{0,1\}^{n\times n}$ and $sk = (\mathbf{S}, \mathbf{E})$.

$\mathsf{Sign}(sk, \mu)$**:** To sign a message $\mu \in \{0,1\}^\star$, do the following:

1. (Commit) Sample $\mathbf{S}', \mathbf{E}' \hookleftarrow U((-B,B]^{n\times n})$ and compute $\mathbf{T}' = \mathbf{AS}' + \mathbf{E}'$;
2. (Challenge) Compute $\mathbf{C} = H(\mu || \mathbf{T}')$;
3. (Response) Compute $\mathbf{Z}_1 = \mathbf{S}' + \mathbf{SC}, \mathbf{Z}_2 = \mathbf{E}' + \mathbf{EC}$.
4. Output $(\mathbf{C}, \mathbf{Z}_1, \mathbf{Z}_2)$.

$\mathsf{V}(vk, (\mathbf{C}, \mathbf{Z}_1, \mathbf{Z}_2), \mu)$**:** To verify the signature:

1. Compute $\mathbf{T}' = \mathbf{AZ}_1 + \mathbf{Z}_2 - \mathbf{TC}$;
2. Accept if $\mathbf{C} = H(\mu || \mathbf{T}')$.

1. Would the scheme be euCMA secure if $\mathbf{S}$ and $\mathbf{E}$ were sampled as vectors in $(-B,B]^n$ instead of $(-B,B]^{n\times n}$?

   ☞ With the latter condition, c is now sampled as a scalar, we can sample random messages and randomness $\mathbf{s}', \mathbf{e}'$ until the hash is 0: then we do not need to know the secret key to sign the message. This could be circumvented using c from a exponentially large space, but this causes other problems as we will see below.

2. Is it hard to find a forgery for this scheme?

☞ A valid signature is a triplet $(\mathbf{C}, \mathbf{Z}_1, \mathbf{Z}_2)$ such that $\mathbf{T}' = \mathbf{A}\mathbf{Z}_1 + \mathbf{Z}_2 - \mathbf{T}\mathbf{C}$ and $\mathbf{C} = H(\mu || \mathbf{T}')$.

Let μ be a message to sign. One can simply sample $\mathbf{T}'$ at random, compute $\mathbf{C} = H(\mu || \mathbf{T}')$, sample $\mathbf{Z}_2$ at random, and compute $\mathbf{Z}_1 \leftarrow \mathbf{A}^{-1}(\mathbf{T}' - \mathbf{T}\mathbf{C} - \mathbf{Z}_2)$. Then $(\mathbf{C}, \mathbf{Z}_1, \mathbf{Z}_2)$ is a valid signature for μ.

Alternative solution: Given a signature $(\mathbf{C}, \mathbf{Z}_1, \mathbf{Z}_2)$ for a message M, one could choose some matrix $\mathbf{S}''$ and compute $\mathbf{Z}_1' = \mathbf{Z}_1 + \mathbf{S}''$ and $\mathbf{Z}_2' = \mathbf{Z}_1 - \mathbf{A}\mathbf{S}''$. Then it holds that $\mathbf{T}' = \mathbf{A}\mathbf{Z}_1' + \mathbf{Z}_2' - \mathbf{T}\mathbf{C}$ and then the signature $(\mathbf{C}, \mathbf{Z}_1', \mathbf{Z}_2')$ is accepted for the message M.

3. Explain how to tweak to above scheme to prevent the above attack by imposing an additional validity check regarding the norm of $\mathbf{Z}_1, \mathbf{Z}_2$. Show that the secret key owner can still sign.

☞ If we require $\mathbf{Z}_1, \mathbf{Z}_2$ to be small, it is hard to find solutions as above as the inverse we compute in the first solution will be large w.o.p., and it is hard to find $\mathbf{S}''$ such that $\mathbf{A}\mathbf{S}''$ is small for the second attack.

4. Show that if the resulting scheme is still not euCMA secure.

☞ We focus on $\mathbf{Z}_1$ to recover $\mathbf{S}$ but the same holds for $\mathbf{Z}_2$.

Let $(\mathbf{C}_i, \mathbf{Z}_i = (\mathbf{s}_i' \quad \mathbf{S}_{i,R}') + (\mathbf{s}_1 \quad \mathbf{S}_\mathbf{R}) \cdot (\mathbf{c}_i \quad \mathbf{C}_i'))$ be the answer to the i-th signing query, where we explicit the first columns $\mathbf{s}_i', \mathbf{s}_1, \mathbf{c}_i$ of each matrix $\mathbf{S}_i', \mathbf{S}, \mathbf{C}_i$.

Then, assuming that $\mathbf{C}_i[1,1] = 1$, it holds that the first column of $\mathbf{Z}_i$ is $\mathbf{s}_i' + \mathbf{s}_1 + \mathbf{S}'\mathbf{c}_i'$ with $\mathbf{c}_i = \begin{pmatrix} 1 \\ \mathbf{c}_i' \end{pmatrix}$.

We query signatures for random messages M until we get k signatures with non-zero topleft value of $\mathbf{C}_i$. Then, if we compute the mean of all leftmost column, it holds that

$$\Pr(|\bar{\mathbf{z}}_1[j] - \mathbf{s}_1[j]| \geq 1) \leq 2\exp\left(-\frac{2k}{q^2}\right),$$

for each index j of the column. If this event is not satisfied, since we work with integers, we can round the mean to the nearest integer and get $\mathbf{s}_1[j]$, which means that we have a non-negligible probability of finding $\mathbf{S}$ if we repeat this for each column, for big enough k (but still polynomial).

The solution to make the scheme secure is to force the distribution of $\mathbf{Z}_1, \mathbf{Z}_2$ to be independent of the secret key. This is done by a technique called rejection sampling, or by noise flooding.

Exercise 3.
In this exercise, we assume we have two cyclic groups G and G_T of the same known prime cardinality p, and a generator g of G. We also assume we have a pairing function $e : G \times G \to G_T$, with the following properties: It is non-degenerate, i.e., $e(g,g) \neq 1$; It is bilinear, i.e., $e(g^a, g^b) = e(g,g)^{ab}$ for all $a, b \in \mathbb{Z}/q\mathbb{Z}$; It is computable in polynomial-time. Note that the bilinearity property implies that $e(g^a, g) = e(g, g^a) = e(g,g)^a$ holds for all $a \in \mathbb{Z}/p\mathbb{Z}$.

1. Show that the Decision Diffie-Hellman problem (DDH) on G can be solved in polynomial-time.

☞ Given g^a, g^b and g^c, test whether $e(g^a, g^b) = e(g^c, g)$. If $c = ab$, then equality holds. If c is uniform, then $e(g^c, g)$ is uniform, and as the pairing is non-degenerate, equality holds with probability $1/p$.

We consider the BLS signature scheme (due to Boneh, Lynn and Shacham), which is as follows:

- `KeyGen` takes as inputs a security parameter and returns G, g, p, G_T and a description of $e : G \times G \to G_T$ satisfying the properties above. All these are made publicly available. Sample x uniformly in $\mathbb{Z}/p\mathbb{Z}$. The verification key is $vk = g^x$, whereas the signing key is $sk = x$.
- `Sign` takes as inputs sk and a message $M \in \{0,1\}^*$. It computes $h = H(M) \in G$ where H is a hash function, and returns $\sigma = h^x$.
- `Verify` takes as inputs the verification key $vk = g^x$, a message M and a signature σ, and returns 1 if and only if $e(\sigma, g) = e(H(M), vk)$.

2. Show that this signature scheme is EU-CMA secure under the Computational Diffie-Hellman assumption (CDH) relative to G, when $H(\cdot)$ is modeled as a random oracle. ☞ Let $\mathcal{A}$ be an adversary against the EU-CMA seucrity of the signature scheme. Let Q be an upper bound on the number of (unique) random oracle queries made by $\mathcal{A}$. We build the following reduction $\mathcal{B}$:

$\mathcal{C}$		$\mathcal{B}$		$\mathcal{A}$
Sample $a, b \hookleftarrow U(\mathbb{Z}_p)$	$\xrightarrow{(g^a, g^b)}$	Set $vk = g^a$ Sample $i^\star \hookleftarrow U([1, Q])$, $i = 1$	$\xrightarrow{vk}$	
			$\xleftarrow{\text{“}RO\text{''}, M}$	
		If $H(M)$ was not set: Set $H(M) = g^{r_M}, r_M \hookleftarrow U(\mathbb{Z}_p)$ if $i \neq i^\star$ Set $H(M) = g^b$ otherwise		
		Increase i by one	$\xrightarrow{H(M)}$	
			$\xleftarrow{\text{“}Sign\text{''}, M}$	
		If not set, set $H(M) = g^{r_M}, r_M \hookleftarrow U(\mathbb{Z}_p)$	$\xrightarrow{(vk)^{r_M}}$	
			$\xleftarrow{\sigma, M}$	
	$\xleftarrow{\sigma}$	If $\texttt{Verify}(vk, M, \sigma)$ and $H(M) = g^b$ Else abort.		

If both conditions at the end are verified, it holds that $e(g^b, g^a) = e(\sigma, g)$, meaning that $\sigma = g^{ab}$, and we win. The answers of $\mathcal{B}$ to RO queries are well distributed, as g is a generator of G of order p.

The answers of $\mathcal{B}$ to signing queries are also well simulated, except if $\mathcal{A}$ queries M such that we set $H(M) = g^b$. In that case, since sign is deterministic, even if we could answer the query correctly, we would fail at the end (ie never get a forgery for M). So in that case, we can abort. Note that outputting a valid forgery for a message M without querying the Random Oracle first is highly improbable, as $H(M)$ is not yet set: the adversary only has probability $1/p$ to guess the correct value of $H(M)$, which is negligible.

Then, assuming that $\mathcal{A}$ has non-negligible probability of winning, it has non-negligible probability of winning by forging a signature for a message it queried the RO for. Since we try to guess which message will be attacked, it holds that $\mathrm{Adv}(\mathcal{B}) \geq \Pr(\mathcal{A}$ wins with a forgery queried to the RO$)/Q$, which is still non-negligible.

In cryptographic applications in which signing is performed very frequently (such as for cryptocurrencies), it is interesting to aggregate many signatures for multiple messages into significantly smaller space than required to store all these signatures.

3. Show that that the BLS signature scheme supports aggregation. ☞

If we have two signature σ_1, σ_2 for messages M_1, M_2 respectively, we can compute their product $\sigma' = \sigma_1 \sigma_2$. To verify that this aggregation is valid, one can check that $(H(M_1)H(M_2))^x = \sigma'$.

We store m signatures in only 1 element of G. Of course, this comes at the price of security.

4. Propose formal definitions for the functionality and security of an aggregate signature scheme. ☞ An aggregate signature scheme is a tuple $\texttt{Gen}, \texttt{Sign}, \texttt{Aggregate}, \texttt{Verify}$ such that

$\texttt{Gen}(1^\lambda)$: Outputs (vk, sk), a verification and secret keys.

$\texttt{Sign}(sk, M)$: Outputs σ, a signature for message M.

$\texttt{Aggregate}(\{\sigma_i, M_i\}_i, vk)$: Outputs σ', an aggregated signature for $\{M_i\}_i$. In particular, σ' must have smaller size than $\{\sigma_i\}_i$.

$\texttt{Verify}(\sigma', \{M_i\}_i, vk)$: Outputs 1 if σ' is an aggregated (or, if there is only one message, simply a) signature for $\{M_i\}_i$. Outputs 0 otherwise.

An aggregated signature scheme is secure if no adversary with access to a signing oracle can forge a valid aggregated signature, such that at least one message was not queried to the signing oracle.