Cryptologie

De tout temps les individus ont souhaité se transmettre des informations de manière secrète pour que d'autres personnes n'en n'aient pas connaissance. Ils ont employé des moyens plus ou moins sophistiqués pour le faire. Par ailleurs, certains individus ont tout intérêt à connaître le contenu d'une information transmise entre deux personnes et utilisent différentes méthodes pour arriver à leur fins.

Ainsi, dans l'antiquité, les grecs rasaient la tête d'un messager, écrivaient un message sur son crâne et attendaient que les cheveux aient repoussés avant de l'envoyer porter ce message à son destinataire.

1. Cryptologie et cryptanalyse

La **cryptologie** regroupe l'ensemble des techniques qui permettent, d'une part de dissimuler le contenu d'un message en le codant afin que seul le destinataire soit capable d'en prendre connaissance en le déchiffrant, d'autre part de décrypter un message destiné à une autre personne. La cryptologie se scinde deux sous domaines : la **cryptographie** qui étudie et conçoit des procédés de chiffrement aussi inviolables que possible et la **cryptanalyse** qui a pour but de déchiffrer des informations dissimulées dans un message crypté. La cryptographie représente le côté défensif de la cryptologie (on protège une information) et la cryptanalyse le côté offensif de celle-ci puisque l'on essaie de décrypter un message chiffré pour l'avoir en clair. Ces deux aspects sont néanmoins très liés puisque pour tester si un message codé supposé inviolable il faut essayer de le déchiffrer, donc faire de la cryptanalyse, sans en avoir la clef. D'autre part les possibilités en cryptanalyse à un moment donné conduisent la cryptographie à imaginer de nouvelles techniques. Toutefois, le destinataire légal d'un message codé doit être capable de le déchiffrer rapidement.

Pour résumer, la cryptographie permet de transformer un message clair en message inintelligible (texte crypté) et la cryptanalyse permet de retrouver le texte clair à partir du texte crypté sans en posséder la clef. Deux voies sont à considérer lorsque l'on considère un message chiffré. Le déchiffrement qui permet au destinataire d'un message codé de retrouver le message en clair en utilisant la clef de déchiffrement et le décryptement où un ennemi qui ne connaît pas la clef essaye de décrypter le message.

Longtemps réservée aux militaires et diplomates, la cryptographie a été un monopole d'état. En France, les moyens de cryptographie ont été considérés comme arme de guerre et interdits jusqu'en 1998. La législation s'est ensuite assouplie et permettait de chiffrer, sans déclaration, des messages avec des clefs de 128 bits au maximum. Depuis 2004, la loi du 21 juin pour la confiance dans l'économie numérique a autorisé l'utilisation de moyens de cryptographie. Le but est de garantir la sécurité du stockage des données et de leur transmission en assurant leur confidentialité, leur authentification et leur intégrité. Néanmoins, pour l'importation ou l'exportation, leur utilisation est soumise à déclaration ou autorisation. La raison de ce changement est que sans cryptographie toute économie numérique serait impossible. Ainsi, il ne serait pas possible d'acheter de façon sûre un objet ou un service sur internet si les données bancaires ou de carte bleue étaient transmises en clair sur le réseau.

2. Principaux problèmes posés

Les méthodes de cryptographie interviennent dans de nombreux domaines lors d'échange d'information entre deux interlocuteurs qui peuvent être des personnes physiques, morales ou des dispositifs électroniques. Ceux-ci sont indiqués dans la figure 1.

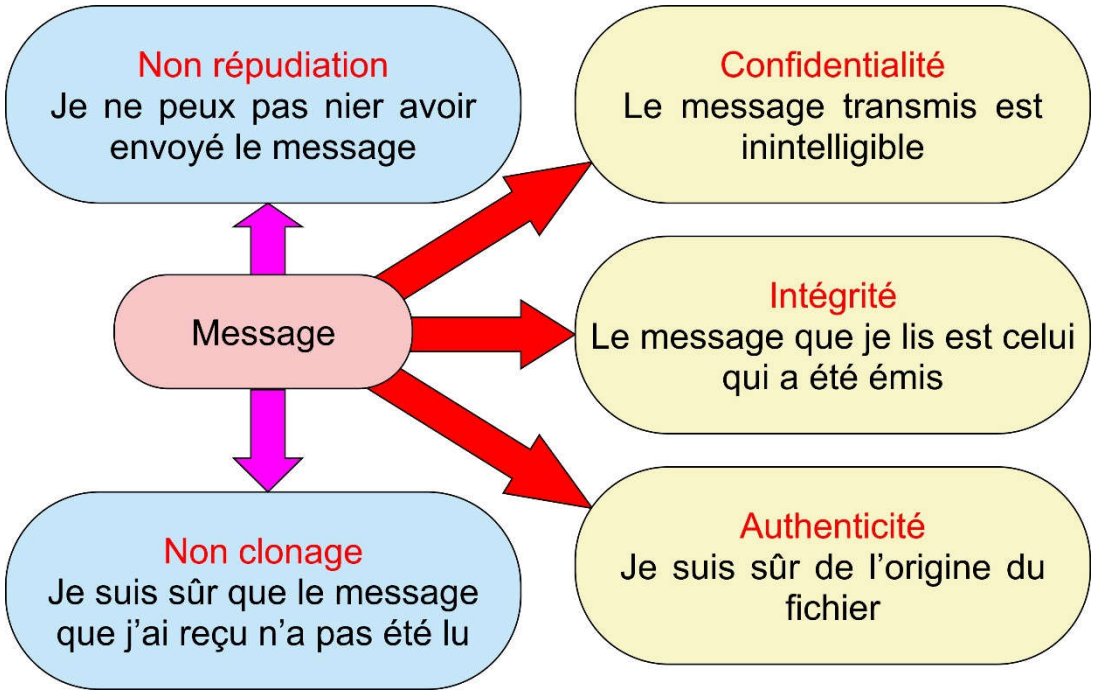

Figure 1.

Le contenu d'un message ne doit pas être lu par tout le monde (confidentialité). Le message ne doit pas avoir été altéré lors de sa transmission (intégrité) et que on doit être sûr de son origine (authenticité). Lorsque l'on reçoit un message, même s'il est identique à celui qui a été émis, il faut être sûr qu'il n'a pas été intercepté et lu (non clonage). Enfin, lorsque c'est moi qui envoie un message, il ne faut pas que plus tard je nie l'avoir envoyé (non répudiation).

Pour répondre à ces conditions, on utilise des méthodes mathématiques similaires à celles employées en cryptographie.

3. Les principes de Kerckhoffs

Pendant longtemps on a, en cryptographie, caché l'algorithme utilisé. C'est par exemple le cas du chiffre de César que l'on peut facilement décoder si l'on connait la méthode utilisée. C'est aussi le cas des grecs qui rasaient le crâne du messager, inscrivaient le message sur celui-ci et laissaient repousser les cheveux avant de l'envoyer porter le message. Pour ces deux exemples, si l'on connaît la méthode employée, il est facile de déchiffrer le message.

Si on ne le connaît pas l'algorithme propriétaire utilisé, il fort possible qu'on pourra le décrypter car il y aura très certainement des failles qui ont auront été laissées par les concepteurs. L'avantage mettre l'algorithme dans le domaine public est de permettre à de nombreuses personnes d'essayer de le déchiffrer ce qui permet de voir si cela est possible et donc de l'améliorer si c'est le cas, sauf bien sûr si celui qui a cassé le code ne

le dit pas. C'est ce qui s'est produit avec la machine Enigma de l'armée allemande dont les messages codés par celle-ci ont été cassé par les britanniques, notamment grâce à Alan Turing.

Il est donc préférable qu'un algorithme soit public plutôt que propriétaire car cela permet d'avoir une meilleure sécurité. Par exemple, les algorithmes de chiffrement propriétaires comme celui du GSM (chiffrement A5/0 et A5/1) ou celui de protection contre la copie des des DVD (CSS Content Scrambling System) ont été cassés en quelques semaines après leur mise en service.

En 1883, AugusteKerckhoffs a publié un article qui a jeté les bases de la cryptographie moderne lorsque celle-ci est utilisée à grande échelle (armée, diplomates, etc.). Il proposait 6 principes que devait satisfaire un système cryptographique. On peut les résumer comme suit :

1. Le système doit être matériellement ou mathématiquement indéchiffrable
2. Il ne doit pas exiger de secret et peut tomber entre les mains de l'ennemi
3. La clef doit pouvoir être retenue facilement
4. Il faut qu'il soit applicable à la correspondance télégraphique
5. Il doit être portatif et ne pas nécessiter plusieurs personnes pour le mettre en œuvre
6. Le système doit être facile à utiliser

Ces principes sont dans leur ensemble toujours valable aujourd'hui même s'ils doivent être adapté aux technologies actuelles.

L'ordinateur représente une rupture importante pour la cryptographie et les principes de Kerckhoffs peuvent aujourd'hui se traduire par les trois principes suivants :

1. La sécurité du système cryptographique doit reposer sur le secret de la clef et non sur celui de l'algorithme.
2. Le déchiffrement sans la clef doit être impossible avec les moyens du moment car nécessitant des temps astronomiques
3. Si l'on connaît le message en clair et le message crypté, il ne doit pas être possible d'en extraire la clef en un temps raisonnable.

4. Congruence

La congruence d'entiers est une notion largement utilisée en cryptographie. Il s'agit d'une relation qui relie deux nombres entiers.

Considérons deux nombres entiers relatifs a, b et un nombre entier $n \geq 2$. On dit que a est **congru** à b **modulo** n, si n divise la différence $b - a$. On note cette propriété

$$a \equiv b \pmod{n}$$

Si par exemple $n = 26$ (le nombre de lettres de l'alphabet), on a $31 \equiv 5 \pmod{26}$ car 31-5=26 est divisible par 26.

Si l'on prend le nombre 157, on a $157 \equiv 1 \pmod{26}$. On a aussi $-2 \equiv 24$ car $24+2$ est divisible par 26.

Dans le premier cas on pouvait écrire $31 = 26 + 5$, dans le second $157 = 6 \times 26 + 1$ et dans le troisième $-2 = 24 - 26.$.

L'ensemble des entiers relatifs est $\mathbb{Z}$. Celui des de tous les éléments de $\mathbb{Z}$ modulo 26 est noté $\mathbb{Z}/26\mathbb{Z}$. Il contient 26 éléments $\{0,1,2,\ldots 25\}$.

Dans le cas général, $\mathbb{Z}/n\mathbb{Z}$ contient n éléments. On représente ainsi chaque nombre entier relatif par un nombre $\in \{0,1,2,\ldots n-1\}$

Tout nombre $a \in \mathbb{Z}$ est donc représenté par un nombre $r \in \{0,1,2,\ldots n-1\}$ qui est le reste de la division euclidienne de a par n c'est-à-dire : $a = dn + r$. On a donc $a \equiv r \pmod{n}$ avec $0 \leq r < n$.

Par exemple : 31^{27}=184827135828240353588176587528159237917 11≡21 mod 26

On peut définir les opérations d'addition et de multiplication dans $\mathbb{Z}/n\mathbb{Z}$. Voici quelques exemples dans $\mathbb{Z}/26\mathbb{Z}$.

Addition

$19+17=36\equiv 10 \pmod{26}$.

Pour l'addition : $75+27=102\equiv 24 \pmod{26}$. On peut aussi calculer le résultat d'une autre manière. On a $75\equiv 23 \pmod{26}$ et $27\equiv 1 \pmod{26}$. Donc le résultat de l'addition peut aussi s'écrire $75 + 27 \equiv 23 + 1 = 24$.

Multiplication

Par exemple : $27 \times 31 = 837 = 32 \times 26 + 5 \equiv 5$

On aurait pu procéder autrement :

$27 \equiv 1$ et $31 \equiv 5$. Donc $27 \times 31 \equiv 1 \times 5 = 5$.

Car :

Si $a \equiv b \bmod n$ et $c \equiv d \bmod n$
Alors $ac \equiv bd \bmod n$

Il y a donc compatibilité entre les opérations d'addition et de multiplication des entiers et celles des entiers modulaires.

Soustraction

$1223-773=450\equiv 8 \bmod 26$
Ou $1223\equiv 1 \bmod 26$ et $773\equiv 19 \bmod 26$
$1223-773\equiv 1-19=-18\equiv 8 \bmod 26$

Division

$\frac{151}{27}=\frac{26\times 5+21}{26+1}$. Donc $\frac{151}{27}\equiv 21 \bmod 26$
Ou $151\equiv 21 \bmod 26$ et $27\equiv 1 \bmod 26$
Donc $\frac{151}{27}\equiv\frac{21}{1}=21 \bmod 26$

Les cas suivants sont plus difficiles.

Par exemple, $\frac{1}{2}\equiv 4 \bmod 7$. En effet, il faut le comprendre comme trouver le nombre a tel que $2a = 1 \bmod 7$ Dans ce cas on trouve $a = 4$.

Mais le plus simple est d'utiliser la table de multiplication modulo 7. Elle est indiquée dans le tableau 1.

×	0	1	2	3	4	5	6
0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6
2	0	2	4	6	1	3	5
3	0	3	6	2	5	1	4
4	0	4	1	5	2	6	3
5	0	5	3	1	6	4	2
6	0	6	5	4	3	2	1

Tableau 1

En utilisant cette table, on voit que $\frac{1}{4} \equiv 2 \bmod 7$.

De même $\frac{2}{3} \equiv 3 \bmod 7$. Pour trouver cela il faut chercher le nombre entier a tel que $3a \equiv 2 \bmod 7$. C'est le nombre entier compris entre 0 et 6 qui divise exactement l'un des nombres suivants : 9,16, 23, etc. Le premier qui satisfait à cette demande et $a = 3$. On a bien $9 \equiv 2 \bmod 7$. On peut également directement trouver le résultat grâce au tableau 24.

On peut également montrer que $\frac{1}{6} \equiv 6 \bmod 7$. Il faut chercher a tel que $6a \equiv 1 \bmod 7$. Pour cela on cherche le nombre entier qui divise exactement par 6 les nombres suivants : 8, 15, 22, 29, 36, etc. On voit que le nombre 26 fait l'affaire et $a = \frac{36}{6} = 6$. On peut aussi utiliser la table de multiplication du tableau 1.

Il faut noter que l'inverse n'existe pas toujours. Ceci est illustré par la table de multiplication modulo 6 du tableau 2.

×	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Tableau 2

On voit alors sur la table de multiplication que :

2^{-1} ; 3^{-1} et 4^{-1} n'ont pas d'inverse $(\bmod 6)$ et que $5^{-1} \equiv 5 \ (\bmod 6)$

Puissance

On peut montrer par récurrence que :

Si $a \equiv b$ alors $a^p \equiv b^p$ pour tout p qui est un entier naturel non nul.

Si l'on reprend l'exemple donné ci-dessus (31^{27}), on a $31 \equiv 5 \bmod 26$. Il suffit donc de calculer $5^{27}=7450580596923828125 \equiv 21 \bmod 26$. Une autre manière de faire si l'on ne veut pas traiter des nombres comportant un grand nombre de chiffres est de procéder comme ci-dessous :

$$31^{27} \equiv 5^{27} \equiv \left(5^3\right)^9 \equiv 21^9 \equiv \left(21^3\right)^3 \equiv 5^3 \equiv 21 \bmod 26$$

Si l'on veut calculer $53^{143} \bmod 26$, ce qui correspond au nombre suivant :

372777209417618919610982765084513544028293043066884835589830684971989387968343583202647562476655855402077155275015288270533854013755250955387400134434823244360647248781473270827505487632952247567048674435870136986012161366333944023450635049068847 7

la tâche est plutôt difficile. Mais si on remarque que $53 \equiv 1 \bmod 26$, alors il suffit de calculer $1^{143}=1 \equiv 1 \bmod 26$.

5. Le chiffre de César

On dit que César (Cæsar) a utilisé un protocole de chiffrage pour transmettre des messages à ses armées. Le chiffrement de César est un décalage de 3 lettres comme il est indiqué dans la figure 2. Ainsi le A devient D, le B devient E, etc.

Dans la figure 2, on donne un exemple de chiffrement de César avec la phrase « LE CHIFFRE DE CESAR EST TRES FACILE A DECODER ». La phrase cryptée est facilement décryptée si l'on connait la valeur du décalage, ici 3 dans le cas du chiffre de César.

Figure 2

On trouve sur internet plusieurs sites permettant de crypter et décrypter des messages en utilisant le principe du chiffrement de César. Il suffit de taper « codage de césar » dans Google.

Le décalage peut être différent de 3. On peut choisir toutes les valeurs comprises entre 0 et 25. La valeur 0 n'a pas grand intérêt puisque le message crypté est identique au message original. On peut également ajouter des symboles à l'alphabet, comme l'espace, le point, la virgule ou le point d'interrogation. Il est alors commode d'avoir une

représentation circulaire de l'alphabet comme le montre la figure 3 car on peut réaliser deux roues concentriques pouvant tourner indépendamment. La roue extérieure contient l'alphabet et la roue intérieure le chiffrement correspondant. Dans l'exemple de la figure 3, il y a un décalage de 3 symboles entre les 2 roues. A donne D et Z la virgule, par exemple. En faisant tourner une des roues par rapport à l'autre, on peut générer l'ensemble des décalages possibles.

Le chiffrement de César et ceux qui en dérivent sont des chiffrement mono-alphabétiques, c'est-à-dire qu'à une lettre correspond toujours la même lettre (ou symbole) chiffrée.

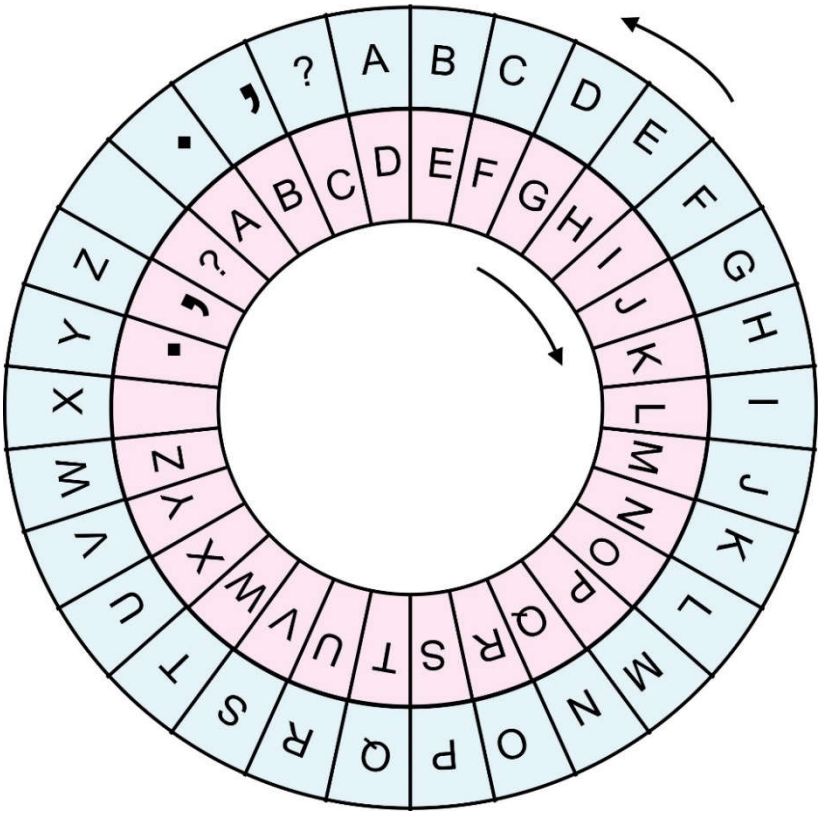

Figure 3

Il est plus commode d'associer à chaque lettre un nombre comme il est indiqué dans le bas de la figure 2 car on peut utiliser un ordinateur pour calculer le message chiffré.

Le chiffrement de type César avec un décalage k (le chiffrement de César correspond à $k = 3$) est une application, C_k, de $\mathbb{Z}/26\mathbb{Z}$ dans $\mathbb{Z}/26\mathbb{Z}$ qui fait correspondre à $x \mapsto x + k$. Cette opération, faite dans $\mathbb{Z}/26\mathbb{Z}$, revient à calculer $x+k \pmod{26}$. Pour $k = 3$ on a $C_3(0) = 3$ et $C_3(24) = 1$.

La fonction de déchiffrement D_k vaut simplement : $D_k = x - k$.

On a $\forall x,\ D_k(C_k(x)) = x.$ La fonction D_k est la fonction réciproque (ou inverse) de C_k.

Pour se transmettre un message, les deux interlocuteurs doivent connaître la clef k.

Il existe 26 valeurs possibles de k mais seulement 25 sont intéressantes. La sécurité d'un tel chiffrement est très faible. Avec un ordinateur, il suffit par exemple d'essayer les 25 clefs possibles et de voir si le message décrypté avec cette clef est lisible.

6. Chiffrement par substitution

Le chiffrement de César est très facile à casser car le nombre de clefs est faible (26, dont une triviale) d'où l'idée d'augmenter le nombre de clefs. Pour cela, à chaque lettre on fait correspondre une autre lettre au hasard. C'est-à-dire que l'on se donne une permutation aléatoire des lettres de l'alphabet comme le montre la figure 4 sur un exemple. Dans celui-ci, A donne J et B donne G, par exemple. Avec cette correspondance on peut chiffrer un message et en utilisant à nouveau cette correspondance en sens inverse, on peut le décrypter. L'intérêt d'utiliser une permutation aléatoire est d'avoir un espace immense de clefs. Il n'est donc plus possible de tester toutes les possibilités.

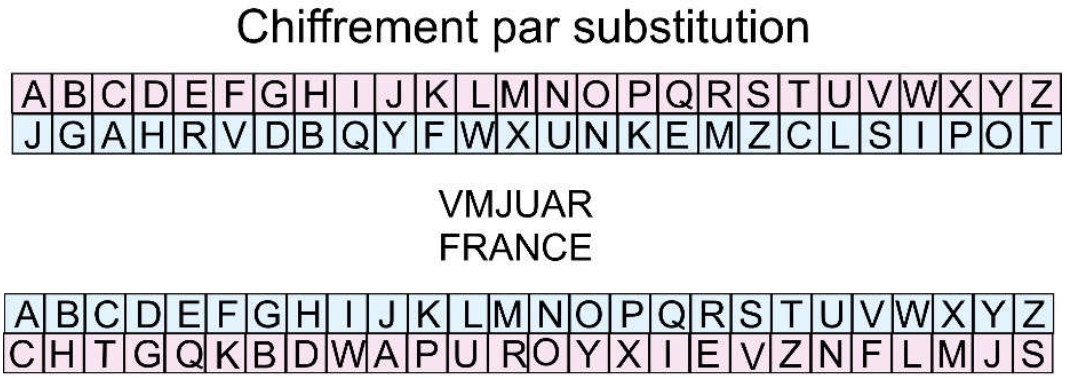

Figure 4

En effet, la clef est longue, ici les 26 lettres de l'alphabet. Il y a $26!$ permutations possibles de 26 lettres soit :

$$403291461126605635584000000 \simeq 4\times10^{26}.$$

Avec un ordinateur rapide pouvant tester 1 million de clefs par seconde, il faudrait environ 12 700 milliards d'années pour tester toutes les clefs (le soleil aura épuisé son combustible dans environ 5 milliards d'années). Il faudrait ensuite choisir, dans cet ensemble énorme, celle qui donne un texte compréhensible. Une attaque directe du message codé est donc sans espoir. Il faut donc procéder de manière indirecte. L'une d'entre elle est de connaître la clef en interceptant son envoi de l'émetteur vers le destinataire. L'autre est d'utiliser le fait que toutes les lettres n'ont pas la même probabilité d'apparaître dans le message. Pour cela il faut pouvoir disposer d'un message assez long.

Ce chiffrement est mono-alphabétique. À une lettre on associe toujours la même lettre dans le message chiffré. Or leur fréquence d'apparition n'est pas la même. Elle dépend de la langue utilisée. Toutefois, pour une langue donnée, et pour la plupart des textes, la probabilité d'apparition d'une lettre est à peu près constante. En français, les lettres apparaissent, par probabilité décroissante de la manière selon l'ordre suivant :

E A S I T N R U L O D C M P V F H G B Q X J Y Z K W

La lettre E est celle qui apparaît le plus souvent et le W celle qui est la moins fréquente. Si l'on tient compte de l'espace qui sépare deux mots et qui apparaît très fréquemment dans le texte, on a la distribution de probabilité (exprimée en pourcentage) donnée

dans le tableau 3 issu des données de l'ouvrage de P.Guillot, la Cryptologie (l'alphabet comprend 27 caractères).

Lorsque la probabilité est 0,0 cela signifie que le pourcentage est inférieur à 0,05%. Ces pourcentages peuvent être différents selon les textes pris en compte. Certains auteurs littéraires ont ainsi écrit des livres sans utiliser la lettre E.

Lettre	Espace	A	B	C	D	E	F	G	H
%	18,4	7,5	0,8	2,5	3,1	14,0	0,9	0,9	0,9
Lettre	I	J	K	L	M	N	O	P	Q
%	6,1	0,3	0,0	5,0	2,2	5,7	4,0	1,9	0,8
Lettre	R	S	T	U	V	W	X	Y	Z
%	5,4	6,4	6,0	5,2	1,3	0,0	0,3	0,2	0,1

Tableau 3

Dans cryptage par substitution deux lettres différentes sont codées par 2 lettres différentes mais une même lettre est toujours codée de la même façon. On parle pour cela de substitution mono-alphabétique. C'est un des inconvénients de ce type de chiffrage qui peut être déchiffré en utilisant les fréquences d'apparition d'une lettre ou de deux lettres consécutives. Cette technique de déchiffrement a été proposé pour la première fois par al-Kindi, au neuvième siècle. Il faut toutefois que le message crypté soit assez long. La clef est difficile à retenir aussi peut-elle être interceptée car le décrypteur devra la stocker quelque part pour pouvoir s'en servir.

Dans ce type de cryptage on cache souvent la longueur des mots car cela facilite le décryptage. Dans la pratique, on supprime donc les espaces entre les mots du texte crypté et on présente souvent le message sous forme de blocs de 5 lettres.

7. Chiffrement de Vigenère

Ce chiffrement a pour but d'éviter qu'une lettre non codée soit toujours représentée par la même lettre codée. Dans le chiffrement de Vigenère, on ne considère plus les lettres individuellement mais on les regroupe en blocs de lettres. Par exemple, si le message initial est :

IL EST DIFFICILE DE DECRYPTER UN MESSAGE

On va le réécrire en groupant les lettres, par exemple par 5, en oubliant les espaces. Cela donne :

ILEST DIFFI CILED EDECR YPTER UNMES SAGE

Les nouveaux espaces séparent les blocs et non plus les mots.

On choisit ensuite une clef constituée de 5 nombres entiers compris entre 0 et 25 dont la longueur est égale à celle du bloc, ici 5 pour l'exemple qui nous intéresse. Prenons par exemple la clef (CLEFS, soit 2,11,4,5,18). Le chiffrement de Vigenère consiste à appliquer à chaque lettre du bloc un chiffrement de César avec un décalage correspondant. On a donc un chiffrement poly-alphabétique car une lettre non codée ne donne pas toujours la même lettre codée.

Dans l'exemple choisi, la première lettre du bloc sera codée avec un décalage de 2, la deuxième avec un décalage de 11, etc. On procède de la même manière pour tous les blocs. Le message codé selon Vigenère sera donc le suivant :

KWIXL FTJKA ETPJV GOIHJ AAXJJ WYQJK ULKJ (message codé)

ILEST DIFFI CILED EDECR YPTER UNMES SAGE (message initial)

On voit que le E est par exemple codé par I dans le premier bloc, par J dans le troisième, par G au début du quatrième. De même, la lettre K du message codé peut correspondre à I (premier bloc) ou à A (second bloc).

Avec les ordinateurs, on traduit d'abord le message en chiffres, on code avec la clef en chiffres et l'on retraduit en lettres. Si, en chiffres, un bloc de longueur k correspond à $\{x_1, x_2, \dots x_k\}$ et la clef à $\{n_1, n_2, \dots n_k\}$, la fonction de chiffrement est :

$$\{x_1, x_2, \dots x_k\} \mapsto \{x_1 + n_1, x_2 + n_2, \dots, x_k + n_k\} = \{y_1, y_2, \dots y_k\}$$

La fonction de déchiffrement sera alors :

$$\{y_1, y_2, \dots y_k\} \mapsto \{y_1 - n_1, y_2 - n_2, \dots, y_k - n_k\} = \{x_1, x_2, \dots x_k\}$$

Pour un bloc de longueur k il y a 26^k clefs possibles, soit 11 881 376 pour une clef de 5 lettres.

Bien que beaucoup plus robuste au décryptage que le chiffrement par substitution, il peut y avoir néanmoins des séquences où la même lettre apparaît codée de la même manière. Par exemple le codage de « encore un prêt » donnera

ENCORE UN PRET (phrase initiale)

ENCOR EUNPR ET (phrase initiale découpée en blocs)

GYGTJ GFRUJ GE (phrase codée)

On voit, sur cet exemple, que le E apparaît au début de chacun des blocs initiaux et va donc donner la même lettre codée. Ceci peut constituer une faille pour décrypter.

Le code du Vigenère a été mis en œuvre par Blaise de Vigenère au XVI^{è} siècle. Il a été cassé par le mathématicien Charles Babbage au XIX^{è} siècle avec une méthode qui n'a été retrouvée que plus tard par Kasiski.

8. Le chiffrement de Vernam

Comme le chiffrement de Vigenère peut être décrypté par la méthode de Babbage, G.Vernam proposa d'utiliser une clef aussi longue que les messages. Cette clef doit être aléatoire et jeté après chaque usage. En effet, si l'on utilisait deux fois la même clef, en mettant les deux messages bout à bout, on aurait affaire à une clef plus petite que l'on pourrait décrypter. G.Vernam a proposé son chiffrement en 1917 et cette méthode a été breveté par les Bell Labs en 1919.

Shannon a démontré par la suite que, si la clef est générée de façon aléatoire et s'il elle a la longueur du message, le chiffrement est inviolable à condition qu'on n'utilise qu'une fois la clef.

Ainsi, pour avoir un chiffrement parfait il faut que la clef ait la même longueur que celle du message et qu'elle soit aléatoire. Illustrons cela sur un exemple. On veut coder AU SECOURS avec la clef ETWIPJKDH. Le tableau 2 donne quelques étapes.

1. On supprime les espaces du message
2. On traduit ce message alphabétique en chiffre (A→0, B→1, etc.)
3. On a traduit la clef alphanumérique
4. Dans chaque colonne on fait la somme modulo 26 de la valeur de la lettre non codée et de la clef.
5. On traduit le résultat codé en digital en alphanumérique.

Message initial	A	U	S	E	C	O	U	R	S
Message (décimal)	0	20	18	4	2	14	20	17	18
Clef (décimal)	4	19	22	8	4	9	10	3	7
Clef (lettres)	E	T	W	I	P	J	K	D	H
Message chiffré (décimal)	4	13	14	12	17	23	4	20	15
Message chiffré (lettres)	E	N	O	M	R	X	E	U	Z

Tableau 4

L'inconvénient de cette méthode est que la clef est aussi longue que le message et que celle-ci doit être transmise de l'émetteur au destinataire. Comme elle doit être changée pour chaque message, cela devient vite très lourd à gérer. En général émetteur et destinataire se mettent d'accord sur une longue liste de clefs secrètes qu'ils se sont échangés auparavant.

9. La machine Enigma

C'est une machine électromécanique utilisée par l'armée Allemande lors de la deuxième guerre mondiale. Elle servait à la fois au chiffrement et au déchiffrement. Elle est constituée de plusieurs anneaux (rotors) qui permettent de faire chacun un chiffrement de type Vigenère. On les utilise comme une machine à écrire pour chiffrer et déchiffrer.

Les polonais d'abord puis les britanniques, notamment grâce à Alan Turing, ont pu déchiffrer les messages de l'armée allemande cryptés par la machine Enigma. Cela a permis de sauver de nombreuses vies chez les alliés.

À la fin de la guerre, les britanniques ont été assez habiles pour faire croire que la machine Enigma produisait des messages indéchiffrables ce qui leur a permis de la vendre à des gouvernements et des industriels. Ils pouvaient ainsi, si c'était nécessaire, avoir connaissance des messages cryptés échangés.

10. Le chiffre de Playfair

Le chiffrement de Lyon Playfair (1854), utilisé notamment par les britanniques lors de la guerre des Boers, a en fait été conçu par Charles Wheastone, son ami. On utilise un carré de 5×5=25 cases pour les 26 lettres de l'alphabet (le w étant remplacé par vv en français). Il s'agit d'un chiffrement de substitution polygrammique. Le tableau donne un exemple de clef.

c	g	p	b	l
q	u	h	r	x
f	m	a	o	k
t	j	n	v	e
z	d	y	i	s

On découpe le message en bigrammes avec un ajout éventuel de « x » si le nombre de lettres est impair. On applique alors les règles suivantes :

- Si 2 lettres sont sur les sommets d'un rectangle on chiffre par les sommets opposés
- Si 2 lettres sont sur la même ligne on prend les 2 lettres situées à leur droite
- Si 2 lettres sont sur la même colonne on prend les 2 lettres situées au-dessous d'elles.

11. Chiffrement à clef jetable (One-Time-Pad)

Les ordinateurs traitent des nombres binaires. La seule méthode de chiffrement qui soit prouvée mathématiquement comme inviolable est celle qui utilise une clef jetable aléatoire unique de même longueur que le message (**chiffrement par clef jetable** ou **one-time-pad** en anglais).

Comme les ordinateurs traitent des informations binaires, c'est-à-dire une suite de 0 et de 1, nous allons expliciter ce type de chiffrement sur un exemple simple dans le tableau 5.

M	0	1	1	0	0	1	0	1	0	0	1	1	1	0	1	0
K	1	0	0	1	1	0	0	1	1	0	1	0	1	0	0	1
C=M⊕K	1	1	1	1	1	1	0	0	1	0	0	1	0	0	1	1
K	1	0	0	1	1	0	0	1	1	0	1	0	1	0	0	1
M=C⊕K	0	1	1	0	0	1	0	1	0	0	1	1	1	0	1	0

Tableau 5

On part d'un message M de 16 bits. On va utiliser une clef K de 16 bit générée de manière aléatoire. Le fait qu'elle soit aléatoire est un point important. On fait ensuite la somme XOR (⊕) bit à bit. Cela donne le message crypté C=M⊕K. Le destinataire

décryptera le message reçu en appliquant à nouveau la clef K. Il obtiendra le message envoyé car M=C⊕K.

L'utilisation d'une clef aléatoire a pour but de transformer le message M en un message aléatoire qu'on ne pourra décrypter sans la clef.

12. Le Chiffre de Hill

Lester S. Hill a publié, en 1929, un chiffre polygraphique analogue à celui de Playfair dans lequel on ne déchiffre pas les lettres une par une mais par paquets. On travaille modulo 26.

Dans le cas général on code un bloc de n lettres, $\ell_1,\ldots \ell_n$ (sous forme d'une une matrice colonne $\mathbf{A}$) à l'aide d'une matrice $\mathbf{M}$ de dimension $n\times n$ convenablement choisie (les éléments de matrice doivent être des entiers inférieurs à 26 et elle doit avoir un inverse modulo 26) en effectuant la multiplication $\mathbf{M}.\mathbf{A}$ modulo 26. On obtient ainsi le bloc codé $L_1,\ldots L_n$, c'est-à-dire un vecteur colonne $\mathbf{C}$:

$$\mathbf{A}=\begin{bmatrix}\ell_1\\ \ldots\\ \ell_n\end{bmatrix}\ ;\ \mathbf{M}=\begin{bmatrix}M_{11} & \ldots & M_{1n}\\ & & \\ M_{n1} & & M_{nn}\end{bmatrix}\ ;\ \mathbf{B}=\begin{bmatrix}L_1\\ \ldots\\ L_n\end{bmatrix}$$

$$\mathbf{B}=\mathbf{M}.\mathbf{A}$$

Nous allons illustrer cette méthode dans le cas le plus simple où l'on regroupe les lettres par paquets de deux. Cette méthode est utilisation de cette méthode cryptographique, les blocs utilisés sont bien sûr plus importants et les calculs plus compliqués puisque si l'on considère des blocs de taille n il faut utiliser des matrices $n\times n$.

Prenons pour matrice de codage :

$$\mathbf{M}=\begin{bmatrix}3 & 2\\ 1 & 3\end{bmatrix}$$

L'inverse d'une matrice 2×2 est donnée par la formule suivante :

$$\mathbf{M}=\begin{bmatrix}a & b\\ c & d\end{bmatrix}\ \Rightarrow\ \mathbf{M}^{-1}=\frac{1}{\Delta}\begin{bmatrix}d & -b\\ -c & a\end{bmatrix}\ \text{avec}\ \Delta=ab-bc$$

Pour la matrice choisie, on a :

$$\mathbf{M}^{-1}=\frac{1}{7}\begin{bmatrix}3 & -2\\ -1 & 3\end{bmatrix}$$

Il faut maintenant calculer $\mathbf{M}^{-1}(\text{mod}\,26)$. On a $\frac{1}{7}\equiv 15\ (\text{mod}\,26)$. Donc :

$$\mathbf{M}^{-1}=\frac{1}{7}\begin{bmatrix}3 & -2\\ -1 & 3\end{bmatrix}\equiv 15\begin{bmatrix}3 & -2\\ -1 & 3\end{bmatrix}(\text{mod}\,26)\equiv\begin{bmatrix}45 & -30\\ -15 & 45\end{bmatrix}(\text{mod}\,26)\equiv\begin{bmatrix}19 & 22\\ 11 & 19\end{bmatrix}(\text{mod}\,26)$$

Supposons que le message que l'on envoie soit « SECRET ». On suppose que les lettres de l'alphabet sont numérotées en partant de 0 (A=0, B=1, etc.). On va découper le

message en groupe de 2 lettres que l'on va successivement crypter. Ces deux premières lettres sont S=18 et E=4.

Les deux premières lettres du message crypté sont obtenues en multipliant le vecteur colonne **A** par **M** :

$$\mathbf{M}.\mathbf{A} = \begin{bmatrix} 3 & 2 \\ 1 & 3 \end{bmatrix}\begin{bmatrix} 18 \\ 4 \end{bmatrix} = \begin{bmatrix} 62 \\ 30 \end{bmatrix} \equiv \begin{bmatrix} 10 \\ 4 \end{bmatrix} \pmod{26}$$

On obtient « KE » pour les lettres cryptées. On répète cette opération pour les 2 groupes suivants. Finalement, le message crypté est « KEOBYJ ».

Pour le décoder on utilise la matrice inverse modulo 26. Pour les 2 premiers caractères « KE », on a :

$$\mathbf{M}^{-1}.\mathbf{B} = \begin{bmatrix} 19 & 22 \\ 11 & 19 \end{bmatrix}\begin{bmatrix} 10 \\ 4 \end{bmatrix} = \begin{bmatrix} 278 \\ 186 \end{bmatrix} \equiv \begin{bmatrix} 18 \\ 4 \end{bmatrix} \pmod{26}$$

On retrouve bien « SE ».

On pourra trouver plus de détails sur le chiffre de Hill et un calculateur pour des matrices 2×2 sur le site http://www.nymphomath.ch/crypto/hill/index.html

13. Complexité

Beaucoup des algorithmes de cryptographie moderne sont basés sur la multiplication de nombres premiers et la factorisation. Voyons cela de plus près avec deux nombres premiers 127 et 457.

La **complexité** est une estimation du nombre d'opérations élémentaires nécessaire pour faire une opération comme l'addition ou la multiplication de deux nombres. Plus le nombre d'opérations élémentaires est élevé, plus le temps de calcul l'est aussi.

Faire l'addition des deux nombres est simple :

127 + 457 = 584

Cela demande, hors retenues, environ 3 opérations. S'il y avait un maximum de retenues cela ferait 6 opérations. Donc, typiquement, pour la somme de deux entiers de n chiffres, on pourrait dire que la complexité varie comme $\sim n$

La multiplication est plus complexe. Il faut faire, hors retenues, 9 multiplications et des additions. On peut dire que la complexité varie comme $\sim n^2$. Pour l'exemple choisi la multiplication donne : 127×457=58 039

Factoriser le nombre 58 039 est cependant une opération beaucoup plus difficile. Pour la meilleure méthode, la complexité varie comme $\exp(4n^{\frac{1}{3}})$.

Pour des nombres premiers de 200 chiffres, par exemple, $n^2 = 40\ 000$ alors que pour la factorisation cette complexité est de 14 423 748 777 soit 360 593 fois plus. Cela veut dire que plus la valeur des nombres premiers sera élevée, plus il sera difficile de factoriser le produit pq de deux nombres premiers p et q. Les temps pour calculer le produit pq et pour décomposer ce produit en facteurs premiers p et q sont donc très différents. C'est à la base de la cryptographie asymétrique. Si les nombres premiers sont

bien choisis, il est pratiquement impossible, dans un temps raisonnable, de trouver p et q.

L'information apportée par la connaissance de p et de q est plus riche que celle de la connaissance du produit pq seul.

14. Fonction à sens unique, fonction trappe

Une **fonction** bijective $f(x)$ à dite à **sens unique** si l'on peut facilement calculer $f(x)$ à partir de x mais si le calcul inverse (obtenir x à partir de $f(x)$) est extrêmement difficile). Il y a une asymétrie entre un sens et l'autre.

Dans le principe, si l'application f permet le chiffrement, l'application inverse f^{-1} permet le déchiffrement. Avec une fonction à sens unique le chiffrement est très rapide mais le déchiffrement demande un temps extrêmement long. Or, si c'est un avantage pour protéger son message codé des pirates, une telle méthode ne peut être utilisée telle quelle pour le déchiffrage.

Pour cela on utilise une **fonction trappe** qui appartient à une famille de fonction à sens unique f_t dépendant d'un index t qui permet de rapidement et facilement calculer f_t si l'on connaît t.

On peut illustrer cela avec l'exemple présenté par l'Université de Lille dans ses vidéos sur la cryptographie.

Soit la fonction $f : x \mapsto x^3 \pmod{100}$. Pour trouver x tel que $x^3 = 11 \pmod{100}$ il faut tester tous les éléments de 0 à 99. Après une recherche fastidieuse on trouve la valeur 71. En effet, $71^3 = 357911 \equiv 11 \pmod{100}$.

Il existe toutefois une fonction trappe qui permet d'obtenir facilement le résultat. Il s'agit de $f : y \mapsto y^7 \pmod{100}$.

Cela fonctionne de la manière suivante. Il suffit de prendre pour y le résultat congru que l'on cherche, c'est-à-dire 11 pour cet exemple. On peut vérifier que $11^7 = 19\ 487\ 171 \equiv 71 \pmod{100}$. On trouve donc que pour l'ensemble des nombres compris entre 0 et 99, la solution 71. En effet, $71^3 = 11 \pmod{100}$.

Cela est bien entendu valable pour d'autres valeurs. En effet, supposons que l'on veuille résoudre $x^3 = 23 \pmod{100}$. La fonction trappe donnera :

$23^7 = 3\ 404\ 825\ 447 \equiv 47 \pmod{100}$ et l'on a $47^3 = 23 \pmod{100}$.

15. Fonction de hachage

Une fonction de hachage transforme une donnée quelconque (message texte, image, musique, etc. en une donnée numérique de taille fixée de faible longueur. La donnée numérique obtenue est l'**empreinte** (ou **signature**) de la donnée initiale.

La fonction de hachage générant l'empreinte (signature, haché) possède les propriétés suivantes :

- La longueur de l'empreinte (signature) est toujours la même, quelle que soit la longueur du message en entrée.
- Cette empreinte doit être unique. Deux messages, même très proches ont une empreinte différente.
- La fonction de hachage doit être une fonction à sens unique afin qu'il ne soit pas possible, à partir de la signature, de remonter au message initial.

En comparant les signatures de deux données, on peut savoir si elles sont identiques ou non. Il y a de multiples usages du hachage. L'empreinte peut servir à s'assurer qu'un téléchargement s'est fait correctement. En effet, il suffit de calculer l'empreinte du fichier téléchargé et de comparer celle-ci à l'empreinte jointe au fichier. Si les deux sont identiques, cela signifie que le téléchargement s'est effectué correctement.

Une autre application concerne le hachage des mots de passe. Certaines sociétés gardent en clair sur leur serveur le mot de passe de leurs clients. Si leurs serveurs sont piratés, les mots de passe le sont aussi ainsi que les coordonnées des clients. C'est la raison pour laquelle les sociétés sérieuses ne stockent plus le mot de passe de leurs clients mais les empreintes de ces mots de passe. On procède alors de la manière suivante :

- Lorsque le client s'inscrit et tape tape son identifiant et son mot de passe, ce dernier est haché et transformé en empreinte au niveau de son ordinateur. Le couple identifiant/empreinte est envoyé sur le serveur du commerçant qui le stocke sur ses serveurs.
- Lorsque le client se connecte à nouveau, il s'identifie avec le couple identifiant/mot de passe. Ce dernier est haché et l'ensemble identifiant/empreinte est envoyé de l'ordinateur du client au serveur du commerçant. Après vérification de l'identifiant et de l'empreinte, le serveur autorise l'accès si tout est correct.

On dit qu'il y a collision si deux données différentes donnent la même empreinte. Dans certain cas, comme celui du stockage de mots de passe et plus généralement celui du stockage de données.

Il existe un grand nombre de fonctions de hachage cryptographiques. Citons l'algorithme MD5 (Message Digest 5) proposé par Ronald Rivest en 1991 ou les algorithmes SHA (Secure Hasch Algorithm). SHA1, conçus par la NSA, utilise une signature de 160 bits et SHA mais n'est plus considérée comme sure, de même que MD5 si les attaquants possèdent des moyens importants. SHA2 a aussi été conçue par la NSA comporte les algorithmes SHA-224, SHA-256 et SHA-512 qui utilisent 224, 256 et 512 bits.

Pour illustrer le hash d'un texte, prenons la phrase de Corneille dans le Cid :

« Nous partîmes cinq cents ; mais par un prompt renfort - Nous nous vîmes trois mille en arrivant au port. ».

Un hash 256 de cette phrase (http://www.convertstring.com/fr/Hash/SHA256) est :

2A9F932B44E7D2D5F60751948A35D9593D5257D0B6D62840E117E32E2066A5D8

Prenons la même phrase dans laquelle nous avons juste enlevé le point final.

« Nous partîmes cinq cents ; mais par un prompt renfort - Nous nous vîmes trois mille en arrivant au port ».

Le même hash devient :

9FAE560ACDE202CDC3ECCEC86A93E3E974D23D0D29BC768ECAC389111DE8DF13

Les deux hash sont très différents alors qu'un point seulement a été enlevé entre le texte intial et le texte modifié.

16. Cryptographie symétrique

Dans une **cryptographie symétrique**, la clef utilisée lors du chiffrement est la même que celle utilisée lors du déchiffrement. Le chiffre de César, Vigenère ou la machine Enigma sont des systèmes de chiffrement symétriques. On qualifie aussi souvent ce type de chiffrement de **système de chiffrement à clef secrète**.

L'émetteur et le destinataire possèdent la même clef pour chiffrer et déchiffrer le message. Le problème majeur de ce système est la transmission de la clef qui doit être changée pour chaque message pour plus de sécurité. L'échange des clefs se fait souvent par des échanges de type « valise diplomatique » mais cela demande une logistique importante et donc des coûts importants.

On peut comparer le chiffrement symétrique à un coffre-fort dans lequel on met le message. L'émetteur et le destinataire sont les seuls à en avoir la clef.

17. Un peu d'arithmétique

Le chiffrement asymétrique est basé sur le fait qu'il est facile de calculer le produit n de deux grands nombres premiers p et q mais qu'il est pratiquement impossible, avec les technologies actuelles, de décomposer n en p et q lorsque ces derniers sont inconnus. Nous allons présenter, dans cette section, quelques outils mathématiques permettant d'aborder la cryptogr aphie symétrique. Il ne s'agit que de notions élémentaires, les problèmes réels étant d'une autre complexité. Cette introduction devrait toutefois permettre au lecteur de toucher du doigt la nature des problèmes posés.

Petit théorème de Fermat

Le petit théorème de Fermat dit que si p est un nombre premier et a un entier appartenant à $\mathbb{Z}$:

$a^p \equiv a \pmod{p}$

Si p ne divise pas a on a alors, en divisant les deux membres par a :

$a^{p-1} \equiv 1 \pmod{p}$

On peut déduire de cela la propriété suivante :

Soient 2 nombres premiers p et q ($p \neq q$) et $n = pq$. Pour tout nombre a qui n'est pas divisible par p ou q ($\text{pgcd}(a,n) = 1$) on a :

$$a^{(p-1)(q-1)} \equiv 1 \pmod{n}$$

On peut vérifier cela sur des nombres simples.

Prenons $p = 3$ et $q = 5 \Rightarrow n = 15$.

$(p-1)(q-1) = 2 \times 4 = 8$. On a donc pour a :

$a = 1, 2, 4, 7, 8 \ldots$

On peut vérifier que :

$8^8 \equiv 1 \pmod{15}$, $7^8 \equiv 1 \pmod{15}$, $4^8 \equiv 1 \pmod{15}$ et $2^8 \equiv 1 \pmod{15}$

PGCD

Le PGCD est le plus grand commun diviseur de deux nombres (GCD en anglais pour Great Common Divisor). Par exemple

pgcd(258,124)=2 car 258=43×3×2 et 124=67×2. Le seul diviseur en commun est 2.

L'algorithme d'Euclide permet de calculer le pgcd de deux nombres :

$$\text{pgcd}(a,b) = \text{pgcd}(b, a \mod b)$$

Si l'on veut calculer le pgcd de 4950 et 4875, on peut soit faire la décomposition

$4950=2\times3^2\times5^2\times11$ et $4875=3\times5^3\times13$. Le pgcd est donc $3\times5^2=75$.

Une autre manière de calculer le pgcd est d'utiliser l'algorithme d'Euclide :

pgcd(4950,4875)=pgcd(4875,4950 mod(4875)) qui vaut 4950-4875=75

Théorème de Bezout

Le théorème de Bezout dit que si a et b sont deux nombres premiers, alors il existe deux nombres entiers relatifs u et v tels que :

$$au + bv = 1$$

Les coefficients u et v sont les coefficients de Bezout.

L'algorithme d'Euclide étendu permet d'obtenir les coefficients de Bezout de l'identité :

$$au + bv = \text{pgcd}(a,b)$$

Lorsque a et b sont premiers entre eux : $\text{pgcd}(a,b) = 1$.

Prenons $a = 13$ et $b = 23$. Dans ce cas on peut vérifier que $u = -99$ et $b = 56$ sont des valeurs qui satisfont l'équation ci-dessus.

Exponentiation modulaire

Nous allons voir maintenant comment on peut plus rapidement calculer $a^k \pmod n$ que par un calcul où l'on évalue directement a^k avant de calculer la valeur $\pmod n$.

Pour cela on commence par développer l'exposant k en puissances de 2 :

$k = \sum_{i=0}^{m} c_i 2^i$. Par exemple si $k = 11$ on a $k = 2^3 + 2^1 + 2^0$

On a alors :

$$a^k = a^{\sum_{i=0}^{m}(c_i 2^i)} = \prod_{i=0}^{m} \left(a^{2^i}\right)^{c_i}$$

Supposons que $a = 7$ et supposons que l'on veuille calculer $a^{11} \pmod{23}$.

On suit le schéma suivant :

$7^{11} = 7^8 \times 7^2 \times 7$

$7 \equiv 7 \pmod{23}$

$7^2 = 49 = 46 + 3 \equiv 3 \pmod{23}$

$7^4 = 7^2 \times 7^2 \equiv 3 \times 3 = 9 \pmod{23}$

$7^8 = 7^4 \times 7^4 \equiv 9 \times 9 = 81 \equiv 12 \pmod{23}$

Au total : $7^{11} \equiv 12 \times 3 \times 7 = 252 \equiv 22 \pmod{23}$

Prenons un cas un peu plus compliqué : $23^{198} \pmod{128}$

On procède de la manière suivante

$198 = 128 + 64 + 4 + 2 = 2^7 + 2^6 + 2^2 + 2^1 = 11000110_2$

$23 \equiv 23 \mod(128)$

$23^2 \equiv 17 \mod(128)$

$23^4 \equiv 33 \mod(128)$

$23^8 \equiv 65 \mod(128)$

$23^{32} \equiv 1 \mod(128)$

$23^{64} \equiv 1 \mod(128)$

$23^{128} \equiv 1 \mod(128)$

Finalement $23^{198} \equiv 1 \times 1 \times 33 \times 17 \equiv 49 \pmod{128}$

18. Cryptographie asymétrique

Dans la **cryptographie asymétrique**, la clef pour chiffrer le message est différente de celle utilisée pour le déchiffrer. Ce type de chiffrement est aussi appelé **système de chiffrement à clef publique**.

Le principe du chiffrement par clef asymétrique a été imaginé en 1976 par Diffle et Hellman. Le premier algorithme basé sur ce principe a été mis en œuvre en 1977 par Rivest, Shamir et Adleman. Il est connu sous le nom de chiffrement RSA. Ce protocole, breveté par le MIT en 1983, est tombé dans le domaine public en 2000.

Dans le chiffrement asymétrique, le chiffrement du message se fait au moyen d'une clef connue de tous, la clef publique. Cependant, chaque destinataire a une clef qu'il garde secrète et qui lui permet de déchiffrer le message. C'est un peu l'analogue d'une boîte aux lettres : tout le monde peut y glisser une lettre mais il n'y a que le destinataire qui en a la clef.

Le concept defonction à sens unique et de fonction trappe dont nous avons parlé plus haut permet de comprendre le principe de la cryptographie asymétrique. Le chiffrement se ferait avec fonction $x \mapsto x^{\mathrm{K}} \pmod{100}$ où K est la clef publique. Celle-ci est vaut $K = 3$ pour l'exemple choisi plus haut. Donc, pour envoyer le message x on utilise la clef publique.

La clef privée permet de déchiffrer les messages codés. Pour l'exemple considéré elle vaut $K' = 7$.

19. Le code RSA

Le chiffrement RSA est basé sur une clef publique, accessible à tous, et une clef privée accessible seulement à celui qui doit décoder le massage. Dans le domaine de la

cryptographie plutôt que de désigner les interlocuteurs qui s'échangent des messages par « A » ou « B », on utilise des prénoms : Alice, Bob, etc.

Supposons que Bob veuille envoyer un message secret à Alice en utilisant une cryptographie asymétrique. Pour cela il faudra réaliser trois étapes.

1. Alice va générer deux clefs : une clef publique que tout le monde pourra utiliser, notamment Bob, et une clef privée qui sera à son usage propre et que personne ne doit connaître.
2. Bob crypte son message avec la clef publique d'Alice et lui envoie son message crypté.
3. Alice se sert de sa clef privée pour décrypter le message.

Nous allons présenter ici de manière simplifiée, le principe du protocole RSA. Nous allons prendre l'exemple présenté dans la vidéo réalisée par l'Université de Lille. La raison de ce choix est que nous recommandons fortement au lecteur cette série de vidéos qui présentent de manière très claire et compréhensive la cryptographie tout en développant les outils mathématiques à un niveau supérieur à ce qui est donné ici. La vidéo relative au système RSA complète donc ce qui va être dit, notamment en explicitant les démonstrations mathématiques.

Génération des clefs

On prend deux nombres premiers distincts p et q avec lesquels on calcule les deux quantités suivantes :

$n = p \times q$ (module de chiffrement) et

$\varphi(n) = (p-1)\times(q-1)$ (indicatrice d'Euler)

On choisit ensuite un exposant e qui soit premier avec $\varphi(n)$ ce qui veut dire que $\text{pgcd}(e, \varphi(n)) = 1.$

On calcule ensuite l'inverse d de e modulo $\varphi(n)$:

$$d \times e \equiv 1 \ (\text{mod}\, \varphi(n))$$

Pour trouver l'inverse on utilise l'algorithme d'Euclide étendu qui permettra d'obtenir les coefficients de l'identité de Bezout. On peut aussi les calculer en utilisant le site internet https://calculis.net/bezout.

Avec ces opérations on crée une clef publique qui est constituée du couple (n, e) et une clef privée qui est d.

Chiffrement

Bob veut envoyer un message secret à Alice. Il va le convertir en un ensemble de chiffres. C'est ce qu'il fera par exemple en codant ses caractères alphanumériques en ASCII mais d'autres codages sont possibles. Le message va générer un ou plusieurs entiers m. Il est nécessaire que $0 \leq m < n$.

On calcule le message chiffré par la formule :

$$x = m^e \ (\text{mod}\, n)$$

en utilisant la clef publique (n, e) et la méthode d'exponentiation rapide décrite plus haut. Une fois le message codé Bob pourra l'envoyer à Alice.

Déchiffrement

Alice a reçu le message chiffré x. Pour le déchiffrer, elle va utiliser sa clef privée d et calculer :

$$m = x^d \pmod{n}$$

Le schéma du codage RSA est indiqué dans la figure 5.

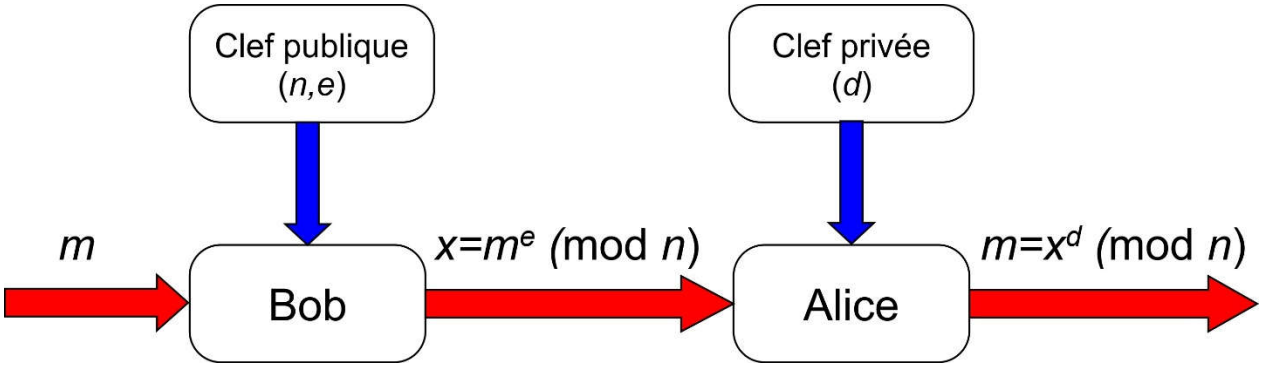

Figure 5

En fait, le cryptage décryptage RSA est basé sur le lemme suivant :

Soit $n = pq$ (p et q sont deux nombres premiers distincts) et d l'inverse de e modulo $\varphi(n)$. Si $x \equiv m^e \pmod{n}$ alors $m \equiv x^d \pmod{n}$

Application

Nous allons choisir l'exemple de l'Université de Lille sur lequel le lecteur pourra avoir plus de détails en regardant la vidéo correspondante.

Dans la pratique les nombres premiers mis en œuvre comportent des centaines de chiffres pour rendre le déchiffrement impossible. Ici les nombres premiers choisis sont très petits afin de rendre les calculs faciles.

Génération des clefs

On choisit $p = 5$ et $q = 17$
$n = p \times q = 5 \times 17 = 85$
$\varphi(n) = \varphi(85) = (p-1) \times (q-1) = 4 \times 16 = 64$
On prend $e = 5$ pour lequel on a bien $\text{pgcd}(e, \varphi(n)) = \text{pgcd}(5, 64) = 1$
la clef publique est donc $(n = 85, e = 5)$

Les coefficients de Bezout peuvent être calculés à partir du site internet cité plus haut.

$5 \times 13 + 64 \times (-1) \equiv 1 \pmod{64}$
Donc $5 \times 13 \equiv 1 \pmod{64}$
La clef privée est $d = 13$.

Cryptage

On va chiffrer me mot $m = 10$.

Le message chiffré sera égal à

$x = 10^5 \pmod{85} = 40 \pmod{85}$
Donc $x = 40$

Décryptage

Pour déchiffrer le message crypté x , on utilise la clef privée d .

$m = x^d \pmod{n} = 40^{13} \pmod{85} = 10 \pmod{85}$

20. Comparaison

Le problème majeur de la cryptographie symétrique est l'échange des clefs. En revanche elle est facile à mettre en œuvre, demande peu de ressources et le chiffrement et déchiffrement sont assez rapides.

La cryptographie asymétrique n'a pas le problème d'échange de clef mais elle est plus complexe à mettre œuvre. Elle nécessite des ressources matérielles coûteuses et s'avère plus lente que la cryptographie symétrique.

Le logiciel de messagerie PGP (Pretty Good Privacy) est une application grand public intéressante d'un système de chiffrement hybride alliant les cryptographies symétriques et asymétriques. Une clef secrète à usage unique est générée pour chaque message. Cette clef est utilisée avec un algorithme de chiffrement symétrique pour chiffrer le message. La clef est ensuite chiffrée par un algorithme de type RSA. Le message et la clef chiffrés sont envoyés simultanément au destinataire. Comme il n'y a que la clef secrète qui est chiffrée de façon asymétrique, le traitement est très rapide.

Philippe Zimmermann a mis à la disposition au public, en 1991, le premier logiciel de chiffrement basé sur cette idée. Cela conduit les douanes américaines à ouvrir une enquête criminelle à son égard pour violation des restrictions sur l'exportation de logiciels cryptographiques. Toutefois, les banques trouvèrent ce système si pratique pour protéger les données communiquées à leurs clients que cela se solda par une procédure sans suite début 1996.

21. L'intrication

L'intrication quantique est un phénomène étonnant. Considérons un système de deux électrons (spin ½) et formons un système quantique qui est une superposition d'un électron dans l'état $|+\rangle$ (projection du spin +1/2) et d'un électron dans l'état $|-\rangle$ (projection du spin -1/2). On va construire l'état intriqué suivant :

$$|+-\rangle + |-+\rangle$$

dans lequel l'un des électrons est dans l'état de spin +1/2 et l'autre dans l'état de spin -1/2 pour le premier état, et l'inverse pour le second. Un tel état ne veut pas dire que la moitié des électrons est dans l'état $|+\rangle$ et l'autre moitié dans l'état $|-\rangle$. Ils sont dans l'état quantique $|+-\rangle + |-+\rangle$.

Supposons maintenant que les deux électrons soient séparés par une grande distance tout en restant dans cet état quantique intriqué (ce qui n'est pas techniquement facile à cause des perturbations extérieures). Si l'on mesure la projection du spin d'un des deux électrons, on ne sait pas quel résultat on va trouver. Il y a 50% de chance de mesurer +1/2 et 50% de chance de mesurer -1/2. Cependant, une fois la mesure faite, on a un résultat. Supposons que l'on ait mesuré +1/2. Dans ce cas une mesure de la projection du spin de l'autre électron donnera nécessairement -1/2, quelle que soit la distance à laquelle il se trouve. Si l'on fait les mesures presque simultanément tout se

passe comme si la connaissance du résultat de la mesure sur l'un des électrons informait l'autre à une vitesse supérieure à celle de la lumière.

Dans la pratique, on utilise des photons (deux états de polarisation) pour effectuer des expériences d'intrication (entanglement en anglais). En 2017, des scientifiques chinois ont fait une expérience d'intrication sur photons séparés 1400 km. En 2013, l'intrication a été démontrée sur des électrons séparés de 1,3 km.

Même si le résultat de la mesure sur un électron se propage instantanément à l'autre particule, cela ne veut pas dire que l'on puisse transmettre de l'information à une vitesse supérieure à celle de la vitesse de la lumière car le résultat d'une mesure est probabiliste et on ne peut pas transmettre par cette méthode un message déterministe. Les expériences d'intrication montrent que des mesures peuvent s'influencer à une vitesse supérieure à la vitesse de la lumière. Toutefois, pour transmettre le résultat d'un point en un autre, il faut utiliser des moyens classiques qui bien sûr ne peuvent pas transmettre de l'information à une vitesse supérieure à celle de la lumière.

22. Cryptographie quantique

La cryptographie quantique n'est pas une méthode dans laquelle on utilise un protocole de cryptage quantique pour transmettre des données mais une méthode de **distribution des clefs**. Elle permet de s'assurer qu'une clef transmise d'un émetteur vers un destinataire n'a pas été interceptée et lue lors de sa transmission. L'intérêt d'utiliser la mécanique quantique pour cette transmission est que toute mesure perturbe profondément un système et ne peut pas passer inaperçue. Toute interception lors de la transmission de la clef sera obligatoirement détectée par le destinataire qui ne l'utilisera pas et demandera la transmission d'une autre clef.

Plusieurs protocoles quantiques pour transmettre les clefs sont utilisés. Celui de Bennet et Brassard proposé en 1994 (BB84) est basé sur la polarisation des photons. Le protocole E91, développé par Artur Ekert en 1991 utilise des photons intriqués.

23. L'ordinateur quantique

Un ordinateur classique travaille avec des **bits** (bit = binary digit). Le bit est la quantité d'information élémentaire et sa valeur peut être de 0 ou 1. Un ordinateur quantique travaille avec des **qubits** ou **Qbits**(quantum bit). Le qubit élémentaire est une superposition entre 2 états $|0\rangle$ et $|1\rangle$:

$$\alpha|0\rangle + \beta|1\rangle$$

Où α et β sont des constantes. On peut former une infinité d'états formés par superposition (figure 6). Lors d'une mesure d'un qubit, on ne peut savoir que s'il est dans l'état $|0\rangle$ ou dans l'état $|1\rangle$. Cependant, la probabilité d'être dans l'un ou l'autre des états dépendra des coefficients α et β.

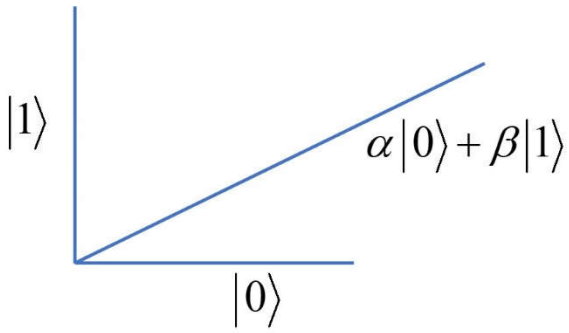

Figure 6

Dans un ordinateur classique on utilise des registres composés d'un certain nombre de bits. Par exemple, avec un registre de 2 bits on pourra avoir les 4 possibilités suivantes : $|00\rangle$, $|01\rangle$, $|10\rangle$ et $|11\rangle$. Un qubit sera une combinaison linéaire de ces quatre valeurs du registre.

Un ordinateur classique fonctionne avec des portes logiques. Un ordinateur quantique fonctionne avec des portes quantiques qui se comportent de manière différente. Un ordinateur quantique n'est intéressant que pour résoudre certains problèmes spécifiques. Par exemple, si on se donne un tableau où pour chaque valeur de x on associe un nombre donné par une fonction $f(x)$ et si l'on cherche la valeur de x correspondant à une valeur de $f(x)$, il faudra, avec un ordinateur classique parcourir l'ensemble du tableau jusqu'à ce que l'on trouve la bonne réponse. Typiquement, si on a un tableau de N lignes, il faudra au maximum N tests. Avec un ordinateur quantique et l'algorithme de Grover, il faudra au maximum $\sqrt{N}$ essais. Si le tableau fait par exemple 100 millions de lignes (10^8) il faudra faire en moyenne 50 millions d'essais avec un ordinateur classique et 5000 avec un ordinateur quantique.

Un intérêt de l'ordinateur quantique, lorsqu'il sera constitué d'un nombre de qubits assez important, est qu'il pourra factoriser beaucoup plus vite un grand nombre en facteurs premiers. Il sera exponentiellement meilleur qu'un ordinateur classique sur ce type de problème grâce à l'algorithme de Schor (1994). Il sera donc possible de casser un code RSA en un temps raisonnable ce qui n'est pas possible actuellement avec les calculateurs classiques. Avec une clef RSA sur 1024 bits, il faudrait environ 1 million d'années de temps CPU pour cracker le code. Avec l'algorithme de Shor et un calculateur quantique cela pourrait être fait en 1 heures.

Pour le moment les ordinateurs quantiques n'ont qu'un petit nombre de qubits. IBM a lancé fin 2017 un ordinateur de 50 qubits mais il en faudrait plusieurs centaines pour avoir la puissance nécessaire pour casser rapidement un code RSA. Lorsque des ordinateurs quantiques assez puissants seront disponibles, le code RSA pourra être cassé en un temps raisonnable. Si l'économie du net (banques, achats en lignes, etc.) n'a pas anticipé cela, la situation risque d'être catastrophique pour tous les échanges mettant en jeu un cryptage asymétrique.

24. Pour en savoir plus

Cryptographie & codes secrets, Bibliothèque Tangente, Hors-série n°26, 2013

P.Guillot, La cryptologie : l'art des codes secrets, EDP Sciences, 2013

B.Martin, Codage, cryptologie et applications, Presses Polytechniques et Universitaires Romandes, 2004

F.Recher, A.Bodin et G.Vantomme, séries de vidéos sur la cryptographie (youtube), Université de Lille

P.Vigoureux, Comprendre les codes secrets, Ellipses, 2010