国 防 科 技 大 学 学 报
第20卷 第4期 JOURNAL OF NATIONAL UNIVERSITY OF DEFENSE TECHNOLOGY Vol.20 No.4 1998

用数域筛法分解大整数*

倪谷炎

(国防科技大学系统工程与数学系 长沙 410073)

摘 要 随着 Fermat 数 F_7 和 F_9 被分解, 一个新的算法被提了出来, 那就是 J. Pollard 提出的 "数域筛法"(NFS). A. K. Lenstra 等人对数域筛法进行了深入的研究, 已经使数域筛法从原来对一些特殊整数的分解发展到对一般整数的分解。本文试图对数域筛法理论及其运行作简要的论述。

关键词 整数分解, 算法, 代数数域

分类号 O156.2

Factoring Large Integers with the Number Field Sieve

Ni Guyan

(Department of Systems Engineering and Mathematics, NUDT, Changsha, 410073)

Abstract With the Fermat numbers F_7 and F_9 being factored into primes, a new algorithm, the number field sieve, is given, which was proposed by Pollard. Manasse and Pollard investigate this algorithm thoroughly, and develop it from the special number field sieve (SNFS) to the general number field sieve (GNFS). In this paper, we describe the new algorithm and explain the NFS implementation

Key words factoring integers, algorithm, algebraic number field

众所周知, 早期的公开密钥 RSA 系统之所以流行, 是因为它是建立在大整数的分解极其困难的理论基础之上, 因而使 RSA 系统有很大的安全性。然而, 随着整数分解算法的不断改进和计算机运算速度的加快, 人们对 RSA 系统的安全性又产生了怀疑。在二战期间, 英国间谍运用古老的大型计算机成功地破译了大量的德国军事情报, 为盟军战胜法西斯赢的了主动。正是由于计算机的迅猛发展, 加密与破密的对抗和数论自身理论的提高, 整数分解的研究也就变得特别地活跃, 算法不断地得到改进。当然, 对费尔马数的分解也同样刺激着整数分解的发展。本文将介绍大整数分解的一种新的算法——数域筛法(NFS). 它是从 J. M. Polloard 对 F_7 进行分解的算法中发展起来的。Lenstra 等人用这个新的方法对 F_9 进行分解并获得成功, 同时给出了对特殊整数进行分解的特殊数域筛法(SNFS)——仅对形如 $r^e - s$ (r 和 s 都比较小)的整数进行分解, 逐渐地发展到对一般的整数进行分解的一般数域筛法(GNFS)。为说明 NFS 的优越性, 我们定义

$$L_x[v,\lambda] = \exp(\lambda(\log x)^v(\log\log x)^{1-v}),$$

其中 x, v 和 λ 为实数, 且 $x > e$. 文[4] 指出, 应用数域筛法分解整数费时 $L_N[1/3, c]$, 当它为特殊整数时, 运行SNFS, 则 $c = (32/9)^{1/3} \approx 1.5263$, 而对于一般的整数运行 GNFS, 则 $c = \dfrac{(92+26\sqrt{13})^{1/3}}{3} \approx 1.9$.

函数 $L_n(v, \lambda)$ 在对现代分解整数算法的复杂性分析中起着很重要的作用。特别地, 有

$$L_n[1,\lambda] = n^{\lambda}, L_n[0,\lambda] = (\log n)^{\lambda},$$

$$\log\log L_n[v,\lambda] = v \cdot \log\log L_n[1,\lambda] + (1-v)\log\log L_n[0,\lambda]$$

* 国防预研项目资助
1997年12月5日收稿
倪谷炎, 男, 1966年生, 讲师

在过去传统的算法中(例如简单的试除法) $v=1$, 也就是说它们花时为 $L_n[1,\lambda]$, 对某些 $\lambda>0$。这些算法所花的时间称为指数时间。而多项式时间算法是指 $v=0$的情况, 对于整数分解还没有找到多项式时间算法。花时为 $L_n[1/2,\lambda]$ 的算法称作是一半指数一半多项式时间算法。而数域筛法花时 $L_n[1/3,\lambda]$ 则偏向于多项式时间。现在猜想有很多分解算法包括连分数算法, 二次筛法, 和椭圆曲线法运行期望时间为 $\ln[1/2,1]$, 而对于类群关系法已得到了证明(参见文[10])。三次筛法(参见文[9] 第七节) 猜想其运行时间会更快一些, 花时 $L_n[1/2,c]$, $\sqrt[3]{2/3}\le c<1$, 它只适应于一些特殊形式的数。

1 数域筛法的思想

一般来说, 要把整数 n 进行分解通常是找到两个整数 x 和 y, 满足 $x^2\equiv y^2 \pmod n$。然后计算 $\gcd(x-y,n)$, 如果 $\gcd(x-y,n)=1$, 表示分解失败, 否则就找到了 n 的两个真因子。各种筛法所不同的是 x, y 的找法不一样。数域筛法的思想是:

(1) 构造代数数域。找一个次数为 $d>1$的首1不可约多项式 f 和整数 m, 使 $f(m)\equiv 0 \pmod n$。设 α 是 f 的一个根, 于是得到扩域 $K=Q(\alpha)$, 作映射 $\varphi: Z[\alpha]\longrightarrow Z/nZ$, 由 $\alpha\longrightarrow(m \bmod n)$ 诱导出来。

(2) 找一个数对 (a,b) 的非空集合 S, 满足以下条件:

i) 对所有 $(a,b)\in S$, $\gcd(a,b)=1$;

ii) $\prod_{(a,b)\in S}(a+bm)$ 是 Z 中的平方数;

iii) $\prod_{(a,b)\in S}(a+b\alpha)$ 是 $Z[\alpha]$ 中的平方数。

(3) 令 $x\in Z$ 是 ii) 中数的平方根, $\beta\in Z[\alpha]$ 是 iii) 中数的平方根, 令 $y\in Z$ 适合 $(y \bmod n)=\varphi(\beta)$, 那么我们得到

$$y^2\equiv x^2 \bmod n$$

于是, 算出 n 的因子 $\gcd(x-y,n)$。

下面就按照这个筛法的基本思想来逐步讨论其实现问题。

2 数域的构造

数域的构造实际上是不可约多项式 $f\in Z[x]$ 的构造。在实践中, 对于十进制字节在110～160之间的数 n, 宜取 $d=5$。理论上取

$$d=((3^{1/3}+o(1))\log n/\log\log n)^{1/3},\ n>2^{2d^2}>1.$$

下面就介绍构造 f 的两种方法:

方法**1** 如果存在一个比较小的整数 a 可以表示为 $an=r^e-s$ 的形式, 这里 r, $|s|$ 都是比较小的整数, 那么可以令 k 是满足 $k\cdot d\ge e$ 的最小正整数, $t=s\cdot r^{kd-e}$, 则得到多项式 $f=x^d-t$, 取 $m=r^k$ 满足 $f(m)\equiv 0 \bmod n$。

方法**2** 以"基 m"的方法找 f。取 r 是一个比较小的正整数, $m=[(rn)^{1/d}]$, 然后把 rn 表示成 m 进制

$$rn=c_dm^d+c_{d-1}m^{d-1}+\cdots+c_0,\ 0\le c_i<m$$

于是得到 $f=c_dx^d+\cdots+c_0$, 并且 $f(m)\equiv 0 \bmod n$。选取 $\sum_{i=0}^{d}c_i^2$ 比较小的作为我们的 f。命题2.1确保我们得到的多项式是首1的。

命题**2.1**[7] 当 $d>1$, $n>d^{d^2}$时, $c_d=1$。

我们通常假设 f 是不可约的。否则 $f=gh$, 那么 $f(m)=g(m)h(m)\equiv 0 \pmod n$。如果 $1<\gcd(g(m),n)<n$, 则 n 被分解; 如果 $n\mid g(m)$, 则以 g 代替 f, 否则以 h 代替 f。关于多项式的分解请参阅文[2]。由方法1得到多项式的可约性可由下面的命题判别。

命题**2.2** 多项式 $f=x^d-t$ 可约的充要条件是存在奇素数 $p\mid d$, 且 t 是 p 次方幂, 或 $2\mid d$ 且 t 是平方数, 或 $4\mid d$ 且 $-4t$ 是4次方幂。

令 α 是 f 的一个根, 得到扩域 $K=Q(\alpha)$, 作自然同态

$$\varphi: Z[\alpha]\longrightarrow Z/nZ.$$

可以由 $\varphi(\alpha)=(m \bmod n)$ 导出。即 $\varphi\left(\sum_{i=0}^{d-1} a_i \alpha^i\right)=\left(\sum_{i=0}^{d-1} a_i m^i \bmod n\right)$.

3 $\prod_{(a,b)\in T_1}(a+bm)$ 是 Z 中的平方数的实现

定义**3.1** 如果整数 x 的任何素因数 $p \le y$, 则称 x 是 $y-$ 光滑的。

我们选定一个正数 B_1作为上界, 集合

$$P=\{p \le B_1: p \text{ 是素数}\}=\{p_1, p_2, \cdots, p_{\pi(B_1)}\}$$

作为因子基。筛集合 $T=\{(a,b): a+bm \text{ 是 } B_1- \text{ 光滑的}\}$。作映射

$$e: T \longrightarrow F_2^{1+\pi(B_1)}$$

如果 $a+bm>0$, 则 $e(a,b)$ 的第一个坐标为0, 否则为1; $e(a,b)$ 的第 $i+1$ 个坐标为 $\operatorname{ord}_{p_i}(a+bm) \bmod 2$, 这里 $\operatorname{ord}_{p_i}(a+bm)$ 指 $a+bm$ 的标准分解中 p_i 的次数。

如果 $\#T>1+\pi(B_1)$, 则可以找出相关组, 即 T 的一个非空子集 T_1, 使得

$$\sum_{(a,b)\in T_1} e(a,b) \equiv 0 \bmod 2$$

于是, $\prod_{(a,b)\in T_1}(a+bm)$ 是 Z 中的平方数。这里要用到高斯消去律, 详细情况参见文[8]。

4 $\prod_{(a,b)\in T_2}(a+b\alpha)$ 是 $Z[\alpha]$ 中的平方数的实现

这是数域筛法的核心部分。先介绍代数数论的有关知识, 再介绍特殊数域筛法的实现, 最后介绍一般数域筛法的实现。

假设 β 是 $Z[\alpha]$ 的理想, β 的范数定义为 $N\beta=\#(Z[\alpha]/\beta)$。当 β 是素理想时, 那么 $Z[\alpha]/\beta$ 是有限域, $N\beta=p^t$, p 为某个素数, t 称为素理想 β 的次数。$t=1$时, β 是称为 $Z[\alpha]$ 的一次素理想。$Z[\alpha]$ 的任意一个一次素理想 β 都一一地对应着一个数对(p,r), 这里 p, r 满足 $f(r)\equiv 0 \bmod p$, $p=N\beta$。在 $Z[\alpha]$ 的数中, 我们感兴趣的是 $a+b\alpha, a, b\in Z, b\neq 0$, 因为它有以下性质:

1. $a+b\alpha$的范数 $N(a+b\alpha)=(-b)^d f(-a/b)$;
2. 若 β 是 $Z[\alpha]$的素理想, 且$a+b\alpha\in\beta$, 那么 β 是 $Z[\alpha]$的一次素理想;
3. 若$p \mid N(a+b\alpha)$, 则存在 $Z[\alpha]$的素理想 β 使的 $N\beta=p$, 且 $a+b\alpha\in\beta$。

记 $R(p)=\{r\in\{0,1,\cdots,p-1\}: f(r)\equiv 0 \bmod p\}$。对任意$(p,r)$, $r\in R(p)$, 定义

$$e_{(p,r)}(a+b\alpha)=\begin{cases} ord_p N(a+b\alpha), & \text{若 } a+br\equiv 0 \bmod p; \\ 0, & \text{若 } a+br\not\equiv 0 \bmod p. \end{cases}$$

不难得到性质

4. $\prod_{p,r} p^{e_{p,r}(a+b\alpha)}=\left|N(a+b\alpha)\right|$, 这里 p 跑遍所有素数, r 跑遍 $R(p)$。

如果 $K=Q(\alpha)$ 的整数环 O_K 是唯一分解环, 且 $O_K=Z[\alpha]$时, 则由上面的4个性质, $a+b\alpha\in Z[\alpha]$ 在 $Z[\alpha]$上的数分解可以按下面的步骤进行:

1. 分解 $N(a+b\alpha)=(-b)^d f(-a/b)=p_1^{e_1}\cdots p_k^{e_k}$;
2. 对于每个 i, 求 r_i, 满足 $a+br_i\equiv 0 \pmod{p_i}$;
3. 作$a+b\alpha$的素理想分解

$$(a+b\alpha)=(p_1, r_1)^{e_1}\cdots(p_k, r_k)^{e_1};$$

4. 由于 O_K 是唯一分解环, 因而每个理想(p_i, r_i) 都是主理想, 即有生成元 π_{p_i}, 使得素理想$(p_i, r_i)=(\pi_{p_i})$。
5. 于是 $a+b\alpha$ 在 $Z[\alpha]$中可分解为

$$a+b\alpha=u\cdot\pi_{p_1}^{e_1}\cdots\pi_{p_k}^{e_k},$$

这里 u 是 $Z[\alpha]$中的某个单位。

定义**4.1** 数域 K 的代数整数环 O_K 的元素 β称作为 $y-$ 光滑的, 是指其范数 $N\beta$ 是 $y-$ 光滑的。

令 U 是 $Z[\alpha]$ 的单位群的生成元集合, B_2是一个正数, $B'=\{(p,r): p\le B_2, r\in R(p)\}$, 并把 U 和 B'

中的元素排好次序作为代数因子基。可以这样将 T_2 实现:

1. 找集合 $T=\{(a,b):a+b\alpha$ 是 B_2- 光滑的$\}$;

2. 作映射 $e:T\longrightarrow F_2^{\#U+\#B'}$。其中 $e(a,b)$ 的前$\#U$ 个坐标分别对应于 $u\in U$ 的指数 $\bmod 2$; $e(a,b)$ 的后$\#B'$ 个坐标分别对应于 $e_{p,r}(a+b\alpha) \bmod 2$, 这里$(p,r)\in B'$

3. 运用高斯消去律找出相关组, 即 T 的子集 T_2。

但是, O_K 是唯一分解环的假设往往是不成立的。这就要求对一般的数域 K 进行讨论。这就碰到以下的4个障碍:

1. $\prod_{(a,b)\in T_2}(a+b\alpha)O_K$ 不一定是一个理想的平方;

2. 即使$\prod_{(a,b)\in T_2}(a+b\alpha)O_K=A^2$, 其中 A 是 O_K 中的某个理想, A 也不一定是主理想;

3. 即使$\prod_{(a,b)\in T_2}(a+b\alpha)O_K=r^2O_K, r\in O_K$, 也不一定得出$\prod_{(a,b)\in T_2}(a+b\alpha)=r^2$。

4. 即使$\prod_{(a,b)\in T_2}(a+b\alpha)=r^2, r\in O_K$, 也不一定得出 $r\in Z[\alpha]$。

第4个障碍可这样解决, 因 $r\in O_K$, 那么 $rf'(\alpha)\in Z[\alpha]$, 所以$f'(\alpha)^2\prod_{(a,b)\in T_2}(a+b\alpha)$ 是 $Z[\alpha]$ 中的平方数。运用以下两个命题可以设法绕过前面三个障碍:

命题**4.1**[7] 如果$\prod_{(a,b)\in T_2}(a+b\alpha)$ 是 O_K 中的平方数, 则对任意的$(p,r)\in B'$, 都有

$$\sum_{(a,b)\in T_2} e_{p,r}(a+b\alpha)\equiv 0 \bmod 2.$$

命题**4.2**[7] 如果$\prod_{(a,b)\in T_2}(a+b\alpha)$ 是 K 中的平方数, 令 q 是奇素数, 对任意的$s\in R(q)$, 满足 $a+bs\not\equiv 0 \bmod q, f'(s)\not\equiv 0 \bmod q$, 那么

$$\prod_{(a,b)\in T_2}((a+bs)/q)=1.$$

这里$(./.)$表示勒朗德符号。

以上两个命题只是给出了找 T^2 的必要条件, 但是目前还没有什么更好的办法。下面我们看看当 O_K 不是唯一分解环时, T_2 是怎么实现的:

1. 找集合 T, 及集合 B 并排序。

2. 找集合 B'' 并排序。这里$\#B''=[3(\log n)/(\log 2)]$,

$$B''=\{(q,s):q\text{ 为素数}, q>B_2, s\in R(q), f'(s)\not\equiv 0 \bmod q\}$$

3. 作映射 $e:T\longrightarrow F_2^{\#B+\#B''}$, $e(a,b)$ 的前$\#B$ 个坐标对应于 $e_{p,r}(a+b\alpha) \bmod 2, (p,r)\in B'$后$\#B''$ 个坐标为0, 如果$((a+bs)/q)=1$, 或1, 如果$((a+bs)/q)=-1, (q,s)\in B''$。

4. 应用高斯消去律找线性相关组, 即 T 的子集 T_2。

由此得出的集合 T_2 并不能一定确保$\prod_{(a,b)\in T_2}(a+b\alpha)$ 是 $Z[\alpha]$ 中的平方数, 但实践证明该算法是可行的。于是我们得到集合 $S=T_1\cap T_2$。

5 平方根的计算

令

$$c=f'(m)^2\prod_{(a,b)\in S}(a+bm),\quad \gamma=f'(m)^2\prod_{(a,b)\in S}(a+b\alpha)$$

而假定 S 是数对(a,b)集合, 满足

(1) $\gcd(a,b)=1$, 对任意的$(a,b)\in S$;

(2) c 是 Z 中的平方数,

(3) γ 是 $Z[\alpha]$ 中的平方数。

我们要计算 c 和 γ 的平方根。

假定$c=x^2, \gamma=\beta^2, x\in Z, \beta\in Z[\alpha]$, 且假定 S 是按照第4节的方式构造出来的, 则有

$$c = f'(m)^2 \prod_{p \le B_1} p^{2e_p} = x^2, x = f'(m) \prod_{p \le B_1} p^{e_p},$$

$$N(\gamma) = N(f'(\alpha))^2 \prod_{p \le B_2} p^{2f_p}, N(\beta) = \pm N(f'(\alpha)) \prod_{p \le B_2} p^{f_p}$$

关于 β 的算法这里介绍两种方法。

算法**5.1** 本算法用来计算 γ 在 $Z[\alpha]$ 上的平方根 β。

1. 把 γ 表示为关于 α 的多项式形式, 次数小于 d;

2. 假定得到奇素数 q, 使 $f \bmod q$ 不可约, 那么 $D = qZ[\alpha]$ 就是 $Z[\alpha]$ 中 d 次素理想, 这里 $f'(\alpha) \notin D$, 且 $a + b\alpha \notin D$。

3. 在有限域 $Z[\alpha]/D$ 中求 $\gamma \bmod q$ 的平方根。其实在这里我们是找一个元素 $\delta_0 (\bmod D)$, 满足 $\delta_0^2 \gamma \equiv 1 (\bmod D)$, 这个 $\delta_0 (\bmod D)$ 是唯一确定的。

4. 从 δ_0 开始, 应用牛顿迭代

$$\delta_j \equiv \frac{\delta_{j-1}(3 - \delta_{j-1}\gamma)}{2} (\bmod D^{2^j})$$

依次求出 $\delta_1, \delta_2, \cdots$ 满足 $\delta_j^2 \gamma \equiv 1 (\bmod D^{2^j})$。直到 q^{2^j} 至少是 β 的系数绝对值的两倍。

5. 计算 $\beta \equiv \delta_j \gamma (\bmod D^{2^j})$。

算法**5.2** 用于计算 γ 的平方根 β, $y = \varphi(\beta)$, 并最终算出 $\gcd(y - x, n)$。

1. 按要求选取适当的 $m_i = q_i^{k_i}$;

2. 对每个 i 计算 $\mathrm{rem}(M_i, m_i)$, a_i 和 $\mathrm{rem}(M_i, n)$, 这里 $M = \Pi_i m_i$, $a_i = 1/M_i (\bmod m_i)$, $\mathrm{rem}(u, v)$ 表示 $u (\bmod v)$;

3. 对每个模 m_i, 计算

$$\gamma_i = f' \prod_{(a,b) \in S} (a + bx) \ (\bmod f, m_i)$$

同时计算 γ_i 的平方根 β_i;

4. 计算 $\beta_i \bmod q_i$ 范数 $N_{1,i}$ 和 $N(\beta) \bmod q_i = N_{2,i}$。如果 $N_{1,i} \neq N_{2,i}$, 则以 $-\beta_i$ 代替 β_i。令 $B_i \in Z[x]$ 次数 $\le d - 1$, $\beta_i \equiv B_i (\bmod f, m_i)$ 计算 $B_i(m) (\bmod m_i)$;

5. 计算 $B(m) \bmod n$;

6. 输出 $\gcd(B(m) - x, n)$。

其中第2, 3, 4步可以进行并行计算。

这两种算法各有利弊。算法5.1相对来说花时少一些, 但会碰到大整数的计算, 算法5.2的花时虽然多一点, 但运算的数字都比较小, 而且可以进行并行处理, 因而应用算法5.2的人比较多。

6 算法总结

由以上的讨论, 我们把数域筛法作个总结:

算法**6.1** 给定一个正整数 n 以及参数 d, u 和 y, 满足 $d > 1$, $n > d^{2d^2}$。本算法试图分解出 n 的一个非平凡因子, 不管成功与否都会停机。

1. 检验 n 是否为一个素数的方幂, 或用 $\le y$ 的数试除[3]。无任那种情况, 若输出素数则停机。

2. 应用 "基 m" 方法找一个首1多项式 $f \in Z[x]$, 次数为 d, 满足 $f(m) \equiv 0 \bmod n$。如果发现 f 是可约的, 用文[2]方法把 f 分解为不可约因子的积, 得到非平凡因子 g, 输出 n 的非平凡因子 $\gcd(g(m), n)$, 并停机。假设 f 是不可约的, α 为 f 的一个根, 计算 $\gcd(f'(m), n)$, 如果得到 n 的一个非平凡因子, 则输出该因子并停机。

3. 用筛法找集合

$$T = \{(a, b) \in Z^2 : \gcd(a, b) = 1, |a| \le u, 0 < b \le u, (a + bm)N(a + b\alpha) \text{ 是 } y - \text{光滑的}\}$$

4. 找因子基 B, B', B''。

$$B = \{\text{素数 } p : p \le y\}$$

$$B' = \{(p,r): p \text{ 为素数}, p \leq y, r \in R(p)\}$$

$$B'' = \{(q,s): q \text{ 为素数}, q > y, s \in R(q), f(s) \not\equiv 0 \bmod q\}, \#B'' = [3\log n/\log 2]$$

5. 作一个矩阵。它的行向量 $e(a,b)$ 定义如下: $e: T \longrightarrow f_2^{1+\#B+\#B'+\#B''}$,

i) $e(a,b)$ 的第一个坐标 $= \begin{cases} 0, \text{如果 } a+bm > 0; \\ 1, \text{如果 } a+bm < 0. \end{cases}$

ii) $e(a,b)$ 接下来 $\#B$ 个坐标 $= ord_p(a+bm) \bmod 2, p \in B$。

iii) 接下来 $\#B'$ 个坐标 $= e_{p,r}(a+b\alpha) \bmod 2, (p,r) \in B'$。

iv) 最后 $\#B''$ 个坐标 $= \begin{cases} 0, \text{如果}((a+bs)/q) = 1; \\ 1, \text{如果}(a+bs)/q) = -1. \end{cases}$

找出相关组, 如果不成功则停机。若成功则令 S 为相关组中对应的数对 (a,b) 集。

6. 把代数整数 $\gamma = f(\alpha)^2 \prod_{(a,b)\in S}(a+b\alpha)$ 表示为关于 α 次数 $< d$ 的多项式形式。求 γ 的平方根 $\beta = \sum_{i=0}^{d-1} b_i \alpha^i$。如果这步不成功则停机。

7. 对整数 c, 满足 $c^2 = f(m)^2 \prod_{(a,b)\in S}(a+bm)$ 计算 $c \bmod n$。

8. 计算 $\gcd(c - \sum_{i=0}^{d-1} b_i m^i, n)$。如果这是一个 n 的非平凡因子, 输出结果, 并停机。否则回到第5步, 重新找 S 集。

其中参数 u 和 y 理论上的取值为

$$u = y = L_n[1/3, (8/9)^{1/3} + o(1)], \text{对 } n \longrightarrow \infty$$

参考文献

1 Lenstra A K, Lenstra H W Jr. The development of the number field sieve. Springer Verlag Berlin, 1993

2 Lenstra A K, Lenstra H W, Lovasz Jr. L. Factoring Polynomials With Rational Coefficients. Math. Ann, 1982

3 Lenstra A K, Lenstra H W Jr. Manasse M. S Pollard J. M. The factorizatiion of the ninth Fermat number. Math. Comp. 1993, 61

4 Lenstra A K et al. The number field sieve, Springer Verlag Berlin, 1993

5 Jean-Marc Couveignes. Computing a square root for the number field sieve, Springer Verlag Berlin, 1993

6 Daniel J Bernstein, Lenstra A K. A general number field sieve implementation, Springer Verlag Berlin, 1993

7 Buhler J P. et aL. Factoring integers with the number field sieve. Springer Verlag Berlin, 1993

8 Wiedeman D. Solving sparse linear equations over finite fields. IEEE Trans. Inform. Theory, 1986, 32

9 Coppersmith D. at al Discrete logarithms in GF(p). Algorithmica 1986, 1: 1–15.

10 Lenstra H W. A rigorous time boud for factoring integers. J. Amer. Math. Soc 1992 5: 483–516

11 Pollard J M. The Lattice Sieve, Springer Verlag Berlin, 1993

12 Mignotte M. Mathematiques pour le calcul formel. Presses Universitaires de France, Paris, 1989

13 Lang S. Algebraic number theory. Addison-Wesley, Reading, Massachusets, 1970

14 Rudol L, Harald N. Finite fields. Addison-Wesley, Reading, Massachusets, 1983

15 Lensrta H W, Tijdeman Jr. R. Computation methods in number theory. second edition, Mathematisch Centrurm Amsterdam, 1984