Dimension algorithmique et chiffrement post-quantique

Gabriel Chênevert,
Département d'Informatique et Mathématiques Appliquées, ISEN Lille
contact par courriel avec l'auteur
page web personnelle

Résumé

Le but de ce texte, accompagnant la conférence plénière prononcée dans la belle salle Émile-Legault du Cégep de Saint-Laurent lors du 62^{e} congrès annuel de l'AMQ, est de présenter quelques éléments de théorie de la complexité algorithmique cruciaux pour la cryptographie dans un langage géométrique original inspiré par le thème du congrès. Nous discuterons notamment des principaux algorithmes utilisés aujourd'hui en cryptographie asymétrique, de la menace que fait peser sur ceux-ci la perspective de l'arrivée de l'ordinateur quantique, ainsi que de quelques pistes pour la sécurité des données dans un éventuel monde dit *post-quantique*.

Mots clés : dimension de Hausdorff, complexité algorithmique, cryptographie, informatique quantique.

1 Dimensions

En cette année (2018) du centenaire de la formulation par Felix Hausdorff (figure 1) de la notion de dimension qui porte aujourd'hui son nom, nous pouvons nous inspirer de son approche pour proposer une version intuitive de la notion de la complexité d'un algorithme, mesure du coût d'utilisation de celui-ci du point de vue calculatoire. L'approche proposée ici n'a pas la prétension d'apporter quoi que ce soit à la théorie standard de la complexité, si ce n'est une interprétation géométrique d'exposants apparaissant dans certaines formules déjà établies.

1.1 Dimension de Hausdorff

La façon de mesurer le contenu d'un objet géométrique s'adapte à la dimension de celui-ci : nous parlons donc de la *longueur* d'une courbe, de l'*aire* d'une surface, du *volume* d'un solide...

FIGURE 1 – Felix Hausdorff (1868–1942)

Le tableau 1 rassemble, pour les quelques premières valeurs de d, les formules familières donnant la mesure $V_d(r)$ du contenu d'une boule de rayon $r > 0$ dans $\mathbf{R}^d$:

$$\mathcal{B}_C(r) := \{\, P \in \mathbf{R}^d \mid \operatorname{dist}(P, C) \leq r \,\}.$$

d	0	1	2	3	4
$\mathcal{B}_C(r)$	point .	segment	disque	boule	hyperboule
$V_d(r)$	cardinal 1	longueur $2r$	aire πr^2	volume $\frac{4\pi r^3}{3}$	hypervolume $\frac{\pi^2 r^4}{2}$

TABLEAU 1 – Mesure des boules de petites dimensions

De façon générale, on établit, par découpage en tranches, la relation de récurrence

$$V_d(r) = 2 \int_0^r V_{d-1}\big(\sqrt{r^2 - t^2}\big)\, \mathrm{d}t,$$

à partir de laquelle on obtient une expression de V_d en termes de la fonction gamma d'Euler :

$$V_d(r) = \frac{\pi^{\frac{d}{2}}}{\Gamma(\frac{d}{2}+1)}\, r^d$$

ayant l'avantage d'avoir du sens pour une valeur réelle positive quelconque de d.

Définition 1 *Étant donné un réel $d \geq 0$ et une partie bornée $\mathcal{A}$ d'un espace métrique (X, ρ), la* mesure de Hausdorff *d*-dimensionnelle *de $\mathcal{A}$ est définie par*

$$\mathfrak{m}_d(\mathcal{A}) := \sup_{\delta>0} \left(\inf \sum_i V_d(r_i) \right),$$

où l'infimum est pris sur tous les recouvrements de $\mathcal{A}$ par des boules $\mathcal{B}_{C_i}(r_i)$ avec $r_i < \delta$.

On montre [15] qu'il existe une unique valeur $\dim(\mathcal{A}) \in [0, +\infty]$ pour laquelle $\mathfrak{m}_d(\mathcal{A}) = 0$ si $d > \dim(\mathcal{A})$ et $\mathfrak{m}_d(\mathcal{A}) = +\infty$ si $d < \dim(\mathcal{A})$; c'est ce nombre, non nécessairement entier, que l'on appelle *dimension de Hausdorff* de $\mathcal{A}$. Ainsi, la seule notion *in fine* pertinente de mesure est celle correspondant à la dimension de l'ensemble considéré :

$$\mathfrak{m}(\mathcal{A}) := \mathfrak{m}_{\dim(\mathcal{A})}(\mathcal{A}),$$

et cette mesure est d'office homogène de degré $\dim(\mathcal{A})$ par rapport aux dilatations :

$$\mathfrak{m}(\alpha\mathcal{A}) = \alpha^{\dim(\mathcal{A})}\, \mathfrak{m}(\mathcal{A}) \qquad \text{pour tout } \alpha > 0.$$

De façon pragmatique, cela mène à penser à la dimension d'un ensemble $\mathcal{A}$ de mesure non nulle comme étant le degré de la formule donnant sa mesure par rapport au facteur d'échelle :

$$\dim(\mathcal{A}) = \log_\alpha \left(\frac{\mathfrak{m}(\alpha\mathcal{A})}{\mathfrak{m}(\mathcal{A})} \right), \qquad \alpha > 0.$$

Dans les cas simples, la dimension de Hausdorff est équivalente à celle de Minkowski qui s'exprime en terme de la fonction $N_\mathcal{A}(r)$ comptant le nombre de boules de rayon r nécessaires pour recouvrir $\mathcal{A}$: on a alors (figure 2)

$$\dim(\mathcal{A}) = -\lim_{r\to 0} \frac{\log N_\mathcal{A}(r)}{\log r};$$

on peut penser à $N_\mathcal{A}(r)$ comme étant (asymptotiquement) inversement proportionnel à $r^{\dim(\mathcal{A})}$.

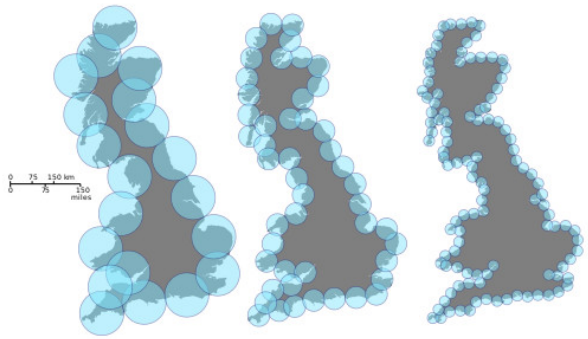

FIGURE 2 – Un exemple classique : mesurer la côte de la Grande-Bretagne

1.2 Dimension algorithmique

Nous pouvons donner un analogue algorithmique de la formule précédente en considérant, pour un algorithme $\mathcal{A}$, le nombre maximal $N_{\mathcal{A}}(\ell)$ d'étapes (dans un modèle calculatoire donné) de calculs effectuées par $\mathcal{A}$ sur une entrée de taille ℓ, et posant

$$\dim(\mathcal{A}) := \lim_{\ell \to \infty} \frac{\log N_{\mathcal{A}}(\ell)}{\log \ell} \qquad \text{(si cette limite existe).}$$

Ainsi, un algorithme pour lequel la fonction $N_{\mathcal{A}}$ est polynomiale de degré d sera-t-il de dimension d ; mais cette notion a l'intérêt d'ignorer les facteurs logarithmiques ayant tendance à pulluer dans les questions de complexité algorithmique. Par exemple, un algorithme avec

$$N_{\mathcal{A}}(\ell) \sim C\, \ell^d\, \log \ell\, (\log \log \ell)^4$$

est également de dimension d ; nous pouvons y penser comme étant *essentiellement polynomial*.

Donnons quelques exemples pour observer cette notion en action.

1.2.1 Énumération

Un algorithme qui énumère tous les points à coordonnées entières dans un ensemble $\mathcal{A} \subseteq \mathbf{R}^n$ est typiquement de dimension $\dim(\mathcal{A})$ par rapport à la taille ℓ de $\mathcal{A}$. Par exemple, énumérer toutes les chaînes de caractères de longueur d formées avec un alphabet à ℓ symboles (ce qui revient à énumérer les points à coordonnées entières dans l'hypercube $\mathcal{H} = [1, \ell]^d$) est de dimension d : il y en a ℓ^d. Par abus conceptuel, on aura envie de considérer une telle énumération comme étant de dimension d même si ℓ est fixé à une petite valeur (par exemple le cas binaire $\ell = 2$).

1.2.2 Primalité

Le test naïf qui consiste à énumérer tous les entiers entre 1 et n pour vérifier s'ils divisent n est de dimension 1 (cas particulier d'un segment dans l'exemple précédent). En exploitant la remarque selon laquelle si n est composé, alors n possède forcément un diviseur inférieur à $\sqrt{n}$, on peut restreindre l'intervalle de recherche et obtenir ainsi un algorithme *fractal*, de dimension $\frac{1}{2}$.

Remarquons, dans l'exemple précédent, que le paramètre utilisé pour décrire la taille de l'entier n était n lui-même, ce qui correspond au nombre de symboles dans sa représentation *unaire*. Dans les applications arithmétiques, il est plutôt d'usage de mesurer la taille de n par le nombre

$$\ell_b(n) := \lfloor \log_b n \rfloor + 1$$

de chiffres nécessaires pour le représenter en base $b \geq 2$.

Notons qu'il existe des tests de primalité polynomiaux par rapport à la taille de n : ceux-ci sont de dimension 0 par rapport à n, mais peuvent être considérés de *dimension logarithmique* finie (par exemple l'algorithme [1] qui est de dimension logarithmique 7,5).

1.2.3 Multiplication matricielle

Si on implémente directement la définition de la multiplication de deux matrices $\ell \times \ell$:

$$c_{ij} = \sum_{k=1}^{\ell} a_{ik} b_{kj},$$

on obtient un algorithme de dimension 3 pour le calcul de $C = A \cdot B$. Notons que des algorithmes de plus faible dimension existent : mentionnons l'algorithme de Strassen (dimension approximative 2,807) et celui de Coppersmith-Winograd (2,376).

1.2.4 Multiplication d'entiers de ℓ chiffres

Exemple semblable au précédent : la méthode naïve apprise à la petite école est un algorithme de dimension 2, mais il existe des algorithmes de plus faible dimension (Karatsuba, dimension 1,585 ; Schönage-Strassen, basé sur la transformée de Fourier rapide, dimension 1).

2 Cryptographie

La cryptographie moderne peut être considérée comme de la théorie de la complexité appliquée : les utilisateurs légitimes (conventionnellement nommés Alice et Bob) cherchent à appliquer des opérations à leurs données pour leur conférer différentes propriétés (confidentialité, authentication, non-répudiation, ...) qu'un attaquant aura du mal à contourner. Ici, l'expression *avoir du mal* est à interpréter de façon quantitative : en théorie, l'attaquant pourrait toujours arriver à ses fins s'ils disposait de suffisamment de ressources (temps, mémoire, capacité de calcul) ; mais l'idée est de rendre les ressources nécessaires disproportionnées par rapport à l'avantage obtenu en cas d'attaque réussie.

2.1 Chiffrement symétrique

En cryptographie symétrique (figure 3), Alice et Bob disposent tous deux d'une clé secrète qu'ils sont les seuls à connaître et appliquent un algorithme de chiffrement à leurs données pour les rendre incompréhensible à l'attaquant qui ne dispose pas de la clé.

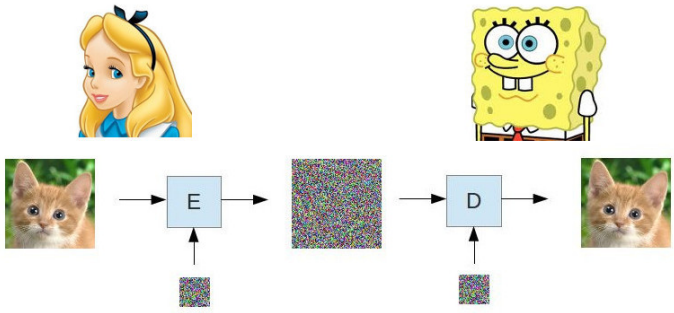

FIGURE 3 – Un cryptosystème symétrique

Formellement, un *cryptosystème symétrique* est la donnée d'un ensemble de messages $\mathcal{M}$, de messages chiffrés $\mathcal{C}$ et de clés $\mathcal{K}$ ainsi que de deux algorithmes

$$E : \mathcal{K} \times \mathcal{M} \to \mathcal{C} \qquad \text{(chiffrement)} \quad \text{et}$$

$$D : \mathcal{K} \times \mathcal{C} \to \mathcal{M} \qquad \text{(déchiffrement)}$$

avec la propriété de *déchiffrement correct* :

$$D(k, E(k, m)) = m \qquad \text{pour toute clé } k \in \mathcal{K} \text{ et message } m \in \mathcal{M}.$$

Intuitivement, on voudrait que E et D soient efficaces du point de vue d'Alice et Bob (disons : de dimension finie par rapport à la *taille* ℓ des clés, habituellement mesurée en bits) mais que l'attaquant ne dispose pas d'une attaque efficace. Sachant que l'attaquant a toujours en principe la possibilité d'essayer toutes les clés puis de déchiffrer (attaque par force brute), il existe forcément toujours des attaques de dimension finie par rapport au *nombre* 2^ℓ de clés. On définit d'ailleurs le *niveau de sécurité* d'un algorithme de chiffrement comme étant la dimension équivalente d'une attaque par force brute permettant de déchiffrer sans connaître la clé.

Par exemple : l'algorithme Rijndael, normalisé aux États-Unis par le NIST sous l'appellation *Advanced Encryption Algorithm* [10], fournit aujourd'hui pour une clé de 128 bits un niveau de sécurité d'environ 126 bits [3].

2.2 Chiffrement asymétrique

En cryptographie asymétrique (figure 4), on dispose cette fois d'une paire de clés associées : une clé k_e pour le chiffrement et une clé k_d pour le déchiffrement, avec la propriété que si (k_e, k_d) est une paire de clés associées, on a

$$D(k_d, E(k_e, m)) = m \qquad \text{pour tout } m \in \mathcal{M}.$$

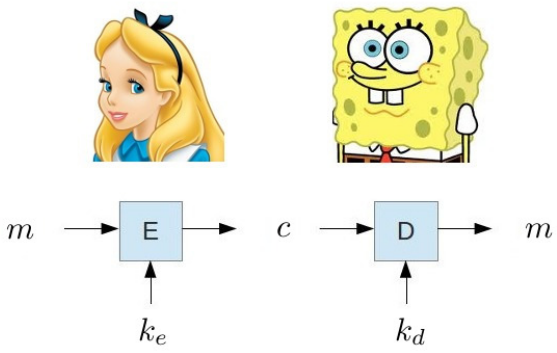

FIGURE 4 – Chiffrement asymétrique

On parle de *chiffrement à clé publique* lorsque k_e est publique et k_d gardée secrète (n'importe quelle Alice peut alors écrire à Bob et lui seul saura déchiffrer le message) ; dualement, le cas où k_e est secrète et k_d publique donne lieu à la *signature électronique* (Alice émet un message qu'elle seule sait produire, et n'importe quel Bob peut vérifier que c'est bien le cas). Si on

aime souvent présenter le premier cas comme application motivante, il est important de noter que d'un point de vue technologique, celle-ci est toujours restée plutôt marginale, alors que la signature électronique est au cœur de plusieurs protocoles sécurisés incontournables aujourd'hui (TLS/SSL, chaîne de blocs, ...).

2.3 RSA

L'algorithme asymétrique le plus connu est certainement celui proposé par Rivest, Shamir et Adleman en 1977 [14] et basé sur l'arithmétique modulaire. Rappelons que celui-ci consiste à se donner un (grand) entier n et à utiliser comme fonctions de chiffrement et de déchiffrement de simples exponentiations modulaires :

$$\begin{cases} E(e,m) \underset{n}{\equiv} m^e, \\ D(d,c) \underset{n}{\equiv} c^d \end{cases}$$

où l'on convient d'écrire $a \underset{n}{\equiv} b$ lorsque n divise sans reste $b-a$.

On vérifie alors aisément à l'aide du théorème des restes chinois et du petit théorème de Fermat que pour obtenir le déchiffrement correct de tous les messages (du moins lorsque n est libre de carrés) il suffit de demander que

$$de \underset{\phi(n)}{\equiv} 1,$$

où ϕ est la fonction indicatrice d'Euler.

Il est important de noter ici qu'on peut calculer efficacement les exponentiations modulaires requises. Par exemple, supposons que l'on cherche à calculer m^e modulo n pour les entiers de taille modérée $m = 492$, $e = 40$, $n = 1679$. On pourrait bien sûr tout d'abord évaluer

$$\begin{aligned} m^{40} = {} & 47709417120732287683866762031447600312014266072755944 7 \\ & 05769087731091689946418876534537357522377875429890457 6 \end{aligned}$$

puis trouver par division euclidienne le reste de la division par 1679 pour trouver 1476. Mais on peut être beaucoup plus efficace en effectuant des mises au carré répétées : on calculera tout d'abord les carrés successifs

$$m = 492,\ m^2 \underset{n}{\equiv} 288,\ m^4 \underset{n}{\equiv} 673,\ m^8 \underset{n}{\equiv} 1278,\ m^{16} \underset{n}{\equiv} 1296,\ m^{32} \underset{n}{\equiv} 616$$

avant d'en déduire

$$m^{40} \underset{n}{\equiv} m^{32} \cdot m^8 \underset{n}{\equiv} 616 \cdot 1278 \underset{n}{\equiv} 1476.$$

De façon générale, on peut réaliser l'exponentiation modulaire à l'aide de $\ell_2(e)$ mises au carré et multiplications modulaires d'entiers à $\ell_2(n)$ chiffres (voir 1.2), ce qui donne pour les algorithmes de chiffrement et de déchiffrement RSA une dimension logarithmique comprise entre 2 et 3.

Du point de vue de l'attaquant : connaissant d ou e, on obtient l'autre aisément comme coefficient de Bézout à l'aide de l'algorithme d'Euclide étendu qui permet de résoudre l'équation diophantine

$$de + k\phi(n) = 1.$$

On a donc intérêt à ce que le calcul de $\phi(n)$ soit **long**, ce qui explique le choix habituel de prendre $n = pq$ avec p et q deux grands nombres premiers afin de mettre l'attaquant dans la situation la plus défavorable ; la meilleure attaque connue à ce jour contre RSA dans le cas général étant de factoriser n pour en déduire

$$\phi(n) = (p-1)(q-1).$$

Pour discuter de la complexité de cette attaque, introduisons la notation L de Pomerance [12] en posant pour $d \in \mathbf{R}$ et $\alpha \in [0,1]$,

$$L_\alpha(n) := \exp\left(d\,(\log n)^\alpha\,(\log\log n)^{1-\alpha}\right).$$

Cette notation permet d'interpoler en quelque sorte entre la dimension algorithmique d ($\alpha = 1$) et la dimension logarithmique d ($\alpha = 0$) [1]. En utilisant celle-ci, on peut décrire l'évolution de la complexité des meilleurs algorithmes de factorisation connus :

— Recherche exhaustive de facteur [4] : L_1 avec $d = \frac{1}{2}$;
— Crible quadratique [5] : $L_{\frac{1}{2}}$ avec $d = 1$;
— Crible du corps de nombres généralisé [9] : $L_{\frac{1}{3}}$ avec $d \approx 1{,}923$.

En observant la courbe obtenue (figure 5), on comprend pourquoi l'utilisation de RSA avec un niveau de sécurité de 128 bits nécessite aujourd'hui l'utilisation d'entiers d'au moins 3000 bits.

2.4 Logarithme discret

Une autre façon d'utiliser l'exponentiation modulaire en cryptographie est de remarquer que, si le calcul de $x \equiv g^y \pmod n$ connaissant g et y est rapide, le problème dit du *logarithme discret* consistant à retrouver y à partir de x (et g) est en général plus ardu. Plusieurs systèmes

1. Notons que la nomenclature utilisée ici est relative à l'entier n lui-même. En terme de sa taille $\ell_2(n)$, on parlera habituellement plutôt de complexité exponentielle lorsque $\alpha = 1$ et polynomiale lorsque $\alpha = 0$.

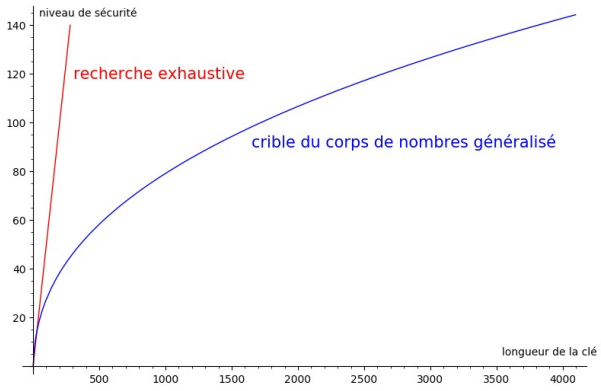

FIGURE 5 – Niveau de sécurité de RSA en fonction de la taille de la clé

asymétriques (protocole d'échange de clés de Diffie-Hellman, chiffrement de ElGamal, algorithme de signature DSA) sont basés sur cette idée.

Ces algorithmes présentent des niveaux de sécurités comparables à celui de RSA pour une taille de n donnée ; mais ce qui est intéressant c'est que dans un groupe abélien quelconque, le meilleur algorithme connu de calcul de logarithme discret est de dimension $\frac{1}{2}$. C'est ce qui explique l'adoption, depuis leur introduction dans les années 80 [8], d'algorithmes basés sur les courbes elliptiques (ECDSA, ECDH) pour lesquelles aucun algorithme de logarithme discret spécifique n'existe : on obtient donc un niveau de sécurité de 128 bits à l'aide de paramètres ayant seulement 256 bits.

3 La menace quantique

Considérée comme pure science-fiction il y a encore quelques années, plusieurs spécialistes estiment aujourd'hui qu'il est *concevable* que nous voyions un jour la construction d'ordinateurs quantiques manipulant l'information de façon radicalement différente des ordinateurs électroniques classiques. Nous n'évoquerons pas ici les concepts physiques sous-jacents, mais nous contenterons de décrire à haut niveau les deux principaux algorithmes quantiques ayant une incidence sur les niveaux de sécurité des cryptosystèmes actuels.

3.1 Algorithme de Grover

Étant donné une fonction quelconque $f : A \to B$ avec $|A| = n$ et $b \in f(A)$, la recherche d'une préimage $a \in A$ telle que $f(a) = b$ prend classiquement n évaluations pour réaliser une recherche séquentielle. Un ordinateur quantique pourrait par contre en trouver une (avec grande probabilité) en seulement $\sqrt{n}$ évaluations de f [6], réduisant ainsi la dimension logarithmique d'une attaque par force brute d'un facteur 2.

La conséquence pour le chiffrement symétrique est simple : pour maintenir un niveau de sécurité constant face à un attaquant disposant d'un ordinateur quantique, il serait nécessaire de doubler la taille des clés (par exemple : passer à AES-256 pour maintenir un niveau de sécurité d'environ 128 bits).

3.2 Algorithme de Shor

Cet algorithme, décrit pour la première fois en 1994 [13], permet de factoriser sur un ordinateur quantique un entier n en

$$O\big((\log n)^2(\log\log n)(\log\log\log n)\big)$$

opérations. Il s'agit donc d'un algorithme de dimension logarithmique (au plus) 2 : avec un ordinateur quantique, déchiffrer RSA *sans* la clé n'est pas vraiment plus long que pour l'utilisateur légitime qui la connaît !

Ce qui rend cette prouesse possible est la facilité qu'aurait un ordinateur quantique à découvrir la période d'une fonction en utilisant la transformée de Fourier quantique. Pour factoriser $n = pq$, on détermine la période de fonctions de la forme

$$f_a(x) \underset{n}{\equiv} a^x$$

jusqu'à ce qu'on en trouve une avec une période paire $2k$. On a donc alors

$$a^{2k} - 1 = (a^k - 1)(a^k + 1) \underset{n}{\equiv} 0$$

et on trouvera alors (avec probabilité $\frac{1}{2}$) les facteurs de n avec

$$\mathrm{PGCD}(a^k - 1, n) \qquad \text{et} \qquad \mathrm{PGCD}(a^k + 1, n).$$

3.3 Ouverture : chiffrement post-quantique

Une variante de l'algorithme de Shor permettant aussi de résoudre rapidement le problème du logarithme discret dans un groupe quelconque, l'arrivée de l'ordinateur quantique aurait

un effet catastrophique sur la cryptographie asymétrique, rendant la plupart des algorithmes en usage actuellement caducs. La communauté cryptographique cherche donc dès maintenant à améliorer d'autres algorithmes, dits *post-quantiques*, pour lesquels aucune attaque quantique sérieuse n'est connue ; le NIST américain prend cette démarche très au sérieux, étant présentement engagé dans un processus de normalisation [11].

Il y a en ce moment quatre grandes pistes de familles d'algorithmes résistants aux attaques quantiques :

— à base de fonctions de hachage ;
— à base de codes correcteurs ;
— à base de réseaux ;
— équations quadratiques à plusieurs variables,

dont plusieurs chercheurs s'affairent à améliorer l'efficacité du point de vue d'Alice et Bob. Une discussion des différentes pistes explorées par la communauté cryptographique ainsi que des problèmes rencontrés serait trop longue pour pour être tentée ici (la lectrice intéressée pourra consulter, par exemple, [2]) ; contentons-nous ici de décrire grossièrement le mécanisme de fonctionnement d'un chiffrement basé sur des réseaux, comme l'algorithme NTRU [7].

On se donne comme paramètre public un réseau

$$\Lambda = \left\{ \sum_i a_i \mathbf{v}_i \,\middle|\, a_i \in \mathbf{Z} \right\},$$

où $\mathbf{v}_1, \ldots, \mathbf{v}_n$ sont des vecteurs linéairement indépendants dans $\mathbf{R}^n$. Pour chiffrer un message $\mathbf{v} \in \Lambda$, on y ajoute un petit bruit $\mathbf{e} \in \mathbf{R}^n$ (figure 6) :

$$\mathbf{c} = \mathbf{v} + \mathbf{e}.$$

À l'inverse, pour déchiffrer : étant donné $\mathbf{c}$, on cherche le vecteur $\mathbf{v}$ du réseau le plus proche, problème algorithmiquement difficile (même pour un ordinateur quantique) en l'absence d'une base adaptée.

Remerciements

L'auteur aimerait en profiter pour remercier et féliciter de nouveau tous ceux ayant, de près ou de loin, contribué à faire du congrès de l'AMQ 2018 une réussite ; ainsi qu'au Bulletin qui lui donne aujourd'hui l'opportunité de développer et étayer cette intervention dont la qualité a pu bénéficier des commentaires éclairés des relecteurs.

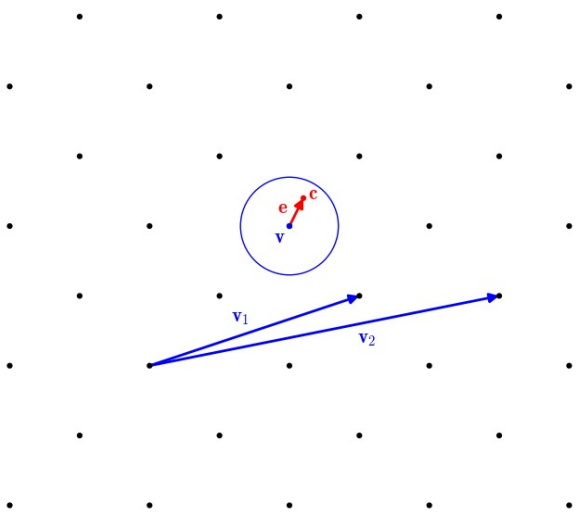

FIGURE 6 – Chiffrement à base de réseau

Références

[1] Agrawal, M., Kayal, N. et Saxena, N. (2004). *PRIMES is in P*. Annals of Mathematics, vol. 160, no. 2, pp. 781–793.

[2] Bernstein, D.J. et Lange, T. (2017). *Post-quantum cryptography – dealing with the fallout of physics success*. Nature 549, pp. 188–194.

[3] Bogdanov, A., Khovratovich, D. et Rechberger, C. (2011). *Biclique cryptanalysis of the full AES*. Advances in Cryptology – ASIACRYPT 2011, Lecture Notes in Computer Science, vol. 7073.

[4] Fibonacci, L. (1202). *Liber abaci*.

[5] Gerver, J. (1983). *Factoring large numbers with a quadratic sieve*. Mathematics of Computation, vol. 41, pp. 287–294.

[6] Grover L.K. (1996). *A fast quantum mechanical algorithm for database search*, Proceedings, 28th Annual ACM Symposium on the Theory of Computing, p. 212.

[7] Hoffstein, J., Pipher, J. et Silverman J.H. (1998). *NTRU : A ring-based public cryptosystem*. International Algorithmic Number Theory Symposium, pp. 267–288.

[8] Koblitz, N. (1987). *Elliptic curve cryptosystems*. Mathematics of Computation, vol. 48, no. 177, pp. 203–209.

[9] Lenstra, A.K. et Lenstra, H.W. Jr. (1993). *The development of the number field sieve*. Lecture Notes in Mathematics 1554, Springer-Verlag.

[10] National Institute of Standards and Technology (nov. 2001). *Specification for the Advanced Encryption Standard*, FIPS 197, récupéré le 15 décembre 2018 au https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf.

[11] National Institute of Standards and Technology (jan. 2017). *Post-quantum cryptography standardization*, récupéré le 15 décembre 2018 au https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Post-Quantum-Cryptography-Standardization.

[12] Pomerance, C. (1982). *Analysis and comparison of some integer factoring algorithms*. Computational Methods in Number Theory, pp. 89–139.

[13] Shor, P. (1997). *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*. SIAM Journal of Computing, vol. 26, pp. 1484–1509.

[14] Rivest R.L., Shamir A. et Adleman L. (1978). *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM, vol. 21, no. 2, pp. 120–126.

[15] Schleicher, D. (2007). *Hausdorff dimension, its properties, and its surprises*. The American Mathematical Monthly, vol. 114, no. 6, pp. 509–528.