Cryptography

JOACHIM VON ZUR GATHEN, JÉRÉMIE DETREY

6. Tutorial: Cryptographic hash functions

Exercise 6.1 (Derivated hash functions).

Let $h_0 \colon \{0,1\}^{2m} \to \{0,1\}^m$ be a collision-resistant hash function with $m \in \mathbb{N}_{>0}$.

Note: in the following, "|" denotes the concatenation of bit strings.

(i) We construct a hash function $h_1 \colon \{0,1\}^{4m} \to \{0,1\}^m$ as follows: Interpret the bit string $x \in \{0,1\}^{4m}$ as $x = (x_1|x_2)$, where both $x_1, x_2 \in \{0,1\}^{2m}$ are words with $2m$ bits. Then compute the hash value $h_1(x)$ as

$$h_1(x) = h_0(h_0(x_1)|h_0(x_2)).$$

Show that h_1 is collision-resistant.

(ii) Let $i \in \mathbb{N}$, $i \geq 1$. We define a hash function $h_i \colon \{0,1\}^{2^{i+1}m} \to \{0,1\}^m$ recursively using h_{i-1} in the following way: Interpret the bit string $x \in \{0,1\}^{2^{i+1}m}$ as $x = (x_1|x_2)$, where both $x_1, x_2 \in \{0,1\}^{2^i m}$ are words with $2^i m$ bits. Then the hash value $h_i(x)$ is defined as

$$h_i(x) = h_0(h_{i-1}(x_1)|h_{i-1}(x_2)).$$

Show that h_i is collision-resistant.

Exercise 6.2 (DLP and hash functions).

The numbers $q = 7541$ and $p = 15083 = 2q+1$ are prime. We choose the group $G = \{z \mid \text{ord } z | q\} < \mathbb{Z}_p^\times$. Let $\alpha = 604$ and $\beta = 3791$ be elements of G.

(i) Show that both elements α and β have order q in $\mathbb{Z}_p^\times$ and (thus) generate the same subgroup.

(ii) Consider the hash function

$$\begin{array}{rrcl} h: & \mathbb{Z}_q \times \mathbb{Z}_q & \longrightarrow & G \\ & (x_1, x_2) & \longmapsto & \alpha^{x_1}\beta^{x_2}. \end{array}$$

Compute $h(7431, 5564)$ and $h(1459, 954)$.

(iii) Find $\log_\alpha \beta$.

(iv) Prove that for any p, q (both prime with q dividing $p-1$) finding a collision of h solves a discrete logarithm in the order q subgroup of $\mathbb{Z}_p^\times$ (which is thought to be difficult...).

Exercise 6.3 (Hash functions for long messages).

We want to use the previous discrete-logarithm-based hash function to build a hash function for messages of arbitrary length, as seen in the lecture.

First of all, we need to tweak our original hash function so that it complies with a few requirements: since we are now working at the bit level, we need a hash function $\tilde{h}$ from $\mathbb{Z}_2^m$ to $\mathbb{Z}_2^t$.

(i) What are the conditions on m and t so that $\tilde{h}$ remains collision-resistant?

(ii) Compute m and t.

(iii) Recall the method for hashing long messages. Give the corresponding pseudo-code.

(iv) Hash the two bit strings 100 and 110001101001001110.

(v) Find a collision of $\tilde{h}$ and of h.

(vi) Conclude on the discrete logarithm of β in base α.