TD 9: Digital Signatures

Exercise 1. *Random Messages and Signatures*

A signature scheme (KeyGen, Sign, Verify) is existentially unforgeable under random-message attack (euRMA-secure) if for all ℓ polynomially bounded with respect to the security parameter n, and all probabilistic polynomial-time adversary $\mathcal{A}$, the success probability of $\mathcal{A}$ in the following game is negligible (as a function of n).

- The challenger runs KeyGen to obtain a key-pair (sk, vk). It samples ℓ messages $m_1, \dots, m_\ell$ uniformly from the (finite) set of messages. It computes $\sigma_1 \leftarrow \texttt{Sign}(sk, m_1), \dots, \sigma_\ell \leftarrow \texttt{Sign}(sk, m_\ell)$. It sends $vk, (m_1, \sigma_1), \dots, (m_\ell, \sigma_\ell)$ to the adversary.
- After receiving the latter from the challenger, the adversary produces a pair (m, σ).
- The adversary succeeds if $m \notin \{m_1, \dots, m_\ell\}$ and if $\texttt{Verify}(vk, m, \sigma) = 1$.

1. Show that an euRMA-secure signature scheme may not be euCMA-secure (existentially unforgeable under Chosen Message Attacks).

The goal of the exercise is to show that an euCMA-secure signature scheme Σ' =(KeyGen', Sign', Verify') for messages in $\{0,1\}^n$ may be built from an euRMA-secure signature scheme Σ =(KeyGen, Sign, Verify) for messages in $\{0,1\}^{2n}$. We consider the following algorithms KeyGen' and Sign'.

- Algorithm KeyGen' runs KeyGen twice, and obtains $(sk^{(0)}, vk^{(0)})$ and $(sk^{(1)}, vk^{(1)})$. It sets $sk' = (sk^{(0)}, sk^{(1)})$ and $vk' = (vk^{(0)}, vk^{(1)})$.
- On inputs $m \in \{0,1\}^n$ and $sk' = (sk^{(0)}, sk^{(1)})$, algorithm Sign' samples $r \in \{0,1\}^n$ and $m^{(0)} \in \{0,1\}^n$ uniformly and independently, sets $m^{(1)} = m^{(0)} \oplus m$, computes $\sigma^{(0)} \leftarrow \texttt{Sign}(sk^{(0)}, r \| m^{(0)})$ and $\sigma^{(1)} \leftarrow \texttt{Sign}(sk^{(1)}, r \| m^{(1)})$, and returns $\sigma' = (r, m^{(0)}, m^{(1)}, \sigma^{(0)}, \sigma^{(1)})$.

2. Propose an algorithm Verify' such that Σ' is correct and not trivially insecure.

3. Show that if r was always set to 0 (instead of being sampled uniformly for every new signature), then Σ' would not be euCMA-secure. Show that if the same r has been sampled twice, an adversary is able to produce signature forgeries.

We now consider a probabilistic polynomial-time euCMA-adversary $\mathcal{A}'$ on Σ', and aim at showing that its success probability is negligible (under the assumption that Σ is euRMA-secure). Let ℓ be the maximum number of signature queries that $\mathcal{A}$ makes. Let $m_1, \dots, m_\ell \in \{0,1\}^n$ be the messages submitted by $\mathcal{A}'$ to the challenger. Let $(r_i, m_i^{(0)}, m_i^{(1)}, \sigma_i^{(0)}, \sigma_i^{(1)})$ be the signature the challenger computes for message m_i. We let $(m_*, \sigma_*) = (m_*, (r_*, m_*^{(0)}, m_*^{(1)}, \sigma_*^{(0)}, \sigma_*^{(1)}))$ denote the forgery produced by the adversary. We define the following events:

- Repeat: "There exist $i \neq j$ such that $r_i = r_j$."
- Forge$^{(0)}$: "$\texttt{Verify}(vk^{(0)}, r_* \| m_*^{(0)}, \sigma_*^{(0)}) = 1$ and $r_* \| m_*^{(0)} \notin \{r_i \| m_i^{(0)} : i \leq \ell\}$."
- Forge$^{(1)}$: "$\texttt{Verify}(vk^{(1)}, r_* \| m_*^{(1)}, \sigma_*^{(1)}) = 1$ and $r_* \| m_*^{(1)} \notin \{r_i \| m_i^{(1)} : i \leq \ell\}$."

4. Show that if $\mathcal{A}'$ succeeds, then at least one of Repeat, Forge$^{(0)}$ and Forge$^{(1)}$ occurs.

5. Give an upper bound on the probability of Repeat.

6. Show that if the probability that $\text{FORGE}^{(0)}$ is non-negligible, then there exists a probabilistic polynomial-time adversary $\mathcal{A}$ against euRMA-security of Σ.

7. Conclude.

Exercise 2. *Binary Trees and Signatures*

The notion of existential unforgeability under single-message attack for a signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{V})$ states that no adversary can output a valid tuple (m', σ) with non-negligible probability by only querying once the signing oracle for m with $m \neq m'$.

The goal of this exercise is to go from euSMA-security to euCMA-security. The idea is, for each bit of the message, to generate two new public keys, sign them using the public key from the previous bit, and use one of them for the next bit (depending on the value of the current bit). This can be seen as building a binary tree.

Let F be a secure PRF. It will come in handy to make sure we use the same randomness to generate the keys (as we do not have memory to store them, from one signature to the next one).

We assume the following about the PRF: its output is long enough to be given to Gen as randomness seed, and there is some one-to-one deterministic padding in the case where the input is too small.

Here is the construction, where $m_{|i}$ denotes the first i bits of m and $m_{|0}$ is the empty word ε:

$\mathsf{Gen}^\star(1^\lambda)$**:** Generate $(vk_\varepsilon, sk_\varepsilon) \leftarrow \mathsf{Gen}(1^\lambda)$ and two PRF keys k, k'. Return $vk = vk_\varepsilon$ and $sk = (sk_\varepsilon, k, k')$.

$\mathsf{Sign}^\star(sk, m)$**:** For $i = 0$ to n do the following: Compute $r_{m_{|i}0} := F(k, m_{|i}0)$, and $r_{m_{|i}1} := F(k, m_{|i}1)$.

Then generate $vk_{m_{|i}1}, sk_{m_{|i}1} \leftarrow \mathsf{Gen}(1^\lambda; r_{m_{|i}0})$ and $vk_{m_{|i}0}, sk_{m_{|i}0} \leftarrow \mathsf{Gen}(1^\lambda; r_{m_{|i}1})$. Then, sign $\sigma_{m_{|i}} \leftarrow \mathsf{Sign}(sk_{m_{|i}}, (\mathsf{vk}_{m_{|i}0}, \mathsf{vk}_{m_{|i}1}); r'_{m_{|i}})$, where $r'_{m_{|i}} \leftarrow F(k', m_{|i})$.

Compute $\sigma_m \leftarrow \mathsf{Sign}(\mathsf{sk}_m, m; F(k', m))$.

Then, return $\Big(\{\sigma_{m_{|i}}, vk_{m_{|i}0}, vk_{m_{|i}1}\}_i, \sigma_m\Big)$.

1. Give a verification algorithm $\mathsf{V}^\star$. How many times does it call V, depending on the message size? How many public keys are manipulated (i.e. generated, used to sign or signed) during one call to $\mathsf{Sign}^\star$?

In order to prove the euCMA-security of this scheme, we introduce the following hybrid H_1: the game is the same as in the euCMA setup (we will call it H_0), except that $F(k, \cdot)$ is replaced by a truly random function, whose table is built adaptively.

2. Show that H_0 and H_1 are indistinguishable.

Then we introduce H_2, which is as H_1 except that this time $F(k', \cdot)$ is replaced by a truly uniform function, whose table is also built adaptively.

3. Show that H_1 and H_2 are indistinguishable.

4. Show that under the euSMA security of the base signature, no adversary has non-negligible advantage in the game H_2.

5. Conclude.

Exercise 3. *Random Oracle Model*

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.

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