Sur quelques évolutions récentes en cryptologie post-quantique

Franck Leprévost [1]

La cryptologie est aussi ancienne que le besoin de garder ses secrets à l'abri d'une part, et d'avoir accès à ceux de son voisin d'autre part. Le fait que ce dernier soit considéré comme ennemi, ou comme ami joue peu lorsqu'on regarde l'histoire du domaine. On consultera avec profit l'excellent ouvrage [6], et on gardera à l'esprit les rapports pour le Parlement Européen sur le réseau Echelon [2] au tournant des années 2000, l'affaire Snowden, ou encore les écoutes de dirigeants politiques ou de journalistes européens.

Ce domaine scientifique est schématiquement constitué de deux composantes aux objectifs opposés (les spécialistes pardonneront la grande simplification des concepts dont cet article fait preuve tout du long). D'un côté la cryptographie s'attache au développement de méthodes visant à assurer la confidentialité de messages ainsi que leur authenticité ou leur intégrité. De l'autre la cryptanalyse cherche à casser les codes employés afin de découvrir le contenu des messages chiffrés, à impersonnaliser les messages, ou à altérer leur contenu plus ou moins subrepticement.

Ces deux directions ont connu d'importantes évolutions au cours des siècles, mais avec des vitesses variables. L'objectif ici n'est pas de les relater toutes, l'ouvrage [6]

1. Ancien chercheur et professeur en mathématiques pures au CNRS, à l'université Joseph Fourier, au Max Planck-Institut für Mathematik à Bonn, à la Technische Universität Berlin (Allemagne), co-auteur du rapport dit « Echelon » pour le Parlement européen, Franck Leprévost est actuellement professeur d'informatique à l'université du Luxembourg, dont il a été le vice-président de 2005 à 2015. Il y dirige le laboratoire d'algorithmique, cryptologie et sécurité et est l'auteur de tutoriels de mathématiques pour l'informatique.

2. https://fr.wikipedia.org/wiki/Echelon.

DOI:10.48556/SIF.1024.21.185

permettant de se faire une idée sur les principales jusqu'aux années 1970 essentiellement, mais de se focaliser sur quelques moments charnières de cette histoire. L'attention est portée sur eux parce qu'ils créent d'une part un avant et un après sans retour puisque la science est incrémentale par essence, et que sa capacité à l'oubli est limitée, et, d'autre part, parce que leur concomitance illustre une accélération de l'état des connaissances.

La cryptographie jusqu'aux années 1970

Jusqu'aux années 1970 — c'est-à-dire sur une période couvrant plus de 3000 ans — le principe utilisé par les cryptosystèmes les plus performants résidait dans l'utilisation d'une clef secrète, servant aussi bien à chiffrer un message qu'à le déchiffrer. C'est essentiellement ce principe qui prévalait par exemple pour la machine Enigma utilisée par les forces allemandes durant la deuxième guerre mondiale. Les clefs secrètes étaient changées fréquemment afin de se prémunir contre les attaques éventuelles du système. Le film *The Imitation Game* [3], donne une idée de cette histoire, qui d'ailleurs rejoint la tragique histoire personnelle du mathématicien Alan Turing (1912 – 1954). La recherche de cryptosystèmes à clef secrète performants s'est poursuivie après la guerre, donnant lieu notamment à la publication du *Data Encryption Standard* (DES) en 1977 par le *National Institute of Science and Technology* (NIST), un organisme que nous retrouverons dans la suite de cet article.

Deux choses sont à noter à cet égard. La première est que, pour la première fois de manière aussi franche, le fonctionnement interne d'un algorithme destiné à devenir un standard était révélé à tous. Cela peut paraître surprenant car le monde du secret est presque ontologiquement allergique à ce type d'approche ouverte. Cependant, en procédant ainsi, le NIST appliquait les recommandations de Kerckhoff (1835 – 1903) selon lesquelles il faut partir du principe que l'ennemi connaît le système (ou le connaîtra), et donc que le secret ne doit résider que dans la clef, et pas dans le maintien dans l'obscurité du cryptosystème utilisé. La seconde est que l'adoubement par le NIST a fait *de facto* du DES le standard des algorithmes de cryptographie à clef secrète à compter de cette date non seulement aux USA, mais quasiment partout sur terre.

Cette domination du DES a duré presque une vingtaine d'années. Toutefois, les recherches se sont démultipliées, donnant lieu à des attaques de plus en plus fortes sur le DES. Le Triple-DES (une variante de DES) a alors été standardisé, ses tailles de clefs plus importantes permettant de résister encore pour un temps à ces attaques. Cependant, le coup d'arrêt est venu en 1999 avec la publication et l'utilisation du DES-Cracker. Cette machine avait été spécifiquement construite pour trouver les clefs secrètes du DES, et parvenait à le faire pour une poignée de dollars (un million de l'époque) et une poignée de jours (deux ou trois en moyenne). Il devint clair que

3. https://fr.wikipedia.org/wiki/Imitation_Game.

les jours du DES ou du Triple-DES étaient comptés : le problème n'était plus seulement la taille devenue trop courte des clefs, mais résidait dans la conception même de ces algorithmes. Les avancées technologiques, l'augmentation exponentielle à l'époque de la puissance des ordinateurs (avec un doublement de la capacité des puces tous les 18 mois) faisaient que les deux ou trois jours nécessaires en moyenne pour trouver une seule clef du DES au moment de l'avènement du DES-Cracker seraient fortement réduits dans les années qui viendraient. Maintenir les choses en l'état laissait planer un risque trop grand sur la confidentialité des communications futures. Ouvrons une parenthèse en notant au passage que l'avènement du DES-Cracker a certainement eu des conséquences sur les communications passées : il est en effet probable que des instances gouvernementales aient stocké au fur et à mesure des années des messages, chiffrés par d'autres États avec le DES ou le Triple-DES, en attendant des jours meilleurs. Même si ces messages n'avaient d'impact que pour des périodes courtes, avoir soudain la possibilité de les lire grâce au DES-Cracker, ou à d'autres machines, a certainement été source de confirmation d'informations ou source de surprises... Quoi qu'il en soit, le NIST avait toutes les raisons de lancer alors un appel à candidatures pour le successeur du DES, l'*Advanced Encryption Standard* (AES), avec des critères plus sévères que ceux ayant prévalu pour le DES, et tenant compte des progrès de la science et ceux prévisibles (à l'époque) de la technologie. L'algorithme à clef secrète gagnant de la compétition fut standardisé en 2001.

Premier moment charnière : l'invention de la cryptographie à clef publique

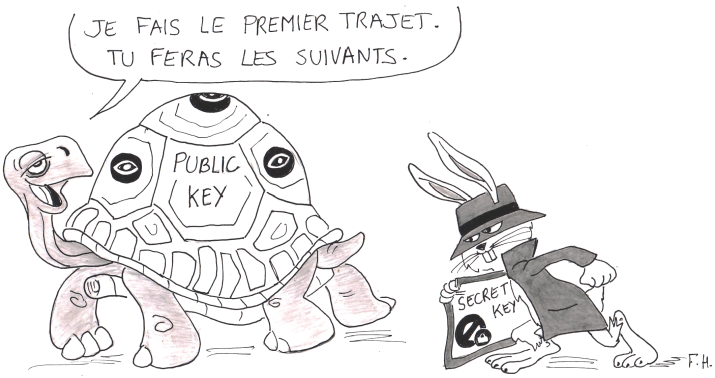
Mais le véritable premier moment charnière de cette histoire ne vient pas de la standardisation du DES ou de l'AES, qui s'inscrivent, d'une certaine manière, dans la lignée de la cryptographie à clef secrète ayant prévalu au cours des millénaires précédents. Il vient de l'invention de la cryptographie à clef publique. Dans un article de 1976 cité plus de 8000 fois depuis sa parution, l'informaticien Diffie et le mathématicien Hellman en posaient la base [4]. L'idée est d'une simplicité biblique. Pourtant il aura fallu des millénaires pour qu'elle émerge. Elle revient à la question suivante : a-t-on nécessairement besoin de la même clef pour chiffrer que pour déchiffrer ? Les cryptosystèmes comme Enigma, DES, AES sont à clef secrète. Le principe de ces algorithmes et machines est que la même clef est utilisée pour ces deux opérations. Elle est connue de l'envoyeur et du destinataire, et c'est tout. Toute autre personne, notamment un attaquant, ne doit pas y avoir accès. Dans le cas contraire, la sécurité des messages est compromise. Maintenant imaginons avec Diffie et Hellman qu'il existe deux clefs par destinataire : une clef publique accessible à tous (publiée par exemple comme le sont les numéros de téléphone dans un annuaire), et une clef

privée connue de cette seule personne. Imaginons encore qu'Alice souhaite communiquer confidentiellement avec Bob (Alice et Bob font partie du jargon standard utilisé en cryptographie), et donc que Bob soit le destinataire auquel Alice envoie un message. Les deux clefs de Bob doivent être conçues de sorte que tout le monde puisse lui envoyer un message chiffré avec sa clef publique (celle qui figure en face du nom de Bob dans l'annuaire), mais que seul Bob puisse déchiffrer le message avec sa clef privée (connue de lui seul). Ce concept novateur, qui suppose évidemment un lien entre clef privée et clef publique, mais qui n'impose plus que ce lien soit l'égalité entre ces clefs (ce concept est clarifié un peu plus loin), est fondamental pour toute la cryptographie qui suivra. Il sera notamment rendu opérationnel dans des travaux de Rivest, Shamir et Adleman, alors au MIT, menant à ce qui s'appellera ensuite l'algorithme RSA d'après les initiales de leurs noms. RSA peut être utilisé aussi bien comme algorithme de chiffrement à clef publique que comme algorithme de signature électronique [10].

Avant de tirer les conséquences de ces découvertes, mentionnons un fait peu répandu mais significatif de l'environnement dans lequel évoluent ces sujets sensibles. D'une certaine manière, ce qui précède est la partie émergée de l'histoire. Il existe une partie beaucoup plus discrète[4] [14]. En effet, tout porte à croire que les britanniques James Ellis, Cliff Cocks et Malcolm Williamson avaient découvert l'essentiel des principes et algorithmes de la cryptographie à clef publique dès 1975 et donc avant quiconque. La raison d'État a imposé le silence sur près d'un quart de siècle, et ce fait n'a été révélé qu'en 1997. En effet, Ellis, Cocks et Williamson travaillaient pour le *Government Communications Headquarters* (GCHQ) à Cheltenham. Le GCHQ était (et est) le successeur de l'établissement secret de Bletchley Park, où Alan Turing travaillait pendant la guerre sur la machine Enigma. Dans ce milieu, le secret des travaux peut prévaloir (notamment sur le principe de Kerckhoff cité plus haut), que l'on soit en temps de guerre active, de guerre froide, ou autre.

Revenons aux conséquences de l'avènement de la cryptographie à clef publique. On aurait pu penser que ces découvertes mettraient au rebut les cryptosystèmes à clef secrète comme DES, Triple DES ou l'AES. En effet, les cryptosystèmes à clef secrète présentent intrinsèquement beaucoup de problèmes : ils ne permettent pas la signature électronique ; les messages sont compromis lorsqu'une clef est perdue (ou volée) ; ils présentent des difficultés pour l'échange des clefs secrètes sur lesquelles les différentes parties doivent convenir ; ils présentent des difficultés lorsqu'il faut inclure un nouveau participant dans une discussion chiffrée, ou au contraire l'en exclure ; etc. La plupart de ces problèmes disparaissent, ou du moins s'atténuent, avec les algorithmes à clef publique pour des raisons un peu longues à expliquer ici. Pourtant, les cryptosystèmes à clef secrète restent utilisés aujourd'hui pour plusieurs raisons. Pour n'en citer qu'une : la rapidité. Les cryptosystèmes à clef secrète sont

4. https://cryptome.org/ukpk-alt.htm.



plus rapides que ceux à clef publique : ils encryptent et décryptent jusqu'à parfois 100 fois plus vite. La situation est donc la suivante : d'une part, les algorithmes de cryptographie à clef secrète sont rapides, mais leur mise en place et leur gestion posent des problèmes importants ; d'autre part, les algorithmes à clef publique résolvent ces problèmes de mise en place et de gestion et font des choses en plus (signature électronique), mais sont lents.

La solution pour sortir de cette alternative est précisément de ne pas en sortir. Il convient en effet de ne pas concevoir les algorithmes de cryptographie à clef secrète et les algorithmes de cryptographie à clef publique comme étant exclusifs l'un de l'autre, mais au contraire de les combiner.

Par exemple, Alice souhaitant communiquer fréquemment avec Bob commencera par récupérer la clef publique de Bob. Puis, Alice construira un message M assez court — pour schématiser encore, imaginez quelque chose comme « *les sanglots longs des violons de l'automne*[5] » — et l'enverra chiffré à Bob, avec la clef publique de Bob bien entendu. De son côté, Bob pourra déchiffrer avec sa clef privée le message M. À ce stade, Alice et Bob connaissent la phrase « *les sanglots longs des violons de l'automne* » : Alice parce qu'elle l'a envoyée, et Bob parce qu'il l'a déchiffrée. Ils sont en outre les seuls à la connaître car toute autre personne ne voit passer qu'un message chiffré entre Alice et Bob. Imaginons que Alice et Bob aient convenu d'utiliser cette phrase connue d'eux seuls comme la clef secrète d'un algorithme à clef secrète, comme AES par exemple, sur lequel ils se sont aussi mis d'accord au préalable, et de l'utiliser pour tous les échanges de la journée. À partir de ce moment et jusqu'à la fin de la journée, aussi bien Alice que Bob n'utilisent plus que l'algorithme à clef secrète AES et la clef secrète commune dont ils disposent (« *les sanglots longs des violons de l'automne* »). Cela leur permet d'échanger beaucoup de messages très vite, grâce à l'AES. Ils ne se servent plus de l'algorithme à

5. https://fr.wikipedia.org/wiki/Chanson_d%27automne.

clef publique du tout de la journée, et ce jusqu'au lendemain matin, moment où ils recommenceront leurs échanges pour convenir d'une nouvelle clef secrète commune avec une autre phrase plus ou moins poétique. Ce changement journalier ou même plus fréquent qu'une fois par jour de la clef secrète d'un algorithme comme AES renforce la sécurité de l'ensemble de l'édifice ainsi mis en place. Dans leur article, Diffie-Hellman proposent une autre manière de procéder, dans laquelle Alice et Bob construisent conjointement la clef secrète M (à la différence du cas précédent, où Alice décide unilatéralement de la clef secrète, qu'elle envoie ensuite à Bob). Ce protocole d'échange de clef de Diffie-Hellman joue un rôle très important dans la suite de l'histoire que nous racontons.

Les méthodes ainsi esquissées, combinant algorithme de cryptographie à clef publique (comme par exemple RSA ou le protocole d'échange de clefs de Diffie-Hellman) et algorithme de cryptographie à clef secrète (comme par exemple AES) — ce qui fait au total une clef publique et une clef privée pour le premier algorithme, et une clef secrète pour le second algorithme — sont, pour l'essentiel, celles qui gouvernent les infrastructures à clef publique utilisées dans le monde aujourd'hui (par exemple LuxTrust [6] au Luxembourg, entreprise dont l'auteur de ces lignes a été membre du conseil d'administration pendant pas loin d'une dizaine d'années). La cryptographie à clef publique a connu un essor commercial, industriel, militaire, étatique colossal dans les années qui suivirent son invention. Ces architectures ont intégré de nombreuses applications. Les plus familières au grand public sont certainement celles ayant trait à l'e-commerce, aux déclarations fiscales en ligne, à l'e-banking pour n'en citer que quelques-unes. Les algorithmes de cryptographie à clef publique sont parmi les briques de base les plus importantes de la chaîne de sécurité des communications. Si ces algorithmes ont des failles, alors toute la chaîne de sécurité s'effondre comme un château de cartes.

Des problèmes mathématiques garants de la sécurité des cryptosystèmes à clef publique

Comment mesure-t-on la solidité de ces briques de base ? Le cœur de la chose réside dans le lien entre clef privée et clef publique des algorithmes de cryptologie à clef publique. Comme signalé plus haut, ce lien n'est pas un signe = comme dans le cas de la cryptographie à clef secrète. Le lien est donné par des opérations mathématiques plus compliquées, et par des problèmes mathématiques complexes à résoudre en pratique.

Ces opérations portant sur des objets mathématiques ont, pour l'essentiel, deux propriétés : elles doivent être « faciles » à faire dans un sens, et « difficiles » ou « très difficiles » à faire en sens inverse. Par exemple, il est facile (et donc rapide) de multiplier l'entier $x = 13$ par l'entier $y = 17$ pour obtenir l'entier $z = 221$. Par contre,

6. https://www.luxtrust.com.

si l'on ne connait que la valeur de l'entier z, il est en pratique plus difficile (et donc plus lent) de retrouver les valeurs des entiers x et y qui le composent (il suffit d'essayer de se faire la main sur un autre exemple, tel $z = 437$). Le problème sous-jacent est le problème de la factorisation des entiers. Le protocole d'échange de clef de Diffie-Hellman repose sur d'autres problèmes mathématiques, impliquant des structures plus compliquées que les entiers, au nombre desquelles figurent les courbes elliptiques. Ici n'est pas le lieu de décrire ce que sont les courbes elliptiques. Toutefois, comme elles reviennent un peu plus tard dans cette histoire, il est nécessaire de citer leur nom.

Mais que signifie « facile », « difficile », voire « très difficile » ? Il est possible d'en donner une définition assez précise en termes de classes de complexité de problèmes, ou plutôt d'algorithmes destinés à résoudre ces problèmes. Disons pour simplifier qu'un problème résolu par un algorithme en temps polynomial en la taille des données est facile, qu'un problème dont le meilleur algorithme connu pour le résoudre nécessite un temps exponentiel est très difficile, et qu'un problème pour lequel le meilleur algorithme connu le résolvant nécessite un temps plus que polynomial mais moins qu'exponentiel (on parle alors de temps sous-exponentiel) est difficile. Le problème de la factorisation des entiers illustré précédemment est difficile. Ce problème spécifique est à l'origine de l'algorithme RSA, utilisé des centaines de millions de fois par jour tant pour la cryptographie à clef publique que pour les signatures électroniques. D'autres problèmes mathématiques ont donné naissance à d'autres algorithmes de chiffrement ou de signature électronique. Notamment les problèmes dits du logarithme discret dans des groupes (au sens mathématique du terme) bien choisis sous-tendent le protocole d'échange de clefs de Diffie-Hellman dans ces mêmes groupes. Les algorithmes de cryptage ou de signature électronique construits sur ces problèmes ont fait l'objet, tout comme RSA, d'un effort mondial de standardisation par différentes organisations. Le domaine a gagné en maturité, les standards restant pour l'essentiel stables tant qu'aucune attaque sérieuse n'est découverte. Plus généralement, l'évolution des standards mondiaux s'effectue en adaptant les recommandations au fur et à mesure de la découverte de cas particuliers de faiblesses (auquel cas des paramètres sont modifiés pour ne pas rentrer dans ces cas de faiblesses ponctuelles). Cette évolution se fait aussi en tenant compte de la montée en puissance des capacités de calcul des ordinateurs. En effet, tout comme on peut imaginer qu'un cambrioleur qui aurait à sa disposition l'ensemble des clefs de la création finirait par rentrer chez vous en les essayant toutes les unes après les autres, un attaquant peut essayer d'attaquer un cryptosystème par ce qu'on appelle *force brute* dans le jargon des spécialistes. Il essaye toutes les combinaisons possibles de clefs jusqu'à ce qu'il trouve la bonne. L'attaque par force brute marche toujours. Avec beaucoup de chance, la première tentative peut d'ailleurs être la bonne. Bien entendu, la question cruciale est le temps nécessaire en moyenne pour trouver cette clef. Avec l'augmentation des capacités de calcul mondial, ne rien faire expose à

des attaques par force brute de plus en plus faciles et de moins en moins onéreuses. Toutefois, l'amélioration des processeurs permet aussi aux cryptosystèmes de remplir leurs missions plus vite, et donc autorisent des tailles de clefs plus grandes pour contrer les attaques par force brute rendues possibles par la puissance croissante des ordinateurs. Par conséquent, les recommandations s'attachent à maintenir un compromis entre l'efficacité du cryptosystème d'un côté, et la faisabilité des attaques de l'autre, en augmentant la taille des clefs périodiquement.

Donc, avec les ordinateurs tels que nous les connaissons actuellement, les algorithmes de cryptographie à clef publique standardisés tels que RSA, ou ceux dérivés des problèmes de logarithme discret dans des groupes bien choisis, dont ceux provenant de courbes elliptiques, continuent d'être utilisés avec des tailles de clefs qui augmentent de temps en temps. La raison de leur pérennité vient de ce que les meilleurs algorithmes pour casser les problèmes mathématiques, sur lesquels reposent leur sécurité, sont en temps sous-exponentiel ou exponentiel. Autrement dit, ces problèmes de factorisation ou de résolution du logarithme discret dans des groupes adaptés sont difficiles ou très difficiles en l'état actuel des connaissances sur les ordinateurs classiques. Un point supplémentaire au passage : de la même manière qu'il n'est guère prudent de mettre tous ses œufs dans le même panier, la communauté cryptographique apprécie d'avoir des algorithmes de cryptage de natures différentes pour réduire l'exposition des utilisateurs aux attaques : une attaque sur un système est en général sans impact sur un autre système dont la sécurité est basée sur un problème mathématique différent.

Deuxième moment charnière : les algorithmes de Shor

Cependant, des efforts très importants sont déployés depuis quelques années pour construire des ordinateurs quantiques (voir plus loin sur l'état actuel de l'art). Si on y arrive, les capacités de calcul de tels ordinateurs seront démultipliées par rapport à celles des ordinateurs actuels. Une ligne de recherche concerne la question de ce que l'on pourrait faire avec de tels ordinateurs. Plus précisément, il s'est agi de construire des algorithmes qui exploitent les caractéristiques quantiques de telles machines, alors même que les machines n'existaient pas (et que l'on ne savait absolument pas si on pourrait en construire un jour). Dans ce contexte, Peter Shor publia à la fin des années 1990 des algorithmes qui résolvent en temps polynomial les problèmes mathématiques sous-jacents aux cryptosystèmes à clef publique usuels — factorisation des entiers, logarithme discret sur les groupes appropriés – sous l'hypothèse d'avoir à disposition un ordinateur quantique performant [13] (voir [8] pour une vulgarisation sur ces algorithmes, voir également [15, 1]). La conséquence est que ces méthodes de Shor rendraient caduques les cryptosystèmes correspondants : RSA, Diffie-Hellman sur le groupe multiplicatif d'un corps fini ou sur le

groupe des points d'une courbe elliptique sur un corps fini. Notons au passage que les efforts nécessaires pour casser ces cryptosystèmes à l'aide des algorithmes de Shor sur un ordinateur quantique peuvent différer de l'un à l'autre [12]. Les algorithmes de Shor et le fait que des investissements substantiels étaient effectués pour tenter de construire des ordinateurs quantiques ont conduit le NIST à se soucier de la sécurité des communications dans un monde post-quantique.

En 2016, le NIST a publié un appel à candidature [7] pour des cryptosystèmes permettant l'encryption, l'échange de clefs, et les signatures électroniques, qui résisteraient à l'existence d'un ordinateur quantique, autrement dit des systèmes basés sur des problèmes mathématiques qui ne s'exposent notamment pas aux attaques du type de celles de Shor. Le processus de sélection devant mener aux nouveaux standards s'est fait par étapes (comme toujours). Parmi les 69 candidatures reçues, le premier tour d'évaluation (décembre 2017 – janvier 2019) a conduit le NIST à retenir 26 propositions pour le deuxième tour. Celui-ci s'est poursuivi jusqu'au mois de juillet 2020. Il a permis au NIST de sélectionner 7 finalistes et 8 candidats alternatifs pouvant participer au troisième tour. Le 5 juillet 2022 est une date importante dans notre narration : ce jour-là, le NIST publia son rapport d'évaluation sur ces 15 propositions [8]. Ses conclusions furent que l'algorithme à clef publique et l'algorithme d'échange de clefs qui serait standardisé serait CRYSTALS-Kyber, et que les trois algorithmes de signature électronique CRYSTALS-Dilithium, Falcon et SPHINCS+ seraient aussi standardisés. Au-delà de ces décisions, le NIST sélectionna aussi quatre algorithmes d'échange de clefs pour un quatrième tour d'évaluation — BIKE, Classic McEliece, HQC et SIKE — et annonça que ces algorithmes pourraient aussi être pris en considération pour une standardisation future. Pour être complet, notons que ce quatrième tour est aussi ouvert à de nouvelles propositions.

Troisième moment charnière : la fin de SIDH

Comme on l'imagine aisément, comprendre comment fonctionnent ces algorithmes, quels sont leurs avantages respectifs, et comment ils se comparent les uns aux autres nécessite un certain effort, dont nous faisons l'économie ici. Mettons toutefois l'accent sur l'un d'entre eux, SIKE, retenu début juillet 2022 pour le quatrième tour d'évaluation et donc susceptible de faire partie des standards (les autres algorithmes retenus ne font pas l'objet du présent article). Les mathématiques sur lesquelles cet algorithme s'appuie portent sur les courbes elliptiques sur des corps finis. De tels objets ont déjà démontré leur puissance par le passé : elles ont

7. https://csrc.nist.gov/News/2016/Public-Key-Post-Quantum-Cryptographic-Algorithms.

8. https://www.nist.gov/publications/status-report-third-round-nist-post-quantum-cryptography-standardization-process.

été utilisées pour prouver la conjecture de Fermat ; elles le sont aujourd'hui pour prouver que des nombres entiers de plusieurs milliers de chiffres sont premiers (pour donner une idée de comparaison, on estime que le nombre de particules dans l'univers observable est de l'ordre d'un nombre ayant au plus 81 chiffres : bien petit à l'égard des milliers de chiffres dont il est question plus haut). Pour ce qui concerne SIKE, il s'agit en l'occurrence, en termes techniques, d'un cryptosystème qui formalise le protocole *Supersingular Isogeny Diffie-Hellman* (SIDH). Le SIDH consiste essentiellement à prendre en compte deux courbes elliptiques et à créer un algorithme à clef publique dont la clef privée est constituée par une application entre ces deux courbes et compatible avec leurs structures algébriques, appelée une isogénie. Ce sujet de recherche, qui s'intègre dans l'ensemble plus vaste de la cryptographie à base de courbes elliptiques, est devenu très actif depuis une bonne quinzaine d'années, donnant lieu à des dizaines de conférences internationales et des centaines d'articles scientifiques.

Quelques jours après la publication du rapport du NIST, trois articles parurent coup sur coup, qui eurent un certain retentissement dans le petit monde de la cryptographie. Le 30 juillet, Castryck et Decru publièrent le premier de la série [3]. De manière indépendante, Maino et Martindale arrivaient à des conclusions comparables le 8 août [9]. Deux jours plus tard, Robert apportait le coup de grâce [11]. Au final, SIDH est cassé en temps polynomial. Il semble donc que le protocole SIDH ait vécu, et qu'il emporte avec lui SIKE. En effet, les auteurs des articles précités ont montré comment casser ce candidat à la standardisation post-quantique sur un ordinateur classique : moins d'une heure de calcul sur un ordinateur portable suffit. Comment en est-on arrivé là ? Dans les trois cas, le point essentiel utilisé par les auteurs est une technique développée par Kani en 1997 pour « coller et séparer » des courbes elliptiques entre elles [7]. Dans un travail commun avec Howe et Poonen [5], nous avions écrit précisément les formules permettant de « recoller » un produit de courbes en une jacobienne de courbe de genre plus élevé. Les auteurs des trois papiers publiés en 2022 utilisent ces considérations très techniques avec beaucoup d'élégance et d'efficacité pour retrouver la clef privée du protocole.

Ouvrons une parenthèse sous la forme d'une méta-remarque et d'une observation menant à une conjecture.

La méta-remarque est que plus un objet mathématique possède de structures algébriques, plus grandes sont les chances de le cerner. Les courbes elliptiques supersingulières, utilisées dans SIDH, offrent des structures plus riches que les autres courbes elliptiques. Ce sont précisément ces structures additionnelles, se traduisant par le S dans SIDH, qui ont été exploitées dans les attaques. SIDH est complètement cassé, et probablement irrécupérable pour les applications cryptographiques, car son design nécessite précisément de telles structures riches.

L'observation est que l'attaque de Shor sur les cryptosystèmes classiques utilise les propriétés quantiques des ordinateurs quantiques, alors que l'attaque sur le cryptosystème post-quantique SIDH ne s'est aucunement faite en exploitant des aspects quantiques (le paragraphe suivant arrive en soutien de cette observation). La conjecture qui se déduit de cette double observation est que des attaques sur certaines instanciations de cryptosystèmes post-quantiques, tels ceux faisant l'objet ou restant en course pour la standardisation du NIST, exploiteront, à la Shor, les capacités quantiques des ordinateurs quantiques. Dit autrement, si des algorithmes comme ceux de Shor cassent les cryptosystèmes classiques en utilisant des propriétés quantiques, on imagine aisément que des cryptosystèmes post-quantiques souffriront de l'exploitation mathématique des propriétés quantiques. Fin de la parenthèse.

Revenons à la fin de SIDH par les attaques de l'été 2022. Galbraith [9], l'un des meilleurs spécialistes mondiaux de ces sujets, positionne ces attaques dans une perspective historique par une déclaration pertinente à la fin de la page de son blog dédié à ce sujet : « *Je veux insister sur le fait que cette attaque est seulement possible en raison de décennies de recherche sur le calcul d'isogénies sur des variétés abéliennes générales. Ces travaux n'étaient pas motivés en premier lieu par leurs applications en cryptographie, mais par le désir de généralisation en mathématiques pures* ». En effet, aussi bien Kani que Howe, Poonen et moi-même (et certainement plusieurs autres chercheurs cités dans les références des articles de l'été 2022) n'avions aucunement en tête les applications qui ont émergé cet été lorsque nous travaillions sur ces objets il y a plus de vingt ans.

Cependant, c'est avec grand plaisir et même, avouons-le, une pointe de fierté (et aussi, avouons-le également, une pointe de regret, celle de ne pas avoir procédé nous-même aux attaques évoquées ci-dessus) que nous voyons (en tout cas que je vois) des travaux en mathématiques pures (si tant est qu'il y ait des mathématiques impures...) menés dans les années 2000 sans pression, sans nécessité d'applications industrielles à court terme, libres de toute autre motivation que la quête du savoir, porter leurs fruits 20 ans plus tard dans un domaine crucial pour la souveraineté des nations, la protection des citoyens et des industries, et même la préservation de la liberté.

Cette histoire illustre ainsi en premier lieu un phénomène bien connu des chercheurs, et souvent ignoré ou — pire — dénigré par nombre de décideurs en matière de politique de recherche, préoccupés de retombées immédiates. Il existe souvent un continuum de la recherche fondamentale aux applications. Mais il faut le laisser mûrir à son rythme. Cela demande de laisser les chercheurs travailler sereinement sur des défis extrêmement difficiles.

Deuxièmement, gouverner c'est prévoir. L'initiative du NIST concernant la cryptographie post-quantique a été lancée en 2016 alors que les qubits — les fameux *quantum bits*, jouant pour les ordinateurs quantiques le rôle que les bits 0 et 1 jouent

9. https://ellipticnews.wordpress.com/2022/08/12/attacks-on-sidh-sike/.

pour les ordinateurs classiques — en étaient encore à leurs balbutiements. Aujourd'hui, plus précisément en date du 11 novembre 2022, le record est détenu par la machine Osprey de IBM opérant sur 433 qubits. IBM a déjà annoncé que des travaux étaient en cours pour monter à 1000 qubits, et est en train de revoir à la hausse ses prévisions dans le domaine. Si l'émergence d'ordinateurs quantiques performants se fait encore attendre, si de nombreuses difficultés techniques restent à surmonter, notamment en ce qui concerne la correction des erreurs [2], si les machines quantiques actuelles ne permettent pas de factoriser de grands nombres (un ordinateur quantique est capable de factoriser 15 et d'obtenir 3×5, idem avec $21 = 3 \times 7$, mais guère mieux pour le moment), les progrès des dernières années sont notables et prometteurs. Si des ordinateurs quantiques performants existent un jour, leur impact ne sera pas uniquement de casser des cryptosystèmes, et de forcer à en créer d'autres. Ils permettront aussi d'aborder de très nombreux problèmes que les ordinateurs classiques actuels ne sont pas en mesure de résoudre, et cela dans des domaines variés : prévision météorologique, recherche nucléaire, médecine, science des matériaux, etc.

Enfin, et ce dernier point est moins universellement partagé à mon grand regret, de manière plus générale, la seule recherche qui compte est celle où le seul mot d'ordre est la qualité, et rien d'autre. Dans tout ce qui précède, dans l'histoire que nous avons contée, nous sommes bien éloignés des délires délétères et destructeurs *woke* sur lesquels nous nous sommes exprimés ailleurs [10]. Ces délires du *wokisme* font pourtant actuellement leur lit jusque dans les universités ou revues scientifiques les plus connues, définissent de nouveaux critères retenus dans les politiques nationales ou européennes de financement de la recherche, guident plus généralement ce que les individus, chercheurs ou pas, doivent penser, et culpabilisent ou décrédibilisent les déviants. Laisser aux tenants de l'idéologie *woke* les rênes du pouvoir équivaut à une programmation de la médiocrité, et au déni du réel, scientifique ou non.

Remerciements

L'auteur remercie A. Atashpendar, A. Biryukov, D. Mestel, L. Goubin, J.-Ch. Pomerol et P. Ryan pour leurs éclaircissements sur des aspects relatifs aux performances actuelles des ordinateurs quantiques, pour leurs indications bibliographiques, et pour leur intérêt pour ce travail.

Références

[1] Mirko Amico, Zain H. Saleem, and Muir Kumph. Experimental study of Shor's factoring algorithm using the IBM q experience. *Physical Review A*, 100(1), jul 2019.

10. https://www.causeur.fr/raison-detre-des-universites-occidentales-a-lheure-du-wokisme-225081.

[2] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. Roads towards fault-tolerant universal quantum computation. *Nature*, 549(7671) :172–179, sep 2017.

[3] Wouter Castryck and Thomas Decru. An efficient key recovery attack on SIDH (preliminary version). Cryptology ePrint Archive, Paper 2022/975, 2022. https://eprint.iacr.org/2022/975.

[4] W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6) :644–654, 1976.

[5] Everett W. Howe, Franck Leprévost, and Bjorn Poonen. Large torsion subgroups of split jacobians of curves of genus two or three. *Forum Mathematicum*, 12 :315–364, 1998.

[6] David Kahn. *The Codebreakers : The Comprehensive History of Secret Communication from Ancient Times to the Internet*. Scribner, rev sub edition, December 1996.

[7] Ernst Kani. The number of curves of genus two with elliptic differentials. *Journal für die reine und angewandte Mathematik*, 1997(485) :93–122, 1997.

[8] Franck Leprévost. Peter Shor, prix Nevanlinna 1998. *La Gazette des mathématiciens*, 81 :40–47, 1999.

[9] Luciano Maino and Chloe Martindale. An attack on SIDH with arbitrary starting curve. Cryptology ePrint Archive, Paper 2022/1026, 2022. https://eprint.iacr.org/2022/1026.

[10] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2) :120–126, feb 1978.

[11] Damien Robert. Breaking SIDH in polynomial time. Cryptology ePrint Archive, Paper 2022/1038, 2022. https://eprint.iacr.org/2022/1038.

[12] Martin Roetteler, Michael Naehrig, Krysta M. Svore, and Kristin Lauter. Quantum resource estimates for computing elliptic curve discrete logarithms. *Advances in Cryptology, ASIACRYPT 2017*, 559 :241–270, 2017.

[13] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5) :1484–1509, oct 1997.

[14] Simon Singh. *The Code Book : The Evolution of Secrecy from Mary, Queen of Scots, to Quantum Cryptography*. Doubleday, USA, 1st edition, 1999.

[15] John A. Smolin, Graeme Smith, and Alexander Vargo. Oversimplifying quantum factoring. *Nature*, 499(7457) :163–165, July 2013.