La protection des informations

Rémy Chrétien et Stéphanie Delaune

Un bon cryptage des données sensibles ne suffit pas à les protéger. Il faut aussi que les protocoles de communication utilisés pour coder ces données soient dépourvus de failles, dans leur logique comme dans leur mise en œuvre.

Le 6 septembre 2013, le monde apprenait avec stupéfaction l'existence de *Bullrun*. Ce programme de l'Agence de sécurité américaine, la NSA *(National Security Agency)*, vise à décoder un grand nombre d'informations cryptées circulant sur Internet. À partir des documents secrets que leur a livrés le lanceur d'alerte Edward Snowden, le *New York Times* et le *Guardian* révélaient qu'une percée technique avait multiplié la puissance de *Bullrun* en 2010, de sorte que les transactions et communications en ligne protégées des banques et de la plupart des grands fournisseurs de service sur Internet étaient désormais accessibles « en grands volumes » aux agences de renseignement *(voir la figure 1)* ; on apprenait aussi que ces agences disposent de superordinateurs pour décrypter par la « force brute » les données qui résistent…

sensibles

© Domaine public

Aujourd'hui, sans vraiment le savoir, nous réalisons tous quotidiennement des opérations de cryptage. Des chiffrements permettent, par exemple, de protéger notre numéro de carte bancaire lors d'un achat en ligne ; on les retrouve dans les protocoles (les procédures de communication) utilisés en téléphonie mobile ou pour lire les passeports électroniques. Pour autant, toutes ces données sensibles, nos données, sont-elles vraiment protégées ? Où sont les points faibles ? Nous allons analyser dans cet article ce qui conditionne la qualité des systèmes de sécurisation des données utilisés de façon routinière.

Sur un réseau, tout se passe comme si les informations protégées ne circulaient qu'à l'intérieur de coffres-forts réalisés dans un matériau solide : le chiffre (le cryptage). Toutefois, quelle que soit la solidité mathématique du chiffrement opéré, encore faut-il que l'assemblage des pièces qui constituent les coffres-forts en circulation ne laisse pas apparaître de fente exploitable par une personne mal intentionnée… Finalement, l'expédition de coffres-forts d'un côté et de leurs clefs de l'autre n'est pas une pratique sans risques, et il arrive qu'on y commette des erreurs, voire qu'on laisse traîner les clefs…

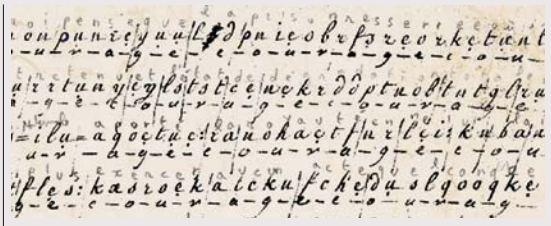
Chiffre et compagnie

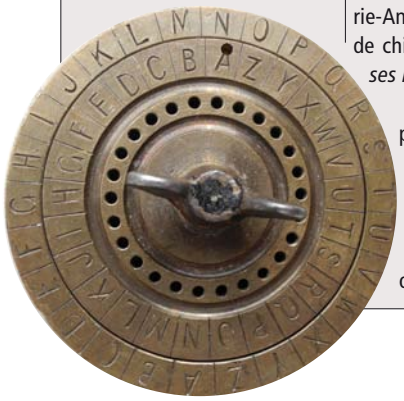
Le chiffrement, nous l'avons dit, est l'analogue du matériau constituant les parois du coffre-fort. Il en existe de nombreux types, chacun caractérisé par des contraintes d'utilisation et des niveaux de sécurité différents. Quoi qu'il en soit, la qualité d'une primitive cryptographique, c'est-à-dire d'un algorithme de cryptage particulier, se mesure à sa robustesse et à sa facilité de mise en œuvre. Tout comme les parois d'un coffre-fort peuvent être plus ou moins épaisses, la robustesse d'une primitive cryptographique peut être plus ou moins importante. Une primitive sera considérée comme faible quand il est facile de découvrir comment inverser son chiffrement, inversion qui permet de révéler les données protégées *(voir l'encadré page suivante)*.

1. CES DÔMES RECOUVRENT LES ANTENNES utilisées (ici au Japon) par le système mondial d'interception des communications privées et publiques des agences de renseignement des États-Unis, du Royaume-Uni, du Canada, de l'Australie et de la Nouvelle-Zélande.

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)*.





Jacques Patarin et Valérie Nachef

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 !

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

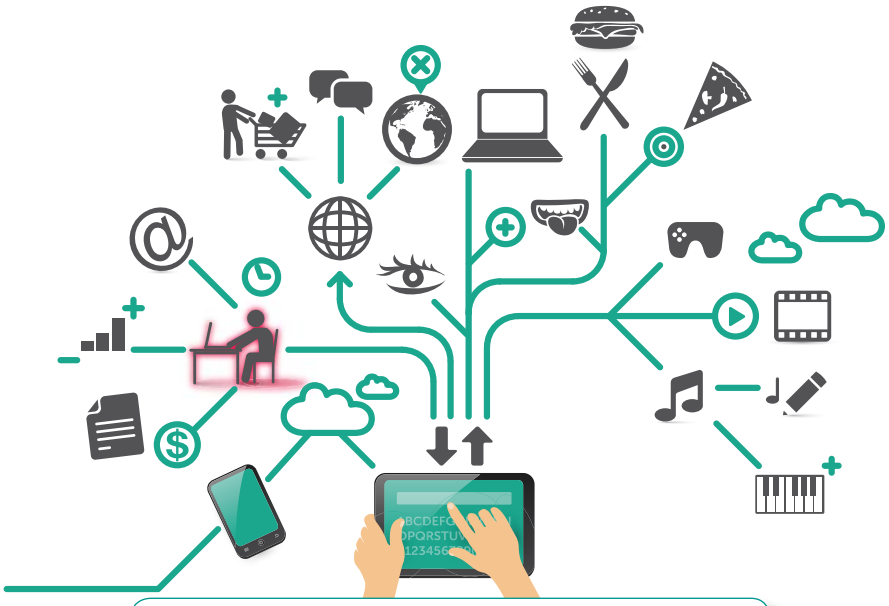
2. CE TABLEAU donne les évolutions recommandées de la longueur des clefs des chiffrements AES et RSA, telles que les ont estimées le cryptologue Arjen Lenstra, de l'École polytechnique fédérale de Lausanne (Suisse), et l'Agence nationale de sécurité des systèmes d'information (ANSSI, France).

© Fedor Selivanov/shutterstock.com

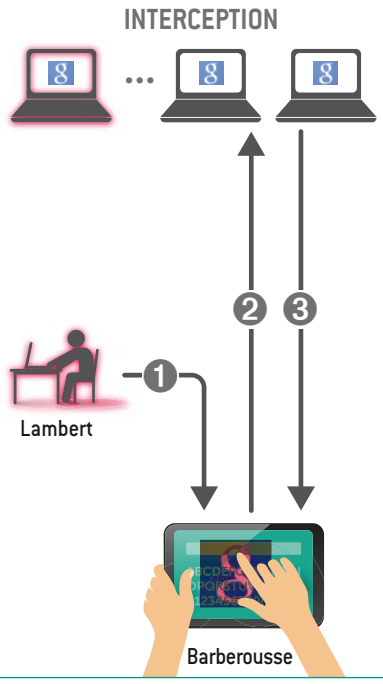
3. LES CRYPTAGES DES COMMUNICATIONS DE TÉLÉPHONIE MOBILE GSM ET GPRS déployés à l'origine *(ici une antenne-relais)* ont été cassés dès 1997. Puis, en 2000, des chercheurs ont prouvé qu'il était même possible de le faire à la volée, de sorte qu'après interception, l'écoute en temps réel des conversations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

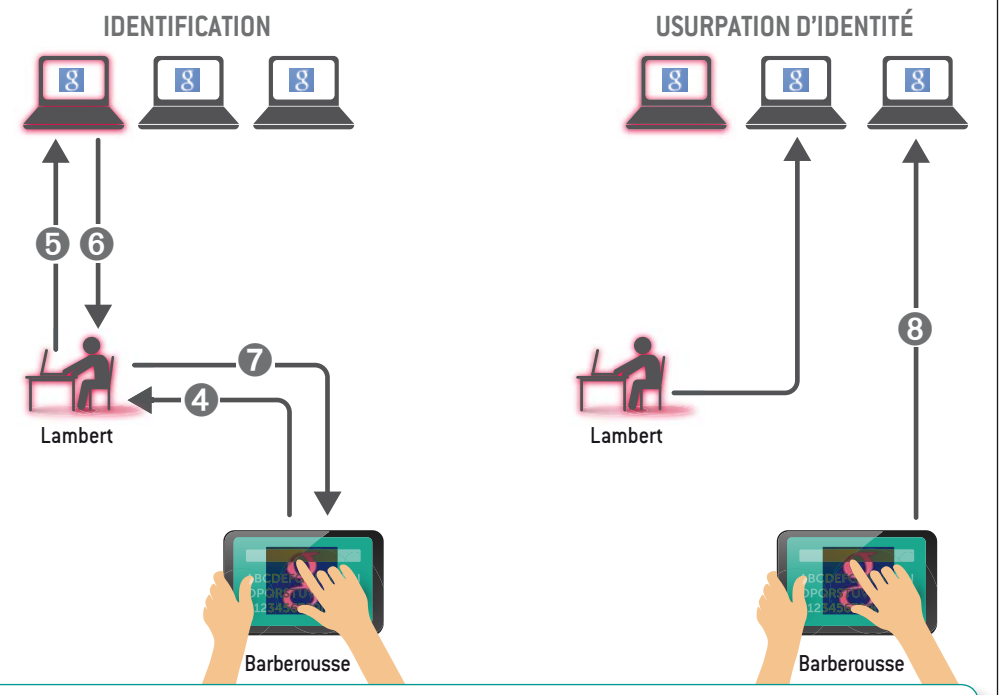
De telles erreurs de conception montrent que, pour éviter ce genre de désastre, d'importants efforts de recherche sont nécessaires avant de déployer un système très utilisé. Il est fréquent que le même algorithme de chiffrement soit employé dans une myriade d'applications différentes, de sorte qu'elles 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

ET D'ACCÈS AUX APPLICATIONS *GOOGLE*



bert. Celle-ci répond par une demande d'identification ❸, que Barberousse répercute à Lambert ❹. Ce dernier s'identifie ❺ auprès du serveur dédié à l'authentification des utilisateurs de *Google*. Ce serveur renvoie ❻ la preuve de l'identité de Lambert à son ordinateur, lequel, conditionné par l'application malveillante de Barberousse, lui communique ladite preuve ❼. Barberousse se sert de cette preuve pour se connecter ❽ par le compte de Lambert à une application *Google*, par exemple l'agenda ou la messagerie de Lambert ; dès lors qu'il s'est connecté avec l'identifiant et le mot de passe de Lambert, ce dernier est aussi simultanément connecté sur l'application sportive, qu'il consulte alors, toujours de bonne humeur, pendant que Barberousse lit son emploi du temps, vole ses secrets ou envoie des messages depuis son ordinateur...

d'erreurs dans la logique présidant à la conception du système.

La première erreur logique possible consiste à envoyer la clef sans la cacher, en d'autres termes à transmettre la clef en clair, ou d'une façon qui permet de la voler. Le vol d'identité résultant d'informations de connexion d'un utilisateur à un service en ligne (une banque en ligne, par exemple) non chiffrées sur Internet fournit un exemple trivial d'une telle erreur logique. Dans ce cas simple, un attaquant n'a qu'à intercepter la communication entre l'usager et son fournisseur de service pour dérober les informations confidentielles permettant de se connecter à la place de l'usager. De telles failles se sont révélées récemment sur des services bien connus du grand public. On peut citer une version ancienne du protocole de connexion des particuliers aux services de *Google* (dit *Google Single Sign-on*), qui a la particularité d'étendre à toutes les applications de *Google* la connexion d'un utilisateur à une seule application.

Ce genre d'attaque permet, par exemple, d'accéder à l'agenda d'une personne après qu'elle a entré son nom d'utilisateur et son mot de passe. Elle repose sur la ruse consistant à pousser l'utilisateur à se connecter à une application malveillante conçue pour l'intéresser, mais secrètement contrôlée par l'attaquant. Il pourra s'agir, par exemple, d'une application affichant les derniers résultats sportifs. L'attaquant intercepte le message envoyé par l'utilisateur pour accéder à cette application. Cette interception est intéressante pour l'attaquant, car ce message a le défaut de ne pas être spécifique à une application donnée, de sorte que l'attaquant peut réutiliser les informations de connexion pour se connecter à d'autres applications depuis le compte *Google* de l'utilisateur, par exemple à sa messagerie, à l'agenda, au service cartographique, à ses archives personnelles, etc. *(voir l'encadré ci-contre)*.

Ce cas illustre qu'en matière de sécurité informatique aussi, le diable est dans les détails, dans des détails de conception infimes, parfois presque invisibles, mais qui peuvent avoir d'importantes répercussions sur la sécurité. Ces failles dans la conception d'un système de protection des données sont d'autant plus graves que la faiblesse qu'elles engendrent affecte des services déjà déployés, et que le déploiement d'un patch ne suffit pas à les réparer. Dans le cas du protocole de connexion aux services de *Google*, changer un message de connexion utilisé par toutes les applications n'est guère réaliste, car cela a des effets à la fois sur la pratique de l'utilisateur et sur l'accessibilité des autres applications. Il est en effet souvent important de maintenir la compatibilité entre les divers protocoles donnant accès à un faisceau d'applications, même si l'un d'eux est défectueux. Une révision complète du système de connexion des services de *Google* a donc été nécessaire pour supprimer la faille.

Que faire pour éviter ce genre de difficulté ? Pour l'essentiel, il s'agit de tester de façon complète, avant de déployer le système de protection, la sûreté des procédures utilisées pour faire circuler les coffres-forts (les informations cryptées) qui protègent les données. Différents modèles théoriques de mise en œuvre existent. Dans le modèle dit symbolique, on procède en supposant d'abord que les primitives cryptographiques choisies sont incassables, puis en représentant par des symboles toutes les opérations effectuées par le système de sécurité. Sur la base d'une définition rigoureuse de la sécurité à obtenir, une technique mathématique rigoureuse permet ensuite de s'assurer que le protocole résiste à toutes les formes d'attaques décrites au sein du modèle utilisé. Une fois cette étape franchie, on peut passer à la mise en œuvre du système de sécurité.

... ou dans l'implémentation

Nous avons maintenant soigneusement choisi des coffres-forts adéquats (des algorithmes de cryptage), organisé scrupuleusement leur transport en nous méfiant de toutes les trahisons volontaires ou involontaires du transporteur (des procédures de communication) et des ruses imaginables

Le passeport biométrique attaqué

Les passeports biométriques français contiennent des puces dites RFID capables de communiquer sans fils. Les données personnelles du voyageur qu'elles contiennent ne deviennent accessibles qu'après que le lecteur de passeport a prouvé son droit de les recevoir en lisant, puis en transmettant à la puce, la clef chiffrée encodée en bas de la première page du passeport. Une fois cette formalité accomplie, le lecteur et le passeport créent une nouvelle clef temporaire pour chiffrer le numéro de passeport, le nom, la date de naissance, la nationalité et la photographie du propriétaire du document.

Toutefois, ce système de sécurité s'est révélé vulnérable. En 2005, il est apparu qu'il était facile de « casser » la clef de sécurité en employant la méthode dite de la force brute. Des failles logiques ont aussi été découvertes dans la conception du système. La première permettait de reconnaître le voyageur en exploitant un message d'erreur envoyé au cours de l'authentification.

En 2010, elle a été corrigée, mais on s'est alors aperçu que la durée exacte du temps de réponse livrait aussi l'identité du voyageur… Plus récemment encore, nous avons montré que la taille et le format des données chiffrées suffisent pour différencier de nombreux passeports, et donc pour les reconnaître dans un contexte particulier. Toutes ces attaques exigent toutefois de savantes mises en œuvre : le contenu des puces RFID de nos passeports est bien mieux protégé que celui de la plupart des puces RFID utilisées au quotidien…

Véronique Cortier,
CNRS-LORIA, Nancy

© Cyril Houshuterstock.com

par les bandits (par les pirates informatiques) ; il nous reste à passer à la pratique (à implémenter le système). Dans cette étape aussi, des failles de sécurité peuvent passer inaperçues. Les spécifications des protocoles n'étant jamais assez exhaustives, il est fréquent que les interprétations qu'en font ceux qui les implémentent confèrent au système certaines particularités, des incompatibilités avec d'autres systèmes, voire des failles de sécurité…

En 2008, on a ainsi découvert une faille dans la version de la suite logicielle *Open*SSH utilisée sur les serveurs *Linux* tournant avec le système d'exploitation *Debian*. Cette suite a été développée par l'équipe du projet *Open*BSD (un projet d'environnement UNIX libre) pour faciliter aux programmeurs l'établissement de connexions sécurisées à partir du protocole SSH (*Secure Shell* en anglais, ou ligne de commande sécurisée). Or il s'est avéré que les paires de clefs RSA de 2 048 bits engendrées à l'aide de la version *Debian* de la bibliothèque d'outils de chiffrement *Open*SSL (utilisée par la suite logicielle *Open*SSH) entre 2006 et 2008 avaient en réalité une robustesse équivalente à celle d'une clef de... 16 bits. De fait, 20 minutes suffisaient pour casser les cryptages SSH pour serveurs *Debian*, et prendre ainsi le contrôle de la machine…Compromises, toutes ces clefs ont dû être changées.

Un autre exemple est celui du passeport électronique français introduit en 2006. Ce passeport incorpore une puce de radio-identification (ou RFID, pour *radio frequency identification*) qui, dans la version biométrique en circulation depuis 2008, contient les photos du détenteur et de deux de ses empreintes digitales. Si les spécifications de l'un des protocoles de communication entre la puce RFID du passeport et la borne de lecture étaient initialement correctes, l'implémentation choisie pour la version française du passeport introduisait une faille. Ses concepteurs avaient prévu qu'en cas d'incident d'authentication du passeport auprès d'une borne, un message d'erreur serait émis pour en préciser la cause.

L'intention était bonne, mais cette circulation supplémentaire d'information s'est révélée suffisante pour qu'un attaquant interceptant les communications puisse identifier le passeport français en voie d'authentification, et ainsi le suivre à la trace à travers le monde *(voir l'encadré ci-contre)*.

Ce type d'attaque ne remet pas en cause la spécification du système et ne concerne que certaines implémentations. Il est donc généralement plus aisé de les bloquer. Bien entendu, le mieux est toujours de les éviter en amont de l'implémentation. Il s'agit, là encore, de tester le système de sécurité en le simulant par un modèle théorique le plus réaliste possible. Malheureusement, tout modèle reste… un modèle, c'est-à-dire un jeu permettant de découvrir des attaques possibles dans un cadre restreint, parce que théorique…

Les failles non encore rendues publiques se monnayent

Notons que les failles dont l'existence n'a pas encore été rendue publique se vendent à prix d'or. Il existe en effet un marché de ce que l'on nomme des vulnérabilités « jour zéro » (le jour 1 étant celui de la publication). Ces attaques restées secrètes sont vendues au plus offrant, le plus souvent à des entreprises ou à des États concernés.

Quand la sécurisation des données repose sur une primitive inviolable, qu'elle a été conçue sans aucune faute logique et implé-

■ BIBLIOGRAPHIE

T. Chothia et V. Smirnov, **A traceability attack against e-passports**, dans *Financial Cryptography*, Springer, vol. 6052, pp. 20-34, 2010.

A. Armando *et al.*, **Formal analysis of SAML 2.0 web browser single sign on : Breaking the SAML-based single sign-on for Google apps.**, 6th ACM Workshop on Formal Methods in Security Engineering (FMSE 2008).

A. K. Lenstra, **Key Lengths, Contribution to the Handbook of Information Security**, Université d'Eindhoven, 2004.

■ SUR LE WEB

Page Internet du LORIA sur le passeport électronique : www.loria.fr/~glondu/epassport/attaque-tailles.html

Cours et séminaires du Collège de France sur la sécurité informatique : www.college-de-france.fr/site/martin-abadi/

mentée sans erreurs, devient-elle parfaitement sûre ? Pas encore. Il reste les attaques indirectes, qui sont comparables à la découverte de la combinaison numérique protégeant un coffre-fort grâce à l'écoute au stéthoscope des bruits de son mécanisme de verrouillage… C'est ce que l'on nomme les attaques par canaux auxiliaires ou canaux cachés.

Ce type d'attaque n'est malheureusement pas l'apanage des films d'espionnage : même un cryptosystème parfaitement réalisé peut être sujet à des fuites d'informations parasites qui, bien analysées et exploitées, suffisent parfois pour le casser. Il arrive ainsi qu'un processeur émette des rayonnemen ts exploitables, qu'il consomme de l'énergie suivant un profil traduisant les tâches de transmissions qu'il est en train d'accomplir, qu'un algorithme de chiffrement ait des temps d'exécution plus ou moins longs selon la valeur de la clef, etc.

De telles fuites ont déjà été utilisées pour casser certaines implémentations du cryptage RSA. Comment ? Les opérations d'élévation à la puissance nécessaires dans cet algorithme peuvent être réalisées soit par multiplication, soit par élévation au carré ; or ces deux opérations induisent des profils de courant électrique différents, lesquels, mesurés et rapportés à leur cause, ont *in fine* permis de déterminer la clef !

Attention aux fuites !

Les failles de ce type sont les plus difficiles à déceler. Les modèles de cryptosystèmes, qui ignorent toute la couche technique sous-jacente au fonctionnement des ordinateurs et des réseaux, sont inutiles pour les repérer. Une fois qu'elles sont connues cependant, il arrive que des contre-mesures simples suffisent à les faire disparaître, par exemple un brouillage supplémentaire de données, une égalisation des signaux ou de la consommation électrique, le blindage des puces pour les rendre opaques à l'espionnage électromagnétique, etc. Autant de défenses qui tiendront jusqu'à ce qu'un autre attaquant découvre encore une fuite insoupçonnée d'information…

La conception et la mise en œuvre d'un cryptosystème impliquent ainsi une science subtile. Cependant, toute la subtilité du monde ne peut rien contre l'utilisateur d'un coffre-fort qui laisse traîner les clefs ou contre la maladresse d'un utilisateur de communications chiffrées qui emploie le même mot de passe trivial (son prénom) pour se connecter au site de son groupe de fans de football, puis à sa banque et ensuite à sa messagerie d'entreprise…

De plus, les révélations du *Guardian* et du *New-York Times* suggèrent que le secret le mieux gardé de l'Internet aujourd'hui serait que les agences de renseignement anglo-saxonnes ont secrètement imposé aux « industries de l'Internet américaines et étrangères » d'insérer des « vulnérabilités dans leurs systèmes de cryptage ». Ces portes dérobées leur assureraient des entrées dans les protocoles de communication sécurisée largement utilisés en ligne, tels HTTPS (le protocole de transfert hypertexte sécurisé), la voix sur IP (téléphonie par Internet) ou encore SSL. Un coffre-fort doté d'une trappe indécelable, mais connue des espions, protégera bien mal nos secrets, c'est clair... ■

La quête d'un support numérique durable

Les disques durs, les CD, les DVD, les mémoires flash... se dégradent en quelques années. Cette situation est largement sous-estimée, mais diverses stratégies et de nouveaux dispositifs sont mis en place.

Franck Laloë et Erich Spitz

Les greniers des vieilles maisons abritent souvent des trésors. Quand on fouille dans des tiroirs ou dans des coffres, il n'est pas rare de retrouver de vieilles photos, des films huit millimètres ou des lettres manuscrites. Aujourd'hui, la plupart des documents, aussi bien les images que les vidéos et les textes, sont enregistrés sur des supports numériques, tels les disques durs des ordinateurs, les CD et les DVD.

À quoi ressembleront les chasses au trésor dans les greniers d'ici 50 ans ? Comment les supports numériques résistent-ils au passage du temps ? Les constructeurs ont annoncé que les supports numériques avaient une longue durée de vie. Les rares études réalisées montrent, au contraire, que ces supports se détériorent en quelques années. Dès lors, se pose la question de la pérennité des données numériques. Comment les conserver ? Ce problème est méconnu du grand public. Pourtant, il touche les documents personnels ayant une valeur sentimentale (les photos et les lettres), mais aussi les informations plus sensibles, telles les données administratives, financières, médicales ou scientifiques.

Aujourd'hui, le numérique est le véhicule dominant de l'information. Il a supplanté les supports analogiques parce qu'il présente de nombreux avantages : une grande densité d'information permettant un gain de place considérable, la facilité des recherches dans l'information stockée, un accès rapide, une circulation quasi instantanée et son universalité – la même clef USB peut contenir des textes, du son, des vidéos, des images et des tableaux de données. La question de la pérennité des données a été reléguée au second plan. Pour l'industrie, la priorité était de développer des dispositifs rapides et contenant des quantités d'informations toujours plus importantes.

La question de la pérennité reste essentielle. Nous aborderons les différents aspects