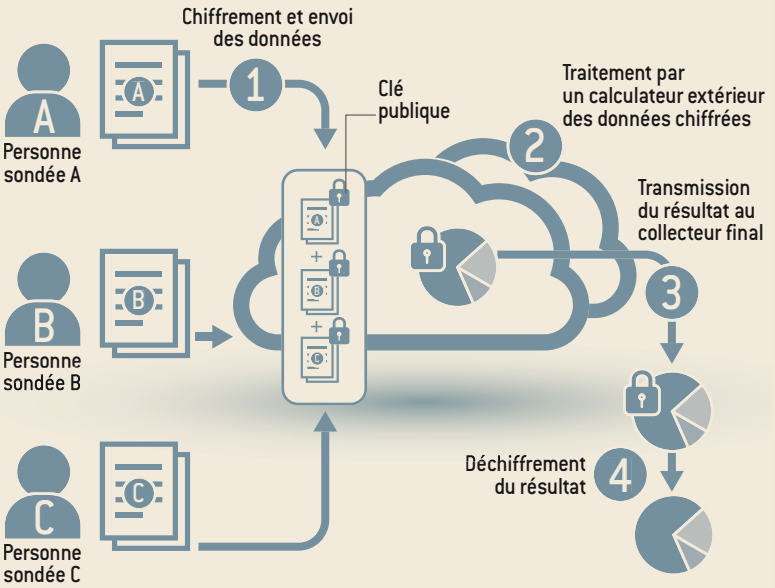
PRÉSERVER LA CONFIDENTIALITÉ

Imaginons que l'on souhaite établir une statistique sur un sujet délicat en utilisant Internet et en garantissant à chacune des personnes sondées que les données individuelles contenues dans les réponses ne seront pas divulguées. La procédure est la suivante. Chaque personne chiffre ses réponses avec la clé publique du collecteur final et les envoie à un calculateur extérieur (1). Ce calculateur traite ces données chiffrées et envoie le résultat au collecteur final (2, 3). Ce dernier déchiffre le résultat et obtient la statistique souhaitée (4). Le calculateur extérieur ne peut pas accéder aux informations privées des sujets de l'enquête, car il ne dispose pas de la clé privée de déchiffrement, et le collecteur final ne peut pas accéder aux données individuelles car il n'a pas eu accès aux informations chiffrées individuelles transmises.



expression, aussi compliquée soit-elle, constituée d'additions et de multiplications. Puisque dès qu'on sait mener des additions et des multiplications de cette façon, on peut exécuter tout algorithme, le problème des calculs secrets par un tiers se ramène à celui de trouver $\mathscr{C}$, $\mathscr{C}'$, $*_1$, $*_2$, vérifiant les propriétés 1 et 2.

Les méthodes de chiffrement vérifiant les propriétés 1 et 2 sont dites «pleinement homomorphes». Le terme vient du langage mathématique où les homomorphismes sont des fonctions qui transforment une opération en une autre, comme l'indique par exemple la propriété 1.

Ces homomorphismes jouent un rôle important en mathématiques et chacun connaît le logarithme qui vérifie $\log(a\times b)=\log(a)+\log(b)$ et qui transforme donc des multiplications en additions (ici, $*_2$ est l'addition habituelle). Avant l'avènement des ordinateurs, c'était le moyen le plus efficace pour mener à la main des calculs numériques, et c'est pourquoi on dressait des tables de logarithmes contenant des millions de données.

En mathématiques, on connaît de nombreux homomorphismes, mais aucun des homomorphismes classiques ne peut servir à concevoir un chiffrement $\mathscr{C}$ homomorphe car l'opération de déchiffrement $\mathscr{C}'$ est alors trop simple ; or une bonne méthode cryptographique est une méthode où il est difficile de trouver $\mathscr{C}'$. Le problème des systèmes de chiffrement pleinement homomorphes n'était donc pas évident au moment où, en 1978, l'idée en a été envisagée par les Américains Ron Rivest et Leonard Adleman et le Grec Michael Dertouzos. On s'est longtemps demandé si de telles méthodes existaient. Le problème a parfois été qualifié de graal de la cryptographie.

UNE PERCÉE DÉCISIVE

Une percée décisive a été réalisée en 2009 par Craig Gentry, de l'université Stanford et aujourd'hui au centre Thomas Watson d'IBM à New York. Craig Gentry a proposé une méthode qui résout théoriquement le problème et a pour cela reçu une bourse de 625 000 dollars de la fondation MacArthur. La méthode qu'il a initialement proposée n'est pas applicable, car elle exige des calculs trop volumineux au moment du chiffrement (la transformation des données avant de les envoyer) et du déchiffrement. De plus, la taille des données que doit manipuler l'opérateur extérieur $\mathbb{O}$, et donc la quantité de calculs qu'il doit mener, est des millions de fois celle nécessaire quand les données ne sont pas chiffrées.

Depuis 2009, une multitude de recherches ont été faites pour trouver des schémas plus économiques. En quelques années, les progrès ont rendu envisageables certaines applications, et l'espoir de méthodes encore plus économiques et faciles à mettre en œuvre suscite de nombreux travaux parmi les spécialistes.

Avant de donner des précisions sur les idées de la percée décisive de Craig Gentry, revenons sur les applications des chiffrements pleinement homomorphes ou même partiellement homomorphes.

L'algorithme RSA (conçu en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman) est une méthode de chiffrement à clé publique : pour chiffrer un message, on utilise la clé publique de >

CRAIG GENTRY, LE PIONNIER

© John D. & Catherine T. MacArthur Foundation

En proposant en 2009 le premier système de cryptage complètement homomorphe, Craig Gentry, aujourd'hui chercheur chez IBM, a révolutionné le domaine. Il a déclenché une multitude de travaux qui, petit à petit, ouvrent des voies pour rendre utilisable son idée, initialement théorique pour l'essentiel. Confier de façon massive des calculs à des opérateurs extérieurs qui ne peuvent pas accéder aux données n'est pas aujourd'hui envisageable. En revanche, peu à peu, il est devenu possible de résoudre des problèmes particuliers n'exigeant qu'une petite quantité de calculs homomorphes (comme pour un vote). Les applications de ces méthodes devraient rapidement s'étendre et aider à accroître la sécurité des réseaux et la protection des données... ce dont on a un grand besoin !

> son correspondant, et lui seul peut déchiffrer le message grâce à sa clé secrète. C'est un bon algorithme, aujourd'hui très utilisé. Il a la propriété d'être homomorphe pour la multiplication: on en déduit une méthode de chiffrement homomorphe pour l'addition, comme celle évoquée au début de l'article, méthode ayant cette fois l'avantage que les données sont correctement masquées à l'opérateur extérieur.

LES VOTES SECRETS

Avec un système additivement homomorphe de ce type, on peut opérer des votes secrets sur le réseau sans se rencontrer. Décrivons un tel procédé.

Il y a trois types d'intervenants: les votants (au plus $N-1$), l'opérateur calculateur, et l'opérateur de dépouillement. L'opérateur de dépouillement choisit un couple clé publique/clé privée et communique la clé publique à chaque votant. Chaque votant choisit alors un nombre entier V de la forme Nk (k entier) s'il veut voter OUI, ou $Nk+1$ (k entier) s'il veut voter NON; puis il chiffre le nombre V choisi avec la clé publique et transmet le nombre chiffré $\mathscr{C}(V)$ à l'opérateur de calcul. Ce dernier fait la somme S de tous les nombres qu'il reçoit et communique S à l'opérateur de dépouillement, lequel déchiffre le résultat grâce à la clé privée, qu'il est seul à connaître. Il trouve ainsi un nombre de la forme $Mk+p$, où p est le nombre de personnes ayant voté NON. Le résultat du vote est alors communiqué à tous.

L'opérateur calculateur, qui ne connaît pas la clé de déchiffrement, ne peut pas connaître les votes individuels; l'opérateur de dépouillement, qui n'a pas eu connaissance des nombres transmis individuellement, ne peut pas lui non plus connaître les votes individuels. Les votants peuvent contrôler le travail de l'opérateur de calcul, s'ils acceptent de se communiquer les $\mathscr{C}(V)$, ce qui ne compromet pas le secret du vote.

Avec un tel système, même si les votants sont séparés par des milliers de kilomètres et ne communiquent que par messages, aucun vote ne circule de manière lisible, et personne ne sait comment les autres ont voté. On peut encore perfectionner de tels systèmes, comme l'expliquent Véronique Cortier et Steve Kremer dans un article de 2013.

Ce type d'applications de la cryptographie partiellement homomorphe est aujourd'hui arrivé à maturité et le système de vote Helios est un logiciel libre développé par les chercheurs de l'université Harvard et de l'université catholique de Louvain. On l'a utilisé pour l'élection du recteur de l'université catholique de Louvain et plusieurs fois lors d'élections étudiantes. L'Association internationale des chercheurs en cryptographie (IACR) l'utilise pour choisir les membres de son bureau. Il est fondé sur le système de chiffrement El Gamal, qui est additivement homomorphe.

UNE ARITHMÉTIQUE « SECRÈTE »

Voici maintenant la description d'une version du chiffrement pleinement homomorphe de Craig Gentry. Cette version est due à Marten van Dijk, Craig Gentry, Shai Halevi et Vinod Vaikuntanathan. Les idées sont simples et n'impliquent que des calculs élémentaires effectués avec des entiers.

La méthode utilise le fait que, pour créer un système de chiffrement pleinement homomorphe, il suffit de définir un système qui chiffre les bits, des 0 et des 1, et de savoir faire exécuter par l'opérateur extérieur un nombre quelconque d'opérations logiques du type XOR (OU exclusif) et ET entre de tels bits chiffrés.

La clé secrète du système, que l'utilisateur doit garder pour lui, comme le C de l'exemple du début de l'article, est ici un long nombre entier impair p, mettons de 500 chiffres binaires. Dans toute la description de la méthode, la taille des entiers manipulés est importante; les opérations ne sont pas faisables à la main, mais ce n'est aujourd'hui pas un problème, puisque n'importe quel téléphone peut manipuler des nombres de plusieurs millions de chiffres.

Un multiple «approximatif» de p est un nombre de la forme $qp+e$, où e est un nombre d'une vingtaine de chiffres binaires introduisant une sorte de bruit. Retrouver p à partir de la donnée d'un ou plusieurs multiples approximatifs de p nécessite des calculs si longs qu'ils sont impossibles à réaliser pratiquement en un temps raisonnable. C'est sur cette impossibilité pratique que se fonde le chiffrement pleinement homomorphe que nous décrivons. Si q comporte beaucoup de chiffres (par exemple plusieurs millions de chiffres binaires), un attaquant