Le cryptage RSA

Nicolas GEORGE

27 juin 1999

Introduction

Le système de cryptage RSA, inventé par trois mathématiciens-informaticiens dont j'oublie toujours le nom (les initiales sont R, S et A) est basé sur certaines propriétés arithmétiques des entiers, et la difficulté de la factorisation en nombres premiers. Je vais ici démontrer totalement la propriété arithmétique, et indiquer comment la mettre en œuvre. Je n'essaierai pas du tout de prouver des questions de complexité algorithmiques.

1 Groupes, propriétés utiles

1.1 Définitions

Soit E un ensemble, et $\star : E \times E \longrightarrow E$ une application. On dit que $\star$ est une loi de composition interne de E.

Le couple $(G, \cdot)$ est un groupe si G est un ensemble, et $\cdot$ est une loi de composition interne de G tels que :

(a) la loi $\cdot$ est associative : $\forall (a,b,c) \in G^3 \ (a \cdot b) \cdot c = a \cdot (b \cdot c)$;

(b) la loi $\cdot$ possède un élément neutre : $\exists e \in G \ \forall x \in G \ e \cdot x = x \cdot e = x$;

(c) tout élément de G possède un inverse : si e est élément neutre, $\forall x \in G \ \exists y \in G \ x \cdot y = y \cdot x = e$.

Remarque. L'associativité permet d'ommettre les parenthèses. L'élément neutre est unique, car si e et e' sont neutres, $e = e \cdot e' = e'$. On le note 1_G. De même, tout $x \in G$ ne possède qu'un inverse, car si y et y' sont inverses de x, $y \cdot x \cdot y' = 1_G \cdot y' = y' = y \cdot 1_G = y$. On le note x^{-1}. On a $(x \cdot y)^{-1} = y^{-1} \cdot x^{-1}$. On ommet souvent le point, et on note $x \cdot y = xy$. De même, on parle souvent du groupe G plutôt que du groupe $(G, \cdot)$.

Une partie $H \subset G$ est un sous-groupe de G si H est stable pour la loi $\cdot$ et si H est un groupe pour la loi restreinte. En pratique, H est un sous-groupe de G si $H \neq \emptyset$, $\forall (x,y) \in H^2 \ xy \in H$ et $\forall x \in H \ x^{-1} \in H$.

Si A est une partie de G, on appelle sous-groupe engendré par A le plus petit sous-groupe de G contenant A (la notion de plus petit étant définie par l'inclusion, le minimum existant car l'intersection de deux sous-groupes est un sous-groupe). C'est le groupe formé des produits finis d'éléments de A et de leurs inverses.

1.2 Théorème de Lagrange

Le théorème de Lagrange est un théorème très important au sujet des sous-groupes des groupes finis.

Si G est un groupe fini (c'est à dire ayant un nombre fini d'éléments), on appelle ordre de G et on note $|G|$ son cardinal.

Théorème. *Si H est un sous-groupe d'un groupe fini G, alors l'ordre de H est un diiseur de l'ordre de G.*

▲. *Une relation binaire $\sim$ sur un ensemble E est une relation d'équivalence si elle est :*

(a) réflexive : $\forall x \in E \ x \sim x$;

(b) symétrique : $\forall (x,y) \in E^2 \ x \sim y \Rightarrow y \sim x$;

(c) transitive : $\forall (x,y,z) \in E^3 \ (x \sim y \wedge y \sim z) \Rightarrow x \sim z$.

Si $x \in E$, on appelle classe d'équivalence de x et on note $\bar{x} = \{y \in E \mid x \sim y\}$. Deux classes d'équivalence sont soit disjointes soit confondues, et $x \sim y \Leftrightarrow \bar{x} = \bar{y}$. L'ensemble $\{\bar{x} \mid x \in E\}$ des classes d'équivalence forme une partition de E. On le note $E/\sim$.

Revenons à notre sous-groupe H de G, et définissons la relation sur G $x \sim y \Leftrightarrow xy^{-1} \in H$. C'est une relation d'équivalence, car $xx^{-1} = 1_G \in H$, $xy^{-1} \in H \Rightarrow (xy^{-1})^{-1} = yx^{-1} \in H$ et $(xy^{-1} \in H \wedge yz^{-1} \in H) \Rightarrow xz^{-1} = xy^{-1}yz^{-1} \in H$.

Soit $y \in G$, soit $f : \left\{ \begin{array}{ccc} H & \longrightarrow & \bar{y} \\ x & \longmapsto & xy \end{array} \right.$. On a $f(x) = f(x') \Rightarrow x = f(x)y^{-1} = f(x')y^{-1} = x'$. Donc f est injective. Si $z \in \bar{y}$, $f(zy^{-1}) = z$, donc f est surjective. Donc f est bijective et $\operatorname{card} \bar{y} = |H|$. Donc toutes les classes d'équivalence ont le même cardinal, et $|G| = \operatorname{card}(G/\sim)|H|$.

1.3 Petit théorème de Fermat pour les groupes

Si $x \in G$ où G est un groupe, et si $n \in \mathbb{N}$, on note x^n le produit de n fois x, et on note $x^{-n} = (x^n)^{-1}$. On a les relations habituelles sur les puissances.

Théorème. *Soit G un groupe fini d'ordre n. Alors $\forall x \in G\ x^n = 1_G$.*

▲. *Soit $x \in G$. On considère le sous-groupe H des puissances de x. Il est fini, donc il existe $p < q$ tel que $x^p = x^q$. Alors $x^{q-p} = 1_G$. Soit N le plus petit entier strictement positif tel que $x^N = 1_G$. Alors $H = \{1_G, x, x^2, ..., x^{N-2}, x^{N-1}\}$, et tous ces éléments sont distincts. En effet, si $p \in \mathbb{Z}$, par division euclidienne, $p = Nq + r$, avec $0 \leqslant r < N$, et $x^p = x^r$, et si $x^p = x^q$, $p = q$ ou $|p - q| \geqslant N$. Donc d'après le théorème de Lagrange, l'ordre N de H divise l'ordre n de G, et $x^n = (x^N)^{\frac{n}{N}} = (1_G)^{\frac{n}{N}} = 1_G$.*

2 Arithmétique

2.1 Entiers modulo p

Soit $p \in \mathbb{N}^*$. Si $(a, b) \in \mathbb{Z}^2$, $(a + kp) + (b + k'p) = (a + b) + (k + k')p$ et $(a + kp)(b + k'p) = ab + (kb + k'a + kk'p)p$. Donc le reste de la division par p de $a + b$ et de ab ne dépendent que du reste de a et de b. On note $\mathbb{Z}_p$ et on appelle groupe des entiers modulo p l'ensemble des entiers «à p près» muni de l'addition.

On note $a \equiv b\ [p]$ le fait a et b ont le même reste par p, et on le lit «a est congru à b modulo p».

Exemple. On peut dresser la table d'addition de $\mathbb{Z}_4$:

+	0	1	2	3
0	0	1	2	3
1	1	2	3	0
2	2	3	0	1
3	3	0	1	2

2.2 Théorème de Bézout

Lemme. *Tout sous-groupe de $\mathbb{Z}$ est de la forme $\{an \mid n \in \mathbb{Z}\} = a\mathbb{Z}$ pour un certain $a \in \mathbb{N}$.*

▲. *Soit H un sous-groupe de Z, distinct de $\{0\}$. Soit $a = \min(H \cap \mathbb{N}^*)$. Soit $x \in H$, $x = aq + r$ avec $0 \leqslant r < a$. Alors $x - aq = r \in H$, $0 \leqslant r < a$, donc $r = 0$. $H = a\mathbb{Z}$.*

Théorème. *Si $(a, b) \in \mathbb{N}^2$ sont premiers entre eux, il existe $(u, v) \in \mathbb{Z}^2$ tels que $au + bv = 1$.*

▲. *Considérons le sous-groupe engendré par a et b, il est de la forme $d\mathbb{Z}$, avec $d > 0$. Donc d divise a et b, donc d divise $\operatorname{pgcd}(a, b) = 1$, $d = 1$. Donc 1 est une somme finie de a, de b et de leurs opposés. Donc $1 = au + bv$.*

2.3 Le petit théorème de Fermat pour les entiers

Théorème. *Si p est un nombre premier, et si $a \not\equiv 0\ [p]$, alors $a^{p-1} \equiv 1\ [p]$.*

▲. *Considérons l'ensemble des entiers non congrus à 0 modulo p muni de la multiplication. La multiplication est associative, 1 est élément neutre. Si $a \not\equiv p\ [p]$, alors a et p sont premiers entre eux, et $au + pv = 1$, alors $au \equiv 1\ [p]$, et $u \not\equiv 0\ [p]$. Donc cet ensemble est un groupe. Il est fini d'ordre $p - 1$, donc tout élément élevé à la puissancce $p - 1$ donne 1.*

2.4 Le théorème des restes chinois

Théorème. *Soient a et b des entiers naturels premiers entre eux. Soient p et q des entiers. Alors il existe un unique r tel que $0 \leqslant r < ab$, $r \equiv p\ [a]$ et $r \equiv q\ [b]$.*

▲. *Considérons $f : \mathbb{Z}_{ab} \longrightarrow \mathbb{Z}_a \times \mathbb{Z}_b$ qui à l'entier n modulo ab associe le couple des entiers n modulo a et n modulo b. $\mathbb{Z}_a \times \mathbb{Z}_b$ possède une structure de groupe par l'addition terme à terme, et f est compatible avec les lois de groupes, c'est à dire $f(m + n) = f(m) + f(n)$. Si $n \equiv 0\ [a]$ et $n \equiv 0\ [b]$, alors $a|n$ et $b|n$, donc $\operatorname{ppcm}(a, b)|n$. Comme a et b sont premiers entre eux, $\operatorname{ppcm}(a, b) = ab$, et $a|n \wedge b|n \Rightarrow ab|n$. Donc $f(n) = 0 \Rightarrow n = 0$. Si $f(m) = f(n)$,*

$f(m-n) = f(m) - f(n) = 0$, donc $n - m = 0$. Donc f est injective. les groupes $\mathbb{Z}_{ab}$ et $\mathbb{Z}_a \times \mathbb{Z}_b$ sont de même ordre ab, donc f est bijective, ce qui implique le théorème.

3 Le théorème du cryptage RSA

Théorème. *Si p et q sont deux nombres premiers entre eux distincts, et pour tout a et k entiers, on a*

$$a^{k(p-1)(q-1)+1} \equiv a \ [pq]$$

▲. *Si $a \equiv 0 \ [p]$, $a^{k(p-1)(q-1)+1} \equiv 0 \equiv a \ [p]$. Si $a \not\equiv 0 \ [p]$, $a^{k(p-1)(q-1)+1} \equiv a \times (a^{p-1})^{k(q-1)} \equiv a \times 1^{k(q-1)} \equiv a \ [pq]$. Donc dans les deux cas, $a^{k(p-1)(q-1)+1} \equiv a \ [p]$. De même, $a^{k(p-1)(q-1)+1} \equiv a \ [q]$. Les nombres p et q sont premiers entre eux, donc ces deux relations déterminent uniquement $a^{k(p-1)(q-1)+1}$ modulo pq. Or on a $a \equiv a \ [p]$ et $a \equiv a \ [q]$. Donc la solution est a.*

4 Mise en œuvre

4.1 Utilisation comme cryptage

On peut transofrmer tout message à transmettre en une suite de nombres dans un certain intervalle. Soit n un nombre à transmettre de la personne A à la personne B.

Supposons qu'on ait deux «grands» nombres premiers p et q, et deux nombres a et b dont le produit soit de la forme $k(p-1)(q-1)+1$. Alors si A calcule $n^a \ [pq]$ et l'envoie à B, qui va calculer $(n^a)^b \ [pq]$ et ainsi retrouver n.

On admettra que retrouver b à partir de pq et b est «intrinsèquement difficile», c'est à dire qu'il faut beaucoup de calculs, et qu'une petite augmentation de la taille des nombres se traduit par une augmentation énorme du temps nécessaire.

Alors on a deux possibilités. B peut avoir publié a et pq et gardé secret b. Dans ce cas, seul B pourra comprendre le message de A. A peut avoir publié b et pq et gardé secret a. Dans ce cas, n'importe qui peut donc lire son message, mais il est le seul à pouvoir l'avoir écrit.

Ces deux possibilités sont rendues possibles par le fait que le cryptage et le décryptage se font exactement par la même fonction. En pratique, A cryptera d'abord avec sa clef privée, puis avec la clef publique de B.

Un défaut de cette méthode est sa lenteur. En général, on se contentera de l'utiliser pour crypter une clef de cryptage traditionnelle, et d'utiliser cette dernière pour crypter le message lui-même.

4.2 Calcul des puissances

Il est totalement inenvisageable de calculer n^a pour ensuite le réduire modulo pq : pour les tailles usuelles, n^a possède plus de chiffres qu'il y a d'atomes dans l'univers observable. La bonne méthode consiste à réduire à chaque étape de l'élévation à la puissance.

De plus, on ne va pas calculer $n \times n \times n \times ... \times n$ a fois. Voici une méthode pour calculer x^y efficacement. Il ne faut pas oublier de réduire modulo pq à chaque multiplication :

```
r = 1
tant que y <> 0
    # résultat = r * x^y
    si y impair
        r = r * x
        y = y - 1
    # résultat = r * x^y
    x = x * x
    y = y / 2
    # résultat = r * x^y
retourner r
```

Cette méthode prend un nombre d'étapes proportionnel au nombre de chiffres de y.

4.3 Création de la clef

La partie la plus difficile consiste à trouver des nombres premier très grands. Le taille typique est d'environ 1000 chiffres en binaire. Attention : l'utilisation d'une clef de plus de 128 chiffres est interdite en France. La méthode classique pour savoir si un nombre est premier, à savoir tester tous les diviseurs, n'est pas envisageable (si elle l'était, le cryptage ne serait pas fiable). On utilise une méthode qui permet de savoir si un nombre p «a des chances» d'être premier : si p est premier, elle dit oui, s'il ne l'est pas, elle a une chance sur quatre de dire quand même oui. En multipliant les tests, on peut atteindre une quasi-certitude assez bonne pour avoir confiance en sa clef.

L'idée est de tester si p vérifie le théorème de Fermat. Mais on rafine un peu en constatant que seuls 1 et -1 peuvent avoir pour carré 1 modulo p, si p est premier. Voici donc la méthode, avec p impair, et $0 < t < p$ un nombre de test (aléatoire) :

```
p-1 = q * 2^r, q impair
n = t^q
si n = 1
    retourner oui
répéter r fois
    si n = -1
        retourner oui
    n = n * n
retourner non
```

Evidemment, c'est une méthode assez lourde. On gagne énormément à tester d'abord de petit diviseurs premiers.

Il faut ensuite trouver des nombres a et b dont le produit soit de la forme $k(p-1)(q-1)$. Ceci revient à trouver les coefficients de Bézout pour a et $(p-1)(q-1)$ en espérant qu'ils soient premiers. Pour celà, on remarque que $\text{pgcd}(x, y) = \text{pgcd}(x, y - x) = \text{pgcd}(x, y mod x)$. Voici donc une adaptation de l'algorithme d'Euclide pour calculer les coefficients de Bézout de x et y :

```
u1 = 1
v1 = 0
u2 = 0
v2 = 1
tant que y <> 0
    # x = u1*x_init + v1*x_init
    # y = u2*x_init + v2*y_init
    q = E(x/y) # E est la partie entière inférieure
    u2' = u1 - q*u2
    v2' = v1 - q*v2
    u1 = u2
    v1 = v2
    u2 = u2'
    v2 = v2'
    x' = y
    y = x mod y
    x = y'
# x = pgcd(x_init,y_init)
retourner (u1,v1)
```

Il n'y a plus qu'à essayer plusieurs a au hasard en attendant que l'un d'eux soit premier avec $(p-1)(q-1)$.

Conclusion

Maintenant, il n'y a plus qu'à choisir un protocole utilisant ces méthodes mathématiques, et à faire un programme rapide. Pour les calculs sur les grands entiers, on peut utiliser la bibliothèque `GNU MP`, qui est libre.