

将椭圆曲线分解算法扩展为三阶段的方案

罗贵文^{1,2}

(1. 中国科学院大学网络空间安全学院, 北京 100049;
2. 中国科学院信息工程研究所信息安全国家重点实验室, 北京 100093)

摘要: 椭圆曲线法是目前使用最广泛的整数分解算法之一, 最早由 Lenstra 于 1985 年提出, 原始的算法只有第一阶段。自其提出以来, 围绕算法和实现的研究层出不穷, 最重要的改进是 Brent 和 Montgomery 提出椭圆曲线法的第二阶段, 这极大地提升了椭圆曲线算法的分解能力和效率。将椭圆曲线法扩展为三阶段, 采用的方法是将第一阶段和第二阶段进行“融合”。对比目前流行的两阶段椭圆曲线法, 改进后的算法有两方面的优点: 一是在保持同两阶段椭圆曲线法参数相同的情况下, 通过增加微不足道的消耗, 提升找到因子的概率; 二是在搜寻同一个因子时, 可以使用较小的“光滑参数”。

关键词: 整数分解; 快速分解; 椭圆曲线法; 光滑参数

中图分类号: TP301

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2018101

Scheme of extending elliptic curve method to three phases

LUO Guiwen^{1,2}

1. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
2. State Key Laboratory of Information Security, Institute of Information Engineering,
Chinese Academy of Sciences, Beijing 100093, China

Abstract: Elliptic curve method for integer factorization (ECM) is one of the most popular integer factorization algorithms, and it was firstly proposed by Lenstra in 1985. The original ECM contained just first phase. Since its invention, researches about the algorithm and implementation emerged up, among which the most important improvement is the extension to two phases proposed by Brent and Montgomery. This improvement tremendously strengthened ECM's capacity and efficiency. Elliptic curve method was extended to three phases. Extension method is kind of like “mixing together” the first phase and second phase. Compared to the best current two phases ECM, the new algorithm shows 2 advantages. First, under the same factorization parameters, the proposed algorithm improves the probability of finding out prime factor at the expense of negligible increasement of time. Second, when searching the same prime factor, the proposed algorithm can utilize smaller “smoothness parameters”.

Key words: integer factorization, fast factorization, elliptic curve method, smoothness parameter

收稿日期: 2018-11-02; 修回日期: 2018-11-28

通信作者: 罗贵文, loughuiwen@iie.ac.cn

基金项目: 国防科技创新特区基金资助项目 (No.Y7H0041102)

Foundation Item: The National Defense Science and Technology Innovation Foundation (No.Y7H0041102)

1 引言

众所周知，整数分解在密码学中扮演着举足轻重的角色。已知的整数分解算法众多，包括经典的试除法、Pollard's rho 算法、Pollard's $p-1$ 算法、Williams' $p+1$ 算法、椭圆曲线法，一般数域筛法，以及适用于量子计算机的 Shor 算法等。在这些算法中，椭圆曲线法是应用最广泛的快速整数分解算法之一，它的运行时间取决于待搜寻的因子大小，因此常常部署在一般数域筛法之前，用以移除待分解整数的较小因子。

自 Lenstra 提出只含有第一阶段的椭圆曲线法以来^[1]，针对它最重要的改进是将其扩展为两阶段，这极大地提升了算法搜寻大素因子的能力。截至目前，利用椭圆曲线法搜寻到的最大素因子是一个 83 十进制位素数，它是由 Propper 于 2013 年分解 $7^{337}+1$ 时得到的。原始的椭圆曲线法只包含“第一阶段”。在文献[2]中，Brent 提出了 2 种将椭圆曲线法扩展为两阶段的方法：一种是“ $p-1$ 第二阶段”；另一种是“生日悖论第二阶段”，他也提到了 Brent-Suyama 扩展以及在第二阶段利用快速多项式求值算法的可能性，这些改进目前已经被广泛应用。在文献[3]中，Montgomery 对 $p-1$ 法、 $p+1$ 法以及椭圆曲线法进行了统一阐释，并提出在第二阶段使用 FFT 扩展，随后，他在文献[4]中对 FFT 扩展做了进一步改进，极大地提升了第二阶段的速度。自此，椭圆曲线法的基本框架已经建立，发展日趋完善。

2005 年，Zimmermann 等在文献[5]中总结了 20 年来椭圆曲线法的发展，并在文末提出一个开放问题：是否有可能设计“第三阶段”，进一步提升椭圆曲线法的分解能力？可惜的是，目前并没有发现这一方向的进展。本文通过将第一阶段和第二阶段进行“融合”，把椭圆曲线法扩展为三阶段，以期提升椭圆曲线法搜寻大素因子的能力。文中提出的新算法可以从两方面加以利用：一是在保持同两阶段椭圆曲线法参数相同的情况下，通过增加少许计算量，提升算法找到因子的概率；二是在搜寻同一个因子时，可以减小“光滑参数”的值。

2 椭圆曲线分解算法

假设本文致力于找出 n 的某个素因子 p 。任意选取 a, b ，方程 $y^2 = x^3 + ax + b \bmod n$ 也蕴含着 $y^2 = x^3 + ax + b \bmod p$ ，且后者在椭圆曲线上定义的加法意义下形成群，而前者则不然。这是因为 Z_n 不是域，在进行椭圆曲线加法的运算过程中，求两点连线的斜率时需要求逆，这一步有可能失效。但恰恰是这点为利用椭圆曲线法分解整数提供了机遇。在 $\bmod n$ 的情形下，按照椭圆曲线定义的加法意义进行计算。若求斜率时，分母在 Z_n 中的逆元不存在，意味着分母和 n 的最大公因数不为 1，该最大公因数是 n 的非平凡因子（该最大公因数可能为平凡因子 n ，但这样的概率很小，并且可以重新选择参数再次进行分解，因此不予考虑）。

由于 p 是未知的，运算只能在 $\bmod n$ 下进行，根据前面的分析，为了找出 n 的非平凡因子，构造逆元不存在的情形。任选椭圆曲线上一点 P_0 ，设在 $\bmod p$ 下椭圆曲线群的阶为 N_p ，那么 $N_p \cdot P_0 = O$ ，这里 O 表示无穷远点。在利用仿射坐标计算 $N_p \cdot P_0$ 时会出现 $\bmod n$ 时逆元不存在的情形。由 Hasse 定理，群的阶

$$N_p = p + 1 + t, |t| \leq 2\sqrt{p}$$

虽然 N_p 未知，但如果 N_p 是“光滑的”，那么最终就能找出 p 。不是每一条椭圆曲线的阶都是“光滑的”，椭圆曲线法是概率算法，光滑参数选取越大，测试的曲线越多，找出因子 p 的概率越大。

如算法 1 所示，目前的椭圆曲线法需要 2 个“光滑参数” B_1, B_2 ($B_1 < B_2$)。如果 N_p 最大的素因子不超过 B_2 ，并且其余的素因子不超过 B_1 ，就称 N_p 为 (B_1, B_2) “光滑的”。在实际实现椭圆曲线法时，为了提升速度，避免每次运算都计算最大公因数和逆元，一般采用 Montgomery 形式下的齐次射影坐标进行运算。算法 1 中的步骤 1)~步骤 4)称为第一阶段（first step 或 first phase），也是 Lenstra 提出椭圆曲线法的最初版本；步骤 5)~步骤 8)称为第二阶段。根据该算法，结合上文的分析，如果

$$N_p \left| \left(\prod_{\{p \leq B_1\}} p^{\left\lfloor \frac{\log(B_1)}{\log(p)} \right\rfloor} \right) \cdot q_1, B_1 < q_1 \leq B_2 \right.$$

那么算法1能够找到 n 的素因子(这只是充分条件),这里所有的 p 和 q_1 均为素数。

算法1 两阶段的椭圆曲线分解算法

输入 既不能被2整除又不能被3整除的整数 n , 以及光滑参数 $B_1 \leq B_2$

输出 n 的因子或FAIL

- 1) 任选椭圆曲线 $E \bmod n$ 及该曲线上一点 $P_0=(x_0:y_0:z_0)$
- 2) 在曲线 $E \bmod n$ 上计算

$$Q = \prod_{\{p \leq B_1\}} p^{\left\lfloor \frac{\log(B_1)}{\log(p)} \right\rfloor} \cdot P_0$$

- 3) 设 $Q=(x_Q:y_Q:z_Q)$, 计算 $g=\gcd(n, x_Q)$
- 4) 如果 $g \neq 1$, 输出 g 并终止, 否则继续
- 5) 对每个素数 $p(B_1 < p \leq B_2)$
- 6) 计算 $(x_P:y_P:z_P)=p \cdot Q$
- 7) 计算 $g=\gcd(n, x_P)$
- 8) 如果 $g \neq 1$, 输出 g 并终止
- 9) 输出FAIL

实现时, 需要进行快速标量乘运算。因为在算法中并没有使用 y 坐标, 所以使用Weierstrass形式并不是最佳选择, 常采用齐次的Montgomery形式进行计算, 这样的好处是在计算倍点时, 可以只计算 x 坐标和 z 坐标^[3,7], 从而实现更快的运算速度。Montgomery形式需要配合差加链(differential addition chain, 即Lucas chain)才能实现倍点运算, 常用的差加链是二进制链, GMP-ECM采用了Montgomery设计的PRAC链^[8-9]。Suyama参数在初始化阶段广泛应用, 方便找出分解成功的椭圆曲线, 重现分解过程。第一阶段计算椭圆曲线 E 上的点 Q , 若第一阶段未能找出 n 的非平凡因子, 即如果 $\gcd(n, x_Q)=1$, 算法1进入第二阶段。第二阶段采用类似“小步大步法(baby step giant step)”的策略, 避免计算每一个倍点^[10-12]。也有采用Edwards曲线实现算法1的尝试^[13-14], 相比较Montgomery形式和Weierstrass形式而言, 使用Edwards曲线可以减少模乘运算, 而且找到因子的概率更高。此外, 最近有研究提出算法1的量子计算

版本^[15]。

3 椭圆曲线法扩展为三阶段

本节提出一种算法1的改进算法, 将椭圆曲线法扩展为三阶段。通过增加较小的计算量, 提高椭圆曲线分解算法做出分解的可能性。如算法2所示, 使用3个“光滑参数” $B_1, B_2, B_3(B_1 < B_2 \leq B_3)$ 。算法2的步骤1)~步骤4)称为第一阶段, 步骤5)~步骤7)称为第二阶段, 步骤8)~步骤11)称为第三阶段。

与算法1对比, 算法2的第一阶段与第三阶段分别相当于算法1的第一阶段和第二阶段。若

$$N_p \left| \left(\prod_{\{p \leq B_1\}} p^{\left\lfloor \frac{\log(B_1)}{\log(p)} \right\rfloor} \right) \cdot q_1 \cdot q_2, B_1 < q_1 \leq B_2, B_2 < q_2 \leq B_3 \right.$$

那么算法2有一定概率找出 n 的素因子, 这里所有的 p 和 q_1, q_2 均为素数。

算法2 扩展到三阶段的椭圆曲线分解算法

输入 既不能被2整除又不能被3整除的整数 n , 以及光滑参数 B_1, B_2, B_3

输出 n 的因子, 或FAIL

- 1) 任选椭圆曲线 $E \bmod n$ 及该曲线上一点 $P_0=(x_0:y_0:z_0)$
- 2) 在曲线 $E \bmod n$ 上计算

$$Q = \prod_{\{p \leq B_1\}} p^{\left\lfloor \frac{\log(B_1)}{\log(p)} \right\rfloor} \cdot P_0$$

- 3) 设 $Q=(x_Q:y_Q:z_Q)$, 计算 $g=\gcd(n, x_Q)$
- 4) 如果 $g \neq 1$, 输出 g 并终止, 否则继续
- 5) 随机选取 r 个素数 $p_1, p_2, \dots, p_r, B_1 \leq p_i \leq B_2, i \leq r$, 计算 $Q_1 = \left(\prod_{i=1}^r p_i \right) \cdot Q$
- 6) 设 $Q_1=(x_{Q_1}:y_{Q_1}:z_{Q_1})$, 计算 $g=\gcd(n, x_{Q_1})$
- 7) 如果 $g \neq 1$, 输出 g 并终止, 否则继续
- 8) 对每个素数 $p(B_1 < p \leq B_3)$
- 9) 计算 $(x_P:y_P:z_P)=p \cdot Q_1$
- 10) 计算 $g=\gcd(n, x_P)$
- 11) 如果 $g \neq 1$, 输出 g 并终止
- 12) 输出FAIL

预先取定整数 B_1, B_2 , 将算法1中的“光滑参数”取为 B_1, B_2 , 将算法2中的“光滑参数”取为 B_1, B_2, B_3 , 算法2的第一阶段与第三阶段

分别等同于算法 1 的第一阶段和第二阶段。如果 N_p 最大的素因子不超过 B_2 , 并且第二大的素因子不超过 B_1 , 那么算法 1 可以找出 n 的素因子 p ; 如果 N_p 最大的素因子不超过 B_2 , 第二大的素因子不超过 B_2 , 并且第三大的素因子不超过 B_1 , 此时算法 1 便束手无策, 但算法 2 却有可能找出 n 的素因子 p 。因此算法 2 新设计的第二阶段增大了椭圆曲线法成功分解的概率, 增加的消耗是 r 个椭圆曲线标量乘运算。

若 N_p 的第二大素因子落在 B_1, B_2 间, 假设 B_1, B_2 间的素数个数为 m , 算法 2 新设计的第二阶段“击中” N_p 的第二大素因子的概率为 (在 $r \ll m$ 的情况下)。

$$\text{Prob} = 1 - \prod_{j=1}^r \left(1 - \frac{j}{m} \right) \sim 1 - \exp \left(1 - \frac{r^2}{2m} \right)$$

要使 $\text{Prob} \geq \frac{1}{2}$, 只需要 $r \geq \sqrt{2m \ln 2}$ 。实践中, 这样的消耗增加是微不足道的。举例来讲, 如果要搜寻一个规模约为 60 十进制位的素因子, GMP-ECM 7.0.4^[6]推荐使用 $B_1 = 2.6 \times 10^8$, $B_2 = 3.2 \times 10^{12}$, 此时, 算法 2 第一阶段中需要的椭圆曲线标量乘运算超过 $\pi(B_1) = 14\,195\,860$ 个, π 表示素数计数函数。而

$$r = \sqrt{2m \ln 2} < \sqrt{2Li(B_2) \ln 2} \approx 4.0 \times 10^5$$

Li 表示对数积分函数。在这种情况下 $r \ll \pi(B_1)$ 。

4 实践

第 3 节阐述了算法 2 在增加少许计算量的情况下, 相比算法 1 提高了成功分解的概率。从另外一个角度考虑, 在搜寻同一个素因子时, 算法 2 可以适当减小“光滑参数”。下面以具体实践来阐述这一论断。

2018 年“第三届全国高校密码数学挑战赛”赛题二要求参赛选手分解一个 432 十进制位的大整数, 在将所有小素因子找出后, 最困难的部分是分解一个 116 十进制位的整数 $n=18311422666613585891834635740997396678045469120835214703400815986739539158582260051188900502060268468088445260535641$ 。

使用算法 2 对 n 进行分解, 分别取 3 个“光滑参数”, $B_1=3 \times 10^6$, $B_2=B_3=1 \times 10^9$, 取

$$r = 8396 > \sqrt{2m \ln 2}$$

利用 Suyama 参数 σ 表示 Montgomery 形式的椭圆曲线。在使用 $\sigma=22\,824\,341$ 所对应的椭圆曲线进行分解时, 成功地搜寻出 n 的 58 十进制位素因子 $p=3457645796086684459525372196446947094107078575533908472103$ 。

该椭圆曲线的阶为 N_p 。

$$N_p = 2^5 \cdot 3 \cdot 5 \cdot 7 \cdot 353 \cdot 1439 \cdot 4231 \cdot 20929 \cdot 28753 \cdot 144553 \cdot 3 \cdot 1533487 \cdot 2283881 \cdot 306842093 \cdot 512194063$$

若按照 GMP-ECM 7.0.4 推荐的搜寻 60 十进制位素因子的参数设定, $B_1=2.6 \times 10^8$, $B_2=3.2 \times 10^{12}$, 并利用算法 1, 使用该椭圆曲线无法成功分解出这一 58 十进制位素因子 p 。

5 结束语

本文将目前流行的两阶段椭圆曲线法扩展为三阶段, 并通过理论分析和实践, 从两方面阐述了改进后算法的优点, 一是在保持同两阶段椭圆曲线法参数相同的情况下, 通过增加微不足道的消耗, 提升找到因子的概率; 二是在搜寻同一个因子时, 可以使用较小的“光滑参数”。此外, 改进后的算法可能有能力搜寻更大的素因子。椭圆曲线法的瓶颈主要在第一阶段消耗过大, 利用算法 2, 可以适当减小 B_1 , 增大 B_2 与 B_3 , 从而增强使用椭圆曲线法搜寻更大素因子的能力。

致谢

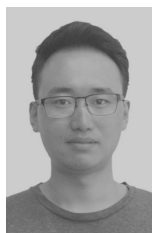
感谢吴宝峰、吕丽君在成文过程中和我进行诸多有益讨论, 并提出宝贵意见。

参考文献:

- [1] LENSTRA H. Factoring integer with elliptic curves[J]. Ann of Math, 1987, 126.
- [2] BRENT R P. Some integer factorization algorithms using elliptic curves[J]. Australian National University Technical Report, 1985, 8:149-163.
- [3] MONTGOMERY P L. Speeding the pollard and elliptic curve methods of factorization[J]. Mathematics of Computation, 1987, 48(177): 243-264.
- [4] MONTGOMERY P L. An FFT extension of the elliptic curve method of factorization[M]. University of California at Los Angeles, 1992.
- [5] ZIMMERMANN P, DODSON B. 20 years of ECM[C]// Interna-

- tional Algorithmic Number Theory Symposium. 2006: 525-542.
- [6] The gmp-ecm development team, GMP-ECM, an implementation of the elliptic curve method algorithm, release 7.0.4[R]. 2017.
- [7] OKEYA K, KURUMATANI H, SAKURAI K. Elliptic curves with the montgomery-form and their cryptographic applications[C]//International Workshop on Practice & Theory in Public Key Cryptography: Public Key Cryptography. 2000.
- [8] MONTGOMERY P L. Evaluating recurrences of form $x_{m+n}=f(x_m, x_n, x_{m-n})$ via Lucas chains[M]. 1983.
- [9] TALL A. A generalization of the Lucas addition chains[J]. Bulletin mathématique de la Société des Sciences Mathématiques de Roumanie, 2012: 79-93.
- [10] MONTGOMERY P L. An FFT extension of the elliptic curve method of factorization[D]. Los Angeles: University of California, 1992.
- [11] JOACHIM V Z G, GERHARD J. Modern computer algebra[M]. Modern Computer Algebra. 2001.
- [12] HANROT G, QUERCIA M, ZIMMERMANN P. The middle product algorithm I[J]. Applicable Algebra in Engineering, Communication and Computing, 2004, 14(6):415-438.
- [13] BERNSTEIN D, BIRKNER P, LANGE T, et al. ECM using edwards curves[J]. Mathematics of Computation, 2012, 82(282):16.
- [14] BERNSTEIN D J, BIRKNER P, JOYE M, et al. Twisted edwards curves[J]. Lecture Notes in Computer Science, 2008, 5023: 389-405.
- [15] BERNSTEIN D J, HENINGER N, LOU P, et al. Post-quantum RSA[C]//International Workshop on Post-Quantum Cryptography. 2017: 311-329.

[作者简介]



罗贵文（1993-），男，重庆人，中国科学院大学硕士生，主要研究方向为椭圆曲线密码学。