UPS – Centre d'Orsay
Mathématiques

MAO – Maîtrise
Année 2003–2004

MAPLE/ 2

Primalité et Cryptologie

Problème I (Pseudo-primalité, Miller-Rabin)

Soit N un entier impair dont on se demande s'il est premier. On pose

$$e := v_2(N-1), \quad q := (N-1)/2^e.$$

Soit a un entier dans $]1, N[$. On dit que N est *pseudo-premier* pour la base a si

$$a^q \equiv 1 \pmod N$$

ou s'il existe $0 \leqslant i < e$ tel que

$$a^{q2^i} \equiv -1 \pmod N.$$

Si aucune de ces conditions n'est remplie, alors N est composé, et a est appelé *témoin* (de non-primalité).

Note : Ce test, proposé par Miller (1976) puis Rabin (1977) est une amélioration du test de Fermat qui cherche a tel qu'on ait $a^{N-1} \not\equiv 1 \pmod N$ [*montrer qu'un tel a est un témoin*].

1) Combien de multiplications dans $\mathbb{Z}/N\mathbb{Z}$ sont elles nécessaire pour déterminer si N est pseudo-premier pour a ? [*les calculs sont faits dans $\mathbb{Z}/N\mathbb{Z}$ avec l'algorithme d'exponentiation binaire. Poser $b := a^q$*]

2) Montrer que, si N est premier, alors il est pseudo-premier pour toute base $a \in]1, N[$.

3) On montre que si N n'est pas premier, il est pseudo-premier pour au plus $1/4$ des bases a. En déduire un test probabiliste fournissant, soit une preuve de non-primalité pour N, soit une forte présomption de primalité (mais pas de preuve).

★★ **Note :** La démonstration n'est pas simple, mais est un bon exercice d'arithmétique : en travaillant dans le 2-Sylow de $(\mathbb{Z}/N\mathbb{Z})^*$, montrer que si n a r diviseurs premiers distincts, alors le nombre de faux témoins (bases fournissant un alibi au non-premier N, pour lesquelles il est pseudo-premier) est inférieur a $1+(2^{rt}-1)/(2^r-1)$, où $t = \min_{p|N}(v_2(p-1))$ [*utiliser le Lemme Chinois et la cyclicité de $(\mathbb{Z}/p^k\mathbb{Z})^*$*]. La proportion est maximale si tous les $v_2(p-1)$ sont égaux à 1 et est majorée par $1/2^{r-1}$; cela règle le cas $r \geqslant 3$. Le cas $r = 1$ ($N = p^k$, $k > 1$) est direct et facile. Le cas $r = 2$ est plus astucieux [*moins de la moitié des b atterrissent dans le 2-Sylow, et au moins la moitié de ceux qui y parviennent sont éliminés par le raisonnement précédent avec $r = 2$*].

★ **4)** On dit que $N > 1$ est un nombre de Carmichael (≈ 1910) si il n'est pas premier, mais qu'il passe le test de Fermat $a^{N-1} \equiv 1 \pmod N$ pour tout $(a, N) = 1$.

a) Montrer le critère de Korselt (1899) : N non premier est de Carmichael ssi il est sans facteur carré et $p - 1 \mid N - 1$ pour tout $p \mid N$, p premier.

b) Trouver tous les nombres de Carmichael inférieurs à 1000. Evaluer la proportion de témoins pour les plus petits d'entre eux.

Note : Les nombres de Carmichael, pour lesquels le test de Fermat est sans intérêt, sont assez rares, mais il en existe une infinité (Alford, Granville et Pomerance, 1994). La question 3) prouve que le test de Miller-Rabin ne souffre d'aucun problème analogue.

5) Indiquer une procédure permettant de trouver un nombre (probablement) premier de n chiffres, avec par exemple $n = 200$ [*pour l'analyse, admettre le théorème des nombres premiers :* $\pi(x) := \#\{p \leqslant x\} \sim x/\log(x)$*, qu'on peut interpréter naïvement par « un entier proche de* x *est premier avec probabilité* $1/\log(x)$ »]

Note : Soit $w(N)$ le plus petit témoin pour N. En développant l'idée de Miller, Bach (1990) a montré que si l'Hypothèse de Riemann Généralisée (GRH) est vraie, alors $w(N) \leqslant 2\log^2(N)$, ce qui rend le test déterministe et polynomial en le nombre de chiffres de N. Sans hypothèse, Burthe (1997) montre qu'il existe un témoin $\leqslant N^{1/(6\sqrt{e})+\varepsilon}$; il prouve aussi que $w(N)$ vaut 2 en moyenne sur les entiers impairs, c'est-à-dire que

$$\sum_{N<X, 2\nmid N} w(N) \sim 2 \sum_{N<X, 2\nmid N} 1 \sim X.$$

Par contre, Alford, Granville et Pomerance (1994) ont montré que $\overline{\lim}_{N\to\infty} w(N) = +\infty$.

Problème II (Preuve de primalité, Pocklington–Lehmer)

Le test de Miller–Rabin ne peut en général pas fournir une preuve de primalité [*ceci dit, Jaeschke (1993) a vérifié que si* $1 < n < 341550071728321 \approx 3.10^{14}$ *est pseudo-premier pour les bases* $(2, 3, 5, 7, 11, 13, 17)$*, alors* n *est premier*]. On étudie ici une méthode simpliste qui permet souvent de conclure, à condition de savoir factoriser $N - 1$.

1) Montrer que N est premier ssi $\varphi(N) = N - 1$.

★ **2)** [Pocklington 1912] On suppose connue la factorisation de $N - 1 = \prod p^{e_p}$. Montrer que si $a^{N-1} \equiv 1 \pmod N$ et $a^{(N-1)/p} \not\equiv 1 \pmod N$ alors p^{e_p} divise l'ordre de a dans $(\mathbb{Z}/N\mathbb{Z})^*$. En déduire que si on peut trouver un tel $a(p)$ pour tout $p \mid N - 1$, alors N est premier [*quel est l'ordre du produit des* $a(p)$ *dans* $(\mathbb{Z}/N)^*$ *?*].

3) Programmer cette méthode [*chercher les racines primitives locales* $a(p)$ *naïvement, comme les racines primitives du TD précédent*].

Note : Des techniques nettement plus sophistiquées, à base de courbes elliptiques (ECPP) ou de sommes de Jacobi (APRCL) (voir [1, Chap. 9]), permettent de prouver la primalité de nombres généraux de plus de 7000 chiffres en quelques mois[1]. La factorisation de $N - 1$ pour de tels nombres est pour le moment parfaitement utopique.

Note : Agrawal, Kayal et Saxena (2002) ont trouvé un algorithme permettant de prouver la primalité de N en temps polynomial [*en* $O(\log^{12+\varepsilon} N)$], inconditionnellement. Leur test, malheureusement pour l'instant essentiellement impraticable, repose sur une façon compliquée de vérifier le théorème (simple) : Si $a \in \mathbb{N}$, $(a, N) = 1$, alors N est premier ssi

$$(x - a)^N \equiv x^N - a \pmod{N\mathbb{Z}[x]}.$$

Problème III (Factorisation et RSA)

Le système RSA (Rivest, Shamir, Adleman 1977) est un des cryptosystèmes à clé publique les plus utilisés, essentiellement pour transmettre des clés symétriques destinées à d'autres protocoles comme DES[2], pour des applications au commerce électronique

[1] Record actuel (07/2003) : 7127 chiffres en 625 jours sur une machine à 1GHz. Le calcul est massivement parallèle : distribué sur 6 machines biprocesseurs à 2.66GHz, il a en réalité pris moins d'un mois.

[2] Data Encryption Standard, cassé en 1998 à cause de la taille insuffisante de son espace de clés. Remplacé par 3DES, puis par AES (Advanced Encryption Standard). Tous ces protocoles utilisent une clé secrète unique, qui sert à chiffrer ou à déchiffrer, d'où le qualificatif symétrique.

(signature) ou dans des logiciels comme PGP ou SSH[3]. Il repose sur le fait qu'on ne connaît pas d'algorithme efficace (polynomial) pour factoriser un entier général $N \in \mathbb{Z}$. Rappelons en rapidement le principe :

Soit $N = pq$ le produit de deux (grands) nombres premiers distincts, de tailles comparables. Soit c (pour *chiffrer*, c'est la clé publique) et d (pour déchiffrer, c'est la clé privée) deux entiers tels que $cd \equiv 1 \pmod{\varphi(N)}$. Soit X un message, vu comme entier $< N$, premier à N. La clé publique c, que tout un chacun peut utiliser, permet de calculer le message chiffré $C := X^c \pmod N$. Pour déchiffrer C, on utilise l'égalité $C^d \equiv X \pmod N$ [*pourquoi ?*] qui, combinée à l'information $0 \leqslant X < N$, détermine X.

La force du système vient de ce que, si p et q sont inconnus, on ne sait pas calculer $\varphi(N) = (p-1)(q-1)$, ni donc d à partir de c. Essentiellement[4], casser RSA revient à factoriser N, ce qui explique l'importance récente prise par les recherches sur la factorisation[5]. L'avantage sur un système symétrique où chiffrer et déchiffrer sont des opérations équivalentes, est qu'arrêter l'espion (découvrir c), n'aide en rien à déchiffrer les messages qu'il a envoyés. La clé publique, destinée au chiffrage des informations, peut-être disséminée arbitrairement, seule la clé secrète doit être protégée.

Signature électronique : tout possesseur de la clé de déchiffrement d peut aussi certifier la provenance du message en envoyant $C := X^d \pmod N$, pour un texte quelconque X. Le destinataire calcule $C^c \pmod N$; s'il obtient un message intelligible, il a la preuve que l'expéditeur connaissait bien la clé secrète, sans que celle-ci soit mise en danger dans le processus.

1) Les conditions $0 \leqslant X < N$ et $(X, N) = 1$ sont-elles restrictives ? Fabriquer un triplet (N, c, d) convenable pour RSA.

2) Ecrire une fonction de (dé)codage, de signature. Évaluer son coût.

3) En supposant qu'un correspondant peu averti accepte de signer un message arbitraire, montrer comment déchiffrer ses messages. En supposant maintenant qu'il refuse de signer un message fixé (mais continue d'accepter les autres), comment obtenir un exemplaire signé du message ?

Problème IV (Logarithme discret, échange de clés)

Soit E un ensemble muni d'une loi multiplicative. L'échange de clé de Diffie–Helmann (1976) est l'ancêtre des cryptosystèmes à clé publique comme RSA. Il est fondé sur le principe suivant : Alice et Bob choisissent ensemble $x \in E$, puis chacun dans leur coin (en secret) un exposant a et b. Puis Alice envoie x^a à Bob, qui lui envoie x^b. A l'aide de son exposant secret, chacun peut reconstituer x^{ab} (la clé) par exponentiation de l'information reçue ($(x^b)^a = (x^a)^b = x^{ab}$). Un espion éventuel ne dispose que de x, x^a et x^b, sans aucun moyen d'obtenir x^{ab} s'il ne devine pas a ou b. Il est donc important qu'il n'y ait pas de fonction logarithme simple définie sur E (en particulier $E = \mathbb{N}$ ne marche pas).

---

[3]PGP = Pretty Good Privacy : chiffreur/déchiffreur gratuit très répandu. SSH = Secure SHell : permet une connexion sécurisée à un serveur informatique, sans que les transactions transitent en clair sur le réseau. Une variante francisée (SSF) est utilisée par l'université d'Orsay pour protéger son réseau.

[4]ce n'est pas tout à fait exact : il existe des attaques qui reconstruisent X sans factoriser N mais, pour l'instant, elles sont inopérantes si des précautions "élémentaires" sont prises.

[5]en particulier NFS (Number Field Sieve), la méthode la plus efficace à ce jour [*record actuel [Décembre 2003] : factorisation d'un module RSA de 174 chiffres décimaux*].

On peut appliquer ce schéma d'échange à $E = (\mathbb{Z}/p\mathbb{Z})^*$, ce qui introduit le problème du logarithme discret : retrouver $x \in \mathbb{Z}/(p-1)\mathbb{Z}$ connaissant g et g^x dans $\mathbb{F}_p^*$. Voici une variante de RSA proposée par ElGammal (1985) où le problème du logarithme discret remplace celui de la factorisation :

Alice choisit une clé secrète $d \in \mathbb{Z}/(p-1)\mathbb{Z}$, un générateur g de $(\mathbb{Z}/p\mathbb{Z})^*$, et publie g et $c := g^d$ (clé publique). Pour chiffrer un message X destiné à Alice, on choisit au hasard $k \in \mathbb{Z}/(p-1)\mathbb{Z}$ et on calcule dans $\mathbb{Z}/p\mathbb{Z}$ le message chiffré $(C_1, C_2) := (g^k, Xc^k)$. Pour déchiffrer, il suffit à Alice de calculer $C_2/C_1^d = X$.

1) Ecrire une fonction de (dé)codage pour la méthode de ElGammal.

★ **2)** On suppose que $p - 1 = \prod q^{e_q}$ est un produit de petits nombres premiers et on veut montrer que l'équation $g^x = a$ est facile à résoudre. Soit l'équation $g_q^{x_q} = a_q$ où $g_q = g^{(p-1)/q}$, $a_q = a^{(p-1)/q}$; montrer que $x = x_q \pmod q$. Comment trouver x_q ? Comment déterminer x modulo q^{e_q} ? Comment déterminer x ?

★★ **Note :** [*description très simplifiée, les algorithmes utilisés en pratique sont nettement plus complexes*] en général, l'équation du logarithme discret $g^x = a$ dans $E = (\mathbb{Z}/p\mathbb{Z})^*$ peut s'attaquer de la façon suivante : on choisit une base de factorisation $B = \{q \text{ premiers}, q \leqslant y\}$ pour un certain y. On calcule g^e pour des valeurs de e prises au hasard, jusqu'à être capable de factoriser sur B l'entier de $]-p/2, p/2[$ congru à g^e modulo p : tous ses facteurs premiers sont inférieurs à y, on dit qu'il est y-friable. On obtient alors une relation dans E : $g^e = \prod q^{a_q}$ que l'on écrit

$$\sum a_q \log_g q \equiv e \pmod{p-1}$$

C'est une équation linéaire, d'inconnues $\log_q$, $q \in B$, que l'on stocke comme i-ème ligne d'une matrice ($a_q \to a_{iq}$, $e \to e_i$). Lorsqu'on a obtenue plus de $|B|$ lignes indépendantes, on peut inverser la matrice (a_{iq}) sur $\mathbb{Z}/(p-1)\mathbb{Z}$ (ce n'est pas un corps, il faut faire attention) et obtenir la valeur des $\log_g q$. Il suffit alors de trouver c tel que ag^c soit y-friable, et on obtient x (en fonction de c et des $\log_g q$ qu'on vient de calculer). Il faut donc ajuster y pour qu'un nombre raisonnable d'entiers $< p$ soient y-friables, tout en se rappelant qu'il faudra obtenir au moins $|B| = \pi(y) \sim y/\log y$ relations.

A partir de résultats de théorie analytique des nombres sur le nombre d'entiers y-friables plus petits que x ($\sim xu^{-u}$, où $u := \log x/\log y$ uniformément dans la région $(\log x)^\varepsilon \leqslant \log y \leqslant (\log x)^{1-\varepsilon}$), on peut estimer qu'il faut de l'ordre de u^u essais pour obtenir chacune des $y/\log y$ factorisations nécessaires. On choisit alors y minimisant l'expression $u^u y/\log y$, pour $x = p$. Le résultat final donne un nombre d'opération espéré inférieur à toute puissance de p quand $p \to \infty$ (c'est un résultat probabiliste). En pratique, on sait calculer des logarithmes discrets sur $\mathbb{F}_p^*$ pour des p de l'ordre de 120 chiffres (en quelques mois), et l'essentiel du temps de calcul est passé à rechercher des relations.

Des idées analogues sont à la base des algorithmes de factorisations modernes. Le but est de trouver x et y tels que $x^2 \equiv y^2 \pmod N$ [*comment en déduire un facteur de N, sauf malchance ?*]. On choisit une base B comme ci-dessus, et on tente de factoriser des carrés sur B : $x^2 = \prod q^{e_q} \pmod N$ où $e_q \in \{0, 1\}$, par exemple pour x proche de $\sqrt{N}$, ce qui implique que x^2 modulo N est petit et a de bonnes chances d'être y-friable. Quand on a assez de relations (plus de $|B|$), on peut éliminer les q (calcul d'un noyau sur $\mathbb{F}_2$) et obtenir une égalité entre deux carrés modulo N [1, Chapitre 10]

Références

[1] H. Cohen, *A course in computational algebraic number theory*, third ed., Springer-Verlag, 1996.