Pas de fausse *blockchain* !

Pour rendre une *blockchain* robuste et impossible à simuler, on utilise une fonction de hachage *h* (publique) qui à tout fichier F associe un code *h*(*F*), une courte suite de caractères dénommée empreinte.

Quand on change un seul caractère de F, le code *h*(F) est totalement changé et ce de façon imprévisible. Il est impossible en pratique de modifier F en F' de manière que *h*(F) = *h*(F') *(voir la rubrique Logique & calcul d'avril 2014).*

On utilise aussi une preuve de travail. C'est une fonction *p* qui, à toute valeur *V* donnée (en général, *V* est une suite de caractères qui détermine un problème), associe une solution *p*(*V*) qui ne s'obtient qu'en menant un long calcul, par exemple d'une durée d'une heure avec une machine de bureau ordinaire. En revanche, vérifier que *p*(*V*) est la bonne valeur associée à *V* est rapide. Calculer le résultat *R* de *p*(*V*) est difficile, mais vérifier que *p*(*V*) = *R* est facile. Un exemple est la décomposition d'un nombre en facteurs premiers, qui est difficile, alors que la vérification que le produit de ces facteurs donne le bon résultat est facile.

Construisons une *blockchain* en concaténant des *blocks* contenant des suites de caractères représentant des messages, ou des transactions : *blockchain* = *block*(1) - *block*(2) - ... - *block*(*n*)

Le *block*(*i*) commence par deux informations particulières *h*(*i*) et *p*(*i*) ajoutées aux messages qui en constituent le contenu. Ces informations *h*(*i*) et *p*(*i*) assurent qu'on ne pourra ni modifier la *blockchain*, ni fabriquer après coup de fausses *blockchains*.

h(*i*) sera l'empreinte de ce qui précède *block*(*i*). Autrement dit, *h*(*i*) = *h*[*block*(1) - ··· - *block*(*i* – 1)].

p(*i*) sera une preuve de travail associée à *h*(*i*) : *p*(*i*) = *p*[*h*(*i*)].

Le calcul des *h*(*i*) et *p*(*i*) se fera progressivement avec l'ajout des *blocks*. Chaque *p*(*i*) exige peut-être une heure de calcul en moyenne, mais résulte d'un calcul collectif qui a pris beaucoup moins qu'une heure par participant. Le calcul de tous les *p*(*i*) se fait progressivement : pour le bitcoin, on a ajouté un *block* toutes les dix minutes depuis janvier 2009 et l'on rétribue ceux qui mènent les calculs.

Quand la *blockchain* comporte de nombreuses pages (la *blockchain* du bitcoin en a environ 300 000), faire une fausse *blockchain* est impossible, car cela demande trop de calculs. Pour 300 000 pages « coûtant » chacune une heure de calcul, il faudrait faire un calcul de 300 000 heures, soit 34 années (et pour le bitcoin, il existe d'autres garde-fous qui font que le calcul est bien plus long). Cependant, vérifier que la *blockchain* est cohérente est facile :
1) On vérifie que chaque *h*(*i*) est convenable en recalculant sa valeur, ce qui est possible car tout est public. C'est rapide et cela assure que rien n'a été modifié dans les *blocks*. 2) On vérifie les *p*(*i*), ce qui est rapide aussi.

Une telle *blockchain* est un objet numérique que chacun peut contrôler rapidement, mais que personne ne peut modifier sans se faire repérer et que personne ne peut imiter, car pour cela il faudrait mener un énorme calcul.

© YiinPhoto/shutterstock.com

de trop gros volumes de données), mais seulement les informations d'enregistrement et d'authentification. Les messages sont gérés par un système de dépôt distribué (c'est-à-dire répartis entre les utilisateurs) avec une table permettant de savoir où se trouvent les informations, comme cela se pratique dans les réseaux pair à pair de partage de musique.

Un projet plus ambitieux, car se voulant le support possible d'applications complexes fondé sur une notion de contrat *(smart-contract)*, est en cours de développement : il se nomme *Ethereum*. La *blockchain* associée émettra une monnaie (des *éthers*) sur le modèle du bitcoin, mais ce ne sera qu'une de ses fonctions.

Relier les chaînes

Une autre avancée toute récente dans ce domaine a été proposée par un groupe de chercheurs associé à Adam Back, inventeur britannique d'un système de preuve de travail *(voir cette rubrique dans Pour la Science d'avril 2014)* utilisé comme élément du protocole du bitcoin. Adam Back et ses collègues ont constaté que le bitcoin évolue très lentement, les décisions pour tout changement se faisant selon un processus où il faut un accord difficile à obtenir de la part de ceux qui travaillent à le surveiller et qui ne sont pas organisés en structure hiérarchique : c'est un problème avec les applications totalement décentralisées dont le contrôle n'est aux mains de personne.

Adam Back a aussi noté que beaucoup d'idées innovantes proposées par des *blockchains* nouvelles n'ont qu'un succès limité ; en valeur, le bitcoin reste très dominant parmi les monnaies cryptographiques. Lui et ses collègues ont donc mis au point une méthode liant les *blockchains*. Ce système de *sidechain* permettra de faire passer des unités monétaires d'une chaîne A vers une autre, B. Elles disparaîtront de la chaîne A pour réapparaître sur B et pourront éventuellement revenir dans A.

Chaque *blockchain* est un petit univers où il est utile de disposer d'une monnaie (comme sur *Namecoin*) ; cependant, faire accepter une nouvelle monnaie et stabiliser