TD 3 – Message authentication codes

Exercise 1. *Suffix-MAC*

Let $H : \{0,1\}^* \to \{0,1\}^n$ be a Merkle-Damgård hash function. Define $\mathsf{SuffixMac}_H : \{0,1\}^\kappa \times \{0,1\}^* \to \{0,1\}^n$ by $\mathsf{SuffixMac}_H(k,m) = H(m\|k)$.

1. **i.** What is the (generic) complexity of finding a collision for (m,m') for H?
   ii. Does the complexity changes if one requires m and m' to be of the same length $\ell > n$?
2. Let (m,m') be a colliding pair for H, with m and m' having the same length.
   i. Give an existential forgery attack for $\mathsf{SuffixMac}_H$ with query cost 1.
   ii. What is the total cost of the attack, if one has to compute (m,m')?
   iii. Is the attack interesting if $\kappa = n/2$? And if $\kappa = n$?

Exercise 2. *CBC-MAC variant*

We recall that CBC-MAC uses a block cipher E, and computes a MAC as follows: Write the input message $m = m_1\|\cdots\|m_B$ and prepend it with one block m_0 encoding the length of m. Then compute $t_0 = E(k,m_0)$ and for $i > 0$, $t_i = E(k, m_i \oplus t_{i-1})$. Finally, output t_B. The main drawback of this (secure) method is that prepending the length requires to know the length in advance. In other words, one cannot begin the computation before getting the full message.

We study a variants of CBC-MAC, and their securities. For each question, Mac denote the current variant, and m_1 and m_2 are two message blocks. We let n be the block length.

1. The first variant simply removes the block m_0 containing the length of m.
   i. Compute an explicit expression for $t_1 = \mathsf{Mac}(m_1)$.
   ii. Compute an explicit expression for $t_2 = \mathsf{Mac}(m_1\|m_2)$, in terms of t_1.
   iii. How can we choose m_2 to get a two-block message with tag t_1?
   iv. Describe an existential forgery attack for Mac. What is its query and time cost?
2. The second variant put the block containing the bit-length as the last block. We still denote the variant by Mac.
   i. Compute an explicit expression for $t_1 = \mathsf{Mac}(m_1)$.
   ii. Compute an explicit expression for $s = \mathsf{Mac}(m_1\|\langle n\rangle\|t)$. Does it depend on m_1?
   iii. Describe an existential forgery attack for Mac, where the attacker requests the t and s as above, as well as another tag $t_2 = \mathsf{Mac}(m_2)$.
3. The third variant does not includes the length of m, but encrypts the last tag with an independent key k': Let $\mathsf{Mac}'((k,k'),m) = E(k', \mathsf{Mac}(k,m))$. Explain (roughly) why the previous attacks are avoided in this solution.