Introduction to Cryptography: Homework 6

October 20, 2021

Requirements about the delivery of this assignment:

- Submit a pdf-document via Brightspace;
- Make sure that you write both name and student number on all documents (not only in the file name).

Deadline: Monday, November 15, 17:00 sharp!

Grading: You can score a total of 100 points for the hand-in assignments. To get full points, please **explain all answers clearly**.

Exercises:

1. **Brightspace quiz.** Make the Brightspace quiz on groups, "Terminology and definitions I". Go to Activities -> Quizzes. [You can attempt the quiz as often as you would like/need.]

2. **Security for sponge-based hash functions.** We build a hash function using a sponge function, where the permutation KECCAK-f is used. KECCAK-f can have values b for the permutation width $b \in \{25, 50, 100, 200, 400, 800, 1600\}$. Give (with explanation) the minimum values for b, the rate r, the capacity c and the output-length n, if we want *at least*

   (a) 50 bits of security against collision attacks;

   (b) 100 bits of security against preimage attacks;

   (c) 75 bits of security against second preimage attacks;

   (d) the security strengths of (a)-(c) simultaneously.

   [For this question, we only consider generic attacks.]

3. **Inner collisions in sponge functions.** In this exercise, we consider the sponge construction F as given in Figure 1. In a sponge, the permutation f is repeatedly applied to a state consisting of an inner and an outer part. As the figure suggests, we start with an all-zero state. Then, during the absorbing phase, we add a message block to the outer part of the current state before applying f. We repeat this process until all message blocks have been "absorbed". During the squeezing phase, we take the outer part of the state as partial output, and then apply permutation f to update the state. Again, we repeat this process until we have obtained an output of the desired length.

   (a) Assume that two input messages $M \neq M'$ result in two equal states at the end of the absorbing phase. Explain that these messages yield a collision for the sponge construction F.

   In the remainder of this exercise, we are going to show that, once we have found two messages $M \neq M'$ (which have a length of $\ell \cdot r$ and $k \cdot r$ respectively) for which only the inner parts of the states are equal at the end of the absorbing phase[1], we can also find two messages $\hat{M} \neq \hat{M}'$ that yield a collision for F.

   (b) Suppose that, at the end of the absorbing phase, the resulting states for M and M' are equal to $x_0 \| y_0$ and $x_1 \| y_1$, such that x_0 and x_1 denote the outer parts and y_0 and y_1 the inner parts, with $y_0 = y_1$. Create two additional r-bit message blocks M_ℓ and M'_k such that $M_\ell \oplus x_0 = M'_k \oplus x_1$.

   (c) Use (b) to give two messages $\hat{M} \neq \hat{M}'$ that result in two equal states at the end of the absorbing phase.

   (d) Show that these messages $\hat{M}$ and $\hat{M}'$ yield a collision for the sponge construction F.

   (e) Suppose $|M_{\ell-1}|, |M'_{k-1}| \neq r$, and hence, padding is used. We use the 10^* padding (like previously used in block ciphers). Show how you can still obtain a collision for F in a similar fashion as (b)-(d).

[1] Technically speaking, it is before the absorbing phase is over, precisely before the last application of the permutation f.

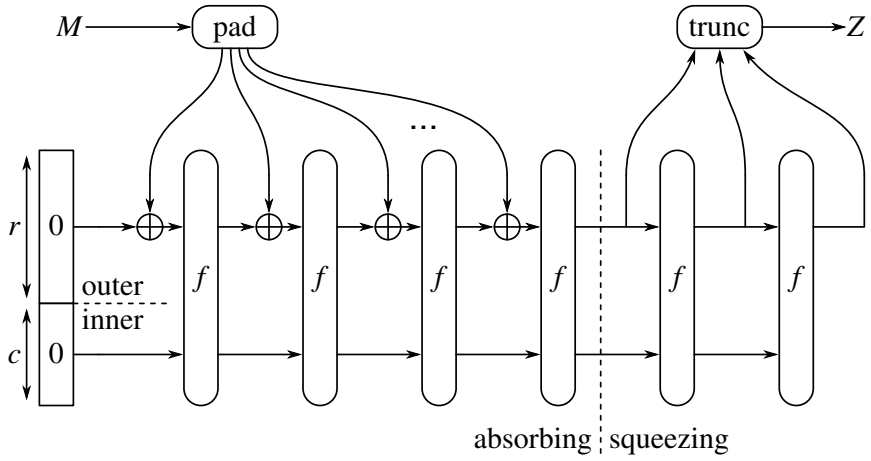

Figure 1: The sponge construction.

Hand in assignments:

1. **(60 points) Hash property reductions.** In this exercise, we investigate security properties from hash functions that can carry over to new constructions.

   (a) Let $\mathrm{h}\colon \{0,1\}^* \to \{0,1\}^n$ be a hash function. Let $\mathrm{H}\colon \{0,1\}^* \to \{0,1\}^n$ be obtained by applying h twice, i.e., $\mathrm{H}(m) = \mathrm{h}(\mathrm{h}(m))$. Suppose that $\mathcal{A}$ is an algorithm that can find a preimage for $m_0 \in \{0,1\}^n$ under H with effort N and probability p. Show that you can also find a preimage for m_0 under h with effort $\leq N+1$ and probability p. 16 pt

   (b) Explain that if h has security strength 128 bits with respect to preimage resistance, then H has security strength at least 127 bits (it is quite close to 128 bits) with respect to preimage resistance. 4 pt

   Let $\mathrm{h}_1\colon \{0,1\}^* \to \{0,1\}^{n_1}$ and $\mathrm{h}_2\colon \{0,1\}^* \to \{0,1\}^{n_2}$ be hash functions. Define $\mathrm{H}_1\colon \{0,1\}^* \to \{0,1\}^{n_1+n_2}$ by setting $\mathrm{H}_1(m) = \mathrm{h}_1(m) \| \mathrm{h}_2(m)$.

   (c) Suppose that both h_1 and h_2 have a security strength with respect to collision resistance of, respectively 128 bits and 96 bits. Explain that the security strength of H_1 with respect to collision resistance is also 128 bits. [It may help to do this in two steps as in (a) and (b).] 20 pt

   Now, define $\mathrm{H}_2\colon \{0,1\}^* \to \{0,1\}^{n_1}$ by setting $\mathrm{H}_2(m) = \mathrm{h}_1(\mathrm{h}_2(m))$.

   (d) Suppose that h_1 has a security strength with respect to collision resistance of 128 bits. Show that for a bad choice of h_2, the security strength with respect to collision resistance of H_2 will be much lower. Give the security strength of H_2 with respect to collision resistance (for your (bad) choice of h_2). 20 pt

2. (**40 points**) **SHADES, a novel MAC function! But how secure is it?** We build a novel MAC function called SHADES from SHA-256 and DES. It works as follows. It takes the input (message) and hashes it with SHA-256. Then, it truncates the resulting digest to its first 64 bits. Subsequently, it encrypts the result with DES_K with K a secret key. Then, the output of DES is truncated to its first 48 bits and that will be the tag. This question is about the quantitative security strength of this function. Attackers can make use of offline queries (i.e., computations) and online MAC generation and verification queries. Each DES computation and each SHA-256 computation counts as one offline query. We denote the total number of offline queries by N, and the total number of online queries by M. Consider the following exhaustive key search attack.

Exhaustive key search on SHADES: Make two generation queries on arbitrary (short) messages m_1 and m_2 and keep the resulting tags T_1 and T_2. Compute (offline) for both messages m_1 and m_2 the SHA-256 digests and truncate the results to 64 bits, that we will call p_1 and p_2. Then, try different random keys K' until one is found that satisfies the following two criteria:

(a) The first 48 bits of p_1 encrypted with $\text{DES}_{K'}$ is T_1.

(b) The first 48 bits of p_2 encrypted with $\text{DES}_{K'}$ is T_2.

These checks can be done with offline computations. A key K' that satisfies both checks is very likely to be the right key: $K' = K$. Using a single pair (m, T) would not be sufficient to determine the key as it is highly likely that, in addition to K, there are many other keys K' for which SHADES maps m to T for any pair m and T.

Answer the following questions and provide for each of them a (short) explanation.

(a) Give the success probability of a single key guess in the exhaustive key search. [Hint: DES has a 56-bit key. You can ignore the complementation property here.] 4 pt

(b) Show that the security strength of SHADES against exhaustive key search is 56 bits. [Hint: In exhaustive key search, you do many key guesses.]

(c) Give the security strength of SHADES against forgery if an attacker can only (blindly) guess tags. 8 pt

(d) You can use collisions to create a forgery. Explain how that would work. [Hint: You do not require a collision on the full output of SHA-256.] 8 pt

(e) Taking into account the attacks in the previous sub-questions, give the security strength of SHADES against tag forgery. 12 pt