TD 9: Digital Signatures (corrected version)

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).

   ☞

   Consider any signature scheme (K, S, V) on the message space $\mathcal{M} = \mathbb{Z}/N\mathbb{Z}$. Then, let $\hat{S}$ be the signature $m \mapsto (S(m), S(m+1))$ (with key generator K and obvious signature verifier).

   Notice that $\hat{S}$ is not euCMA-secure: an adversary $\mathcal{A}$ can ask the signature of m and $m+2$, obtaining $S(m), S(m+1), S(m+2), S(m+3)$, and forging the valid signature $(S(m+1), S(m+2))$ for $m+1$.

   However $\hat{S}$ is euRMA-secure. Having $\hat{S}$-signatures $(m_1, \sigma_1), \dots, (m_q, \sigma_q)$ for random messages $m_1, \dots, m_q$ just means having S-signatures $(m_1, S(m_1)), (m_1+1, S(m_1+1)), \dots, (m_q, S(m_q)), (m_q+1, S(m_q+1))$. Thus, except in the case where $m_j = m_i + 2$ for some $i \neq j$, which is unlikely, forging a new signature for $\hat{S}$ is not easier than forging a new signature for S.

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.

   ☞

   Algorithm Verify' is defined as follows. On input (m, σ'), parse σ' as $(r, m^{(0)}, m^{(1)}, \sigma^{(0)}, \sigma^{(1)})$. Then output 1 if and only if $m = m^{(0)} \oplus m^{(1)}$ and $\texttt{Verify}(vk^{(0)}, (r \| m^{(0)}, \sigma^{(0)})) = 1$ and $\texttt{Verify}(vk^{(1)}, (r \| m^{(1)}, \sigma^{(1)})) = 1$.

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.

   ☞

   If r is always 0. Choose any message m and ask for its signature $(0, m^{(0)}, m^{(1)}, \sigma^{(0)}, \sigma^{(1)})$. Then apply the same strategy as in the next paragraph.

   If the same r has been sampled twice. The adversary has two signatures with the same r: $(r, m_0^{(0)}, m_0^{(1)}, \sigma_0^{(0)}, \sigma_0^{(1)})$ for a message m_0 and $(r, m_1^{(0)}, m_1^{(1)}, \sigma_1^{(0)}, \sigma_1^{(1)})$ for a message m_1. Then a valid signature for the message $m_0^{(0)} \oplus m_1^{(1)}$ (which is m_0 or m_1 with negligible probability $1/2^{n-1}$) is $(r, m_0^{(0)}, m_1^{(1)}, \sigma_0^{(0)}, \sigma_1^{(1)})$.

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 $m_1, \ldots, m_\ell \in \{0,1\}^n$ 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.

☞

We assume that Repeat does not occur, and we will show that either Forge$^{(0)}$ or Forge$^{(1)}$ occurs. Since $\mathcal{A}'$ succeeds, we have $\texttt{Verify}(vk^{(0)}, r_* \| m_*^{(0)}, \sigma_*^{(0)}) = 1$ and $\texttt{Verify}(vk^{(1)}, r_* \| m_*^{(1)}, \sigma_*^{(1)}) = 1$. And since Repeat does not occur, we have $r_* = r_i$ for at most one value of i.

Case 1: $r_* \neq r_i$ for all i. Then we have both Forge$^{(0)}$ and Forge$^{(1)}$.

Case 2: $r_* = r_i$ for some (unique) i. If both $m_*^{(0)} = m_i^{(0)}$ and $m_*^{(1)} = m_i^{(1)}$, then we have $m_* = m_i$, which is not possible. So we have either $m_*^{(0)} \neq m_i^{(0)}$ (and Forge$^{(0)}$ occurs) or $m_*^{(1)} \neq m_i^{(1)}$ (and Forge$^{(1)}$ occurs).

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

☞

This is a birthday paradox instance: we have ℓ values $r_1, \ldots, r_\ell$ sampled uniformly and independently from the set $\{0,1\}^n$ of size 2^n. Repeat is the event that two of these values are equal, which has probability at most $\ell^2/2^{n+1}$ (note that this is negligible in n).

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

☞ We reduce to the unforgeability of Σ. Let $\mathcal{A}$ that does the following. On input a verification key vk and ℓ signatures $\sigma_1, \ldots, \sigma_\ell$ for messages $m'_1, \ldots, m'_\ell$, it sets $vk^{(0)} := vk$ and runs the keygen to get $vk^{(1)}, sk^{(1)}$. It then calls $\mathcal{A}'$ on $vk^{(0)}, vk^{(1)}$. Everytime $\mathcal{A}'$ requests a signature for message m_i, let r_i be the first n bits of m'_i and $m_i^{(0)}$ the last n bits of m'_i. Then set $\sigma_i^{(0)} = \sigma_i$, $m_i^{(1)} = m_i^{(0)} \oplus m_i$ and $\sigma_i^{(1)} \leftarrow \mathsf{Sign}(sk^{(1)}, r_i || m_i^{(1)})$. Finally, return the signature $(r_i, m_i^{(0)}, m_i^{(1)}, \sigma_i^{(0)}, \sigma_i^{(1)})$. Eventually, when $\mathcal{A}'$ outputs a forgery $m, (r_\star, m_\star^{(0)}, m_\star^{(1)}, \sigma_\star^{(0)}, \sigma_\star^{(1)})$, output $r_\star || m_\star^{(0)}, \sigma_\star^{(0)}$.

Indeed, if Forge$^{(0)}$ happens, then this is a valid forgery, by definition of the event. Then under the euRMA-security of the Σ signature scheme, this event has negligible probability.

Note that this can be adapted to swap the role of 0 and 1 and this also proves that Forge$^{(1)}$ has negligible probability of happening.

7. Conclude.

☞

The probability that $\mathcal{A}'$ succeeds is bounded from above by $\Pr[\text{Repeat} \cup \text{Forge}^{(0)} \cup \text{Forge}^{(1)}] \leq \Pr[\text{Repeat}] + \Pr[\text{Forge}^{(0)}] + \Pr[\text{Forge}^{(1)}]$. Since Σ is assumed to be euRMA-secure, from the questions above we obtain that this is negligible.

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 $\mathsf{Sign}^\star$? ☞ $\mathsf{V}^\star(vk, m, \sigma)$: Check that all signatures in σ are consistent with the message they sign.

i.e. check $\mathsf{V}(\mathsf{vk}_m, m, \sigma_m)$ and $\mathsf{V}(\mathsf{vk}_{m_{|i}}, \sigma_{m_{|i}b}, \mathsf{vk}_{m_{|i}b})$ for both $b \in \{0,1\}$ and all $i = 0$ to $n-1$.

Return 1 if and only if all signatures are accepted by V.

In total, we call V $2n+1$ times and we manipulate $2n+1$ public keys: vk_ε and every $\mathsf{vk}_{m_{|i}b}$.

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.

☞ Assuming that we have a distinguisher $\mathcal{A}$ between H_0 and H_1, an attacker against the security of the PRF can run $\mathsf{Gen}^\star(1^\lambda)$, discard k and replace it with the PRF oracle from its challenger. Then, it calls $\mathcal{A}$ and answers its signing queries using the secret key from $\mathsf{Gen}^\star$ and when the PRF oracle. In the case where the PRF oracle is the PRF, then $\mathcal{A}$ is in the setup of H_0, and when the oracle is truly random, then $\mathcal{A}$ is in the setup of H_1. This means that $\mathcal{A}$ and this new attacker have the same advantage if we set the attacker to return the same bit as $\mathcal{A}$.

Then under the security of the PRF, these two games 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.

☞ We apply the same kind of reduction as in the previous question. This time, the difference is that calls to $F(k', \cdot)$ are replaced with calls to the PRF oracle. As previously, under the security of the PRF, the two games are inditinguishable.

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

☞ Let $\mathcal{A}^\star$ be an adversary against H_2. Let $Q^\star$ be an upper bound on the number of signing queries from $\mathcal{A}^\star$.

As we have shown in question 1, excluding the base key vk_ε, at most $2n$ keys are manipulated for a signature. Let $Q = 2nQ^\star + 1$ be an upper bound on the number of pairs (vk, sk) needed to answer all of $\mathcal{A}^\star$'s signing queries. Note that this number is still polynomial: the idea is then to guess for which key there will be a valid forgery done, i.e. finding the weakest point of the tree.

Let $\mathcal{A}$ be an adversary against the euSMA security of the base scheme that does the following on input vk.

1. Choose $i^\star \hookleftarrow U([1, Q])$ and set $\mathsf{vk}^{i^\star} = \mathsf{vk}$.
2. For $i = 0$ to Q, $i \neq i^\star$, sample $(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^\lambda)$.
3. Call $\mathcal{A}^\star$ on $\mathsf{vk}_\varepsilon = \mathsf{vk}^1$.
4. For each signing query m of $\mathcal{A}^\star$, do the following for $j = 0$ to $n-1$. If $\mathsf{vk}_{m_{|j}1}, \mathsf{vk}_{m_{|j}0}$ and $\sigma_{m_{|j}}$ were not set, then set the first two to be the two next unused keys vk^i and vk^{i+1}. Then compute $\sigma_{m_{|j}} \leftarrow \mathsf{Sign}(\mathsf{sk}_{m|j}, \mathsf{vk}^i || \mathsf{vk}^{i+1})$ (up to calling the signing oracle instead). Then if σ_m is not yet defined, compute a signature $\sigma_m \leftarrow \mathsf{Sign}(\mathsf{sk}_m, m)$ (up to calling the signing oracle instead). Return the full signature $\sigma_m^\star$ to $\mathcal{A}^\star$.
5. When $\mathcal{A}^\star$ outputs a valid forgery $m, \sigma'_m, \{\sigma_{m_{|j}}, \mathsf{vk}_{m_{|j}0}, \mathsf{vk}_{m_{|j}1}\}$, in particular m was never queried as there is only one signature per message, we do the following:
   - If there exists some minimal $j^\star$ such that $\mathsf{vk}'_{m_{|j^\star}b} \neq \mathsf{vk}_{m_{|j^\star}b}$ (or it was not defined by $\mathcal{A}$), then it holds that there exists some i, by minimality of $j^\star$ such that $\mathsf{vk}^i = \mathsf{vk}_{m_{|j}} = \mathsf{vk}'_{m_{|j}}$. Then, if $i = i^\star$, output $(\mathsf{vk}'_{m_{|j}0} || \mathsf{vk}'_{m_{|j}1}, \sigma'_{m_{|j}})$.
   - Otherwise, it holds that $\mathsf{vk}_m = \mathsf{vk}'_m = \mathsf{vk}^i$ for some i (and no signature except σ'_m is a forgery). If $i = i^\star$, return (m, σ'_m).

Note the following: the fact that we reuse the public keys from one message to another does not change $\mathcal{A}^\star$'s view: actually, this is what is done in the sign protocol as the same randomness is used each time. Since we have sampled enought public keys at the beginning, it holds that $\mathcal{A}^\star$'s view is the same as in H_2. Thus $\mathcal{A}^\star$ finds a forgery in this simulation with the same probability as in H_2.

Moreover, if it finds a forgery, either of the two aforementioned cases arise, and in both cases we return a valid forgery. This concludes the reduction.

5. Conclude.

☞ Let an adversary $\mathcal{A}$ against H_2. A distinguisher $\mathcal{B}$ between H_1 and H_2 can call $\mathcal{A}$ and return H_2 if and only if $\mathcal{A}$ returns a valid forgery. Then the advantage of $\mathcal{A}$ against H_1 is at most its advantage in H_2 plus the advantage of $\mathcal{B}$.

Then $\mathcal{A}$ still has negligible advantage against H_1. The same reduction holds for H_0: any ppt adversary has negligible advantage against H_0. This concludes the proof of the euCMA-security of this scheme.

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.

☞ 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 ranom 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.