Prof. Dr. Rudolf Mathar, Dr. Michael Reyer, Jose Leon, Qinwei He

Exercise 4

Friday, November 17, 2017

Problem 1. *(Collision in hash functions)* Consider the following function:

$$h: \{0,1\}^* \to \{0,1\}^*, \; k \mapsto \left(\left\lfloor 10000\left((k)_{10}(1+\sqrt{5})/2 - \left\lfloor (k)_{10}(1+\sqrt{5})/2 \right\rfloor\right)\right\rfloor\right)_2.$$

Here, $\lfloor x \rfloor$ is the floor function of x (round down to the next integer smaller than x). For computing $h(k)$, the bitstring k is identified with the positive integer it represents. The result is then converted to binary representation.
(example: $k = 10011$, $(k)_{10} = 19$, $h(k) = (7426)_2 = 1110100000010$)

a) Determine the maximal length of the output of h.

b) Give a collision for h.

Problem 2. *(number of messages and hardware resources of two hash functions)* Consider two hash functions, one with an output length of 64 bits and another one with an output length of 128 bits.

For each of these functions, do the following:

a) Determine the number of messages that have to be created to find a collision with a probability larger than 0.86 by means of the birthday paradox.

b) Determine the hardware ressources required for this attack in terms of memory size, number of comparisons, and number of hash function executions.

Problem 3.

a) Explain the four requirements for a cryptographic hash function

Let $h : \{0,1\}^* \to \{0,1\}^n$ be a function that is a second-preimage and collision free. Let $h^{'} : \{0,1\}^* \to \{0,1\}^{n+1}$ be a function given by the rule:

$$h'(m) = \begin{cases} 0 \parallel m & m \in \{0,1\}^n, \\ 1 \parallel h(m) & \text{otherwise} , \end{cases}$$

where the symbol $\parallel$ denotes concatenation.

b) Show that h' is not preimage resistant, but still second-preimage resistant.

Let the input data be of the form $X = (X_0, X_1, X_2, \ldots, X_{n-1})$ where each X_i is a byte. Consider the following hash function: $h\{0,1\}^8 \to \{0,1\}^8$

$$h(X) = X_0 \oplus X_1 \oplus X_2 \oplus \ldots \oplus X_{n-1},$$

where $\oplus$ stands for bitwise addition modulo two.

c) Considering $X \neq 0$, is this a secure hashing method in the sense that collisions are hard to find? Substantiate your answer.

Consider the following signature scheme. The scheme operates on $\mathbb{Z}_q$ for a large prime q and a generator g of $\mathbb{Z}_q$. A user has a private key $\alpha \in \mathbb{N}$ and a public key $X = g^{\alpha} \mod q$.

To sign a message m, one first computes $h = H(m)$ for some hash function H. Then one computes $z = \alpha/h$ with $h^{-1} \mod q - 1$ (assuming h $\neq 0$). The signature is $s = g^z \mod q$. The verification of signature s consists of checking that $s^h \equiv X \mod q$.

d) Will valid signatures be accepted?

e) Is it feasible to sign an arbitrary message without knowing α?