素因数分解の世界記録はいかに作られたか

A Backstage Story about the New World Record of Integer Factorization

青木和麻呂

Abstract

NTT ＋ Bonn 大＋ EPFL の研究チームは 2007 年 5 月に小さな因子のない 1,000 ビット（10 進 300 けた）を超える合成数の素因数分解を完了したことを報告した．今回分解した合成数は $2^{1039}-1$ といった特殊な形であるので，直ちに 1,000 ビット程度の法を持つ RSA 署名などに影響を及ぼすものではないが，1999 年に作られた 512 ビット合成数の素因数分解に次ぐ一里塚として意義深いものである．今回の分解実験は技術的には予想外に順調に進んだが，これは過去の記録達成のための経験があったからにほかならない．本稿では過去の素因数分解実験における純粋な数学的問題以外に起きた問題や解決などについて記述し，どう前述の記録につながったかについて報告する．

キーワード：素因数分解，数体篩（ふるい）法，PC クラスタ，故障

1. はじめに

NTT は 2007 年 5 月に，Bonn 大及び EPFL と共同で小さな因子がない 1,000 ビットを超える合成数の分解に成功したことを報告した[(1),(2)]．分解対象は $(2^{1039}-1)/5{,}080{,}711$ といった特別な形ではあるが従来の記録である 911 ビットを 100 ビット以上超える 1,017 ビット合成数の分解であった．筆者は，好運にもこの成果にかかわることができ，大きな達成感を味わった．本稿ではこの記録に至るまでに経験したことを中心に巨大数の素因数分解[(用語)]計算について述べる．

2. 素因数分解問題

素因数分解問題とは狭義には与えられた合成数 N に対し，自明な因子である 1 と N 以外の素因子を求める問題である．この操作を繰り返せば N の完全な素因数分解を求めることができる．

素因数分解は，義務教育で習ったように 2, 3, 5, 7, …と小さい素数から順に N を割っていけばできる．この方法は最悪の場合 $\sqrt{N}$ 以下の素数すべてについて試す必要があるので，N の長さ（$=\log_2 N$）の指数の時間を要することから効率が悪い．しかし，現在においても，既知の最も効率が良い方法でも準指数時間を要する方法しかないため，素因数分解問題は計算量的観点からは困難である．なお，準指数時間とは多項式時間と指数時間の中間であり

$$L_N[s,c]=\exp((c+o(1))(\log N)^s(\log\log N)^{1-s})$$

を用いて表されることが多い．$o(1)$ は漸近的（$N\to\infty$）に 0 に近づく関数である．ここで $o(1)$ を無視すると $s=0$ のときは

$$L_N[0,c]=(\log N)^c$$

となり，多項式，$s=1$ のときは

$$L_N[1,c]=N^c$$

となり，指数関数となる．準指数時間と呼ばれるのは計算量が $0<s<1$ を用い $L_N[s,c]$ で表される場合である．

素因数分解アルゴリズムには大きく分けて実行時間が主に N のみに依存する方法と，N の最小因子 p に依存する方法がある．既知の方法で漸近的に最も高速な方法は，それぞれ数体篩（ふるい）法[(用語)]とだ円曲線法であり，計算

青木和麻呂　正員　日本電信電話株式会社 NTT 情報流通プラットフォーム研究所
Kazumaro AOKI, Member (NTT Information Sharing Platform Laboratories, NIPPON TELEGRAPH AND TELEPHONE CORPORATION, Musashino-shi, 180-8585 Japan).
電子情報通信学会誌　Vol.91 No.6 pp.462-468 2008 年 6 月

量はそれぞれ $L_N[1/3, 1.9]$ と $L_p[1/2, 1.4]$ と評価されている．小さな因子がない場合は数体篩法の方が高速であり，小さな因子がある場合はだ円曲線法の方が高速となる．なお，素数判定は本小特集第2章の「素数とアルゴリズム」にあるように多項式時間の方法もあり素因数分解に比べ非常に容易である．したがって，因子の大きさに依存する素因数分解アルゴリズムの利用を考えると2番目に大きな因子が素因数分解の難しさを決めることになり，単純に N の大きさからだけでは素因数分解の難しさを決められない．例えば $N=2p$ で p が素数の場合は，Nを2で割り p の素数判定を行うだけで完全な素因数分解ができる．このようなことから1.では「小さな因子がない」と回りくどい書き方が必要であった．

3. 数体篩法

数体篩法は1990年にLenstraにより提案され，1990年代に完成した素因数分解アルゴリズムである(3)．数体篩法以前の素因数分解アルゴリズムは計算量が $L_N[1/2, c]$ となるものしか知られておらず，初めて $L_N[1/3, c]$ を達成したアルゴリズムとして注目された．提案当初は $N=b^n\pm1$ (b は小)といったような特別な形の合成数にしか適用できなかったが，改良が加えられ，形の制限のない合成数も分解可能となった．この提案当初の特別な形のみに適用可能な方法を特殊数体篩法(SNFS：Special Number Field Sieve)と呼び，分解対象の形の制限のない方法を一般数体篩法(GNFS：General Number Field Sieve)もしくは単に数体篩法(NFS：Number Field Sieve)と呼ぶ．

数体篩法が提案されたころは小さな因子のない素因数分解の世界記録は100けた程度であり，従来より知られている $L_N[1/2, c]$ の素因数分解法の方が高速であること，また数体篩法は非常に複雑であり実装が困難であることから理論的な興味でしかないと考えられてきた．しかし，世界記録が130けたになるころには数体篩法自体の細かな改良もあり，それ以上の大きさの数の分解には圧倒的に数体篩法の方が従来の素因数分解法より高速であると認識された．

用語解説

素因数分解　整数を素数の積に表すこと．例えば60の素因数分解は $2\times2\times3\times5$ である．

数体篩法　素因数分解方法の一つ．小さな因子がない合成数に対し既知の素因数分解方法では最も高速である．

PCクラスタ　複数のPCをネットワークでつないだ計算機群．従来のスーパコンピュータと異なり安価なPCを用いて構成することにより問題によっては費用対効果が高い計算能力を得ることができる．

メルセンヌ数　$M_n=2^n-1$ という形で表される数．M_n が素数であるためには n が素数である必要があるが十分ではない．

ここでは簡単に数体篩法のアルゴリズムを説明する(詳細は文献(3)を参照)．

数体篩法は以下のステップからなる．

① 多項式選択
② 篩
③ フィルタリング(filtering)
④ 線形代数
⑤ 平方根

特殊数体篩法は N が特別な形であることからほぼ自動的に多項式選択ができる．またその多項式が特殊な形をしているので計算量は $L_N[1/3, 1.5]$ となる．一方，一般数体篩法の計算量は $L_N[1/3, 1.9]$ となる．この計算量評価において律速段階となるのは篩と線形代数である．また，数学的にはフィルタリングは線形代数の一部であるが，実際に実装実験する場合は線形代数から独立して特別に実装した方がその後の線形代数処理をけた違いに高速に処理できるようになるのでここでは分けて記述した．

ここで説明したように特殊数体篩法と一般数体篩法の実装上の違いは多項式選択のみである．特殊数体篩法は確かに特定の形にしか対応していないが，特殊数体篩法による分解実験の知識はほとんどそのまま一般数体篩法に適用でき，どちらの結果も以後の数体篩法実装実験に役立つ経験となる．

以下にそれぞれのステップについて実装の観点から簡単に説明する．

3.1 多項式選択

整数を係数とする多項式の集合 $\mathbf{Z}[x]$ から互いに素で既約な多項式 $f(x)$ と $g(x)$ を選ぶ．$f(x)$ と $g(x)$ は mod N で共通根 M を持つ必要がある．つまり

$$f(M)\equiv g(M)\equiv0 \pmod N$$

である．多項式の次数の合計の最適な値は N が大きくなるにつれて大きくなる．N が200けたぐらいでは7ぐらいが最適である．多項式を選択する際にはなるべく係数の絶対値が小さなものを選ぶ必要がある．現在のところ $\deg f+\deg g>4$ については特殊な場合を除いて $\deg g=1$ とする方法しか知られていない．

多項式選択の素朴な方法としては $\deg g=1$，$\deg f=d$ とし，$M=[N^{1/(d+1)}]$ ととり，N を M 進展開

$$N=\sum_{i=0}^{d} c_iM^i$$

した係数を用いて

$$f(x) \leftarrow \sum_{i=0}^{d} c_i x^i, \quad g(x) \leftarrow x - M$$

とする M 進展開法がある．この素朴な方法による多項式選択でも $L_N[1/3, 1.9]$ の計算量を達成できる．実際の分解には $f(x)$ や $g(x)$ の係数を素朴な方法より小さく取れるなど，後続の計算の効率が上がるような多項式選択法が幾つか提案されている（例えば文献(4)）．

3.2 篩

篩は関係式収集ともいわれ，多項式選択で生成された f, g と a, $b \in \mathbf{Z}$ を用い，$|f(-a/b)(-b)^{\deg f}|$ と $|g(-a/b)(-b)^{\deg g}|$ が共に「小さな」素数のみの積でかける互いに素な a, b の組を多数見つけることが目的である．ここで関係式を見つける際にエラトステネスの篩と類似の方法を用いることにより効率的に (a, b) を集められるので篩と呼ばれる．ここで「小さな」素数の集合を因子基底(factor base)と呼ぶ．後のステップの技術的問題から f と g の因子基底に属する素数は別のものとして処理される．篩では (a, b) の組を因子基底の大きさと同程度の個数集める必要がある．したがって，因子基底を大きく取れば取るほど適当な (a, b) が必要とする関係を満たす可能性が上がるが，集めなければならない (a, b) の組も増大する．因子基底の大きさは N によって適切な大きさが選択される．

篩の計算量は数体篩法全体の計算量と同じオーダを必要とし，また，実際の実装実験でも最も計算量を要することから慎重に実装する必要がある．現在の多くの篩の実装では，因子基底を確保するぐらいの数倍のメモリ量を要するが世界記録を目指すレベルでも普通の PC で実行可能な大きさとなる．また，(a, b) の範囲ごとに独立に計算可能であることから，容易に分散計算が可能である．したがって，篩処理を高速に実現するには，高速な篩プログラムが利用可能との前提で，ある程度以上のメモリを実装した多数の PC をいかに集めるかが勝負となる．

3.3 フィルタリング

フィルタリングでは篩で得られた多数の (a, b) から $\mathbf{F}_2(=\{0, 1\})$ 要素の行列を生成する．数学的にはフィルタリングはガウスの消去法を実行しているにすぎないが，巨大なデータをいかに効率良く処理するかが課題となる．実際，1. で述べた 1,017 ビット合成数の分解では篩出力は gzip 圧縮をしても 1TByte 近くになっている．これだけ巨大なデータを長期間かけて生成していると，中にはファイルの破損や誤った結果も含まれたりすることなどがよくある．フィルタリングでは後続の処理のために，このような破損データを検知し，以後の計算が無駄にならないよう対応することも重要である．フィルタリングではデータフォーマットが正しいかどうかの確認や，メモリにすべては展開できない状況での巨大データの整合性確認などがプログラムのほとんどを占め，数学的な処理に至るまでが長く，プログラム作成の気力を維持するのが大変であった．

3.4 線形代数

線形代数の計算量も，篩と同様，数体篩法全体と同じオーダと評価されている．実際の分解実験においては，160 けたぐらいまでは篩の計算量に比べそれほど多くの計算量を要していなかったがそれ以上の記録を見ると，篩の計算量の 20 ～ 40% もの計算量を要している．この線形代数は $\mathbf{F}_2$ 上の巨大な疎行列で表される連立一次方程式の独立な非自明解を 10 個程度以上見つけるのが目的である．現在のところ，この線形代数を処理する疎結合の効率の良い分散計算法は知られていない．つまり，分解対象の合成数が大きくなると，単純に計算機を多数そろえたとしても計算可能かどうかは不明である．線形代数をどう処理するかは今後の大きな課題である．なお，1. に書いた分解実験では実質四つの疎結合の PC クラスタ(用語)を用いた計算を行った．これは多数のクラスタには拡張できるかどうかは自明ではないので今後の研究の進展が期待される．

3.5 平方根

多項式選択で選択された多項式で定義される数体上の巨大数の平方根をノルムの素因数分解情報を用いて計算する．数体篩法で唯一数論の知識を必要とする部分である．実行時間はさほどかからないが，無視できない長さのプログラムを要し，また，この処理を実行しないことには素因数分解ができないことから数論を知らない単なるプログラマーにとっては苦痛な作業となる．

4. 分 解 実 験

この章では 1. で述べた世界記録達成に至るまでに印象に残った素因数分解実験について述べる．

この章で述べる分解対象はすべて Cunningham project[5] から選んだ．Cunningham project は $b^n \pm 1$ といった形で，$b=2, 3, 5, 6, 7, 10, 11, 12$ に制限し n を大きくした数の素因数を追い求めている．Cunningham project では $b,n+$ や $b,n-$ という表記を用いそれぞれ b^n+1 と b^n-1 を表す．更に，

$$2^{4n+2}+1=(2^{2n+1}-2^{n+1}+1)(2^{2n+1}+2^{n+1}+1)$$

といった自明な分解があるものは，分解先をそれぞれ 2,1826 L や 2,1642 M など，LやM を付けて明記することもある．

Cunningham project で取り扱う数は先に書いたように $b^n \pm 1$ という形をしていることから，まさに特殊数体篩法の適用対象に思われる．しかし，$b^n \pm 1$ という形をしていたとしても，多数の因子が知られていて，残りの合成数が $b^n \pm 1$ に比べて圧倒的に小さい場合は一般数体篩法の方が高速に分解できる．

4.1 c164 in 2,1826L

この節では Cunningham project にある 2,1826L と呼ばれる数の 164 けたの未分解因子の分解[6]を試みたときに経験したことについて述べる．

c164 in 2,1826L は CRYPTREC[7]の支援を受け本格的に数体篩法の実装を立教大の木田教授及び富士通研の下山研究員と始めてから初めて世界記録更新を目指そうと試みた合成数である．2,1826L は $2^{913}-2^{457}+1$ で，既に 3，4，5，11，12，15，28，36 けたの素因子が知られており，残った未分解部分が 164 けた(545 ビット)の合成数であった．2,1826L 自体は Cunningham 数から選んだこともあり特殊数体篩法の適用可能な数である．特殊数体篩法は既知の小さな因子による恩恵は受けられないので一般数対篩法により残った 164 けたの因子を分解するのが最も高速と考えた．

この数に目標を絞って分解作業を開始したのは 2003 年 10 月下旬である．実は 2002 年 9 月末に ECC 2002 という会議で Bonn 大の Franke 教授を中心とするチームが RSA-576 という 576 ビットの合成数に対し分解実験を行っており既に篩作業の 30%が終わっているとの情報があった．また電子メールでの問合せに対し，2003 年 9 月末には既に篩が終わっているとの回答を得ていた．つまり，c164 が分解できたとしても，短い期間の世界記録であることは折り込み済みであり，時間との勝負であった．

具体的な篩作業は 10 月末から 12 月初旬までかけて行った．篩には電通大に設置した 64 台，立教大の約 40 台，NTT の 30 台を中心とし，その他数台の PC を用いた．CPU は PentiumⅡ 400MHz から Pentium 4 3.06GHz の様々なものを用いた．要した計算量はおよそ 7 Pentium 4 2.53GHz･年だった．続いてのフィルタリング作業は 11 月末から 12 月初旬にかけ約 3 日で行った．篩結果からおよそ 750 万行のほぼ正方の巨大疎行列を構成した．なお，この作業中である 11 月 30 日から 12 月 4 日にかけて開催された Asiacrypt 2003 という暗号関連の国際会議の途中で，12 月 12 日に素因数分解関連のワークショップを 12 月 12 日にユトレヒトで開催するとの告知があった．この情報を聞き，RSA-576 の分解発表に合わせてのワークショップではないかとの想像から残された時間はほとんどないのではないかとの焦りが共同研究者の間で走った．とはいえ，この情報を得たからといって何か分解を加速させるような魔法があるわけではない．人的作業を可能な限り無駄なく効率良く進めるしかない．電通大設置の PC で得られた行列データを ADSL 上りで転送すると時間がかかるということで外付けハードディスクに保存し人間が立教大まで運ぶことにより高速なデータ転送を実現するなど精いっぱいの努力のもと 12 月 3 日から線形代数の作業に入った．

一方，競争相手の RSA-576 の分解については 12 月 3 日 14 時ごろ(GMT＋1)に USENET の sci.crypt ニュースグループに結果が報告された．その後，12 月 4 日 16 時ごろ(GMT-8)，電子メールで素因数分解関連研究者に告知された．筆者自身は 12 月 5 日 16 時ごろ(GMT ＋9)に sci.crypt の投稿内容を見てこの事実に気が付いた．報告は短いものであったが確かに素因子が記録されており，何度検算しても正しいものであった．我々はいまだ線形代数の計算中であり，がっくりとしたが，せっかく続けたこともあり後続の作業を進めた．

c164 分解の線形代数は立教大に設置されたギガビットイーサネットで接続した 16 台の PC を用い行っていた．12 月 15 日に計算が終了した．この後の平方根計算は数時間もあれば計算可能なはずであったが，あいにくまだプログラムが完成していなかった．既に世界記録が取れないことが確定していたので，余り急がずにプログラムを完成させ，実際に分解できたのは 12 月 18 日．世界記録に 2 週間ほど及ばなかった．確かに分解時点では世界第 2 位の記録ではあったが，非常に悔しいものがあった．

4.2 2,1642M

4.1 で c164 in 2,1826L の分解を述べた．これで終わるのも悔しいので何か別の世界記録をねらえないものかと考えた．RSA-576 の分解では，ずっと連続して計算をしていたわけではないだろうがおよそ 3 年もかけての記録達成であるので，再び一般数体篩法の世界記録をねらうためには相当量の計算量が必要と考えられ現実的とは思えなかった．そこで暗号学的な意味では印象が薄くはなるが，特殊数体篩法での世界記録更新を考えた．その対象として選んだのは Cunningham project で因子を募集している 2,1642M[8]である．

2,1642M は $2^{821}+2^{411}+1$ であり，248 けた(822 ビット)の合成数である．分解作業開始前に一つも因子が知られていなかった．この分解作業を始めるにあたって気にすべき点は二つあった．一つは，CRYPTREC の支援により電通大に設置された 64 台の PC は 2004 年 2 月末までの 2.5 か月ほどしか使えないこと．もう一つは，NFSNET と呼ばれる世界中からボランティアベースで計算資源提供者を求めて数体篩法により素因数分解を行っているチームとの競合であった．

NFSNET は，その性格上，分解対象及び進行状況を公開しており，競合相手の進行状況は容易に把握できた．

当時 NFSNET は 2,811－(＝$2^{811}-1$) と呼ばれる合成数の分解を進めており篩作業中であった．2003 年 10 月から篩作業を進めており，12 月には NFSNET も篩作業に参加していた RSA-576 の分解発表があり，一挙に知名度が上がったことから参加者が増え，それまでの倍程度の計算能力を保持していた．ただ，彼らの分解対象である 2,811－は既に 6 けたの因子が知られていたことから残った未分解合成数は 239 けた(793ビット)であり，当時の特殊数体篩法による世界記録である 244 けた(809ビット)に及んでいなかった．とはいえ，先に述べたように特殊数体篩法では既知の因子は分解に役立たないので特殊数体篩法としての難しさは 245 けた(811 ビット)相当ということで，そういう意味では世界記録になり得る分解対象設定であった．一方，我々の分解対象は 248 けた(822 ビット)ということで，仮に NFSNET の分解が先に終了したとしても世界記録になる問題設定であった．

2,1642M の篩処理は，c164 in 2,1826L 分解の線形代数作業の途中である 2003 年 12 月中旬から，線形代数に用いない計算機を用いて始めた．計算には CRYPTREC の支援により電通大に設置された 64 台の PC 及び，立教大設置の約 40 台の PC を用いた．今回の計算では電通大設置計算機については 2 月末までの期限があるとはいえ余裕があったので，それ以上の計算機はかき集めなかった．なお NFSNET の分解作業であるが，篩が一向に進まずいつまでたっても線形代数に進まないことからこのまま先に我々の結果の発表となるとだまし討ちのようなので，NFSNET へ我々の計算対象合成数の大きさ及び進行状況を電子メールで伝えた．その結果，NFSNET の計算能力はそれまでのおよそ倍になり，篩処理が加速された．

我々の篩作業は 2004 年 2 月初旬に終わった．要した計算量は 8.2 Pentium 4 2.53GHz･年であった．後続のフィルタリングは 2 月初旬に行い 770 万行の疎行列を構成した．線形代数は 2 月 11 日から 2 月 24 日にかけての 14 日間行った．これは電通大設置の 100 baseT 接続の 16 台の PC を用いて行った．その後，平方根の計算を 2 月 25 日に行ったところなぜか分解に成功しなかった．原因を解析したところ，篩プログラムにたまに起こるバグがあり，ごみデータが篩結果に混入していた．結果，ごみデータを篩結果から取り除きフィルタリングからやり直すことになった．が，電通大設置の 64 台もの PC の利用期限が 2 月いっぱいだったことから後続の計算の計算機確保が課題となった．この時点で NFSNET へ我々の分解がまだ 2 週間以上かかりそうな旨連絡したところ，彼らの篩能力がそれまでの 1.5 倍に上がった．

さて，2,1642M の分解作業であるが，フィルタリング作業のやり直しを 2 月下旬に行い 740 万行の疎行列が生成できた．前回のフィルタリングより小さな行列となったのは若干，前回のフィルタリング入力に間に合わなかった篩出力があり，それを利用したからであった．後続の線形代数は立教大設置のギガビットイーサネットで接続された 16 台の PC を用いて行った．この PC は電通大設置のときのものと異なり，PC の仕様がまちまちであり使いづらいものであった．とはいえ，3 月 1 日に開始し，途中の操作ミスやハードウェアトラブルもあったとはいえ 3 月 10 日までの延べ 10 日で計算終了した．が，しかし行列のランクが行数よりも多くなるというあり得ない結果が得られた．

ぼう然としながら，途中のハードウェアトラブルからメモリが怪しいとにらみ memtest 86＋により半日ほどテストし，幾つかの問題があった PC の構成を換え再び線形代数の処理に入った．3 月 16 日に開始し 3 月 25 日までの延べ 10 日間で計算終了したが，またランクが行数よりも多いという結果になった．このときには NFSNET の篩は終わっていた．

2,1642M の分解作業の途中から，素因数分解専用として NTT に 32(＋α)台構成の PC クラスタの設置作業が始まった．立教大 PC での結果がうまく得られるとの確信が得られなかったことから設置作業中の 16 台の PC に割り込んで線形代数を 3 月 19 日に開始した．最初の数時間の計算から予測される計算終了は 3 月 29 日(月)であった．3 月 26 日(金)まで順調に計算が進んでいることを確認して退社し，月曜日に喜び勇んで出社して console を確認したところ，異常終了している．調べたところ 3 月 27 日(土)に PC の 1 台がディスククラッシュで停止していた．これだけの台数の PC があれば初期不良としては当然あり得ることであるが，年度内に計算が終わらなかったことは残念であった．

気を取り直し 3 月 29 日に計算を再開した．線形代数のプログラムは，このような障害に備えて数時間ごとに中間結果をディスクに保存し，その結果から再開できるようになっていたので，4 月 2 日(金)の早朝には結果が出る見込みであった．結果をちょっとでも早く見たいと思い 4 月 2 日に，ほぼ初電で出社してみたところ，半数近くの PC が停止していた．確認したところ，4 月 2 日(金)の午前 1：20 ごろ落雷があり瞬電したようだった．

最初のプログラムのバグによる線形代数計算の無駄は自分の問題であるので仕方ないとしても，その後の機器不良や天災による計算のやり直しは相当こたえた．天候にまで嫌われたのかとがっくり．とはいえ，もう一歩なので気を持ち直して 4 月 3 日に計算を再開し，当日深夜には計算終了した．待ち望んだ正常終了．延べ 15 日，実質 9.5 日の計算であった．

後続の平方根計算は立教大設置の 8 台の PC を並列に動作させて計算した．4 時間後，最初の答えが出たが，数体篩法のアルゴリズム上，あり得ない解が出力された．

またもがっくりしながら，線形代数出力の別の解では

うまくいく可能性にかけて，また平方根計算を開始．その間に原因解析．平方根計算の入力パラメータに誤りがあることに気が付き最初に得られた平方根を調整して計算したところ…ようやく分解に成功．喜び勇んで，すぐに結果を素因数分解に興味ある関係者に報告した．

なお，NFSNET の 2,811－の分解も線形代数のトラブルがあり，実際に分解できたのはその 2 か月後であった．

4.3 c307 in 2,1039－

2,1039－は，Cunningham 数でもあり，$2^p - 1$ と表せるいわゆるメルセンヌ数(用語) M1039 である．分解作業開始時点で既に 7 けたの因子が知られており，残りの 307 けたが未分解合成数 c307 であった．この c307 の分解については，十分な経験と余裕を持って臨み，また，これまで競争となった Bonn 大のチーム及び数体篩法の開発者の一人である EPFL の Lenstra 教授とチームを組んだことから技術的な問題はほとんど発生せず順調に計算できた．

実際の分解実験は，かなり「やった」だけの作業に近いものがあったが，技術的には

- 線形代数をネットワーク的に離れた複数の PC クラスタを用いたこと，
- また一つの PC クラスタで複数の計算を並列に走らせたこと

が新しい．これまで線形代数は一箇所に密結合の計算資源を集中させる必要があったが，今回の技術の進展が更に進むことにより，将来の分解実験では線形代数が律速になるとの予想を覆すことを期待したい．

c307 の分解は，1999 年に達成された 512 ビットの一般数体篩法の分解の後の一里塚として考えられていた kilo-bit SNFS，つまり 1,000 ビットを超える特殊数体篩法による分解を達成したいとの思いから始まった．これは単にやればよいというものではなく，実は，分解対象の設定が自明ではない．これまでの分解でも考慮しなければならない問題であったが，この kilo-bit SNFS では特に苦労したので紹介したい．

素因数分解の記録更新のためには分解対象の選び方が重要である．適当に選ぶと，実は「先に因子を作ってから掛け算をして作っただけではないか」との疑いを晴らすことができず，またそうでないとしても小さな因子があり，実はだ円曲線法など別の簡単な方法で分解した後，つじつまを合わせるだけのデータをそろえただけではないかとの疑念を抱かせる．このようなことから，通常，分解候補は RSA challenge number といった第三者により作られたことが明らかな数か，Cunningham 数や分割数など「自然に」作られた数が選ばれる．

さて，kilo-bit SNFS を実行するためには，特殊数体篩法の対象となる数を選ぶ必要があった．これは Cunningham 数から選べばよい．ただし，小さな因子を持っていると先に述べた疑念がわくので，十分にだ円曲線法により小さな因子を排除しておく必要がある．また，特殊数体篩法は未分解合成数の大きさではなく $b^n \pm 1$ の大きさにより計算量がほぼ決まるので余り大きな $b^n \pm 1$ を選ぶのは効率的ではない．といった条件のもと，Cunningham 数から候補の数を幾つか選び出した．そのうち，10,371－，2,2062M，2,2038M はだ円曲線法による若干の計算で 50 けた前後の因子が見つかり条件から外れた．残った候補のうち R311 と呼ばれる，10 進数表記で 1 が 311 個並ぶ数，Cunningham 数としては 10,311－と表される数が業界としても kilo-bit SNFS としての最有力候補として考えられており，これについてだ円曲線法による分解を試みた．

特殊数体篩法で分解を始める前には，だ円曲線法によりおよそ 2/9 のけた数の因子が見つかるぐらいの計算量を要するのが適切との経験則がある．それによると R311 は 311 けたの数なので，69(≈311×2/9)けた程度の因子が見つかる計算量をかける必要があった．また，当時のだ円曲線法による見つかった因子の世界記録は 66 けたではあったが，利用可能計算資源などの制約から 65 けた程度の因子を見つけるのに最適なパラメータで計算を開始した．

事前に若干の予備計算を行い 2005 年 6 月から本格的にだ円曲線法により R311 の分解作業を開始した．計算は 40 台ほどの素因数分解専用に用いることができる PC に加え NTT 社内の PC の空き CPU 時間の供出も呼び掛け 100 台程度が参加し，合計 140 台ほどが用いられた．計算開始から，およそ 3 か月後の 8 月末に 64 けたの因子を発見．因子が見つからないことを期待して始めた計算なのでがっかり．もちろん，このまま因子を見つけられないまま数体篩法の計算に進んでいると，他チームがだ円曲線法により先に分解したり，そうでないにしても，だ円曲線法で分解されたとの疑念が残るような分解となったので，そういう意味では好運であったが準備期間を合わせると 3 か月以上の計算が，ある意味無駄になったのはかなりこたえた．せめて，だ円曲線法による世界記録になっていれば，まだ慰められたとは思うが，残念ながら当時としてはだ円曲線法により見つかった因子の大きさでは世界第 2 位の記録でしかなかった．

次の候補としては M1061(＝2^{1061}－1)が業界としては最有力候補であった．M1061 は因子が一つも知られておらず，kilo-bit を超える条件も満たしているが 61 ビットも超えていて若干大きく感じた．更に業界的に有力候補ということは他チームとの競合になり得ることからも避けたかった．そこで，特殊数体篩法としての難しさとしては 2^{10}(＝1,024)を超え，未分解合成数の大きさとして 10^3 を超えるものを当たったところ M1039 (＝ 2^{1039} －

1)があった．M1039 自体は 2^{10} も超えているが，未分解因子は 1,017 ビットであり，2^{10} ビットを下回っているので，若干印象が悪い．しかし，仮に M1039 が分解された後に，未分解因子としても 2^{10} ビットを超えているものが分解されたときに M1039 の評価はどうなるか想像を巡らせたところ，M1039 分解の評価の方が高いように思われたので，M1039 を候補に選択した．

この後の分解作業は順調に進んだ．だ円曲線法により小さな因子がないことの確認にはおよそ 128 Opteron 2.2GHz･年，篩に 95 Pentium D 3.0GHz･年，線形代数に 36 Pentium D 3.0GHz･年を要し，分解に成功した．

5. メモリ使用量について

数体篩法の計算量は通常ビット演算量，つまり時間でまずは評価されるがメモリも問題にしなければならない．特に線形代数の処理では行列を記憶するメモリをどのように確保するかが問題となる．4.3 で紹介した M1039 の分解では，$\mathbf{F}_2$ 上の疎行列とはいえ 95 億もの非ゼロ要素があった．行列の大きさはおよそ 6,672 万行のほぼ正方行列であったので，行位置だけ保存するとしても 26 ビット必要であるので，合計 29GByte 弱は最低でも必要となる．実際にはその他のデータも必要なので更に大きなメモリが必要である．最近のハードディスクは高速になったとはいえ，まだまだメモリの方が数けた高速であるので，できれば実メモリに展開したい，ということから，行列計算のための PC クラスタの実メモリ合計は最低でも 29 GByte は必要ということになる．

現在知られている線形代数の計算法では PC クラスタのノード数に対し漸近的にはルート倍，つまりノード数が k 倍になっても計算時間は $1/\sqrt{k}$ 倍しか短くならない．行列計算のメモリ量が必要だからといって単純にノード数を増やしても余り効率が良くないことに注意しなければならない．

漸近的な観点からはどうなるのであろうか．通常の一般数体篩法の計算量は $L_N[1/3, (64/9)^{1/3}]$ で，必要メモリ量は $L_N[1/3, (8/9)^{1/3}]$ と評価されている．このとき実行時間を増やして，代わりに必要メモリ量を減らすことを考えてみる．なお，$L_N[s, c]$ の c は，同じ s に対してのみ大小の意味を持つので以下では c を省略して $L_N[s]$ と表記する．このとき，使用メモリ量を $L_N[s]$ $(0 \leq s \leq 1/3)$ と置き数体篩法の計算量の評価法をたどると計算時間は $L_N[1-2s]$ となる．確かに，計算時間を増やせば必要メモリ量は減らせるが，減らしたメモリ量以上に実行時間が必要となることが分かる．今後の半導体技術など計算能力向上が計算能力とメモリ量についてどのように伸びていくかでこの評価のどちらが足を引っ張るかについては注意深く検討する必要がある．

6. おわりに

本稿では，筆者がかかわった巨大数の幾つかの素因数分解実験中に経験したことについて紹介した．確かに数体篩法は，巨大なアルゴリズムであり，すぐには実装・実験できるようなものではなかったが，近年では GGNFS や msieve などの公開実装が現れている．これらを世界記録をねらうような大規模計算に使うにはまだ不十分な点があるが，アルゴリズム全体の動きを理解するには十分である．また M1039 の分解ではおよそ 6,672 万行の疎行列で定義される $\mathbf{F}_2$ 上の連立方程式の非自明解を求めることになり，苦労したが，実は大規模計算の観点からは既に解決されている問題であり簡単に計算できたりするのではないかとの疑念もぬぐえない．今後，様々な分野の人がアルゴリズムの改良や実装方法の工夫に参加し，記録の進展が加速されることを期待する．

文　献

（1） 日本電信電話株式会社，“暗号方式の安全性検証に有効とされる「素因数分解」において世界記録を更新，” NTT ニュースリリース， 2007.
http://www.ntt.co.jp/news/news07/0705/070521a.html

（2） K. Aoki, J. Franke, T. Kleinjung, A.K. Lenstra, and D.A. Osvik, “A kilobit special number field sieve factorization,” Advances in Cryptology—ASIACRYPT 2007, Lect. Notes Comput. Sci., vol.4833, pp.1-12, Springer-Verlag, Berlin, Heidelberg, New York, 2007.

（3） The Development of the Number Field Sieve, A.K. Lenstra and H.W. Lenstra, Jr., Eds., Lecture Notes Math., vol. 1554, Springer-Verlag, Berlin, Heidelberg, 1993.

（4） T. Kleinjung,“On polynomial selection for the general number field sieve,” Math. Comput., vol.75, no.256, pp.2037-2047, 2006.

（5） http://www.cerias.purdue.edu/homes/ssw/cun/

（6） K. Aoki, Y. Kida, T. Shimoyama, Y. Sonoda, and H. Ueda, “GNFS164,” 2003.
http://www.rkmath.rikkyo.ac.jp/~kida/gnfs164e.htm

（7） http://www.cryptrec.jp/

（8） K. Aoki, Y. Kida, T. Shimoyama, Y. Sonoda, and H. Ueda, “SNFS248,” 2004.
http://www.rkmath.rikkyo.ac.jp/~kida/snfs248e.htm

（平成 20 年 1 月 29 日受付　平成 20 年 3 月 19 日最終受付）

青木　和麻呂（あおき　かずまろ）（正員）

平 5 早大・理工・数学卒．平 7 同大学院修士課程了．同年 NTT 入社．以来，暗号安全性の研究に従事．うち平 3 年度に通信放送機構に出向し CRYPTREC 事務局運営．博士(理学)．SCIS’95・’96 論文賞，平 9 年度本会学術奨励賞，平 17 年度情報処理学会業績賞各受賞．