37 ♦♢♢ Nombres premiers jumeaux

Bob publie sa clé publique RSA $(N,e) = (5183, 11)$.

Sachant que Bob a l'habitude de prendre pour nombres premiers (p,q) des nombres premiers *jumeaux* c'est-à-dire tels que $q = p+2$, calculez p et q.

38 ♦♦♢ RSA avec p et q trop proches

Supposons que n soit un entier produit de deux nombres premiers p et q proches (on peut toujours supposer que $p > q$). On pose $t = \frac{p+q}{2}$ et $s = \frac{p-q}{2}$.

1. Montrer que :
   (a) $n = t^2 - s^2$,
   (b) t est légèrement supérieur à la racine carrée de n.

   On peut utiliser ces informations pour factoriser n.
   Notons $\lceil \sqrt{n} \rceil$ la partie entière par excès de $\sqrt{n}$: $\lceil \sqrt{n} \rceil - 1 < \sqrt{n} \leqslant \lceil \sqrt{n} \rceil$.

   L'algorithme de Fermat permet de trouver p et q à partir de n dans ce cas là, le voici :

   (a) $A \leftarrow \lceil \sqrt{n} \rceil \qquad (\in \mathbb{N})$
   (b) $x = A^2 - n \qquad (\in \mathbb{N})$
   (c) Tant que x n'est pas un carré
      i. $A \leftarrow A + 1$
      ii. $x \leftarrow A^2 - n$
   (d) Retourner $p = A + \sqrt{x}$ et $q = A - \sqrt{x}$

   On sait que lorsque A vaudra $t = \frac{p+q}{2}$ alors $x = t^2 - n = s^2$ sera un carré.
2. Appliquer cet algorithme pour factoriser :
   (a) $1\,253\,311$;
   (b) $6\,212\,933$;
   (c) $27\,171\,013$;
   (d) $1\,995\,107\,033$;
   (e) $3\,649\,574\,023$;
   (f) $14\,780\,385\,761$.
3. Déterminer la complexité de cet algorithme, en fonction de p et de n.
4. Déterminer le nombre d'itérations lorsque p diffère de $\sqrt{n}$ de moins de $\sqrt[4]{4n}$.

39 ♦♦◊ Attaque par module commun sur RSA

Supposons qu'Alice et Bob partagent le même nombre $n = p \times q$ mais des clefs (e_A, d_A) et (e_B, d_B) différentes.

On suppose, de plus, que e_A et e_B sont premiers entre eux (ce qui est le plus général).

Supposons alors qu'Alice et Bob chiffrent un même message m et qu'Oscar intercepte les deux messages chiffrés : $c_A = m^{e_A} \mod n$ et $c_B = m^{e_B} \mod n$ qu'il sait être deux chiffrements du même message m.

Montrer qu'Oscar peut alors retrouver le message m sans factoriser n.

• Applications numériques :

1. Posons $n = 1763; e_A = 43; e_B = 5; c_A = 149$ et $c_B = 413$ retrouver m (sans factoriser n).
2. Posons $n = 667; e_A = 4; e_B = 13; c_A = 277$ et $c_B = 275$ retrouver m (sans factoriser n).
3. Posons $n = 551; e_A = 25; e_B = 11; c_A = 325$ et $c_B = 238$ retrouver m (sans factoriser n).

40 ♦◊◊ De $\phi(n)$ à la factorisation

Montrer simplement comment la connaissance de $\phi(n) = (p-1)(q-1)$ (la fonction d'Euler) permet de remonter à la factorisation de n.

41 ♦◊◊ Le temps de factorisation des grands entiers

Le meilleur algorithme connu à ce jour pour factoriser les grands entiers est le GNFS (General Number Field Sieve). Son facteur de travail, pour factoriser un entier n est donné par la formule :

$$W(n) = k \exp^{c(\log_2 n)^{\frac{1}{3}}(\log_2 \log_2 n)^{\frac{2}{3}}}$$

avec : $c = \sqrt[3]{\frac{64}{9}}$ et k est une constante qui dépend de la qualité du programme.

1. Calculer k sachant qu'en 1999, un entier de 512 bits a été factorisé par une équipe internationale avec un facteur de travail de 8000 Mips-années.
2. Quel est le facteur de travail nécessaire pour factoriser un entier de 768 bits ? Et 1024 bits ?

42 ♦♦◊ Oubli de p et q

La clé publique de Bob est $(N, e) = (81070877, 127)$. Bob a oublié les premiers p et q à l'aide desquels il avait déterminé N ainsi que sa clé privée d, mais il se rappelle que $\phi(N) = 81052860$.

1. Retrouver p, q et d.
2. En quoi le choix des premiers p et q est maladroit ?
   Montrer comment Oscar peut les retrouver rapidement. Détailler ses calculs.