Problem sheet 6 for Course 02231, 2025

These practice problems have the purpose of helping you understand the material better and learning the skills that are necessary to analyze cryptographic constructions, and sometimes to prepare you for the next class. All answers should be supported by a written justification. To gauge whether a justification is sufficient, ask yourself if your peers would be convinced by it without additional explanations.

We denote vectors as $\mathbf{x} \in \{0,1\}^\lambda$. By $\mathbf{x}[i]$ we denote the ith index of $\mathbf{x}$, where $i \in \{1,\ldots,\lambda\}$. As in the lecture, we write $k \leftarrow K$ if k is sampled from the set K such that it can be each element from K with equal probability $1/|K|$. For two strings x, y we use $x|y$ to denote the string obtained from concatenating x with y.

Exercise 1. (**Length extension attacks for Merkle-Damgård**)

Let $h : \{0,1\}^{\ell+\lambda} \to \{0,1\}^\lambda$ be a collision-resistant compression function.
In the class, we mentioned the Merkle-Damgård construction for hashes. Let us quickly recap how it works:

$MDPad_\ell(m)$**:**

1. Let e be $|m|$ in binary form, where e consists of ℓ bits.
2. While $|m|$ is not a multiple of ℓ: $m := m|0$.
3. Output $m|e$

$MDHash(m)$**:**

1. Let $m_1,\ldots,m_\tau = MDPad_\ell(m)$ where m_i has length ℓ bits.
2. Set $y_0 := 0^\lambda$.
3. For $i = 1,\ldots,\tau$: $y_i = h(m_i|y_{i-1})$.
4. Output y_τ.

1. Consider that we want to use the Merkle-Damgård construction as a MAC. For this, we redefine $MDHash(m)$ (which samples $y_0 \in \{0,1\}^\lambda$) as $MDMAC(k,m)$ which works like $MDHash(m)$, but instead sets $y_0 := k$. Show that given $(m, MDMac(k,m))$ it is easy to find (m', t') such that $t' = MDMac(k,m')$ where $m \neq m'$.

   Hint: The easiest way is to derive m' from m by making it a little bit longer. Maybe the function $MDPad_\ell$ can help you...

❓ Exercise 2. (Simplify Merkle-Damgård?)

Let $h : \{0,1\}^{\ell+\lambda} \to \{0,1\}^{\lambda}$ be a collision-resistant compression function.
Let us define a simplified version of Merkle-Damgård as follows:

$SMDPad_\ell(m)$**:**

1. If $|m| \leq \ell + \lambda$ then output $m|0^{\ell+\lambda-|m|}$.
2. Else Let $m = m_1|m_2$ where m_1 is of length λ.
3. While $|m_2|$ is not a multiple of ℓ: $m_2 := m_2|0$.
4. Output $m_1|m_2$

$SMDHash(m)$**:**

1. Let $m_1, \ldots, m_\tau = SMDPad_\ell(m)$ where m_1 has length λ bits and $m_2, \ldots, m_\tau$ ℓ bits
2. $y_1 = m_1$
3. For $i = 2, \ldots, \tau$: $y_i = h(m_i|y_{i-1})$.
4. Output y_τ.

1. Show that this simplified construction of Merkle-Damgård is not collision-resistant and construct two inputs m, m' such that $m \neq m'$ while $SMDHash(m) = SMDHash(m')$.

   Hint: m and m' do not have to be of the same length. If m is the longer message, then a starting point might be to choose a part of m' based on y_2 that is generated during $SMDHash(m)$...

❓ Exercise 3. (No homomorphism in Collision-resistant hashing)

Let $H : \{0,1\}^* \to \{0,1\}^{\lambda}$ be a function such that for all strings x, y of identical length, it holds that $H(x) \oplus H(y) = H(x \oplus y)$. This means that computing the hash of the coordinate-wise XOR of x and y is the same as hashing x and y separately using H and XOR-ing the resulting strings. Cryptographers would refer to such a function H as being *homomorphic*.

1. Show that H cannot be collision-resistant, by constructing an adversary that can distinguish $L^H_{CR-real}$ from $L^H_{CR-fake}$ (as defined in the lecture) with probability essentially 1.

   Hint: Consider the values $H(1) \oplus H(1)$ or $H(11) \oplus H(11)$...

Exercise 4. (**PRPs and collision resistance**)

Let $F : \{0,1\}^\lambda \times \{0,1\}^B \to \{0,1\}^B$ be a pseudorandom permutation on blocks of length B. Define the function $H : \{0,1\}^{\lambda+B} \to \{0,1\}^B$ as $H(x|y) := F(x,y)$ where x is of length λ and y of length B. This means that x is the key of the PRP F while y is the input to the permutation.

1. Show that H cannot be collision-resistant, by constructing an adversary that can distinguish $L^H_{CR-real}$ from $L^H_{CR-fake}$ (as defined in the lecture) with probability essentially 1.

   Hint: F is a pseudorandom permutation. Which operations are efficiently computable according to the definition?