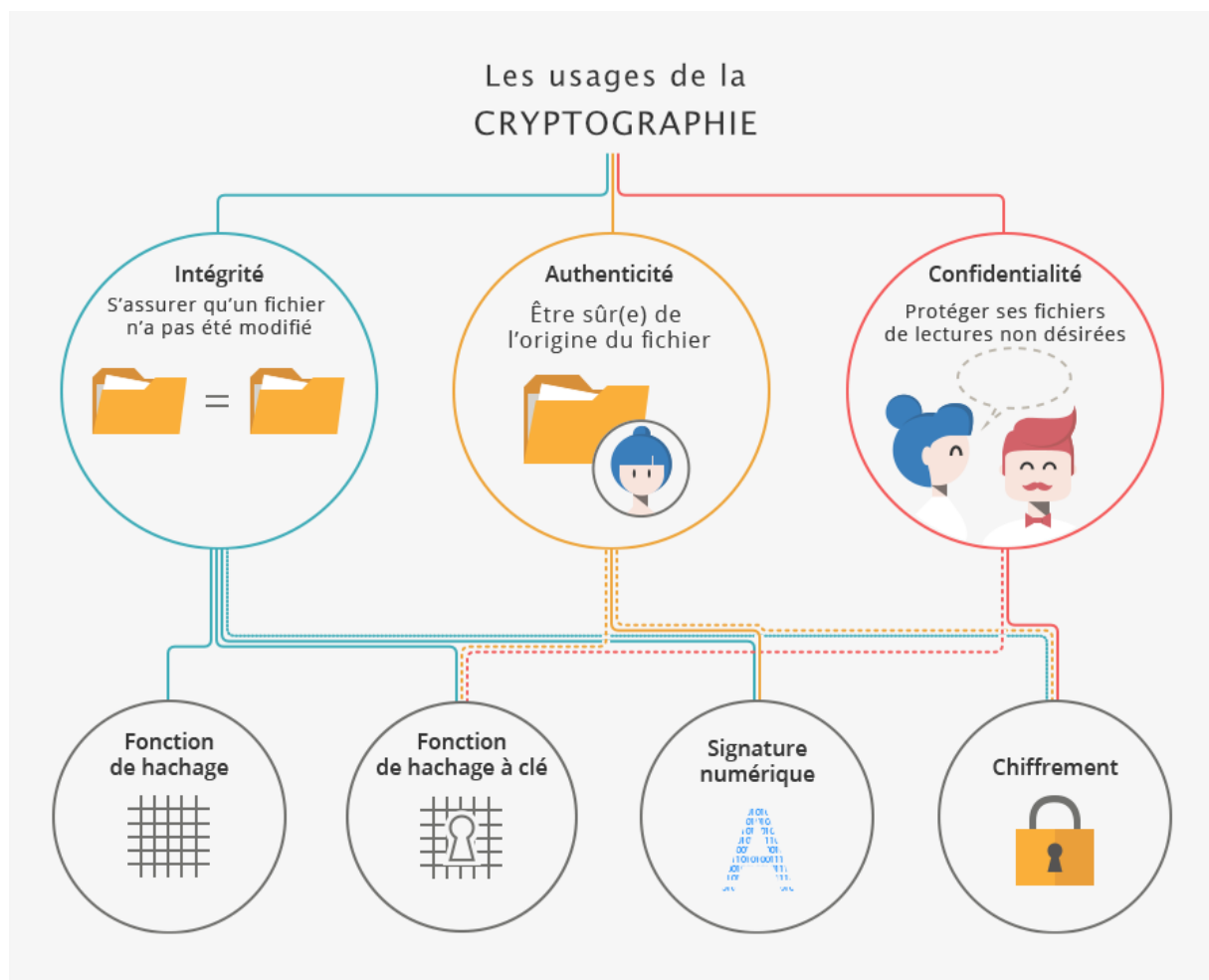


Qu'est-ce que la cryptographie ?

Historiquement, la cryptologie correspond à la science du secret, c'est-à-dire au chiffrement. Aujourd'hui, elle s'est élargie au fait de prouver qui est l'auteur d'un message et s'il a été modifié ou non, grâce aux signatures numériques et aux fonctions de hachage.

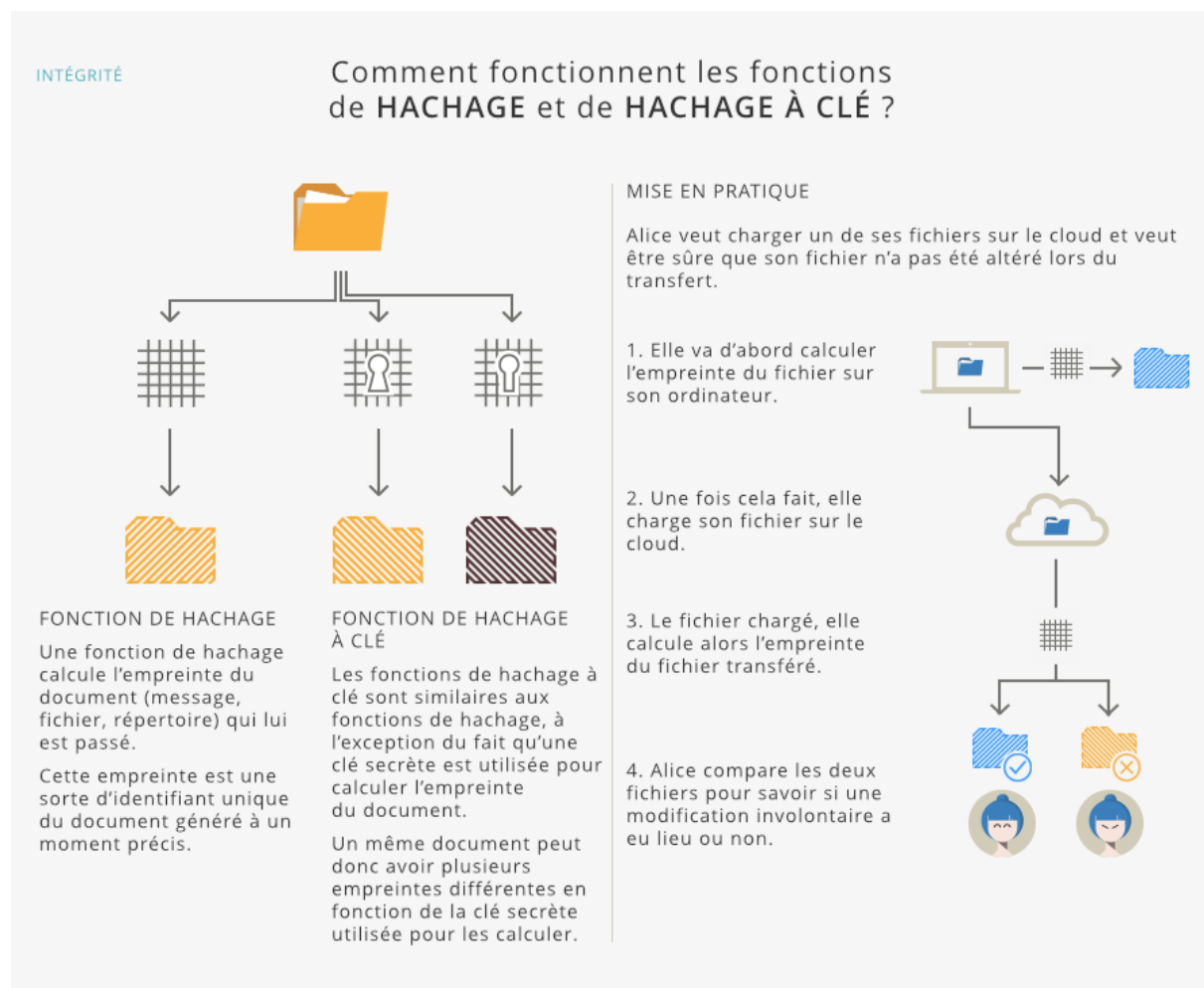


La cryptologie ne se limite plus aujourd'hui à assurer la **confidentialité** des secrets. Elle s'est élargie au fait d'assurer mathématiquement d'autres notions : assurer l'**authenticité** d'un message (qui a envoyé ce message ?) ou encore assurer son **intégrité** (est-ce qu'il a été modifié ?).

Le hachage

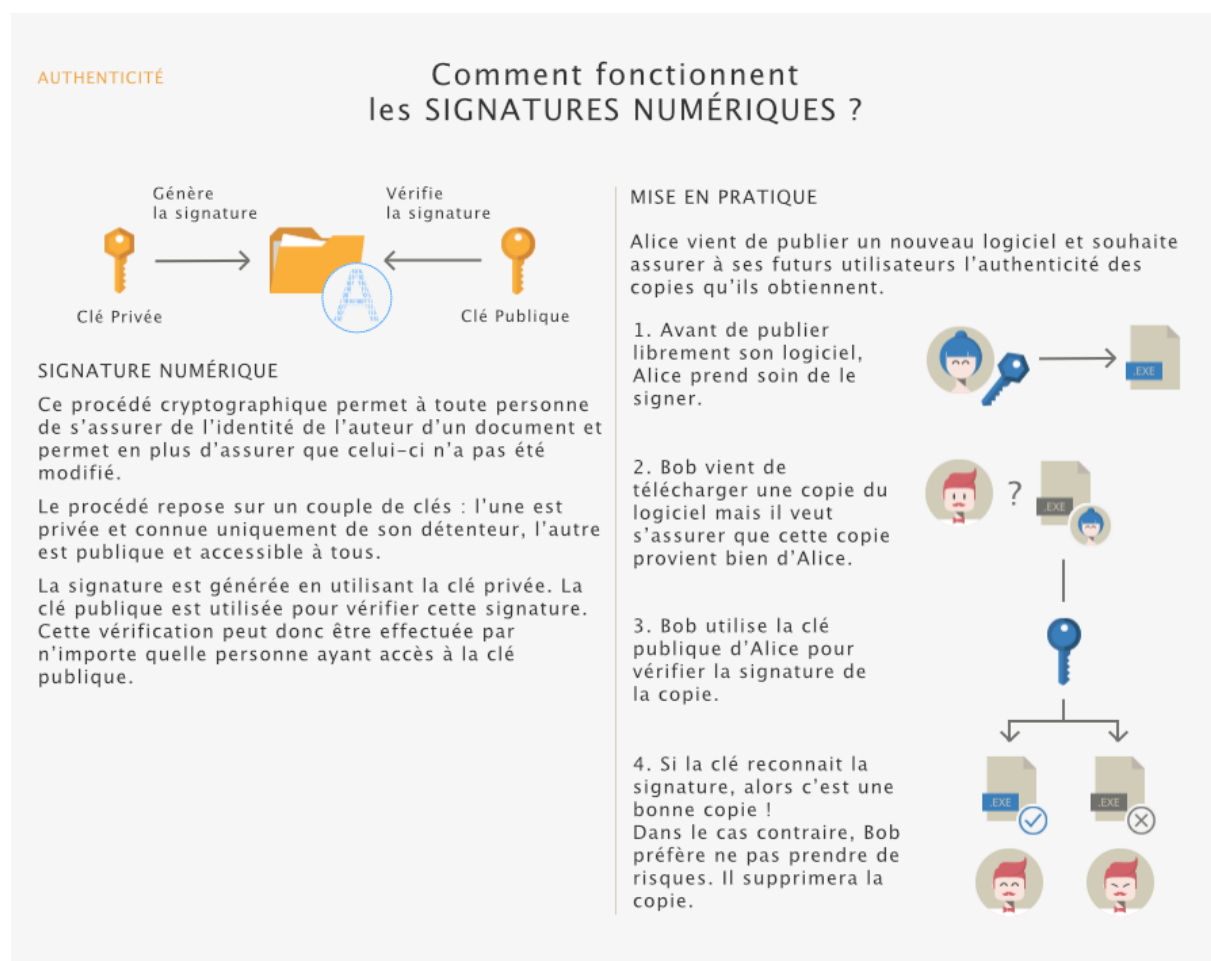
Une « **fonction de hachage** » permettra d'associer à un message, à un fichier ou à un répertoire, une empreinte unique calculable et vérifiable par tous. Cette empreinte est souvent matérialisée par une longue suite de chiffres et de lettres précédées du nom de l'algorithme utilisé, par exemple « SHA2 » ou « SHA256 ».

Il existe aussi des « **fonctions de hachage à clé** » qui permettent de rendre le calcul de l'empreinte différent en fonction de la clé utilisée. Avec celles-ci, pour calculer une empreinte, on utilise une clé secrète. Pour deux clés différentes l'empreinte obtenue sur un même message sera différente.



La signature

Au même titre que pour un document administratif ou un contrat sur support papier, le mécanisme de la « **signature** » - numérique - permet de vérifier qu'un message a bien été envoyé par le détenteur d'une « clé publique ». Ce procédé cryptographique permet à toute personne de s'assurer de l'identité de l'auteur d'un document et permet en plus d'assurer que celui-ci n'a pas été modifié.



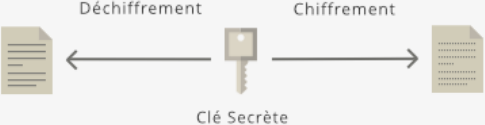
Le chiffrement

Le chiffrement d'un message permet justement de garantir que seuls l'émetteur et le(s) destinataire(s) légitime(s) d'un message en connaissent le contenu. C'est une sorte d'enveloppe scellée numérique. Une fois chiffré, faute d'avoir la clé spécifique, un message est inaccessible et illisible, que ce soit par les humains ou les machines.

CONFIDENTIALITÉ

Comment fonctionne le CHIFFREMENT ?

← Déchiffrement
Clé Secrète
→ Chiffrement





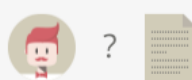

CHIFFREMENT SYMÉTRIQUE

Le chiffrement symétrique permet de chiffrer et déchiffrer un fichier avec la même clé, dite secrète. Pour s'échanger un message il faut donc que les deux parties partagent la même clé.

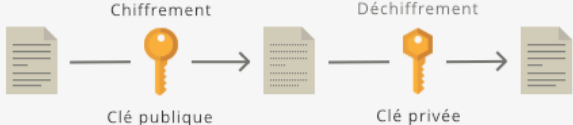
MISE EN PRATIQUE

Alice vient d'enregistrer la liste des cadeaux de Noël de sa famille sur l'ordinateur familial. Elle souhaite être la seule à pouvoir y accéder.

1. Pour ce faire, Alice chiffre la liste en utilisant sa clé secrète.
2. Plus tard dans la journée, Bob trouve la liste et cherche à l'ouvrir.
3. Malheureusement pour lui, Bob est incapable de lire la liste car il ne possède pas la clé secrète.
4. La liste est donc bien protégée. Seule Alice peut réussir à la déchiffrer et la lire !



→ Chiffrement
Clé publique
← Déchiffrement
Clé privée



CHIFFREMENT ASYMÉTRIQUE


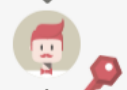

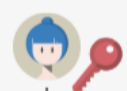
Le chiffrement asymétrique repose sur l'utilisation d'une paire de clés : une publique et une privée.

La clé publique, accessible à tous, est utilisée pour chiffrer les fichiers. Seule la clé privée permet de déchiffrer ces fichiers, celle-ci étant connue que d'un seul individu.

MISE EN PRATIQUE

Alice, hackeuse, vient de découvrir des informations d'intérêt public. Elle veut les transmettre à Bob, journaliste, pour qu'il enquête.

1. Alice vient de récupérer la clé publique de Bob. Elle l'utilise pour chiffrer son document.
2. Elle l'envoie à Bob.
3. Bob reçoit le document et le déchiffre à l'aide de sa clé privée.
4. Une fois le document déchiffré, il rédige un article puis le publie dans son journal.



Les Failles

Les préoccupations en matière de sécurité poussent les développeurs à recourir de plus en plus fréquemment au chiffrement. Sauf que ces derniers n'en maîtrise pas toujours l'implémentation, conduisant à introduire de nouvelles failles dans les applications. Les vulnérabilités dans les protocoles de chiffrement sont la seconde cause la plus commune de failles dans les applications, derrière les défauts dans la qualité du code. La plupart des logiciels touchés par ces vulnérabilités sont des applications Web.

Les failles cryptographiques seraient ainsi plus courantes que d'autres problématiques bien connues, comme l'injection SQL. Elles se nichent notamment dans des validations impropres de certificats, le stockage en clair d'informations sensibles, l'utilisation d'une longueur de clef insuffisante, la présence de clefs cryptographiques codées en dur ou encore dans la vérification incorrecte des signatures cryptographiques.

Selon les études, les développeurs veulent chiffrer, notamment pour répondre aux législations en matière de protection des données, mais ne savent pas s'y prendre. Il y a clairement une absence de formation appropriée dans ce domaine, ce qui aboutit à créer « *un faux sentiment de sécurité*. Nombre de développeurs ont tendance à penser qu'il leur suffit d'appeler une librairie de chiffrement pour sécuriser leurs données. Mais l'implémentation comporte de nombreux pièges.

Accuser les seuls développeurs est toutefois un peu trop simple. La faute est partagée avec les concepteurs de librairies de crypto, pensées pour des spécialistes du sujet et non pour des développeurs lambda. Une lacune dont OpenSSL semble avoir pris conscience, prévoyant de réduire la complexité de l'API et d'améliorer la documentation.