# Security
**Assignment 5, Monday, October 10, 2011**

**Handing in your answers:** There are two options:

1. Put your solutions in Pim's post box (in the white cabinet, lower door, near the printer at the station-side-end of the corridor on the second floor of the Huygens building). Before handing in, make sure:
   - your names and student numbers are written on the document.
2. Send your solutions by e-mail to `pim@cs.ru.nl` with subject '*assignment 5*'. This e-mail should only contain a single PDF document as attachment. Before handing in, make sure:
   - your names and student numbers are included in the document (since it will be printed),
   - your names are part of the filename (example: PimVullers_FlavioGarcia-5.pdf), and
   - the file is a PDF document.

**Deadline:** Monday, October 17, 8:30 sharp!

**Properties of cryptographic hash functions:** A cryptographic hash is a function $h$ that assigns to a bit string another bit string (usually of fixed length, but this is not a formal requirement here), and has the following four properties:

(E) $h$ can be computed efficiently (efficiency),

(O1) *given a bit string* $y$ it is infeasible to determine a bit string $x$ such that $h(x) = y$ (pre-image resistance),

(O2) *given a bit string* $x$ it is infeasible to determine a different bit string $x'$ ($x \neq x'$) such that $h(x) = h(x')$ (2nd pre-image resistance),

(C) it is infeasible to determine *any two different bit strings* $x$ *and* $x'$ ($x \neq x'$) such that $h(x) = h(x')$ (collision resistance).

1. **(6 points)** Given a function $H$ that meets (E), (O1), (O2) and (C), determine for the following functions $h$ (that possibly make use of $H$) which of the properties apply. Only give the labels of the properties (no additional explanation), for example '(a) E and O1'.
   (a) The function $h(x) = 1$. (The function $h$ returns constant output)
   (b) The function $h(x) = x$. (The function $h$ just returns its input)
   (c) The function $h(x_1 \dots x_{2n}) = (x_1 \oplus x_{n+1}) \cdot (x_2 \oplus x_{n+2}) \cdots (x_n \oplus x_{2n})$, where $x_i$ are the individual bits of the input and $\oplus$ denotes the XOR or exclusive-or operation. (The function $h$ computes the bitwise exclusive-or of the first and the second half of its input.)
   (d) The function $h(x_1 ... x_{2n}) = x_1 ... x_n \oplus H(x_{n+1} ... x_{2n})$. (The function $h$ computes the bitwise exclusive-or of the first half and the hash of the second half of its input.)
   (e) The function $h(x_1 ... x_{2n}) = x_1 ... x_n \oplus H(x_1 ... x_{2n})$. (The function $h$ computes the bitwise exclusive-or of the first half of its input and the hash of its input.)
   (f) The function $h(x_1 \dots x_n) = \begin{cases} H(x_2 \dots x_n) & \textbf{if } x_1 = 0 \\ H(x_n \dots x_2) & \textbf{if } x_1 = 1 \end{cases}$
   (Depending on the first input bit the function $h$ hashes the remaining bits in the same or in reverse order using $H$.)

2. **(4 points)** Alice and Bob play a game where they need to answer a few multiple choice questions with either I, II, III or IV. The questions are e-mailed to them by Eve. They do not trust Eve in checking their answers and decide to e-mail their answers directly to each other. This poses a problem; who is going to send the answers first? Bob might, for instance, change his answers after receiving the answers from Alice.

   Alice and Bob decide to use a secure hash function $h$ that has the properties (E), (O1), (O2) and (C). Their intention is to hide their actual answer and, at the same time, commit to an answer such that they cannot change it afterwards. Let, for question $i$, $a_i \in \{\text{I}, \text{II}, \text{III}, \text{IV}\}$ be the answers of Alice and $b_i \in \{\text{I}, \text{II}, \text{III}, \text{IV}\}$ be the answers of Bob.

   Their scheme is as follows:

   1. $A \longrightarrow B$ : $h(a_i) = x_i$
   2. $B \longrightarrow A$ : $h(b_i) = y_i$
   3. $B \longrightarrow A$ : $b_i$
   4. $A$ : **verify** $h(b_i) = y_i$
   5. $A \longrightarrow B$ : $a_i$
   6. $B$ : **verify** $h(a_i) = x_i$
   7. $\ldots$ : continue at step 1 with the next question $(i + 1)$

   When they successfully go through all steps they both think that cheating is impossible.

   (a) Does this scheme prevent cheating (yes/no)? If yes, explain why. If no, give an attack.

   (b) In another game the questions are no longer multiple choice. So Alice and Bob will have to send some long text answers instead of just I, II, III or IV. They still want to use the same scheme. Does this scheme, in this new setting, prevent cheating (yes/no)? If yes, explain why. If no, give an attack.

   (c) This scheme consists of two phases. First Alice and Bob commit to an answer in steps 1 and 2 and in steps 3 to 6 they reveal and verify them. Is the order of the steps *within* these phases important to prevent cheating (by Alice or Bob)? For example, can we just switch the combination of steps 3 and 4 with the combination of steps 5 and 6, or would this break the scheme? Consider this for both the multiple choice game as well as the one with open questions. Briefly explain your answer.