CRYPTA - TD 2
2020-2021

Énoncé

Exercice 1 : Extraction de racine carrée modulo N

Notons p un nombre premier et $f \geq 1$ un entier. Nous rappelons le critère d'Euler généralisé qui garantit qu'un entier x premier avec p est un carré modulo p^f si et seulement si $x^{(p-1)/2} \equiv 1 \bmod p$.
Soit N un entier dont la décomposition en facteur premiers est $N = q_1^{f_1} \dots q_d^{f_d}$ où $q_i \in \mathbb{P}$ sont des nombres premiers deux à deux distincts et $f_i \geq 1$ pour $i \in \{1, \dots, d\}$.

1.a] Montrer qu'un entier x est un carré modulo N dès que $x^{(q_j-1)/2} \equiv 1 \bmod q_j$ pour tout $j \in \{1, \dots, d\}$.

1.b] Montrer qu'un tel carré modulo N a exactement 2^d racines carrées.

1.c] Montrer que s'il existe un algorithme $\mathcal{A}$ capable d'extraire des racines carrées dans $(\mathbb{Z}/N\mathbb{Z})$ en temps τ, alors il existe un algorithme $\mathcal{B}$ qui retourne un diviseur propre de N en temps espéré $O(\tau)$.

Exercice 2 : Factorisation et logarithme discret

Soient p et q deux nombres premiers distincts et soit $N = pq$. Soit α un élément inversible de $(\mathbb{Z}/N\mathbb{Z})$. Notons α_p et α_q les images de α dans $(\mathbb{Z}/p\mathbb{Z})^*$ et $(\mathbb{Z}/q\mathbb{Z})^*$ (respectivement).

2.a] Montrer que l'ordre de α dans $(\mathbb{Z}/N\mathbb{Z})^*$ est égal au plus petit multiple commun de l'ordre de α_p dans $(\mathbb{Z}/p\mathbb{Z})^*$ et de l'ordre de α_q dans $(\mathbb{Z}/q\mathbb{Z})^*$.

2.b] Notons $d = \text{pgcd}(p-1, q-1)$. Montrer qu'il existe un élément d'ordre $\varphi(N)/d$ dans $(\mathbb{Z}/N\mathbb{Z})^*$ où $\varphi(N) = (p-1)(q-1) = \#(\mathbb{Z}/N\mathbb{Z})^*$ est la fonction indicatrice d'Euler de N.
Dans la suite nous supposons que $p > 3$ et $q > 3$ et que $\text{pgcd}(p-1, q-1) = 2$.

2.c] Soit γ un élément d'ordre $\varphi(N)/2$ de $(\mathbb{Z}/N\mathbb{Z})^*$ et soit a le logarithme discret de $\gamma^N \bmod N$ en base γ. Montrer que $N - a = \varphi(N)$.

2.d] Écrire un algorithme polynomial qui prend en entrée N et a et retourne les facteurs premiers p et q de N.

Exercice 3 : Sécurité des bits de la fonction RSA

Soient N un module RSA, e un nombre entier premier avec $\varphi(N)$ et $f_{N,e}$ la fonction RSA associée. Considérons la fonction $\delta : \mathbb{Z}_N \longrightarrow \{0, 1\}$ définie par

$$\delta(x^e) = \begin{cases} 0 & \text{si } 0 \leq x \bmod N \leq N/2, \\ 1 & \text{si } N/2 \leq x \bmod N \leq N-1. \end{cases}$$

3.a] Montrer l'équivalence

$$\delta((2x)^e \bmod N) = 0 \quad \Longleftrightarrow \quad x \in \left[0, \frac{N}{4}\right[\cup \left[\frac{N}{2}, \frac{3N}{4}\right[$$

et généraliser en caractérisant les $x \in \mathbb{Z}_N^*$ tels que $\delta((2^t x)^e) \bmod N) = 0$.

3.b] Montrer comment transformer tout algorithme polynomial calculant $\delta(z)$ pour tout $z \in \mathbb{Z}_N^*$ en un algorithme polynomial qui inverse la fonction RSA.

3.c] Considérons la fonction $\gamma : \mathbb{Z}_N \longrightarrow \{0,1\}$ définie par

$$\gamma(x^e) = \begin{cases} 0 & \text{si } x \text{ est pair,} \\ 1 & \text{si } x \text{ est impair.} \end{cases}$$

Montrer comment transformer tout algorithme polynomial calculant $\gamma(z)$ pour tout $z \in \mathbb{Z}_N^*$ en un algorithme polynomial qui inverse la fonction RSA.