

問題適合型高精度計算ライブラリの開発

プロジェクト代表:

長谷川秀彦@筑波大学図書館情報メディア系

発表者:

後 保範@早稲田大学

プロジェクトの進捗状況

- 計算誤り検証付きライブラリの開発 :NG
- RSA 暗号(1024ビット)の強度推定:
今日のメイントピックス Good
- 自動チューニング機構付き線形計算ライブラリの開発 :NG
- 精度保証付き線形計算ライブラリの開発 :NG
- 4倍精度線形計算ライブラリの開発 :NG

RSA 暗号の強度推定

- 初年度
RSA暗号ふるい処理の高速化と評価
- 2年目
標数2の線形方程式解法の作成と高速化
- 3年目(今回)
代数的平方根の計算とRSA暗号の強度推定

RSA 暗号の強度推定の背景

- (1) 現在の暗号 (RSA暗号) や認証システムは多数桁数(1024ビット, 10進309桁)の**因数分解の困難さ**を利用している
- (2) 現在の多数桁数の因数分解の世界記録は、**RSA-768**(10進232桁)。2010年1月、NTT他5カ国共同、GNFSを使用
- (3) 「**暗号の2010年問題**」: RSA暗号を含む現在の暗号システムの変更が必要

RSA-768(232桁)の計算時間

項目	台数・年	比(%)	対象年
ふるい処理	1500	90	初年度
標数2の線形計算	155	9	2年目
利用関数の探査	20	1	対象外
<u>代数的平方根計算</u>	<u>1</u>	<u>0</u>	<u>今回</u>
その他	1	0	今回

注) AMD64 (2.2Ghz, 1コア換算)

行列サイズ: 192,796,550 * 192,795,550

RSA暗号の解読方法

(1) ふるい方式の原理

$A^2=B^2 \pmod{N}$ を多数の素数から作成

(2) GNFS (一般数体ふるい法)

(a) B^2 側を多項式の代数的平方根に変更

$f(m)=N$ で、 $f(x)=ax^6+bx^5+cx^4+\dots+g$

(b) A^2 側を1次式の代数的平方根に変更

$f(M)=0 \pmod{N}$, $h(L)=0 \pmod{N}$

$h(x) = ux+v$

GNFSの計算手順

- 効率の良い多項式 $f(x)$ と1次式 $h(x)$ を探す
- ふるいで $f(x), h(x)$ が共に分解できる整数 s, t
($sx+t$)を基底要素数以上集める
- $\Pi(sx+t)$ が $f(x)$ と $h(x)$ の基底要素の2乗となるよう、標数2の線形計算で整数 s, t を選択
- 代数的平方根計算:
 $F(x)^2 = \Pi(sx+t) \pmod{f(x)}$, $H(x)^2 = \Pi(sx+t) \pmod{h(x)}$ となる $F(x), H(x)$ を求める
- $A^2 = B^2 \pmod{N}$ の形にし、 N を因数分解

代数的平方根の規模 (RSA-768)

- N (分解対象数) : 10進232桁
- 多項式 $f(x)$ の係数 (a,b,\dots,g) : 10進15~45桁
- 基底要素 $(sx+t)$ の係数 (s,t) : 10進1~12桁
- $\Pi(sx+t)$ の $(sx+t)$ の数 : 1億個
- $\Pi(sx+t) \pmod{f(x)}$ の係数 : 10進50億桁
- 多項式 $F(x)$ の係数 : 10進25億桁

代数的平方根の計算方法

- 1次式の平方根

剰余は整数なので中国剰余定理で計算

- 多項式(6次式)の平方根

剰余は5次式

- (a) 中国剰余定理

桁数は短い。途中の符号の選択が困難

- (b) 多数桁の連立非線形ニュートン法(今回使用)

単純だが、50億桁(RSA-768)の計算が必要

連立非線形ニュートン法

- $G(x) = Ax^5 + Bx^4 + \dots + F = \Pi(sx+t) \pmod{f(x)}$
- $F(x)^2 = G(x) \pmod{f(x)}$ なる $F(x)$ を求める
ここで $F(x) = ax^5 + bx^4 + cx^3 + \dots + f$
- $F(x)^2 = g_1(x)x^5 + g_2(x)x^4 + \dots + g_6(x) \pmod{f(x)}$ なる展開式より $g_1(x) = A, \dots, g_6(x) = F$ が成立
- $g_1(x) = A, \dots, g_6(x) = F$ から未知数 a, b, \dots, f を反復計算

ニュートン法の計算回数

- ニュートン法による初期値の選定
10進1万桁計算で収束する初期値を探索
ニュートン法に占める割合は1%以下
- 整数解の選定(6次式)
平均約3個の解を求める必要がある
- RSA暗号を解読
解読確率1/2、平均で約2回計算が必要

多数桁乗算方式

- 1要素(32ビット)に2進32桁で多数桁を表現
- すべて**整数**(int64,int32)を使用
(ES2は**64ビット整数剰余が高速**)
- 乗算は**整数FMT**(高速剰余変換)を使用
(**整数FMTは容易にベクトル化できる**)
- 乗算結果の有効桁数を増やすために、4回の乗算を**中国剰余定理**で重ね合わせる
- **桁上げ計算のベクトル化に工夫が必要**

桁上げ計算のベクトル化対策

- 桁上げ計算は、データ依存性があるため、ベクトル化困難
- 桁上げ(2進32桁単位)要素数が非常に多いため要素数 n を L ずつに分割($n=M \times L$)
- M に1要素追加し、 $(M+1) \times L$ 要素で表現
- 計算を桁上げ方向から、分割方向に変更してベクトル化

ベクトル化桁上げ計算(主要部)

```
for (j = 0; j<M; j++) {  
#pragma cdir nodep(A,B,C,Cover) on_adb(Cover)  
    for (i=0; i<L i++) {  
        Apval    = A[i][j] + Cover[i];  
        Cover[i] = (Apval < A[i][j]);  
        C[i][j]  = Apval + B[i][j];  
        Cover[i] += (C[i][j] < Apval);  
    }  
}
```

ES2による代数的平方根の計算

- RSA-768(10進232桁)
6次多項式で、50億桁の計算
1ノードで3時間
- RSA-1024(10進309桁、推定)
7～8次多項式で、約2000億桁の計算
32ノードで約10時間

ES2によるRSA暗号解読推定

- RSA-768(10進232桁): 2010年解読済み
 - ふるい : 2ノード・年
 - 線形方程式 : 4ノードで800時間
 - 代数的平方根 : 1ノードで3時間
- RSA-896(10進270桁): 未解読
 - ふるい : 60ノード・年
 - 線形方程式 : 16ノードで8ヶ月
 - 代数的平方根 : 8ノードで5時間

PCクラスタとの比較

	ふるい処理	標数2線形計算	代数的平方根
PCクラスタ	キャッシュ処理	ブロックランチョス法	中国剰余定理
ベクトル	メモリ全体処理	工夫が必要	計算法を変更
並列化	超大並列が可能	通信量大	通信量中
特色	素数飛び参照	値が0と1の行列	数千億桁計算
ES2:PC	約800倍	約500倍	約700倍*

まとめ

- RSA暗号解読(ふるい、線形計算、代数的平方根)はすべて**整数演算**を使用
- ベクトル演算率: 98.6% (ベクトル化率: **99.98%**)
- ES2でのRSA-1024解読の必要資源量(推定)
 - ふるい: 1ノード (2,000ノード・年)
 - 標数2の線形計算: 64ノード(6年)
 - 代数的平方根: 32ノード(10時間)(推定)
- PCと比べ、規模が大きくなるとより高速となる