La Cryptographie

&

Le RSA

Vincent S. – 9VSB 3

1.1 Introduction

Depuis toujours, l'homme a éprouvé le besoin de cacher ou de protéger des informations privées ou confidentielles. Pour ce faire, il s'est alors servi de la cryptographie.

Lors de l'Antiquité, la cryptographie était restreinte à un petit groupe de personnes qui avaient les capacités d'imaginer et de développer une telle idée (n'oublions pas qu'il n'y avait là aucun manuel ou aucune aide quelconque car la cryptographie n'existait pas encore vraiment). Ainsi, le risque que les messages codés ne soient découverts et décodés par d'autres personnes était faible. En effet, la plus grande partie des gens n'auraient même pas pu imaginer ce concept et même s'ils y avaient songé, les risques restaient minimes étant donné que la majorité de la population était illettrée.

Par exemple en Egypte, seuls les scribes et les personnes importantes ayant reçues une certaine éducation savaient lire et écrire les hiéroglyphes.

Dans le reste du monde la situation restait à peu près semblable jusqu'à environs 50 av. J.-C. L'homme prend enfin l'initiative de développer les techniques de protections des informations confidentielles de façon plus efficace.

Après la seconde guerre mondiale la situation a considérablement changé. L'éducation, l'accès à l'information et la connaissance sont devenus accessibles à presque tout le monde, avantageant énormément la créativité et la nouveauté et donc la complexité. Les méthodes de cryptographies se sont décuplées et la difficulté de cassage du code également. Cependant, la cryptographie a évolué uniquement dans des milieux fermés tels les gouvernements, les services secrets ou les armées. C'est pourquoi, pendant tant d'années, elle est restée une science secrète.

De nos jours en revanche, il y a de plus en plus d'informations dans les milieux publiques qui doivent rester confidentielles (les informations échangées par les banques par exemple). La cryptographie a commencé à être utilisée à des fins personnels peu après la création de l'Internet. En effet le cyberespace s'est énormément développé et s'est ouvert au grand public permettant par exemple l'envois de messages électroniques. Certaines personnes désirent alors que le contenu de ces messages reste discret. Cela est devenu possible grâce à certains logiciels qui sont distribués gratuitement sur l'Internet, permettant au grand public de se servir de la cryptographie.

Dans ce document se situe à la première partie, des explications concernant la cryptographie en général, il y a quelques définitions qui permettent de mieux comprendre les termes techniques. Il y a également une liste des algorithmes de cryptographie les plus connus, un historique assez vigoureux permettant de constater l'évolution des techniques au fil du temps et cette partie se conclu avec une observation de la place de la cryptographie de nos jours.

Ensuite, dans la seconde partie, se trouve décrite une technique de cryptage, le RSA. Son historique est décrit. si cette méthode est semblable à une autre, et pourquoi elle est ou n'est plus efficace de nos jours. L'algorithme y est démontré en expliquant chaque opération et en les commentant.

En annexe se trouve le code source d'une une petite application en langage C++ que j'ai créé afin d'aider à comprendre le fonctionnement de cet algorithme. Une brève description de ce que fait ce programme est aussi présente. Evidemment, la lecture du code n'aidera pas quelqu'un ne connaissant pas ce langage, même si le niveau de ce programme est débutant.

Voici comment est organisé mon dossier :

1.2 Définitions

La cryptographie désigne l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre incompréhensibles. Le fait de coder un message de telle façon à le rendre secret s'appelle chiffrement. La méthode inverse, consistant à retrouver le message original, est appelée déchiffrement.

La cryptanalyse est le fait d'essayer de déchiffrer ou de trouver une fonction qui permette le déchiffrement du message, que la clé de déchiffrement soit connue ou non. On peut aussi appeler cette action le « cassage ».

La cryptologie est la science qui étudie les différents aspects de la cryptographie et de la cryptanalyse.

Coder un message est le fait de transformer un message en une suite de nombres tandis que **Crypter un message** est le fait transformer, à l'aide d'un algorithme de chiffrement, un message déjà codé.

Le chiffrement

Le chiffrement est l'action de transformer une information claire, compréhensible de tout le monde, en une information chiffrée, incompréhensible. Le chiffrement est toujours associé au déchiffrement, l'action inverse. Pour ce faire, le chiffrement est opéré avec un algorithme à clé publique ou avec un algorithme à clé privée.

Le chiffrement par substitution consiste à remplacer dans un message un ou plusieurs bloc (de lettres par exemple) par d'autres blocs.
Il existe entre autre quatre types de substitutions :

- Mono-alphabétique **–** Cette méthode remplace chaque lettre du message par une autre lettre de l'alphabet. (par exemple **César**)
- Poly-alphabétique **–** Cette méthode utilise une suite de chiffres mono-alphabétiques (la clé) réutilisée périodiquement. Le principe est très simple : on chiffre la première lettre avec un premier alphabet, la seconde avec un second alphabet etc... (par exemple **Trithème** et **Vigenère**)
- Homophonique **–** Cette méthode fait correspondre à chaque lettre du message en clair un ensemble d'autres caractères. Par exemple la lettre A pourrait être chiffrée par 21, 25 et 26 et la lettre B 22, 23 et 24. Ainsi aucun chiffre n'apparaît plus souvent qu'un autre et donc rend le décryptage plus difficile.
- Polygramme **–** substitue un groupe de caractères dans le message par un autre groupe de caractères.

Les algorithmes

L'algorithme est une suite d'opérations et d'instructions à suivre pour exécuter une opération précise. Il est la structure de base d'un programme informatique mais on peut le trouver dans la vie de tous les jours comme par exemple les recettes de cuisine

Les algorithmes symétriques aussi appelés algorithmes à clés privées (ou secrètes). Lorsque l'on crypte une information à l'aide d'une clé secrète, le destinataire utilisera la même clé secrète pour décrypter. Il est donc nécessaire que les deux interlocuteurs se soient mis d'accord sur une clé privée auparavant, ce qui est un fort désavantage. Les algorithmes asymétriques ont été inventé afin d'éviter ce problème d'échange de clés secrètes préalable.

Voilà quelques algorithmes symétriques :

- **AES** (Advanced Encryption Standard)
- **Blowfish**
- **DES** (Data Encryption Standard)
- **IDEA** (International Data Encryption Standard)
- **RC2**, **RC4**, **RC5**, **RC6** (Rivest Cipher)
- **SEAL** (Software Optimized Encryption Algorithm)
- **TripleDES**

Les algorithmes asymétriques aussi appelés algorithmes à clés publiques et à clés privées. C'est à dire que pour crypter un message, on utilise la clé publique (connue de tous) et le message codé est envoyé au destinataire. Ce dernier se sert de sa clé privée (censée être connue de lui seul) pour décrypter le message. On évite enfin d'échanger clés de chiffrement et clés de déchiffrement, ce qui est un avantage.

Voilà quelques algorithmes asymétriques :

- **Diffie-Hellman**
- **DSA** *(Digital Signature Algorithm)*
- **RSA** *(Rivest, Shamir et Adleman)*

Les algorithmes de hachage sont des fonctions mathématiques qui convertissent une chaîne de caractères d'une longueur quelconque en une chaîne de caractères de taille fixe (appelée digest ou empreinte). On appelle emprunte le résultat d'une fonction de hachage.

- **MD2, MD4, MD5** (Message Digest)
- **RIPE-MD 128**
- **SHA1** (Secure Hash Algorithm)
- **Tiger**

1.3 Histoire de la Cryptographie

Les premières méthodes de chiffrement [Antiquité]

1900 av. J.-C. Un scribe égyptien utilise des hiéroglyphes qui ne sont pas standards racontant la vie de son maître. Le but n'était pas de rendre le texte incompréhensible mais plutôt de lui donner un caractère plus solennel.

1600 av. J.-C. Le premier « document » chiffré connu remonte à l'Antiquité. Il s'agit d'une tablette d'argile, retrouvée en Irak. Un potier y avait gravé sa recette secrète en supprimant des consonnes et en modifiant l'orthographe des mots.

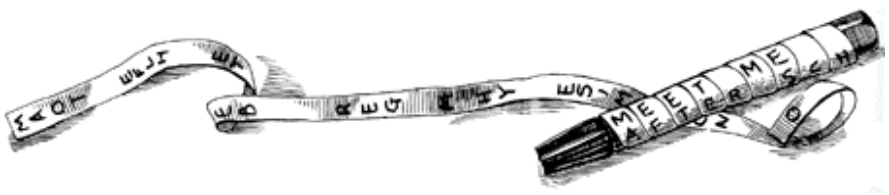
600 av. J.-C. Un roi de Babylone écrit sur le crâne rasé de ses esclaves, attend que leurs cheveux aient repoussé, et les envoie à ses généraux. Il suffit ensuite de raser à nouveau le messager pour lire le texte.

~ 600 av. J.-C. La Mésopotamie, grande civilisation de l'antiquité, avait atteint un niveau cryptologique étonnamment moderne. On a retrouvé en Iran des fragments de tablettes où des nombres correspondaient à des mots.

500 av. J.-C. Des scribes hébreux emploient le ATBASH, un simple algorithme de chiffrement par substitution utilisant l'alphabet renversé, afin de transcrire le livre de Jeremiah. (Par exemple bonjour devient « ruojnob »).

487 av. J.-C. Des grecs utilisent une scytale, aussi appelé bâton de Plutarque (historien et moraliste de la Grèce Antique). Il s'agit d'un bâton autour duquel on enroulait une longue et mince bande de cuir sur laquelle on écrivait notre message secret (souvent codé). Une fois la bande déroulée, il était difficile de retrouver le message. Seule le destinataire, connaissant le diamètre du bâton de celui ayant servi à écrire le message, pouvait le déchiffrer.

La principale faiblesse de ce système est qu'un bâton de diamètre approximativement égal suffit à déchiffrer le texte. La sécurité reste donc sur le secret autour du procédé de chiffrement du message.



Les premiers systèmes cryptographiques [Ancien Âge]

150 av. J.-C. Polybe, historien grec, imagine procédé de chiffrement très innovant pour son temps. Cette méthode utilise un système de transmission basé sur un carré de 25. Cependant l'alphabet français contient 26 lettres donc il faut supprimer une lettre, en général il s'agit du W.

Les cryptologues modernes ont vu dans cette méthode plusieurs caractéristiques très intéressantes dont la conversion de lettres en chiffres, la représentation de chaque lettre par deux éléments séparés.

50 av. J.-C. Jules César utilise une substitution dans l'alphabet pour les communications gouvernementales. En effet il décalait la lettre qu'il souhaitait coder de 3 lettres vers la droite (en revenant au début de l'alphabet si besoin).

200 Le papyrus de Leyde est le plus ancien manuscrit connu concernant l'alchimie. Il utilise un algorithme de chiffrement pour cacher les parties importantes de certaines recettes.

1499 Jean Trithème (1462-1516), ancien abbé, est considéré comme un des pères de la cryptographie. En effet, il est l'auteur d'un des premiers systèmes poly-alphabétiques et il a créé une technique de stéganographie (fait de cacher un message au sein d'un autre message) où les lettres sont remplacées par des mots choisis de manière à former, par leur réunion, une prière par exemple

1560 Blaise de Vigenère (1523 - 1596) est l'auteur de l'un des premiers systèmes de substitution poly-alphabétique, il utilise donc une clé. Cette méthode restera dominante pendant trois siècles. Sa particularité est qu'il n'utilise non pas un alphabet, mais 26 alphabets décalés pour chiffrer un message. On peut résumer ces décalages avec un carré de Vigenère. Ce chiffre utilise une clé qui définit le décalage pour chaque lettre du message.

La Cryptographie moderne [Temps Modernes à Nos Jours]

1918 Gilbert Vernam met au point l'algorithme One Time Pad (traduisez masque jetable) aussi appelé chiffre de Vernam.

En effet il fonctionne sur le même principe que le chiffrement de Vigenère, avec quelques règles supplémentaires : Une clé ne doit être utilisée qu'une seule fois, elle doit être de la même taille que le message et elle doit être générée aléatoirement. Ce système est donc reconnu comme étant l'algorithme de chiffrement le plus sécuritaire.

Cependant la communication des clés pose problème car n'oublions pas que la clé doit être de la même taille que le message codé, il est donc hors de question d'utiliser Internet comme passerelle. Pour donner un exemple, on peut penser à la valise diplomatique que les gouvernements utilisent pour communiquer les clés privées de façon sûre à leurs ambassades.

1923 Le Dr Arthur Scherbius, hollandais résidant en Allemagne, met au point une machine nommée Enigma qui sert à encoder des messages. Pendant la guerre, des versions d'Enigma sont utilisées pour pratiquement toutes les communications radio allemandes ainsi que pour les communications télégraphiques. Même les bulletins météos sont codés avec Enigma. Elle continuera à être utilisée dans l'armée encore jusqu'en 1939 (date à laquelle le code de cette machine à été cassé).

1976 IBM publie un algorithme basé sur Lucifer (l'une des premières méthodes de chiffrement moderne destiné à un usage civil). Il devient le DES (Data Encryption Standard). C'est un chiffrement qui transforme des blocs de 64 bits avec une clé secrète de 56 bits au moyen de permutations et de substitutions. Le DES est considéré comme étant raisonnablement sécuritaire.

1978 Le RSA est inventé par Ronald L. Rivest, Adi Shamir et Leonard M. Adleman. Il est un des plus populaire des systèmes à clés publiques.

1992 Le MD5 (Message Digest 5) est développé par Ronald L. Rivest. Il s'agit d'une fonction de hachage. Très utilisée sur l'Internet mais n'est pas considéré comme étant un algorithme sûr.

2000 L'AES (évolution de l'algorithme Rijndael) devient le standard du chiffrement avancé pour les organisations du gouvernement des Etats-Unis.

2005 La cryptographie quantique, qui repose sur la physique quantique, serait considérée comme sûre à presque 100%. Ce sera peut-être la méthode du futur car elle est toujours en cours d'expérimentation.

1.4 Observations Et Conclusion

La liste ci-dessus est bien évidemment un résumé des plus grandes avancées et des découvertes les plus significatives concernant la cryptographie. Il y a bien d'autres personnes qui ont créé des cryptosystèmes plus ingénieux les uns que les autres. J'ai décrit ceux qui, d'après-moi, étaient les plus importants et ceux qui ont permis d'améliorer l'intégrité et la sécurité des informations cryptées.

Ce qu'on peut constater, c'est que depuis environs le début du 20^{e} siècle, les techniques et les recherches se sont décuplées. La cryptographie est rapidement passée de science secrète à outil public. Depuis les récentes études sur la cryptographie quantique, on peut s'attendre à une sécurité quasi inviolable et donc, sûrement une baisse du piratage informatique. Développons un peut le point de la cryptographie quantique. Afin de se donner une idée du fonctionnement de cette méthode, on sait qu'elle repose sur la physique quantique, qu'il est, d'après cette dernière, théoriquement impossible de connaître la vitesse et l'endroit où se site un électron par rapport à un atome à un moment précis. Plus on essaie de définir la vitesse, plus le lieu est incertain et vice-versa. N'oublions pas que pour l'année 2005, il est impossible de trouver ces deux paramètres, mais peut-être que dans 10ans, 20ans, voir 200ans (pourquoi pas), on aura trouvé le moyen de les définir.

Cela pour dire que les méthodes de cryptages se complexifient au fil du temps, mais les méthodes de cassages également. Donc peut-être que dans quelques années, certaines méthodes seront obsolètes alors d'autres les remplaceront et ainsi de suite.

La cryptographie est une technique qui change progressivement en fonction des conditions politiques, économiques et des législations des pays (loi concernant la protection de la vie privée etc…). On peut aussi dire que la cryptographie n'évoluerait pas si personne n'essayait de casser les codes de cryptages, car il n'y aurait plus aucune raison de les améliorer étant donné que personne ne parviendrait à les casser. Les méthodes de cryptage seraient donc jugées sûres à 100% et elles le resteraient en attendant que quelqu'un réussisse à passer les mesures de sécurités

2.1 Le RSA - Introduction

Le RSA est l'un des plus connus des algorithmes asymétriques. C'est à dire qu'on utilise une clé publique que tout le monde connaît qui permet de crypter le texte et une clé privée, dont seul le destinataire du message est censé la connaître afin de décrypter le message. Ce qui est important avec ce système, c'est qu'*on ne peut pas* retrouver la clé privée à partir de la clé publique.

Il a été inventé en 1978 par Ronald L. Rivest, Adi Shamir et Leonard M. Adleman, des chercheurs au MIT (Massachusetts Institute of Technology).

Cette méthode est très utilisée de nos jours par exemple, dans les navigateurs Internet afin de garantir la sécurité de certains sites ou encore pour chiffrer des emails. Il est aussi le standard de chiffrement du secteur bancaire de plusieurs pays.

Afin de crypter, nous avons besoin d'un message codé (il faut donc que le message d'origine soit transformé en nombres). Le RSA utilise l'arithmétique des congruences modulos n, c'est à dire qu'il utilise les restes des divisions. Il est alors possible, pour certains nombres, de trouver deux autres entiers qui nous permettent, par le calcul de puissance, de retomber sur le reste de départ (ainsi un premier calcul crypte et un second nous ramène sur ce chiffre de départ, donc décrypte). Un des principaux avantages du RSA, c'est qu'il crypte par blocs de caractères, donc il ne conserve pas la fréquence d'apparition des lettres.

Ce qu'utilise le RSA

Ci-dessous, nous allons voir quelques notions de mathématiques afin de comprendre ce qui est utilisé avec le RSA.

- **Les nombres entiers naturels**

  On dit qu'un nombre est « entier naturel » quand ce nombre existe dans la nature (qu'on peut l'utiliser pour compter des objets). Par exemple, on peut dire qu'il y'a 1 arbre ou 2 arbres mais pas 1.5 arbre ou 0,8 arbre. « 0, 1, 2, 4, 5, 6, 7... 25, 904 etc... » sont des entiers naturels. Avec le RSA on considérera uniquement les nombres entiers naturels.

- **Les nombres premiers**

  Un nombre premier est un entier naturel plus grand que 1 et qui ne possède que deux diviseurs : lui-même et 1. Tout nombre entier est le produit de deux nombres premiers. Les premiers permettent alors de produire tous les entiers. Par exemple, les nombres « 2, 3, 5, 7, 11, 13 » sont premiers.

- **Le modulo**

La notation du modulo est : **a _ b (mod, c)**
Soit b est le reste de la division de a par c

Pour donner un exemple de l'utilisation du modulo, on peut imaginer de partager 269 francs entre 11 personnes. On souhaite distribuer l'argent équitablement. Chaque personne doit donc recevoir le plus possible et la même somme. On commence par donner un franc à chaque personne, et ceci à tour de rôle. Au bout de 11 tours, on aura remis 24 francs à chacune des 11 personnes mais il reste encore 5 francs qui ne peuvent pas être partagées équitablement. Ces 5 francs restant, c'est 269 modulo 11 (où 5 est le résultat de la division euclidienne, car 269 = 24*11 + 5)

On sait aussi que 11 _ 1 (mod, 5) car si prend le reste de la division de 11 par 5 on obtient :

```
  1 1  _5
       ‾‾
- 1 0   2
‾‾‾‾‾‾
    1
```

Ce qui donne 11 = 5*2 + 1

Essayons avec une division sans reste :

```
  3 6  _6
       ‾‾
- 3 6   6
‾‾‾‾‾‾
    0
```

Ce qui donne 36 = 6*6 + 0 et donc, 36 _ 0 (mod, 6).

- **Les théorèmes et algorithme**

Avec le RSA, afin de prouver l'existence de certains nombres, on devra utiliser des théorèmes. Ils seront décrits au moment voulu. Cependant, voici les deux théorèmes qui sont utilisés :

- Le Petit Théorème de Fermat
- Le Théorème de Bezout

2.2 Fonctionnement

Le Cryptage

On travaille principalement avec des nombres premiers (donc entiers naturels). A priori, il est assez facile de dire si un petit nombre est premier ou non. Cependant, dès que ce nombre devient plus conséquent, il est très rapidement difficile de dire s'il s'agit d'un premier ou non. Voici un atout principal du RSA.

- On choisit donc deux nombres premiers très grand p et q qui serviront à former les clés publiques et privées.

- On calcul n, qui est un constituant de la clé publique et de la clé privée en faisant : **n = p*q**.

- Ensuite on calcul e, qui fait partie de la clé publique, avec :
  e est un nombre premier avec (p-1)*(q-1).

- La couple **(n, e)** forme ainsi la clé publique de chiffrement.

On peut commencer le cryptage. On va se servir de la clé publique de chiffrement. Tout d'abord, il faut un message préalablement codé. On doit donc transformer en nombres le message d'origine (en utilisant la valeur ASCII de chaque lettre ou encore en remplaçant chaque lettre par son rang dans l'alphabet par exemple).

On a un message codé nommé M. Il faut ensuite découper le message en blocs strictement inférieurs à n. On nomme alors chacun des bloques codés Mi, le texte crypté C et chacun des bloques cryptés Ci. Dans la réalité, les blocs sont très longs et les clés contiennent une centaine de chiffres (voir plus).

- Afin de crypter, on fait **Ci _ Mi^e (mod, n)**. Autrement dit, Ci (étant le bloc de texte crypté), équivaut au reste de la division de Mi^e par n.

Le Décryptage

- Il s'agit désormais de calculer, à partir de p et q, la clé privée d. Pour cela, il faut satisfaire l'équation **e*d (mod, (p-1)*(q-1)) = 1**.

- Avec la clé d générée, on peut décrypter bloc par bloc le texte crypté. On retrouve alors le message M.

- **Mi _ Ci^d (mod, n)**. Autrement dit, Mi (étant le bloc de texte décrypté), équivaut au reste de la division de Ci^d par n. On retrouve alors nos blocs codés et il ne reste plus que les transférer vers leur équivalant dans l'alphabet défini.

Dans la Pratique

La norme RSA était de choisir un nombre n de 155 chiffres. En 1999, un nombre de 155 chiffres (qui servait justement de clé à un système RSA) a été décomposé en produit de deux premiers de 78 chiffres chacun. Le temps cumulé de calcul a été évalué à 8'000 millions d'instructions par secondes durant un an. Depuis, la Société *RSA Data Security* recommande des nombres de 309, voire 617 chiffres. Les deux nombres premiers secrets p et q sont si grands (ils dépassent les 100 chiffres) qu'il est quasi impossible de les retrouver connaissant le nombre public n et donc il est quasi impossible de retrouver la clé privée d.

Pour le chiffrement, on utilise généralement une combinaison de RSA et d'un algorithme symétrique. On chiffre tout d'abord les données à l'aide d'une clé symétrique aléatoire puis ensuite on chiffre cette clé à l'aide de la clé. Le destinataire déchiffrera cette clé symétrique avec sa clé RSA privée et ensuite il déchiffrera les données avec sa clé symétrique.

Si on souhaitait doubler la taille de la clé RSA, on multiplierait par 3 le temps des opérations utilisant la clé publique, par 8 le temps des opérations utilisant la clé privée et par 16 le temps de génération des clés. On voit encore une fois pourquoi il est préférable qu'on utilise le RSA avec un algorithme symétrique.

Standards

Le RSA fait partie de nombreux standards tels que :

- ISO 9796
- ANSI X9.31 rDSA
- ITU-T X.509 (standard de sécurité)
- ETEBAC 5 (standard français du secteur bancaire et financier)
- X9.44 draft (standard américain du secteur bancaire et financier)
- SWIFT (standard interbancaire international)

2.3 Démonstration Mathématique

- **On commence donc par définir p et q qui :**

  - Doivent être des nombres entiers naturels premiers.
  - Doivent être le plus grand possible et ils ne doivent pas être très proches l'un de l'autre par question de sécurité, de façon qu'ils ne soient pas facilement retrouvés à l'aide de n.
  - Doivent être connus que par le récepteur du message, puisqu'ils constituent la solution de génération des clés.

- **On définit la clé publique n avec :**

  n = p*q

- **On définit la clé publique e avec :**

  e [est premier avec] (p–1)*(q–1)

  Il y a alors plusieurs possibilités pour le choix de e mais il doit bien évidemment être inférieur à **(p–1)*(q–1)**

- **Le couple (n, e) forme la clé publique de cryptage**.

  Tout le monde peut la connaître afin de crypter un message, mais « personne » ne peut décrypter les messages sans la clé privée.

- **On nomme M, Mi, C, Ci**

  M est le message d'origine déjà codé.
  Mi est un bloc du message d'origine **M** (avec Mi < n)
  C est le message crypté
  Ci est un bloc du message cryptée **C** (avec Ci < n)

- **Maintenant il faut trouver la clé privée d.**

  - **Preuve de l'existence de d et k**

Le petit théorème de Fermat dit que si p est un nombre premier alors 1 est le reste de la division de Mi^{p-1} par p, soit **Mi^{p-1} _ 1 (mod, p)** (Pour 1 < Mi < n).

Le fait que e et d soient des entiers naturels tel que :
ed – k(p–1)*(q–1) = 1 et e premier avec (p–1)*(q–1) est prouvé avec le Théorème de Bezout.

Théorème de Bezout :

x et y sont premiers entre eux s'il existe u et v dans _ tel que :
xu + vy = 1. On donne : **x=e** et **y=(p–1)*(q–1)**
donc pour que **e*u + v(p–1)*(q–1) = 1** il faut que **e** soit premier avec **(p–1)*(q–1)** et justement, c'est le cas.

Avec **e*u + v(p–1)*(q–1) = 1** on a alors u et v dans _, c'est à dire qu'ils ne sont pas forcément réels. Comment alors trouver u et v entiers réels de manière que **u=d** et **v=k** ? On peut avoir u comme entier réel mais alors v serait forcément négatif, et donc il ne pourrait pas valoir k...

Notons tout de même ce que cela donnerait :
e*d + (–v) * ((p–1)*(q–1)) = 1

Afin d'avoir v entier, la solution est évidente :
e*d – v(p–1)*(q–1) = 1

v peut désormais valoir k tel que **e*d – k(p–1)*(q–1) = 1**

Théorème de Fermat :

Ce théorème utilise la notion de factoriel et de combinaisons d'éléments, ce qui dépasse mon niveau de mathématique. C'est pourquoi je ne prouverai que brièvement ce théorème.

Brève preuve du Théorème de Fermat :

Puisque e est premier avec (p-1)*(q-1) il existe des entiers d et k tel que e*d – k(p–1)(q-1) = 1.

On peut enfin résoudre d avec :

e*d = k(p–1)(q–1) +1
ou encore
e*d = 1 (mod, (p–1)(q–1))

Car : **k(p–1)(q–1) +1 _ 1 (mod, (p–1)(q–1))**

Par exemple avec **p = 3**, **q = 5**, **(p–1)(q–1) = 8**.
Donc **k*8 +1 _ 1 (mod, 8)**
k*8 +1 = 2*8 +1 = 17

Ainsi :

```
  1 7  _8
       __
- 1 6   2
_______
    1
```

Et **17 _ 1 (mod, 8)**

donc **e*d – k(p–1)(q–1) = 1** (avec 0 < d < (p-1)*(q-1))

Afin de trouver d, il faut connaître (p–1)(q–1). Si on connaît p et q, il est facile de le calculer, or si quelqu'un voulait décoder un message, il lui sera difficile de trouver p et q seulement à l'aide de n (Evidemment p, q et n doivent être très grands). Néanmoins il existe des algorithmes pour trouver p et q mais ils sont très lents surtout si n l'est aussi...

- **<u>Preuve que si p et q sont premiers on a :</u>**

Pour tout Mi < n : $\mathbf{Mi}^{\mathbf{k*(p-1)(q-1)+1}} \_ \mathbf{Mi\ (mod, n)}$ (modulo n, car n = p*q)

<u>Preuve :</u>

$\mathbf{Mi}^{\mathbf{k*(p-1)(q-1)+1}} \_ \mathbf{Mi} * \mathbf{Mi}^{\mathbf{k*(p-1)(q-1)}}$ **(mod, n)** (loi des exposants)
Donc $\mathbf{Mi}^{\mathbf{k*(p-1)(q-1)}} \_ \mathbf{1(mod, n)}$

Où $\mathbf{Mi}^{\mathbf{ed}} \_ \mathbf{Mi\ (mod, n)}$ car :
k(p–1)(q–1) +1 = e*d

Car d'après le théorème de Fermat $\mathbf{Mi}^{\mathbf{p-1}} \_ \mathbf{1\ (mod, p)}$
et $\mathbf{Mi}^{\mathbf{q-1}} \_ \mathbf{1\ (mod, q)}$
(Evidemment, avec **p<n**, et donc **q<n**...)

Donc Mi^{ed} _ Mi (mod, p) et Mi^{ed} _ Mi (mod, q)

Dans ce cas,
$Mi^{(p-1)(q-1)}$ _ 1 (mod, n)

1 modulo n car n = p*q, et 1 car le PGDC (Plus Grand Diviseur Commun) de deux nombres premiers est forcément 1…

Mi est le reste de la division de **$Mi^{k * (p-1)(q-1) +1}$** par **n**.
(où **p*q = n** et **k(p–1)(q–1) +1 = e*d**).
On peut alors dire que **$Mi^{k * (p-1)(q-1) +1} = Mi^{ed}$**

On revient donc au point où :
Mi^{ed} _ Mi (mod, n)

Dans ce cas si **Ci _ Mi^{e} (mod, n)**
Alors pour retrouver la valeur initiale de Mi on utilise :
Mi _ Ci^{d} (mod, n)

Le cryptage se fait alors avec Ci _ Mi^{e} (mod, n)
Et le Décryptage avec Mi _ Ci^{d} (mod, n)

2.3 Un Exemple Simple

- On prend un alphabet simple de quatre lettres :
  A – B – C

- On code cet alphabet de façon que :
  A = 1
  B = 2
  C = 3

- On choisit les deux nombres premiers p et q :
  p = 2
  q = 5

- On calcule la clé publique n avec :
  n = p*q = 2*5 = 10

- On définit la clé publique e de manière que e soit premier avec (p–1)*(q–1).
  On a alors :
  (p–1)*(q–1) = (2–1)*(5–1) = 1*4 = 4
  e [premier avec] 5
  e = 3

- On calcul d avec e*d – k((p–1)(q–1)) _ 1 par exemple :
  e*d – k((p–1)(q–1)) _ 1
  3*d – k(4) _ 1
  3*3 – 2*4 _ 1 car 9 = 3 * 3 + 1

  - **donc la clé privée d = 3**

- Clé publique : (n, e) = **(10, 3)**
- Clé privée : (n, d) = **(10, 3)**
  (dans la pratique, la clé publique n'est jamais égale à la clé privée)

- Cryptage du message : **Ci _ Mi^{e} (mod, n)**

Message : **BAC**
Message Codé : **3 2 4**

- Crypter A : Mi = A = 1
  Ci _ 1^3 (mod, 10) = **1**

- Crypter B : Mi = B = 2
  Ci _ 2^3 (mod, 10) = **8**

- Crypter C : Mi = C = 3
  Ci _ 3^3 (mod, 10) = **7**

Message Crypté : **8 1 7**

- Décryptage du message : **Mi _ Ci^d (mod, n)**

Message Crypté : **8 1 7**

- Décrypter 8 : Ci = 8
  Mi _ 8^3 (mod, 10) = **2**
- Décrypter 1 : Ci = 1
  Mi _ 1^3 (mod, 10) = **1**
- Décrypter 7 : Ci = 7
  Mi _ 7^3 (mod, 10) = **3**

Message Codé retrouvé : **2 1 3**
Message décodé : B A C

2.4 A Propos

Les atouts principaux du RSA sont le fait que ce langage travaille avec les nombres premiers. Or factoriser un très grand nombre premier en deux autres prend beaucoup de temps.

Trouver de grands nombres premiers est lent. Dans la plupart des cas on utilise des algorithmes probabilistes de primalité afin de déterminer p et q, comme le test de Miller-Rabin. Ces nombres ont donc une certaine probabilité de primalité.

Le fait que le RSA utilise une clé publique et une clé privée est un bon point étant donné que le problème de communications des clés est résolu (contrairement aux algorithmes symétriques).

De plus, avec cette méthode, on crypte par bloc de caractères, ce qui empêche à un cryptanalyste d'utiliser l'analyse des fréquences afin d'essayer de casser les clés.

Le problème avec une telle méthode, c'est qu'il faut utiliser de grandes clés, et leur génération prend quand même un certain temps. De plus le cryptage également, puisqu'on travaille avec de très grandes clés, donc puissances (c'est pourquoi on l'utilise avec un algorithme symétrique, comme cité plus haut)

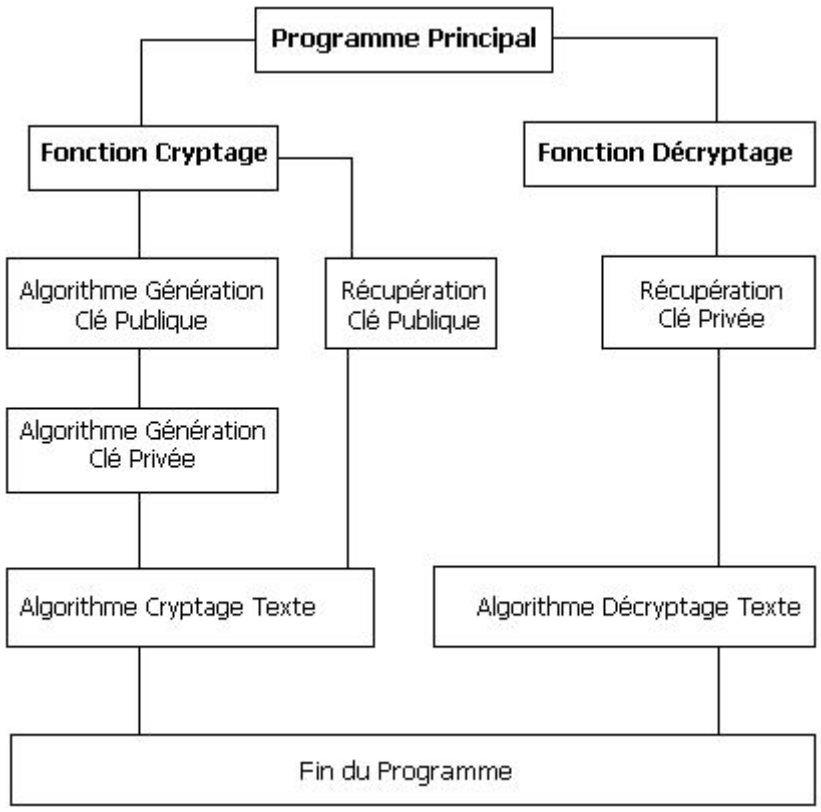
Il existe des algorithmes pour casser les clés, mais ceux-ci sont très lents et donc la plupart des fois inefficaces dans des délais raisonnables.

Annexe Le C++

Le C++ est un langage de programmation évolué du « C », inventé par un ingénieur nommé Dennis Ritchie puis amélioré par un autre chercheur. La particularité de ce langage était le fait qu'il était à la fois *un langage système*, c'est à dire qu'on pouvait l'utiliser pour accéder à toutes les ressources systèmes d'un ordinateur (mémoire, registres, ports etc…) et à la fois un langage évolué (qui utilise des syntaxes très proche de l'anglais, facilitant l'écriture et la mise à jour des programmes).

Débutant en programmation, je me suis aidé des ouvrages « *Le Langage C++*, par Stéphane Dupin, aux éditions *CampusPress* » ainsi que « *C++ par la pratique*, par Jean-Cédric Chappelier et Florian Seydoux, des *presses polytechniques et universitaires romandes* ». Dans mon programme, il s'agit de programmation procédurale (ou modulaire), consistant à partager le code en plusieurs fonctions (ou procédures). Chaque fonction contient un certain nombre d'informations. En bref, le programme correspond à l'assemblage de plusieurs fonctions qui communiquent entre elles.

Voici un plan du programme :



Annexe La Partie Programmation

Ce programme propose de générer la clé publique ainsi que la clé privée. Concernant la génération de la clé publique, le programme prend deux nombres au hasard, qu'il nommera *p* et *q*. Ensuite, *p* et *q* sont entrés dans une fonction qui teste leur primarité, et s'ils ne le sont pas, ils sont modifiés dans une boucle jusqu'à ce qu'ils soient des nombres premiers.

Génération des clés

Ensuite *e* et *n* sont générés pour former la clé publique, puis ils sont enregistrés dans un fichier texte. Ensuite il est temps de générer la clé de déchiffrement *d*. Pour ce faire, j'ai créé une boucle qui assigne à *d* la valeur de *(p-1)*(q-1)* (qui dont être un nombre impair, puisque *d* est premier) et qui décrémente *d* de 2 tant que l'équation **e*d modulo (p-1)*(q-1)==1** n'est pas correcte. Cette technique est de loin efficace, surtout dès qu'on travaille avec de plus grands nombres, le calcul est très long…. Continuons, *d* et *n* forment la clé privée et sont également enregistrés dans un fichier texte. Il ne reste plus que le cryptage.

Cryptage du texte

Afin de crypter les lettres d'un message, j'ai décidé de leur donner leur valeur *ASCII*. Par exemple, la lettre « A » vaut 65, le signe « ! » vaut 33 ou encore le chiffre « 4 » vaut 52. Lors du cryptage, il faut donc faire :
[Valeur ASCII de la lettre à Crypter]^{e} modulo n.

J'ai rencontré beaucoup de problèmes à ce niveau là, car travailler avec des grands chiffres, soit une grande puissance, surcharge rapidement l'ordinateur et prend trop de temps. J'ai essayé plusieurs façons pour programme quelque chose qui fasse ce calcul sans prendre trop de temps mais en vain, j'ai donc cherché sur l'Internet et dans des livres et j'ai fini par trouver quelque chose qui semblait faire ce que je souhaitais. Je précise aussi que ce n'est pas moi qui aie créé ce bout de code, je l'ai simplement adapté et j'ai réfléchi afin de comprendre comment il fonctionne.

Conclusion Personnelle

En écrivant ce dossier, j'ai appris qu'il existait plusieurs types de cryptages, j'ai appris à voir les différences entre tel ou tel cryptage. Je me suis également rendu compte de la place qu'a la cryptographie de nos jours. J'ai aussi appris à travailler avec les classes de restes (modulos) et avec les nombres premiers. Ceci m'a permis d'apprendre à utiliser quelques théorèmes et à travailler avec l'algorithme d'Euclide (pas présent dans le dossier, faute de temps pour faire une démonstration qui tienne la route).

J'ai donc eu du plaisir à écrire ces lignes et à réfléchir comment les écrire, même si quelques-unes unes d'entres-elles m'ont posé quelques petits problèmes. Je pense que ce qui est important c'est de bien comprendre le rôle de la cryptographie et de bien comprendre le procédé de fonctionnement du RSA.

Pourquoi avoir choisi le RSA ? C'est simple, j'en avais déjà entendu parlé et un de mes objectifs principaux était d'améliorer mes connaissances en mathématiques, notamment avec les nombres premiers et les modulos (et un peu de factorisation) mais surtout au niveau programmation en C++, le RSA était donc bienvenu.

La partie programmation en C++ du projet m'a permis de correctement comprendre le fonctionnement de la technique du RSA, même si la sécurité de mon cryptage est quasiment nulle étant donné que le programme crypte caractère par caractère, et donc en quelques sortes, reste un algorithme de substitution simple (dans ce cas il suffit d'utiliser l'analyse de fréquences des caractères dans un texte et d'avoir les bases en cryptanalyse afin de décrypter). Cependant l'efficacité du cryptage n'est pas très importante (en tout cas pour un code de niveau débutant). Ce qui est important c'est de réussir à générer les clés de chiffrement et de déchiffrement ainsi que de réussir à crypter un caractère.

En conclusion, ce travail m'a été très agréable et si c'était à refaire, je le referai.

Vincent S.

Bibliographie

- La cryptographie à clé publique – Principe de fonctionnement ..... http://www.bibmath.net
- Tutoriel sur la cryptographie ..... http://www.uqtr.ca
- Comprendre l'ordinateur – Cryptographie : l'algorithme RSA ..... http://www.sebsauvage.net
- Histoire de la cryptographie - Wikipedia ..... http://fr.wikipedia.org
- Le cryptosystème RSA ..... http://www.apprendre-en-ligne.net
- Arithmétique modulo m ..... http://www.apprendre-en-ligne.net
- La cryptographie et le RSA ..... http://www.cryptosec.lautre.net
- Cryptographie_dossier_college ..... http://mathadora.free.fr
- Donnée d'un TP de Terminale S Spécialité (France) – La cryptographie à clés publique : le système RSA
- Décrypt'Or – 32 Jeux de décodage par substitution simple (aux éditions Hatier – A.F.J)
- Le langage C++ (aux éditions CampusPress)
- C++ par la pratique (des presses polytechniques et universitaires romandes)