> le 30 novembre 2019 à midi, qui donnerait un 0, donc *pile*; 3,5 °C pour le 1er décembre 2019 à midi, qui donnerait un 1, donc *face*; 5,2 °C pour le 2 décembre 2019 à midi, qui donnerait un 0, donc *pile*. La suite des tirages est donc, à partir du 28 novembre: *pile-face-pile-face-pile*. On peut opérer un tirage à pile ou face par téléphone tous les jours à midi.

Cette balise est très probablement aléatoire: il n'y a aucune raison de penser que les nombres pairs (ou impairs) soient favorisés, ni qu'aucune séquence particulière de *pile* ou *face* soit plus fréquente qu'une autre de même longueur. Elle est bien imprévisible, car les prévisions météorologiques pour Paris ne sont pas assez précises pour qu'on puisse savoir à l'avance ce qui va venir comme premier chiffre après la virgule de la température à midi. La balise est vérifiable *a posteriori*: tout le monde peut aller consulter le site mentionné pour voir ce qui y est indiqué. On peut raisonnablement faire confiance au sérieux des opérateurs du site pour qu'ils ne changent pas les données inscrites et qu'ils les maintiennent accessibles assez longtemps.

Trois problèmes persistent cependant avec cet aléa public.

(1) Prévoir le résultat du tirage devient possible quand on s'approche de midi, car la température varie continûment. Il faudrait donc convenir au moins quelques heures avant midi que l'on va jouer à pile ou face et attendre midi pour que s'effectue le tirage.

(2) Le procédé n'est pas rapide. En considérant d'autres lieux de relevé, d'autres heures, d'autres données (pression, hygrométrie, etc.) on pourrait faire mieux, mais on resterait limité à quelques bits par minute au plus.

(3) Peut-on faire totalement confiance à ce qui est publié par le site de Météo-France? Est-ce qu'il ne se pourrait pas qu'un employé, par négligence, erreur de manipulation ou soudoyé par l'un des joueurs, modifie le relevé publié pour produire des températures autres que celles vraiment relevées? L'un des joueurs ne pourrait-il pas s'introduire sur le site et en modifier les chiffres publiés? Nous verrons par la suite que nous pouvons pallier cette incertitude sur la stabilité et la sécurité de ce qui est inscrit.

Les cours de bourses, les taux de change, les résultats sportifs, le contenu de ce qui est publié par les médias (journaux, radio, télévisions, pages web), sont d'autres sources publiques d'aléas et vérifiables *a posteriori*. Certaines sont assez rapides, mais aucune ne donne une assurance de stabilité supérieure aux données publiées par Météo-France. Avec elles, on aura donc en définitive un aléa public ayant les mêmes imperfections qu'avec Météo-France.

AUTRES SOURCES PARTAGEABLES D'ALÉAS

Conscients de l'intérêt de disposer de meilleures sources publiques d'aléas, certains sites internet en proposent. Ainsi, le site random.org rend disponible des nombres aléatoires de toutes sortes depuis 1998.

Ce site, conçu et réalisé par Mads Haahr, du Trinity College, à Dublin, propose des nombres aléatoires qu'il qualifie de «véritables», car

2 LES FONCTIONS DE HACHAGE

Une fonction de hachage est une fonction qui, à un x donné (une suite de caractères ou un nombre) de longueur quelconque, associe une valeur $f(x)$ de taille fixe, l'« empreinte de x ». Le passage de x à $f(x)$ se fait par une série d'opérations qui commencent en découpant x en morceaux (d'où le nom de hachage). Ces morceaux sont rassemblés et emmêlés de façon à obtenir un brouillage aussi parfait que possible de l'information présente dans x. Pour la fonction classique de hachage SHA-256, la longueur de $f(x)$ est de 256 bits.

Les fonctions de hachage à usage cryptographique, comme SHA-256, ont les propriétés suivantes.
(a) Elles sont « à sens unique » : le passage de x à $f(x)$ est un calcul rapide, mais pour un y donné, trouver x tel que $y = f(x)$ est impossible en pratique.
(b) Il est impossible en pratique de trouver x et x' différents avec $f(x) = f(x')$.
(c) Elles produisent des valeurs assimilables à des valeurs aléatoires uniformes 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 $f(x')$ sont sans aucun lien apparent.

Exemple simple. On transforme la donnée x en une suite de 0 et de 1 que l'on découpe 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 centraux :
x = 00010010100101010100100101
→ 00010010 + 10010101 + 00100101 = 11001100
→ $f(x)$ = 0011.
Cet exemple est bien sûr trop simple pour avoir les propriétés (a), (b) et (c).

Exemples d'utilisation.
– Authentifier un utilisateur sans garder son mot de passe *(voir l'illustration ci-contre)*.
– Assurer l'intégrité d'un fichier x. Si l'on dépose un gros fichier sur un serveur, 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, laquelle doit être la même que celle publiée sur le site.
– Engagement d'une donnée sans la révéler. Vous choisissez un nombre que vous ne devez plus changer car, par exemple, vous allez participer à une enchère simultanée ; vous calculez son empreinte et la communiquez à l'organisateur. Le site d'enchère recueille les empreintes des autres participants. Les empreintes sont toutes rendues publiques et ceux qui les ont proposées ne peuvent donc plus rien changer. À cet instant, personne ne connaît les enchères proposées (car f est impossible à inverser), et donc, même si les soumissions ne sont pas simultanées, ceux qui participent ignorent les propositions des autres. La phase d'enchère terminée, chacun révèle son nombre. Le calcul des empreintes des nombres révélés permet de vérifier qu'il n'y a pas eu de changements. Tout s'est donc déroulé comme si chacun avait écrit sa proposition sur un papier et avait placé celui-ci dans une enveloppe cachetée qui n'a été ouverte qu'au dernier moment.

En cryptographie, les fonctions de hachage sont conçues avec soin et soumises à des tests variés. Des compétitions pour concevoir les nouvelles générations de fonctions de hachage sont organisées. Ainsi est née la fonction SHA-3, disponible depuis 2015 (voir https://en.wikipedia.org/wiki/SHA-3).

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

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/

L'AUTEUR

LOÏC MANGIN
rédacteur en chef adjoint
à *Pour la Science*

OÙ SE CACHE LE TRIFLUORORMÉTHYLE ?

Un artiste-plasticien rend hommage à un nouveau type de réaction chimique, respectueux de l'environnement, qui aidera à préparer des composés inédits comportant un groupe trifluorométhyle, souvent intéressant en pharmacologie.

Q u'ont en commun l'antidépresseur Prozac, l'antipaludéen Lariam, l'anti-VIH Sustiva… ? Outre leur effet dirigé contre quelque chose, leurs principes actifs (respectivement fluoxétine, méfloquine, éfavirenz et fipronil) sont tous dotés d'un groupe chimique trifluorométhyle (CF_3). À la liste précédente, on aurait pu ajouter un autre « produit » qui partage cette particularité : une œuvre récente (*voir page ci-contre*) de Patrick Barrès, enseignant-chercheur au Laboratoire de recherche en audiovisuel – savoirs, praxis et poïétiques en art, à l'université de Toulouse ! Où est le trifluorométhyle ?

Tout commence dans les laboratoires de Géraldine Masson, de l'Institut de chimie des substances naturelles, à l'université Paris-Saclay, et celui d'Emmanuel Magnier, de l'institut Lavoisier, à l'université de Versailles-Saint-Quentin. Avec leurs collègues, ils ont récemment mis au point une méthode de synthèse de composés trifluorométhylés, à l'aide de réactions plus respectueuses de l'environnement.

La piste suivie est celle de la photocatalyse d'oxydo-réduction dont l'un des pionniers fût, au début des années 1980, par Alain Deronzier, de l'université de Grenoble. L'idée est d'utiliser un photocatalyseur qui, éclairé, passe dans un état excité et peut alors donner ou recevoir un électron et ainsi démarrer une réaction, voire plusieurs, et dans le cas qui nous intéresse un processus dit «radicalaire» (un seul électron est engagé).

Patrick Barrès s'est emparé du résultat des chimistes et a souhaité le représenter symboliquement. De fait, les quatre zones colorées correspondent à autant de réactifs mis en jeu: des alkènes, c'est-à-dire des molécules carbonées à double liaison (*en bleu*); un composé carbonylé, comportant une double liaison entre un atome de carbone et un d'oxygène (*en vert*); le triméthylsilyl azide $(CH_3)_3SiN_3$, noté $TMSN_3$ (*en jaune*); enfin, le réactif d'Umemoto qui apporte le groupe trifluorométhyle (*en rouge*). La spirale découpée brouille les cartes et, en apportant une dynamique à l'œuvre, renvoie à la réaction elle-même. Quelle est-elle?

Grâce au photocatalyseur, ici le $[Ru(bpy)_3(PF_6)_2]$, le rouge et le bleu s'associent, le jaune et le vert font de même, et

© Patrick Barrès