Courbes elliptiques

Rémi Vaucher

Epita Lyon

2022

Le code RSA

Création des clés (par un seul parti seulement):

- On choisit p et q deux entiers naturels premiers distincts (premières clés secrètes)
- On calcule $n = p.q$ (première clé publique)
- On calcule $\phi(n) = \phi(pq) = \phi(p)\phi(q) = (p-1)(q-1)$ (car p et q premier entre eux)
- On choisit $e < \phi(n)$ tel que $\phi(n) \wedge e = 1$ (deuxième clé publique)
- On calcule $d = e^{-1}[\phi(n)]$ (deuxième clé secrète)

Conclusion: $PubK = (n, e)$ et $Seck = (p, q, d)$

Le code RSA

Chiffrement:

$$C = M^e[n]$$

Déchiffrement:

$$M = C^d[n]$$

Pourquoi? **Petit théorème de Fermat:** Si M n'est pas multiple de k, alors $M^{k-1} = 1[k]$

Casser le code RSA

Si je connais les clés publiques, est ce que je peut retrouver (p, q, d)?
Connaissant juste n, puis je retrouver p et q? (retrouver d devient ensuite facile)
C'est facile, il suffit de factoriser n sachant qu'il est le produit de 2 entiers premier (donc il n'admet strictement aucun autre diviseurs)

DEAL!

Très bien. Factorisons 38009, dans la joie et la bonne humeur s'il vous plait!

$\phi(n)$: un secret bien gardé

RSA est très compliqué à craquer si l'on prends p et q très grand. Mais il est important que $\phi(n)$ reste secret (il est encore plus difficile à calculer). En effet:

$$\phi(n) = (p-1)(q-1)$$

Posons $q = \frac{n}{p}$. Notre équation devient (après quelques étape):

$$p^2 + p(\phi(n) - 1 - n) + n = 0$$

... qui admet pour racine exactement p et q!

Algorithmes de factorisation:

- Algorithme $p-1$ de Pollard
- Algorithme ρ de Pollard (encore lui)
- Problème du logarithme discret.

On raisonnera toujours avec p et q des entiers premiers distincts et différents de 1.

Algorithme $p-1$ de Pollard

Principe: Si $N = pq$, $a = b[N]$ implique $a = b[p]$. En particulier (toujours par le petit Fermat) si $e = 0[p-1]$ et $a \neq 0[p]$, alors $b = a^e[N]$ implique $b = a^e = 1[p]$.
En conséquence: $b - 1 = 0[p]$ (mais attention, par forcément mod q, et c'est même peu probable) $\Rightarrow p = (b-1) \wedge N$.

Algorithme $p-1$ de Pollard

Soit $N \in \mathbb{N}$. On va choisir B un seuil de *friabilité* (on considère que c'est le maximum que peut atteindre un facteur premier de N)

- $\rightarrow$ On calcule $e = B!$ (pour être "sûr" que $p-1$ est dedans)
- $\rightarrow$ On choisit un élément $0 < a < N$
- $\rightarrow$ On calcule $a \wedge N$ (on sait jamais). S'il est différent de 1: c'est fini, on a trouvé un diviseur de N.
- $\rightarrow$ On calcule $b = a^e[N]$ (par exponentiation binaire).
- $\rightarrow$ On calcule $(b-1) \wedge N$
  - Si il vaut N, je ne dirais pas que c'est un échec, je dirais plutôt que ça n'a pas marché.
  - Si il vaut 1, même chose.
  - Sinon, on a trouvé p.
- $\rightarrow$ Si l'algorithme a échoué, on recommence avec un B plus grand.

Algorithme ρ de Pollard

- On veut factoriser $N = pq$. On crée une fonction simple, rapide à calculer, mais pas triviale non plus (habituellement $x^2 + 1$) de $\mathbb{Z}/N\mathbb{Z}$ dans $\mathbb{Z}/N\mathbb{Z}$.
- On va créer une suite $x_n = f(x_{n-1})$. Comme on se trouve dans $\mathbb{Z}/N\mathbb{Z}$, à un moment notre suite va être cyclique (c'est le moment de faire une illustration!).
- Maintenant on regarde $x_n[p]$. On ne peut pas la calculer, par contre on sait qu'elle est cyclique (car $x_n[N]$ l'est). Sa période est au plus celle de $x_n[N]$, et est bien souvent plus petite (aka le paradoxe des anniversaires). Si c'est le cas, il existe deux indices i, j tels que $x_i = x_j[p]$, mais $x_i \neq x_j[N]$. On a donc $x_i - x_j = 0[p]$ et donc p divise $(x_i - x_j) \wedge N \neq N$ (comme $x_i \neq x_j[N]$). Et donc $(x_i - x_j) \wedge N = p$.

Algorithme ρ de Pollard

$\rightarrow$ On choisit $x_0 \in \mathbb{Z}/N\mathbb{Z}$ aléatoirement.

$\rightarrow$ Tant que $d = (x_{k+1} - xk) \wedge N = 1$, on calcule les x_k.

$\rightarrow$ On retourne le premier $d \neq 1$.