Pour qu'il y ait un intérêt pratique à participer à ce contrôle collectif sans lequel la cryptomonnaie n'existe plus, une récompense est donnée à intervalles réguliers à l'une des machines tenant les comptes. Aujourd'hui par exemple, 25 *bitcoins* sont distribués toutes les dix minutes par le protocole *Bitcoin*. Cette machine gagnante est désignée par un procédé fondé sur une preuve de travail dont le fonctionnement doit être parfait pour que le protocole résiste aux attaques. C'est la plus belle application à ce jour de l'idée des preuves de travail, et de nombreuses caractéristiques de *Bitcoin* en dépendent. Donnons quelques précisions.

Chaque machine candidate essaie de résoudre un problème dont les paramètres sont fixés par le protocole *Bitcoin* et qui est formulé sans intervention d'un service centralisé. Il s'agit d'un problème du type « trouver un $x$ tel que $f_p(x) < y$ » évoqué plus haut. La fonction $f$ à sens unique est connue de tous, et c'est la fonction de hachage SHA-256. On l'utilise beaucoup en cryptographie et les spécialistes s'accordent à considérer qu'elle a les propriétés voulues pour servir dans une preuve de travail. La première machine qui résout le problème gagne la compétition et emporte les 25 *bitcoins*. Le protocole ajuste régulièrement le seuil donné par le nombre $y$, qui fixe la difficulté de la preuve de travail en prenant en compte les concours précédents, de telle façon que le temps moyen entre deux concours soit de dix minutes environ.

## Des preuves de travail pour les bitcoins

Cette émission contrôlée, prévisible et faible de *bitcoins* toutes les dix minutes est la seule façon d'en créer. Cela a pour intérêt que les détenteurs de *bitcoins* sont certains de ne pas être victimes des phénomènes inflationnistes provoqués par l'émission massive de devises, comme c'est parfois le cas pour les monnaies contrôlées par les banques centrales.

Participer à ce concours porte le nom de « minage », car le travail de calcul fait par les machines évoque celui d'un mineur dans une mine d'or, qui le conduit à s'enrichir s'il a la chance de trouver une pépite. Les informaticiens propriétaires des machines participant à ce concours permanent, répété toutes les dix minutes, se nomment entre eux des « mineurs ».

Il est important de comprendre que puisqu'il n'y a pas d'autorité centrale gérant les *bitcoins* (il en va de même pour les autres cryptomonnaies), l'acceptation de la preuve de travail se fait collectivement par tous les mineurs. Quand l'un d'eux a réussi à relever le défi de trouver un $x$ tel que $f_p(x) < y$, il envoie un message sur le réseau, qui prévient les autres mineurs. Ceux-ci vérifient alors, chacun de son côté, que la solution proposée est correcte et se remettent au travail pour résoudre le défi suivant.

Le temps nécessaire pour qu'un gagnant soit trouvé n'est pas fixé avec précision : la

### 2. Le hachage

Une fonction de hachage associe, à un $x$ donné (une suite de caractères ou un nombre) de longueur quelconque une valeur $f(x)$ de taille fixe nommée « empreinte de $x$ » ou « condensé de $x$ » ou *hash*. Le passage de $x$ à $f(x)$ se fait par une série d'opérations qui commencent en général en découpant $x$ en petits morceaux (d'où le nom de hachage), qui sont ensuite rassemblés et emmêlés de façon à obtenir un mélange aussi parfait que possible. Pour la fonction classique de hachage SHA-256, la longueur de $f(x)$ est de 256 bits ; il y a donc $2^{256} = 1{,}157 \times 10^{77}$ valeurs possibles, ce qui est considérable.

Les fonctions de hachage à usage cryptographique ont les propriétés suivantes.

*(a)* Elles sont « à sens unique » : le passage de $x$ à $f(x)$ se calcule facilement et rapidement, mais pour un $y$ donné, trouver un $x$ tel que $y = f(x)$ est impossible en pratique.

*(b)* Il est impossible en pratique de trouver deux valeurs $x$ et $x'$ différentes telles que $f(x) = f(x')$.

*(c)* Elles produisent des valeurs assimilables à des valeurs aléatoires uniformes (prises dans l'ensemble des valeurs possibles) et, en particulier, même si $x$ et $x'$ sont deux suites de caractères n'ayant qu'un seul caractère différent, les valeurs $f(x)$ et $f(x')$ n'ont aucun lien apparent.

Voici un exemple. La donnée $x$ est transformée en une suite de 0 et de 1 qui est découpée en morceaux de longueur 8. Les morceaux sont ensuite additionnés comme des nombres en base 2. On ne garde du résultat que les quatre chiffres en positions 3, 4, 5 et 6 à partir de la droite :

$x$=01010010100101010010010101 0 → 01010010 + 10010101 + 00100101 + 010 = 100001110 → $f(x)$ = 0011.

Cet exemple est bien sûr trop simple pour avoir les propriétés *(a)*, *(b)* et *(c)*. La fonction de hachage SHA-256 mélange soigneusement les bits de la donnée $x$ selon un algorithme public (voir http://en.wikipedia.org/wiki/SHA-2).

Les fonctions de hachage cryptographiques ont de multiples usages dont voici trois exemples.

– Elles permettent d'assurer l'intégrité d'un gros fichier $x$, auquel on lui associera son empreinte $f(x)$, beaucoup plus courte. Celui qui téléchargera $x$ s'assurera que $x$ lui parvient sans erreur et en totalité en calculant l'empreinte du fichier qu'il aura reçu, qui doit être la même que celle publiée sur le site. Si $x$ a mal été copié, il le saura, car la probabilité que la modification n'ait pas changé l'empreinte est infinitésimale.

– Au lieu de mémoriser les mots de passe d'accès aux comptes sur un service en ligne, celui-ci ne mémorise que les empreintes des mots de passe. Le serveur peut vérifier que les utilisateurs donnent les bons mots de passe, mais si un pirate accède à sa liste d'empreintes, il ne pourra rien en faire.

– Pour produire une suite de nombres aléatoires dépendant d'un fichier $x$, on calculera par exemple $f(x.1)$, $f(x.2)$,... où $x.i$ désigne le fichier $x$ suivi de l'écriture du nombre $i$ (par exemple en binaire). En changeant le $x$, on a autant de suites de nombres aléatoires qu'on veut (ce qui permet par exemple de chiffrer des messages).

L'importance des fonctions de hachage est telle, en cryptographie, qu'un grand soin est mis à les élaborer et à les soumettre à toutes sortes de tests et d'attaques. Le NIST, aux États-Unis, propose des fonctions bien contrôlées. La fonction SHA-256 utilisée par *Bitcoin* en est une. Le NIST publie aussi des recommandations concernant le choix et l'usage des fonctions de hachage. Il a récemment organisé une compétition pour concevoir une nouvelle génération de fonctions de hachage qui a abouti à la norme SHA-3 en 2012. Voir : http://www.nist.gov/itl/csd/ct/hash_competition.cfm.



© Pour la Science - n° 438 - Avril 2014