3

LES FONCTIONS VÉRIFIABLES À DÉLAI

Savoir qu'un calcul ne peut pas être fait rapidement est parfois utile. Les fonctions vérifiables à délai, ou VDF (pour *verifiable delay function*), ont été introduites en cryptographie il y a trois ans par Dan Boneh, Joseph Bonneau, Benedikt Bünz et Ben Fisch, des universités Stanford et de New York. Il est probable que ce nouveau concept cryptographique jouera un rôle important dans l'avenir.

Une VDF, comme une fonction de hachage, prend une donnée quelconque et la transforme en une donnée aléatoire, mais pour cela elle opère un calcul complexe impossible à mener rapidement et qui, en particulier, ne se parallélise pas. Il faut par exemple 30 secondes à la VDF pour faire le calcul, même en utilisant les plus puissants outils disponibles. De plus, connaissant le résultat d'un calcul fait par la VDF, il est facile de vérifier qu'il est correct.

Bien utilisées, ces VDF rendent impossible pour un site émetteur d'une balise aléatoire de tricher. L'idée est que le site qui produit ses propres données et les mélange à des données extérieures pourrait tricher s'il savait quelles données émettre pour que, combinées avec celles reçues, il produit un résultat biaisé. Or si le site émetteur apprend trop tardivement ce qui résulte des données qu'il mélange avec d'autres, il lui sera impossible de choisir ce qu'il propose dans le mélange pour fausser le résultat. Détaillons le protocole.

Imaginons une balise qui émet des séquences de bits aléatoires toutes les 60 secondes en mélangeant celles qu'elle propose (et qu'elle peut truquer) à d'autres reçues périodiquement de l'extérieur (impossibles à truquer, car publiques). Voici le détail du protocole entre l'instant t et l'instant $t + 60$ secondes que va appliquer la balise aléatoire, procédure vérifiable *a posteriori* par tous.

De l'instant t à l'instant $t + 30$
La balise reçoit des données d_1, d_2, d_3, ..., d_n, qu'elle va chercher sur d'autres balises ou qu'on lui envoie. Ces données sont rendues publiques.

À l'instant $t + 30$
La balise fixe sa propre donnée Q (tirée par exemple d'un dispositif quantique) et en publie l'empreinte $f(Q)$ (où f est une fonction de hachage).
La balise regroupe les données reçues et la sienne en une donnée D.

De l'instant $t + 30$ à l'instant $t + 60$
La balise calcule le résultat de la VDF appliquée à D.
La VDF ne peut pas faire le calcul plus rapidement qu'en 30 secondes. Personne d'autre, durant ces 30 secondes, ne peut le faire, car Q est resté secret.

À l'instant $t + 60$
La balise publie le résultat R du calcul et la donnée Q restée jusque-là cachée. Avec une telle procédure, chacun peut vérifier instantanément que :
– s'il avait envoyé une donnée, elle a été prise en compte dans D.
– la balise n'a pas changé sa donnée Q entre l'instant $t + 30$ et l'instant $t + 60$.
– le calcul de R a été fait comme il fallait.

Celui qui propose une donnée ne peut pas biaiser le résultat, car quand il propose sa donnée, il ne dispose pas de Q qui y sera mélangée. La balise elle-même ne peut pas biaiser le résultat ; en effet, quand elle commence le calcul avec D, elle ne peut plus changer sa donnée Q (qui est dans D) car elle en a publié l'empreinte, et elle ne sait pas avant l'instant $t + 60$ l'effet que sa donnée Q produit sur le résultat aléatoire final.

Aujourd'hui, de nombreux travaux sont menés pour trouver des VDF les plus simples possible et les plus sûres. (voir https://vdfresearch.org/).

> sur des méthodes cryptographiques avancées pour sécuriser les émissions.

Les balises évoquées sécurisent et améliorent la vérifiabilité *a posteriori* de leurs émissions. Quelles sont les méthodes mises en œuvre ?

GARANTIR L'IMPOSSIBILITÉ DE MODIFICATION

La première idée est la signature. Les données aléatoires produites peuvent être signées par un procédé cryptographique qui assure que ce qu'on lit sur le site internet n'a pas été écrit ou réécrit par un autre acteur que le site producteur. Les signatures numériques sont vérifiables par tous, mais seul l'organisme producteur (qui indique sa clé publique permettant le contrôle) peut (grâce à sa clé privée) produire les signatures.

Pour garantir que les nombres publiés ne soient pas modifiés après coup, on peut utiliser une fonction de hachage (*voir l'encadré 2*). En même temps que la balise aléatoire publie le k-ième nombre aléatoire N_k, elle publie un autre nombre E_k qui est l'« empreinte », par une fonction de hachage cryptographique, de $E_{k-1}N_k$ (la concaténation de E_{k-1} et de N_k). Cette publication lie toutes les publications antérieures, car la publication à l'étape k dépend de celle à l'étape $k-1$, qui elle-même dépend de la publication à l'étape $k-2$, etc. Le procédé rend visible toute modification qui serait apportée aux N_k après leur publication.

NOUVEAUX PERFECTIONNEMENTS

Une méthode pour assurer que les bits utilisés ne soient pas soumis à des manipulations est l'utilisation combinée de plusieurs balises aléatoires indépendantes. Comment le faire et qu'est-ce que cela donne comme garantie ?

– Considérons par exemple quatre balises produisant chacune 512 bits toutes les minutes ; notons-les A, B, C et D ;

– Pour connaître le résultat combiné des quatre balises, vous opérez une addition modulo 2 des bits des quatre séries : pour chaque k, vous regardez si le nombre de 1 produits en position k par les balises A, B, C et D est pair ; si c'est le cas, le bit en position k de la combinaison est 0 ; sinon, c'est 1.

La séquence de 512 bits obtenue est un mélange très intéressant. En effet, d'une part, la qualité du hasard produit sera bonne si l'un au moins des sites produit un hasard convenable. D'autre part, pour truquer cette combinaison des quatre balises, il faudrait que l'attaquant truque les quatre sites émetteurs à la fois, ou en truque un en ayant connaissance de ce que les autres publient à l'instant du trucage.

Les blockchains, dont celles des monnaies cryptographiques bitcoin ou ether, ont été proposées comme sources d'aléas publics

infalsifiables et vérifiables *a posteriori*. L'utilisation de fonctions de hachage au sein des pages ajoutées périodiquement aux fichiers blockchains semble introduire un hasard dans ces blockchains qui par ailleurs sont publiques et infalsifiables car conservées en de multiples exemplaires lisibles par tous. Cependant, une étude publiée en 2018 par Cécile Pierrot, de Sorbonne Université, à Paris, et Benjamin Wesolowski, de l'École polytechnique fédérale de Lausanne (EPFL), a montré qu'en réalité de telles sources pourraient être manipulées par des acteurs assez puissants. Même si cela peut avoir un certain intérêt d'utiliser l'aléa des blockchains en le mêlant à d'autres sources, une balise aléatoire ne doit pas se fonder sur cette méthode seule.

Plus intéressant, le cryptologue Arjen Lenstra, de l'EPFL, et Benjamin Wesolowski, ont récemment introduit deux idées nouvelles susceptibles de renforcer la confiance envers les balises aléatoires modernes.

La première idée est de permettre aux acteurs qui le souhaitent de participer à ce qui est produit. Entre deux émissions successives, on permettrait à quiconque le souhaite d'envoyer des nombres à la balise. Tous ces nombres seraient regroupés en un seul dont on calculerait l'empreinte, et l'on mêlerait celle-ci aux résultats du mécanisme local (quantique ou autre) de la balise, par exemple par le biais d'une somme modulo 2 comme quand on mélange plusieurs balises. De plus, le site de la balise autoriserait la vérification que les nombres envoyés sont bien pris en compte.

Le fait de pouvoir envoyer ses propres données assurerait que, même si tous les autres intervenants et la balise forment une association dont le but est de vous tromper, elle ne réussira pas. Le fait que vous interveniez ne vous donne heureusement pas le pouvoir de manipuler la balise et cela même en vous coordonnant avec tous les intervenants extérieurs, car la composante locale (avec une somme modulo 2) rend vaine toute tentative de biais.

DES FONCTIONS DONT LE CALCUL NÉCESSITE UN DÉLAI

La seconde idée est plus subtile, mais la plus importante peut-être pour assurer qu'aucune manipulation n'est possible, même par la balise. Elle utilise le concept de *fonction vérifiable à délai* ou VDF (le sigle anglais), objet d'intenses recherches depuis trois ans.

Une VDF est une fonction qui, lorsqu'on lui confie un fichier quelconque, le transforme en une séquence de chiffres aléatoires, comme le fait une fonction de hachage (*voir l'encadré 3*). Mais en plus, elle le fait de telle façon que le calcul ne peut pas être mené en moins d'une certaine durée (par exemple 30 secondes), et que la vérification de son résultat final est quasi instantanée: calculer est long, vérifier est rapide.

Même si une balise aléatoire combine des sources extérieures, selon la première idée, un problème persiste qu'une VDF va résoudre. Le problème est qu'au moment où la balise mélange les données extérieures (qu'elle ne contrôle pas) et une donnée locale, par exemple présentée comme quantique, elle pourrait s'arranger, connaissant les données extérieures, pour que la combinaison avec ce qu'elle introduit (prétendument tiré d'un mécanisme quantique), conduise à un résultat final biaisé comme elle le souhaite. En clair, à cause de la donnée locale qu'elle introduit, la balise garde une possibilité de tricherie. Comment rendre impossible cette manipulation?

Voici la solution d'Arjen Lenstra et Benjamin Wesolowski. La balise exige que les données externes qu'elle prendra en compte lui soient parvenues durant les 30 premières secondes de l'intervalle de temps de 60 secondes entre deux émissions (en t, et en t+60). Elle-même, à l'instant t+30, engage sa donnée (par exemple quantique) en publiant son empreinte. La balise combine les données extérieures et la sienne et, à partir du résultat, effectue le calcul de la VDF qui exige 30 secondes. Le résultat de ce calcul est connu et publié par la balise à l'instant t+60 (il ne peut pas l'être avant) avec la donnée dont la balise n'avait communiqué que l'empreinte à l'instant t+30. Tout le monde peut vérifier instantanément que le calcul fait par la balise est correct (c'est la propriété de vérifiabilité instantanée) et qu'elle n'a pas changé sa donnée (puisqu'elle en avait publié l'empreinte).

Les acteurs extérieurs n'ont pas pu agir sur le résultat, car ils ne connaissaient pas la donnée de la balise, restée secrète jusqu'à t+60, et que le résultat en dépend.

Plus important, la balise n'a pas pu proposer une donnée conduisant à un résultat qui l'arrange. En effet, dès l'instant t+30, elle a figé la donnée qu'elle proposait (en en publiant l'empreinte) et il lui a fallu 30 secondes pour savoir comment cette donnée combinée aux autres données venant de l'extérieur (qu'elle ne connaissait pas entièrement avant t+30) produisait le résultat. Dans le protocole, ni un fournisseur de données externes ni la balise elle-même ne peuvent tricher. Le résultat dépend de toutes les données et il est donc vraiment aléatoire, si un seul des fournisseurs de données (local ou externe) en livre qui soient aléatoires.

On dispose ainsi de méthodes cryptographiques pour concevoir et réaliser des balises aléatoires fiables et sécurisées. Rien ne semble interdire ou limiter les applications exigeant des balises parfaitement sécurisées. On peut désormais exiger des sites internet de jeux, des loteries et lotos divers qu'ils se fondent de manière transparente sur de telles balises. ■

BIBLIOGRAPHIE

K. Pietrzak, **Simple verifiable delay functions**, *IACR Cryptology ePrint Archive*, n° 627, 2018.

D. Boneh *et al.*, **Verifiable delay functions**, dans H. Shacham et A. Boldyreva (éd.), *Advances in Cryptology – CRYPTO 2018*, Springer, 2018.

C. Pierrot et B. Wesolowski, **Malleability of the blockchain's entropy,** *Cryptography and Communications,* vol. 10(1), pp. 211-233, 2018.

A. Lenstra et B. Wesolowski, **Trustworthy public randomness with sloth, unicorn, and trx**, *Int. J. of Appl. Crypt.*, vol. 3(4), pp. 330-343, 2017.

J. Bonneau *et al.*, **On Bitcoin as a public randomness source**, *IACR Cryptology ePrint Archive*, n° 1015, 2015.

La balise aléatoire de Mads Haarh : www.random.org

La balise aléatoire du NIST : https://beacon.nist.gov

La balise aléatoire de l'université du Chili : https://random.uchile.cl/en/

Un site collectant les derniers travaux sur les VDF : https://vdfresearch.org/