Séminaire Bourbaki Mars 2003
55ème année, 2002-2003, n° 917

LA PRIMALITÉ EN TEMPS POLYNOMIAL
[d'après Adleman, Huang ; Agrawal, Kayal, Saxena]

par **F. Morain**

1. INTRODUCTION

Le théorème fondamental de l'arithmétique affirme que tout entier positif s'écrit comme un produit de puissances de nombres premiers distincts de manière unique, à l'ordre près des facteurs. On sait depuis Euclide qu'il existe une infinité de nombres premiers, même si la démonstration de ce résultat ne fournit pas de grands nombres premiers de manière constructive. Pendant des millénaires, trouver les facteurs premiers d'un entier esthétique a été une motivation pour le développement de méthodes de décomposition en facteurs premiers. Les nombres $F_n = 2^{2^n} + 1$ (nombres de Fermat), ou $M_m = 2^m - 1$ (nombres de Mersenne) ont ainsi servi de pierres de touche. Nul doute que la conjecture de Fermat sur la primalité de tous les F_n a stimulé bien des recherches, à l'époque d'Euler ou bien encore aujourd'hui. On renvoie le lecteur intéressé à [74] pour un aperçu de la période précédant l'arrivée des ordinateurs. Les calculs étant faits à la main (à l'exception de plusieurs tentatives de mécanisation), les spécialistes de l'époque savaient bien ce qui était *faisable* ou *infaisable*. L'exploit réalisé par Lucas en prouvant la primalité de M_{127} (39 chiffres décimaux) est resté inégalé jusqu'à l'arrivée des ordinateurs.

Ceux-ci ont apporté avec eux une puissance de calcul prodigieuse, et on peut imaginer sans trop de difficulté la joie des pionniers de la théorie algorithmique des nombres qui ont trouvé les premiers nombres premiers de Mersenne depuis Lucas ! Au fil des ans, la théorie de la complexité est née et a grandi, son but étant de classifier les problèmes faciles et les problèmes difficiles. Pour simplifier, un problème est facile quand on lui connaît un algorithme de résolution dont le temps de calcul est polynomial en la taille de l'entrée. Une différentiation importante s'est faite entre algorithmes déterministes et algorithmes randomisés, ces derniers pouvant tirer à pile ou face pour prendre des décisions. La théorie de la complexité s'est ramifiée au cours des ans, et il serait présomptueux de la résumer en quelques lignes. Nous avons toutefois rassemblé dans la section 2 de cet article les

notions essentielles pour notre propos, ce qui nous permettra de signaler au fil de l'article les interprétations qu'on peut donner des théorèmes dans le langage de la complexité.

S'il fallait encore une motivation pour étudier les algorithmes de primalité, nul doute que nous ferions appel aux plus gros consommateurs de nombres premiers, à savoir les cryptographes, qui utilisent désormais quotidiennement des nombres premiers dans leurs cryptosystèmes [53], à commencer par le fameux RSA [4].

L'un des buts de cet article est de présenter l'algorithme récent d'Agrawal, Kayal et Saxena, qui montre que la primalité (ou plutôt le problème de décision estPremier?) est décidable en temps polynomial déterministe. Avant d'arriver à cela, il ne nous a pas paru inutile de revenir sur les vingt dernières années (grosso modo depuis l'article fondateur de la primalité moderne [47]), marquées entre autres par l'utilisation des courbes algébriques en primalité, fournissant le premier algorithme randomisé de primalité fonctionnant en temps polynomial. Ces considérations expliquent le titre choisi pour l'article.

Nous n'oublierons pas non plus les aspects pratiques des tests de primalité. Les progrès là encore ont été foudroyants dans les vingt dernières années, en liaison avec les avancées théoriques.

Les ouvrages traitant de nombres premiers et de factorisation sont nombreux. Outre l'incontournable [65], nous ferons référence principalement à [24] et [29].

Notations : dans ce qui suit, p et q dénoteront des nombres premiers, et N l'entier dont la primalité doit être étudiée.

2. BRÈVE INTRODUCTION À LA THÉORIE DE LA COMPLEXITÉ

Même si nous avons délibérement choisi de ne pas regarder la primalité à travers le prisme de la théorie de la complexité, il convient de donner quelques pistes pour comprendre les différentes règles du jeu (d'informatique théorique) qui interviennent. Le livre [58] est une bonne référence pour ce qui suit. On trouvera dans [32] une section sur la primalité et la complexité.

La façon la plus simple de mesurer la complexité d'un algorithme est par son temps de calcul. Le paramètre d'entrée est généralement la taille des données fournies à l'algorithme. On cherche alors une fonction de cette taille, notée n, qui donne le résultat. À titre d'exemple, l'addition de deux polynômes de degré $n-1$ à coefficients dans $K = \mathbb{F}_2$ est n additions dans K, soit $f_K(n) = n$.

Deux algorithmes différents réalisant la même opération seront comparés à l'aide de leur fonction de coût respective. Une fois cet ordre de grandeur des calculs établis, la détermination des constantes sera primordiale dans la mesure de l'efficacité pratique des algorithmes.

Par exemple, l'algorithme classique de multiplication de deux polynômes de degré n prendra $O(n^2)$ multiplications d'éléments de K. Si le corps K le permet, on peut utiliser

la transformée de Fourier rapide (FFT), et on obtient alors un temps de calcul $O(n \log n)$ (voir [41, 32, 29]).

Le but est maintenant, pour chaque problème de calcul donné, de déterminer quelle est la meilleure fonction de coût possible pour la résolution du problème. Cela permet de répartir les problèmes dans différentes *classes de complexité*. Par exemple, la classe $\mathcal{T}(n^2)$ désigne la classe des problèmes pour lesquels il existe un algorithme de résolution en temps au plus $O(n^2)$ pour une entrée de taille n. Les problèmes de la classe $\mathsf{P} = \cup_{k \geqslant 1} \mathcal{T}(n^k)$ sont ceux dont le temps de calcul est au plus polynomial en n : c'est la classe des problèmes faciles par excellence. Ceux de $\mathsf{EXP} = \cup_{k \geqslant 1} \mathcal{T}(2^{n^k})$ nécessitent un temps de calcul exponentiel en n. Une autre classe permet de simplifier certains énoncés. Si $f(n) \to \infty$, on note $\tilde{O}(f(n)) = \cup_{k \geqslant 1} O(f(n) \log(f(n))^k)$. Par exemple, les algorithmes de multiplication à base de FFT décrits ci-dessus montrent que la multiplication est dans $\tilde{O}(n)$.

Pour le moment, nous avons sous-entendu que les algorithmes que nous utilisions étaient *déterministes*, comme dans le cas de l'addition ou de multiplication de polynômes. Cette contrainte est très forte. Considérons un problème de base de la théorie des nombres, qui est celui de la recherche d'un non résidu quadratique dans $(\mathbb{Z}/p\mathbb{Z})^*$ pour p premier. Sans l'hypothèse de Riemann, on ne sait pas trouver un tel résidu en temps polynomial déterministe, alors qu'on sait qu'un élément sur 2 n'est pas un carré et qu'il suffit donc de tirer au sort quelques valeurs pour en trouver une bonne. Un algorithme tirant des nombres au hasard a ainsi une probabilité de succès plus grande que 1/2.

Les problèmes considérés ci-dessus sont des problèmes de *calcul*. Un autre type de problème très important en complexité est celui des problèmes de *décision*. Celui qui nous intéressera dans la suite est bien sûr estPremier?. Quand on ne dispose pas d'algorithmes déterministes pour répondre à la question, on peut chercher des algorithmes randomisés, qui peuvent s'aider de tirages à pile ou face. La classe des algorithmes randomisés qui nous intéresse au premier chef est celle dite de Monte Carlo polynomiale, notée RP(pour *random polynomial*). Un tel algorithme permet de répondre en temps polynomial à la question

"x appartient-il[1] à l'ensemble E ?"

Si $x \notin E$, alors l'algorithme répond toujours que x n'est pas dans E. Si $x \in E$, alors l'algorithme le reconnaît avec probabilité supérieure à 1/2. Itérant alors l'algorithme en faisant k choix indépendants aboutit à rendre la probabilité de succès de l'algorithme itéré aussi proche qu'on veut de 1.

[1] On parle d'appartenance à un *langage* en théorie de la complexité.

Les algorithmes de primalité sont répartis en deux catégories : les *tests de composition* répondent à la question estComposé ? (on en verra un exemple avec le test de Solovay-Strassen). Les *tests de primalité* répondent quant à eux à la question estPremier ? (c'est le cas de l'algorithme AKS).

La classe RP fournit souvent des algorithmes raisonnables pour résoudre un problème donné. En primalité, il se trouve que les tests de composition sont généralement très rapides en pratique, contrairement aux tests de primalités qui sont souvent plus lourds à mettre en œuvre. Ces tests peuvent souvent fournir un *certificat*, c'est-à-dire des éléments qui permettent de convaincre un observateur extérieur de la justesse du calcul. Nous verrons à la section qui suit des exemples de ces concepts.

Il existe d'autres classes de complexité, la plus intéressante étant celle appelée ZPP(pour *zero-probability polynomial*). Elle contient les problèmes de décision pour lesquels existe un algorithme de type *Las Vegas*. Un tel algorithme répond "oui", "non", ou "je ne sais pas". La probabilité qu'il réponde "je ne sais pas" peut être rendue aussi petite que possible. Cette classe est l'intersection de RP et de co-RP(co-RPest la classe des problèmes pour laquelle on peut décider de la non appartenance en temps polynomial randomisé). La classe ZPP contient les problèmes qui sont "moralement résolus" par des algorithmes randomisés.

3. FERMAT, LUCAS, LEHMER

3.1. Tests de composition

Le plus simple de ces tests est fondé sur le petit théorème de Fermat : si a est premier avec N et $a^{N-1} \not\equiv 1 \bmod N$, alors N n'est pas premier. Cela nous fournit une preuve de composition, qui ne consiste pas en un facteur de N. On peut fabriquer un algorithme de composition de la façon suivante :

fonction estComposéAvecFermat(N)

1. Choisir $a \neq 0$ au hasard dans $\mathbb{Z}/N\mathbb{Z}$.
2. Calculer $g = \mathrm{pgcd}(a, N)$; **si** $g > 1$, **alors** retourner (`oui`, g est un facteur de N).
3. **si** $a^{N-1} \not\equiv 1 \bmod N$, **alors** retourner (`oui`, a) **sinon** retourner `je ne sais pas`.

Proposition 3.1. — *La probabilité d'échec est* $P(N)/(N-1)$ *où* $P(N) = \prod_i \mathrm{pgcd}(p_i - 1, N-1)$ *si* $\prod_i p_i^{\alpha_i}$ *est la décomposition de* N *en facteurs premiers avec* $p_i \neq p_j$.

Démonstration. — Il est plus facile d'estimer la probabilité de succès de l'algorithme. La probabilité de trouver un diviseur de N à l'étape 2 est donnée par :

$$\left(1 - \frac{\varphi(N)}{N-1}\right)$$

où $\varphi(N) = \mathrm{Card}((\mathbb{Z}/N\mathbb{Z})^*)$ est la fonction indicatrice d'Euler.

On laisse au lecteur le soin de démontrer que le nombre de $a \in (\mathbb{Z}/N\mathbb{Z})^*$ satisfaisant

$$a^{N-1} \equiv 1 \bmod N$$

est $P(N)$. La probabilité de trouver un a prouvant la composition de N à l'étape 3 est donc :

$$\frac{\varphi(N)}{N-1}\left(1 - \frac{P(N)}{\varphi(N)}\right).$$

En additionnant les deux termes, on trouve le résultat. □

Que conclure de cette proposition ? Tout d'abord, si N est premier, alors l'algorithme échoue toujours à prouver la composition de N, ce qui est rassurant (en effet $P(N) = N - 1$). Ensuite, la fonction $P(N)$ gouverne le succès de l'algorithme. Cette quantité est non triviale, quand N est *nombre pseudopremier en base* a, c'est-à-dire qu'il satisfait $a^{N-1} \equiv 1 \bmod N$ pour a fixé. Il peut même arriver que $P(N) = \varphi(N)$, c'est le cas quand N est un *nombre de Carmichael*, dont on sait qu'il existe une infinité [7]. La probabilité d'échec de l'algorithme est donc proche de 1 pour ces nombres, ce qui fait que cet algorithme ne suffit pas à prouver que estPremier? est dans RP.

Pour faire mieux, on fait appel au théorème d'Euler, qui nous dit que si N est un nombre premier et a un entier premier avec N, alors

$$a^{(N-1)/2} \equiv \left(\frac{a}{N}\right) \bmod N \tag{1}$$

où $\left(\frac{a}{N}\right)$ désigne le symbole de Legendre. Le nombre a étant fixé, un nombre composé N vérifiant (1) (où $\left(\frac{a}{N}\right)$ désigne cette fois le symbole de Jacobi) est appelé *nombre pseudo-premier d'Euler en base* a (noté ppE-a). On pose

$$A_N = \left\{a \in (\mathbb{Z}/N\mathbb{Z})^*, a^{(N-1)/2} \equiv \left(\frac{a}{N}\right) \bmod N\right\}.$$

Si N est premier, alors $A_N = (\mathbb{Z}/N\mathbb{Z})^*$. Par contre, Lehmer [43] a montré, que si N est composé, A_N est un sous-groupe strict de $(\mathbb{Z}/N\mathbb{Z})^*$.

Le test de composition de Solovay et Strassen [70] est le suivant :

fonction estComposé(N)

1. Choisir a au hasard dans $\mathbb{Z}/N\mathbb{Z}\backslash\{0\}$.
2. Calculer $g = \text{pgcd}(a, N)$; **si** $g > 1$, **alors** retourner (`oui`, g est un facteur de N).
3. **Si** (1) n'est pas satisfaite **alors** retourner (`oui`, Nn'est pas ppE-a) **sinon** retourner `je ne sais pas`.

Cet algorithme est bien dans RP, puisque la probabilité de choisir un bon a est au moins $1/2$ (quand N est composé) par le théorème de Lehmer. Quand il réussit, l'algorithme renvoie un *témoin de composition* a.

Miller est allé plus loin [55] (on consultera avec profit [46, 48]). Si l'hypothèse de Riemann pour les fonctions L de Dirichlet pour les caractères réels est vraie, alors le plus

petit témoin est plus petit que $c(\log N)^2$. Bach [10] a montré que $c = 2$ était suffisant. L'algorithme de primalité correspondant est alors :

fonction estPremierAvecMiller(N)

1. **pour** $a = 2$ **à** $2(\log N)^2$ **faire**
   i) calculer $g = \text{pgcd}(a, N)$; **si** $g > 1$ **alors** retourner (`non`, g est un facteur de N).
   ii) **si** l'équation (1) n'est pas satisfaite **alors** retourner (`non`, Nn'est pas ppE-a) ;
2. retourner `oui`.

Quelle est la complexité de cet algorithme ? Calculer a^e dans un groupe prend $O(\log e)$ opérations de groupe. Multiplier deux entiers de taille $\log N$ prend $O((\log N)^2)$ ou encore $\tilde{O}(\log N)$, la division euclidienne a le même coût. Cela conduit à un temps de calcul $O((\log N)^5)$ ou $\tilde{O}((\log N)^4)$.

Tant que ces hypothèses de Riemann ne sont pas prouvées, cet algorithme ne peut être utilisé dans la pratique et il faut se tourner vers d'autres méthodes.

D'autres tests de composition ont été proposés, comme par exemple l'algorithme préféré des cryptographes, celui d'Artjuhov-Miller-Rabin [8, 55, 63], dont la probabilité de succès est $\geqslant 3/4$. Nous renvoyons à [29] pour une plus longue liste.

3.2. **Preuve de primalité**

Les seules méthodes connues et utilisées pour prouver la primalité des entiers jusqu'à la fin des années 1970 étaient basées sur le théorème de Fermat, et des généralisations obtenues par Lucas et Lehmer. Nous ne nous intéresserons pas ici au côté pratique des algorithmes, ni aux avancées plus récentes, préférant renvoyer à la littérature [19, 20, 29].

Théorème 3.2. — *N est premier si et seulement si $(\mathbb{Z}/N\mathbb{Z})^*$ est cyclique d'ordre $N-1$.*

Proposition 3.3. — *Supposons connus les facteurs premiers p_i de $N-1$. Le nombre $g \in (\mathbb{Z}/N\mathbb{Z})^*$ est d'ordre $N-1$ si et seulement si $g^{N-1} \equiv 1 \bmod N$ et $g^{(N-1)/p_i} \not\equiv 1 \bmod N$ pour tout i.*

On utilise généralement ce théorème de concert avec un résultat de Pocklington

Théorème 3.4. — *Soit s tel que $s \mid N-1$ et a tel que $a^{N-1} \equiv 1 \bmod N$, et pour tout q premier divisant s, $\text{pgcd}(a^{\frac{N-1}{q}} - 1, N) = 1$. Alors tout diviseur premier p de N vérifie $p \equiv 1 \bmod s$.*

Le corollaire est le plus utile :

Corollaire 3.5. — *Avec les hypothèses du théorème précédent, si $s > \sqrt{N}$, alors N est premier.*

L'utilisation pratique de ces résultats nécessite la factorisation de $N-1$ qui est difficile à déterminer, le meilleur algorithme de factorisation connu à ce jour, le crible algébrique [45, 29] ayant une complexité sous-exponentielle. Même si cette factorisation est connue, il reste à trouver g, ce qu'on ne sait pas faire actuellement en temps polynomial déterministe, à moins d'admettre l'hypothèse de Rieman. Pour l'anecdote, l'algorithme randomisé de Shor [69] qui factorise en temps polynomial dans le modèle quantique ne s'attaque qu'au problème de la factorisation [22].

Malgré tout, ces tests permettent de traiter les nombres tels que $N-1$ soit facilement factorisable, c'est-à-dire de la forme $N_1 \prod p_i^{\beta_i}$, ou les p_i sont de petits nombres premiers, et N_1 probablement premier au sens de la partie 2.1. On construit ainsi une suite décroissante d'entiers $(N_i)_{1 \leqslant i \leqslant k}$ avec $N_0 = N$, et N_{i+1} un facteur probablement premier de $N_i - 1$. La primalité de N_k étant prouvée, il ne reste plus qu'à remonter pour prouver de proche en proche celles de tous les N_i, en terminant par N_0. Une telle suite est appelée DOWNRUN par Selfridge.

La proposition 3.3 est utilisée en complexité. En effet, la donnée de g et des facteurs de $N-1$ accompagnés de leurs certificats respectifs permet de montrer que le problème estPremier? appartient à la classe NP (voir [62]).

On peut généraliser cette idée de base au cas où on chercherait à construire un corps fini de degré plus élevé, en utilisant la factorisation de $N+1$, ou celle de $\Phi_k(N)$ pour le k-ième polynôme cyclotomique. Cela donne par exemple l'algorithme de primalité de Lucas-Lehmer qui est très rapide pour les nombres de Mersenne. Ce n'est pas un hasard si ces nombres font régulièrement la une des journaux. Ainsi, le plus grand nombre premier connu, à l'instant où je tape ces lignes[2], est $M_{13466917}$, qui a 4053946 chiffres décimaux[3].

Tous ces tests apparaissent maintenant comme des cas particuliers d'un théorème de Lenstra, énoncé pour la première fois dans [47] et développé dans [50].

THÉORÈME 3.6. — *Soit $s > 0$. Soit $\mathbf{A}$ un anneau contenant $\mathbb{Z}/N\mathbb{Z}$ comme sous-anneau. Supposons qu'il existe $\alpha \in \mathbf{A}$ tel que :*

$$\begin{aligned} \alpha^s &= 1, \\ \alpha^{s/q} - 1 &\in \mathbf{A}^* \text{ pour tout } q \text{ premier } \mid s, \\ \Psi_\alpha(X) = \prod_{i=0}^{t-1} (X - \alpha^{N^i}) &\in \mathbb{Z}/N\mathbb{Z}[X], \text{pour un certain } t > 0. \end{aligned}$$

Alors : $\forall r \mid N, \exists i, r = N^i \bmod s$.

Le cas le plus classique d'application est celui où $\mathbf{A}$ est une extension d'anneau, construite à l'aide d'un polynôme à coefficients dans $\mathbb{Z}/N\mathbb{Z}$. Si N est premier, $\mathbf{A}$ n'est autre que le corps fini $\mathbb{F}_{N^t}$. En particulier, le cas $t = 1$ nous redonne le théorème 3.3.

[2] 24 février 2003

[3] `http ://www.utm.edu/research/primes/largest.html`

4. VINGT ANS DE PRIMALITÉ

Par opposition à la période précédente, les vingt dernières années ont vu l'éclosion et le développement de deux tests de primalité pratiquables pour des nombres entiers *généraux* (par opposition aux nombres pour lesquels $N-1$ ou $N+1$ ont des factorisations faciles). Le premier algorithme historiquement, celui des sommes de Gauss/Jacobi, se décline en deux versions, déterministe ou randomisé, avec un temps de calcul quasi-polynomial, mais ne fournit pas de certificat. La recherche d'algorithmes polynomiaux a été ainsi une motivation pour chercher de nouvelles approches.

Le second algorithme utilise les courbes elliptiques et hyperelliptiques. Il est de type Monte Carlo polynomial. Il en existe également une version pratique utilisant des courbes elliptiques à multiplication complexe, fournissant un certificat, mais dont la complexité polynomiale est heuristique.

4.1. Sommes de Gauss, Jacobi

En 1980, Adleman [1] ébauche un algorithme de primalité utilisant des lois de réciprocité d'ordre supérieur, dans une version randomisée. Rejoint par Pomerance et Rumely, ces travaux conduisent à l'article [3] dans lequel une version déterministe est également présentée. Un temps de calcul $O((\log N)^{c \log\log\log N})$ est prouvé pour les deux versions.

Décrivons brièvement les idées de base. Il n'est pas dans notre propos de donner tous les détails, contenus dans les articles déjà mentionnés ou bien dans [24]. Soient p, q des nombres premiers, $p^k \mid\mid q-1$, $\mathrm{pgcd}(pq, N) = 1$. Soit χ un caractère multiplicatif d'ordre p^k et de conducteur q :

$$\chi \ : \mathbb{F}_q^* \to \mathbb{C}$$

défini par sa valeur en g, un générateur de $\mathbb{F}_q^*$, par $\chi(g) = \zeta_{p^k}$, une racine primitive p^k-ième de l'unité fixée. La somme de Gauss attachée à χ est :

$$\tau(\chi) = \sum_{x=1}^{q-1} \chi(x)\zeta_q^x$$

à valeur dans $R = \mathbb{Z}[\zeta_{p^k}, \zeta_q]$. C'est un objet bien connu en théorie de la cyclotomie (voir par exemple [38]). On construit alors un test de pseudoprimalité en utilisant la proposition élémentaire suivante :

PROPOSITION 4.1. — *Si N est premier,* $\mathrm{pgcd}(N, pq) = 1$, *alors*

$$\frac{\tau(\chi)^N}{\tau(\chi^N)} = \chi(N)^{-N}, \tag{2}$$

la relation s'interprétant dans $\mathbb{Z}/N\mathbb{Z}[\zeta_{p^k}, \zeta_q]$.

L'idée de l'algorithme est alors de tester les identités (2) pour tous les couples (p, q) avec $p^k \mid\mid q-1$ choisis de sorte que $s = \prod_{q \in \mathcal{Q}} > \sqrt{N}$. Si elles sont toutes vérifiées (ainsi

que des conditions techniques négligées ici), alors tout diviseur r de N appartient au groupe multiplicatif $\langle N \bmod s\rangle$, d'ordre $t = \mathrm{ppcm}_{q\in\mathcal{Q}}(q-1)$. Il ne reste plus qu'à vérifier qu'aucun élément de cet ensemble n'est un diviseur premier non trivial de N.

Le coût de l'algorithme est déterminé par cette dernière phase, avec un coût t, qui domine le reste de l'algorithme, constitué de (nombreuses) opérations sur des polynômes de relativement petit degré. Pour diminuer le coût, on commence par chercher t le plus petit possible, avec le plus de diviseurs possibles, et tel que

$$s(t) = \prod_{q-1|t} q$$

soit plus grand que $\sqrt{N}$. Un théorème d'Odlyzko et Pomerance prouve le temps de calcul intrinsèque de l'algorithme.

THÉORÈME 4.2. — *Il existe $c_1, c_2 > 0$ tels qu'on puisse trouver t convenable avec*

$$(\log N)^{c_1 \log\log\log N} \leqslant t \leqslant (\log N)^{c_2 \log\log\log N}.$$

H. W. Lenstra, Jr., a amélioré l'algorithme, dans une série d'articles dont le premier est [47]. Avec H. Cohen [26], il donne une version beaucoup plus pratique de l'algorithme, remplaçant l'utilisation originale des sommes de Gauss par celle des sommes de Jacobi, qui se prêtent beaucoup mieux aux calculs (elles sont naturellement dans $\mathbb{Z}[\zeta_{p^k}]$). Cet algorithme, implanté par H .Cohen et A. K. Lenstra [25], est le premier à avoir prouvé efficacement la primalité de nombres quelconques de 100 à 200 chiffres décimaux, ce qui représentait un progrès remarquable. Par la suite, les pistes esquissées à la fin de [47] et développées dans [49, 50], ont été explorées avec succès par W. Bosma & M. -P. van der Hulst [18] ainsi que par P. Mihăilescu [54]. Elles concernent principalement l'utilisation pratique du concept d'extension cyclotomique d'un anneau.

4.2. **Courbes elliptiques**

4.2.1. *Préparation théorique.* Soit E une courbe elliptique définie sur $\mathbb{Z}/p\mathbb{Z}, p > 3$. L'ensemble des points de E est $E(\mathbb{Z}/p\mathbb{Z}) = \{(x : y : z) \in \mathbb{P}^2(\mathbb{Z}/p\mathbb{Z}), y^2z = x^3 + axz^2 + bz^3\}$, sur lequel on définit la loi de groupe habituelle, notée $+$. La multiplication par k sur E sera notée $[k]$.

On sait d'après Hasse que le cardinal de $E(\mathbb{Z}/p\mathbb{Z})$ vérifie :

$$|\#E(\mathbb{Z}/p\mathbb{Z}) - (p+1)| < 2\sqrt{p}.$$

Inversement, d'après Deuring (cf. aussi [72]), pour tout entier t vérifiant $|t| < 2\sqrt{p}$, il existe une courbe E dont l'ensemble des points sur $\mathbb{Z}/p\mathbb{Z}$ a cardinal $p+1-t$. Malheureusement, ce théorème ne fournit pas d'algorithme efficace pour trouver E en fonction de t. Si c'était le cas, la primalité serait facile [60].

Pour pouvoir utiliser des courbes elliptiques en primalité, il est nécessaire de pouvoir calculer la cardinalité d'une courbe dans $\mathbb{Z}/p\mathbb{Z}$. Les méthodes élémentaires (formule dite

de Lang et Trotter, algorithme de Shanks) pour cela ne marchent pas en temps polynomial déterministe, et il a fallu attendre l'algorithme de Schoof [67] pour résoudre le problème. En quelques mots, cet algorithme consiste à calculer $t_\ell = t \bmod \ell$ pour de petits nombres premiers ℓ en nombre suffisant pour que $\prod \ell > 4\sqrt{p}$, ce qui permet de déterminer t par application du théorème Chinois.

Il faut également définir ce qu'est une courbe elliptique sur $\mathbb{Z}/N\mathbb{Z}$ quand N est composé, ce que Lenstra a fait dans [51]. On définit

$$E(\mathbb{Z}/N\mathbb{Z}) = \{(x : y : z) \in \mathbb{P}^2(\mathbb{Z}/N\mathbb{Z}), y^2z = x^3 + axz^2 + bz^3\}.$$

On notera $\Delta(E) = 4a^3 + 27b^2$ et on demande que Δ soit inversible modulo N. Notons que le point $O_E = (0 : 1 : 0)$ appartient à $E(\mathbb{Z}/N\mathbb{Z})$. On l'appelle point origine de la courbe. On définit alors une loi de groupe sur $E(\mathbb{Z}/N\mathbb{Z})$ en utilisant des formules données dans [42].

On remarque que si p premier divise N, alors on peut réduire E modulo p et envoyer $E(\mathbb{Z}/N\mathbb{Z})$ dans $E(\mathbb{Z}/p\mathbb{Z})$ par réduction des coordonnées modulo p ; le point origine O_E se réduit en le point à l'infini de la réduction de E modulo p.

Il nous reste maintenant à donner un théorème de primalité.

THÉORÈME 4.3. — *Soient m et s deux entiers tels que $s \mid m$, E une courbe elliptique sur $\mathbb{Z}/N\mathbb{Z}$ et P un point de $E(\mathbb{Z}/N\mathbb{Z})$. Si $[m]P = O_E$ et si pour tout diviseur premier q de s, on a $[m/q]P = (X : Y : Z)$ et $\mathrm{pgcd}(Z, N) = 1$, alors pour tout diviseur premier p de N, on a $\#E(\mathbb{Z}/p\mathbb{Z}) \equiv 0 \bmod s$.*

COROLLAIRE 4.4. — *Si $s > (\sqrt[4]{N} + 1)^2$, alors N est premier.*

Ces résultats généralisent les énoncés de Pocklington. On peut désormais utiliser des facteurs premiers de cardinaux de courbes elliptiques pour fabriquer un DOWNRUN. On en déduit également un certificat de primalité généralisé.

4.2.2. *L'algorithme de Goldwasser et Kilian.* Cet algorithme a été introduit dans [35] (voir la version finale en [36]). Il est décrit à la figure 1. Cet algorithme est paramétré par les deux nombres B_1 et B_2. Le nombre B_1 gouverne le nombre d'essais que l'on s'autorise dans l'algorithme. Pour avoir un algorithme dans RP, ce nombre doit être polynomial en $\log N$. Nous y revenons plus loin. Le nombre B_2 sert à contrôler le nombre d'essais nécessaires à trouver un bon point sur E. Si N est premier, alors la probabilité de trouver un P convenable à cette étape est plus grande que $1/2$.

Que se passe-t-il quand N est composé ? L'algorithme de Schoof teste des identités entre polynômes, à la recherche de t_ℓ. Lors des calculs, il peut rarement apparaître un facteur premier de N, quand une inversion modulo N échoue. Plus sûrement, aucune valeur de t_ℓ ne sera trouvée et cela fournira une preuve de composition pour N. On pourrait imaginer de définir un nombre pseudopremier de Schoof pour (E, ℓ) comme étant un nombre pour

fonction estPremierAvecGK(N)
1. **répéter** B_1 fois
   (a) choisir a, b au hasard modulo N et calculer $g = \text{pgcd}(4a^3 + 27b^2, N)$;
   (b) **si** $g = N$ **alors** aller à (a) ;
   (c) **si** $g \neq 1$ **alors** retourner (`non`, g) ;
   (d) soit E la courbe elliptique d'équation $y^2 = x^3 + ax + b$ définie sur $\mathbb{Z}/N\mathbb{Z}$;
   (e) calculer $m = \#E(\mathbb{Z}/N\mathbb{Z})$ avec l'algorithme de Schoof ;
   (f) **si** $m = 2q$, avec q probablement premier **alors**
      (α) **répéter** B_2 fois
         • choisir au hasard $P \neq O_E$ sur E ;
         • **si** $[m]P \neq O_E$ **alors** retourner `non` ;
         • calculer $[q]P = (X_q : Y_q : Z_q)$ et $g = \text{pgcd}(Z_q, N)$;
         • **si** $g = 1$ **alors** aller à (β) ;
         • **si** $1 < g < N$ **alors** retourner (`non`, g) ;
      (β) **si** GK(q)`==oui` **alors** retourner `oui` ;
2. retourner `je ne sais pas`.

Fig. 1. L'algorithme GK.

lequel on trouve une valeur de t_ℓ. À titre d'exemple, le plus petit N pseudopremier pour $E : y^2 = x^3 + x + 1$ et $\ell = 3$ est $N = 3481 = 59^2$.

L'analyse de l'algorithme GK fait appel au résultat suivant, prouvé par Lenstra [52].

Théorème 4.5. — *Pour $S \subset \mathbb{N}$, soit $S'_p = S \cap [p - \sqrt{p}, p + \sqrt{p}]$. Il existe $c > 0$ tel que*

$$\text{Prob}(\#E(\mathbb{Z}/p\mathbb{Z}) \in S) \geqslant \frac{c(\#S'_p - 2)}{\sqrt{p}\log p},$$

la probabilité se calculant en choisissant (a, b) uniformément dans $\mathbb{Z}/p\mathbb{Z}$ de sorte que $4a^3 + 27b^2 \neq 0$.

Pour analyser GK, on s'intéresse alors à $S = \{m = 2q, q \text{ premier}\}$, ce qui nous ramène à étudier S'_p. Pour assurer la terminaison de GK dans tous les cas, il faut conjecturer le résultat suivant :

Conjecture 4.6. — *Il existe $c_1, c_2 > 0$ tels que pour $x > x_0$: $\pi(x + \sqrt{x}) - \pi(x) \geqslant \frac{c_2\sqrt{x}}{\log^{c_1} x}$.*

Avec cette conjecture, il nous suffirait de prendre pour B_1 une quantité polynomiale en $\log N$. On obtiendrait alors :

Théorème 4.7. — *Si la conjecture 4.6 est vraie, alors* GK *termine en temps polynomial pour tous les nombres.*

La conjecture est beaucoup plus forte que ce que peut apporter l'hypothèse de Riemann classique, mais suivrait de la conjecture de Cramér.

On ne sait d'ailleurs pas prouver que S'_p est non vide. Rappelons que le plus petit exposant δ pour lequel on puisse prouver que $\pi(x+x^\delta)-\pi(x)>0$ est $\delta=0.525$ (voir [12]). Il est clair cependant que certains petits intervalles doivent contenir au moins un nombre premier. En utilisant les résultats de Heath-Brown [37] sur la différence entre nombres premiers consécutifs, il est possible de montrer :

THÉORÈME 4.8. — GK *termine en temps* $O((\log N)^9)$ *en moyenne pour les nombres premiers* $\leqslant x$, *sauf pour ceux de l'ensemble* $\mathcal{E}(x)$ *de cardinal*

$$\#\mathcal{E}(x) \ll \frac{x/\log x}{2^{2\frac{\log\log x}{\log\log\log x}}}.$$

4.2.3. *L'amélioration d'Adleman et Huang.* Superficiellement, cette amélioration (décrite en [2, p. 136]), consiste à remplacer la condition $m=2q$ par $m=\ell q$ avec ℓ premier petit. En utilisant les mêmes travaux de Heath-Brown, et d'autres de Pomerance [59], ils sont à même de montrer que si $\mathcal{E}'(x)$ est l'ensemble des entiers plus petits que x non prouvables par leur algorithme GK-AH, alors :

THÉORÈME 4.9. — *Pour* x *suffisamment grand,* $\#\mathcal{E}'(x)<x^{15/16}$.

4.3. **Courbes de genre 2**

Comme l'algorithme GK-AH pourrait ne pas terminer, Adleman et Huang ont eu l'idée d'utiliser les courbes de genre 2, pour lesquelles les cardinalités des groupes associés varient cette fois dans un intervalle beaucoup plus grand, $[x^2-x^{1.5},x^2]$ au lieu de $[x,x+x^{0.5}]$ comme en genre 1.

Soit p un nombre premier >3. Nous nous intéresserons aux courbes $C(f)$ définies sur $\mathbb{Z}/p\mathbb{Z}$ par une équation du type $y^2=f(x)$ où f est un polynôme de degré 6 à coefficients dans $\mathbb{Z}/p\mathbb{Z}$ et sans racines multiples. Nous noterons $\mathcal{D}(f)$ le cardinal de la jacobienne de $C(f)$. Par Hasse-Weil, on sait que $(\sqrt{p}-1)^4<\mathcal{D}(f)<(\sqrt{p}+1)^4$.

On peut maintenant imaginer généraliser l'algorithme de Schoof au calcul de $\mathcal{D}(f)$, ce qui fournit là encore un algorithme polynomial déterministe. Comme dans le cas du genre 1, l'algorithme ne fournit pas de résultat exploitable quand N est composé, ce qui permet de le repérer.

Nous avons maintenant tous les outils pour pallier le problème de terminaison (hypothétique, rappelons-le) de GK. L'idée de base consiste à construire une suite de nombres premiers $(q_i)_{1\leqslant i\leqslant r}$ tels que la primalité de q_r entraîne celle de q_{r-1}, ..., celle de q_1 celle de N. Contrairement à ce qu'il se passe classiquement, cette fois, la suite q_i sera *croissante*! En effet, on commence par construire q_1 comme le cardinal $\mathcal{D}(f_1)$ d'une jacobienne définie sur $\mathbb{Z}/N\mathbb{Z}$. De même q_2 est construit comme un cardinal premier $\mathcal{D}(f_2)$ d'une jacobienne définie sur $\mathbb{Z}/q_1\mathbb{Z}$, etc. L'algorithme s'arrête quand un des q_i est prouvable par GK-AH.

Nous utilisons ainsi dans notre algorithme un sous-algorithme qui est de type Monte Carlo polynomial.

Le calcul de la probabilité de succès de cet algorithme n'est pas une mince affaire, et nécessite de commencer par généraliser le théorème 4.5. Une fois cela fait, il ne reste plus qu'à estimer le nombre de nombres premiers dans un intervalle très grand, cette fois $I(x) = [x^2 - x^{1.5}, x^2]$. Il suffit d'utiliser un résultat de Iwaniec et Jutila [39], pour minorer le nombre de nombres premiers dans $I(x)$ par $cx^{1.5}/\log x$ pour x assez grand. C'est là l'une des clefs de l'algorithme AH.

On calcule alors le nombres de r-uplets $(q_1, \ldots, q_r)$ construits par l'algorithme à partir d'un nombre initial $q_0 = p$. Le calcul montre que pour $r = 3$, la proportion des bons r-uplets devient significative, et l'algorithme a une probabilité de réussite plus grande que $1/(\log p)^c$ pour un certain entier c, ce qui achève de démontrer :

THÉORÈME 4.10. — *Il existe un algorithme de type Monte Carlo polynomial qui prouve la primalité de tous les entiers.*

Il existe donc finalement un algorithme de Las Vegas pour la primalité.

4.4. **La primalité en pratique**

4.4.1. *Les sommes de Jacobi.* L'algorithme des sommes de Jacobi a été relativement peu implanté. Une des raisons est qu'il s'avère difficile de convaincre de la véracité des calculs effectués, puisqu'aucun certificat n'est délivré. Il faut programmer soi-même l'algorithme, ce qui est chose délicate et qui ne convainc que son auteur. C'est un cas où on rêverait que la technologie de la preuve de correction des programmes soit opérationnelle à ce niveau (voir [21] pour une piste).

L'entier t utilisé dans la description de l'algorithme est le paramètre critique. On doit calculer des sommes de Jacobi associées à des nombres premiers q tels que $q - 1 \mid t$, donc potentiellement de la taille de t. Pour fixer les idées, un nombre de 100 chiffres décimaux nécessite $t = 8400$; si $N \approx 10^{1000}$, alors $t \approx 10^8$ et $N \approx 10^{6000}$ nécessite plutôt $t \approx 6 \cdot 10^9$. L'une des caractéristiques de l'algorithme est la possibilité de précalculer ces sommes une fois pour toutes. Notons que la taille mémoire du résultat n'est pas un problème, même si le temps de calcul et la mémoire nécessaire pour ces précalculs peuvent être très grands. Pour ces grandes tailles, l'utilisation de la factorisation dans les corps cyclotomiques devient rentable [71].

Au-delà de ces premières remarques, il faut utiliser les améliorations présentées dans [18, 54] pour aller plus vite. Parmi celles-ci, notons l'utilisation de facteurs de $N^w - 1$ pour w petit. P. Mihăilescu détient le record actuel avec le nombre $N = 2^{10000} + 177$ (3011 chiffres décimaux) depuis novembre 1997.

4.4.2. *Les courbes.* L'algorithme de Schoof n'est pas implantable tel quel. Il a fallu attendre les améliorations d'Elkies et Atkin pour disposer d'un algorithme de calcul de la

cardinalité de courbes elliptiques fonctionnant en temps polynomial, pratique et efficace, mais cette fois randomisé (voir [68, 30, 56, 17] par exemple). Malgré ces améliorations, l'algorithme SEA ne permet pas d'envisager de traiter des nombres de plus de quelques centaines de chiffres décimaux, et l'algorithme GK, même dopé par ces changements, n'est pas efficace.

Dès 1986, Atkin a eu l'idée d'utiliser la réduction de courbes elliptiques à multiplication complexe, dont on sait calculer la cardinalité facilement. Ces courbes remplacent les courbes aléatoires utilisées dans GK-AH. L'algorithme, appelé ECPP est décrit dans [9, 24, 29] (voir également [28]), des améliorations données au cours du temps par l'auteur (consulter sa page web par exemple). ECPP a été analysé de façon heuristique dans [44] : la complexité en temps devrait être de l'ordre de $O((\log N)^{6+\varepsilon})$ ou $\tilde{O}((\log N)^5)$ avec de l'arithmétique rapide ($\tilde{O}((\log N)^4)$ avec des précalculs, voir également [57]).

Deux implantations sont connues : celle de l'auteur, maintenue depuis 12 ans, et dont une version se trouve dans le logiciel MAGMA, la seconde réalisée par Marcel Martin (PRIMO[4]). Le record actuel, dû à l'auteur, est la primalité du nombre $2177^{580} + 580^{2177}$, qui a 6016 chiffres décimaux (cf. [57])[5].

L'algorithme AH est-il pratique ? Rappelons qu'il sert essentiellement de bouée de sauvetage théorique pour l'algorithme GK-AH. Bien sûr, on pourrait imaginer y avoir recours dans le cas où on ne trouverait pas de bonnes courbes dans ECPP. Pour cela, il faudrait songer à remplacer la recherche des petits facteurs de nombres de la taille de N, par des nombres de taille N^2, ou même N^8, tout cela avec de multiples allers et retours entre les différents algorithmes...

Les problèmes d'implantation sont pour le moment formidables. Les meilleures implantations d'algorithmes à la Schoof pour des courbes de genre 2 permettent au plus de traiter le cas de $p = 5 \cdot 10^{24} + 41$ (voir le récent record de Gaudry et Schost annoncé dans la liste `NMBRTHRY@LISTSERV.NODAK.EDU` allant au-delà de [33]). Quand bien même on mettrait au point une variante d'ECPP en genre 2, appelons-là HECPP (et qui serait elle aussi heuristique), les calculs liés à la multiplication complexe dans ce cadre sont encore loin d'être aussi faciles et agréables que dans le cas du genre 1 (cf. [73]).

4.5. **Conclusion sur la pratique**

Pour des nombres relativement petits, disons plus petits que 10^{1000}, les deux algorithmes (sommes de Jacobi, ECPP) ont des temps de calcul raisonnables, n'exécedant pas une dizaine d'heures sur une machine normale. Le problème pratique de prouver la primalité est donc résolu. Pour des nombres plus grands, les résultats dépendent de beaucoup trop de paramètres pour que l'auteur se risque à des réponses définitives.

[4] `http ://www.ellipsa.net/`

[5] Le précédent record était détenu par Jose Luis Gomez Pardo, avec PRIMO, pour un nombre de 5878 chiffres décimaux, en février 2003 ; cf. `http ://www.ellipsa.net/pages/primorecord.html`.

5. LES TRAVAUX D'AGRAWAL, KAYAL, SAXENA

5.1. Première idée

Dans [5], les auteurs présentent un test de composition très simple utilisant la proposition suivante :

PROPOSITION 5.1. — *N est premier si et seulement si le polynôme $P(X) = (X+1)^N - X^N - 1$ est identiquement nul modulo N.*

Démonstration. — Il suffit pour cela de se rappeler que si N est premier, alors $N \mid \binom{N}{k}$ pour tout entier k, $0 < k < N$. Si N est composé, soit p un de ses facteurs premiers, et a la valuation p-adique de N. Alors $\binom{N}{p}$ est divisible exactement par p^{a-1}. □

Cette proposition ne conduit pas à un algorithme efficace, puisque le polynôme $P(X)$ a un degré trop élevé. Par contre, on peut élaborer un test de composition de la façon suivante :

fonction estPremierAB(N)

1. Si N est une puissance de nombre entier, retourner `non`.

2. Choisir un polynôme aléatoire $Q(X)$ de degré $O(\log N)$ à coefficients dans $\mathbb{Z}/N\mathbb{Z}$. Si $(X+1)^N \equiv X^N + 1 \bmod (Q(X), N)$ alors retourner `oui`, sinon retourner `non`.

Les auteurs montrent alors que la probabilité d'échec de l'algorithme est bornée par $1 - 1/(4 \log N)$, là où Solovay et Strassen donnent $1/2$. Ils conjecturent également :

CONJECTURE 5.2. — *Si N est composé, alors il existe $1 \leqslant r \leqslant \log N$ tel que $P(X)$ n'est pas divisible par $X^r - 1$ modulo N.*

Notons que le test généralise le test de Fermat pour $r = 1$, dans ce cas il est équivalent à $2^N \equiv 2 \bmod N$. Si N passe le test pour $r = 1$, il le passe aussi pour $r = 2$ (car $(X+1)^N = X^N + 1 \bmod (X+1, N)$ quand N est impair).

L'auteur a testé cette conjecture sur les nombres pseudopremiers[6] en base 2 plus petits que 10^{12} à l'aide de MAPLE. Il y a 24 nombres qui passent les tests pour $1 \leqslant r \leqslant 8$, 8 étant la valeur maximale atteinte. Le plus petit est 597717121, le plus grand 880731910801 (notons que $\log 597717121 \approx 20.20$).

5.2. Le théorème original

C'est celui que l'on peut extraire de [6], suivant en cela [13] (remplacé depuis par [14]) et inspiré en partie par [64]. L'idée est de réussir à combiner des tests de pseudoprimalité pour obtenir une preuve de primalité pour N.

[6] Grâce au fichier envoyé il y a fort longtemps par S. S. Wagstaff, Jr.

THÉORÈME 5.3. — *Soient N un entier qui ne s'écrit pas M^k pour M et k entiers, $k > 1$. Soient s un entier positif, r un nombre premier et q le plus grand facteur premier de $r-1$. On suppose que*

(i) [condition arithmétique] $N^{(r-1)/q} \bmod r \notin \{0, 1\}$ *;*

(ii) [condition combinatoire]

$$\binom{q-1+s}{s} > N^{2\lfloor\sqrt{r}\rfloor}. \tag{3}$$

(iii) [divisibilité élémentaire] N n'a pas de facteur premier $p \leqslant s$;

(iv) [tests de pseudoprimalité] $(X-a)^N \equiv X^N - a \bmod (X^r-1, N)$ *pour tout* $1 \leqslant a \leqslant s$.

Alors N est premier.

Démonstration. — Supposons que N soit composé. Il existe un facteur premier p de N tel que $p^{(r-1)/q} \bmod r \notin \{0, 1\}$, car sinon cela contredirait (i). D'après (iii), on a également $p > s$.

Par réduction modulo p, la propriété (iv) implique que pour tout a dans $\mathcal{A} = \{1, \ldots, s\}$:

$$(X-a)^N \equiv X^N - a \bmod (X^r - 1, p).$$

Comme p est premier, on a aussi

$$(X-a)^p \equiv X^p - a \bmod (X^r - 1, p).$$

L'idée est de combiner ces deux relations.

LEMME 5.4. — Soient m_1 et m_2 deux entiers. Si

$$(X-a)^{m_1} \equiv X^{m_1} - a \bmod (X^r - 1, p),$$

$$(X-a)^{m_2} \equiv X^{m_2} - a \bmod (X^r - 1, p),$$

alors $(X-a)^{m_1 m_2} \equiv X^{m_1 m_2} - a \bmod (X^r - 1, p)$.

Démonstration. — Il existe $g(X) \in \mathbb{F}_p[X]$ tel que

$$(X-a)^{m_2} - (X^{m_2} - a) = (X^r - 1)g(X).$$

D'où :

$$(X^{m_1} - a)^{m_2} - (X^{m_1 m_2} - a) = (X^{m_1 r} - 1)g(X^{m_1}).$$

Comme $X^r - 1 \mid X^{m_1 r} - 1$, on en déduit

$$(X-a)^{m_1 m_2} \equiv (X^{m_1} - a)^{m_2} \equiv X^{m_1 m_2} - a \bmod (X^r - 1, p).$$

□

On déduit de cela qu'en fait, pour tous les entiers positifs i, j, et tout $a \in \mathcal{A}$:

$$(X-a)^{p^i N^j} \equiv X^{p^i N^j} - a \bmod (X^r - 1, p).$$

On utilise alors l'argument combinatoire suivant. Posons $L = \{p^i N^j, 0 \leqslant i, j \leqslant \lfloor \sqrt{r} \rfloor\}$. Comme N n'est pas une puissance d'un nombre entier, il ne s'écrit *a fortiori* pas comme p^k. Par suite, tous les éléments de L sont distincts et $\#L = (\lfloor \sqrt{r} \rfloor + 1)^2 > r$. Il existe donc deux éléments qui sont congrus modulo r, ce qu'on écrit :

$$m_1 = p^{i_1} N^{j_1}, m_2 = p^{i_2} N^{j_2} = m_1 + kr, (i_1, j_1) \neq (i_2, j_2)$$

avec par exemple $m_1 \leqslant m_2$. On obtient alors pour tout $a \in \mathcal{A}$:

$$(X-a)^{m_2} \equiv X^{m_1 + kr} - a \equiv X^{m_1} - a \equiv (X-a)^{m_1} \bmod (X^r - 1, p).$$

Il ne nous reste plus qu'à démontrer que $m_1 = m_2$, ce qui impliquera $N = p^t$, contredisant l'hypothèse faite.

À ce stade, on choisit un facteur irréductible du r-ième polynôme cyclotomique modulo p, soit $h(X)$. On sait que $h(X)$ est de degré d, l'ordre de p modulo r (voir par exemple [38]). Par le choix de p, $d \geqslant q$. On considère $F = \mathbb{F}_p[X]/(h(X))$ le corps fini à p^d éléments, et $\theta = X \bmod (h(X), p)$. Par hypothèse :

$$\forall\, a \in \mathcal{A}, (X-a)^{m_1} \equiv (X-a)^{m_2} \bmod (h(X), p).$$

On introduit alors le monoïde S de F^* engendré par les $\theta - a$ pour $a \in \mathcal{A}$.

Lemme 5.5. — Soit T le sous-ensemble de S formé des produits $\prod_{a=1}^{s} (\theta - a)^{\alpha_a}$ avec

$$\sum_{a=1}^{s} \alpha_a \leqslant q - 1, \alpha_a \geqslant 0. \tag{4}$$

Alors

(i) les éléments de T sont tous distincts ;

(ii) le cardinal de T est $\binom{q-1+s}{s}$.

Démonstration. — (i) Tous les $X - a$ pour $a \in \mathcal{A}$ sont irréductibles et distincts dans $\mathbb{F}_p[X]$, puisque $p > s$. Tous les polynômes $\prod_{a=1}^{s} (X-a)^{\alpha_a}$ avec les α_a satisfaisant (4) sont de degré strictement plus petit que d, et deux à deux distincts dans $\mathbb{F}_p[X]$. Deux d'entre eux ne peuvent être égaux dans F que s'ils diffèrent d'un multiple de $h(X)$, ce qui est impossible car leur degré est $< q \leqslant d$.

(ii) C'est un résultat classique de combinatoire. On met les solutions de (4) en bijection avec les parties à s éléments de l'intervalle $[1, q-1+s]$ de la façon suivante. Si (α_a) est une solution, on pose $\beta_1 = \alpha_1 + 1$, $\beta_2 = \alpha_1 + \alpha_2 + 2$, …, $\beta_s = \alpha_1 + \alpha_2 + \cdots + \alpha_s + s$. La suite β_a vérifie $1 \leqslant \beta_1 < \beta_2 < \cdots < \beta_s \leqslant q - 1 + s$. Le nombre de suites (β_a) possibles est bien la quantité annoncée, appelée traditionnellement nombre de combinaisons avec répétition [27]. □

Retour au théorème.

Les éléments de S et *a fortiori* de T sont tous racines du polynôme $Y^{m_1} - Y^{m_2} = Y^{m_1}(Y^{m_2-m_1} - 1) = Y^{m_1}P(Y)$. Comme $m_2 - m_1 \leqslant N^{2\lfloor\sqrt{r}\rfloor} < \binom{q-1+s}{s} \leqslant \#T$, le polynôme $P(Y)$ a plus de racines que son degré, donc il est identiquement nul et $m_1 = m_2$. □

Il est facile de déduire un algorithme du théorème 5.3, si on se donne r et s, ce que nous ferons dans la section qui suit.

PROPOSITION 5.6. — *Le temps de calcul de* AKS *est* $O(sr^2(\log N)^3)$ *si on utilise des algorithmes classiques de multiplication et* $\tilde{O}(sr(\log N)^2)$ *si on utilise de la multiplication rapide, aussi bien pour les polynômes que pour les entiers.*

Démonstration. — Le temps de calcul est largement dominé par le temps passé à vérifier la condition (iv). Pour chaque valeur de a, le calcul de $(X-a)^N \bmod (X^r-1, N)$ demande $O(\log N)$ multiplications $A(X)B(X) \bmod (X^r - 1)$ où $A(X)$ et $B(X)$ ont degré $O(r)$. La réduction ne coûte rien car $X^r - 1$ est creux. Notant $\mathcal{P}(N, r)$ le temps nécessaire à multiplier deux polynômes de degré r à coefficients dans $\mathbb{Z}/N\mathbb{Z}$, le temps de calcul est donc $O(s(\log N)\mathcal{P}(N, r))$. En utilisant les algorithmes naï fs de multiplication, on obtient un temps de calcul $O(sr^2(\log N)^3)$. Avec des FFT partout, cela devient $\tilde{O}(sr(\log N)^2)$. □

5.3. **Le choix de r et s**

Pour achever de démontrer que la primalité est décidable en temps polynomial déterministe, il nous faut montrer que l'on peut choisir r et s comme puissances de $\log N$.

On commence par minorer brutalement $\binom{q-1+s}{s}$ par $(q/s)^s$, puis on impose $q \geqslant 2s$. Si on choisit s tel que $2^s \geqslant N^{2\lfloor\sqrt{r}\rfloor}$, alors la condition (ii) est certainement vérifiée. Par suite, il suffit d'assurer l'existence d'un nombre premier r tel que $r-1$ ait un facteur premier q plus grand que $4\sqrt{r}\log N/\log 2$.

Notant $P(n)$ le plus grand facteur premier de n, il faut donc être capable d'estimer le nombre de nombres premiers r ayant une grande valeur de $P(r-1)$. Intéressons-nous à la quantité

$$\mathcal{P}_\delta(x) = \#\{p \text{ premier } \leqslant x, P(p-1) > x^\delta\}$$

pour $\delta > 0$. Nous nous intéressons ici au cas où il y a suffisamment de nombres premiers dans l'ensemble, c'est-à-dire au cas où $\mathcal{P}_\delta(x) \geqslant c_\delta\pi(x)$ pour x assez grand. Suite aux travaux de Fouvry [31] ainsi que de Baker et Harman [11], on sait que la plus grande valeur de δ pour laquelle $\mathcal{P}_\delta(x) \geqslant c_\delta\pi(x)$ est $\delta = 0.676$. Pour notre propos, toute valeur de $\delta \geqslant 2/3$ ou même $> 1/2$ (en utilisant [34] comme suggéré par Pomerance) suffirait.

Nous aurons également besoin de majorations effectives de $\pi(x)$. Celles de [66] donnent : $x/\log x < \pi(x) \leqslant \gamma x/\log x$ pour $x \geqslant 114$.

Nous sommes maintenant prêts à montrer :

THÉORÈME 5.7. — *Soit $\delta \in]1/2, 0.676]$ et $\alpha = 2/(2\delta - 1)$. Il existe deux constantes c_1 et $c_2 > c_1$ telles qu'il existe un nombre premier r dans l'intervalle $I_\alpha = [c_1(\log N)^\alpha, c_2(\log N)^\alpha]$ tel que $r-1$ ait un facteur premier $q \geqslant 4\sqrt{r}\log N/\log 2$ et pour lequel l'ordre de N modulo r soit divisible par q.*

Démonstration. — On commence par remarquer que si $r \in I_\alpha$, on a $r^\delta \geqslant 4\sqrt{r}\log N/\log 2$, les exposants des $\log N$ étant égaux, à condition de prendre $c_1^\delta \geqslant 4c_2^{1/2}/\log 2$ et aussi $c_2 > (4/\log 2)^{2/(2\delta-1)} = (4/\log 2)^\alpha$ (puisque c_2 doit être plus grand que c_1).

Appelons *convenables* les nombres premiers cherchés et comptons leur nombre $\mathcal{R}(N)$. Nous noterons pour alléger $L = \log N$. Ce nombre est :

$$\begin{aligned}
\mathcal{R}(N) &= \mathcal{P}_\delta(c_2 L^\alpha) - \mathcal{P}_\delta(c_1 L^\alpha) \\
&\geqslant \mathcal{P}_\delta(c_2 L^\alpha) - \pi(c_1 L^\alpha) \\
&\geqslant c_\delta \frac{c_2 L^\alpha}{\log c_2 + \alpha \log L} - \pi(c_1 L^\alpha) \\
&\geqslant \frac{L^\alpha}{(\log c_2 + \alpha \log L)(\log c_1 + \alpha \log L)}(c_\delta c_2 \log c_1 - \gamma c_1 \log c_2 + (c_\delta c_2 - \gamma c_1)\alpha \log L).
\end{aligned}$$

Il suffit de trouver c_1 tel que $c_1 < c_\delta c_2/\gamma$. N'oublions pas que $c_1 \geqslant (4c_2^{1/2})^{1/\delta}$. Ces deux relations sont compatibles à partir du moment où $c_2^{1-1/(2\delta)} \geqslant 4^{1/\delta}\gamma/c_\delta$, ce qui est faisable quand $1 - 1/(2\delta) > 0$. Par suite

$$\mathcal{R}(N) \geqslant c_3 \frac{(\log N)^\alpha}{\log\log N}$$

pour une certaine constante $c_3 > 0$ et N assez grand.

On cherche maintenant un r de l'intervalle tel que $r-1$ ait un facteur premier q vérifiant $N^{(r-1)/q} \not\equiv 1 \bmod r$. Pour cela, on considère le produit :

$$\Pi = (N-1)(N^2-1)\cdots(N^v-1).$$

Il suffit de trouver r convenable et $q \mid r-1$ tels que $(r-1)/q \leqslant v$ et r ne divise pas Π. Or le nombre de diviseurs premiers de Π est au plus $\log N^{v(v+1)/2}/\log 2 \leqslant v^2 \log N$. Si $v^2 \log N < \mathcal{R}(N)$, alors on est sûr de l'existence d'un r convenable ne divisant pas Π. Il suffit de prendre $v = c_4(\log N)^{(\alpha-2)/2}$ pour cela.

La condition $(r-1)/q \leqslant v$ sera satisfaite si

$$c_2^{1-\delta}(\log N)^{(1-\delta)\alpha} \leqslant c_4(\log N)^{(\alpha-2)/2}$$

c'est-à-dire $c_4 \geqslant c_2^{1-\delta}$. □

On a donc montré, en prenant $r = O((\log N)^\alpha)$, $s = O((\log N)^{\alpha/2+1})$, que :

THÉORÈME 5.8. — *Pour tout $\delta \in]1/2, 0.676]$, il existe un algorithme de primalité déterministe dont le temps de calcul est $O((\log N)^{(8\delta+1)/(2\delta-1)})$ si on utilise de l'arithmétique naïve, et $\tilde{O}((\log N)^{6\delta/(2\delta-1)})$ si on utilise les FFT.*

COROLLAIRE 5.9. — estPremier?$\in$ P.

Si l'on prend $\delta = 2/3$, on obtient les exposants 19 et 12. Remarquons que si on pouvait faire tendre δ vers 1, on obtiendrait au mieux $\tilde{O}((\log N)^6)$. Notons qu'une façon d'obtenir $\delta = 1$ serait de prouver l'existence d'un nombre suffisant de nombres premiers de Sophie Germain, i.e. r tel que $(r-1)/2$ soit également premier.

5.4. **L'après AKS**

De multiples améliorations de toutes natures ont fleuri depuis la parution de l'article, concernant des généralisations et améliorations de l'algorithme de base. D. Bernstein maintient une page résumant les différents travaux [14]. La situation est loin d'être stabilisée à la date où je tape ces lignes.

Les résultats basés sur les théorèmes de Fouvry ou Baker/Harman ne sont pas effectifs, ce qui fait qu'on ne sait pas à partir de quelle taille de N les estimations s'appliquent. De plus, les contraintes imposées au nombre premier r auxiliaire sont très fortes.

H. W. Lenstra, Jr. (communication personnelle, cf. également [14]), a amélioré l'algorithme au point de prouver un temps de calcul effectif en $\tilde{O}((\log N)^{12})$ ou bien $\tilde{O}((\log N)^8))$ en utilisant de nouveau le théorème de Fouvry. Des résultats similaires ont été trouvés par S. David (communication personnelle). Rapidement, l'amélioration essentielle tient à une relaxation des contraintes sur r, qui peut être maintenant un entier quelconque. Une condition combinatoire ressemblant à (3) est utilisée en ses lieux et place.

D'autres améliorations ont été apportées par d'autres chercheurs, comme la lecture de [14] en fait foi.

5.5. **Vers un algorithme pratique**

L'algorithme AKS n'est pas un algorithme efficace dans sa version de base, essentiellement parce que les degrés des polynômes en jeu sont bien trop grands. En effet, au minimum, r doit être de taille $O((\log N)^2)$ pour satisfaire les conditions du théorème. On constate de plus que la complexité prouvée est pire que celle des sommes de Jacobi pour les nombres de taille raisonnable, ou bien celle d'ECPP, même heuristique. Les récentes avancées semblent augurer de l'existence d'une version randomisée d'AKS qui sera peut-être plus rapide dans la pratique.

Signalons pour commencer que l'article [6] contient une conjecture intrigante, tant elle paraît simple.

CONJECTURE 5.10. — *Soit r un entier tel que $r \nmid N^2 - 1$. Si*

$$(X+1)^N \equiv X^N + 1 \bmod (N, X^r + 1) \tag{5}$$

alors N est premier.

Si cela était vrai, alors on disposerait d'un algorithme déterministe en $\tilde{O}((\log N)^3)$. Cette conjecture a été testée numeriquement pour des nombres plus petits que 10^{10} (cf. [40]). Nous extrayons de ce travail le résultat suivant. Supposons que $N = 10k + 3$. Alors la relation (5) est équivalente aux deux équations :

$$5^{(N-1)/2} \equiv -1 \bmod N, \left(\frac{\theta+1}{\theta-1}\right)^{(N+1)/2} \equiv -1 \bmod N$$

où $\theta^2 - 5 = 0$. Ces deux relations demandent donc à N d'être pseudopremier d'Euler en base 5 et pseudopremier de Lucas pour θ. Cela n'est pas sans rappeler les combinaisons de tests proposés dans [61] (voir aussi [29, Ex. 3.41, p. 156]) et pour lesquels aucun contre-exemple n'est connu à ce jour.

P. Berrizbeitia [16], suivi de Q. Cheng [23], a proposé une amélioration dans le cas où $N - 1$ admet un facteur premier r pour lequel $r^\alpha \mid\mid N - 1$, $r \geqslant \log^2 N$. On provoque une explosion combinatoire à l'aide des racines r-ièmes de l'unité dans $\mathbb{Z}/N\mathbb{Z}$, ce qui donne une complexité en $\tilde{O}((\log N)^4)$ pour l'algorithme qui est randomisé.

Suivant en cela un schéma classique, passer de $N-1$, puis à $N+1$ (comme Berrizbeitia), il ne reste plus qu'à généraliser au cas où il existe deux entiers e et d tel que $e \mid N^d - 1$, ce qu'a fait Bernstein [15]. La complexité tombe encore à $\tilde{O}((\log N)^4)$ pour une version randomisée, cette fois pour tous les nombres. Dans la suite logique, P. Mihăilescu (communications personnelles) a annoncé très récemment un algorithme combinant AKS et cyclotomie.

6. CONCLUSIONS

Nous avons tenté de faire un tour d'horizon du problème de la primalité, en mêlant considérations théoriques (complexité, mathématiques), et algorithmiques, voire pratiques. Maintenant que estPremier? est montré être dans P, l'histoire s'arrête-t-elle là ? Sans doute pas. Tout d'abord, la chasse au meilleur exposant ne fait que commencer. En outre, une version pratique de AKS n'est encore que frémissante. L'avenir dira quand elle détrônera les méthodes ancestrales.

Remerciements. Je remercie G. Hanrot pour sa relecture précise et ses nombreuses suggestions, P. Gaudry pour sa relecture et de nombreuses discussions sur le sujet. Merci à tous ceux avec qui j'ai eu des discussions sur certains points des différents algorithmes : H. Cohen, S. David, H. W. Lenstra, Jr., P. Mihăilescu, J.-L. Nicolas, C. Pomerance, O. Ramaré, J. Rivat.

RÉFÉRENCES

[1] L. M. Adleman. On distinguishing prime numbers from composite numbers. In *Foundations of computer science*, pages 387–406. IEEE, 1980. 21st FOCS, Syracuse, USA, Proceedings.

[2] L. M. Adleman and M.-D. A. Huang. *Primality testing and Abelian varieties over finite fields*, volume 1512 of *Lecture Notes in Math.* Springer–Verlag, 1992.

[3] L. M. Adleman, C. Pomerance, and R. S. Rumely. On distinguishing prime numbers from composite numbers. *Ann. of Math. (2)*, 117:173–206, 1983.

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

[5] M. Agrawal and S. Biswas. Primality and identity testing via chinese remaindering. In *Proc. Ann. IEEE Symp. Found. Comp. Sci.*, pages 202–209, 1999.

[6] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. Preprint; available at `http://www.cse.iitk.ac.in/primality.pdf`, August 2002.

[7] W. R. Alford, A. Granville, and C. Pomerance. There are infinitely many Carmichael numbers. *Ann. of Math. (2)*, 139(3):703–722, May 1994.

[8] M. Artjuhov. Certain criteria for the primality of numbers connected with the little Fermat theorem (russian). *Acta Arith.*, 12:355–364, 1966/67.

[9] A. O. L. Atkin and F. Morain. Elliptic curves and primality proving. *Math. Comp.*, 61(203):29–68, July 1993.

[10] E. Bach. Explicit bounds for primality testing and related problems. *Math. Comp.*, 55(191):355–380, July 1990.

[11] R. C. Baker and G. Harman. The Brun-Titschmarsh theorem on average. In *Proceedings of a conference in Honor of Heini Halberstam*, volume 1, pages 39–103, 1996.

[12] R. C. Baker, G. Harman, and J. Pintz. The difference between consecutive primes. II. *Proc. London Math. Soc. (3)*, 83(3):532–562, 2001.

[13] D. Bernstein. An exposition of the Agrawal-Kayal-Saxena primality-proving theorem. Preprint, August 2002.

[14] D. Bernstein. Proving primality after Agrawal-Kayal-Saxena. `http://cr.yp.to/papers/aks.ps`, January 2003.

[15] D. Bernstein. Proving primality in essentially quartic expected time. `http://cr.yp.to/papers/quartic.ps`, January 2003.

[16] P. Berrizbeitia. Sharpening "Primes is in P" for a large family of numbers. `http://arxiv.org/abs/math.NT/0211334`, November 2002.

[17] I. Blake, G. Seroussi, and N. Smart. *Elliptic curves in cryptography*, volume 265 of *London Math. Soc. Lecture Note Ser.* Cambridge University Press, 1999.

[18] W. Bosma and M.-P. van der Hulst. *Primality proving with cyclotomy*. PhD thesis, Universiteit van Amsterdam, December 1990.

[19] J. Brillhart, D. H. Lehmer, and J. L. Selfridge. New primality criteria and factorizations of $2^m \pm 1$. *Math. Comp.*, 29(130):620–647, 1975.

[20] J. Brillhart, D. H. Lehmer, J. L. Selfridge, B. Tuckerman, and S. S. Wagstaff, Jr. *Factorizations of* $b^n \pm 1$, $b = 2, 3, 5, 6, 7, 10, 11, 12$ *up to high powers*. Number 22 in Contemporary Mathematics. AMS, 2 edition, 1988.

[21] O. Caprotti and M. Oostdijk. Formal and efficient primality proofs by use of computer algebra oracles. *J. Symbolic Comput.*, 32:55–70, 2001.

[22] H. F. Chau and H.-K. Lo. Primality test via quantum factorization. *Internat. J. Modern Phys. C*, 8(2):131–138, 1997.

[23] Q. Cheng. Primality proving via one round in ECPP and one iteration in AKS. `www.cs.ou.edu/~qcheng/paper/aksimp.pdf`, January 2003.

[24] H. Cohen. *A course in algorithmic algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer–Verlag, 1996. Third printing.

[25] H. Cohen and A. K. Lenstra. Implementation of a new primality test. *Math. Comp.*, 48(177):103–121, 1987.

[26] H. Cohen and H. W. Lenstra, Jr. Primality testing and Jacobi sums. *Math. Comp.*, 42(165):297–330, 1984.

[27] L. Comtet. *Analyse combinatoire*. Presses Universitaires de France, 1970.

[28] D. A. Cox. *Primes of the form* $x^2 + ny^2$. John Wiley & Sons, 1989.

[29] R. Crandall and C. Pomerance. *Prime numbers – A Computational Perspective*. Springer Verlag, 2000.

[30] N. D. Elkies. Elliptic and modular curves over finite fields and related computational issues. In D. A. Buell and J. T. Teitelbaum, editors, *Computational Perspectives on Number Theory: Proceedings of a Conference in Honor of A. O. L. Atkin*, volume 7 of *AMS/IP Studies in Advanced Mathematics*, pages 21–76. American Mathematical Society, International Press, 1998.

[31] E. Fouvry. Théorème de Brun-Titschmarsh; application au théorème de Fermat. *Invent. Math.*, 79:383–407, 1985.

[32] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.

[33] P. Gaudry and R. Harley. Counting points on hyperelliptic curves over finite fields. In W. Bosma, editor, *Algorithmic Number Theory*, volume 1838 of *Lecture Notes in Comput. Sci.*, pages 313–332. Springer Verlag, 2000. 4th International Symposium, ANTS-IV, Leiden, The Netherlands, July 2000, Proceedings.

[34] M. Goldfeld. On the number of primes p for which $p + a$ has a large prime factor. *Mathematika*, 16:23–27, 1969.

[35] S. Goldwasser and J. Kilian. Almost all primes can be quickly certified. In *Proc. 18th STOC*, pages 316–329. ACM, 1986. May 28–30, Berkeley.

[36] S. Goldwasser and J. Kilian. Primality testing using elliptic curves. *Journal of the ACM*, 46(4):450–472, July 1999.

[37] D. R. Heath-Brown. The differences between consecutive primes. *J. London Math. Soc. (2)*, 18(1):7–13, 1978.

[38] K. Ireland and M. Rosen. *A classical introduction to modern number theory*, volume 84 of *Graduate Texts in Mathematics*. Springer, 1982.

[39] H. Iwaniec and M. Jutila. Primes in short intervals. *Ark. Mat.*, 17(1):167–176, 1979.

[40] N. Kayal and N. Saxena. Towards a deterministic polynomial-time primality test. Technical report, IIT Kanpur, 2002.
`http://www.cse.iitk.ac.in/research/btp2002/primality.html`.

[41] D. E. Knuth. *The Art of Computer Programming: Seminumerical Algorithms*. Addison-Wesley, 2nd edition, 1981.

[42] H. Lange and W. Ruppert. Complete systems of addition laws on abelian variety. *Invent. Math.*, 79:603–610, 1985.

[43] D. H. Lehmer. Strong Carmichael numbers. *J. Austral. Math. Soc. Ser. A*, 21:508–510, 1976.

[44] A. K. Lenstra and H. W. Lenstra, Jr. Algorithms in number theory. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume A: Algorithms and Complexity, chapter 12, pages 674–715. North Holland, 1990.

[45] A. K. Lenstra and H. W. Lenstra, Jr., editors. *The development of the number field sieve*, volume 1554 of *Lecture Notes in Math.* Springer, 1993.

[46] H. W. Lenstra, Jr. Miller's primality test. *Inform. Process. Lett.*, 8(2):86–88, 1979.

[47] H. W. Lenstra, Jr. Primality testing algorithms (after Adleman, Rumely, Williams). *Séminaire Bourbaki*, 576, 1980-1981.

[48] H. W. Lenstra, Jr. Primality testing. In *Computational methods in number theory, Part I*, pages 55–77. Math. Centrum, Amsterdam, 1982.

[49] H. W. Lenstra, Jr. Primality testing with Artin symbols. In *Number theory related to Fermat's last theorem (Cambridge, Mass., 1981)*, volume 26 of *Progr. Math.*, pages 341–347. Birkhäuser Boston, Mass., 1982.

[50] H. W. Lenstra, Jr. Galois theory and primality testing. In I. Reiner and K. W. Roggenkamp, editors, *Orders and their applications*, volume 1142 of *Lecture Notes in Math.*, pages 169–189. Springer, 1984. Proc. of a conference, Oberwolfach, June 3–9, 1984.

[51] H. W. Lenstra, Jr. Elliptic curves and number-theoretic algorithms. In *Proceedings of the International Congress of Mathematicians, Vol. 1, 2 (Berkeley, Calif., 1986)*, pages 99–120, Providence, RI, 1987. Amer. Math. Soc.

[52] H. W. Lenstra, Jr. Factoring integers with elliptic curves. *Ann. of Math. (2)*, 126:649–673, 1987.

[53] A. Menezes, P. C. van Oorschot, and S. A. Vanstone. *Handbook of applied cryptography*. CRC Press, 1997.

[54] P. Mihăilescu. *Cyclotomy of rings and primality testing*. Diss. ETH No. 12278, Swiss Federal Institute of Technology Zürich, 1997.

[55] G. L. Miller. Riemann's hypothesis and tests for primality. In *Proc. 7th STOC*, pages 234–239, 1975.

[56] F. Morain. Calcul du nombre de points sur une courbe elliptique dans un corps fini : aspects algorithmiques. *J. Théor. Nombres Bordeaux*, 7:255–282, 1995.

[57] F. Morain. Implementing the asymptotically fast version of the elliptic curve primality proving algorithm. In preparation, June 2003.

[58] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1995.

[59] C. Pomerance. Analysis and comparison of some integer factoring algorithms. In H. W. Lenstra, Jr. and R. Tijdeman, editors, *Computational methods in number theory*, pages 89–140. Mathematisch Centrum, Amsterdam, 1982. Mathematical Center Tracts 154/155.

[60] C. Pomerance. Very short primality proofs. *Math. Comp.*, 48(177):315–322, 1987.

[61] C. Pomerance, J. L. Selfridge, and Samuel S. Wagstaff, Jr. The pseudoprimes to 25.10^9. *Math. Comp.*, 35(151):1003–1026, 1980.

[62] V. R. Pratt. Every prime has a succint certificate. *SIAM J. Comput.*, 4:214–220, 1975.

[63] M. Rabin. Probabilistic algorithms for testing primality. *J. Number Theory*, 12:128–138, 1980.

[64] J. Radhakrishnan. Primes in P. *Bull. of the EATCS*, 78:61–65, October 2002.

[65] P. Ribenboim. *The new book of prime number records*. Springer-Verlag, 1996.

[66] J. B. Rosser and L. Schoenfeld. Approximate formulas for some functions of prime numbers. *Illinois J. Math.*, 6:64–94, 1962.

[67] R. Schoof. Elliptic curves over finite fields and the computation of square roots mod *p*. *Math. Comp.*, 44:483–494, 1985.

[68] R. Schoof. Counting points on elliptic curves over finite fields. *J. Théor. Nombres Bordeaux*, 7:219–254, 1995.

[69] P. W. Shor. Algorithms for quantum conputation: Discrete logarithms and factoring. In *Proceedings 26th Annual ACM Symposium on Theory of Computing (STOC)*, pages 124–134, Montreal, Canada, 1994. ACM.

[70] R. Solovay and V. Strassen. A fast Monte-Carlo test for primality. *SIAM J. Comput.*, 6(1):84–85, 1977. Erratum, ibid, volume 7, 1, 1978.

[71] P. van Wamelen. Jacobi sums over finite fields. *Acta Arith.*, 102.1:1–20, 2002.

[72] E. Waterhouse. Abelian varieties over finite fields. *Ann. Sci. École Norm. Sup.*, 2:521–560, 1969.

[73] A. Weng. Constructing hyperelliptic curves of genus 2 suitable for cryptography. *Math. Comp.*, 72:435–458, 2003.

[74] H. C. Williams and J. O. Shallit. Factoring integers before computers. In *Mathematics of Computation 1943–1993: a half-century of computational mathematics (Vancouver, BC, 1993)*, volume 48 of *Proc. Sympos. Appl. Math.*, pages 481–531. Amer. Math. Soc., Providence, RI, 1994.

F. Morain

Laboratoire d'Informatique
École polytechnique (LIX)
CNRS/UMR 7650, INRIA-Futurs
F-91128 Palaiseau Cedex
E-mail : `morain@lix.polytechnique.fr`