tirés du bruit atmosphérique capté par radio. On peut par exemple obtenir, à la demande, le résultat d'un lancer de deux dés, ou le mélange d'un paquet de 52 cartes, ou même une image aléatoire faite de pixels noirs et blancs.

Random.org affirme extraire 12000 bits par seconde du bruit atmosphérique, et c'est à partir d'eux que les autres services sont fournis. Si on le souhaite, les nombres que le site communique sont mémorisés, mais ce n'est pas très facile car le site n'est pas conçu pour jouer le rôle de balise aléatoire sécurisée émettant périodiquement. Par ailleurs, on ne sait pas vraiment comment le bruit atmosphérique est capté et transformé en données aléatoires. Utiliser random.org pour les applications n'exigeant pas une certitude d'honnêteté et de sécurité totale sera une assez bonne solution, mais, tel qu'il est conçu, ce site internet n'est pas en mesure de fournir assez de garanties pour des applications où par exemple des sommes importantes d'argent seraient en jeu.

LA BALISE DU NIST: UNE LISTE DE 512 BITS CHAQUE MINUTE

C'est sans doute pourquoi l'institut américain des normes et de la technologie, le NIST (National Institute of Standards and Technology), travaille sur ce problème depuis 2011 et propose une balise aléatoire, conçue cette fois comme un émetteur public et périodique d'aléas dont les données produites sont protégées et restent consultables sans limitation de temps. La balise du NIST publie chaque minute une liste de 512 bits aléatoires (*voir* https://beacon.nist.gov/home). Et le NIST encourage la création d'autres balises indépendantes.

Le NIST utilise les bits émis par un dispositif quantique. Il fournit un moyen de vérifier que les bits publiés ne sont pas modifiés une fois publiés et affirme que son site est protégé contre toute intervention extérieure, ce qui assure que personne ne manipule le mécanisme de production ou de publication.

Si l'on considère que le NIST est un tiers de confiance parfait, on peut utiliser ce qu'émet sa balise. Par exemple, si des sites de jeux qui utilisent des nombres aléatoires vont les y rechercher en indiquant comment, on pourra avoir confiance en eux et vérifier qu'il n'y a pas tricherie, car il est improbable que le NIST se laisse corrompre par eux. Malgré tout, comme le notent de nombreux spécialistes, le NIST ne donne aucun moyen pour contrôler que les bits qu'il publie sont produits par des phénomènes quantiques et qu'aucune manipulation n'est opérée par l'institut sur ce qu'il émet. Le NIST ayant déjà été soupçonné de tentatives de trucage liées à des outils de génération d'aléas (*voir* https://en.wikipedia.org/wiki/Dual_EC_DRBG), il est impossible de recommander d'utiliser sa balise pour les applications exigeant une sécurité absolue.

Remarquons que, sur sa page, le NIST indique en rouge qu'il ne faut pas utiliser sa balise pour choisir des mots de passe. La recommandation est tout à fait justifiée et s'applique à tous les générateurs publics d'aléas. En effet, les données publiées restent accessibles (c'est nécessaire pour mener des contrôles *a posteriori*), ce qui signifie que si votre mot de passe a été engendré par une balise publique, quelqu'un cherchant à le deviner a accès aux mêmes nombres aléatoires, ce qui facilite son travail. Pour vos mots de passe, utilisez des générateurs privés et si possible indépendants de votre machine!

D'autres sites se présentent comme des balises aléatoires sécurisées et réussissent à faire mieux que la balise du NIST.

Le Laboratoire de sécurité computationnelle et de cryptographie appliquée de l'université du Chili (https://random.uchile.cl/en/) propose une balise dont les bits dépendent à la fois d'une source quantique locale et de données extérieures (données sismiques, radio de l'université du Chili, blockchain Ethereum, Twitter). Ce site est l'une des meilleures balises actuelles: il utilise le principe décrit plus loin et dans l'encadré 3.

Le site brésilien d'Inmetro, l'Institut national de la métrologie et de la qualité (www4.inmetro.gov.br/) et le projet collaboratif *League of entropy* (www.cloudflare.com/leagueofentropy/) offrent divers outils de génération de hasard privé et public. Là encore, ils s'appuient >

AUTHENTIFICATION SANS CONSERVATION DES MOTS DE PASSE

Première connexion : enregistrement de l'identifiant de l'utilisateur et de l'empreinte du mot de passe

Jean-Paul choisit le mot de passe JP1234 et le communique au serveur

Le serveur calcule l'empreinte du mot de passe reçu, trouve X2e9qq92 et mémorise le couple Jean-Paul/X2e9qq92.

Connexion ultérieure : vérification par le serveur que celui qui se connecte est la bonne personne

Jean-Paul se connecte au serveur et indique son identifiant ainsi que son mot de passe

Le serveur calcule l'empreinte du mot de passe reçu et vérifie qu'elle est bien la même que celle du couple qu'il a enregistré Jean-Paul/X2e9qq92

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