Rapport Méthode CHIFFREMENT & DÉCHIFFREMENT AVEC RSA

RSA ? C'est quoi au juste ?

Le chiffrement RSA (nommé par les initiales de ses trois inventeurs) est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman. RSA a été breveté par le Massachusetts Institute of Technology (MIT) en 1983 aux États-Unis.

Comment ça fonctionne ?

Vue d'ensemble

De manière générale le procédé RSA est schématiser par un échange entre deux personnes qu'on va on nommer Rayan et Sonia par exemple.

- Sonia veut transmettre un message à Rayan.
- Rayan crée deux clés :
  - ✓ une clé de chiffrement publique qu'il transmet à Sonia.
  - ✓ une clé de déchiffrement privée qu'il conserve soigneusement.
- Sonia utilise la clé publique pour chiffrer le message, et le transmets à Rayan.
- Rayan utilise la clé privée pour déchiffrer le message reçu.

Envoie un message chiffré avec la clé public.

Reçois le message et le déchiffre avec la clé privé.

Génération des clés

La génération des clés se fait selon les étapes suivantes :

- Soient deux grands nombres premiers « aléatoirement » choisis : *p* et *q*.
  Exp : p= 17483 , q=65951

- Notons : *n = p*q* et *φ = (p-1)*(q-1)*
  Exp : n = 17483*65951= 1153021333 et ϕ = (17483-1)*(65951-1)= 1152937900

- Soient *d* un grand entier « aléatoirement » choisi, premier avec *φ*. Et *e* l'inverse de *d* modulo *φ*.

- La clé publique de chiffrement est le couple (n,e), la clé privée de déchiffrement le couple (n,d).

Chiffrement/Déchiffrement

Chiffrement

- Avant d'être chiffré, le message original doit être décomposé en une série d'entiers *M* de valeurs comprises entre 0 et *n-1*.
- Pour chaque entier *M* il faut calculer *C≡M^e [n]*.
- Le message chiffré est constitué de la succession des entiers *C*.

Déchiffrement

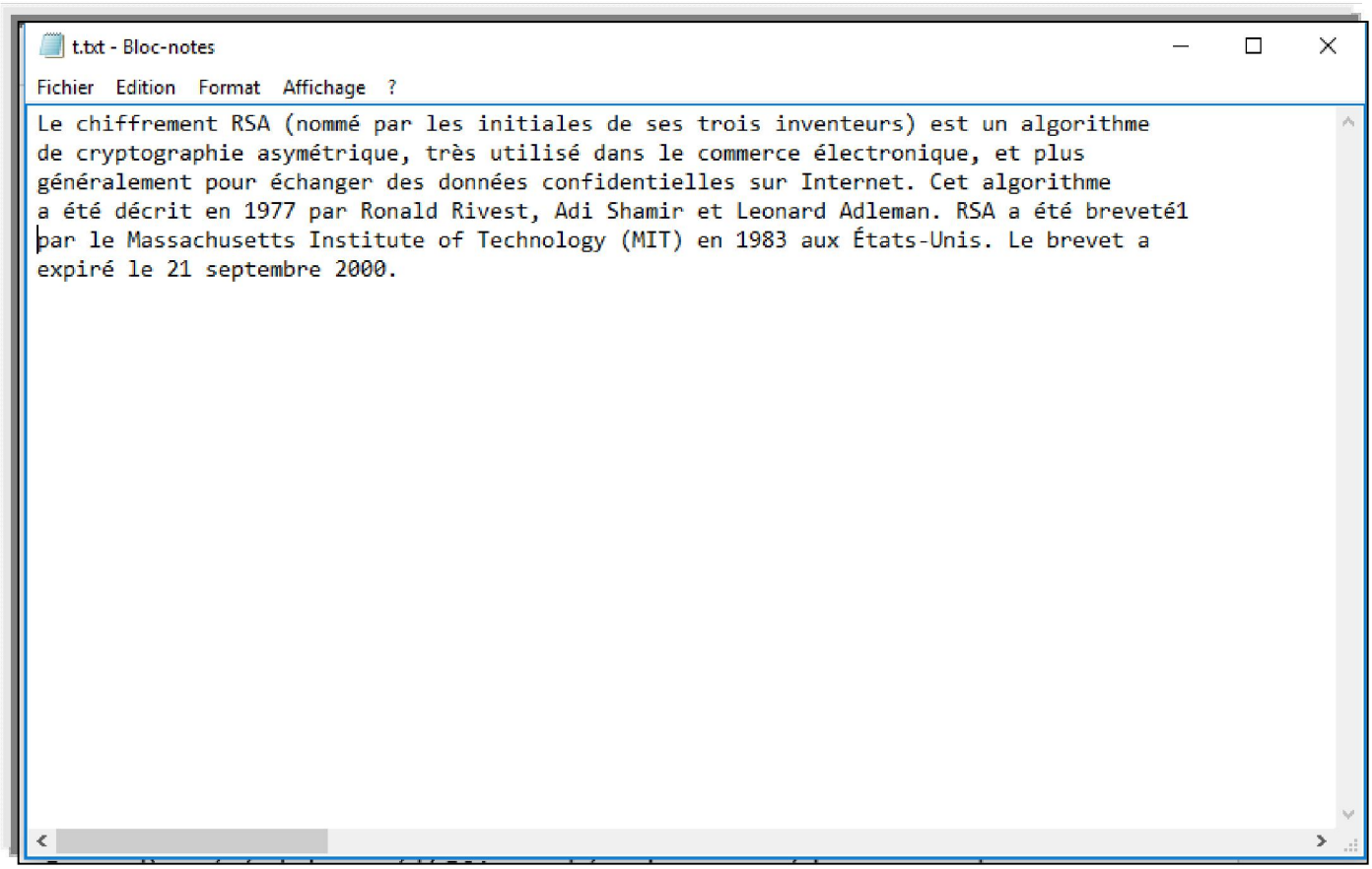
- Conformément à la manière dont il a été chiffré, le message reçu doit être composé d'une succession d'entiers *C* de valeurs comprises entre 0 et *n-1*.
- Pour chaque entier *C* il faut calculer *M≡C^d [n]*.
- Le message original peut alors être reconstitué à partir de la série d'entiers *M*.

Application du chiffrement RSA

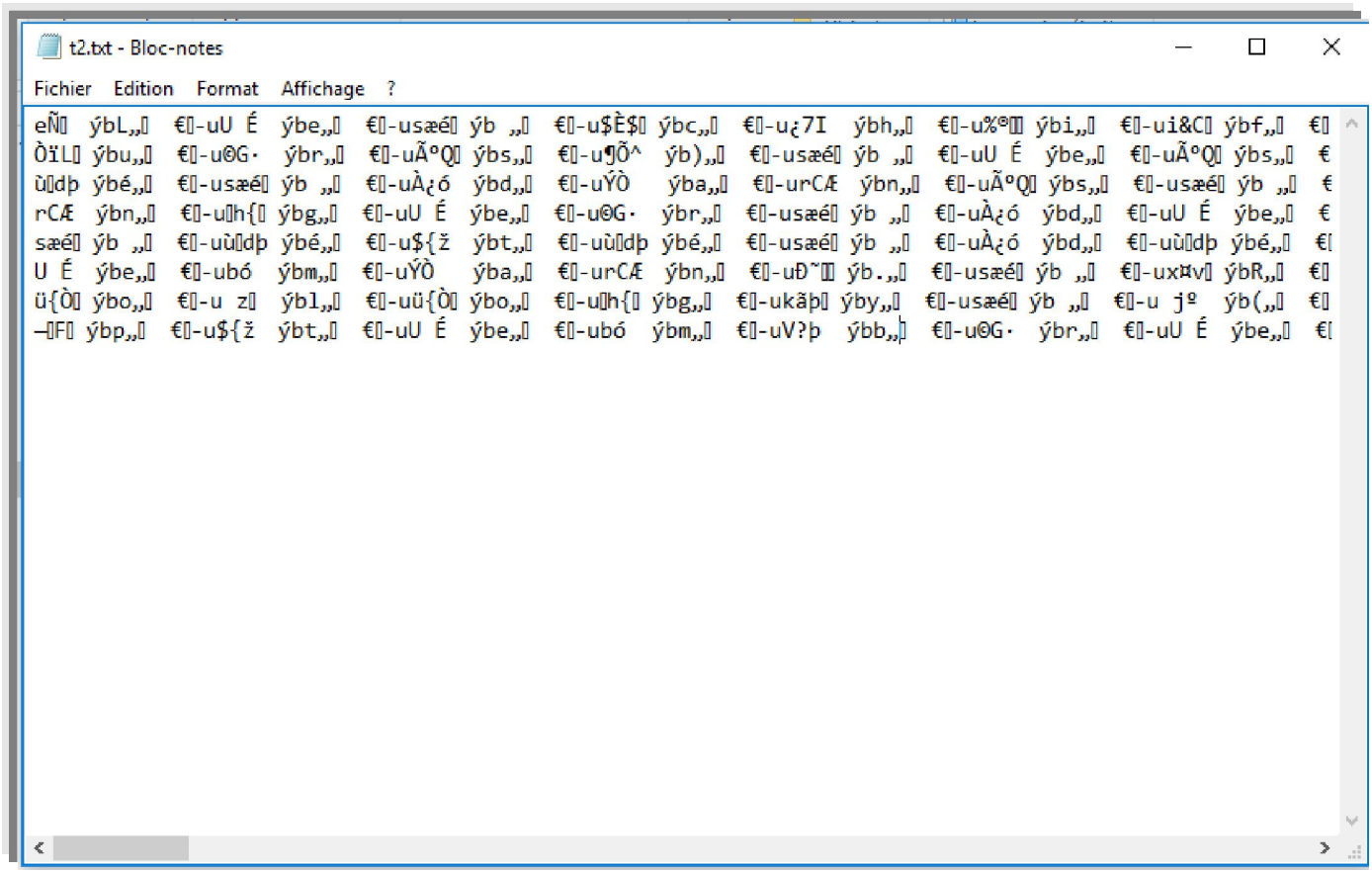
Dans notre cas l'algorithme RSA sera appliqué pour crypter un fichier texte.
Le fichier contient des caractères codée sous ascii donc la variation des valeurs de ces caractères vas de 0 à 255, c.à.d. $0 < M < 255$.

Dans l'exemple qui va suivre nous allons chiffrer un fichier nommé ' t.txt ' vers un fichier destination qui sera nommé ' t2.txt '.

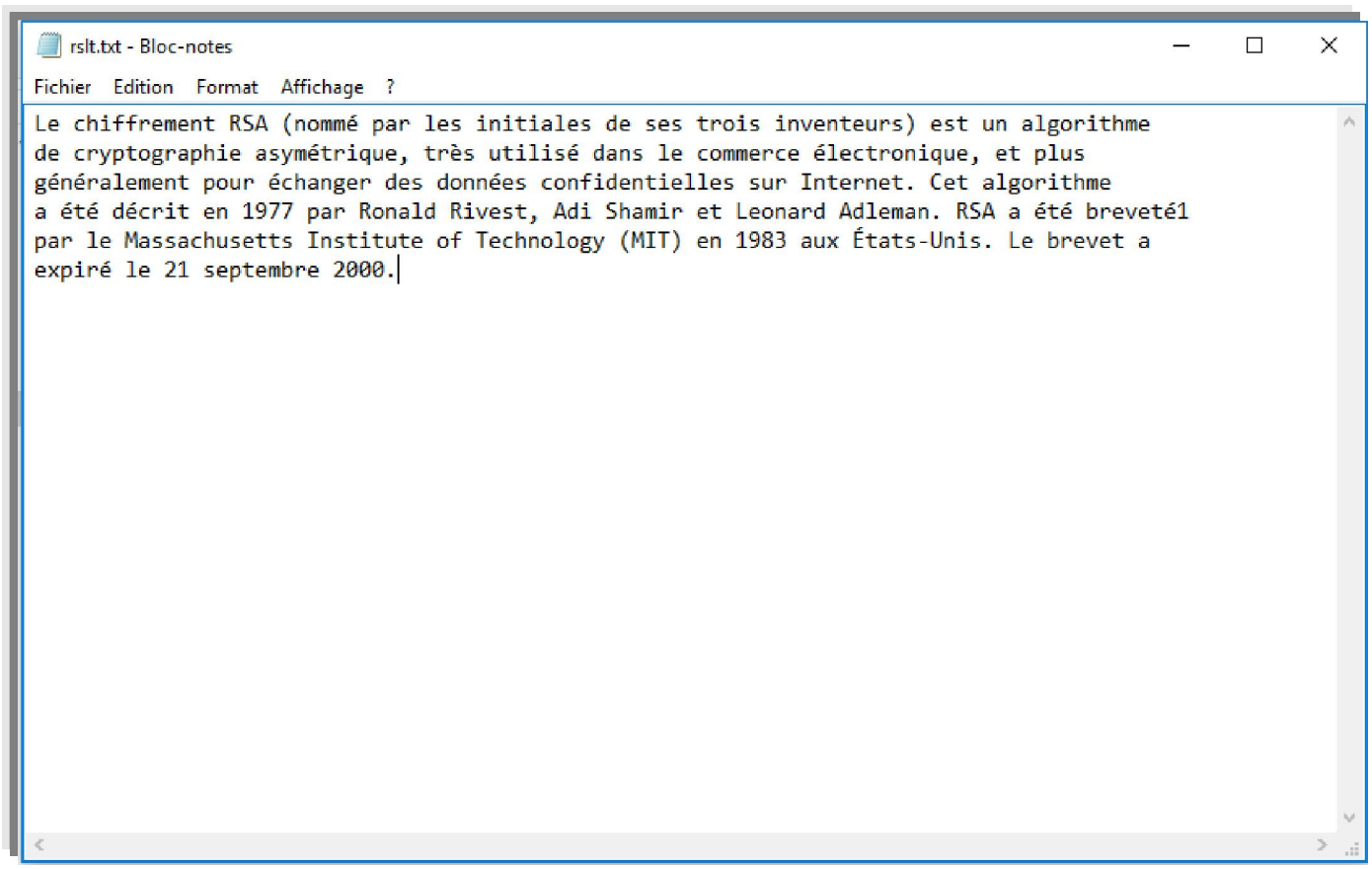
Dans l'étape suivante ce dernier ' t2.txt ' sera déchiffrer vers un fichier destination qui sera nommé ' rslt.txt '.



Ci-dessus on a le fichier 't.txt' qui sera chiffrer vers la destination 't2.txt'.



Après chiffrement le contenus du fichier ‘t2.txt’ ne ressemble en aucun cas à sa source ‘t.txt‘ donc le message a bien était crypté et il est maintenant illisible.



Apres déchiffrement le contenus du fichier ‘rslt.txt’ est identique à celui de ‘t.txt’ donc le décryptage c’est effectué avec succès.

Synthèse (Avantages et Inconvénients)

Le cryptage RSA demeure de nos jours le système de cryptographie à clé publique le plus utilisé et le plus fiable. En effet si l'on considère une clef RSA de 512Bits celle-ci sera cassée au bout d'environ 107 millions d'années par une machine modeste (architecture x86, ~1Ghz), une clef de 1024Bits sera quant à elle 1.3x10^154 fois plus complexe.
En revanche il est a noté que cette méthode sollicite une consommation conséquente des ressources mémoires un domaine où son concurrent direct le DES (Data Encryption Standard) fais nettement mieux. Alors évidemment cet inconvénient n'en est pas un sur un ordinateur récent mais il le devient quand il est question de ce type de cryptage pour une utilisation embarquée de type carte bancaire ou smartphone à titre d'exemple.

Ceci dit, en théorie le cryptosystème RSA n'est pas inviolable, il "suffit juste" de pouvoir factoriser n en produit de facteurs premiers. Cependant la puissance actuelle des supercalculateurs ne permet pas de réaliser ce travail à une échelle de temps raisonnable à partir du moment où n dépasse 150 chiffres.