Sécurité des Systèmes d'information
Exercise sheet 5 : Hash functions and MACs

20 November 2019

All answers should be justified.

Definition of a hash function

A hash function $h : X \to Y$ should respect two rules : always give a fixed finite size answer, whatever the input and being easy to compute.
But in order to be useful from a cryptographic point of view, it should also respect the three following properties :

- **Preimage resistance** : Given $y \in Y$, it is impossible to find a $x \in X$ such that $h(x) = y$.
- **Second preimage resistance** : Given a $x \in X$ and a $y \in Y$ such that $h(x) = y$, it is impossible to find $x' \neq x$ such that $h(x') = h(x) = y$. We also call this property "weak collision resistance".
- **Collision resistance** : It is impossible to find distincts $x, x' \in X$ such that $h(x) = h(x')$. We also call this property "strong collision resistance".

Note : When we use the word "impossible" in the preceding properties, we really mean impossible in a computable way.

Exercise 1 : Hash functions

- Does the preimage resistance imply the second preimage resistance ? and conversely ?
- Does the second preimage resistance imply the collision resistance ? and conversely ?
- Let $h_1(x) = x \mod n$, where n is a big integer. Which properties does the function h_1 satisfy ?

- Let $x = x_1 ... x_n$ be a sequence of octets. We let $+$ describe the addition bit per bit, mod 2 (that is xoring two octets). Let $h_2(x) = x_1 + ... + x_n$. Which properties does the function h_2 satisfy ?

- Let x and the operation $+$ be defined as before, and let $*$ be the multiplication operation mod 16 on blocks of 4 bits (That is, in order to multiply two octets, we split it in half and perform multiplication on each half separately) :
  For instance, $5 * (7A)_{16} = (5 * (7)_{16} \mod 16) \parallel (5 * (A)_{16} \mod 16) = 35 \mod 16 \parallel 50 \mod 16 = (32)_{16}$.
  We define $h_3 = n * x_1 + (n-1) * x_2 + ... + 1 * x_n$. Which properties does the function h_3 satisfy ?

Exercise 2 : Message Authentication Codes

We will use a block cipher E_k in CBC mode to create MACs. We consider a CBC without IV (or with IV = 0)

- Let the MAC be defined as follow :

$$\begin{cases} t_1 = E_k(m_1) \\ t_{i+1} = E_k(m_{i+1} \oplus t_i) \end{cases}$$

  We consider a message divided into two blocks $m = m_1 \parallel m_2$ and we define its MAC $t = t_1 \parallel t_2$. Use these informations to create a false MAC, that is a new message m' and its corresponding MAC t' without any knowledge of the key.

- Let $M = m_1 \parallel m_2 \parallel ... \parallel m_n$ be a message divided into n blocs, and let $C = c_1 \parallel c_2 \parallel ... \parallel c_n$ be the CBC cipher computed in the following way :

$$\begin{cases} c_1 = E_k(m_1) \\ c_{i+1} = E_k(m_{i+1} \oplus c_i) \end{cases}$$

  We define the following MAC (based on key K) :

$$MAC = E_K(m_n \oplus E_K(m_{n-1} \oplus E_K(... \oplus E_K(m_2 \oplus E_K(m_1))...)))$$

  Is it possible to falsify this MAC ? Which method would you use in order to insure the integrity of the couple authentication/cipher ? What if we were to use this function MAC as a hash function ? Would it respect the collision resistance property ?