Cryptographie symétrique

La **cryptographie symétrique**, également dite **à clé secrète** (par opposition à la cryptographie à clé publique), est la plus ancienne forme de chiffrement. On a des traces de son utilisation par les Égyptiens vers 2000 av. J.-C. Plus proche de nous, on peut citer le chiffre de Jules César, dont le ROT13 est une variante.

1 Clé et sécurité

L'un des concepts fondamentaux de la cryptographie symétrique est la *clé*. Une clé est une donnée qui (traitée par un algorithme) permet de chiffrer et de déchiffrer un message. Toutes les méthodes de chiffrement n'utilisent pas de clé. Le ROT13, par exemple, n'a pas de clé. Quiconque découvre qu'un message a été codé avec cet algorithme peut le déchiffrer sans autre information. Une fois l'algorithme découvert, tous les messages chiffrés par lui deviennent lisibles.

Si l'on modifiait le ROT13 en rendant le décalage variable, alors la valeur de ce décalage deviendrait une clé, car il ne serait plus possible de chiffrer et déchiffer sans elle. L'ensemble des clés possibles comporterait alors 25 décalages (26 décalages si l'on considère le décalage nul).

Cet exemple montre le rôle et l'importance de la clé dans un algorithme de chiffrement ; et les restrictions qu'elle implique. Auguste Kerckhoffs (*La cryptographie militaire*, 1883) énonce le principe de Kerckhoffs : pour être sûr, l'algorithme doit pouvoir être divulgué. En outre, il faut aussi que la clé puisse prendre suffisamment de valeurs pour qu'une attaque exhaustive — essai systématique de toutes les clés — soit beaucoup trop longue pour être menée à bien. Cela s'appelle la sécurité *calculatoire*.

Cette sécurité calculatoire s'altère avec le progrès technique, et la puissance croissante des moyens de calcul la fait reculer constamment. Exemple : le DES, devenu obsolète à cause du trop petit nombre de clés qu'il peut utiliser (pourtant 2^{56}). Actuellement, 2^{80} est un strict minimum. À titre indicatif, l'algorithme AES, dernier standard d'algorithme symétrique choisi par l'institut de standardisation américain NIST en décembre 2001, utilise des clés dont la taille est au moins de 128 bits soit 16 octets, autrement dit il y en a 2^{128}. Pour donner un ordre de grandeur sur ce nombre, cela fait environ $3{,}4\times10^{38}$ clés possibles ; l'âge de l'univers étant de 10^{10} années, si on suppose qu'il est possible de tester 1 000 milliards de clés par seconde (soit $3{,}2\times10^{19}$ clés par an), il faudra encore plus d'un milliard de fois l'âge de l'univers. Dans un tel cas, on pourrait raisonnablement penser que notre algorithme est sûr. Toutefois, l'utilisation en parallèle de très nombreux ordinateurs, synchronisés par internet, fragilise la sécurité calculatoire.

Cette notion de sécurité calculatoire pose la question de la sécurité absolue. On sait depuis Claude Shannon et son article *Communication theory of secrecy system* (1949) que le chiffrement de Gilbert Vernam qui consiste à ajouter au message en clair une clé de la même longueur (voir XOR) est parfaitement sûr. C'est le seul pour lequel nous soyons capables de prouver une telle chose. L'inconvénient est que pour chiffrer un message de n bits, il faut au préalable avoir échangé une clé de n bits avec le destinataire du message, et cela par une voie absolument sûre, sinon chiffrer devient inutile. Très peu de cas nécessitent un tel système, mais c'était toutefois le système utilisé pour le Téléphone rouge entre le Kremlin et la Maison-Blanche.

2 Petite taxinomie du chiffrement symétrique classique

Jusqu'aux communications numériques, les systèmes utilisaient l'alphabet et combinaient *substitutions* — les symboles sont changés mais restent à leur place — et *transpositions* — les symboles ne sont pas modifiés mais changent de place.

La substitution est dite *monoalphabétique* quand l'algorithme de codage n'utilise aucun autre paramètre que la lettre à coder, de sorte qu'une lettre est toujours remplacée par la même lettre (relation 1→1). C'est le cas d'un algorithme à décalage simple. Quand l'algorithme de codage utilise un (ou plusieurs) autre(s) paramètre(s) (ex : sa position dans le message), chaque lettre à coder peut alors être remplacée par plusieurs lettres différentes selon les cas (relation 1→n). On parle alors de substitution *polyalphabétique* — e.g. le chiffre de Vigenère, Enigma.

La substitution peut utiliser la méthode du décalage, où chaque lettre est transformée en la lettre n positions plus loin dans l'alphabet, en rebouclant, i.e. la lettre suivant 'z' est 'a'. On parle de décalage simple — est également connu sous le nom de chiffre de Jules César- quand le décalage est identique pour toutes les lettres du message. Avec le chiffre de Blaise de Vigenère, on applique un nombre quelconque n de décalages, le premier décalage est utilisé pour chiffrer la lettre numéro 1, puis la $1+n$,

1+2n, … le second décalage pour la lettre numéro 2, 2+n, 2+2n, … Usuellement, la valeur de ces décalages est donnée par un mot de longueur n dont la i^e lettre donne la valeur du i^e décalage. Clarifions par un exemple.

Message clair : wikipedia Mot clé : crypto Message chiffre : yzixisfzy

Un 'a' dans le mot clé correspond à un décalage de 0, un 'b' à un décalage de 1, etc. Dans notre exemple, la clé a 6 lettres, donc les lettres 1 ('w') et 7 ('d') sont chiffrées par le même décalage, à savoir 2.

La machine Enigma utilisée par les Allemands durant la Seconde Guerre mondiale est également basée sur les substitutions, mais avec un mécanisme beaucoup plus sophistiqué.

Une autre forme de la substitution est le dictionnaire : au lieu de changer les symboles du message un à un, ce sont des mots entiers que l'on remplace.

Pour les transpositions on modifie l'ordre des symboles du texte clair. Une technique consiste à se donner un mot clé, à écrire le message sous ce mot clé et à lire le texte en colonne, par ordre alphabétique.

Message : wikipediaestuneencyclopedielibre Mot clé : crypto on écrit sous wikipe le mot clé diaest uneenc yclope dielib re**** lettre du mot clé (ordre alphabétique) coprty on ordonne les weiipk colonnes dteisa ucenne yeocpl dbliie r**e** Message chiffré : wduydr etceb* ieeol* iincie psnpi* kaele*

Les astérisques sont ajoutés pour le déchiffrement et les espaces dans le message chiffré uniquement pour la lisibilité. Le message, s'il était par exemple envoyé à un destinataire qui connaît le mot clé, serait le suivant :

Message chiffré : wduydretceb*ieeol*iinciepsnpi*kaele*

3 Techniques modernes

Depuis l'avènement du numérique, les paradigmes du chiffrement symétrique ont bien changé. D'une part, la discipline s'est formalisée, même si la conception de système de chiffrement garde inévitablement un aspect artisanal. En effet dans ce domaine, la seule chose que l'on sache prouver est la résistance face à des types d'attaques connues, pour les autres… D'autre part, la forme du texte chiffré ayant changé, les méthodes ont suivi. Les algorithmes modernes chiffrent des suites de bits.

On distingue deux types d'algorithmes, les algorithmes en blocs, qui prennent n bits en entrée et en ressortent n , et les algorithmes à flots, qui chiffrent bit par bit sur le modèle du chiffre de Vernam. Dans ce dernier cas, l'algorithme engendre une suite de bits qui est ajouté (cf. XOR) à la suite binaire à chiffrer. Les techniques utilisées pour générer la suite que l'on ajoute -- appelée la suite chiffrante -- sont diverses. Elles peuvent utiliser des registres à décalage à rétroaction linéaire, composés de façon non linéaire (par exemple A5/1 ou E0, mais pas RC4 qui est ou a été très répandu) ... ou utiliser un chiffrement par bloc en mode avec un mode opératoire adapté.

La seconde famille d'algorithmes, ceux en blocs, est en général construite sur un modèle itératif. Ce modèle utilise une fonction F qui prend une clé k et un message M de n bits. C'est cette fonction F qui est itérée un certain nombre de fois, on parle de nombre de tours. À chaque tour, la clé k utilisée est changée et le message que l'on chiffre est le résultat de l'itération précédente.

$$C_1 = F(k_1, M)\,;$$
$$C_2 = F(k_2, C_1)\,;$$
$$\dots$$
$$C_r = F(k_r, C_{r-1})\,;$$

Les clés k_i utilisées sont déduites d'une clé maître K qui est la quantité secrète que doivent partager émetteur et destinataire. L'algorithme générant ces clés à partir de K est appelé l'algorithme de cadencement de clés.

Pour qu'un tel système puisse fonctionner, la fonction F utilisée doit être une permutation, c'est-à-dire qu'il faut pour toute clé k et message M pouvoir recalculer M à partir de $F(k, M)$, autrement le déchiffrement n'est pas possible et par conséquent on ne dispose pas d'un algorithme utilisable. Formellement, cela signifie qu'il existe une fonction G vérifiant

$$G(k, F(k, M)) = M$$

La sécurité d'un tel système repose essentiellement sur deux points, l'algorithme de cadencement de clé et la robustesse de la fonction F . Si l'algorithme de cadencement est mal conçu, les k_i peuvent être déductibles les unes des autres, ou mal réparties, … Dire de la fonction F qu'elle est robuste signifie qu'on la suppose difficile à inverser sans connaître la clé k ayant servi dans le calcul de $C = F(k, M)$. En d'autres termes, connaissant seulement C , F et G , on ne doit pas pouvoir retrouver le message M , si ce n'est en effectuant une recherche exhaustive de la clé k , c'est-à-dire en calculant

$$X = G(k, C$$

$$Y = F(k, X)$$

et cela pour toutes les clés k jusqu'à ce que l'on en trouve une pour laquelle Y est égal à C . On est alors assuré d'avoir le message M qui n'est autre que X . Le problème étant que si k est constitué de l bits, il faut en moyenne $2^l/2 = 2^{l-1}$ essais. En prenant l assez grand, on peut être sûr que cela n'est pas réalisable en pratique : supposons que l'on puisse essayer 10^9 (un milliard) clés par seconde, soit environ 2^{30}, il y a 31 557 600 secondes par an, soit

2^{25}, en conséquence on peut tester 2^{55} clés par an. Si on prend pour l une valeur de 80 bits, il faudrait 2^{24} ans, plus de 16 millions d'années.

Une technique très répandue pour fabriquer des fonctions F est celle du schéma de Feistel. Dans ce schéma, le message à chiffrer est découpé en 2 blocs de n/2 bits, $M = (L, R)$ et le message chiffré est

$$C = (R, L \oplus f(k, R))$$

où le '$\oplus$' est le XOR et f est une fonction quelconque, on n'a plus à supposer que c'est une permutation. En effet, on peut retrouver M à partir de la clé k

1) connaissant C , on connaît R qui est sa partie gauche,

2) on calcule $f(k, R)$,

3) on ajoute le résultat du calcul précédent à la partie droite de C , et on retrouve L ,

cela sans restriction sur f . Clairement, dans ce schéma, la robustesse de F repose sur la fonction f .

4 Voir aussi

- Cryptologie
- Cryptographie asymétrique
- Cryptographie hybride
- Chiffre de Vigenère
- TAREC
-  Portail de la cryptologie

5 Sources, contributeurs et licences du texte et de l'image

5.1 Texte

- **Cryptographie symétrique** *Source :* https://fr.wikipedia.org/wiki/Cryptographie_sym%C3%A9trique?oldid=114354568 *Contributeurs :* Hashar, Calo, Dtcube, Snark Boojum, Alvaro, Looxix, Orthogaffe, Vincent Ramos, Kelson, Oz, HasharBot, Koyuki, Robbot, Archibald, Sanao, MedBot, Sam Hocevar, Phe-bot, Laurent PELLISSIER, Aither, Mickaël Delahaye, ClementSeveillac, Jef-Infojef, Vincnet, Clemux, Sinar~frwiki, Sador, Sherbrooke, Antoinetav, Gribeco, RobotE, Vazkor, Lmaltier, RobotQuistnix, YurikBot, Eskimbot, Edeluce, Esprit Fugace, Loudubewe, Malost, Manu1400, MetalGearLiquid, Abcd-international, Callyope, Proz, NicoV, Giltir, Schtroumpf neutralisateur, JAnDbot, Ce`dric, Esperanto94, Dfeldmann, Eybot, Salebot, Akeron, DodekBot~frwiki, Smaret, VolkovBot, Chicobot, Ptbotgourou, Félix Potuit, SieBot, OKBot, OuechTonton, BOTarate, ZetudBot, Luckas-bot, GrouchoBot, Mezek, ArthurBot, Cantons-de-l'Est, Ziron, Xqbot, RibotBOT, MastiBot, Tuxien, EmausBot, Mlpoandmlpo, WikitanvirBot, Bdc43, Lmao, Metamorforme42, YFdyh-bot, Razibot, Addbot et Anonyme : 32

5.2 Images

- **Fichier:Crypto_key.png** *Source :* https://upload.wikimedia.org/wikipedia/commons/5/5b/Crypto_key.png *Licence :* LGPL *Contributeurs :* Transféré de fr.wikipedia à Commons. *Artiste d'origine :* Original téléversé par Dake sur Wikipedia français Later versions were uploaded by Croquant at fr.wikipedia.

5.3 Licence du contenu

- Creative Commons Attribution-Share Alike 3.0