Il existe différents algorithmes asymétriques. L'un des plus connus est le **RSA** (de ses concepteurs *Rivest*, *Shamir* et *Adleman*). Cet algorithme est très largement utilisé, par exemple dans les navigateurs pour les sites sécurisés et pour chiffrer les emails. Il est dans le domaine public.

L'algorithme est remarquable par sa simplicité. Il est basé sur les nombres premiers.

Pour encrypter un message, on fait: **c = m^e mod n**

Pour décrypter: **m = c^d mod n**

m = message en clair
c = message encrypté
(e,n) constitue la clé publique
(d,n) constitue la clé privée
n est le produit de 2 nombres premiers
^ est l'opération de mise à la puissance (a^b : a puissance b)
mod est l'opération de modulo (reste de la division entière)

Créer une paire de clés

C'est très simple, mais il ne faut pas choisir n'importe comment **e**,**d** et **n**. Et le calcul de ces trois nombres est tout de même délicat.

Voici comment procéder:

1. Prendre deux nombres premiers **p** et **q** (de taille à peu près égale). Calculer **n = pq**.
2. Prendre un nombre **e** qui n'a aucun facteur en commun avec **(p-1)(q-1)**.
3. Calculer **d** tel que **ed mod (p-1)(q-1) = 1**

Le couple **(e,n)** constitue la clé publique. **(d,n)** est la clé privée.

Un exemple

Allons-y ! Commençons par créer notre paire de clés:

Prenons 2 nombres premiers au hasard: **p = 29**, **q = 37**

On calcul **n = pq = 29 * 37 = 1073**

On doit choisir **e** au hasard tel que **e** n'ai aucun facteur en commun avec **(p-1)(q-1)**:

(p-1)(q-1) = (29-1)(37-1) = 1008

On prend **e = 71**

On choisit **d** tel que **71*d mod 1008 = 1**

On trouve **d = 1079**

On a maintenant nos clés :

- La clé publique est **(e,n) = (71,1073)** (=clé d'encryptage)
- La clé privée est **(d,n) = (1079,1073)** (=clé de décryptage)

On va encrypter le message '**HELLO**'. On va prendre le code ASCII de chaque caractère et on les met bout à bout:

m = 7269767679

Ensuite, il faut découper le message en blocs qui comportent moins de chiffres que **n**. **n** comporte 4 chiffres, on va donc découper notre message en blocs de 3 chiffres:

726 976 767 900
(on complète avec des zéros)

Ensuite on encrypte chacun de ces blocs:

726^71 mod 1073 = 436
976^71 mod 1073 = 822
767^71 mod 1073 = 825
900^71 mod 1073 = 552

Le message encrypté est **436 822 825 552**. On peut le décrypter avec d:

436^1079 mod 1073 = 726
822^1079 mod 1073 = 976
825^1079 mod 1073 = 767
552^1079 mod 1073 = 900

C'est à dire la suite de chiffre **726976767900**.
On retrouve notre message en clair **72 69 76 76 79** : '**HELLO**'.

Dans la pratique

Dans la pratique, ce n'est pas si simple à programmer:

- Il faut trouver de grands nombres premiers (ça peut être très long à calculer)
- Il faut obtenir des nombres premiers **p** et **q** *réellement* aléatoires (ce qui est loin d'être évident).
- On n'utilise pas de blocs aussi petits que dans l'exemple ci-dessus: il faut être capable de calculer des puissances et des modulos sur de très grand nombres.

En fait, on utilise jamais les algorithmes asymétriques pour chiffrer toutes les données, car ils sont trop longs à calculer : on chiffre les données avec un simple algorithme symétrique dont la clé est tirée au hasard, et c'est cette clé qu'on chiffre avec un algorithme asymétrique comme le RSA.

P.G.P.

Si vous voulez encrypter vos fichiers, je vous recommande l'excellent logiciel **PGP** (Pretty Good Privacy) ou gpg (GNU Privacy Guard).

D'autres algorithmes, d'autres programmes...

Il existe d'autres algorithmes asymétriques, dont l'**ECC** (Elliptic Curve Cryptosystems, Encryptage par Courbe Elliptique). Ce système est basé sur une courbe paramétrique qui passe par un certain nombre de points de coordonnées entières. Ce n'est pas encore très développé, mais il est prometteur.

Il existe également **Diffie-Hellman**, de plus en plus préféré à RSA. (Diffie-Hellman avait rapidement été adopté par la communauté opensource quand RSA n'était pas encore dans le domaine public).