Introduction:

La cryptographie est devenue une notion omniprésente en matière de communication car des protections sont nécessaires dès que l'on éprouve le besoin de transmettre des informations confidentielles, que ce soit pour des raisons privées ou militaires comme on l'imagine généralement, mais aussi pour des raisons commerciales. En effet, aujourd'hui, par réseau informatique ou cartes bancaires, la plupart les échanges financiers ont besoin de cette science pour éviter toutes fraudes, ainsi qu'en télécommunication, par le biais des cartes à puce des portables, par exemple.

Mais la cryptographie n'est pas seulement une science contemporaine ; ses origines sont très anciennes : En effet, dans la Grèce antique, on utilisait déjà des méthodes de codage qui ont évolué au fur et à mesure des siècles puis des années, augmentant complexité et sécurité.

Mais la majorité des systèmes présentent une difficulté commune : ils sont ***à clef privée***, c'est à dire qu'ils nécessitent une communication préalable des deux interlocuteurs pour se mettre d'accord sur ***le choix de la clef privée*** qu'ils devront utiliser l'un et l'autre ; pour coder, et, symétriquement, pour décoder. Le problème est donc cette communication préalable qui devra être sûre, car si un pirate entre en possession de cette clef, il pourra décoder tous les messages qu'il verra passer.

D'où l'idée d'utiliser des systèmes de codage ne nécessitant pas de ***clef privée commune*** : l'avantage est que l'on n'aura plus cette difficulté de transmission préalable. Pour cela plusieurs propositions peuvent venir :

_ Tout d'abord, utiliser des systèmes dits ***à clefs publiques.***

_ Ou encore des algorithmes sans clefs, c'est à dire sans clef privée commune.

Système RSA:

L'idée d'un système ***à clef publique*** est de trouver un algorithme difficilement inversible de telle sorte que ***tout le monde puisse coder***, mais qu'il n'y ait ***qu'une personne qui puisse décoder.***

Par exemple Alice communiquera son adresse et la clef à utiliser ; ***la clef publique***. Quand Bob voudra lui envoyer un message, il le codera avec cette clef publique et ainsi plus personne ne pourra lire le message sauf Alice , puisqu'elle est la seule à savoir inverser son algorithme, au moyen de sa ***clef secrète.***

Voyons l'exemple du système « **RSA** » :

A/ Historique :

Le système de cryptographie RSA est considéré comme le code public le plus sûr. Cette méthode a été créée en 1977 par Rivest, Shamir et Adleman, d'où le nom de RSA. Il est utilisé dans le monde entier et sert aujourd'hui à des protections de toutes sortes notamment dans les cartes bancaires.

B/ Théorie :

Etude Préalable de l'expediteur :

On choisit 2 nombres premiers p et q avec lesquels on calcule n=p*q puis ?(n)=(p-1)*(q-1) , ou ? est la fonction d'Euler (c'est à dire le nombre d'entiers premiers avec n et plus petit que n). On cherche alors k tel que k<?(n) et k premier avec ?(n), puis on calcule ? tel que k*? ?1 mod [?(n)]. Les nombres k et n sont publiés : Ce sera ***la clef publique***. Par ailleurs, ? sera ***la clef secrète*** d'Alice.

Codage :

Pour envoyer un message à Alice, Bob le transforme en nombre, puis un ensemble de nombres inférieurs à n, puis pour chacun de ces nombres m il calcule $c=m^k$ mod [n] et il envoie les nombres c à Alice.

Décodage :

Pour décoder le message, Alice doit calculer $c^?$ mod [n] et, grâce au petit théorème de Fermat, on montre qu'elle va retrouver m et, en le recomposant, le message initial.

C/ Echange de clef :

On peut utiliser cette méthode pour éviter le problème de la transmission préalable et se communiquer des clefs afin d'utiliser un ***codage symétrique***, c'est à dire un système ***à clef privée***.

Soit n et k les clefs publiques d'Alice. Elle va choisir un nombre aléatoire x inférieur à n et calculer $X= k^x$ mod [n] qu'elle enverra à Bob. De son coté, celui ci choisira son aléa y et élèvera X à la puissance y pour obtenir C. Par ailleurs il enverra $Y= k^y$ mod [n] à Alice qui, en élevant ce nombre à la puissance x trouvera également C.

Nos deux interlocuteurs se seront mis d'accord sur une clef C sans donner le moindre renseignement à un éventuel pirate. En effet celui ci, placé entre les deux, n'aura pu avoir que n et k,qui sont publics, et X et Y, ce qui ne permet pas de trouver C ni de retrouver x ou y. Donc Alice et Bob pourront maintenant communiquer à l'aide d'un ***algorithme à clefs privée*** en utilisant la clef C.

D/ Application :

Sur Maple…

Effectuons une application numérique, avec des nombres de taille modeste : on choisit k et 2 nombres premiers p et q. On a les valeurs indiquées (p=389, q=167 et k=17). Puis on calcule n et ?(n) (n=64 963 et ?(n)=64 408) et on résout une petite équation pour trouver un intermédiaire a. On trouve a=7 ce qui nous donne ? (? =26 521).

On utilise des procédures de passage lettre/nombre classiques : Ce sont de simples changements de base ; à chaque caractère on associe un nombre. Ici on travaille avec 114 caractères. On a entré des procédures de codage de type RSA dont on pourra détailler le fonctionnement plus tard, éventuellement.

On a effectué ici une application. Et on peut remarquer que cela fonctionne également dans l'autre sens, c'est à dire que dans le cadre d'***une signature*** : Alice peut prouver qu'elle est l'auteur d'un message en diffusant son message après l'avoir codé avec ses clefs, ? et n. Ainsi tout le monde pourra décoder en utilisant les clefs publiques, k et n, et lire le message en étant sur que Alice en est l'auteur, puisqu'elle est la seule à connaître ? .

Mais cette méthode de calcul prend du temps à cause de l'élévation à des puissances élevées. Pour des applications plus sérieuses, on sera amené à introduire une procédure d'exponantiation modulaire bien plus rapide.

E/ Exemple de piratage :

Cassage pratique, sur Maple :

Pour de petits nombres, il est très facile de pirater un code RSA ... Il suffit de factoriser le nombre n comme ici (n=64 963) ; on trouve p et q instantanément et on en déduit ?, puis a, et enfin ? et on peut finalement pirater.

Mais la procédure de factorisation de Maple ne fonctionne aisément que pour des nombres inférieurs à 10^35. Par contre, il est relativement facile de trouver de grands nombres premiers, même sous Maple.

La fiabilité du RSA repose sur l'écart entre la ***difficulté de factorisation*** et la ***facilité à trouver de grands nombres premiers***.

A propos de cassage, on peut citer le cassage du RSA-129 en 1994, ou encore des récents travaux de Serge Humpich.

Conclusion:

La cryptographie ***à clefs publiques*** présente donc des avantages incontestables par rapport aux méthodes ***à clefs privées***. Mais il n'existe que peu de méthodes couramment utilisées ; on a vu celle du RSA, la plus connue, et on sait les difficultés auxquelles elle est confrontée avec les progrès informatiques. Mais on peut considerer que, malgré quelques faiblesses, il semble toujours très fiable, du moins pour de grands nombres.

Mais il existe d'autres méthodes ***à clefs publiques*** purement théoriques ou encore des méthodes ***sans clefs privée commune*** qui ne connaissent pas d'application pratique ... alors est-ce impossible ou est-ce que certaines idées n'ont tout simplement pas été approfondies ?