RSA 暗号解読向け 0-1 行列の疎行列直接解法

後　保範（早稲田大学）

Direct sparse solver of 0-1 matrix for RSA decryption.

Yasunori Ushiro (Waseda University)

概要：RSA 暗号解読では、大規模な 0-1 疎行列の線形計算が必要である。現在はブロックランチョス法による反復解法が使用されている。暗号解読全体の高速化方式を検討すると、従属解を多数必要な場合がある。その場合は反復解法より直接解法が有利である。そのため、大規模な 0-1 行列の疎行列直接解法の計算方法を検討した。

1. はじめに

RSA 暗号は現在の情報社会において欠かせない技術である。また、現在使用している 1024 ビットの RSA 暗号が安全性の問題で近いうちに使えなくなるという「2010 年問題」が懸念されている。RSA 暗号は多数桁の因数分解の困難性を利用している。現在、知られている手法では、1024 ビットの RSA 暗号の解読はスーパーコンで数年かかると予想されているが、計算機の高速化により数年先には解読される可能性が高くなっている。2010 年 1 月に NTT 他 4 カ国の共同により、Opteron(2.2GHz)換算で約 1700 年・コアの演算量を使用して RSA-768(10 進 232 桁)が解読された。RSA 暗号の解読計算はほとんど PC クラスタで行われている。

2. RSA 暗号

インターネットで情報を安全に送るために、暗号化が用いられている。暗号化の方式には、共通鍵方式と公開鍵方式があり、両者の利点を用いたハイブリッド方式が使用されている。共通鍵方式は処理が高速であるが、暗号鍵と復号鍵が同じため鍵が安全に送れない。一方、公開鍵方式は暗号鍵と復号鍵が異なるため、復号鍵は送付しなくてもよいが、共通鍵方式より処理速度が遅い。そのため、公開鍵方式を使用して、共通鍵方式の鍵を送付し、情報本体は共通鍵方式で暗号化して送信されている。

現在使用している公開鍵方式は、1978 年に R. L. Rivest, A. Shamir, L. M. Adelman の 3 名により発見され、RSA 暗号と呼ばれている。これは多数桁の因数分解が非常に困難なことを利用している。RSA 暗号は 2 つの大きな素数 P,Q 及び e をランダムに作成し、N=P×Q、F=(P-1)×(Q-1)を計算し、$D=e^{-1} \pmod F$を求め、N と e を暗号鍵として使用し、N と D を復号鍵に使用する。N と e は秘密にする必要はなく、公開鍵として自由に送信できる。一方、D は復号するための秘密鍵で、この鍵は送信しない。送付したい情報を N 以下の数値の列に分けて、それぞれを暗号化して送信し、受け取り側で復号して元に戻す。分けられて N 以下の数値を M とし、暗号文 C を $C=M^e \pmod N$で変換し送付する。受信は暗号文 C を $M=C^D \pmod N$の変換で元の数値 M に復元する。復号は 2 つの素数に P,Q にオイラーの定理、$M^{(P-1)(Q-1)}=1 \pmod N$を適用することで得られる。現在 N は 1024 ビット(10 進 309 桁)以上の値が使用され、P,Q はビット

数が少しだけ異なるランダムな素数が使用される。

3. RSA 暗号の解読方法

RSA 暗号を解読するには、合成数 N を 2 つの素数 P と Q に因数分解する必要がある。現在、高速に因数分解する方法としてふるい法が知られている。10 進 100 桁までは複数 2 次多項式ふるい法(MPQS)が、それ以上は一般数体ふるい法(GNFS)が効率的と言われている。GNFS も多項式 f(x)単独から、1 次式 g(x)と多項式 f(x)の組合せに改良されてきている。GNFS(一般数体ふるい法)による RSA 暗号の解読手順を下記に示す。分解する合成数を N とする。

(a) g(M)=0 (mod N), f(M)=0 (mod N)を満たす 1 次式 g(x)と多項式 f(x)の探査

(b) ふるい処理(素数などの基底の選定、ふるいによるデータ収集と行列作成)

(c) 剰余 2 を施した 0-1 行列の線形方程式の解の計算

(d) 多項式(f(x))を法とする代数平方根の計算

(e) $a^2-b^2=0$ (mod N)を構成し、N を因数分解

RSA 暗号解読における計算量の例を表 1 に示す。これは GNFS で RSA-768(10 進 232 桁)を解読した例で、0-1 線形計算の次元数は 192,796,550×192,795,550 である。

表 1. RSA-768(10 進 232 桁)の計算量

処理項目	計算量(台数・年)	構成比率(%)
利用関数の探査	20	1
ふるい処理	1500	90
0-1 行列の線形計算	155	9
代数平方根の計算	0.1 以下	0
その他	0.1 以下	0

注)計算量は ADM64(2.2Gh)の 1 コアで 1 年の計算が 1 台数・年

4. 代数的数の平方根の計算

GNFS で 2 次式以上の多項式には代数的平方根の計算が必要である。計算時間は、ふるい処理や 0-1 行列線形計算に比較して圧倒的に少ない。また、2 次式と多項式(5,6,7 次式)では代数平方根の性質及び計算法が大きく異なる。2 次式は直接計算できるが、代数平方根 a(x)は $a(x)^2=R\cdot b(x)$ (mod f(x))の形となる。ここで、R は小さい素数の積。一方、5,6,7 次式では計算に多くの工夫が必要であるが、R は 1 になる利点がある。その原因は 2 次式では 1 次式の乗算結果の mod f(x)が整数になるケースが多く、平方剰余の追加で R が 1 にできない。R が 1 にできないと、0-1 行列の線形計算で、多くの解を求める必要が発生する。1 次式の剰余(mod sx+t)は整数なので、1 次式の代数平方根は、整数の平方根になる。0-1 行列計算は、各基底のベキを 2 にする組合せ(従属解)を求めているので、平方根は各基底のベキを 2 で割ればよい。GNFS は 2 次式と多項式(5,6,7 次式)の組合せでも良いが、現在利用されていないのは 0-1 行列計算で多数の解を必要とするためである。

5. 0-1 行列の線形計算

(1) 0-1 行列線形計算の概要

0-1 行列の線形計算は、ふるいで得られたデータから各基底（素数、イデアル）のベキを偶数になるような組合せを選ぶ目的で行う。計算方法は反復法と直接法があり、現在は全て反復法が使用されている。データ数を n、基底数(少量の平方剰余を含む)を m とすると、n は m より少し多く求める。n×m の行列を A とする。反復法では少し横長の $A^Tx=0$ のゼロでない解 x を求める。反復法は通常 64 個の解 x をブロックランチョス法で計算する。直接法は行列 A にガウスの消去法を適用する。ガウス消去は疎行列の形のまま行う。n が 1 億程度のとき A の 1 行当たりの非ゼロ数は数百である。このため大規模な A の非ゼロ率は $1/10^5$ 以下である。

(2) 疎行列直接解法が必要な理由

単独の多項式や 1 次式と多項式(5,6,7 次式,f(x))による GNFS では、代数平方根 $a(x)^2=b(x) \pmod{f(x)}$が存在し、確率 1/2 で因数分解できる。このため、同時に 64 個の解を求める反復解法で十分である。一方、2 次式と多項式(5,6,7 次式,f(x))による GNFS では、代数平方根は $a(x)^2=R\cdot b(x) \pmod{f(x)}$の形になり、R の部分を平方にするには、多数の線形計算の解が必要になる。また、GNFS のふるい考えると、因数分解と自明解(N=1・N と分解)の確率を 1/2 に保つことが、より高速なアルゴリズム開発の妨げとなる場合がある。因数分解の確率が下がると、確実に因数分解するには、多数の線形計算の解が必要になる。そのため、1 回で 64 個の解ではなく、数千、数万の解を求める必要性が発生する。反復解法で 64 個の解を求めているのは、0-1 の解は 64 ビット整数に 64 個保存でき、計算も 1 命令(XOR)で 0-1 要素を 64 個同時計算ができるためである。これを増やすと、計算量は増加分(64 個単位)だけ増加し、メモリー容量も増える。一方、疎行列直接解法は、ガウス消去法の過程で、非ゼロ率は増加するが、求める解の数には計算量もメモリー量もほとんど依存しない。例えば、m(基底数)が 1 億なら、n(データ数)を 1 億 10 万(ほんの少しだけ縦長行列)にして、ガウス消去法を適用すると、10 万以上の解が計算できる。1 億の行列では縦横比を 1/1000 だけ長くすれば良い。

6. 直接解法の概要

n×m の 0-1 行列 A にガウス消去法を適用する。ここで、n>m である。0-1 行列なので消去は、加算して mod 2 処理し、値は 0 か 1 となる。通常の消去と異なり、消去は消去行以下全てに対して行う。消去列が全てゼロなら、消去列を 1 ずらす。n>m なので、消去が終了しても消去行の下に、行が残る。残った行数だけ従属解が求まる。枢軸は必ず 1 に保つために軸交換が発生する。枢軸と交換情報を 2 本(長さ n と m)に残す。A は各行毎に、1 の列番号だけ持たせた、疎行列の形で消去処理を行う。疎行列の要素の数(1 の数)は消去により増加する。

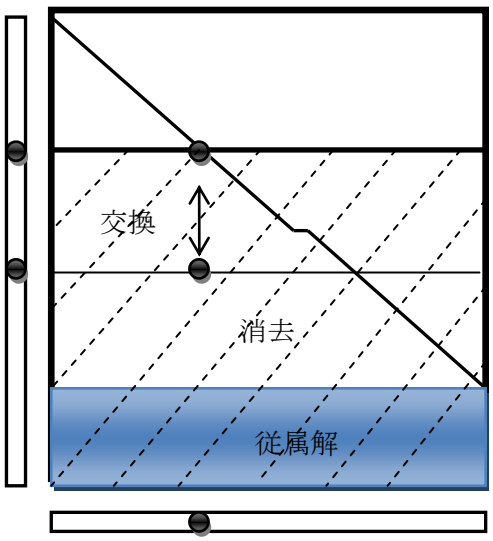

図 1. ガウス消去法

7. 0-1 疎行列直接解法の工夫点

疎行列直接解法は、行と列の番号付けの方法によりメモリー量と計算量が左右される。番号変換の方法については別に発表する。ここでは、番号変換を行う基礎の疎行列ガウス消去の考え方を示す。0-1 行列のため、密行列では 64 要素が 1 命令(XOR)で計算できる。一方、疎行列で扱うと 1 命令で 1 要素の計算となる。そのため、1 の要素の比率が 1%以上なら密行列処理の方が、高速になる。行列は非常に疎で、数億次元なら 1 要素の比率は 0.001%程度となる。また、行方向は均等に近い分布で、列方向は 2 から、m/2 と行当たりの 1 要素の数が大きく変化する。

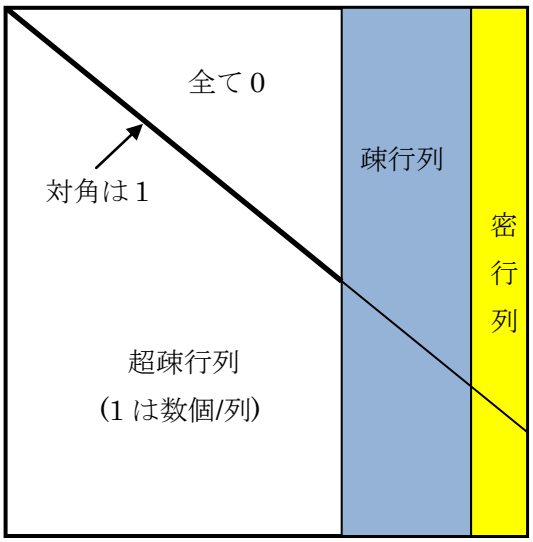

図 2.　0-1 疎行列のガウス消去の形

このため、一部の列は密行列として扱う。また多くの列は 1 要素が数個から十数個なため、対角に 1 要素を並べ、それらの列の上三角部分は、全てゼロ要素になるようにする。3 個の部分に分けて、処理する場合の考え方を図 2 に示す。ガウス消去計算は超疎行列の 1 要素を対角要素で全て消去する。次に、この消去変換を疎行列の部分に対して行う。その次に疎行列部分のガウス消去計算を行う。密行列部分は、超疎行列による消去変換、疎行列による消去変換を行ってから、密行列部分のガウス消去計算を行う。詳細処理は発表時に示す。

8. 参考文献

[1] 木田 祐司, “数体ふるい法による素因数分解”, 2003 年, http://www.rkmath.rikkyo.ac.jp/~kida/nfs_intro.pdf

[2] 小国 力編, 行列計算ソフトウェア「疎行列用オーダリング法」“, 1991 年, 丸善

[3] Y. Ushiro, H. Hasegawa, “Acceleration of the Processing of Linear Equations in Characteristic 2 for RSA Decryption”, Annual Report of Earth Simulator Center, 165-170, 2013 年

[4] Y. Ushiro, H. Hasegawa, “Acceleration of the Processing of Linear Equations in Characteristic 2 for RSA Decryption”, Annual Report of Earth Simulator Center, 165-170, 2012 年

[4] Y. Ushiro, H. Hasegawa, “RSA Decryption using Earth Simulator”, Annual Report of Earth Simulator Center, 167-171, 2011 年

[6] 後保範,”ベクトル計算機による RSA 暗号ふるいの高速化”, 京大数理解析研講究録 1733, 101-117, 2011 年 3 月