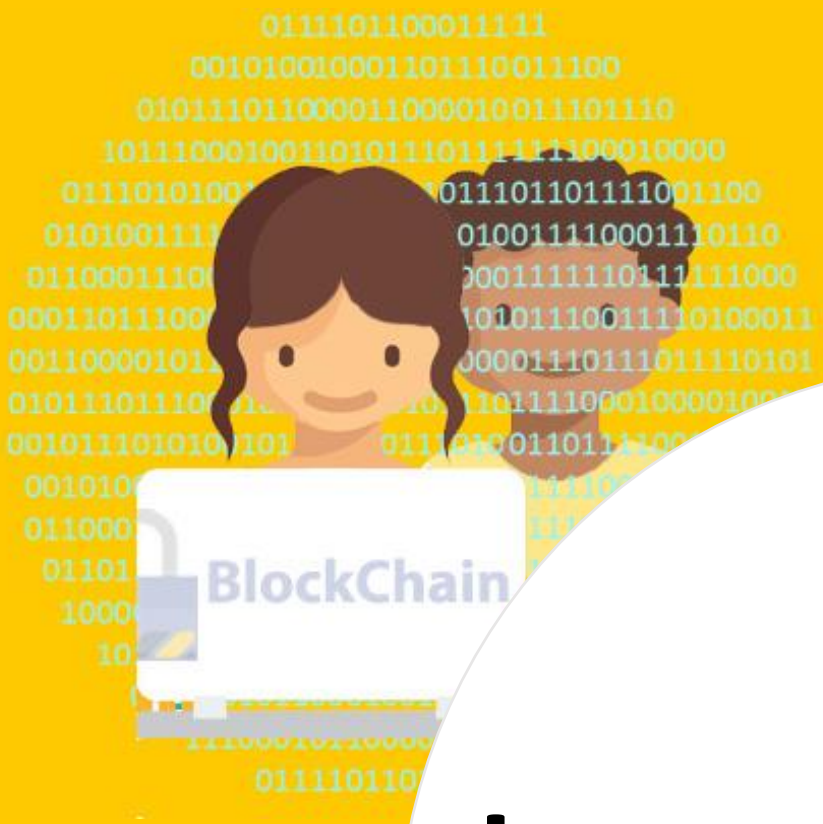
Bee Bryte
Energy Intelligence

10 min



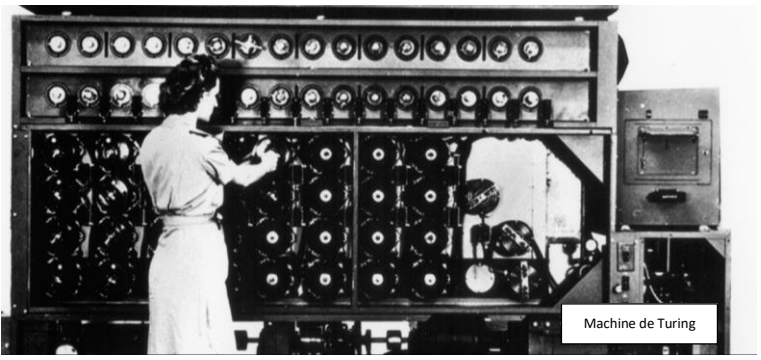
La cryptographie déchaînée

Maxime LAHARRAGUE, Pierre-Jean LARPIN

30 Mai 2018

La cryptographie déchaînée

Dans un monde où les flux d'information s'accroissent de manière exponentielle, il devient essentiel de disposer d'outils performants permettant de garantir certains critères de sécurité (authenticité, intégrité, confidentialité, …) en matière d'échange et de stockage des données. Parmi les outils disponibles, la « BlockChain » focalise aujourd'hui toutes les attentions et les passions.


Machine de Turing

Construction mathématique particulièrement complexe, et objet de nombreux articles et ouvrages dans l'actualité, la BlockChain pourrait profondément transformer une multitude de domaines, et apparaît comme une composante essentielle du développement des modes de communication de demain. Ses « briques de base » ne sont pourtant pas nouvelles et appartiennent au vaste et fascinant domaine de la cryptographie. Nous proposons dans cet article de revenir aux sources et d'explorer cette science du secret.

Pour le développement de sa solution d'intelligence énergétique, notre société BeeBryte s'est très tôt intéressée aux procédés et technologies cryptographiques constituant les bases de la sécurité de tout système d'information (confidentialité, authenticité et intégrité). Utilisés séparément ou en combinaison, ces technologies sont déjà largement intégrées dans notre SaaS, notamment pour protéger les communications entre notre passerelle IoT (Box) et notre plateforme Cloud.

La cryptographie est une discipline qui cherche à protéger l'intégrité et l'authenticité d'un message ainsi que sa confidentialité vis-à-vis de tiers. Utilisée dans des domaines divers et variés, la cryptographie adresse de nombreuses fonctionnalités, en particulier la transmission sécurisée de données (chiffrement), la génération de secrets entre utilisateurs, l'authentification (ou signature) d'un message, ou encore sa protection contre toute altération.

Nous allons ici présenter quelques briques fonctionnelles essentielles à la mise en œuvre de protocoles cryptographiques, comme ceux sous-tendant la BlockChain.

Nous ne détaillerons pas les théories mathématiques complexes qui se cachent derrière les outils exposés. De façon générale, elles reposent sur des opérations relativement simples à effectuer dans un sens, mais extrêmement difficiles à inverser, sauf à recourir à de la « force brute », c'est-à-dire l'exploration systématique de larges combinatoires, limitée par la puissance de calcul disponible. Nous laissons le lecteur souhaitant explorer plus avant les mathématiques sous-jacentes, consulter l'information abondante disponible sur Internet à ce sujet.

PRINCIPAUX CONCEPTS CLES

Chiffrement symétrique – Algorithme AES

Le chiffrement symétrique est fondé sur le partage d'une clé commune entre les interlocuteurs. Cette clé est utilisée par l'envoyeur pour crypter son message, et seul l'usage de cette même clé permet de déchiffrer le message – clé que seuls le destinataire et l'expéditeur du message connaissent.

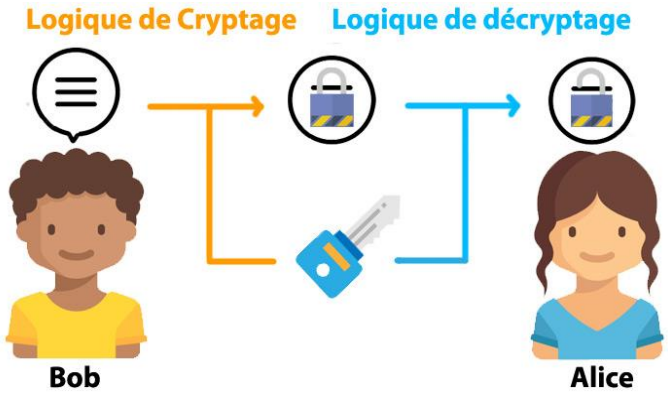
Le chiffrement utilisé permet de transformer du texte brut en des blocs incompréhensibles sans la clé. Toutefois, ces algorithmes présentent l'inconvénient majeur de nécessiter le partage préalable d'une clé commune, en toute confidentialité. Cette difficulté peut toutefois être surmontée à l'aide de techniques comme l'échange de clés Diffie-Hellman.

L'algorithme de chiffrement symétrique le plus couramment utilisé est l'Advanced Encryption Standard (AES), notamment adopté par la NSA aux Etats-Unis. Cet algorithme définit un certain nombre de transformations vouées à être réalisées sur des données stockées dans un tableau. La longueur de la clé utilisée (128, 192 ou 256 bits) détermine la force du chiffrement.

<u>Illustration d'un cas pratique typique :</u>

- Bob veut envoyer un message à Alice, avec qui il partage un secret qu'ils vont utiliser chacun de

leur côté comme clé de chiffrement. Ils prennent bien soin de ne communiquer leur clé partagée à aucune autre tierce partie.



- Bob crypte son message en utilisant la clé de chiffrement, puis envoie le message crypté à Alice.
- Utilisant la clé de chiffrement également, Alice est en mesure de décrypter le message qu'elle reçoit et accède ainsi au message initial de Bob.

Chiffrement asymétrique – Algorithme RSA

En chiffrement asymétrique, chaque utilisateur possède une paire de clés : une clé publique, diffusée à tous les utilisateurs, et une clé privée, qu'il est le seul à connaître. Chaque message chiffré par l'une des deux clés peut uniquement être déchiffré par l'autre clé constituant la paire.

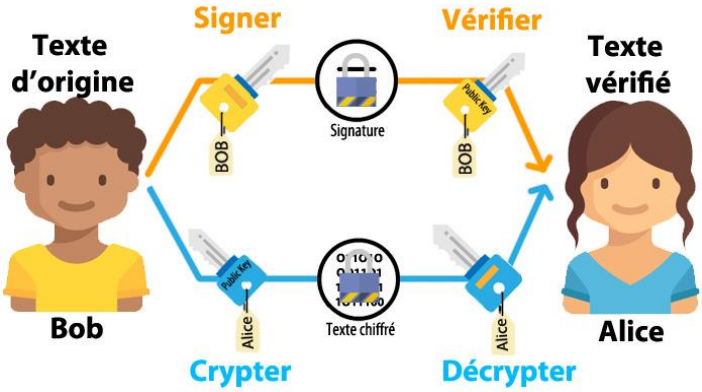
Les algorithmes de chiffrement asymétrique sont fondés sur des problèmes mathématiques complexes n'admettant pas toujours de solution efficace, c'est pourquoi ils sont généralement utilisés pour chiffrer de faibles quantités de données. Le chiffrement asymétrique est principalement utilisé dans 2 situations :

- **Pour certifier l'identité de l'expéditeur**

  Considérons le cas où Bob veut transmettre un message et garantir au(x) destinataire(s) de son message qu'il est bien l'auteur de celui-ci. En le chiffrant avec sa clé privée, Bob a créé une signature digitale qui lui est propre. N'importe qui peut déchiffrer son message en utilisant sa clé publique et confirmer que le message provient bien de Bob. En effet, l'utilisation d'une autre clé publique retournerait un message erroné.

- **Pour envoyer contenu protégé à un destinataire**

  Cette fois-ci, Bob veut envoyer un message à Alice en s'assurant qu'elle seule pourra le lire. En chiffrant son message avec la clé publique d'Alice, Bob s'assure que seule Alice, unique détentrice de la clé privée associée, saura le déchiffrer et le lire.



Les fonctions de hachage

En cryptographie, les fonctions de hachage sont des fonctions mathématiques qui prennent en entrée tout type de donnée et génèrent un code (le Hash) de longueur fixe, caractéristique de la donnée d'entrée.

Une fonction de hachage possède plusieurs propriétés particulières qui en font un outil aux vastes applications en sécurité informatique :

- Elle est déterministe : pour une valeur d'entrée donnée, la fonction de hachage renvoie toujours la même valeur de sortie. Ainsi, le hash constitue en quelque sorte une signature de taille fixe du message… mais une signature non-exclusive.
- En effet, plusieurs messages différents peuvent avoir le même Hash, mais la probabilité de collisions (i.e. probabilité que deux messages différents aient le même hash) est extrêmement faible.

  Cette propriété permet d'authentifier un message : si on connaît le hash de signature d'un message et qu'on reçoit indépendamment un exemplaire de ce message, après calcul de son hash et comparaison positive avec la signature, son authenticité est quasi certaine.
- Une fonction se hachage est en général hyper-sensible aux conditions initiales : la moindre modification du contenu en entrée produit un hash totalement différent. Cette propriété permet

– en plus de l'authenticité – de garantir l'intégrité d'un message, puisque la plus petite altération invalide son hash.

- Elle n'est pas directement inversible : calculer le hash correspondant à un message est aisé, mais l'inverse, c'est-à-dire trouver un message correspondant à un hash, n'est pas possible par calcul direct. La seule solution est de hacher, de manière systématique, des messages différents jusqu'à tomber sur le hash recherché, cette méthode étant physiquement limitée par la capacité de calcul disponible.

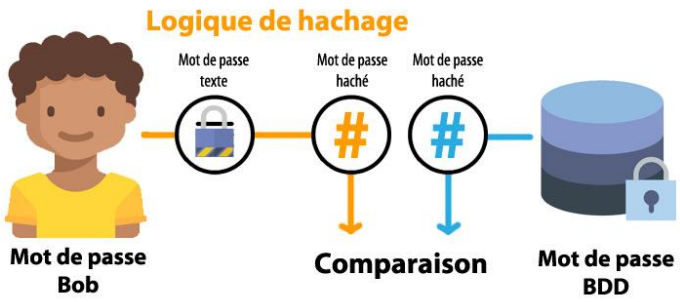


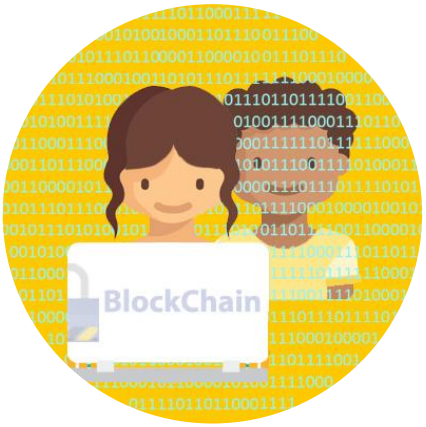
Illustration d'un cas d'utilisation typique :

- Bob souhaite envoyer un message à Alice, et veut s'assurer que le contenu de son message n'est pas altéré au cours de la transmission.
- Alice, de son côté, veut s'assurer que le message qu'elle reçoit correspond bien à ce que Bob voulait lui transmettre.
- Bob hache le message qu'il souhaite envoyer à Alice. Il lui envoie son message ainsi que le hash correspondant.
- Lorsqu'elle reçoit le message, Alice prend connaissance du message en clair de Bob, et hache de son côté le message qu'elle vient de lire. En comparant le hash qu'elle a obtenu avec celui de Bob qu'il lui a transmis, elle est en mesure de savoir si le message qu'elle a lu n'a pas été corrompu lors de la transmission.

Conclusion

Les quelques techniques cryptographiques élémentaires que nous avons exposées constituent des briques de base à partir desquelles sont élaborés des systèmes plus complexes comme la BlockChain, mais peuvent (et doivent) aussi être envisagés individuellement pour adresser un grand nombre de fonctionnalités de sécurisation des données.

Cette mise en perspective nous paraissait essentielle pour aborder ultérieurement de façon dépassionnée les applications possibles de la BlockChain au secteur de l'énergie, avec ses atouts et potentiels mais aussi ses contraintes et les limites de sa pertinence.



BeeBryte développe des solutions utilisant l'IoT, l'Intelligence Artificielle et la BlockChain pour que les bâtiments industriels et commerciaux, les stations de recharge de véhicules électriques et les éco-quartiers consomment l'énergie de manière plus intelligente, moins chère et plus efficacement tout en réduisant leur empreinte carbone ! BeeBryte a une équipe de 20 personnes en France et à Singapour et est soutenue par Intel, BPI & l'ADEME. Depuis sa création en 2015, ses solutions ont reçu de nombreux prix, tels qu'EDF Pulse, Start-up Energy Transition Award, et le label GreenTech Verte.

Si vous voulez participer à la révolution énergétique, n'hésitez pas à nous contacter :

contact@beebryte.com
www.twitter.com/BeeBryteGroup
www.beebryte.com