Chiffrement et clef de chiffrement

Chiffrer consiste à mélanger de façon plus ou moins inextricable les informations élémentaires d'un texte (lettres, chiffres, bits…). Le chiffre employé par Jules César dans ses correspondances secrètes fournit un exemple simpliste de chiffrement : César se contentait de décaler l'alphabet latin de trois lettres. Son propre nom, CESAR, devenait ainsi FHVDU. Le chiffre de César peut être étendu aux 26 décalages alphabétiques possibles, ce qui explique que des appareils à chiffrer ainsi aient été vendus aux commerçants et aux militaires jusqu'au XIXe siècle *(ci-dessous)*. Il peut aussi être généralisé aux chiffrements polyalphabétiques, qui consistent à substituer les lettres de l'alphabet par d'autres tirées du même alphabet, voire d'un autre. Pendant sa captivité, la reine Marie-Antoinette employa ce type de chiffre *(ci-dessus, l'une de ses lettres)*.

Tout chiffrement implique un algorithme de déchiffrement. Dans le cas du chiffre de César, il suffit de décaler l'alphabet de trois lettres dans le sens opposé pour déchiffrer.

Dans la pratique cryptologique, l'algorithme de (dé)chiffrement est public, de sorte que la sécurité réside dans la clef qui spécifie le chiffrement effectué. Cette clef est le nombre 3 dans le cas du chiffre de César, ce qui est bien trop simple pour résister à des essais de déchiffrement systématiques (la force brute). Pour produire une clef plus difficile à découvrir, on pourra par exemple décaler de 3 lettres la première phrase, de 1 lettre la deuxième, de 4 la troisième, puis de 1, de 5, de 9, de 2, de 6, de 5,… et nommer π la clef résultante. Mais cela serait encore bien trop simple !

Jacques Patarin et Valérie Nachef

Après la robustesse, le deuxième critère à prendre en compte dans le choix d'une primitive cryptographique est la facilité avec laquelle on pourra la mettre en œuvre dans la situation à traiter. Si placer dans un appartement ou une maison particulière un coffre-fort aux parois d'acier de dix centimètres d'épaisseur et pesant cinq tonnes semble une précaution exagérée, ce n'est pas le cas dans une banque. De même, les contraintes qui accompagnent l'emploi d'une primitive cryptographique restreignent ses applications.

Les systèmes de sécurisation des données embarqués dans un téléphone portable, par exemple, sont limités en mémoire et en puissance de calcul, ce qui y interdit les primitives trop complexes ; la situation est différente si les données à protéger ont vocation à être stockées sur un serveur.

Une fois la primitive choisie, il reste à fixer la taille de la clef de chiffrement à utiliser. Une telle clef consiste en une série de bits (de valeurs 0 ou 1) déterminant la façon dont s'opère le cryptage avec un algorithme donné. Dans le cas des algorithmes dits symétriques, la clef spécifie aussi le déchiffrement. Dans celui des algorithmes dits asymétriques, elle ne sert qu'au chiffrement, et une clef supplémentaire est nécessaire au déchiffrement. Quoi qu'il en soit, un principe général prévaut : plus une clef est longue, plus il sera difficile de casser le chiffrement correspondant et le cryptosystème dont elle est le cœur.

L'ESSENTIEL

- Les données stockées ou circulant en masse sur Internet sont protégées par des systèmes de sécurité.
- Les attaques peuvent porter soit sur leur couche cryptographique, soit sur leur couche logique.
- La solidité de la première dépend de la longueur de la clef de chiffrement ; celle de la seconde, des détails de la mise en œuvre du système.
- Les attaquants tentent aussi d'exploiter toute fuite involontaire d'information que présente le système.

Hubert Berberich

Attaques par force brute

Ce principe général s'explique facilement si l'on considère le cas d'un algorithme symétrique public – c'est-à-dire connu et employé par qui veut. Supposons ce mode de cryptage contrôlé par une clef comportant N bits. Pour le casser, il existe toujours, en dernier ressort, la méthode dite de la force brute : elle consiste tout simplement à essayer, avec l'aide d'un ordinateur, toutes les clefs possibles, soit 2^N clefs. Dans le cas d'une clef de 320 bits, on aura donc $2^{320} \simeq 2 \times 10^{96}$ clefs (plus que d'atomes dans l'Univers…) à tester. Face à de tels nombres, la méthode de la force brute paraît bien illusoire. Mais la prudence s'impose : avec l'augmentation incessante de la puissance de calcul des superordinateurs massivement parallèles (à très nombreux processeurs calculant en même temps), l'expérience a montré que l'on a pu casser des primitives d'usage courant, dont les clefs étaient considérées jusque-là comme assez longues pour être sûres…

L'algorithme de cryptage AES *(Advanced Encryption Standard)*, par exemple, s'emploie aujourd'hui avec des clefs de 128, 192 ou 256 bits. Cet algorithme fait subir aux blocs successifs de données un triturage numérique complexe comprenant des permutations, des rotations dans un espace mathématique abstrait, etc., le tout répété plusieurs fois selon la longueur de la clef. En 2011, Christian Rechberger, de l'Université technique de Copenhague, et deux collègues ont publié une méthode d'attaque d'AES-128 (avec clef de 128 bits) environ quatre fois plus rapide que la force brute. Son déroulement implique à peu près 2^{126} (environ 10^{38}) opérations pour trouver la clef. Un ordinateur effectuant dix milliards d'opérations par seconde mettrait de l'ordre de 3×10^{20} années pour faire le calcul, soit plusieurs milliards de fois l'âge de l'Univers : on comprend qu'AES-128 soit encore considéré comme sûr…

Ce cas illustre que plus la clef d'un algorithme est longue, plus il faut de temps

et de ressources de calcul pour le décrypter. La longueur de la clef détermine donc le niveau de sécurité du cryptage.

L'histoire des attaques contre l'algorithme RSA montre l'influence de la longueur de la clef sur la sécurité. Nommé d'après ses inventeurs Ron Rivest, Adi Shamir et Leonard Adleman, RSA est un algorithme asymétrique à clef publique (la clef de cryptage est connue de tous) dont la robustesse découle de la difficulté à décomposer les grands nombres en facteurs premiers. Le cryptage RSA a été utilisé dès les années 1980 avec des clefs de 320 bits par le groupement des banques émettrices de cartes bancaires. Or une version de RSA à clefs de 320 bits a été cassée à la fin des années 1990 par Serge Humpich. Cet ingénieur français a exploité une faille logique du système de sécurité de ce moyen de paiement tel que spécifié en 1983 (la puce contenue dans les cartes vérifie elle-même le code), puis cassé l'instance RSA utilisée dans les cartes bancaires, obtenant à partir de là des cartes fonctionnelles, mais non reliées à un compte bancaire…

Avant 1983, les cryptologues consultés par le groupement des banques émettrices de cartes avaient préconisé une clef d'au moins 640 bits, mais les banques ne les avaient pas suivis. Aujourd'hui, la taille minimale acceptable pour des clefs RSA est de 1 024 bits, tandis que la taille recommandée est de 2 048 bits…

Ce cas et d'autres illustrent que, dans la pratique, il faut non seulement que les clefs des algorithmes d'usage courant soient assez longues, mais aussi qu'on augmente au fil du temps leurs longueurs à mesure que les moyens de calcul deviennent plus puissants *(voir la figure 2)*.

La clef doit être secrète, pas l'algorithme

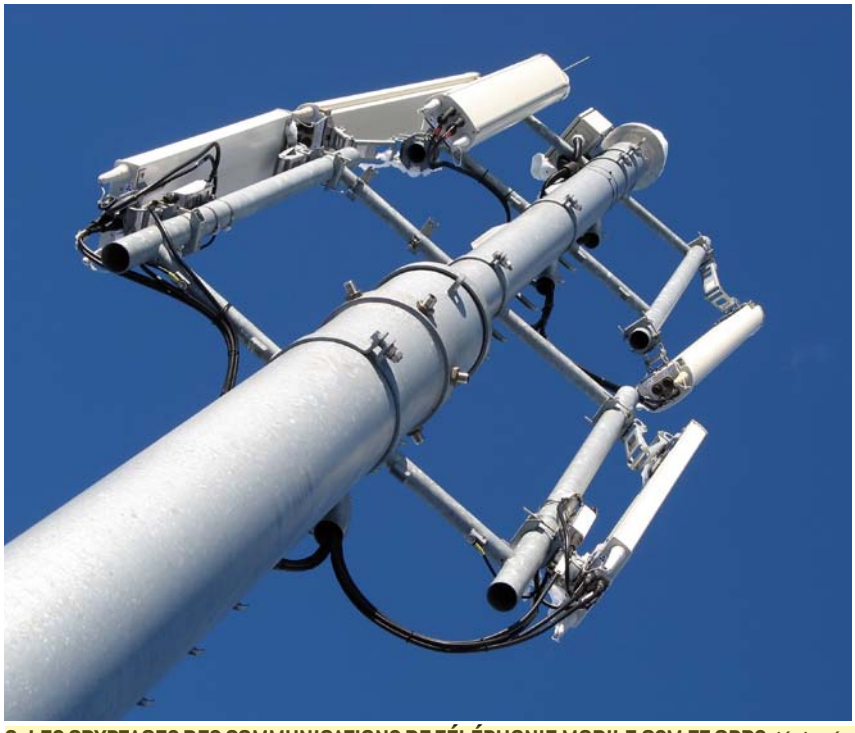
Comment mesurer la sécurité inhérente à un algorithme de chiffrement à un moment donné ? Pour ce faire, le mieux est de le soumettre librement aux attaques. L'expérience a en effet montré qu'il est préférable de rendre publics les algorithmes de cryptage (le fonctionnement du chiffre de César il y a 2 000 ans, ou celui de RSA aujourd'hui), car ces derniers finissent toujours par être connus des attaquants – lesquels ne peuvent parvenir à leurs fins que s'ils réussissent à obtenir la clef par un vol ou par un calcul. C'est pourquoi le cryptologue militaire néerlandais Auguste Kerckhoffs a énoncé en 1883 le principe qui porte son nom : la sécurité d'un système de cryptage ne doit reposer que sur le secret de sa clef. Ainsi, la confiance que les cryptologues accordent à une primitive découle soit de sa résilience aux méthodes de cryptanalyse développées pour en découvrir la clef raisonnablement vite, soit de la difficulté de résoudre les problèmes mathématiques qui lui sont associés ; cette confiance n'a donc rien d'absolu et change avec le temps (en général en diminuant...).

À ce propos, les algorithmes symétriques A5/1 et A5/2 de chiffrement à la volée, utilisés dans les communications hertziennes de téléphonie mobile selon les normes GSM et GPRS, fournissent un intéressant cas d'école. Les spécifications de ces algorithmes fondés sur un générateur de nombres pseudo-aléatoires ont été initialement maintenues secrètes. Mais David Wagner, de l'Université de Californie, et des collègues sont parvenus à les révéler en 1999. L'écoute en temps réel des communications sur téléphones mobiles est rapidement devenue possible, alors que les systèmes A5/1 et A5/2 avaient déjà été déployés sur la planète entière…

AES (chiffrement symétrique)		
Année	Longueur minimale (en bits)	
	Lenstra	ANSSI
2014	78	100
2015	79	100
2017	80	100
2020	82	100
2025	85	128

RSA (chiffrement asymétrique)		
Année	Longueur minimale (en bits)	
	Lenstra	ANSSI
2014	1309	2048
2015	1350	2048
2017	1435	2048
2020	1568	2048
2025	1805	4096

2. CE TABLEAU donne les évolutions recommandées de la longueur des clefs des chiffrements AES et RSA, telles que les ont estimées le cryptologue Arjen Lenstra, de l'École polytechnique fédérale de Lausanne (Suisse), et l'Agence nationale de sécurité des systèmes d'information (ANSSI, France).

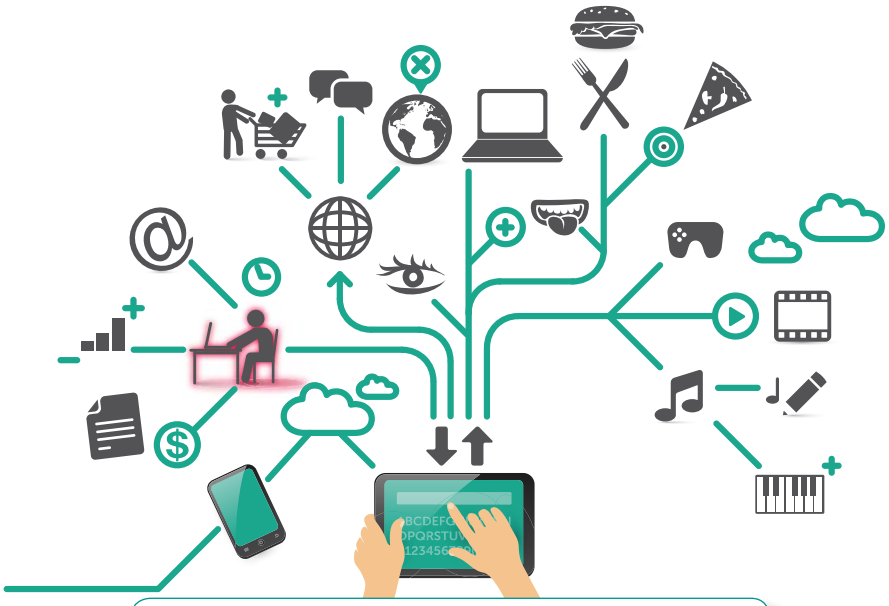


© Fedor Selivanov/shutterstock.com

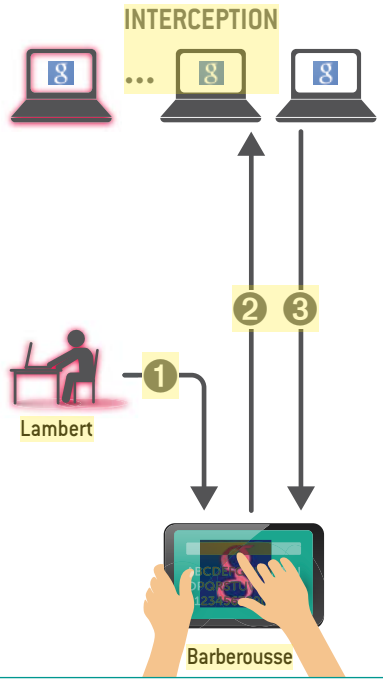
3. LES CRYPTAGES DES COMMUNICATIONS DE TÉLÉPHONIE MOBILE GSM ET GPRS déployés à l'origine *(ici une antenne-relais)* ont été cassés dès 1997. Puis, en 2000, des chercheurs ont prouvé qu'il était même possible de le faire à la volée, de sorte qu'après interception, l'écoute en temps réel des conversations GSM et GPRS sur la planète entière est devenue possible.

UN EXEMPLE DE VOL DE DONNÉES PERSONNELLES SUR INTERNET

L'usurpation d'identité sur Internet prend de nombreuses formes. Nous avons tous reçu des courriels dans lesquels des escrocs nous demandent des données sur notre identité ou nos moyens de paiement en se faisant passer pour l'administrateur de l'un des sites que nous fréquentons (banque, librairie en ligne, etc.). La ficelle grossière de cette escroquerie a son équivalent en temps réel, qui est décrit ci-dessous.



Transactions bancaires, achats de morceaux de musique, etc. : un individu a aujourd'hui de nombreuses activités en ligne et doit pour cela s'identifier souvent. Cette réalité rend la confidentialité sur Internet cruciale pour le fonctionnement de la société ; elle explique aussi que les escrocs font de nombreuses tentatives pour usurper notre identité ou voler nos mots de passe en temps réel.



Levé de bonne humeur, Lambert se connecte ❶ grâce à son compte *Google* à une nouvelle application affichant en ligne les résultats de l'équipe de France de basket. Mais il ignore que ce service a été créé par Barberousse, un escroc qui lui fait faire bien plus que seulement afficher des résultats sportifs... Barberousse utilise l'identité de Lambert pour demander ❷ une connexion à l'une des applications du compte *Google* de Lam-

LES AUTEURS


Stéphanie DELAUNE, chargée de recherche au CNRS, travaille au LSV (Laboratoire de spécification et vérification), le laboratoire d'informatique de l'École normale supérieure de Cachan.

Rémy CHRÉTIEN est doctorant au LSV et au Laboratoire lorrain de recherche en informatique et ses applications (LORIA).

De telles erreurs de conception montrent que, pour éviter ce genre de désastre, d'importants efforts de recherche sont nécessaires avant de déployer un système très utilisé. Il est fréquent que le même algorithme de chiffrement soit employé dans une myriade d'applications différentes, de sorte qu'elles deviennent toutes vulnérables si l'algorithme est attaqué avec succès. Et le nombre énorme de systèmes impliqués ainsi que la complexité inhérente aux primitives cryptographiques rendent impraticables, en général, les corrections à la volée (par exemple par un patch, c'est-à-dire une section de code correctrice).

Une fois le coffre-fort choisi, il importe de maîtriser son expédition, ce qui n'a rien de facile. En effet, une fois effectué le choix des primitives cryptographiques adaptées à la situation, il reste à les mettre en œuvre sans commettre d'erreurs ou de maladresses fatales pour la sécurité des données. La construction d'un cryptosystème s'accompagne en effet de spécifications fixant la façon dont ses composantes échangent des informations, qui sont le plus souvent chiffrées, mais pas toujours. Comme une erreur dans le choix du coffre-fort peut être fatale, un mauvais choix de système de transport (de protocole de communication) annule vite la sécurité apportée par le blindage (par le chiffrement).

Des failles dans la couche logique...

Lorsqu'on traite les problèmes d'organisation et de mise en œuvre d'un système de sécurisation des données, on quitte la couche cryptographique pour entrer dans ce que l'on nomme la couche logique de sécurité. L'emploi d'un algorithme de cryptage dépourvu des faiblesses mentionnées plus haut ne garantit pas l'absence