

Algorithm p-1 de Pollard

Nous avons vu en cours que la factorisation d'entiers RSA quelconques est un problème difficile, néanmoins solvable en temps *subexponentiel* à l'aide des algorithmes de crible. Toutefois, si l'entier RSA est mal généré, il existe des cas plus faciles.

Soit $N = PQ$ un module RSA où P et Q sont premiers et de même taille. Soit B une borne à choisir plus tard. L'algorithme $P - 1$ de Pollard s'exécute selon les étapes suivantes :

1. Calculer $M = \prod_{\text{premiers } q \leq B} q^{\lfloor \log_q B \rfloor}$ (i.e., calculer le produit des puissances de nombres premiers inférieures à B)
2. Sélectionner a au hasard premier avec N
3. Calculer $G = \text{GCD}(a^M - 1, N)$
4. Si $G < N$ renvoyer G ; sinon renvoyer « échec »

Question 1. On suppose que $P - 1$ est B -ultrafriable (B -powersmooth), c'est-à-dire que toutes les puissances de nombres premiers p^ν dans sa décomposition en facteurs premiers sont plus petites que B . Montrer que M est multiple de $P - 1$.

Question 2. On suppose aussi que Q n'est pas B -powersmooth. Montrer que l'algorithme renvoie P avec bonne probabilité.

Question 3. Donner une borne sur la complexité de l'algorithme, à B fixé, puis à B variable.

Lors de la génération de « bons » nombres premiers on impose ainsi que $P - 1$ ait au moins un « grand » facteur premier. Toutefois, la méthode de factorisation ECM (basée sur les courbes elliptiques) a rendu l'algorithme $P - 1$ obsolète, et fonctionne aussi bien lorsque $P - 1$ est powersmooth ou ne l'est pas.

Dans la méthode ECM, on utilise en effet le groupe des points d'une courbe elliptique quelconque définie sur \mathbb{Z}_N . Ce groupe est d'ordre variable, mais proche de N . Cette variabilité permet de tomber avec forte probabilité sur un ordre friable (smooth), contrairement à l'algorithme $P - 1$ dans lequel le choix du groupe est contraint. On calcule donc les multiples d'un point de la courbe jusqu'à tomber sur un élément non-inversible, qui doit apparaître assez tôt à cause du petit théorème de Fermat. Cet algorithme est de complexité subexponentielle.

Autour de RSA

On rappelle le schéma de chiffrement RSA basique.

KeyGen(1^n) choisit un module N qui est le produit de deux premiers de n bits, avec deux entiers e et d tels que $ed = 1 \pmod{\phi(N)}$. $\text{pk} = (N, e)$; $\text{sk} = (N, d)$
Enc($m \in \mathbb{Z}_N^*, (N, e)$) renvoie $c = m^e \pmod{N}$
Dec($c \in \mathbb{Z}_N^*, (N, d)$) renvoie $m = c^d \pmod{N}$

Nous avons déjà vu que ce RSA basique n'est pas IND-CPA. Dans cet exercice nous explorons quelques autres attaques sur ce schéma.

Question 4. Soit $N = PQ$ un produit de deux premiers distincts. Montrer que si $\phi(N)$ et N sont connus, alors on peut retrouver p, q en temps polynomial.

Question 5. Montrer que si $m \in [0, N^{1/e}]$ alors on peut facilement décrypter (= retrouver le message sans connaître la clé privée, par opposition à déchiffrer).

Question 6. Une racine de l'unité modulo N est un entier x tel que $x^2 = 1 \pmod{N}$.

1. Combien y a-t-il de racines de l'unité modulo N ?

2. Supposons que l'on connaisse (N, e, d) (mais pas la factorisation de N). Montrer qu'on peut calculer une racine de l'unité modulo N . On admet qu'elle est non-triviale avec bonne probabilité.
3. En déduire qu'on peut factoriser N .

Question 7. Fixons un module N et supposons qu'un serveur centralisé donne aux utilisateurs des paires (e_1, d_1) et (e_2, d_2) formant des clés RSA valides (exposants privés et publics). Pourquoi est-ce une mauvaise idée ?

Question 8. Soit $(N_1, e), \dots, (N_e, e)$ les clés publiques de e utilisateurs différents. Un même message m est chiffré e fois, avec chacune de ces clés publiques. Montrer qu'un attaquant peut retrouver m à partir de l'observation des chiffrés $c_i := \text{Enc}(m, (N_i, e))$.

Question 9. On essaie maintenant d'éviter l'attaque de la question précédente. On a $m < \sqrt{N_i}$ mais on force chaque utilisateur à utiliser une modification de son message m , sous la forme d'un décalage δ_i connu. L'attaquant n'observe donc plus que les chiffrés de $m + \delta_1, m + \delta_2, \dots, m + \delta_e$.

On admet le théorème de Coppersmith :

Theorem 1. Soit $f \in \mathbb{Z}[X]$ un polynôme unitaire de degré e et N un entier. S'il existe une racine x_0 de f modulo N telle que $|x_0| \leq N^{1/e-\varepsilon}$, alors il est possible de retrouver x_0 en temps polynomial en $\log N$ et $1/\varepsilon$.

Montrer comment retrouver m .

Fonction Indicatrice d'Euler

On rappelle que l'indicatrice d'Euler est définie par $\phi(N) = |\mathbb{Z}_N^*|$, l'ordre du groupe \mathbb{Z}_N^* . Dit autrement, c'est le nombre d'entiers de $[1; N]$ qui sont premiers avec N .

Question 10. Soit p un nombre premier, montrer que $\phi(p) = p - 1$.

Question 11. Soient p, q premiers entre eux. Montrer que $\phi(qp) = \phi(p)\phi(q)$.

Question 12. Soit p un premier et $e \geq 1$ un entier. Montrer que $\phi(p^e) = p^{e-1}(p - 1)$.

Question 13. Soit $N = \prod_i p_i^{c_i}$ où les p_i sont des premiers distincts, $c_i \geq 1$. Montrer que $\phi(N) = \prod_i p_i^{c_i-1}(p_i - 1)$;