多倍長平方数の高速探査法

後　保範
（神奈川大学）

はじめに

- DRM（分割剰余法）発見：20年前 3-Gp

  π計算：AGM（算術幾何平均）

  AGM → DRM + ラマヌジャン公式

- FSS（高速平方数探査）発見：今年 単独

  RSA暗号解読：GNFS（一般数体ふるい法）

  GNFS → FSS + 平方差法
  （研究中）

何万回/s可能？(4GhのPC)

- Yが平方数かのチェック

$$Y = (M + x)^2 - N \quad = h^2$$

N: 2048ビット(618桁)の数

M: Nの平方根を切り上げた整数

x : 0,1,2,…と増加させる

平方数でないYの値は不要

Nの値の一例(2048ビット)

N=

25677387604979174745650113439241870319948692169987586987064217095280862955992909072300653168631621794286691440902481665331169514458834441618096858537080112176897909414971637999785470540208172478953628569638268116333771307187785435113239691953227878985856787628817148629285215167591341060629360680706276666777410080181671792674052335016894122101678401287903067557480695012549698890769654847894705525022833939558199158970554038822782042239650321691935109625751322550734744674624989223688999884343191185936615580541795939979086396690437969864999069285791953748131240075717217658164159648096674467244073982670709358392 9

何万回/sか単純推定

- Yの値の計算

4Ghz / (32^2 / 2) = 8M/s = 800万回/s

乗算数(int)　　2演算/hz

700万
(実測)

- Yが平方数か判定（平方根を求めて）

平方根がY計算の4倍 = 200万回/s

200万
(実測)

平方数判定(何回/s)

$Y=(M+x)^2-N$の平方数判定

60兆　回/s
(実測)

3千万倍

Nは2048ビット
200万回/s(平方根計算)

工夫(コロンブスの卵)

- 平方剰余を利用
- 既知 ＝ 整数の平方数判定
  多倍長関数(gmp)のmpz_perfect_square_p
- 工夫 ＝ 希少を探す → 多項式に適用
  素数40個で平方数候補を1/兆に絞る
  数千京個の探査でも多項式計算は約200回
  64,27,25,49の剰余は更に効果大

平方数探査(原理)

- 平方剰余

  整数P,xで$u^2=x \pmod P$の整数解uが存在?

  Pで、uが存在 → xは平方剰余(値=1)

  存在せず → xは平方非剰余(値=0)

- Pで0ならxは平方数でない → 多項式

- P,x<Pで0-1テーブル化 ($G_P[x]$) → $F_P[x]$

  $G_8[x]$ = [1,1,0,0,1,0,0,0]

  $G_{15}[x]$ = [1,1,0,0,1,0,1,0,0,1,1,0,0,0,0]

平方数探査 (具体例)

- $f(x) = (M+x)^2 - N$

  N = 640463357731254796 3, M=2530737754

  x <240で$(M+x)^2=T^2$ (mod N)となるx,Tを探す。
- P = 8,15,7,11,13,17とする。

f(0)～f(16)まで17個のf(x)を計算

各Pでf(x) (mod P)をP個計算

各Pで平方判定テーブル$F_P[x]$を作成

注目は
数と
繰り返し

$F_P[x]$のセット方法

- P=8　(x=0,1,…,7)

  $G_8[x]$ = [1,**1**,0,0,1,0,0,0]

  f(x) (mod P) = [**1**,6,5,6,**1**,6,5,6]

  $F_8[x]$ = [**1** 0 0 0 **1** 0 0 0]

Pで固定

f(x)依存

- P=15 (x=0,1,…,14)

  $G_{15}[x]$ = [1,1,0,0,1,0,**1**,0,0,1,1,0,0,0,0]

  f(x) (mod P) = [3,12,8,**6**,**6**,8,12,3,11,**6**,3,2,3,**6**,11]

  $F_{15}[x]$ = [0,0,0,**1**,**1**,0,0,0,0,**1**,0,0,0,**1**,0]

平方数探査 ($F_p(x)$)

$F_8[x]$ = [1,0,0,0,1,0,0,0]

$F_{15}[x]$ = [0,0,0,1,1,0,0,0,0,1,0,0,0,1,0]

$F_7[x]$ = [1,0,1,0,1,1,0]

$F_{11}[x]$ = [0,1,0,1,1,0,1,1,0,1,0]

$F_{13}[x]$ = [0,1,0,0,1,0,1,1,1,1,0,1,0]

$F_{17}[x]$ = [1,1,1,0,0,0,0,0,1,1,1,0,1,0,0,1,0]

注) $F_p[x]$はPで繰り返し利用可能

平方数探査 (候補)

- $F_8[x]$で$x<8$の平方数候補： x = (0, 4)
- $F_{15}[x]$で$x<120$の候補(120で繰り返す)

  x = (4,24,28,48,64,84,88,108)
- $F_7[x]$で$x<240$の候補

  x = (4,28,84,124,**144**,168,184,208,228)
- $F_{11}[x]$,$F_{13}[x]$,$F_{17}[x]$で$x<240$の候補

  x = 144 → $(M+x)^2=T^2 \pmod N$, T=855021

RSA暗号解読の応用例

- MPQS(AをA^2に変更)でdを複数選ぶ

  $((A^2x+b)^2-N)/A^2=a^2d$, $b^2=N \pmod{A^2}$ --- (1)
- xを動かしyが平方数となるx,yを2つ求める

  $y=((dx+c)^2-N)/d$, $c^2=N \pmod{|d|}$
- 2つのx,cをx_1,x_2,c_1,c_2、yの平方根をh_1,h_2

  $(dx_1+c_1)^2=dh_1^2$, $(dx_2+c_2)^2=dh_2^2 \pmod N$ →

  $S^2=T^2 \pmod N$, $S=(dx_1+c_1)h_2$ で$T=(dx_2+c_2)h_1$

  注) 一つは(1)式と同一で、新規は一つだけ

応用例(具体的例)

- N=22991079409115583 1
- A=2731の例 (A=2861他でも$S^2=T^2$ (mod N))

  x=51で a=15, d=-49979238, cは8個

  c_1= 6444595, c_2= 32295925

  → **x_1=1**, x_2=7, **h_1= 67353**, h_2=40965 →

  $25738988755623^2=2311402318845^2$ (mod N)

  → N = 455570569 × 504665599

注) aA=40965=h_2で、x_2,h_2は(1)式と同じ

数値実験(環境)

- パソコン

  HP Desktop 870 (Intel Core i7 6700k)

  4Ghz, 8GB
- システム: Windows10 (64ビット)
- コンパイラー

  Cygwin Ver. 2.7 gcc (64ビット), 最適化(-O3)
- 多倍長関数: gmp のmpz_t

数値実験(テストデータ)

- Nは2048ビット (N=P × Q)
  Pはランダムで、Qを調整(探査回数)
- 2種類の方法
  Fermat法　　: $y=(M+x)^2 - N$　　$(P/Q \cong 1)$
  拡張Fermat法: $y=(M+x)^2 - 4abN$　　$(P/Q \cong a/b)$
- 探査回数が4倍づつ増加するデータ
  A-1,A-2,A-3,A-4 (Fermat法)
  B-1,B-2,B-3,B-4 (拡張Fermat法)

数値実験(計算法)

- 平方根(A-1,B-1で100億回探査)
  yを計算し、その平方根から平方数か判定
- 剰余法(A-1,B-1で100億回探査)
  yを計算し、gmpの平方数評価関数で判定
- FSS (Nを因数分解するまで探査)
  考案したFSS(高速平方探査法)で判定
  平方数候補(1/数百億)はyと平方根を計算

測定結果(Fermat法)

方式	データ	探査数 (百億回)	計算 (s)	探査/s (億回)	高速化 (倍)
平方根	A-1	1	5,147	0.02	1
剰余法	A-1	1	1,335	0.07	4
FSS	A-1	900,000	135	700,000	34,000,000
	A-2	3,600,000	594	600,000	31,000,000
	A-3	14,400,000	2,499	600,000	30,000,000
	A-4	57,600,000	7,497	800,000	40,000,000

$Y=(M+x)^2-N$, Nは2048ビット

測定結果(拡張Fermat法)

方式	データ	探査数 (百億回)	時間 (s)	探査/s (億回)	高速化 (倍)
平方根	B-1	1	5,300	0.02	1
剰余法	B-1	1	1,465	0.07	4
FSS	B-1	1,800,000	112	1,600,000	85,000,000
	B-2	7,200,000	561	1,300,000	68,000,000
	B-3	28,700,000	2,377	1,200,000	64,000,000
	B-4	114,700,000	5,656	2,000,000	108,000,000

$Y=(M+x)^2-4abN$, a=1017,b=1231、Nは2048ビット

B-4の平方数探査1 (115京回)

- 合成数N(2048ビット)の因数分解
  拡張 Fermat 法 + FSS (Ver. 8)
  $Y = (M+x)^2 - abN$ (a=1019, b=1231)
  Yが平方数: $Y=h^2$, $g=M+x$ → $g^2=h^2 \pmod N$
  最大探索回数: 10 Exa = 10^{19}
  FSSに使用した素数(P): 32個 ($2^7,3^3,5^2,7,11,\ldots$)
  Yの総計算回数: 137回
  Y (mod P)の計算回数: 2021回

B-4の平方数探査2 (115京回)

- 初期探査(7素数(2^7,3^3,5^2を含む))

  探査: 14.7億, 候補: 16.1万, 削減: 1/9千
- 中間探査(+2素数)

  探査: 6425億, 候補: 1774万, 削減: 1/4万
- 最終探査(+23素数)

  探査: 115京, 候補: 720万 → 平方数判定

  削減: 1/1600億

B-4のN(合成数)

N=
25677387604979174745650113439241870319948692169987586987064217095280862955992909072300653168631621794286691440902481665331169514458834441618096653718919037974637239192749774537792817942978360570513101166733021143008513795392327563304325050993308695779665136072593388015897479804732752568497960089822767151252557699166820228078607982571460614826457133501700637182817549258980767075406408523914787931962356438883296803312387013026992481894919527201138794532384546763185668385783880360321437897202286603418514320652492534062323879516492203743515179770144529988947195323580111950157298979476220823162503958438421236729

B-4の因数分解(115京回探査)

P=145791929591232287489275134536823875882511534844886233972041773955685324499067659355737367435219776275691359706660105337787796768884621535639712814414896336660924310021811719221344573070406754334649949786471208800372307511439908140679051110975412378472667752792254088884793376561260843710839571790913460377479

Q=176123518475767365946317655166663583131866240818503389616863026240872065219187721949865259384450976050418119527869077203941882063294375966994960504929060639996070318400007499258372495278852428986057852847610923166483759711913067321208352861746819143236558364875744117006650037189885830658554695750616522507 51

おわりに

- [拡張]Fermat法でFSSの高速性を評価
- 2048ビットで平方根法の3千万倍～1億倍
- FSSは60兆回～200兆回/sの判定が可能
- RSA暗号解読でFSSはふるい法に代わり得る
- FSSのRSA暗号解読への応用は初期段階
- 1024ビットRSA暗号解読

  GNFSは百億の0-1疎行列で、スパコンが必要

  FSSは数GBのGPU多数をネット接続で可能