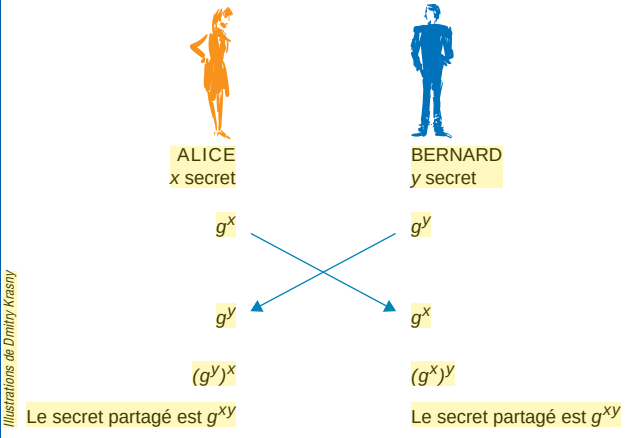
un message à Alice, il le chiffre avec sa clé secrète, puis il chiffre le résultat en utilisant la clé publique d'Alice. À la réception, Alice effectue les étapes inverses : elle déchiffre le message avec sa clé secrète, puis elle déchiffre le résultat avec la clé publique de Bernard. Si elle parvient à lire le message, elle est certaine que c'est bien Bernard qui l'a écrit.

Ces opérations de chiffrement et de déchiffrement nécessitent des myriades d'opérations mathématiques, mais des logiciels exécutables par des ordinateurs personnels automatisent le processus ; l'utilisateur n'a qu'à cliquer sur les boutons «chiffrer» ou «déchiffrer» (ainsi mon programme PGP, l'acronyme de *Pretty Good Privacy*, soit «confidentialité raisonnable», est distribué gratuitement sur le réseau Internet ; son utilisation en France est maintenant autorisée).

Cryptographie à clé publique



Illustrations de Dmitry Krasny

Pendant des siècles, la cryptographie a été handicapée par le problème de l'échange des clés. Supposons que Bernard ait envoyé un message chiffré à Alice. Cette dernière ne pourra lire le message que si elle reçoit également la clé de codage. Les systèmes cryptographiques à clé publique évitent cet envoi, grâce à l'utilisation de mathématiques ingénieuses.

Avec l'algorithme de Diffie et Hellman, Alice utilise son nombre secret x pour calculer g^x (où g est un nombre public) et elle envoie cette valeur à Bernard. De son côté, Bernard utilise son nombre secret y pour calculer g^y et envoie le résultat à Alice. À la réception, Alice peut alors calculer $(g^y)^x$, tandis que Bernard calcule le même nombre $(g^x)^y$. Cette quantité devient leur clé secrète partagée.

Un intrus qui aurait intercepté la valeur g^x d'Alice, ainsi que la valeur g^y de Bernard peut en déduire les valeurs secrètes de x et de y. Aussi, pour déjouer toute indiscrétion, Alice et Bernard utilisent la fonction modulo, qui donne le reste d'une division (par exemple, 14 modulo 4 est égal à 2 : le reste de la division de 14 par 4 est égal à 2). Cette astuce assure le secret, car, au lieu d'envoyer g^x à Bernard, Alice lui transmet le nombre g^x modulo p, à partir duquel un pirate aura bien du mal à retrouver x, même s'il connaît g et p.

En compliquant l'algorithme de Diffie et Hellman, on a obtenu des systèmes qui engendrent deux clés complémentaires, l'une secrète (x pour Alice) et l'autre publique (à savoir g, p et g^x modulo p). La clé secrète déchiffre le message chiffré par la clé publique, mais la clé utilisée pour chiffrer un message ne peut être utilisée pour le déchiffrer. Ainsi, Bernard peut utiliser la clé publique d'Alice (qu'elle a donnée à tout le monde) pour chiffrer un message qui lui est destiné, mais seule Alice peut déchiffrer ce message à l'aide de sa clé secrète.

Malgré ces progrès, la cryptographie à clé publique a deux graves limitations. D'une part, la technique est relativement lente, de sorte qu'en pratique on ne peut chiffrer les messages trop longs. D'autre part, des structures caractéristiques du message en clair subsistent dans le message chiffré. Détectables dans le texte chiffré, ces fragments sont utilisables par des spécialistes, qui peuvent reconstituer la totalité des messages.

Chevaux de bataille symétriques

Pour ces raisons, le chiffrement est souvent assuré par des techniques symétriques, plus rapides et plus sûres, et la cryptographie à clé publique ne sert que lors de l'étape d'échange des clés symétriques. Par exemple, Bernard chiffre son message avec une clé symétrique rapide et sûre. Comme il doit envoyer à Alice la clé qu'il a utilisée, il la chiffre avec la clé publique d'Alice et attache le résultat à son message chiffré. Alice pourra déchiffrer la clé symétrique en utilisant sa clé secrète. Muni de la clé symétrique, elle déchiffrera alors le reste du message de Bernard.

Pour authentifier ses messages, Bernard ne les signe pas non plus à l'aide des techniques à clé publique ; il extrait du message un condensé numérique, ou «empreinte digitale», à l'aide d'une transformation mathématique, nommée fonction de hachage. Cette transformation associe à un message de longueur quelconque un message de longueur fixe, telle que 160 bits (un bit est un chiffre binaire, égal à 0 ou à 1). Les programmes cryptographiques sont conçus de manière que deux messages différents n'aient pas le même condensé. Autrement dit, l'empreinte digitale est unique, deux messages différents ayant à peu près certainement deux condensés différents.

Après avoir calculé le condensé de son message, Bernard le chiffre avec sa clé secrète. Il envoie cette «signature» en même temps que le reste de son message chiffré. Lorsqu'elle reçoit ce condensé chiffré, Alice le déchiffre grâce à la clé publique de Bernard. Elle compare alors le résultat obtenu avec le résultat qu'elle calcule elle-même après déchiffrement du message. Si les deux résultats sont identiques, elle conclut que le message n'a pas été modifié et que Bernard en est bien l'expéditeur.

Pour chiffrer un message destiné à circuler sur le réseau Internet, on le découpe souvent en blocs de taille fixe (généralement 64 ou 128 bits), puis on chiffre individuellement les blocs successifs. Ce système de chiffrement par blocs crypte plusieurs fois chaque bloc (le nombre d'itérations dépend des algorithmes utilisés). Chaque itération fait intervenir des permutations circulaires (la permutation d'un groupe de trois blocs «xtv» devient «tvx») et des substitutions («tvx» engendre «cb2»). Une partie de la clé participe à la transformation des données au cours de chaque itération.

Quand on fournit des morceaux de texte identiques à un système de chiffrement, on obtient des blocs identiques. Aussi, afin d'éviter le maintien de structures qui faciliteraient la découverte du message, les algorithmes par blocs utilisent un chaînage : les blocs qui viennent d'être chiffrés servent à déchiffrer les blocs suivants. Le chiffrement d'un bloc de texte dépend alors de tous les blocs qui le précèdent.

La longueur des clés de chiffrement par blocs est généralement égale à 56, 128 ou 256 bits. Les méthodes les plus connues sont DES (*Data Encryption Standard*, soit

«standard de chiffrement de données»), triple-DES, CAST, IDEA et *Skipjack*. Les algorithmes par blocs sont au centre de la plupart des recherches actuelles en cryptographie.

La clé est la clé

En cryptographie, l'étape la plus sensible est la création des clés. Pour qu'un système soit le plus sûr possible, les clés doivent être des nombres réellement aléatoires, imprévisibles par un éventuel pirate. Les séquences pseudo-aléatoires déterministes que les ordinateurs engendrent pour les jeux ou les simulations sont inutilisables, car ils sont insuffisamment aléatoires. La seule manière d'engendrer de vrais nombres aléatoires est d'utiliser un «bruit» physique, telle la désintégration radioactive.

À l'aide d'un ordinateur, on fabrique difficilement des nombres aléatoires aussi bons que ceux issus de la physique. On obtient toutefois des résultats raisonnables quand on utilise l'intervalle de temps, en microsecondes, qui sépare la frappe de deux touches d'un clavier : cette valeur est impossible à prédire. Les données obtenues ainsi ne sont pas assez aléatoires pour fabriquer directement des clés, mais peuvent être transformées pour accroître le désordre.

La seule méthode cryptographique dont on ait prouvé la parfaite sécurité est le chiffrement de Vernam, où la longueur de la clé est égale à la longueur du message. Dans cette méthode, on utilise une séquence aléatoire pour chiffrer le message bit par bit, c'est-à-dire que le 34e bit de la clé code le 34e bit du message. La clé doit être parfaitement aléatoire, elle ne peut pas être une séquence pseudo-aléatoire issue d'un algorithme déterministe, sinon le code pourrait facilement être cassé. Le chiffrement de Vernam est rarement utilisé, en raison de difficultés pratiques : la clé, qui est aussi longue que le message, doit être envoyée au destinataire par un canal de transmission sûr. De surcroît, on ne peut utiliser chaque clé qu'une seule fois, sinon un pirate pourrait la décrypter.

Si la taille des clés détermine l'efficacité des systèmes cryptographiques, la conception des systèmes est également très importante. Considérons un simple code de substitution où tous les A seraient transformés en W, tous les B en K, tous les C en Q, et ainsi de suite. Le nombre de façons différentes d'arranger l'alphabet est égale à 26! (ce nombre, qui se lit «factorielle 26», est égal à $26 \times 25 \times 24 \ldots 3 \times 2 \times 1$), soit environ 4×10^{26} (un nombre de 26 chiffres)! On ne peut tester toutes les combinaisons une à une, mais, lorsque j'étais enfant, je m'amusais à casser ce type de code avec un papier et un crayon : je regardais simplement la lettre la plus fréquente et je supposais qu'il s'agissait d'un E ; puis je cherchais la deuxième lettre la plus fréquente et je supposais que c'était un S, etc. Ainsi, malgré un espace des clés très grand, ce type de code est très simple.

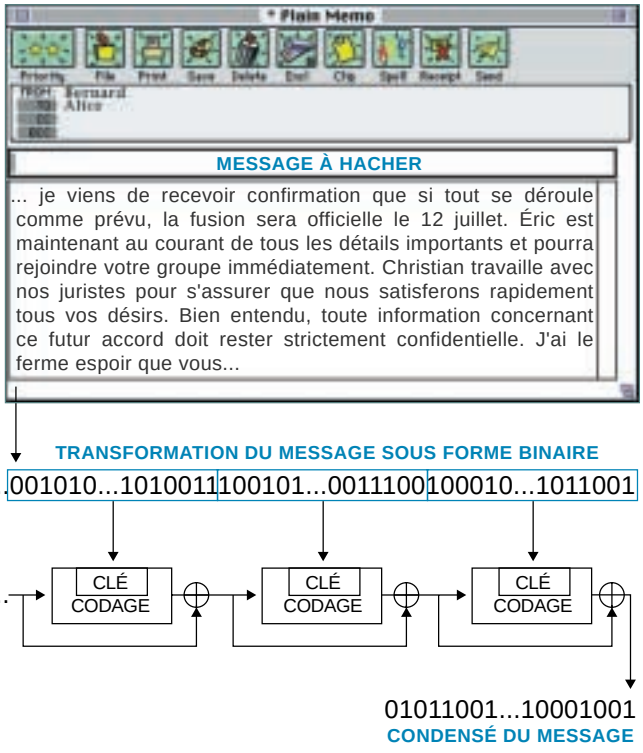
En fait, la sécurité d'un système cryptographique n'est pas proportionnelle à la taille de la clé. Pour les systèmes par blocs, par exemple, la relation est plutôt exponentielle : un bit de plus à la clé oblige un pirate éventuel à doubler le nombre d'essais. L'effort nécessaire est élevé au carré quand la taille de la clé est doublée. Pour casser un système qui utilise une clé de longueur 128 bits, il faut en moyenne 2^{127} (soit $1{,}7 \times 10^{38}$) opérations.

Pour les algorithmes à clé publique, la difficulté augmente plus lentement. L'espace des clés augmente moins vite qu'une loi exponentielle, mais plus vite qu'une loi de puissance ; le doublement de la longueur de la clé n'élève pas au carré l'effort à fournir pour casser le code, mais le travail reste considérable. Par exemple, dans le système RSA, les algorithmes modernes de factorisation peuvent faire beaucoup mieux que d'essayer toutes les combinaisons de nombres premiers possibles pour factoriser un nombre de grande taille. Le code de Diffie et Hellman est, lui aussi, «sous-exponentiel». Par exemple, une clé de 3 000 bits pour RSA ou Diffie-Hellman est aussi difficile à décrypter qu'une clé de 128 bits d'un système de chiffrement par blocs.

Le chiffrement par blocs n'est toutefois pas invulnérable. En 1998, la Société *Electronic Frontier Foundation* a construit, pour moins de 1,5 million de francs, un ordinateur massivement parallèle qui a cassé un message chiffré par DES en explorant toutes les clés à 56 bits en moins d'une semaine.

La force brute n'est pas la seule façon de percer un code. Des méthodes mathématiques élaborées et des outils statistiques raccourcissent le travail. Certaines méthodes recherchent, par exemple, un motif particulier dans le message chiffré. Ces méthodes statistiques sont classées en trois catégories selon ce qui est connu du texte clair et du texte chiffré.

Quand le pirate connaît seulement le texte chiffré, il n'a rien pour le guider dans sa recherche de la clé, si bien que même un système simple de chiffrement résiste aux



2. UNE FONCTION DE HACHAGE crée un condensé de message, semblable à une empreinte digitale utilisée pour détecter les contrefaçons. Le texte d'un message est d'abord converti en chiffres binaires (la lettre A peut être représentée par 00000, la lettre B par 00001, le C par 00010, etc.) ; puis la suite de caractères qui en résulte est découpée en blocs de taille fixe. Chacun des blocs sert alors de clé dans un système de codage. La sortie finale est le condensé du message original. Quelle que soit la longueur du message initial, le condensé a toujours la même taille. C'est une opération à sens unique, car il est quasi impossible de retrouver le message original à partir de son condensé. De surcroît, avec cet algorithme, deux messages donnent presque toujours des condensés différents et il est impossible de trouver un autre message ayant le même condensé : ce condensé est une «empreinte digitale» du message.