# Comment ça marche les bitcoins (1)

Crédit photo : Cory Doctorow

Début d'une série d'articles à l'attention des curieux et surtout des journalistes qui se plaisent à parler de « protocole obscur et non public ». Pas de chance, c'est totalement public, par contre, c'est anéfé très obscur. Un peu de théorie du chiffrement pour commencer.

On distingue trois grandes familles de chiffrement :

**Le symétrique :**

Il en existe tout un tas de sortes. Le plus simple pour expliquer le principe est le ROT13 (ROT pour « rotation », qui est la méthode, et 13 est la clé qui représente le nombre de rotations). Prenons un exemple de message à chiffrer : « secret ». Considérons l'alphabet a, b, c, d, e, f… et appliquons une rotation sur chaque lettre avec un décalage de 13.

Le E va donc devenir R, le S va devenir F, etc.

Une fois chiffré, notre « secret » va donc devenir « frperg ». Si le destinataire du message connaît le principe du chiffrement (décaler les lettres de l'alphabet) et la clé (13 décalages) il peut retrouver le mot d'origine.

Evidemment, dans la vraie vie, le ROT13 n'est pas utilisé, sauf pour expliquer le principe ou cbhe f'nzhfre à snver raentre yrf traf. Le plus répandu actuellement est l'AES, qui implique des décalages, des manipulations matricielles, des rotations et des polynômes.

**L'unidirectionnel :**

On parle aussi de hachage. L'idée est de convertir une chaîne de caractère en une autre sans pouvoir faire marche arrière. C'est une fonction intéressante pour stocker des informations auxquelles on n'a pas besoin d'accéder mais qu'on peut vouloir vérifier. L'usage typique est le stockage des mots de passe : on n'a pas besoin de connaître le mot de passe mais uniquement de pouvoir vérifier si celui qu'on nous propose est le bon.

Comment ça marche ? Reprenons notre mot « secret » comme mot de passe et attribuons à chaque lettre un chiffre (par exemple de la table ASCII) : le « e » est 101, le « s » est 115, etc. Notre mot « secret » peut donc s'écrire 115 101 99 114 101 116.

A présent, additionnons tous ces chiffres, on obtient 646. Prenons le reste de la division par 256 (nous travaillons avec des octets, les chiffres vont donc de 0 à 255), on obtient 134. Impossible, avec ce simple chiffre, de remonter au mot « secret ». On peut par contre le stocker et conserver la méthode pour l'obtenir. Si quelqu'un vient nous demander si « coucou » est le bon mot de passe, on fera l'addition, on obtiendra 654, le reste de la division par 256 sera 142 et on saura que « coucou » n'est pas le bon mot de passe.

L'intérêt est de faire en sorte que deux chaînes différentes ne donnent pas le même résultat. Si on se cantonne à additionner des chiffres et à ne garder que le résultat de la division par 256, on risque de tomber sur des chaînes différentes donnant le même chiffre (par exemple, « ddpdd » donne, comme « secret », un reste de 134). Dans la vraie vie, la méthode est un peu plus complexe. On utilise par exemple SHA2.

**L'asymétrique :**

L'idée de base est de palier au problème du système symétrique en permettant de séparer en deux la clé utilisée : une partie servira à chiffrer, l'autre à déchiffrer. Du coup, on peut disséminer la clé permettant le chiffrement tout en conservant pour soi l'autre clé permettant de déchiffrer le message ce qui facilite l'usage courant puisque n'importe qui peut chiffrer un message et vous l'envoyer sans risque qu'un autre puisse le déchiffrer.

Le principe général se base sur les nombres premiers. On en choisit deux, par exemple 3559 et 67. La multiplication des deux donne 238453. A partir de ce chiffre, il est aujourd'hui mathématiquement impossible de retrouver 3559 et 67 à moins d'essayer les multiplications une par une de tous les nombres premiers connus entre 0 et 238453. Dans la vraie vie, on utilisera bien entendu des nombres premiers beaucoup plus grands pour corser la difficulté, par exemple 282755483533707287054752184321121345766861480697448703443857012153264407439766013042402571 et 370332600450952648802345609908335058273399487356359263038584017827194636172568988257769601.

Appliquez ensuite un certain nombre d'opérations mathématiques que je n'ai pas le courage de vulgariser ici, vous obtiendrez la clé privée (à garder pour vous) et la publique (à distribuer). Si quelqu'un a (ou trouve) une explication bien fichue, je suis preneur :)

Deux usages distincts découlent de cette théorie :

- Le chiffrement : une personne lambda qui a connaissance de votre clé publique peut chiffrer un message que seule votre clé privée pourra déchiffrer. C'est ce qui permet, par exemple avec PGP, d'échanger des emails chiffrés en toute discrétion.
- La signature : votre clé privée peut également servir à chiffrer un message que seule la clé publique saura déchiffrer, permettant de s'assurer que vous êtes bien l'expéditeur d'un message.

Un petit mot sur ce principe de signature, car je sens qu'il n'est pas clair dans votre esprit :

Admettons que vous ayez fourni votre clé publique à Tatie Martine. Elle sait donc déchiffrer un message que vous auriez chiffré avec votre clé privée. Votre but n'est pas de fomenter un attentat mais uniquement de faire en sorte qu'elle puisse s'assurer que les emails que vous lui envoyez viennent bien de vous. Il n'est donc pas question de chiffrer totalement le message mais uniquement une signature de celui-ci. Le principe de signature est le suivant :

1. Vous composez votre message (du texte, par exemple, disons « le message »)
2. Vous passez ce message dans un système de chiffrement unidirectionnel. Si on reprend celui ci-dessus, ça donne un reste de division de 214 (dans la vraie vie, ça ressemblera plus à quelque chose
   comme 681c58fa6edcc97a180ad594ca456ae81ef45a3f86408472c378c0e990d22378, si on utilise SHA-2)
3. Vous chiffrez ce résultat avec votre clé privée
4. Vous envoyez le message original en clair accompagné du résultat de ce chiffrement
5. Tatie Matine lit votre prose transmise en clair et se dit « c'est pas possible, c'est pas lui qui m'a envoyé ça !! »
6. Elle passe votre message dans le même système de chiffrement unidirectionnel et trouve 214.
7. Elle déchiffre la signature avec votre clé publique et retrouve 214.
8. Elle sait donc que « le message » vient de vous.

Bien entendu, si Tatie Martine dispose elle aussi d'un couple clé privée / clé publique, les fonctions de signature et de chiffrement peuvent être combinées, permettant à la fois à votre message d'être signé par votre clé privée ET chiffrée par la clé publique de Tatie Martine, qui va ensuite le déchiffrer avec sa clé privée et vérifier la signature avec votre clé publique.

Comme je sens bien que je vous ai déjà donné mal à la tête, on va s'arrêter là pour aujourd'hui. On parlera, dans le prochain article, de comment tout ceci s'applique aux bitcoins et plus particulièrement à leur mystérieuse fabrication.