# 13/11/2023. HAI709I : Fondements cryptographiques de la sécurité. Test de mi-parcours : une solution

**Question 6.** Soit $\Pi = (Gen, Enc, Dec)$ un schéma de chiffrement sûr au sens calculatoire. On considère l'expérience suivante :

- on produit une clef secrète aléatoire $k$ pour le paramètre de sécurité $n$, $k \leftarrow Gen(1^n)$
- nous produisons au hazard un message ouvert $m = x_1 \dots x_n \in \{0,1\}^n$ (avec la distribution uniforme, c'est-à-dire que chaque message peut être choisi avec la probabilité $1/2^n$)
- on calcule un message chiffré $e = Enc(m, k)$
- Adversaire obtient $1^n$ et le message chiffré $e$, et essaie de deviner quel est le XOR des premiers 10 bits du message ouvert $x_1 \oplus \dots \oplus x_{10}$ et renvoie le résultat $j \leftarrow Adv(1^n, e)$.

Le succès d'Adversaire est défini comme

$$\textbf{succès} = \begin{cases} 1, & \text{si } j = x_1 \oplus x_2 \oplus \dots \oplus x_{10} \\ 0, & \text{sinon.} \end{cases}$$

Montrer que pour tout algorithme $Adv$ calculable en temps polynômial, il existe un fonction négligeable $g(n)$ telle que

$$\text{Prob}[\textbf{succès} = 1] \leq \frac{1}{2} + g(n).$$

(Cela signifie qu'étant donné le message chiffré, Adversaire ne peut pas apprendre en temps polynômial le XOR des 10 premiers bits du message ouvert.)

**Reminder :** Let us recall the «Standard Game» between Alice and Adversary.

Let $\Pi = \langle \text{Gen}(), \text{Enc}(), \text{Dec}() \rangle$ be an encryption scheme, where $\mathcal{M}, \mathcal{E}, \mathcal{K}$ are the spaces of *clear messages*, *encrypted messages*, and *secret key* respectively. We consider the following game between an adversary and Alice.

- Adversary uses an algorithm $Adv_1()$ that chooses two clear messages $m_a, m_b \in \mathcal{M}$ ;
- Alice chooses at random $i \in \{a, b\}$ (with equal probabilities), samples a secret key $k \leftarrow \text{Gen}()$, and computes the encrypted message $e = \text{Enc}(m_i, k)$ ;
- Adversary computes $j \in \{a, b\}$ using another algorithm $j \leftarrow Adv_2(m_a, m_b, e)$.

The success of the adversary is defined as follows :

$$\textbf{success} = \begin{cases} 1, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases}$$

In words : the adversary prepared a pair of messages $m_a, m_b$ ; Alice decides which message to encrypt ; then the adversary tries to understand which of the messages was encrypted.

**Definition.** An encryption scheme $\Pi = \langle \text{Gen}(), \text{Enc}(), \text{Dec}() \rangle$ is computationally secure (sûr au sens calculatoire) if for every *standard game* between Adversary and Alice (following the rules explained above) such that the adversary's algorithms $Adv_1$ and $Adv_2$ are computable in polynomial time, the gap

$$\left| \text{Prob}[\text{succes}] - \frac{1}{2} \right|$$

is a negligible function.

**Solution for Exercise 6.** For the sake of contradiction we assume that there exists a polynomial time computable algorithm $Adv$ such that

$$\text{Prob}[\textbf{succès} = 1] = \frac{1}{2} + g(n),$$

where $g(n)$ is *not* negligibly small. We are going to use this assumption to show that $\Pi$ is *not* computationally secure [see the definition of a computationally secure scheme above]. To this end, we use the existing algorithm $Adv$ to construct a pair of algorithms $Adv_1$ and $Adv_2$ that win in the «Standard Game» from the definition of computationally secure scheme with a probability $\frac{1}{2} + g(n)$ (which will contradict the definition of security).

$Adv_1$ : given the input $\underbrace{111\ldots1}_{n}$, we select a random string $m_a = x_1 \ldots x_n \in \{0,1\}^n$ and a string $m_b = y_1 \ldots y_n \in \{0,1\}^n$ that differs from $x_1 \ldots x_n$ in exactly one bit, and this bit must be in one of the first ten positions. Observe that $m_a = x_1 \ldots x_n$ as well as $m_b$ are uniformly distributed on $\{0,1\}^n$ (but, of course, they are not independent of each other).

By the construction, the XOR of the bits $x_1, \ldots, x_{10}$ and the XOR of the bits $y_1, \ldots, y_{10}$ are different (one of these sums of bits is odd and another one is even).

$Adv_2$ : given an encrypted message $e$, we apply the algorithm $Adv(e)$ [we use $Adv$ as a black box, we do not know how it works] and obtain the result $j$. Then we select among the strings $m_a$ and $m_b$ one for which the XOR of the first ten bits is equal to $j$ and return this string as the final result.

**Claim 1 :** If we plug $Adv_1$ and $Adv_2$ in «Standard Game» explained above, then $e = Enc(m_i, k)$ is an encoding of a uniformly chosen $n$-bit string. Indeed, $m_a$ and $m_b$ are both uniformly distributed, and in the game Alice chooses at random one of them, so the the chosen by Alice clear message is also uniformly distributed string of $n$ bits.

**Claim 2 :** In our game, with a probability $\frac{1}{2} + g$, the result of $Adv(e)$ is the correct value of the XOR of the first 10 bits of clear message chosen by Alice. This follows from two facts :

— the chosen clear messages is uniformly distributed (see Claim 1)

— for a uniformly distributed random clear message $m$, a randomly chosen key $k$, and the corresponding encrypted message $e = Enc(m, k)$, the result $Adv(e)$ returns the XOR of the first ten bits of $m$ with probability $\frac{1}{2} + g$ (this is our initial assumption about $Adv$).

**Claim 3 :** If we know the XOR of the first 10 bits of the clear message, we know for sure which clear message was encrypted ($m_a$ or $m_b$). This is because by the construction, the XORs of the first 10 bits for $m_a$ and $m_b$ are different.

Combining all three claims together we conclude that Adversary succeeds in the «Standard Game» with probability $\frac{1}{2} + g$. This contradicts the definition of security of the scheme $\Pi$ if $g(n)$ if not negligibly small.