In *Nouvelles Méthodes Mathématiques en Cryptographie*, Fascicule *Journées Annuelles*, pages 23–53. Société Mathématique de France, June 2007.

ALGORITHMES POUR RÉSOUDRE LE PROBLÈME DU LOGARITHME DISCRET DANS LES CORPS FINIS

par

Antoine Joux & Reynald Lercier

Résumé. — Avec la publication d'un grand nombre de schémas cryptographiques à base d'accouplements de Weil sur courbes elliptiques ou à base de tores algébriques, la résolution du problème du logarithme discret dans le groupe multiplicatif d'un corps fini fait l'objet d'un intérêt renouvelé. Dans ce texte, nous nous intéressons à trois algorithmes récents qui permettent de résoudre ce problème en toute généralité, sans limitation sur le degré ou la caractéristique du corps, et qui sont de complexités similaires à celles des algorithmes connus pour les corps premiers et les corps de caractéristique deux.

Abstract **(Algorithms to solve the finite field discrete logarithm problem)**
Numerous publications on the use in cryptography of elliptic curve Weil pairings and algebraic torus renew interest in solving the finite field discrete logarithm problem. In this article, we state three recent algorithms to solve this problem in full generality, without any limit on the degree or the characteristic of the field, and with the same complexities as those known for prime or characteristic two fields.

Table des matières

Classification mathématique par sujets **(2000).** — 11T99, 11Y05, 11Y16, 12E20, 68Q25, 94A60.
Mots clefs. — Cryptographie, Complexités d'Algorithmes, Corps Finis, Logarithmes Discrets.

1. Introduction

Soit G un groupe fini cyclique et g l'un de ses générateurs, alors pour tout élément y de G, il existe un entier positif x tel que $y = g^x$. Le plus petit de ces entiers est appelé l'index, ou le logarithme discret, de y en base g. Il est souvent noté $\log_g(y)$. De façon analogue à la fonction du logarithme népérien, la fonction $\log_g$ satisfait modulo le cardinal de G la formule $\log_g yz = \log_g y + \log_g z$ dont une application immédiate permet de ramener le calcul de la loi de groupe à une addition. Obtenir le produit de deux éléments d'un corps fini par cette technique est séduisant, mais il est apparu très vite que cela est inapproprié pour des corps de grandes tailles, car il n'est plus possible de calculer par avance une table. En fait, si d'un point de vue algorithmique, calculer y à partir de x est facile, en particulier par la méthode communément appelée « exponentiation binaire » [21], l'inverse est aujourd'hui considéré comme difficile pour certains groupes.

La cryptologie, avec l'invention de la cryptographie à clef publique au milieu des années 70, a su tirer parti de cette difficulté. On cherche à y mettre en évidence des algorithmes et protocoles cryptographiques ayant, d'un côté une complexité petite, fonction polynomiale de la taille de la clef, et d'un autre côté une sécurité essentiellement équivalente à la résolution d'un problème pour lequel aucun algorithme de complexité polynomiale en la taille de la clef n'est connu. Les schémas de type OAEP [5] sont typiques de cette démarche. On dispose d'une réduction qui ramène leur sécurité au problème de la non inversibilité de la permutation à trappe RSA [4], dont l'étude nécessite de quantifier précisément la difficulté de factoriser le produit de deux nombres premiers. Les meilleurs algorithmes connus pour la factorisation d'entiers N étant d'une part de complexité asymptotique sous-exponentielle, égale à $e^{(\sqrt[3]{64/9}+o(1))\,\ln^{\frac{1}{3}} N \ln^{\frac{2}{3}} \ln N}$ pour le crible algébrique [26], et d'autre part de complexités réelles connues sur des instances précises (*cf.* TAB. 1 où l'unité « GIPS years » correspond approximativement à un ordinateur effectuant un milliard d'opérations élémentaires par seconde pendant un an), on peut en déduire ce que seraient les complexités réelles pour des entiers de tailles bien supérieures à ceux que l'on peut espérer attaquer. En confrontant ensuite le résultat à des heuristiques sur les puissances de calculs réalisables dans le futur, à partir d'estimateurs comme, par exemple, la loi de Moore selon laquelle à coût égal les capacités des ordinateurs doublent tous les 18 mois, on arrive à des propositions motivées pour les tailles de clefs. Pour illustrer, nous donnons en troisième ligne de TAB. 2 extrait de [28], des dimensionnements RSA recommandables pour les années futures.

Le protocole interactif de négociation de clef dû à Diffie-Hellman [12] et ses nombreux dérivés à des fins, entre autres, de chiffrement ou de signature, motivent de même l'étude de familles de groupes finis de toutes tailles qui sont à la fois « pratiques » et « sûrs ». On doit disposer d'algorithmes de complexité polynomiale, pour d'abord, étant donnée une taille $\lceil \ln \ell \rceil$, obtenir une représentation explicite d'un

Nombre	Taille (chiffres)	Quand	Complexité (GIPS year)	Méthode	Qui
C116	116	1990	0.3	MPQS	Lenstra *et al.*
RSA-120	120	Juin 1993	0.8	MPQS	Lenstra *et al.*
RSA-129	129	Avril 1994	5	MPQS	Lenstra *et al.*
RSA-130	130	Avril 1996	1	NFS	Lenstra *et al.*
RSA-140	140	Fév. 1999	2	NFS	Lenstra *et al.*
RSA-155	155	Août 1999	8	NFS	Lenstra *et al.*
C158	158	Janv. 2002	3.4	NFS	Franke *et al.*
RSA-160	160	Mars 2003	2.7	NFS	Franke *et al.*
RSA-576	174	Déc. 2003	13	NFS	Franke *et al.*
C176	176	Mai 2005	49	NFS	Aoki *et al.*
RSA-640	193	Nov. 2005	80	NFS	Franke *et al.*
RSA-200	200	Mai 2005	170	NFS	Franke *et al.*

TABLE 1. Records de factorisation d'entiers

groupe G à ℓ éléments, et pour ensuite évaluer la loi de groupe, tout en s'assurant au contraire que la résolution du problème du logarithme discret, ou de l'une de ses variantes Diffie-Hellman, nécessite des algorithmes de complexité non polynomiale. En fait, les meilleurs algorithmes génériques pour calculer des logarithmes dans un groupe arbitraire G de cardinal ℓ sont de complexité en temps au mieux $O(\ell^{\frac{1}{2}})$, exponentielle donc en $\ln \ell$ [34]. Pour atteindre cette borne, on applique dans une première phase la méthode de Pohlig-Hellman pour ramener le calcul du logarithme discret recherché à celui de logarithmes discrets dans les sous-groupes cycliques de G dont les ordres sont des diviseurs de ℓ premiers entre eux. On déduit ensuite dans une seconde phase chacun de ces logarithmes discrets, de façon déterministe avec la méthode des « pas de bébés, pas de géants », ou mieux de façon probabiliste avec des méthodes de type « Pollard-ρ ». Dans le pire cas, *i.e.* ℓ premier, la première phase est triviale, la seconde est de complexité celle qui est annoncée. Les groupes les plus utilisés dans les applications sont ceux qui sont définis par des courbes elliptiques données dans un corps fini [30], car les algorithmes de résolution connus sont génériques, de complexités exponentielles donc en fonction de la taille du groupe.

Ici, nous nous focalisons sur les groupes multiplicatifs de corps finis $\mathbb{F}_q$, à l'origine de la cryptographie Diffie-Hellman, mais maintenant moins populaires, du moins dans une utilisation directe, en raison d'attaques sous-exponentielles spécifiques. Cependant, un intérêt renouvelé se fait sentir, en particulier pour mesurer la sécurité de nombre de schémas cryptographiques récents faisant usage d'accouplements de Weil. Dans cette voie, par mimétisme avec la factorisation d'entiers, il est nécessaire de

quantifier l'efficacité d'un algorithme de résolution, d'abord par sa complexité asymptotique, fonction de la taille $\ln q$ du corps, ensuite par les tailles de corps atteignables en pratique, de façon à avoir une estimation du temps d'exécution pour tout q.

L'essentiel des travaux publiés sur le sujet concerne des corps finis pour lesquels soit le degré n, soit la caractéristique p, sont considérés comme fixe : typiquement les corps premiers $\mathbb{F}_p$ ou les corps de caractéristique deux $\mathbb{F}_{2^n}$. Ainsi, on connaît pour le cas des corps premiers un algorithme de complexité identique à celui du crible algébrique pour la factorisation d'entiers, *i.e.* $e^{(\sqrt[3]{64/9}+o(1))\,\ln^{\frac{1}{3}} p \ln^{\frac{2}{3}} \ln p}$ [**33**], et des algorithmes de complexité similaire pour $\mathbb{F}_{2^n}$, mais avec une constante plus petite, précisément $e^{(\sqrt[3]{32/9}+o(1))\,n^{\frac{1}{3}} \ln^{\frac{2}{3}} n}$ [**3, 18**]. En tenant compte des mêmes heuristiques sur les puissances de calculs réalisables à moyen et long terme, on arrive aussi à une appréciation crédible de la difficulté du logarithme discret dans ces cas (*cf.* Tab. 2).

Année	1982	2000	2005	2010	2020	2050
Cryptographie symétrique	56	70	74	78	86	109
Logarithmes discrets dans $\mathbb{F}_p$ ou $\mathbb{F}_{2^n}$ & factorisation d'entiers	417	952	1149	1369	1881	4047
Logarithmes discrets sur des courbes elliptiques dans $\mathbb{F}_p$ ou $\mathbb{F}_{2^n}$	105	132	139	146	161	206

Table 2. Estimations sur les tailles de problèmes atteignables de 1982 à 2050 (en bits, extrait de [**28**])

Il faut attendre le début des années 2000 avec la publication de nouveaux schémas, ceux à base de tores algébriques [**25, 27, 32**] et ceux à base d'accouplements de Weil (et ses variantes) sur courbes elliptiques [**15, 6, 7**], pour que l'on considère sérieusement l'usage de corps de caractéristiques et degrés moyens, de tailles croissantes en fonction de la taille des clefs. Du coup, l'étude des problèmes du logarithme discret correspondants, sans limitation sur le degré ou la caractéristique, est récente [**18, 19**] et nous nous proposons dans ce texte d'en faire état. Pour l'essentiel, nous décrivons trois algorithmes. Ils résultent tous d'une méthode de crible appelée « index-calculus » [**29**] (ou « calcul d'index ») dont nous présentons, un exemple à l'appui, les grandes idées dans le paragraphe 2. Sur cette base, le premier algorithme que nous explicitons, dans le paragraphe 3, permet de traiter des corps finis de petites caractéristiques et le deuxième algorithme, dans le paragraphe 4, est destiné aux corps de grandes caractéristiques. Nous rentrons dans le vif du sujet avec le paragraphe 5, en généralisant ces algorithmes au cas des corps de caractéristiques et degrés qui tendent simultanément vers l'infini en fonction de $\ln q$. Nous montrons ainsi qu'ils sont, pour les corps dont $\ln p$ est négligeable (resp. prépondérant) devant toutes fonctions de la frontière $O(\ln^{\frac{1}{3}} q \ln^{\frac{2}{3}} \ln q)$ (resp. $O(\ln^{\frac{2}{3}} q \ln^{\frac{1}{3}} \ln q)$), de complexités identiques à celles déjà connues pour des corps de caractéristiques

ou degrés fixés, *i.e.* $e^{(\sqrt[3]{32/9}+o(1))\,\ln^{\frac{1}{3}} q \ln^{\frac{2}{3}} \ln q}$ (resp. $e^{(\sqrt[3]{64/9}+o(1))\,\ln^{\frac{1}{3}} q \ln^{\frac{2}{3}} \ln q}$). Puis, pour la plage intermédiaire, nous exhibons un troisième algorithme, de complexité $e^{(\sqrt[3]{128/9}+o(1))\,\ln^{\frac{1}{3}} q \ln^{\frac{2}{3}} \ln q}$. Enfin, nous terminons par l'analyse des complexités aux frontières et mettons en évidence, de façon surprenante, un algorithme de complexité $e^{(\sqrt[3]{3}+o(1))\,\ln^{\frac{1}{3}} q \ln^{\frac{2}{3}} \ln q}$ lorsque $\ln p = \sqrt[3]{1/9}\ln^{\frac{1}{3}} q \ln^{\frac{2}{3}} \ln q$.

2. Algorithmes de type « Index calculus »

De nombreux algorithmes, parmi lesquels ceux qui sont connus à ce jour comme les plus efficaces pour factoriser des entiers, résoudre le problème du logarithme discret dans des corps finis, ou certaines familles de courbes algébriques, calculer le cardinal du groupe de classe d'un corps de nombre, *etc.* sont de type « index-calculus ». On attribue à Kraitchik la découverte de ce procédé [**22, 23**].

Dans un premier temps, nous dégageons les grandes lignes de cette méthode, appliquée au calcul de logarithmes discrets. Il apparaît que son efficacité dépend fortement des options prises à l'initialisation. Nous précisons donc ensuite les choix qui sont à l'origine des algorithmes des paragraphes 3, 4 et 5.

2.1. Principes. — Les méthodes de type « index-calculus » procèdent comme suit pour résoudre le problème du logarithme discret dans un sous-groupe cyclique $< g >$, de cardinal ℓ, d'un corps fini $\mathbb{F}_q$.

Étape 1 (crible) : on fixe un sous-ensemble $\mathcal{S} = \{\gamma_1, \ldots, \gamma_{|\mathcal{S}|}\}$ de $< g >$ appelé la « base de lissité » et l'on cherche des relations faisant intervenir les éléments de $\mathcal{S}$ de la forme $\prod_{(\epsilon,\gamma)\in\mathbb{Z}\times\mathcal{S}} \gamma^{\epsilon} = 1$, ce qui conduit à des équations du type

$$\sum_{(\epsilon,\gamma)\in\mathbb{Z}\times\mathcal{S}} \epsilon \log_g \gamma = 0 \pmod{\ell} . \tag{1}$$

Étape 2 (algèbre linéaire) : quand on a suffisamment de relations (1), on calcule des résidus $\log_g \gamma$ en inversant modulo ℓ le système linéaire correspondant.

Étape 3 (résolution) : pour déduire le logarithme discret d'un élément quelconque y de $< g >$, on essaie des entiers aléatoires ν jusqu'à ce que $g^{\nu} y$ s'écrive à son tour comme un produit d'éléments de $\mathcal{S}$, c'est-à-dire $g^{\nu} y = \prod_{(\epsilon,\gamma)\in\mathbb{Z}\times\mathcal{S}} \gamma^{\epsilon}$. Alors, $\log_g y = -\nu + \sum_{(\epsilon,\gamma)\in\mathbb{Z}\times\mathcal{S}} \epsilon \log_g \gamma \pmod{\ell}$.

Clairement, la complexité en temps associée dépend de la stratégie utilisée pour fixer la base $\mathcal{S}$ et pour rechercher des relations.

2.2. Étude d'un cas d'école. — Nous nous plaçons dans $\mathbb{F}_p$ avec $\ell = (p-1)/2$, un nombre premier. Étant donné $\gamma \in \mathbb{Z}/p\mathbb{Z}$, on définit aussi $\gamma^{\uparrow}$ comme étant le plus petit entier positif de la classe de congruence γ modulo p. Adjoindre à un générateur g l'ensemble des nombres premiers inférieurs à une borne B est un choix naturel pour la base de lissité $\mathcal{S}$. Si la factorisation sur $\mathbb{Z}$ d'entiers de la forme $g^{\nu^{\uparrow}}$ pour des entiers

ν choisis aléatoirement ne fait intervenir que des nombres premiers de $\mathcal{S}$, on en déduit alors une relation du type (1). Quand on en a suffisamment, *i.e.* au moins autant que d'éléments de $\mathcal{S}$, l'algèbre linéaire et la résolution finale sont réalisées comme indiqué au paragraphe 2.1.

2.2.1. Exemple. — Supposons que l'on souhaite résoudre le problème du logarithme discret dans le sous-groupe cyclique $< 1193 >$ de $\mathbb{F}_{10007}$, d'ordre $\ell = 2 \cdot 5003$. Choisissons comme base $\mathcal{S} = \{2, 3, 5, 7, 11, 13, 17\}$, on trouve assez vite huit relations multiplicatives ne faisant intervenir que des éléments de $\mathcal{S}$:

$$1193^{15^\uparrow} = 2 \cdot 3 \cdot 7 \cdot 11\,,\ 1193^{36^\uparrow} = 7^2 \cdot 11^2\,,\ 1193^{41^\uparrow} = 17^3\,,$$
$$1193^{47^\uparrow} = 2 \cdot 11 \cdot 13 \cdot 17\,,\ 1193^{73^\uparrow} = 3 \cdot 5 \cdot 11 \cdot 13\,,\ 1193^{74^\uparrow} = 2^5 \cdot 3^2 \cdot 5^2\,,$$
$$1193^{78^\uparrow} = 2^6 \cdot 3 \cdot 7^2\,,\ 1193^{80^\uparrow} = 2^3 \cdot 5^2\,.$$

En combinant ces équations, il n'est alors pas difficile de voir que

$$2 = 1193^{4764}\,,\ 3 = 1193^{236}\,,\ 5 = 1193^{7903}\,,\ 7 = 1193^{638}\,,$$
$$11 = 1193^{4383}\,,\ 13 = 1193^{2560}\,,\ 17 = 1193^{3349}\,.$$

Maintenant, pour connaître le logarithme discret d'un élément y n'appartenant pas à la base il nous reste à trouver une puissance de 1193, qui multipliée par y, se décompose sur $\mathcal{S}$. Pour par exemple $y = 8964$, on a

$$(y \cdot 1193^{12})^\uparrow = 2^2 \cdot 3^3 \cdot 5 \cdot 17\,,\ \text{et donc } 8964 = 1193^{1464}\,.$$

2.2.2. Complexité asymptotique. — Une fois rappelés quelques résultats sur la densité des nombres friables, il est relativement aisé de prouver que l'algorithme précédent est de complexité sous-exponentielle.

2.2.2.1. Rappels sur les fonctions L. — Étant donnés des réels N, ν et $\lambda \neq 0$, définissons pour N tendant vers l'infini, la quantité

$$\mathcal{L}_N(\nu, \lambda) = e^{\lambda (\ln N)^\nu (\ln \ln N)^{1-\nu}}\,.$$

On utilise ces fonctions pour des estimations asymptotiques de complexité et classiquement, on abrège $L_N(\nu, \lambda)$ pour $\mathcal{L}_N(\nu, \lambda + o(1))$ ou encore $L_N(\nu)$ pour $\mathcal{L}_N(\nu, \lambda + o(1))$. Pour $\nu = 0$, on trouve une complexité polynomiale de degré λ en la taille $\ln N$. Au contraire, pour $\nu = 1$, on a une complexité exponentielle (en $\ln N$) égale à N^λ. Pour une valeur de ν intermédiaire, on fait souvent référence à une complexité sous-exponentielle. Ces fonctions vérifient,

$$\begin{aligned} L_N(\nu_1) L_N(\nu_2) &= L_N(\max(\nu_1, \nu_2)) \text{ si } \nu_1 \neq \nu_2 \text{ et} \\ L_N(\nu, \lambda_1) L_N(\nu, \lambda_2) &= L_N(\nu, \lambda_1 + \lambda_2)\,. \end{aligned}$$

Dans la suite, nous utilisons la notion d'entiers ou de polynômes « B-lisses ». Cela signifie que ces entiers ou ces polynômes ont leurs facteurs premiers plus petits que la borne B. On a en corollaire d'un théorème de Canfield, Erdös et Pommerance [9], le résultat suivant sur les entiers B-lisses.

Theorème 2.1. — *Étant donné des constantes ν, λ, ω et μ telles que $1 \geqslant \nu > \omega > 0$ et $\lambda, \mu > 0$, la probabilité qu'un entier de l'ordre de $L_N(\nu, \lambda)$ ait tous ses facteurs premiers plus petit que $L_N(\omega, \mu)$ est, lorsque N tend vers l'infini, asymptotiquement égale à*

$$L_N(\nu - \omega, -\lambda \cdot \frac{\nu - \omega}{\mu}) .$$

2.2.2.2. Analyse. — Supposons que $B = L_p(\theta)$ pour $0 \leqslant \theta \leqslant 1$. On cherche des entiers de l'ordre de $L_p(1)$ qui soient B-lisse. La probabilité d'occurrence de telles relations est $L_p(1-\theta)$, il en faut $L_p(\theta)$ et donc, le temps de la première étape est $L_p(\max(1-\theta, \theta))$.

Le temps de la seconde étape, celle de l'inversion modulaire, est polynomial en le cardinal de $\mathcal{S}$, c'est-à-dire égal à $L_p(\theta)$.

Enfin, dans la dernière étape, on recherche un entier B-lisse de l'ordre de $L_p(1)$, ce qui est réalisable en temps $L_p(1-\theta)$.

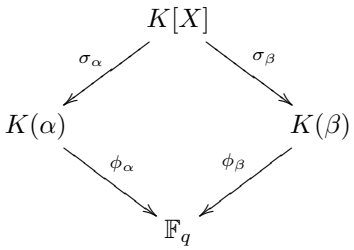
Le temps total est ainsi $L_p(\max(\theta, 1-\theta))$. Il vaut donc au minimum $L_p(1/2)$, réalisé pour $\theta = 1/2$.

Remarque 2.1. — *Une analyse attentive permet de s'assurer que la complexité de cet algorithme probabiliste ne repose sur aucune heuristique. Ce n'est malheureusement plus le cas des algorithmes qui suivent, principalement parce que l'on cherche à y factoriser des entiers ou des polynômes de tailles plus petites, qui ne sont plus complètement aléatoires.*

2.3. Généralisations. — Considérer la lissité d'objets directement dans $\mathbb{F}_q$ est inapproprié, simplement parce que tout élément y est une unité. C'est pourquoi, déjà dans la version naïve du paragraphe précédent, on se place par l'intermédiaire de l'opération $^{\uparrow}$ dans $\mathbb{Z}$. Les algorithmes les plus modernes de calcul de logarithme discrets contournent la difficulté en obtenant des bases de lissité par réduction modulo des idéaux premiers dans l'anneau des entiers d'un corps global. Les corps globaux considérés sont soit les corps de nombres pour les algorithmes efficaces dans le cas des corps de grandes caractéristiques et l'on parle de « Number Field Sieve » (NFS), soit les corps de fonctions algébriques pour les corps de petites caractéristiques et l'on parle de « Function Field Sieve » (FFS). Les objets utilisés dans les deux cas sont suffisamment simples pour que l'on puisse présenter ces algorithmes d'un point de vue « global » à l'aide du dictionnaire classique :

corps de nombres	$\leftrightarrow$	corps de fonctions,
nombres algébriques	$\leftrightarrow$	fonctions algébriques,
idéaux	$\leftrightarrow$	places,
groupe des classes	$\leftrightarrow$	groupe de Picard,
	etc.	

En fait, la situation peut être résumée par le diagramme commutatif



où le corps K est égal au localisé en p (resp. en un polynôme f irréductible de degré n) de $\mathbb{Q}$ (resp. $\mathbb{F}_p[X]$), où α et β sont des entiers algébriques (resp. fonctions algébriques) dont les polynômes minimaux ont une racine commune γ dans $\mathbb{F}_q$ (γ est racine de f quand $K = \mathbb{F}_p[X]_{(f)}$) et où les flèches sont définies par $\sigma_\alpha : X \longrightarrow \alpha$, $\sigma_\beta : X \longrightarrow \beta$, $\phi_\alpha : \alpha \longrightarrow \gamma$ et $\phi_\beta : \beta \longrightarrow \gamma$. Tout comme pour $\mathbb{F}_q$, considérer la lissité d'objets dans les anneaux $\mathbb{Q}_{(p)}[X]$ ou $\mathbb{F}_p(t)_{(f)}[X]$ conduit à une impasse, car la majeure partie des éléments y est irréductible. Par contre, ce n'est plus le cas au niveau intermédiaire, celui des corps de nombres ou de fonctions. C'est pourquoi les bases de lissité $\mathcal{S}$ qui sont utilisées par la suite proviennent d'idéaux premiers de normes bornées dans l'anneau des entiers de ces corps.

Schématiquement, ces algorithmes recherchent des $(t+1)-$uplets $(u_0, u_1, \ldots, u_t)$ de rationnels ou de polynômes de $\mathbb{F}_p[t]$ tels que les idéaux principaux de générateurs $u_0+u_1\alpha+\cdots+u_t\alpha^t$ et $u_0+u_1\beta+\cdots+u_t\beta^t$ se factorisent simultanément sur l'ensemble des idéaux premiers de $\mathcal{S}$. Lorsque l'une de ces relations est découverte, il est possible de la réduire modulo un idéal ou une place de corps résiduel égal à $\mathbb{F}_q$ pour obtenir une égalité entre les deux « factorisations réduites ».

À ce stade, plusieurs questions se posent et sont l'objet des paragraphes suivants. Les principales sont les suivantes.

- D'un point de vue théorique, comment lever les obstructions liées à l'utilisation d'anneaux non factoriels avec des unités non triviales ?
- D'un point de vue algorithmique, quels sont les meilleurs choix pour les corps de nombres ou de fonctions et, une fois ces choix faits, comment organiser efficacement les calculs ?

3. Algorithmes pour les corps de caractéristiques fixées

L'algorithme que nous avons présenté au paragraphe 2.2 dans le cas des corps premiers peut être généralisé à tout corps fini. Dans le cas des corps dont la caractéristique est fixée lorsque l'on fait tendre le cardinal vers l'infini, il n'est pas difficile de montrer, de façon similaire, mais à partir de probabilités de factorisation pour des polynômes plutôt que pour des entiers, que la complexité asymptotique est aussi égale à $L_q(1/2)$.

Dès 1984, Coppersmith exhibe un algorithme de complexité asymptotique meilleure, en $L_q(1/3, c)$ avec c variant entre $(32/9)^{\frac{1}{3}}$ et $4^{\frac{1}{3}}$ selon que le degré du corps est plus ou moins proche d'une puissance de p [10]. C'est à notre connaissance, tout problème confondu (factorisation, logarithmes discrets, *etc.*), le premier algorithme de type index-calculus avec une complexité plus faible que $L_q(1/2)$.

Des travaux d'Adleman, dix ans plus tard, éclairent la problématique sous un jour nouveau en montrant qu'une méthode de type index-calculus construite à l'aide de corps de fonctions algébriques conduit aussi à un algorithme en $L_q(1/3)$, mais de constante plus grande, puisque de complexité égale à $L_q(1/3, (64/9)^{\frac{1}{3}})$ [1].

Plus proche de nous, Adleman avec l'aide de Huang, améliore en 1999 son algorithme et arrive à une complexité uniforme en $L_q(1/3, (32/9)^{\frac{1}{3}})$ [3]. Malheureusement, cet algorithme, comme le précédent, est en pratique nettement moins efficace que l'algorithme de Coppersmith. En fait, il faut attendre 2002 avec une idée de Joux et Lercier pour au final obtenir un algorithme de même complexité asymptotique [16], et qui est en pratique plus rapide que la méthode de Coppersmith (*cf.* TAB. 3).

Nous faisons un tour d'horizon de ces méthodes dans le paragraphe 3.1 et nous les illustrons dans le paragraphe 3.3 en donnant quelques éléments sur un calcul que nous avons mené pour $\mathbb{F}_{2^{607}}$ en 2005 (*cf.* TAB. 3). Nous nous focalisons ensuite sur un dernier algorithme, publié en 2006, de Joux et Lercier [18]. Celui-ci est tout comme notre « cas d'école » assez simple à comprendre, en particulier on n'y a plus besoin de corps de fonctions. Du coup, il est très facile à mettre en œuvre et, bonne surprise, sa complexité est aussi en $L_q(1/3, (32/9)^{\frac{1}{3}})$. Nous décrivons donc cet algorithme de façon détaillée dans le paragraphe 3.2, exemple à l'appui. L'analyse de sa complexité est évoquée au paragraphe 5.

Corps	Taille (chiffres)	Quand	Complexité (GIPS year)	Méthode	Qui
$\mathbb{F}_{2^{401}}$	121	1992	0.2	COPPERSMITH	Gordon, McCurley
$\mathbb{F}_{2^{521}}$	157	2002	0.4	FFS	Joux, Lercier
$\mathbb{F}_{2^{607}}$	183	2002	20	COPPERSMITH	Thomé
$\mathbb{F}_{2^{607}}$ $\mathbb{F}_{2^{613}}$	183	2005	1.6	FFS	Joux, Lercier

TABLE 3. Records pour le problème du logarithme discret dans $\mathbb{F}_{2^n}$

3.1. Cribles à l'aide de corps de fonctions algébriques. — On se donne deux corps de fonctions algébriques. Tout d'abord $\mathbb{F}_p(X)$ défini par $\mathbb{F}_p(X)[Y]/(a_1(X)Y - a_0(X))$, et le corps défini par une courbe plane $\mathcal{C}$ de jacobienne $\mathcal{J}$ définie par un polynôme $H(X, Y)$ sur $\mathbb{F}_p$ tel que le numérateur de $H(X, a(X))$, où $a(x) = a_0(X)/a_1(X)$,

soit divisible par un polynôme irréductible $f(X)$ de degré n. Nous notons γ, une racine de $f(X)$ dans $\mathbb{F}_q$.

Dans $\mathbb{F}_p(X)$, il est immédiat de passer des places aux fonctions génératrices de ces derniers, car elles sont principales, et l'on peut réduire simplement dans $\mathbb{F}_q$ par $(u+vY) \to u+va(\gamma)$. Dans $\mathcal{C}$, par contre, le passage des places aux fonctions ne peut se faire qu'une fois élevé à la puissance le cardinal de la jacobienne de la courbe. Pour toute place $\mathfrak{p}$ de $\mathcal{C}$, il existe une fonction $\pi_\mathfrak{p} \in \mathbb{F}_p(X,Y)$ telle que $\mathfrak{p}^h = \operatorname{div}(\pi_\mathfrak{p})$ où $h = |\mathcal{J}(\mathbb{F}_q)|$. À une place $\mathfrak{p}$ correspond donc l'élément $\pi_\mathfrak{p}(\gamma, a(\gamma))^{1/h}$ de $\mathbb{F}_q$. La base de lissité se déduit alors de

$$\mathcal{S} = \{\text{places à l'infini}\} \cup \{\pi \in \mathbb{F}_p[X], \pi \text{ irréductible de degré } < B_\alpha\} \cup \left\{\mathfrak{p}, \begin{array}{l}\text{places irréductibles de degré relatif 1}\\ \text{au-dessus de leur restriction à } \mathbb{F}_p(X)\end{array}, \deg(\mathfrak{p}) < B_\beta\right\},$$

où B_α et B_β sont deux paramètres donnés.

L'étape 1 de la méthode générique du paragraphe 2 consiste à rechercher des couples $(u(X), v(X)) \in \mathbb{F}_p[X]^2$ tels que $u(X) + v(X)\cdot X$ et $\operatorname{div}(u(X) + v(X)\cdot Y)$ soient tous deux $\mathcal{S}-$lisses, car alors

$$\left(U \prod_{\pi \in S} \pi(\gamma)^{e_\pi}\right)^{p-1} = (u(\gamma) + v(\gamma)\cdot a(\gamma))^{p-1} = \left(\prod_{\mathfrak{p}\in S} \pi_\mathfrak{p}(\gamma, a(\gamma))^{e_\mathfrak{p}/h}\right)^{p-1}.$$

Le coefficient c de la complexité heuristique $L_q(1/3, c)$ de ces algorithmes est surtout fonction de la taille des coefficients et des degrés des polynômes qui définissent les corps de fonctions utilisés. On dispose de quatre constructions.

3.1.1. Méthode de Coppersmith (1984). — Soit $\mathbb{F}_{2^n} \simeq \mathbb{F}_2[X]/(f(X))$ où $f(X) = X^n + a(X)$ et $a(X)$ a un degré bas. Alors, on définit avec d une puissance de deux et $e > n/d$,

$$H(X,Y) = Y^d + X^{de-n}a(X) \quad (= Y^d + X^{de} (\operatorname{mod} f(X))).$$

Cette construction n'est optimale que lorsque n est une puissance de deux. Cependant la méthode est séduisante en pratique, car la moitié des relations nécessaires s'obtient automatiquement (elles proviennent de la factorisation, modulo tout polynôme irréductible de $\mathbb{F}_2$, de $Y^d + X^{de-n}a(X)$ en un polynôme de degré un élevé à la puissance d). Elle conduit à $\sqrt[3]{32/9} \leqslant c \leqslant \sqrt[3]{4}$.

3.1.2. Méthode de Adleman (1994). — Soit $\mathbb{F}_q \simeq \mathbb{F}_p[X]/(f(X))$, la méthode comme décrit par Adleman est une adaptation de la méthode de la « base m » utilisée pour factoriser des entiers avec NFS.

On choisit un polynôme $a(X)$ au hasard de degré $n/(d+1)$ pour un degré d fixé par avance. On écrit $f(X)$ comme $f(X) = \sum_{i=0}^{d} h_i(X)a(X)^i$ et au final les corps de fonctions sont définis par $Y - a(X)$ et $H(X,Y) = \sum_{i=0}^{d} h_i(X)Y^i$. Adleman ajoute

huit conditions techniques sur H, la plupart d'entre elles pour mieux contrôler les places à l'infini de la courbe définie par $\mathcal{C}$. Cela donne $c = \sqrt[3]{64/9}$.

3.1.3. Méthode de Adleman et Huang (1999). — Soit $\mathbb{F}_q \simeq \mathbb{F}_p[X]/(f(X))$, Adleman et Huang proposent une construction similaire à celle de NFS pour factoriser des entiers de forme spéciale. Soit $f(X) = X^n + a(X)$ où $a(X)$ est de petit degré, alors les polynômes de définition des corps de fonctions algébriques sont simplement $Y - X^e$ et

$$H(X,Y) \quad = \quad Y^d - X^{ed-n}a(X)\,,$$

où $e = \lceil n/d \rceil$ pour un paramètre d fixé par avance. Cela donne $c = \sqrt[3]{32/9}$.

3.1.4. Méthode de Joux et Lercier (2002). — L'idée derrière cette dernière construction est d'effectuer le calcul à l'envers. On fixe d'abord un polynôme $H(X,Y)$ de degré d en Y et de bas degré en X. On choisit au hasard $a_0(X)$ et $a_1(X)$ de degrés au plus $\lfloor n/d \rfloor$ jusqu'à ce que $f(X) = a_1(X)^d H(X, a_0(X)/a_1(X))$ soit irréductible de degré n. Le polynôme à l'origine du second corps de fonctions est alors $a_1(X)Y - a_0(X)$ et $\mathbb{F}_q$ est défini par $\mathbb{F}_p[X]/(f(X))$.

On combine cette construction avec l'utilisation de courbes $C_{a,b}$ comme suggéré par Matsumoto (1999). Dans ce cas, on a une unique valuation à l'infini. On a aussi le résultant en Y de $H(X,Y)$ et $a_1(X)Y - a_0(X)$ qui est exactement égal à $f(X)$. Cela donne $c = \sqrt[3]{32/9}$.

3.2. Variante sans corps de fonctions. — Cet algorithme, extrait de [18], prend en entrée un corps fini $\mathbb{F}_q$ de caractéristique p et degré n, ainsi qu'en paramètre supplémentaire un entier strictement positif D borné par n. Nous renvoyons le lecteur au paragraphe 5 pour la détermination d'un optimum pour D quand q est donné.

3.2.1. Définition. — Il se décompose en quatre parties : l'initialisation, le crible, l'algèbre linéaire et la résolution finale.

3.2.1.1. Initialisation. — Soient $d_\alpha \approx \sqrt{nD}$ et $d_\beta \approx \sqrt{n/D}$, deux entiers avec $d_\alpha d_\beta \geqslant n$. On choisit deux polynômes $f_\alpha(X)$ et $f_\beta(X)$ à coefficients dans $\mathbb{F}_p$ de degrés d_α et d_β tels que le polynôme $f_\beta(f_\alpha(X)) - X$ ait un facteur irréductible $f(X)$ de degré n. Ce facteur nous fournit une représentation polynomiale pour $\mathbb{F}_q$, autrement dit $\mathbb{F}_q \simeq \mathbb{F}_p[X]/(f(X))$.

3.2.1.2. Crible. — Soit α une racine de $f(X)$ dans $\mathbb{F}_q$, soit $\beta = f_\alpha(\alpha)$, nous avons donc $\alpha = f_\beta(\beta)$. De n'importe quel polynôme bivarié $h(X,Y)$ on peut déduire les polynômes univariés $h_\alpha(X) = h(X, f_\alpha(X))$ et $h_\beta(Y) = h(f_\beta(Y), Y)$ avec $h_\alpha(\alpha) = h_\beta(\beta)$ dans $\mathbb{F}_q$. Par ailleurs les polynômes univariés $h_\alpha(X)$ et $h_\beta(Y)$ peuvent être factorisé en polynômes de bas degré qui, une fois substitué X par α ou Y par β, conduisent à une relation dans $\mathbb{F}_q$ entre polynômes de bas degré en α et β.

En tenant compte du fait que $\deg(h_1) \leqslant d_\alpha \cdot \deg_Y(h) + \deg_X(h)$ et que $\deg(h_2) \leqslant d_\beta \cdot \deg_X(h) + \deg_Y(h)$, nous considérons des polynômes bivariés $h(X,Y) = u(X)Y - v(X)$ où, afin d'équilibrer le degré des polynômes $h_\alpha(X)$ et $h_\beta(Y)$, les polynômes $u(X)$ et $v(X)$ à coefficients dans $\mathbb{F}_p$ sont de degré D. Pour éviter d'avoir des relations redondantes, on choisit de plus des polynômes $u(X)$ unitaires. Notons $\mathcal{S}_D$, l'ensemble des polynômes irréductibles et unitaires de degré au plus D, alors dans le cas favorable où $h_\alpha(X)$ et $h_\beta(Y)$ ont tous leurs facteurs irréductibles dans $\mathcal{S}_D$, on a une relation

$$U_\alpha \prod_{\pi \in \mathcal{S}_D} \pi(\alpha)^{e_{\alpha,i}} = h(\alpha,\beta) = U_\beta \prod_{\pi \in \mathcal{S}_D} \pi(\beta)^{e_{\beta,i}}$$

avec U_α et U_β deux éléments de $\mathbb{F}_p^*$, que l'on peut faire disparaître si on élève les deux côtés de ces relations à la puissance $(p-1)$-ième. Un choix naturel pour la base de lissité est donc $\mathcal{S} = \{\pi(\alpha) \mid \pi(X) \in \mathcal{S}_D\} \cup \{\pi(\beta) \mid \pi(X) \in \mathcal{S}_D\}$.

3.2.1.3. Algèbre linéaire. — Une fois collecté un minimum de $|\mathcal{S}|$ relations, il devient possible d'inverser le système linéaire correspondant modulo $(q-1)/(p-1)$ (*cf.* paragraphe 3.2.3). Au final, on dispose ainsi du logarithme discret des éléments de $\mathcal{S}$, en base $g \in \mathcal{S}$ de notre choix.

Remarque 3.1. — *Lorsque $D = 1$, le système n'est pas de rang plein sur les rationnels puisqu'il y a exactement d_α et d_β inconnues respectivement à gauche et à droite dans nos relations (en comptant les multiplicités). L'espace des solutions contient donc un sous-espace vectoriel de dimension un, ayant pour base le vecteur avec pour composantes d_β et d_α pour respectivement les inconnues de gauches et de droites. Il est facile de lever cette ambiguïté en ajoutant à ces relations une équation d'un autre type, par exemple un polynôme linéaire en α dont l'expression en β se factorise complètement sur $\mathcal{S}$.*

3.2.1.4. Résolution finale. — Étant donné de façon arbitraire un élément y de $\mathbb{F}_q$, l'idée principale, à ce stade, consiste à trouver un entier ν tel que que yg^ν, considéré comme un polynôme de degré au plus $n-1$, se factorise en polynômes de degré au plus $\sqrt{Dn}/2$. La difficulté à laquelle nous sommes maintenant confrontés est d'exprimer ces polynômes comme un produit d'éléments de $\mathcal{S}$.

Soit $r(X)$ l'un de ces polynômes de bas degré, on recherche maintenant avec $u(X)$ et $v(X)$ des polynômes de degré au plus $\sqrt{Dn}/2$ choisis tels que $r(X)$ divise $u(X)f_\alpha(X) - v(X)$, un polynôme bivarié $u(X)Y - v(X)$ tel que $(u(X)f_\alpha(X) - v(X))/r(X)$ et $u(f_\beta(Y))Y - v(f_\beta(Y))$ se factorisent simultanément en facteurs de degré plus petit que celui de $r(X)$. En réitérant la méthode un nombre suffisant de fois (que l'on peut montrer comme étant petit dans l'analyse), on arrive au final à exprimer y comme un produit d'éléments contenus dans $\mathcal{S}$.

3.2.2. Exemple. — On souhaite calculer des logarithmes discrets dans $\mathbb{F}_{17^4}$. Prenons pour cela $D = 1$, $f_\alpha(X) = X^2 + 1$ et $f_\beta(X) = X^2 + X + 1$, si bien que $f(X) =$

$f_\beta(f_\alpha(X)) - X = X^4 + 3\,X^2 + 16\,X + 3$ est irréductible. On note α une racine de $f(X)$ dans $\mathbb{F}_q$, $\beta = f_\alpha(\alpha)$ et donc $\alpha = f_\beta(\beta)$.

On énumère ensuite les 17^3 polynômes de la forme $h(X,Y) = (X+u_0)\cdot Y + (v_1\cdot X + v_0)$ avec $u_0, v_0, v_1 \in \mathbb{F}_p$ et l'on recherche parmi ces derniers ceux tels que $h(X, f_\alpha(X))$ et $h(f_\beta(Y), Y)$ se factorisent simultanément en facteurs de degré un. Une recherche rapide montre qu'il existe 95 telles relations, nous donnons dans Tab. 4 quarante d'entre elles, sous la forme condensée de 6-uplets. À un 6-uplet $(\alpha_0, \alpha_1, \alpha_2, \beta_1, \beta_2, \beta_3)$, correspond l'équation $(\alpha - \alpha_0)\cdot(\alpha - \alpha_1)\cdot(\alpha - \alpha_2) = (\beta - \beta_0)\cdot(\beta - \beta_1)\cdot(\beta - \beta_2)$. Typiquement, de $(4, 13, 14, 0, 5, 11)$, on déduit que $(\alpha - 4)\cdot(\alpha - 13)\cdot(\alpha - 14) = \beta\cdot(\beta - 5)\cdot(\beta - 11)$.

(4,13,14,0,5,11)	(4,9,13,0,6,10)	(4,7,13,0,2,14)	(4,6,13,0,7,9)
(3,4,13,0,1,15)	(6,7,9,4,11,13)	(2,7,15,2,5,14)	(1,9,16,2,6,10)
(10,11,15,0,6,11)	(3,7,14,2,10,14)	(4,5,12,4,9,12)	(0,7,12,0,5,13)
(7,10,14,5,11,16)	(6,13,15,6,8,14)	(2,4,15,4,5,12)	(0,4,16,9,13,16)
(0,9,11,2,7,11)	(1,6,11,0,3,16)	(7,10,13,3,13,16)	(0,8,13,2,5,8)
(6,7,11,2,3,14)	(1,4,6,6,13,15)	(4,8,9,4,12,14)	(1,10,12,1,4,6)
(3,5,16,4,7,16)	(1,2,15,0,5,16)	(1,6,7,8,10,12)	(3,4,14,4,10,12)
(8,10,15,2,10,12)	(8,13,15,9,15,16)	(5,9,11,0,1,4)	(1,3,16,1,2,15)
(1,2,13,4,8,15)	(5,12,14,5,9,11)	(7,15,16,8,13,15)	(7,9,10,6,10,16)
(0,12,13,4,11,16)	(3,8,9,1,14,15)	(2,8,16,1,3,7)	(6,7,8,0,1,6)

Table 4. Relations issues d'un crible pour $\mathbb{F}_{17^4}$

On ajoute à ces relations, l'équation $\alpha - 14 = (\beta - 5)\cdot(\beta - 11)$ de façon à ce que le système linéaire obtenu soit de rang maximal, et l'on trouve le logarithme discret des éléments $\alpha - \alpha_0$ et $\beta - \beta_0$ pour tout $\alpha_0, \beta_0 \in \mathbb{F}_q$ modulo chacun des facteurs de $17^4 - 1 = 2^6 \cdot 3^2 \cdot 5 \cdot 29$.

Typiquement, si on choisit comme base g l'élément primitif $\alpha - 2$, on trouve que le logarithme discret de α (resp. β) en base g est égal, modulo 29, à 28 (resp. 17). Un calcul analogue modulo 2, 3 et 5 permet finalement de déduire que $\alpha = g^{40773}$ et $\beta = g^{39718}$.

3.2.3. Mise en œuvre. — La difficulté essentielle lors de la programmation de cet algorithme sur ordinateurs est la phase d'algèbre linéaire. Pour la réaliser, on combine généralement une élimination gaussienne structurée (introduite par Lamachia et Odlyzko en 1991) pour réduire significativement la dimension du système à résoudre, avant de terminer le calcul avec une méthode itérative comme la méthode de Lanczos ou de Wiedemann.

3.2.3.1. Élimination gaussienne structurée. — L'élimination gaussienne structurée telle que décrite dans [24] consiste à choisir à chaque étape pour pivot dans l'algorithme de Gauss celui qui intervient le moins de fois, afin de densifier au minimum la matrice résultat.

Typiquement, lors d'un pivotage avec une ligne de poids w dans une colonne de poids plus petit que k, la variation estimée sur le poids total est égale à $(k-1)((w-1)-1)-w = (w-2)(k-2)-2$. On choisit donc l'entrée avec la valeur $(w-2)(k-2)$ minimale comme pivot. Après chaque étape, $(w-2)(k-2)$ doit être recalculé, mais grâce à une représentation adaptée de la matrice nous pouvons le faire incrémentalement, donc efficacement.

3.2.3.2. Algorithme de Lanczos. — Multiplier la matrice et sa transposée conduit à une matrice symétrique à partir de laquelle on peut définir un produit scalaire. L'algorithme de Lanczos [24] peut être vu comme un procédé d'orthogonalisation pour ce produit scalaire. Il permet de trouver le vecteur $\mathbf{x}$ solution de l'équation matricielle $A\mathbf{x} = \mathbf{y}$ par projection sur la base orthogonale calculée. Son coût principal est deux fois la dimension de la matrice par le coût de la multiplication de la matrice avec un vecteur dont les éléments sont des nombres entiers de même taille que $(q-1)/(p-1)$.

Malheureusement, cet algorithme se distribue mal, en particulier sur un réseau de stations de travail. Jusqu'à présent, les essais de parallélisation ont porté pour l'essentiel sur le produit matrice-vecteur. Du coup, ils nécessitent l'utilisation de machines massivement parallèles.

3.3. Expérimentations en grandeur nature. — Posons $H(X,Y) = Y^5 + Y + X^2 + 1$, et $a_1(X)Y + a_0(X) = (X^{121} + X^8 + X^7 + X^5 + X^4 + 1)\cdot Y + 1$. Leur résultant en Y, $f(X)$, est un polynôme irréductible de degré 607, dont on note γ une racine dans $\mathbb{F}_{2^{607}}$. On choisit alors comme base de lissité pour le crible,

- 1 000 000 places irréductibles de degré relatif un au-dessus de leurs restrictions à $\mathbb{F}_2(X)$ du côté du polynôme de degré 5,
- 765 925 places irréductibles du côté du polynôme de degré 1.

Le crible a fourni, après 18 jours de calcul sur une machine avec 16 processeurs DEC alpha cadencés à 1.15 GHZ un total de 1 898 338 équations avec 1 556 351 inconnues. Après l'élimination Gaussienne structurée, il reste 365 928 équations avec 364 927 inconnues comprenant 104 312 947 entrées non nulles (environ 286 inconnues par équation).

Appliquer l'algorithme de Lanczos à ce système conduit en dix jours aux logarithmes de petits polynômes irréductibles. Typiquement, on trouve,

$$\begin{aligned}\log_\gamma(\gamma+1) = \; & 10609492128595622222624937026242966957231352347767233020066622 \\ & 23675011082943476527357064661792736790795355204136627165628980 \\ & 66366551334582985153118169319000331715715367479012889579679 53,\end{aligned}$$

ou encore,

$$\log_\gamma(\gamma^2+\gamma+1) = \begin{array}{l}252004893491706977983106205029565158306449725598536526421876277776038590761926757324360558160692922785915155428024088164161518786778594735877418361412794450150811428693269839070316351\end{array}.$$

Pour terminer, on a calculé le logarithme discret de $y(\gamma) = \lfloor 2^{605}\pi \rceil = \gamma^{606}+\gamma^{605}+\ldots+\gamma^4+\gamma^2$. En particulier, on trouve en quelques heures,

$$\begin{aligned} y_1(\gamma) &= \texttt{208d2107247a35}\cdot\texttt{a10da7a6f73e3}\cdot\texttt{e48f497e83}\cdot\texttt{9355bfccf}\cdot \\ &\quad \texttt{3ed885367}\cdot\texttt{2c6d3d}\cdot\texttt{9b7d}\cdot\texttt{7123}\cdot\texttt{5375}\cdot\texttt{47b5}\cdot\texttt{35b9}\cdot\texttt{a29}\cdot\texttt{7}, \\ y_2(\gamma) &= \texttt{46da9a2b29f69}\cdot\texttt{43050a0219}\cdot\texttt{307c17f1b}\cdot\texttt{79cac967b}\cdot\texttt{6a513017}\cdot \\ &\quad \texttt{e4cf755}\cdot\texttt{14bcdbf}\cdot\texttt{53a4a3}\cdot\texttt{2c3999}\cdot\texttt{747}\cdot\texttt{2d1}\cdot\texttt{fd}\cdot\texttt{b}^2\cdot\texttt{3}\cdot 2^4 \end{aligned}$$

tels que $\texttt{155b93b}^{27}\cdot y(\gamma) = y_1(\gamma)/y_2(\gamma)$, où nous notons de façon concise les polynômes sur $\mathbb{F}_2[\gamma]$ à l'aide d'entiers écrits en hexadécimal (par exemple, $\texttt{b}$ représente $\gamma^3+\gamma+1$). À partir de cette relation, on tire,

$$\log_\gamma y(\gamma) = \begin{array}{l}194891399758968486429687093757270550092585346683729020126283742527056226784304085164731800423582020004466178421521409573307372675772079931586693537551194866503272438476718291852872030\end{array}.$$

4. Algorithmes pour les corps de degrés fixés

La méthode de Kraitchik (*cf.* paragraphe 2.2) a d'abord été améliorée en 1986 par Coppersmith *et al.* [**11**] avec un algorithme de complexité $L_q(1/2,1)$. Il faut attendre 1993 et les travaux de Schirokauer [**33**], suite à une proposition de Gordon [**14**], pour disposer pour la première fois d'un algorithme de complexité asymptotique en $L_q(1/3)$, en fait $L_q(1/3,(64/9)^{\frac{1}{3}})$, pour des corps finis de degrés fixés lorsque le cardinal du corps tend vers l'infini.

Nous présentons ici ces deux méthodes, d'abord pour le cas de corps premiers dans le paragraphe 4.1. Nous montrons ensuite dans le paragraphe 4.2 les modifications qu'il faut apporter pour les corps de degrés supérieurs. Nous illustrons l'ensemble avec des éléments relatifs à une expérimentation de grande ampleur au paragraphe 4.3.

4.1. Méthodes pour les corps premiers. — Afin de faciliter la compréhension, nous introduisons d'abord la méthode des entiers de Gauss avec le formalisme qui sera nécessaire à la présentation dans un second temps du crible algébrique.

4.1.1. Méthode des entiers de Gauss. — La méthode dite des entiers de Gauss [**11**] utilise comme corps de nombres $\mathbb{Q}$ et un corps de nombres quadratique imaginaire. On choisit un petit entier b tel que modulo p, $-b$ soit égal au carré d'un élément a.

On écrit par ailleurs $a = a_0/a_1 \bmod p$ avec $a_0, a_1 \simeq O(\sqrt{p})$ et l'on considère les corps de nombres $\mathbb{Q}(\alpha) \simeq \mathbb{Q}$ et $\mathbb{Q}(\beta)$ respectivement définis par

$$f_\alpha(X) = a_1 X - a_0 \text{ et } f_\beta(X) = X^2 + b$$

dont on note les anneaux d'entiers $\mathcal{O}_\alpha \simeq \mathbb{Z}$ et $\mathcal{O}_\beta$.

Il est alors facile de donner dans $\mathbb{Q}(\alpha)$ un sens en termes d'entiers algébriques à une factorisation en idéaux premiers et d'envoyer ces entiers dans $\mathbb{F}_p$ par, $(u+v\alpha) \to u+va$. Pour les idéaux de $\mathcal{O}_\beta$, la situation est à peine plus difficile lorsque le groupe des classes de $\mathcal{O}_\beta$ est de cardinal un puisque alors, pour tout idéal $\mathfrak{p}$ de O_β,

$$\exists (u_\mathfrak{p}, v_\mathfrak{p}) \in \mathbb{Z}^2 \text{ tel que } \mathfrak{p} = (u_\mathfrak{p} + v_\mathfrak{p}\beta)\,.$$

L'application de réduction est simplement donnée par $u_\mathfrak{p} + v_\mathfrak{p}\beta \to u_\mathfrak{p} + v_\mathfrak{p} a$. Grâce à ces définitions, on peut obtenir la base de lissité à partir de

$$\mathcal{S} = \{\text{nombres premiers } p < B_\alpha\} \cup \{\text{générateurs du groupe des unités}\} \cup \{u_\mathfrak{p} + v_\mathfrak{p} a, \mathfrak{p} \text{ idéal premier de } O_\beta, \mathrm{Norm}(\mathfrak{p}) < B_\beta\}\,,$$

où B_α et B_β sont des bornes qui conduisent à des probabilités de lissité suffisamment hautes. L'étape 1 de la méthode générique consiste ici à rechercher des couples (u, v) tels que $(u + v\alpha)$ et $(u + v\beta)$ soient simultanément lisses. Alors

$$U_\alpha \prod_{p \in \mathcal{S}} p^{e_p} = u + va = U_\beta \prod_{\mathfrak{p} \in \mathcal{S}} (u_\mathfrak{p} + v_\mathfrak{p} a)^{e_\mathfrak{p}} \text{ où } U_\alpha \text{ et } U_\beta \text{ sont des unités}\,.$$

Les calculs majeurs réalisés avec ce procédé sont donnés dans Tab. 5.

Taille (chiffres)	Quand	Complexité (GIPS year)	Qui
58	1991	0.01	Lamacchia, Odlyzko
85	1996	0.10	Weber
90	1998	0.07	Joux, Lercier

Table 5. Records avec la méthode des entiers de Gauss pour le problème du logarithme discret dans $\mathbb{F}_p$

4.1.2. Crible algébrique général. — Le crible algébrique est une généralisation de la méthode des entiers de Gauss à des corps de nombres arbitraires $\mathbb{Q}(\alpha)$ ou $\mathbb{Q}(\beta)$. On a toujours, en corollaire du théorème de Dedekind, les factorisations

$$(u + v\alpha) = \prod_{\mathfrak{p} \in \mathcal{S}} \mathfrak{p}^{e_\mathfrak{p}} \text{ et } (u + v\beta) = \prod_{\mathfrak{p} \in \mathcal{S}} \mathfrak{p}^{e_\mathfrak{p}}\,.$$

Le problème est que l'on ne peut plus les réduire aussi facilement dans $\mathbb{F}_p$. Les obstructions proviennent ici du groupe des classes et des unités de ces corps qui, pour certains de ceux qui sont considérés en pratique, ne peuvent pas être calculés. On doit

à Schirokauer une méthode permettant de lever cette difficulté. Les plus gros calculs réalisés à ce jour reposent sur ces idées (*cf.* TAB. 6).

Taille (chiffres)	Quand	Complexité (GIPS year)	Qui
25	1994	0.00	Weber
65	1995	0.01	Weber
85	1998	0.05	Weber
100	1999	0.05	Lercier, Joux
110	2000	0.20	Lercier, Joux
130	2005	1.5	Lercier, Joux
160	2007	55	Franke *et al.*

TABLE 6. Records avec le crible algébrique général pour le problème du logarithme discret dans $\mathbb{F}_p$

Nous passons maintenant en revue les grandes étapes d'un crible algébrique général :

- trouver de bons corps de nombres (*cf.* paragraphe 4.1.2.1),
- cribler efficacement (*cf.* paragraphe 4.1.2.2),
- inverser le système linéaire (*cf.* paragraphe 4.1.2.3),
- résoudre des instances particulières du logarithme discret (*cf.* paragraphe 4.1.2.4).

4.1.2.1. Stratégies de définition des corps de nombres. — Soient $f_\alpha(X)$ et $f_\beta(X)$ les polynômes de définition des entiers algébriques α et β. Ces polynômes doivent satisfaire les conditions suivantes :

- $f_\alpha(X)$ et $f_\beta(X)$ sont irréductibles sur $\mathbb{Z}$,
- les coefficients de $f_\alpha(X)$ et $f_\beta(X)$ sont premiers entre eux,
- $f_\alpha(X) \neq \pm f_\beta(X)$,
- $\exists a \in \mathbb{F}_p^*,\ f_\alpha(a) = f_\beta(a) = 0$.

La lissité des idéaux $(u + v\alpha)$ et $(u + v\beta)$ étant d'autant meilleure que leurs normes sont petites, il est de plus naturel de rechercher des polynômes $f_\alpha(X)$ et $f_\beta(X)$ avec des coefficients aussi petits que possible. Cependant, le fait d'avoir une racine commune a dans $\mathbb{F}_p$ implique que p doit diviser le résultant de $f_\alpha(X)$ et $f_\beta(X)$.

Quatre méthodes sont connues pour trouver des polynômes $f_\alpha(X)$ et $f_\beta(X)$ vérifiant les conditions précédentes.

- La méthode des entiers de Gauss [**11**] : elle conduit à $f_\alpha(X) = a_1X - a_0$ et $f_\beta(X) = X^2 + b$, avec les coefficients a_i de l'ordre de $p^{\frac{1}{2}}$ et $b = O(1)$.
- La méthode de Montgomery [**13**] : elle conduit à $f_\alpha(X) = a_2X^2 + a_1X + a_0$ et $f_\beta(X) = b_2X^2 + b_1X + b_0$, avec les coefficients a_i et b_i de l'ordre de $p^{\frac{1}{4}}$.
- La décomposition en base a [**26**] : elle conduit à $f_\alpha(X) = X - a$ et $f_\beta(X) = \sum_{i=0}^{d} b_iX^i$, avec a et les coefficients b_i de l'ordre de $p^{\frac{1}{d+1}}$ et $\sum_{i=0}^{d} b_ia^i = p$.

– La méthode de Joux-Lercier [17] : on choisit un polynôme $f_\beta(X)$ de degré $d+1$ avec des coefficients aussi petits que possible ayant une racine a dans $\mathbb{F}_p$. Puis, on applique l'algorithme LLL au réseau dont des générateurs sont donnés, en colonne, par la matrice

$$\begin{pmatrix} W & Wa & W(a^2 \bmod p) & \cdots & W(a^d \bmod p) & Wp \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{pmatrix},$$

où W est une constante arbitraire, suffisamment grande. On trouve un vecteur court $(0, a_0, \ldots, a_d)^t$ tel que les coefficients a_i sont de l'ordre de $p^{\frac{1}{d+1}}$. Ainsi, puisque $\sum_{i=0}^{d} a_i a^i = 0 \bmod p$, on peut poser $f_\alpha(X) = \sum_{i=0}^{d} a_i X^i$. Avec ces polynômes, les normes sont plus petites que celles qui sont obtenues avec les méthodes précédentes.

4.1.2.2. Collecte des relations. — Avec un crible, plutôt que de tester successivement la lissité de chacun des idéaux principaux $(u + v\alpha)$ ou $(u + v\beta)$ avec les idéaux de la base de lissité, on procède à l'inverse. On marque dans le tableau d'abscisses u et d'ordonnées v, les idéaux principaux multiples des idéaux premiers de la base de lissité. En fait, puisque les idéaux premiers que nous considérons sont de degré un, il existe une racine ρ de $f_\alpha(X) \bmod \pi$ où $\pi = \mathrm{Norm}(\mathfrak{p})$ et les entiers multiples de $\mathfrak{p}$ appartiennent au réseau $L_\mathfrak{p}$ défini par la matrice $\begin{pmatrix} \rho & \pi \\ 1 & 0 \end{pmatrix}$.

Lorsque l'on crible sur de grands tableaux, ce qui est le cas en pratique, on préfère généralement travailler dans le sous-réseau défini par un idéal $\mathfrak{q}$. Cette technique s'appelle un crible avec « spécial-q ». Précisément, on considère les entiers algébriques $u + v\alpha$ qui sont multiples d'un idéal premier $\mathfrak{q}$ de norme moyenne. De tels entiers algébriques appartiennent à un réseau $L_\mathfrak{q}$. Comme par ailleurs tout entier algébrique multiple d'un idéal premier $\mathfrak{p}$ de la base appartient aussi à un réseau $L_\mathfrak{p}$, on est finalement ramené à énumérer les points du réseau $L_\mathfrak{q} \cap L_\mathfrak{p}$ dont il est aisé de déterminer une base.

Remarque 4.1. — *En pratique, il est parfois préférable d'utiliser comme base de ces réseaux des vecteurs courts pour accélérer le crible.*

4.1.2.3. Algèbre linéaire. — Soient $\mathcal{S}_\alpha$ et $\mathcal{S}_\beta$ deux ensembles d'idéaux premiers de degré un et de petites normes dans $\mathcal{O}_\alpha$ et $\mathcal{O}_\beta$. Supposons que le crible donne $|\mathcal{S}_\alpha| + |\mathcal{S}_\beta| + O(1)$ couples d'équations de la forme

$$(u + v\alpha) = \prod_{\mathfrak{p} \in \mathcal{S}_\alpha} \mathfrak{p}^{e_\mathfrak{p}} \text{ et } (u + v\beta) = \prod_{\mathfrak{p} \in \mathcal{S}_\beta} \mathfrak{p}^{e_\mathfrak{p}}.$$

Grâce alors a une inversion matricielle réalisée modulo $p-1$, nous avons,

$$\forall \mathfrak{p} \in \mathcal{S}_\alpha,\ \mathfrak{p} = I_\mathfrak{p}^{p-1} \prod_{(u,v)} (u+v\alpha)^{e_{u,v}} \text{ et } \forall \mathfrak{p} \in \mathcal{S}_\beta,\ \mathfrak{p} = I_\mathfrak{p}^{p-1} \prod_{(u,v)} (u+v\beta)^{e_{u,v}}\,,$$

et il reste $|\mathcal{S}_\beta| + O(1)$, $|\mathcal{S}_\alpha| + O(1)$ équations supplémentaires de la forme

$$I_\alpha^{p-1} = \prod_{(u,v)} (u+v\alpha)^{e_{u,v}} \text{ et } I_\beta^{p-1} = \prod_{(u,v)} (u+v\beta)^{e_{u,v}}$$

(ou les idéaux I_α, I_β et $I_\mathfrak{p}$ sont non principaux). On peut pour chaque équation, $I_\alpha^{p-1} = \prod_{(u,v)}(u+v\alpha)^{e_{u,v}}$, calculer, avec (r_1, r_2) la signature de $Q(\alpha)$, r_1+r_2-1 quantités $(\lambda_1, \ldots, \lambda_{r_1+r_2-1})$ appelées « maps » de Schirokauer. Nous référons à [33] pour une définition précise de ces « maps », notons simplement qu'elles sont analogues à des logarithmes ℓ-adiques de $(u+v\alpha)$ pour les facteurs premiers ℓ de $p-1$. Grâce à une phase d'algèbre linéaire modulo $(p-1)$, il est alors facile de calculer une combinaison adéquate des dernières r_1+r_2-1 équations avec chacune des premières $|\mathcal{S}_\beta|$ équations pour obtenir $|\mathcal{S}_\beta|$ nouvelles équations telles que leur vecteur de « maps » est nul.

Alors, Schirokauer explique que si la conjecture de Leopoldt est vraie, ces équations ne font intervenir que des entiers algébriques, *i.e.* $\exists \delta \in \mathcal{O}_\alpha$ tel que $\delta^{p-1} = \prod_{(u,v)}(u+v\alpha)^{e_{u,v}}$.

Remarque 4.2. — *Pour faciliter l'exposition, nous avons découpé ici l'algèbre linéaire en quelques sous-étapes. En « grandeur nature », il est bien entendu que nous la réalisons en une seule fois.*

4.1.2.4. Résolution finale. — Il est naturel de chercher à exprimer le logarithme visé en fonction du logarithme de petits nombres premiers. Simplement, on ne peut utiliser que des nombres premiers qui se décomposent complètement dans les corps de nombres $\mathbb{Q}(\alpha)$ ou $\mathbb{Q}(\beta)$. Dans une situation typique, ces corps de nombres sont de degrés deux et trois. Dans un corps quadratique, environ la moitié des nombres premiers se décomposent en idéaux premiers de degré un, et en prenant en compte celui de degré trois, on augmente assez peu les nombres premiers finalement utilisables. Dans cette voie, rappelons qu'une technique classique pour calculer le logarithme discret d'un élément y particulier est de rechercher deux nombres entiers y_1 et y_2 de taille proche de $\sqrt{p}$ tels que $y = y_1/y_2 \bmod p$. On essaie de nombreux tels couples (y_1, y_2) jusqu'à ce que l'un d'eux soit lisse. En fait, on améliore sensiblement l'efficacité de la recherche en utilisant des techniques de crible. Une fois réduit le réseau $\begin{pmatrix} y & p \\ 1 & 0 \end{pmatrix}$, nous avons alors des nombres entiers y_1, y_2, y_1' et y_2' tels que

$$\forall (\kappa, \eta) \in \mathbb{Z}^2,\ y = \frac{y_1}{y_2} = \frac{y_1'}{y_2'} = \frac{\kappa y_1 + \eta y_1'}{\kappa y_2 + \eta y_2'} \bmod p\,,$$

et la méthode du crible par vecteurs s'applique pour trouver des nombres entiers κ et η tels que $\kappa y_1 + \eta y'_1$ et $\kappa y_2 + \eta y'_2$ soient simultanément lisses sur l'ensemble des nombres premiers qui se décomposent dans $\mathbb{Q}(\alpha)$ ou $\mathbb{Q}(\beta)$.

Une amélioration permet de se débarrasser de la contrainte sur les nombres premiers utilisés dans le crible. Elle présuppose que le groupe de Galois du corps de nombres de degré le plus élevé soit cyclique et de cardinal égal à un nombre premier. Pour le cas du degré trois, par exemple, il suffit que le discriminant du polynôme soit un carré. Tout nombre premier non inerte se décompose alors en idéaux premiers de degré 1. Dans ce cas, étant donné un élément y dont on cherche le logarithme discret, on peut essayer de l'écrire sous la forme $y = (u_0 + u_1 a + \cdots + u_d a^d)/(v_0 + v_1 a + \cdots + v_d a^d) \bmod p$, où, en utilisant une réduction de réseau, u_0, u_1, …, u_d et v_0, v_1, …, v_d sont des entiers de taille proche de $O(p^{1/(2d+2)})$ premiers entre eux. Dans ce cas, il n'est pas difficile de montrer que les idéaux principaux $(u_0 + u_1\beta + \cdots + u_d\beta^d)$ et $(v_0 + v_1\beta + \cdots + v_d\beta^d)$ se décomposent en idéaux de degré un dans $\mathbb{Q}(\beta)$. Le bénéfice d'un crible alors réalisé dans $\mathbb{Q}(\beta)$ est que les idéaux premiers facteurs de ces entiers algébriques sont systématiquement dans la base de lissité de $\mathbb{Q}(\beta)$.

D'un point de vue plus algorithmique, notons que l'on peut très bien conserver à l'issue du crible des relations dont les idéaux premiers ne sont pas tous dans la base de lissité de $\mathbb{Q}(\beta)$. Ces derniers, s'ils ne sont pas de normes trop importantes, servent alors de « spécial-q » dans un crible annexe faisant intervenir $\mathbb{Q}(\alpha)$. On peut aussi autoriser des idéaux de grande taille du côté $\mathbb{Q}(\alpha)$, on a ainsi au final un véritable arbre de cribles successifs.

4.2. Extension aux corps non premiers. — Cet algorithme, qui est extrait de [19], prend en entrée un corps fini $\mathbb{F}_q$ et quatre paramètres : $E > n$ le degré du second corps de nombre, t le degré des éléments sur lesquels nous allons cribler, B une borne sur les bases de lissité et S une borne sur l'espace de crible. Nous renvoyons aussi le lecteur au paragraphe 5 pour la détermination d'un optimum pour E, t, B et S quand q est donné.

4.2.1. Définition. — Il se décompose en quatre parties : l'initialisation, le crible, l'algèbre linéaire et la résolution finale. Les phases d'algèbre linéaire et de résolution finale étant identiques à celles du paragraphe précédent, nous passons ces dernières sous silence.

4.2.1.1. Initialisation. — On choisit un polynôme unitaire et irréductible de degré n à coefficients entiers, $f_\alpha(X)$, avec de petits coefficients. En particulier, les racines de f_α dans $\mathbb{F}_q$ forment base polynomiale. Il nous faut alors construire un polynôme f_β, de degré E à coefficients entiers tel que f_α divise f_β modulo p, mais pas sur les rationnels. La première condition assure que f_α et f_β ont n racines communes dans $\mathbb{F}_q$. Dans le cas où $E = n$, choisir pour f_β le polynôme $f_\alpha + p$ convient.

Quand $E > n$, on commence par fixer un polynôme f_0 de degré n, avec de petits coefficients et irréductible sur $\mathbb{F}_p$. On choisit alors une constante W arbitraire et l'on pose $f_\alpha(X) = f_0(X + W)$. Le plus grand coefficient de f_α est de l'ordre de W^n. Par réduction de réseau, on recherche alors un polynôme f_β de degré E et de coefficients plus petits que W^n, tel que f_α divise f_β modulo p. Cela peut-être fait en réduisant le réseau donné par les générateurs (en colonne) suivants :

$$\left(\mathbf{f_\alpha(X)} \quad \mathbf{Xf_\alpha(X)} \quad \mathbf{X^2 f_\alpha(X)} \quad \cdots \quad \mathbf{X^{E-n} f_\alpha(X)} \quad \mathbf{p} \quad \mathbf{pX} \quad \mathbf{pX^2} \quad \cdots \quad \mathbf{pX^E} \right).$$

On peut montrer qu'avec l'algorithme LLL, lorsque $2^{(E+1)/4} p^{n/(E+1)} \leqslant W^n$, on obtient de cette façon un polynôme $f_\beta(X)$ avec des coefficients de l'ordre de $W^n \approx p^{n/(E+1)}$.

4.2.1.2. Crible. — Les paramètres t, S et B déterminent le crible. Nous considérons pour cette phase des $(t+1)$-uplets $(u_0, \cdots, u_t)$ d'entiers premiers entre eux avec $|u_i| \leqslant S$ tels que les normes de $\sum_{i=0}^t u_i \alpha^i$ et $\sum_{i=0}^t u_i \beta^i$ soient B-lisses. Chacune de ces normes est égale au résultant de deux polynômes $\sum_{i=0}^t u_i X^i$ et $f_\alpha(X)$ (resp. $f_\beta(X)$). Il est bien connu que le résultant peut être obtenu comme le déterminant d'une $(n+t) \times (n+t)$ matrice formée de t colonnes contenant les coefficients de f_α et de n colonnes contenant les coefficients de l'entier. En utilisant l'inégalité d'Hadamard, nous pouvons borner la norme par ${n^{t/2} t^{n/2} {B_a}^n {B_f}^t}$, où B_a est une borne supérieure sur les valeurs absolues des u_i et B_f une borne semblable pour les coefficients de f_α (resp. f_β).

Lorsque que la norme de $z_1 = \sum_{i=0}^t u_i \alpha^i$ (resp. $z_2 = \sum_{i=0}^t u_i \beta^i$) est B-lisse, les seuls idéaux principaux dans la factorisation de l'idéal (z_1) (resp. (z_2)) sont au-dessus de nombres premiers $\pi < B$ et l'on obtient une relation. La réduction dans le corps fini de ces relations est tout comme dans le paragraphe 4.1.2.3 rendue possible par l'introduction des « maps » de Schirokauer.

4.2.2. Exemple. — On souhaite ici calculer des logarithmes discrets dans $\mathbb{F}_{1193^2}$. On choisit tout d'abord $f_\alpha(X) = X^2 + 3$ et $f_\beta(X) = X^2 + p + 3$.

On énumère ensuite les couples (u, v) d'entiers premiers entre eux avec $|u| \leqslant 100$ et $0 \leqslant v \leqslant 40$ et l'on ne conserve que les couples (u, v) tels que les idéaux principaux $(u + v\alpha)$ et $(u + v\beta)$ se factorisent simultanément en idéaux premiers de normes bornées par 47 et 97. Une recherche rapide montre qu'il existe 85 tels couples, que nous donnons dans Tab. 7. Par exemple, on peut vérifier à partir de l'entrée $(-94, 37)$ que $\mathrm{Norm}(-94 + 37\alpha) = 7 \cdot 43^2$ et $\mathrm{Norm}(-94 + 37\beta) = 2^4 \cdot 3 \cdot 5 \cdot 19^3$.

Nous sommes ici en présence de deux corps quadratiques imaginaires, les unités de nos deux corps sont donc triviales et il n'est pas nécessaire d'ajouter de « maps » de Schirokauer. On peut donc inverser tel que le système obtenu modulo les facteurs premiers de $(p^2 - 1)/(p - 1)$.

Typiquement, modulo 199, on trouve que la matrice correspondante a bien un noyau de dimension 1, dont il est possible, une fois fixé un élément primitif de $\mathbb{F}_{p^2}$, de déduire des logarithmes discrets. En fait, le groupe de classe du corps défini par α est

(-94,37)	(-89,13)	(-89,17)	(-82,1)	(-79,2)	(-73,1)	(-72,1)	(-69,2)
(-67,9)	(-65,11)	(-58,1)	(-53,1)	(-42,1)	(-37,1)	(-33,14)	(-31,3)
(-28,11)	(-26,3)	(-23,1)	(-23,12)	(-23,13)	(-20,1)	(-17,2)	(-16,1)
(-15,1)	(-13,1)	(-13,3)	(-13,4)	(-13,27)	(-12,1)	(-11,17)	(-10,9)
(-9,7)	(-8,17)	(-7,1)	(-6,1)	(-5,1)	(-5,23)	(-4,3)	(-2,1)
(-1,1)	(-1,2)	(0,1)	(1,1)	(1,2)	(2,1)	(4,3)	(5,1)
(5,23)	(6,1)	(7,1)	(8,17)	(9,7)	(10,9)	(11,17)	(12,1)
(13,1)	(13,3)	(13,4)	(13,27)	(15,1)	(16,1)	(17,2)	(20,1)
(23,1)	(23,12)	(23,13)	(26,3)	(28,11)	(31,3)	(33,14)	(37,1)
(42,1)	(53,1)	(58,1)	(65,11)	(67,9)	(69,2)	(72,1)	(73,1)
(79,2)	(82,1)	(89,13)	(89,17)	(94,37)			

TABLE 7. Relations issues d'un crible pour $\mathbb{F}_{1193^2}$

trivial (alors que celui qui est défini par β contient 8 éléments), ses idéaux premiers sont donc générés par un entier algébrique dont le logarithme discret correspond à une composante d'un vecteur du noyau. Par exemple, si on choisit comme élément primitif $g = (a+11)/2$ où a est l'une des racines de x^2+3 (g provient d'un générateur d'un idéal premier de norme 31), on trouve aisément que $\log_g(2) = 0 \bmod 199$, $\log_g((a+3)/2) = 0 \bmod 199$, $\log_g(-a+2) = 90 \bmod 199$, *etc.*

4.3. Expérimentations en grandeur nature. — Nous décrivons maintenant, brièvement, un calcul réalisé en 2005 pour un nombre premier de 130 chiffres qui utilise le crible algébrique. Soit $p = \lfloor 10^{129}\pi \rfloor + 38914$, *i.e.*

$$\begin{aligned} p \;=\; & 31415926535897932384626433832795028841971693993751058209749445923 \\ & 07816406286208998628034825342117067982148086513282306647093883523\,, \end{aligned}$$

nous cherchons alors à calculer le logarithme discret en base 2 de $y = \lfloor 10^{129}e \rfloor$.

La phase de crible consiste à trouver des couples (u, v) tels que l'idéal principal $(u+v\beta)$ soit de norme lisse dans le corps de nombres défini par $X^3+12\,X^2-13\,X+3$ et tels que $(u\xi + v\alpha)$ soit de norme lisse dans le corps de nombres défini par

$$\begin{aligned} X^2 - 16909365834968487790611823369329079646668885 \cdot X + \\ 2007725147818668820255327215363313111556122 2 \cdot \xi\,, \end{aligned}$$

où $\xi = 8258891862441565856565120580503979696050188$. Ici, le groupe de Galois du polynôme de degré 3 est d'ordre 3. Le corps de nombres correspondant a deux unités fondamentales, $9\,\beta^2 + 117\,\beta - 41$ et $3\beta - 1$, son groupe de classe est d'ordre un.

Le crible donna, après treize jours de calculs sur une machine avec 16 processeurs DEC alpha cadencés à 1.15 GHZ, 1 782 228 équations avec 1 556 351 inconnues. La base de lissité est constituée d'idéaux de normes plus petites que 18 815 207 (1 200 000 de nombres premiers) pour le corps de degré 2 ou plus petites que 5 799 977 (400 000

nombres premiers) pour l'autre corps. Puisque les idéaux premiers sont tous principaux pour le corps de degré trois, on a tenu compte explicitement de ces unités pour normaliser les équations. Pour le corps de degré deux, par contre, il nous a fallu normaliser avec deux « maps » de Schirokauer.

L'application d'une élimination Gaussienne structurée pour réduire notre système à 433 181 équations en 432 172 inconnues avec 35 323 996 entrées non nulles nécessita quinze minutes. La phase critique est l'inversion finale avec l'algorithme de Lanczos. Notre version parallélisée de cet algorithme a pris 8 jours pour 16 processeurs. À l'issue, nous avons le logarithme pour des générateurs d'idéaux de petites normes. En particulier, nous avons des logarithmes pour certains petits nombres premiers. Par exemple,

$$\begin{aligned}\log_2(3) &= 1893502264238571226544216278485664370791846468296985770573613690141150005617588988077853204617146221819982684416360486509655807744\,,\\ \log_2(11) &= 4237102728490621953875844147019736370886375829289178182680995735016364851941957664105425017609514863871805571511626538943085325555\,.\end{aligned}$$

Pour finalement calculer des instances particulières du logarithme discret, on tire avantage du groupe de Galois de $\mathbb{Q}(\beta)$. Précisément, nous avons trouvé en quelques heures deux entiers algébriques y_1 et y_2 tels que, $2^2 y = y_1/y_2 \bmod p$, et tels que, avec γ, la racine commune modulo p des polynômes de définition des deux corps de nombres,

$$\begin{aligned}y_1 &= (-1+\gamma)(-\gamma^2-11\gamma+4)(\gamma^2+7\gamma-1)(38\gamma^2-5\gamma-14)(-58\gamma^2+51\gamma-10)\\ &\quad(-19968234615452\gamma^2-253095266814753\gamma+88773815898184)(-103527432682334747\gamma^2\\ &\quad-1277679011473992359\gamma+909588468749207990)(52211650\gamma^2+648088367\gamma-410299369)\\ &\quad(-10551006859\gamma^2-133733301857\gamma+46901909959)(9\gamma^2+117\gamma-41)^{-12}(3\gamma-1)^{37}\,,\\ y_2 &= (-1+2\gamma)^2(-250393155\gamma^2-3090199419\gamma+2200155182)(-17257853\gamma^2-218912874\gamma+\\ &\quad 74493007)(-6714524974\gamma^2-85147186243\gamma+29313654043)(-110524082322\gamma^2-\\ &\quad 1346736343362\gamma+1196134561937)(-1643288515585\gamma^2-20830007491278\gamma+\\ &\quad 7286141994812)(9\gamma^2+117\gamma-41)^{-17}(3\gamma-1)^{49}\,.\end{aligned}$$

Alors, en utilisant une suite de cribles avec des spécial-q de tailles décroissantes, grâce à une heure de calcul pour chacun, on déduit

$$\begin{aligned}\log_2(y) &= 2113848822378679565759046301222860744437727641443507757730839547200952585495202128754210118376422361373301079194266697766848291 09\,.\end{aligned}$$

En conclusion, le temps dont nous avons eu besoin pour calculer le logarithme discret d'un élément arbitraire modulo un nombre premier de 130 chiffres sur une machine avec 16 processeurs DEC alpha cadencés à 1.15 GHZ est approximativement de 12 heures, une fois le crible (13 jours) et l'algèbre linéaire (8 jours) exécutés.

5. Algorithmes pour les corps de caractéristiques et degrés variables

Nous avons exhibé dans les paragraphes 3 et 4 deux algorithmes pour résoudre le problème du logarithme discret dans un corps fini : un algorithme de type FFS (*cf.* paragraphe 3.2) et un algorithme de type NFS (*cf.* paragraphe 4.2). Ces algorithmes prennent en entrée un corps fini $\mathbb{F}_q$ et des paramètres supplémentaires D et E, t, B, S. Jusqu'à présent, nous avons considéré que soit la caractéristique, soit le degré, sont fixés lorsque q tend vers l'infini. Ici, en suivant [**18, 19, 20**], nous allons plus loin en explicitant selon la taille relative de p et n de quelle façon ces paramètres doivent être définis en fonction de q pour que l'algorithme obtenu soit systématiquement de complexité asymptotique $L_q(1/3)$. Nous améliorons ainsi un vieux résultat de Adleman et DeMarrais qui proposait dans ce cadre des algorithmes de complexités seulement $L_q(1/2)$ [**2**].

En fait, on peut montrer que la complexité de la résolution finale est toujours inférieure aux complexités du crible et de l'algèbre linéaire. En ce qui concerne ces dernières, d'une part, les cribles consistent à tester la lissité de polynômes définis sur des corps finis de degrés bornés ou d'entiers algébriques de normes bornées. Ce test pouvant être réalisé par des algorithmes de complexité négligeable dans ce contexte, le coût du crible est simplement égal au nombre d'éléments testés. D'autre part, la complexité asymptotique de l'algèbre linéaire est donnée par le carré du cardinal de la base de lissité. Enfin, il ne faut pas oublier que nous sommes contraints à avoir autant de relations après le crible que le nombre d'éléments de la base de factorisation.

En se donnant p comme égal à $\exp(\Lambda(q))$ où $\Lambda(q)$ est une fonction positive, et donc, n comme égal à $\ln(q)/\Lambda(q)$, l'analyse met en évidence 5 cas (*cf.* FIG. 1).

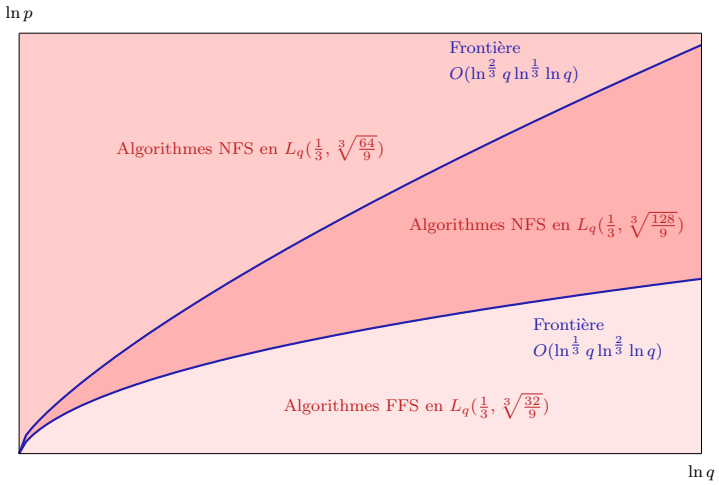


FIGURE 1. Zones de complexités

- Cas du « Function Field Sieve ».
  - Si $\Lambda \in o((\ln q)^{1/3}(\ln\ln q)^{2/3})$, *i.e.* p est asymptotiquement négligeable devant toute fonction de la forme $L_q(1/3,\mu)$, nous avons un algorithme de complexité $L_q(1/3, \sqrt[3]{32/9})$ (*cf.* paragraphe 5.1).
- Cas du « Number Field Sieve ».
  - Si $\Lambda \in \omega((\ln q)^{2/3}(\ln\ln q)^{1/3})$, *i.e.* p est asymptotiquement prépondérant devant toute fonction de la forme $L_q(2/3,\mu)$, nous avons un algorithme de complexité $L_q(1/3, \sqrt[3]{64/9})$ (*cf.* paragraphe 5.2).
  - Si $\Lambda \in o((\ln q)^{2/3}(\ln\ln q)^{1/3}) \setminus O((\ln q)^{1/3}(\ln\ln q)^{2/3})$, *i.e.* p est asymptotiquement négligeable (resp. prépondérant) devant toute fonction de la forme $L_q(2/3,\mu)$ (resp. $L_q(1/3,\mu)$), nous avons un algorithme de complexité $L_q(1/3, \sqrt[3]{128/9})$ (*cf.* paragraphe 5.3).
- Cas aux frontières.
  - Si $\Lambda = \mu(\ln q)^{1/3}(\ln\ln q)^{2/3}$ pour une constante positive μ, *i.e.* p est asymptotiquement équivalent à une fonction de la forme $L_q(1/3,\mu)$, les algorithmes sont de complexité $L_q(1/3,c)$ avec c variant en fonction de μ, au minimum égale à $\sqrt[3]{3}$ (*cf.* paragraphe 5.4).
  - Si $\Lambda = \mu(\ln q)^{2/3}(\ln\ln q)^{1/3}$ pour une constante positive μ, *i.e.* p est asymptotiquement équivalent à une fonction de la forme $L_q(2/3,\mu)$, les algorithmes sont de complexité $L_q(1/3,c)$ avec c variant en fonction de μ, au minimum égale à $\sqrt[3]{64/9}$ (*cf.* paragraphe 5.5).

Il s'avère que les algorithmes obtenus sont aussi très efficaces en pratique. Il suffit pour s'en convaincre de consulter TAB. 8. Nous reprenons d'ailleurs brièvement dans le paragraphe 5.6 le calcul que nous avons mené en 2005 dans $\mathbb{F}_{65537^{25}}$.

Corps	Taille (chiffres)	Quand	Complexité (GIPS year)	Méthode	Qui
$\mathbb{F}_{370801^{18}}$	101	2005	0.4	TORI	Lercier, Vercauteren
$\mathbb{F}_{65537^{25}}$	121	2005	$\simeq 0$	FFS	Joux, Lercier
$\mathbb{F}_{370801^{30}}$	168	2005	0.1	FFS	Joux, Lercier
$\mathbb{F}_{p^3}$	120	2006	1.2	NFS	Joux, Lercier, Smart,Vercauteren

TABLE 8. Records pour le problème du logarithme discret dans $\mathbb{F}_{p^n}$

5.1. Corps de petites caractéristiques. — En toute généralité, l'algorithme FFS du paragraphe 3.2 considère des bases de lissité d'au plus $2p^D$ polynômes irréductibles (en α et β). On sait, par ailleurs, que la probabilité qu'un polynôme de degré $\sqrt{nD}$ ait tous ces facteurs de degré au plus D, est égale à

$$(n/D)^{-\frac{1}{2}\sqrt{n/D}} \cdot (n/D)^{-\frac{1}{2}\sqrt{n/D}}.$$

Le nombre attendu de relations est donc au total de l'ordre de $p^{2D+1}(n/D)^{-\sqrt{n/D}}$, qui doit être supérieur à $2p^D$. Chacune de ces relations ayant de l'ordre de $\sqrt{Dn}$ termes, on arrive à une complexité asymptotique pour l'algèbre linéaire égale à $O(p^{2D}\sqrt{Dn})$.

Quand $\Lambda \in o((\ln q)^{\frac{1}{3}}(\ln\ln q)^{\frac{2}{3}})$, un bon choix pour D est $\delta\,(\ln q)^{\frac{1}{3}}(\ln\ln q)^{\frac{2}{3}}/\Lambda(q)\,$, où δ est une constante à déterminer, la plus petite possible. Avoir suffisamment de relations impose que $D^{3/2}\ln p \leqslant \sqrt{n}\ln n$, ou de façon équivalente que $\Lambda(q) \leqslant \ln^{1-\delta^{3/2}}(q)$. Choisir $\delta = \sqrt[3]{4/9}$ convient, d'où on déduit une complexité asymptotique $L_q(1/3, 2\delta) = L_q(1/3, \sqrt[3]{32/9})$.

5.2. Corps de grandes caractéristiques. — Nous considérons ici l'algorithme du paragraphe 4.2 avec $t = 1$ et $S = B$. Les paramètres sont donc le degré E et la borne de lissité B. Dans cet algorithme, les coefficients de f_α et f_β sont de l'ordre de $R = p^{n/(E+1)} = q^{1/(E+1)}$.

Choisissons donc $E = \varepsilon\,(\ln q/\ln\ln q)^{1/3}$ et $B = L_q(1/3,\theta)\,$, avec ε et θ, deux constantes à déterminer de façon à ce que la complexité finale soit la plus petite possible.

Asymptotiquement, bornons les normes du côté de f_α par RB^n et celles du côté de f_β par RB^E, le produit des normes est alors borné par $B^{n+E}q^{2/(E+1)}$. Avec notre hypothèse sur Λ, le degré n est égal à $\ln q/\Lambda(q) \in o((\ln q)^{1/3}/(\ln\ln q)^{1/3})$, il est donc négligeable devant E et le produit des normes est équivalent à $L_q(2/3, \theta\varepsilon + 2/\varepsilon)$. Il est minimal pour $\varepsilon = \sqrt{2/\theta}$ et égal à $L_q(2/3, 2\sqrt{2\theta})$ dans ce cas.

La probabilité de lissité étant de $L_q(1/3, -(1/3)\cdot 2\sqrt{2/\theta})$ et le nombre de relations collectées lors du crible devant être plus grand que le cardinal de la base de lissité, on a $2\theta - (1/3)\cdot 2\sqrt{2/\theta} \geqslant \theta$. On en déduit que θ doit être au minimum égal à $\sqrt[3]{8/9}$, ce qui conduit à la complexité asymptotique attendue de $L_q(1/3, \sqrt[3]{64/9})$.

5.3. Corps de caractéristiques et degrés moyens. — Nous considérons ici aussi l'algorithme du paragraphe 4.2. Mais p étant plus petit, relativement à n, l'espace de crible avec un choix de paramètre identique à celui du paragraphe 5.2 est insuffisant et nous manquons de relations à la fin du crible. Pour éviter ce problème, nous allons faire croître t vers l'infini avec q pour augmenter l'espace de crible et choisir $E = n$ de façon à ce que les normes pendant le crible soient les plus petites possible. Les paramètres sont donc la borne B sur les éléments de la base de lissité, le degré t des entiers sur lesquels nous criblons et enfin la borne S sur les coefficients de ces entiers.

Le crible consiste à tester la lissité de S^{2t+1} entiers, la probabilité desquels d'être B-lisse étant asymptotiquement égale à $n^{t/2}t^{n/2}S^n$ et $n^{t/2}t^{n/2}S^n p^t$. Nous choisissons pour B, le plus proche entier de $L_q(1/3,\theta)$, pour t, le plus proche entier de $\tau\,(\ln q)^{2/3}(\ln\ln q)^{1/3}/\Lambda(q)\,$, et enfin pour S, le plus proche entier de $\exp(\sigma\Lambda(q)(\ln\ln q)^{1/3}/\ln q^{1/3})$, avec θ, τ et σ des constantes à déterminer de façon à ce que la complexité finale soit minimale.

Avec un tel choix, le produit des normes est asymptotiquement borné par $S^{2n}p^t$, *i.e.* $L_q(2/3, 2\sigma+\tau)$. La probabilité de lissité est alors $L_q(1/3, -(1/3)\cdot(2\sigma/\theta+\tau/\theta)$, et son produit par l'espace de crible S^t, doit être au moins égal à B. Enfin, nous devons égaliser la complexité du crible, *i.e.* S^t, et celle de l'algèbre linéaire, *i.e.* B^2.

La traduction en équations de ces considérations conduit à $\theta = (2\sigma+\tau)/3\theta = \sigma\tau/2$. Une fois τ éliminé, on trouve $\theta = (1+\sqrt{1+6\,\sigma^3})/3\sigma$, dont un minimum est obtenu pour $\sigma = \sqrt[3]{36/27}$. Il suit $\theta = \sqrt[3]{16/9}$, et $\tau = \sqrt[3]{32/3}$, dont on déduit que la complexité de l'algorithme est égale à $L_q(1/3, 2\theta) = L_q(1/3, \sqrt[3]{128/9})$.

5.4. Corps à la frontière $p = L_q(1/3, \mu)$. — À cette frontière, il nous faut examiner à la fois la complexité de l'algorithme FFS du paragraphe 5.1 et celle de l'algorithme NFS du paragraphe 5.3.

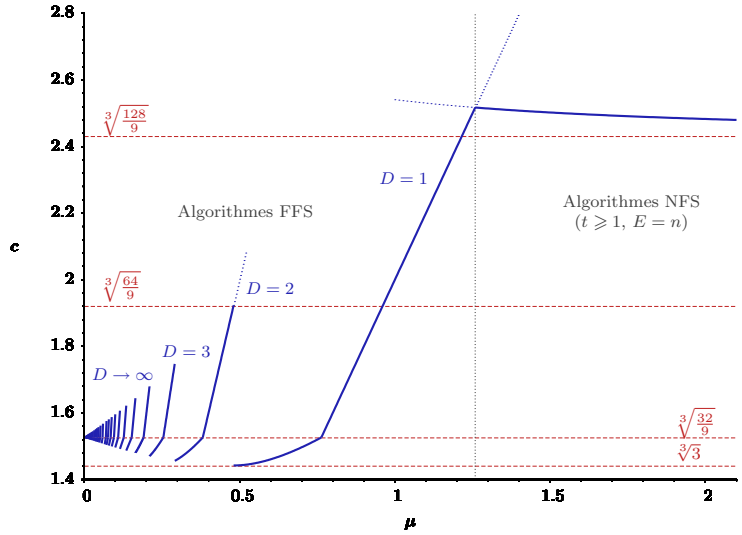


FIGURE 2. Complexités $L_q(1/3, c)$ à la frontière $p = L_q(1/3, \mu)$

En ce qui concerne FFS, excepté que l'on considère maintenant que E est une constante positive fixe, et que donc on a une famille d'algorithmes paramétrée par E, l'analyse est proche de celle du paragraphe 5.1. Elle est résumée sur la partie gauche de FIG. 2. On peut y noter que la plus petite complexité $L_q(1/3, \sqrt[3]{3})$, est réalisée pour $E = 1$ et $\mu = 3^{-\frac{2}{3}}$.

En ce qui concerne NFS, avec un choix pour B identique à celui du paragraphe 5.3, et un choix pour t et S égal à $t \simeq (\tau/\mu)\,(\ln q/\ln\ln q)^{1/3}$ et $S \simeq (\ln q)^{\sigma\tau}$, on arrive à une complexité qu'il est difficile d'écrire ici en toute généralité de façon concise, mais qui comme le suggère FIG. 2, tend comme attendu vers $L_q(1/3, \sqrt[3]{128/9})$ quand μ est très grand.

5.5. Corps à la frontière $p = L_q(2/3, \mu)$. — À cette seconde frontière, il nous faut cette fois examiner la complexité de l'algorithme NFS du paragraphe 5.3 (avec $E = n$ et $t \geqslant 1$) et celle de l'algorithme NFS du paragraphe 5.2 (avec $E \geqslant n$ et $t = 1$)

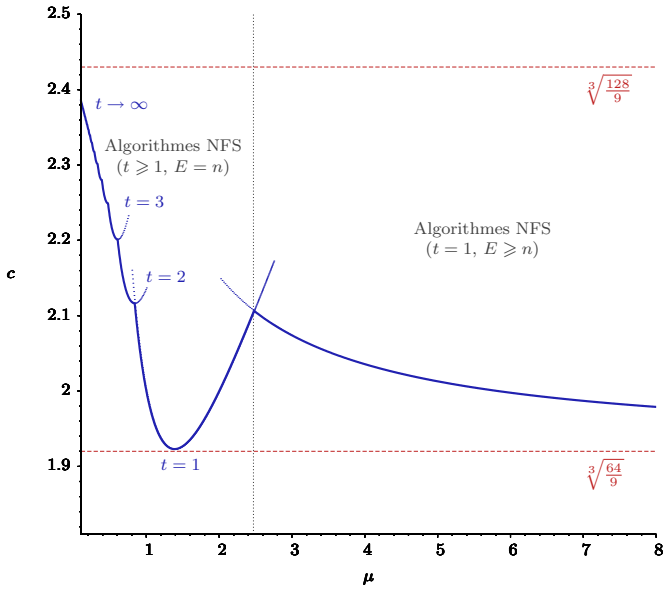


FIGURE 3. Complexités $L_q(1/3, c)$ à la frontière $p = L_q(2/3, \mu)$

Dans le premier cas, on a à considérer une famille d'algorithmes paramétrés par une constante t. Si on choisit alors $B = L_q(1/3, \theta)$ et $S = L_q(1/3, 2\theta/(t+1))$, on retrouve une complexité de $L_q(1/3, (64/9)^{1/3})$ quand $t = 1$ et la limite $L_q(1/3, \sqrt[3]{128/9})$ quand $t \to \infty$. La succession de ces complexités, tronquées à leurs parties utiles, est tracée sur la partie gauche de FIG. 3.

Dans le second cas, on conserve dans la nouvelle analyse les paramètres E et B inchangés, *i.e.* $E = \varepsilon \sqrt[3]{\ln q / \ln \ln q}$ et $B = L_q(1/3, \theta)$. On obtient alors une complexité qui correspond à la partie droite de FIG. 3 et qui, quand μ est très grand, tend vers $L_q(1/3, \sqrt[3]{64/9})$.

5.6. Expérimentations en grandeur nature. — Nous nous intéressons ici au corps fini $\mathbb{F}_{65537^{25}}$, qui nous semble typique du point de faible complexité $L_q(1/3, \sqrt[3]{3})$ déterminé dans l'analyse du paragraphe 5.4. Le cardinal de ce corps a environ 120 chiffres décimaux, sa factorisation est $q = 65536 \cdot 3571 \cdot 37693451 \cdot 137055701 \cdot 10853705894563968937051 \cdot p_{247}$.

On choisit ici pour f_α et f_β, les polynômes $f_\alpha(X) = X^5 + X + 3$ et $f_\beta(X) = -X^5 - X - 1$, si bien que $f(X) = X - f_\beta(f_\alpha(X))$ est un polynôme irréductible de degré 25 qui nous permet de représenter polynomialement $\mathbb{F}_{65537^{25}}$. Soit α, l'une de ses racines, on pose $\beta = f_\alpha(\alpha)$.

Dans le crible, nous avons utilisé comme espace de recherche des polynômes bivariés de la forme $y-(u\,X+v)$, avec u et v dans $\mathbb{F}_{65537}$, que l'on cherche à écrire en α et en β, comme un produit de polynômes de degré $D=1$. On trouve par exemple

$$(\alpha+2445)\cdot(\alpha+9593)\cdot(\alpha+31166)\cdot(\alpha+39260)\cdot(\alpha+48610) \quad = \\ -2(\beta+43449)\cdot(\beta+18727)\cdot(\beta+17129)\cdot(\beta+1946)\cdot(\beta+49823)\,.$$

La phase de crible est en fait très rapide, environ deux minutes sur un ordinateur portable PENTIUM cadencé à 1.6 GHZ.

Après une élimination gaussienne structurée, nous avons à résoudre un système linéaire creux de 79 466 équations en 78 465 inconnues avec un petit peu moins de 4 millions d'entrées. On a pu inverser ce système modulo $\ell = q/(65536\cdot 3571)$ en un peu moins de deux jours sur le même ordinateur. Nous avons ainsi le logarithme discret d'éléments de la forme $\alpha - a_0$ and $\beta - b_0$ modulo ℓ. Par exemple

$$\log_\alpha(\alpha+1) = 95805410880093234842298898214533393829434304594545362348240373548352401735322970633432318492972385332094443948 5 \bmod \ell\,,$$

$$\log_\alpha(\beta) = 46495712756925209185601240503381083970050573012881700517185566862384316422897306135296316764963935552585468876 91 \bmod \ell\,.$$

La dernière phase consiste à choisir un élément arbitraire y de $\mathbb{F}_{65537^{25}}$,

$$y \quad = \quad \sum_{i=0}^{24} (\lfloor \pi\cdot 65537^{i+1}\rfloor \bmod 65537)\,\alpha^i = 41667\alpha^{24}+\cdots+9279\,,$$

et de calculer son logarithme discret. On exprime d'abord y en utilisant des polynômes de degré au plus 3, puis on complète ce calcul par des cribles secondaires pour exprimer ces polynômes de degré 3 comme produits d'éléments de la base. Au final, on trouve en base $g=3\alpha$ (α n'est pas primitif), en moins d'une heure,

$$\log_g(y) = 40537369450524407445879885072715457733779105170746399357547363481852609028577772820085371649268383536448936947412841469 99\,.$$

6. Conclusion

Il apparaît finalement que le problème du logarithme discret est de difficulté plutôt uniforme, puisque dans tous les cas nous disposons d'algorithmes de complexité en $L_q(1/3)$ pour le résoudre, même si en raffinant on s'aperçoit que certains corps sont plus aisés que d'autre. On exhibe en effet un algorithme de complexité $L_q(1/3, \sqrt[3]{3})$ lorsque $\ln p = \sqrt[3]{1/9}\ln^{\frac{1}{3}} q \ln^{\frac{2}{3}}\ln q$.

Ces résultats permettent bien sûr de choisir avec circonspection le dimensionnement cryptographique nécessaire pour atteindre un niveau de sécurité suffisant. Ainsi, il semble que certaines parmi les propositions publiées dans [7, 31, 8] doivent être reconsidérées à la lumière de nos algorithmes.

Mais plus largement, on constate parfois dans d'autres domaines que la complexité d'algorithmes prenants comme entrée un corps fini $\mathbb{F}_q$ n'est donnée que pour des corps de degrés ou caractéristiques fixés. Citons par exemple, la multiplication, l'irréductibilité ou la factorisation de polynômes, le comptage de points sur courbes algébriques, *etc.* Il nous paraîtrait intéressant, dans le même esprit, d'élargir leur étude à des corps finis sans limitation sur la caractéristique ou sur le degré.

Références

[1] L. Adleman – « The function field sieve », in *Proceedings of ANTS-I*, Lecture Notes in Computer Science, vol. 877, 1994, p. 108–121.

[2] L. Adleman & J. DeMarrais – « A subexponential algorithm for discrete logarithms over all finite fields », in *Advances in Cryptology – CRYPTO '93* (D. Stinson, éd.), Lecture Notes in Computer Science, vol. 773, Springer, 1993, p. 147–158.

[3] L. Adleman & M. Huang – « Function field sieve method for discrete logarithms over finite fields », in *Information and Computation*, vol. 151, Academic Press, 1999, p. 5–16.

[4] L. Adleman, R. Rivest & A. Shamir – « A method for obtaining digital signatures and public-key cryptosystems », *Communications of the ACM* **21** (1978), no. 2, p. 120–126.

[5] M. Bellare & P. Rogaway – « Optimal asymmetric encryption », in *Advances in Cryptology – EUROCRYPT '94* (A. DeSantis, éd.), Lecture Notes in Computer Science, vol. 950, Springer, 1994, p. 92–111.

[6] D. Boneh & M. Franklin – « Identity based encryption from the Weil pairing », in *Advances in Cryptology – CRYPTO 2001* (J. Killian, éd.), Lecture Notes in Computer Science, vol. 2139, Springer, 2001, p. 213–229.

[7] D. Boneh, B. Lynn & H. Shacham – « Short signatures from the Weil pairing », in *Advances in Cryptology – ASIACRYPT 2001* (C. Boyd, éd.), Lecture Notes in Computer Science, vol. 2248, Springer, 2001, p. 514–532.

[8] ———, « Short signatures from the Weil pairing », *Journal of Cryptology* **17** (2004), no. 4, p. 297–319.

[9] E. Canfield, P. Erdös & C. Pomerance – « On a problem of Oppenheim concerning "factorisatio numerorum" », *Journal of Number Theory* **17** (1983), p. 1–28.

[10] D. Coppersmith – « Fast evaluation of logarithms in fields of characteristic two », *IEEE Transactions on Information Theory* **30** (1984), no. 4, p. 587–594.

[11] D. Coppersmith, A. Odlyzko & R. Schroppel – « Discrete logarithms in $\mathbb{F}_p$ », *Algorithmica* **1** (1986), p. 1–15.

[12] W. Diffie & M. Hellman – « New directions in cryptography », *IEEE Transactions on Information Theory* **22** (1976), no. 6, p. 644–654.

[13] R. Elkenbracht-Huizing – « An implementation of the number field sieve », Tech. Report NM-R9511, CWI, 1995.

[14] D. Gordon – « Discrete logarithms in $\mathbb{F}_p$ using the number field sieve », *SIAM Journal on Discrete Mathematics* **6** (1993), p. 124–138.

[15] A. Joux – « A one round protocol for tripartite Diffie-Hellman », in *Proceedings of ANTS-IV*, Lecture Notes in Computer Science, vol. 1838, 2000, p. 385–394.

[16] A. Joux & R. Lercier – « The function field sieve is quite special », in *Proceedings of ANTS-V* (C. Fieker & D. Kohel, éds.), Lecture Notes in Computer Science, vol. 2369, Springer, 2002, p. 431–445.

[17] ———, « Improvements to the general number field sieve for discrete logarithms in prime fields. A comparison with the Gaussian integer method », *Mathematics of Computation* **72** (2003), no. 242, p. 953–967.

[18] ———, « The function field sieve in the medium prime case », in *Advances in Cryptology – EUROCRYPT 2006* (S. Vaudenay, éd.), Lecture Notes in Computer Science, vol. 4004, Springer, 2006, p. 254–270.

[19] A. Joux, R. Lercier, N. Smart & F. Vercauteren – « The number field sieve in the medium prime case », in *Advances in Cryptology – CRYPTO 2006* (C. Dwork, éd.), Lecture Notes in Computer Science, vol. 4117, Springer, 2006, p. 326–344.

[20] ———, « An $L_{q^n}(1/3)$-algorithm to compute discrete logarithms over all finite fields $\mathbb{F}_{q^n}$ », 2007, In Preparation.

[21] D. Knuth – *The art of computer programming*, Addison-Wesley, 1973.

[22] M. Kraitchik – *Théorie des nombres*, Paris, Gauthier-Villars, 1922.

[23] ———, *Recherches sur la théorie des nombres*, Paris, Gauthier-Villars, 1924.

[24] B. LaMacchia & A. Odlyzko – « Solving large sparse linear systems over finite fields », in *Advances in Cryptology – CRYPTO '90* (A. Menezes & S. Vanstone, éds.), Lecture Notes in Computer Science, vol. 537, Springer, 1990, p. 109–133.

[25] M. Lennon & P. Smith – « LUC : A new public key system », in *IFIP TC11 Ninth International Conference on Information security IFIP/Sec*, 1993, p. 103–117.

[26] A. Lenstra & H. Lenstra (éds.) – *The development of the number field sieve*, Lecture Notes in Mathematics, vol. 1554, Springer, 1993.

[27] A. Lenstra & E. Verheul – « The XTR public key system », in *Advances in Cryptology – CRYPTO 2000* (M. Bellare, éd.), Lecture Notes in Computer Science, vol. 1880, Springer, 2000, p. 1–19.

[28] ———, « Selecting cryptographic key sizes », *Journal of Cryptology* **14** (2001), p. 255–293.

[29] K. McCurley – « The discrete logarithm problem », in *Proc. Symp. in Applied Mathematics*, Cryptology and Computational Number Theory, no. 42, 1990, p. 49–74.

[30] V. Miller – « Use of elliptic curves in cryptography », in *Advances in Cryptology – Crypto '85* (H. Williams, éd.), Lecture Notes in Computer Science, vol. 218, Springer, 1986, p. 417–426.

[31] K. Rubin & A. Silverberg – « Supersingular abelian varieties in cryptology », in *Advances in Cryptology – CRYPTO 2002* (M. Yung, éd.), Lecture Notes in Computer Science, vol. 2442, Springer, 2002, p. 336–353.

[32] ———, « Torus-based cryptography », in *Advances in Cryptology – CRYPTO 2003* (D. Boneh, éd.), Lecture Notes in Computer Science, vol. 2442, Springer, 2003, p. 349–365.

[33] O. Schirokauer – « Discrete logarithms and local units », *Philosophical Transactions of the Royal Society A* **345** (1993), p. 409–423.

[34] V. Shoup – « Lower bounds for discrete logarithms and related problems », in *Advances in Cryptology – EUROCRYPT '97* (W. Fumy, éd.), Lecture Notes in Computer Science, vol. 1233, Springer, 1997, p. 256–266.

Antoine Joux, DGA & Université de Versailles St-Quentin-en-Yvelines, PRISM, 45, avenue des États-Unis, 78035 Versailles Cedex, France

Reynald Lercier, DGA/CELAR, La Roche Marguerite, 35174 Bruz Cedex, France