Le chiffrement, ou l'apport de la cryptologie à la sécurisation du stockage, de la transmission et du traitement des données

Par Louis GOUBIN
Professeur à l'Université Versailles Saint-Quentin-en-Yvelines – Université Paris-Saclay, directeur du groupe de recherche « Cryptologie et sécurité de l'information » au laboratoire LMV (UMR CNRS 8100)

Le développement des techniques nécessaires au stockage, à la transmission et au traitement des données numériques crée un besoin de plus en plus aigu de sécurisation de ces données. La cryptologie, que l'on appelle souvent la science du secret, apporte des réponses solides, souvent des preuves mathématiques, à la question de la confidentialité des données, et par là même à la protection de la vie privée. Nous proposons ici une excursion à travers la problématique du chiffrement des données, depuis les principes cryptographiques de base jusqu'à des applications plus complexes nécessitant de pouvoir effectuer des calculs sur les données chiffrées. Des perspectives nouvelles s'ouvrent grâce à des techniques récentes ; l'un des grands défis à relever en la matière est d'arriver à articuler sécurité et conformité réglementaire, notamment lorsqu'il s'agit de chaînes d'approvisionnement dans l'industrie ou de plateformes mettant en œuvre une analyse parfois inquiétante des données personnelles de leurs utilisateurs.

Le contexte du chiffrement

« Le système doit être matériellement, sinon mathématiquement, indéchiffrable. Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi. [...] Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exigent pas le concours de plusieurs personnes » (Auguste Kerckhoffs, *La cryptographie militaire*, 1883 [6]).

On peut considérer que les principes édictés par Kerckhoffs à la fin du XIX^{e} siècle marquent la naissance de la cryptologie, au sens moderne du terme. On peut en tout cas les apercevoir en filigrane tout au long du développement de la théorie et de la pratique du chiffrement, qui a accompagné l'invention de techniques de plus en plus cruciales pour le stockage, la transmission et le traitement des données. Le texte qui suit invite à une excursion – forcément trop brève ! – à travers la problématique du chiffrement, depuis les principes cryptographiques de base jusqu'à des applications plus complexes nécessitant de pouvoir effectuer des calculs sur des données chiffrées.

Tout d'abord, intéressons-nous à la cryptologie. Il est maintenant naturel de postuler que le système « puisse sans inconvénient tomber entre les mains de l'ennemi », et donc de considérer des algorithmes cryptographiques connus de tous, les informations secrètes étant concentrées dans une clé. Longtemps limité à la cryptographie symétrique, ce principe – poussé à l'extrême – a donné naissance à la notion de cryptographie asymétrique, sachant que la nature ouverte des systèmes les rend en outre bien plus vulnérables, puisque l'attaquant peut dans certains cas avoir un contrôle complet sur la plateforme d'exécution et l'implémentation logicielle elle-même.

« Il faut qu'il soit portatif » : la cryptographie est de plus en plus déployée dans les applications exécutées sur des périphériques portables, tels que des ordinateurs portables, des tablettes ou des *smartphones*. On ne pourrait trouver plus belle illustration de ce contexte que la carte à microprocesseur (la « carte à puce »[1]). C'est encore le meilleur moyen que l'on ait trouvé pour garder secrète une clé au sein d'un dispositif embarqué, tout en proposant des capacités réelles de calcul cryptographique.

[1] Rendons ici hommage à Michel Ugon, disparu le 28 décembre 2021. Ingénieur de génie et véritable père de la carte à puce, c'est en grande partie lui qui a fait de la France le berceau incontesté de cette technologie qui a inondé le monde.

« Le système doit être matériellement, sinon mathématiquement indéchiffrable » : cette phrase préfigure la théorie de la complexité, qui conduit à identifier des problèmes mathématiques « difficiles », au sens informatique du terme. Ce sont les briques de base à partir desquelles on peut construire de nouveaux cryptosystèmes aux propriétés inédites, notamment pour tenir compte des contraintes de mémoire ou de temps de calcul propres aux dispositifs mis en jeu. Le défi des chercheurs est alors de trouver un compromis acceptable entre le niveau de sécurité et le niveau de flexibilité du chiffrement.

La cryptologie

La cryptographie et la cryptanalyse

Pour analyser le sens du mot « cryptologie », il est utile de se référer à son étymologie, qui s'appuie sur les mots grecs χρυπτός (*kryptos* = caché) et λόγος (*logos* = discours, traité). On peut ainsi définir la cryptologie comme la science du secret : elle s'appuie sur la possibilité de transmettre un message, tout en dissimulant son contenu pour un observateur indiscret.

Plus généralement, dans sa conception moderne, la cryptologie a essentiellement pour objet l'étude de trois problématiques qui en constituent les véritables piliers [15] : la confidentialité, l'authenticité et l'intégrité de l'information. Pour répondre à ces trois exigences, trois concepts importants ont émergé au fil du temps et sont considérés aujourd'hui comme les fonctionnalités fondamentales de la cryptologie :

- le chiffrement, qui permet de cacher l'information contenue dans un message ;
- la signature électronique, qui permet de prouver l'identité de l'auteur d'un message et de garantir la non-répudiation ;
- l'authentification, qui permet de prouver son identité lors d'un contrôle d'accès.

Pour compléter cette typologie, remarquons qu'il est d'usage de distinguer deux facettes de la cryptologie :

- La cryptographie, qui consiste à concevoir et à mettre au point des mécanismes cryptologiques adaptés afin d'assurer une ou plusieurs des trois notions de la sécurité décrites précédemment[2]. Concrètement, ces mécanismes sont en général décrits de façon algorithmique et font appel à des notions mathématiques, allant des probabilités à la théorie des nombres, en passant par la théorie de la complexité, la combinatoire, la théorie des codes correcteurs d'erreurs, la théorie de la réduction des réseaux, les corps finis, les polynômes multivariables, les courbes elliptiques et hyperelliptiques, la géométrie algébrique, etc.
- La cryptanalyse, qui consiste à évaluer la résistance des méthodes mises au point par la cryptographie pour contrer les attaques. Une partie consiste à définir des scénarios d'attaque et à les appliquer pour mettre à l'épreuve les algorithmes. Il peut s'agir d'attaques purement mathématiques ou bien d'attaques faisant, en outre, intervenir la façon dont l'algorithme est implanté dans un système électronique (on parle alors d'attaques physiques).

Le chiffrement symétrique

Dans la cryptographie symétrique, les clés de chiffrement et de déchiffrement sont identiques (voir la Figure 1 ci-dessous). Elles doivent donc toutes deux être gardées secrètes, ce qui fait que l'on parle également ici de cryptographie à clé secrète.

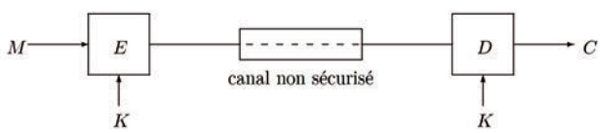

Figure 1 : Le chiffrement symétrique – Schéma général (la clé secrète commune est notée *K*).

C'est le domaine le plus ancien de la cryptographie (il est aussi connu sous le nom de cryptographie conventionnelle). De nombreux algorithmes appartiennent à cette catégorie, dont les deux plus importants actuellement sont évoqués ci-après.

En 1977, le gouvernement américain[3] a publié et standardisé l'algorithme DES (Data Encryption Standard [8,9]), qui a fait l'objet d'un effort de cryptanalyse considérable depuis sa conception. Il est toujours considéré comme excellent : ainsi, même si la taille de la clé (56 bits) s'avère aujourd'hui trop courte, la variante appelée Triple-DES est encore utilisée pour certaines applications, en particulier pour les transactions bancaires.

L'algorithme AES (Advanced Encryption Standard [1, 10]) a progressivement remplacé le DES depuis sa standardisation en 2001. Conçu pour être à la fois plus rapide et plus polyvalent que le Triple-DES, il a fait l'objet d'une grande attention parmi les chercheurs. À ce jour, aucune attaque efficace contre lui n'a été identifiée.

La cryptographie asymétrique

Un nouveau paradigme

La cryptographie asymétrique repose sur le recours à deux clés nécessairement différentes, d'où le qualificatif d'asymétrique. Dans son principe, cette technologie de cryptographie consiste à rendre publique la clé qui sert à la fonction de chiffrement. La clé de déchiffrement doit, quant à elle, rester bien entendu secrète sous peine de perdre complètement tout espoir d'assurer la confidentialité des messages. Cette branche de la cryptologie est souvent appelée également cryptographie à clé publique.

[2] Ainsi que d'autres objectifs plus spécialisés, tels l'anonymat, le *broadcasting*, le *traitor tracing*, etc.

[3] Plus précisément, le National Bureau of Standards (NBS), qui est l'ancêtre du NIST (le National Institute of Standards and Technology).

Remarquons qu'un attaquant disposant du message chiffré C connaît tout à la fois la fonction qui a servi à chiffrer ce message et la clé (publique) qui a été utilisée (voir la Figure 2 ci-après). C'est même encore plus inquiétant, puisqu'en général, il existe une relation connue entre la clé publique et la clé privée ! La situation semble donc paradoxale ; et avec le recul, on s'aperçoit qu'une solution n'a pu émerger qu'avec le développement de la théorie de la complexité, qui rend plausible l'existence de problèmes mathématiques intrinsèquement difficiles. C'est ainsi qu'en théorie (c'est-à-dire en supposant que l'ennemi dispose d'une puissance de calcul infinie), il n'est pas impossible de reconstituer le message M en clair à partir de son équivalent chiffré C. Mais, en pratique, un tel décryptage sous-entend une puissance de calcul que l'on espère suffisamment grande pour dépasser les capacités supposées de tout attaquant.

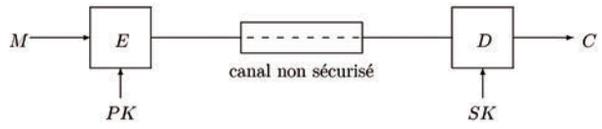

Figure 2 : Le chiffrement asymétrique – Schéma général (la clé publique et la clé privée sont notées respectivement *PK* et *SK*).

Ainsi, ce n'est qu'en 1976, que Whitfield Diffie et Martin Hellman, dans leur célèbre article fondateur [2], ont montré la possibilité théorique de la cryptographie à clé publique, en l'illustrant dans le cas particulier d'un protocole d'échange de clés[4]. L'algorithme RSA est reconnu comme étant le premier algorithme publié[5] de chiffrement asymétrique réellement utilisable.

Par rapport aux systèmes symétriques, le chiffrement à clé publique présente le grand avantage de ne pas nécessiter un accord préalable entre les interlocuteurs qui souhaitent échanger des messages. Mais la plus grande nouveauté apportée par la cryptographie asymétrique est la possibilité de concevoir des protocoles d'authentification et de signature, répondant ainsi de manière spectaculaire aux besoins d'intégrité et d'authenticité.

L'exemple du schéma RSA

L'algorithme RSA, inventé par Ronald Rivest, Adi Shamir et Leonard Adleman [12], a été présenté publiquement pour la première fois dans le numéro d'août 1977 de la revue *Scientific American* [4]. C'est encore actuellement le cryptosystème à clé publique le plus utilisé dans le monde. On le retrouve dans un très grand nombre de produits commerciaux liés à la sécurisation des échanges de données sur Internet, à la protection de la confidentialité et de l'authenticité des courriers électroniques, au paiement électronique au moyen de cartes à puce, etc.

Mathématiquement, on peut décrire l'algorithme RSA de la manière suivante. On commence par choisir l'exposant public e[6]. On utilise ensuite un générateur de nombres aléatoires pour obtenir deux nombres premiers p et q, tels que e soit premier avec p - 1 et avec q - 1. Si l'on pose $n = p \times q$, la clé publique est alors constituée de e et de n, alors que la clé secrète (ou privée) est composée de p et q. La fonction de chiffrement est alors définie par : $f{:}x \rightarrow y = x^e \bmod n$, et la fonction de déchiffrement par : $f^{-1}{:}y \rightarrow x = y^d \bmod n$, où d est une valeur qui doit rester secrète.

La fonction f est donc conçue pour être facilement inversible lorsque l'on connaît la « trappe » d. Casser la fonction RSA consiste à trouver un moyen de calculer $f^{-1}(y)$, alors que l'on ne dispose pas de l'exposant secret d. La seule stratégie d'attaque connue consiste à retrouver p et q à partir de n. De nombreux algorithmes spécialisés ont été inventés pour résoudre ce problème de factorisation. La Figure 3 ci-après illustre la puissance nécessaire[7] pour mettre en œuvre la meilleure méthode connue, en fonction de la taille du modulo n. Empiriquement, on fait généralement l'hypothèse (connue sous le nom de « Loi de Moore ») que la puissance des ordinateurs double tous les 18 mois. On peut alors prédire (s'il n'y a pas de découverte théorique nouvelle concernant les techniques de factorisation[8]) que les clés RSA de 1 024 bits seront cassées vers l'an 2034, et celles de 2 048 bits vers l'an 2079...

Nombre de Mips.ans disponibles	Taille maximale des clés RSA «factorisables»
$4,8 \times 10^{-2}$	251
$4,9 \times 10^{-1}$	292
$4,9$	337
$5,0 \times 10^{1}$	385
$5,0 \times 10^{2}$	438
$5,0 \times 10^{3}$	494
$\mathbf{1,0 \times 10^{4}}$	512
$5,1 \times 10^{4}$	555
$5,1 \times 10^{5}$	620
10^{8}	784
10^{11}	1035
10^{16}	1551
10^{20}	2057

Figure 3 : Puissances nécessaires pour factoriser le modulo RSA.

(4) Notons qu'un protocole présentant les mêmes propriétés avait été décrit un peu auparavant par Ralph Merkle [7]. Les *puzzles de Merkle* affichaient toutefois un moins bon rapport sécurité/performance.

(5) On sait aujourd'hui [3] qu'au sein du service du chiffre britannique, James Ellis avait établi, dès janvier 1970, la possibilité de la cryptographie asymétrique. Par la suite, Clifford Cocks a inventé en 1973 une variante du RSA, avant que Malcolm Williamson décrive à son tour, en 1974, une variante du futur protocole d'échange de clés Diffie-Hellman.

(6) Des exemples courants sont e = 3, e = 17, e = 257 ou e = 65537.

(7) Mips signifie « million d'instructions élémentaires par seconde ». 1 Mips.an représente le nombre d'instructions élémentaires exécutées par une machine qui en exécute 1 million par seconde, et que l'on fait tourner pendant 1 an. Ainsi, 1 Mips.an équivaut plus ou moins à $31{,}5.10^{12}$ instructions élémentaires.

(8) Ni de saut technologique tel que l'apparition d'ordinateurs quantiques qui pourraient factoriser efficacement, comme l'a montré Peter Shor en 1994 [14].

Utilisation du RSA pour le chiffrement

Même si la fonction RSA est solide, la façon dont on l'utilise pour obtenir un cryptosystème capable d'effectuer du chiffrement n'est pas neutre. De manière générale, il doit non seulement être impossible, en pratique, de retrouver un message clair à partir de son équivalent chiffré – hormis, bien entendu, pour l'utilisateur légitime du système –, mais il doit également être impossible de connaître la moindre information sur le message clair (ou sur la signature). C'est ce que l'on appelle la sécurité sémantique.

Il est facile de comprendre que la fonction RSA n'est pas suffisante à elle seule pour garantir cette sécurité. La fonction f jouit en effet d'une propriété de multiplicativité : $f(x.y) = f(x).f(y)$, ce qui ouvre la voie à certaines attaques comme celle élaborée par Johan Håstad en 1988, lequel a montré qu'un attaquant peut toujours retrouver le message M, si celui-ci est envoyé à un nombre suffisant de destinataires utilisant le même exposant public e.

Il faut par conséquent faire très attention à la manière dont on applique la fonction RSA pour chiffrer un message M. Dans la pratique, on commence par « formater » le message M au moyen d'une transformation φ qui fait intervenir un nombre aléatoire. Ainsi, le message chiffré est obtenu sous la forme : $C = f(\varphi(M,r))$, où r est une valeur aléatoire. Pour déchiffrer, on calcule $\varphi(M,r) = f^{-1}(C)$, où φ a été conçue pour qu'il soit facile de retrouver M à partir de $\varphi(M,r)$. La théorie des protocoles de chiffrement RSA est aujourd'hui bien maîtrisée, au point que l'on connaît aujourd'hui des transformations φ qui sont prouvées comme étant « saines ». En d'autres termes, si la fonction f est mathématiquement solide, et que l'on utilise l'une de ces « bonnes » transformations de φ, alors il n'existe pas d'attaque mathématique plus simple que celle consistant à trouver un moyen d'inverser la fonction f[9].

Réaliser des calculs sur des données chiffrées ?

Le chiffrement homomorphe

Les données que l'on considère sont, en général, dans l'un des trois états suivants : au repos, en transit ou en cours d'utilisation. Les algorithmes de chiffrement évoqués jusqu'ici se placent dans les deux premiers cas : les données au repos ou en transit ne changent pas activement ; elles ont la même valeur lorsqu'on les déchiffre que quand on les a chiffrées.

En revanche, les données en cours d'utilisation n'ont pas cette propriété. Presque toutes les opérations que l'on peut envisager sur les messages chiffrés modifient la valeur du message correspondant converti en clair. Il est dès lors difficile de s'assurer que ce message en clair change de la « bonne manière ». Habituellement, il s'agit même d'une exigence de sécurité pour les algorithmes de chiffrement, qui sont conçus pour détruire toute relation entre le message en clair et le message chiffré correspondant.

Inversement, la possibilité d'effectuer des opérations mathématiques sur des données chiffrées implique qu'il doit exister une relation entre les messages en clair et les messages chiffrés. Il doit être possible d'ajouter deux messages chiffrés ou de les multiplier et d'avoir le même résultat que celui chiffré obtenu en effectuant la même opération (addition ou multiplication) sur les deux messages en clair correspondants. En outre, une telle opération doit être réalisée de façon à rester cachée pour un observateur, au sens où l'observation de telles opérations mathématiques réalisées sur des messages chiffrés ne doit rien révéler sur les messages en clair correspondants.

Du point de vue cryptographique, il s'agit de construire un schéma de chiffrement complètement homomorphe (en anglais, *Fully Homomorphic Encryption*, FHE), c'est-à-dire permettant à tout utilisateur de calculer, à partir d'un ensemble de données chiffrées : $c_1, \ldots, c_n$ (correspondant à des données en clair : $m_1, \ldots, m_n$), une donnée chiffrée c correspondant à une certaine fonction $F(m_1, \ldots, m_n)$ des données en clair, sans que cet utilisateur connaisse ces données elles-mêmes (voir la Figure 4 de la page suivante). Bien que Ronald Rivest, Leonard Adleman et Michael Dertouzos aient conjecturé l'existence d'un tel schéma dès 1978 [11], il a fallu attendre 2009 pour que Craig Gentry [5] propose la première solution convaincante[10]. D'autres solutions sont alors apparues rapidement, tirant parti de la notion mathématique de réseaux euclidiens, qui avaient déjà fait leurs preuves pour la conception de nombreuses primitives cryptographiques[11].

Il est toutefois à noter que la tension entre les conditions techniques de ces constructions[12] et la sécurité des problèmes difficiles sous-jacents, oblige à choisir des paramètres très grands pour dimensionner le système, ce qui résulte en des schémas de chiffrement d'une grande complexité[13]. Le défi – encore actuellement – est d'améliorer ces schémas[14] pour les rendre pratiques, tout en restant sûrs.

[9] Précisons que ces résultats sont vrais sous l'hypothèse que les fonctions de hachage utilisées dans la définition de φ sont elles-mêmes, dans un certain sens, « parfaites » (c'est ce que l'on appelle le *modèle de l'oracle aléatoire*).

[10] Remarquons que si l'on se limite à un seul type d'opérations à réaliser sur les chiffrés (on parle de chiffrement partiellement homomorphe), des solutions étaient déjà bien connues, à commencer par la fonction RSA qui est homomorphe pour la multiplication.

[11] Et qui sont à la base – comme les systèmes de polynômes multivariés, les codes correcteurs d'erreurs et les isogénies de courbes elliptiques – de nouveaux algorithmes dits post-quantiques, c'est-à-dire résistants face à d'éventuelles attaques menées par des individus équipés d'ordinateurs quantiques.

[12] En particulier, ceux liés à l'opération de *bootstrapping* introduite par Craig Gentry.

[13] Par exemple, dans la version de la bibliothèque de calcul HElib C++ publiée en 2018 par IBM, les opérations portant sur les chiffrés sont environ 1 million de fois plus lentes que les mêmes opérations réalisées sur les textes en clair correspondants. Un calcul qui prendrait une seconde dans le cas de textes en clair prendrait en moyenne 11,5 jours pour être réalisé en recourant à la version 2018 de HElib.

[14] La bibliothèque de calcul HElib d'IBM a ainsi gagné un facteur 100 millions entre 2015 et 2018.

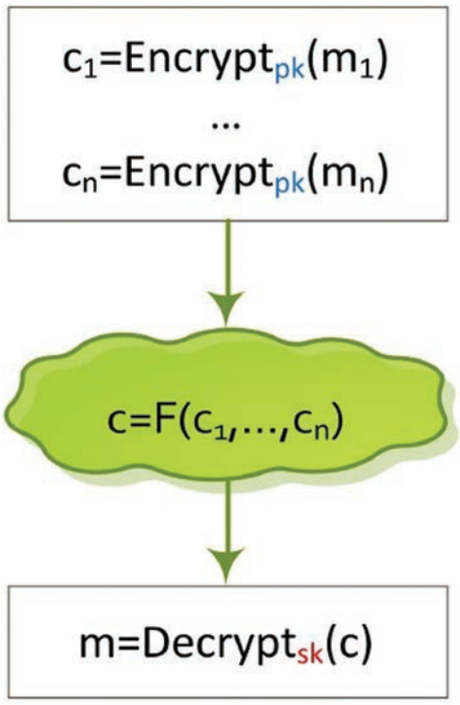

Figure 4 : Principe du chiffrement complètement homomorphe (FHE).

Vers de nouvelles applications

Il est facile de voir qu'un tel schéma peut servir à construire des protocoles respectant la vie privée (*privacy-preserving*) : un utilisateur peut ainsi stocker des données chiffrées sur un serveur[15] et autoriser ce même serveur à effectuer des opérations sur ces données, sans avoir à lui révéler la teneur même de ces données.

Plus concrètement, cette capacité à pouvoir effectuer un traitement sur des données chiffrées a pour potentiel de résoudre de nombreux défis commerciaux majeurs auxquels sont confrontées les entreprises de tous les secteurs.

La plupart des entreprises font appel à des tiers de confiance dans le cadre de l'exercice de leurs activités. Ces sous-traitants, fournisseurs, etc. ont souvent besoin d'accéder aux données sensibles et exclusives de l'entreprise pour pouvoir faire leur travail. Des événements récents ont illustré les risques liés à des chaînes d'approvisionnement non sécurisées et montré comment les cybercriminels ciblent le maillon le plus faible de celles-ci pour atteindre leurs objectifs. Cela signifie pour une organisation que confier ses données sensibles à un partenaire peut l'exposer à des vols de données qui peuvent être pour elle coûteux et dommageables. Le chiffrement homomorphe peut l'aider à se protéger contre ces risques liés à des failles de sécurité dans la chaîne d'approvisionnement : si toutes les données fournies à un tiers de confiance pour opérer un traitement sont chiffrées, un vol de ces données ne présente dès lors qu'un risque minime pour l'entreprise. Cela permet à une organisation d'externaliser le traitement de ses données critiques avec un risque minimal.

Ces dernières années, le paysage réglementaire de la protection des données est devenu de plus en plus complexe. De nouvelles réglementations, telles que le Règlement européen sur la protection des données (RGPD), ont accordé aux personnes concernées de nouveaux droits et imposé des responsabilités et des restrictions supplémentaires aux entreprises. Une règle du RGPD avec laquelle de nombreuses entreprises sont aux prises est l'exigence que les données des citoyens de l'Union européenne (UE) restent au sein de l'UE ou ne puissent être utilisées que dans des pays ou des entreprises dont les normes de sécurité des données sont équivalentes à celles de l'UE. L'arrêt Schrems II de 2020 [13] a invalidé l'un des principaux moyens par lesquels les flux de données entre l'UE et les États-Unis étaient justifiés dans le cadre du RGPD, ce qui a causé d'importants problèmes à de nombreuses entreprises américaines comptant parmi leurs clients des citoyens de l'UE. Des lois comme le RGPD stipulent clairement que leurs exigences ne s'appliquent pas aux données chiffrées. Avec le chiffrement homomorphe, une entreprise pourrait potentiellement stocker et traiter des données en recourant à des systèmes se trouvant en dehors de l'UE, puis les déchiffrer en faisant appel uniquement à des serveurs situés dans des espaces géographiques répondant aux exigences du RGPD.

L'analyse de données est pour de nombreuses entreprises une façon pour elles de générer des revenus. Si des entreprises sont en mesure de fournir des services « gratuits », c'est parce qu'elles collectent des informations sur leurs utilisateurs, qu'elles traitent celles-ci et les vendent à des tiers à des fins de publicité ciblée. Cependant, cette monétisation des données personnelles est controversée. De nombreuses personnes sont mécontentes des pratiques de ces entreprises qui conduisent à créer des profils détaillés les concernant sans qu'elles n'aient de visibilité ni de contrôle sur les données collectées et sur la manière dont elles sont utilisées. Le chiffrement homomorphe fournit une solution potentielle à ce problème : une entreprise pourrait effectuer les analyses de données dont elle a besoin, sans avoir la possibilité de visualiser les données d'origine ou même d'y accéder. Si les clés de chiffrement sont contrôlées par les utilisateurs, alors cela ouvre la possibilité d'une publicité qui soit à la fois privée et ciblée.

Bibliographie

[1] DAEMEN J. & RIJMEN V., *AES proposal: Rijndael,* https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf

[2] DIFFIE W. & HELLMAN M. E. (1976), "New Directions in Cryptography", *IEEE Transactions on Information Theory*, vol. IT-22, pp. 644-654.

[3] ELLIS J., "The story of non-secret encryption", article écrit en 1977 et publié après la mort de l'auteur en 1997, https://cryptome.org/jya/ellisdoc.htm

[4] GARDNER M. (1977), "A new kind of cipher that would take millions of years to break", *Scientific American*, août, pp. 120-124.

[5] GENTRY C. (2009), "Fully homomorphic encryption using ideal lattices", in *Proc. of STOC*, pp. 169-178.

[15] Typiquement dans le contexte du *cloud computing* (ou « informatique en nuage »).

[6] KERCKHOFFS A. (1883), « La cryptographie militaire », *Journal des sciences militaires*, vol. IX, janvier, pp. 5-38, et février, pp. 161-191.

[7] MERKLE R. C. (1978), "Secure Communication over Insecure Channels", *CACM*, vol. 21, n°4, pp. 294-299.

[8] NATIONAL BUREAU OF STANDARDS (NBS), Data Encryption Standard (DES), Federal Information Processing Standards Publication (FIPS PUB) 46, Washington, DC, 1977.

[9] NATIONAL BUREAU OF STANDARDS (NBS), Data Encryption Standard (DES), Federal Information Processing Standards Publication (FIPS PUB) 46-3, Gaithersburg MD, 1999, http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf

[10] NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST), Advanced Encryption Standard, Federal Information Processing Standards Publication (FIPS PUB) 197, décembre 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[11] RIVEST R. L., ADLEMAN L. M. & DERTOUZOS M. L. (1978), *On Data Banks and Privacy Homomorphisms. In Foundations of Secure Computation*, Academia Press.

[12] RIVEST R. L., SHAMIR A. & ADLEMAN L. M. (1978), "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, vol. 21, n°2, pp. 120-126.

[13] Schrems II – Arrêt rendu par la Cour de justice de l'Union européenne dans l'affaire C-311/18 – Data Protection Commissioner contre Facebook Ireland Ltd et Maximillian Schrems, 16 juillet 2020, https://curia.europa.eu/juris/document/document.jsf?text=&docid=228677

[14] SHOR P. W. (1997), "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", *SIAM Journal on Computing* 26, pp. 1484-1509.

[15] STERN J. (1998), *La Science du secret*, Odile Jacob.