Number Field Sieve at NTU

Cryptography

Number Field

- Choose irreducible $f(x) = c_d x^d + c_{d-1} x^{d-1} + \ldots + c_1 x + c_0$ and $g(x) = e_1 x + e_0$ such that $f(m) \equiv g(m) \equiv 0 \pmod N$ for some integer m
- Let α be a complex root of $f(x)$
  - $\boldsymbol{Q}(\alpha)$ is a finite field extension of $\boldsymbol{Q}$
  - $\boldsymbol{Q}(\alpha)$ is the ***number field*** associated to α
- Define a ring homomorphism φ: $\boldsymbol{Z}[\alpha] \longrightarrow \boldsymbol{Z}_N$ by $\varphi(\alpha) \equiv m \pmod N$
  - φ is a homomorphism since $f(\alpha) = 0$ and $f(m) \equiv 0 \pmod N$
  - $\varphi(a - b\alpha) \equiv a - bm \pmod N$

Squares in Both Sides

- Suppose we find a set S consisting of relatively prime pairs (a, b) satisfying both:

$$\prod_{(a,b)\in S}(a - b\alpha) = \beta^2 \quad \text{for some } \beta \in Z[\alpha]$$

(called the **algebraic side**)

$$\prod_{(a,b)\in S}(a - bm) = y^2 \quad \text{for some } y \in Z$$

(called the **rational side**)

- Let $x = \varphi(\beta)$, then we have $x^2 = \varphi(\beta)^2 = \varphi(\beta^2)$
  $= \varphi(\prod(a - b\alpha)) \equiv \prod(a - bm) = y^2 \pmod N$
- Hence N is factored with probability $\geq \frac{1}{2}$

Norm Function

- The ***norm*** $N(\beta) = \prod \sigma_i(\beta)$ is defined for every $\beta \in \boldsymbol{Q}(\alpha)$
  - $\sigma_i(\beta)$: conjugates of β
- Properties of $N(x)$
  - $N(x) \in \boldsymbol{Z}$ for every $x \in \boldsymbol{Z}[\alpha]$
  - Norm is a multiplicative function: $N(xy) = N(x)\,N(y)$
- ***Sieving*** is to find enough ***relations*** (a, b) : both of the following two numbers are smooth
  - $N(a - b\alpha) = \prod(a - b\alpha_i) = b^d \prod(a/b - \alpha_i)$
    $= \boldsymbol{b^d f(a/b)} = c_d a^d + c_{d-1} a^{d-1} b + \ldots + c_1 ab^{d-1} + c_0 b^d$
  - $\boldsymbol{b\,g(a/b)} = e_1 a + e_0 b$

First Degree Prime Ideals

- Suppose the prime ideal $P \mid < a - b\,\alpha >$
- $p = N(P) \mid N(< a - b\,\alpha >) = b^d f(a/b)$
  $= c_d a^d + c_{d-1} a^{d-1} b + \ldots + c_1 a b^{d-1} + c_0 b^d$
- Therefore
  1) $p \mid b$ and $p \mid c_d$ (not if $c_d = 1$) *or*
  2) $p \mid f(r)$ with $a \equiv br \pmod p$
- Associate P with the pair (p, r)
- There is a bijection between the following two sets
  - The set of first degree prime ideals
  - The set of (p, r) pairs
    - p is a prime
    - $r \in R(p) = \{ r \mid 0 \le r \le p-1 \text{ with } p \mid f(r) \}$

Squares on Algebraic Side

- Find smooth $N(a - b\,\alpha)$
- $N(< a - b\,\alpha >) = p_1^{e1}\, p_2^{e2} \ldots p_k^{ek}$
- Find $r_i \in R(p_i)$ such that $p_i \mid (a - br_i)$
- Then $< a - b\,\alpha > = (p_1, r_1)^{e1} (p_2, r_2)^{e2} \ldots (p_k, r_k)^{ek}$
- So we can find the set S such that

$$\prod_{(a,b)\in S} < a - b\alpha > = \prod (p_i, r_i)^{e_i} = I^2, \; I \text{ is an ideal}$$

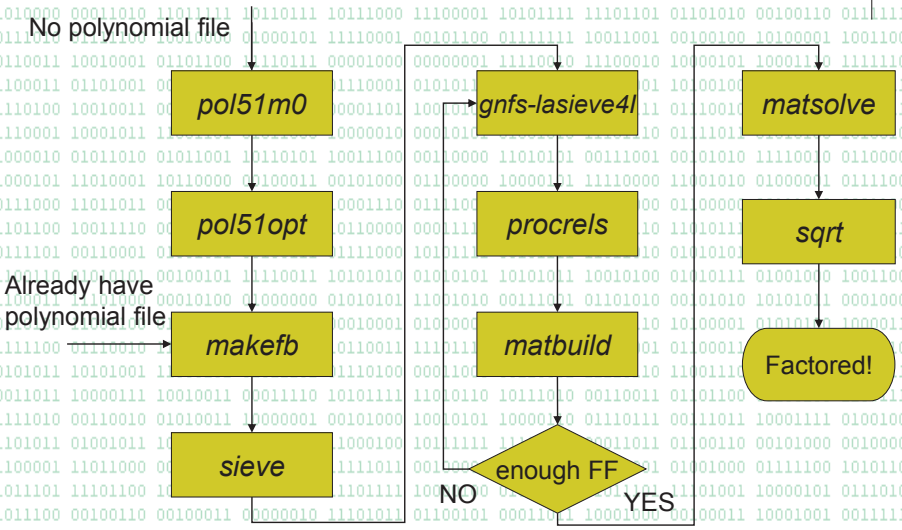
- If I is a principal ideal, $I = < \beta >$,

$$\prod_{(a,b)\in S} < a - b\alpha > = I^2 = < \beta^2 >, \text{ then } \prod_{(a,b)\in S} (a - b\alpha) = \beta^2$$

Major Steps of GNFS

1. Polynomial Selection
   - Find 'good' polynomials to speed up sieving
2. Sieving
   - Find sufficiently many relations to produce a matrix
3. Matrix Reduction
   - Find linear dependencies mod 2 among the rows of the matrix
4. Square Root
   - Calculate a square root in the number field for each dependency, until the factorization is found

Open Source GGNFS



因數分解 RSA-155

- 分解 512 位元 “challenge number” **RSA-155**
  - 1999 年分解紀錄
    - 超過六個月
    - 使用一台超級電腦, 和大約 300 台 PC 或工作站
      - 由 17 位來自世界各地的專家完成
  - 2008 年
    - 少於三天
    - 使用兩台建置於國立臺灣大學的超級電腦

實作成果摘要

RSA-155/512	HP cluster		IBM p595
因數分解	50 核心	424 核心	24 核心
多項式選取	24.0 小時	3.0 小時	
篩法	76.2 小時	< 20 小時	
矩陣化簡			39.7 小時
開平方根	3.0 小時	3.0 小時	
因數分解全程	142.9 小時 (5.95日)	< 65.7小時 (少於2.74日)	

超級電腦

- 超級電腦為何跑得快？
  - 一顆超強 CPU？ Of course not…
  - 很多顆正常 CPU，想辦法合作
- 程式要平行化！
  - 不能只是 single thread

程式平行化

- 兩大平行化模式
  - MPI (Message Passing Interface)
    - 用網路溝通
    - 溝通速度慢
    - 多台一起跑 (幾百台ok!)
  - OpenMP (Open Multi-Processing)
    - Multi-thread
    - 溝通速度快
    - 有極限 (p595上頂多用64顆CPU)

HP cluster

- 建置日期：2006/11
- 完工日期：2007/05
- 運算節點：106
  - HP DL145G3 Server
  - Dual Intel Xeon 3.0GHz
  - 4GB memory
  - Infiniband connection
- 效能:
  - Rpeak: 4900GFlops
  - Rmax: 3022GFlops
- 公告/決標金額: 1990/1890萬元
- 適合對象：
  - 已透過 MPI 平行化之程式
  - 需要大量 CPU 數目之程式

IBM p595 SMP (symmetric multi-processing)

- 建置日期：2006/11
- 運算節點：
  - IBM p595
  - 64*Power5 1.9GHz CPU
  - 256GB memory
  - AIX 5.3
- 效能:
  - Rpeak: 486GFlops
  - Rmax: 421GFlops
- 公告/決標金額: 1990/1950萬元
- 適合對象：
  - 已透過 OpenMP 平行化之程式
  - 大量記憶體需求之程式

奇怪…

- 價錢一樣…CPU量差那麼多？
  - IBM p595: 64 CPUs
  - HP cluster: 106 × 4 = 424 CPUs
- 兩台根本不一樣
  - p595：一台大電腦，64顆CPU，256GB RAM
  - HP：106台電腦用網路連起來

p595強項

- 64顆CPU access同一塊記憶體
  - 非常適合OpenMP加速
  - SMP 大型主機
- RAM超大，256GB
  - 適合跑需要很多記憶體的程式
- 銀行使用
  - 金額synchronize，無法分開算

p595弱項

- 也只有64顆CPU...
  - 無法再更加速...
  - 可以跑MPI程式，但CPU太少...
- 個別CPU速度慢
  - 才1.9GHz
  - 工作站的電腦跑的可能還比較快

HP cluster 強項 / 弱項

- 424顆core
  - MPI超適合
- 106台電腦記憶體獨立
  - OpenMP非常不適合
    - 最多四個 threads
  - 無法使用大量記憶體

➢ 沒有萬能的超級電腦

- OpenMP: 可能用p595較好
- MPI: 可能用HP cluster較好

GNFS 四大步驟

1. 「多項式選取」 Polynomial Selection
   - 尋找兩個不可約 (irreducible) 多項式 f_1 和 f_2 使得
     1. 在 modulo N 之下，f_1 和 f_2 有相同的根 m
     2. 因數基底完全分解 (smooth) 之多項式值越多越好
   - f_1 和 f_2 對 sieving 效率影響極大
   - 須兼顧 “size property” 與 “root property” 二項特性
   - GGNFS的程式Pol5 分為 Pol51m0 和 Pol51opt 兩部份
     - 前者操作 Kleinjung 演算法，輸出由 (a_5, p, d) 三數唯一決定的 f_1 和 f_2，具有好的 size property；後者執行 Murphy 的演算法，改善其 root property

Polynomial Evaluation

- Actual Murphy_E value

$$\frac{6}{\pi^2}\int_A \rho\left(\frac{\log(f(x,y))+\alpha_1}{\log(B_1)}\right)\rho\left(\frac{\log(g(x,y)+\alpha_2}{\log(B_2)}\right)dxdy$$

- Approximations
  - Root Property $\sum_{p\ small}\left(1-r(f,p)\frac{p}{p+1}\right)\frac{\log(p)}{p-1}$
  - Size Property $\frac{1}{2}\log\left(\iint_S y^d f(x/y)\,dxdy\right)$
  - L^2 sup-norm $\sqrt{\int_0^1\int_0^1\left(f\left(\frac{sx}{y}\right)\times\left(\frac{y}{\sqrt{s}}\right)^d\right)^2 dxdy}$

Polynomials for RSA-155 (512 bits)

a) Factorization in 1999:

- $f(x) =$
  $119377138320x^5$
  $-80168937284997582x^4$
  $-662698522341185744 45x^3$
  $+11816848430079521880356852x^2$
  $+74596615800717864439197430 56x$
  $-40679843542362159361913708405064$
- $g(x) =$
  $x - 391230797211680007713134490 81$
- Murphy_E: 1.905e-12

b) Kleinjung's paper in 2006:

- $f(x) =$
  $4985820x^5$
  $+15578368316860x^4$
  $-5137488762804904 87x^3$
  $-10211574130795357032973 44x^2$
  $-39893111467231678678251 49900x$
  $+14658919460374074323550710377995600$
- $g(x) =$
  $87494555574829559x$
  $-29394756538965034296055627 0613$
- Murphy_E: 2.792e-12

Polynomials for RSA-155 (512 bits)

c) Our result obtained on the HP cluster at NTU

- $f(x) =$ $83772000x^5$ ← We ran from 1 to 10^9
  $-55340006499600x^4$
  $-578998746640536264 78x^3$
  $+42764560282021639254 79457x^2$
  $+22350079225298842053344018218 00x$
  $-14068501632188545244303052842 00079$
- $g(x) =$ $205409829331650555 7x - 167185341081359137443707501330$
- Murphy_E: 2.980e-12

- b) is 46% better than a)
- c) is 56% better than a)

Overall Performance

Number	Optimal c_5 value	Total Time (hrs)
RSA-155	83772000	1176.5
D155_01	56847420	1098.2
D155_02	106560360	984.5
D155_03	255172680	1117.2
D155_04	218225280	1154.5
D155_05	90300420	1167.8
D155_06	91654080	1114.0

- Select good polynomials for 155-digit numbers in 24 hours with 50 cores, or in **3 hours** with **400 cores**
- Terminating Condition: Running Time or c_5 Range

GNFS 四大步驟

2. 「篩法」 **Sieving**

- 運算量最大、執行總時間最長的步驟
- 尋找夠多的 (a, b)，使得 $b^{\deg(f_1)} f_1(a/b)$ 和 $b^{\deg(f_2)} f_2(a/b)$ 同時分別被代數端與有理數端的因數基底分解
  - 我們稱滿足上述條件的數對 (a, b) 為 "relation"
- Sieving 有兩種方式: lattice sieve 和 line sieve
- Line Sieve 簡單地在 (a, b) 平面上水平尋找，但隨著 b 增加，(a, b) 是 smooth 機率很快下降
- Lattice Sieve 不是尋找整個 (a, b) 平面，只找 $N(a - b\alpha)$ 可被 special q 整除的(a, b)，即代數端基底中的質因數

Performance

- RFB = Rational Factor Base
- AFB = Algebraic Factor Base
- 50 cores: 76 hr 13 min
- 424 cores: < 20 hours

Statistics Comparison between Ours and [4]		
	Ours	[4]
RFB Line Sieve_1 (1)	20 000 000	44 000 000 (2)
AFB Line Sieve_1	40 000 000	110 000 000 (2)
RFB Line Sieve_2	–	8 000 000 (3)
AFB Line Sieve_2	–	25 000 000 (3)
RFB Lattice Sieve	20 000 000	16 777 216 (4)
AFB Lattice Sieve	40 000 000	16 777 216 (4)
% Relations Found by Line Sieve	< 1%	29%
% Relations Found by Lattice Sieve	> 99%	71%
Calendar Time	76 hours 13 minutes	3.7 months
Total CPU Time	147 days 22 hours	35.7 years
Total Relations (no duplicates)	69 523 978	85 534 688
Large Primes Limit	1 073 741 824	1 000 000 000

GNFS 四大步驟

3. 「矩陣化簡」Matrix Reduction
   - GGNFS 的 matrix reduction 再區分為三大部份
   1. **matbuild**：篩法結束後，合併或刪掉有 large prime 的 relations，建構矩陣
   2. **matprune**：亦稱 “filtering”，對上一步所得到巨大而稀疏 (絕大部份元素是零) 的矩陣進行特別處理，降低 size 和 weight，以節省下一步的執行時間
   3. **matsolve**：亦稱 “finding dependencies”，解線性方程組，構成完全平方數
      - 高斯消去法不適用，因所需記憶體太大且執行時間過長
      - 以 Block Wiedemann 或 Block Lanczos 為主

Matrix Reduction

- Matrix Pruning
- Matrix Solving (Block Lanczos)
- Performed on the IBM p595 with 24 cores out of 64

	Matrix Size	Weight
Before Pruning	3703230 × 4245939	574694874
After Pruning	3420140 × 3438720	449074181

Statistics Comparison between Ours and [4]		
	Ours	[4]
Matrix Building	7 hours 20 minutes (2)	N / A
Matrix Pruning	2 hours 14 minutes (3)	N / A
Matrix Building and Pruning	9 hours 34 minutes	1 month (4)
Matrix Solving	37 hours 31 minutes (3)	224 hours (5)

Optimization of *matprune*

```
for(i = 0; i < omp_thread_count; i++){
s32 *one_threads_row = rowMember_big[i];
#pragma omp for
for(j = 0; j < M -> numRows; j++){
  if( rowMember[j] != -2 ){
    if( one_threads_row[j] == -2 ){
      rowMember[j] = -2;
    }else if( one_threads_row[j] >= 0 ){
      if( rowMember[j] >= 0 ){
        rowMember[j] = -2;
      }else{
        rowMember[j] = one_threads_row[j];
      }
}
}
}
```

- 以「電路」取代 if-else

```
int x = rowMember[j];
int y = one_threads_row[j];
int A = ( x == -1 );
int B = ( y == -1 );
rowMember[j] += A * ( y – x )
    + (!( A | B )) * (-2 – x );
```

- 與左方的程式碼等價

Parallelization of *matsolve*

Matsolve 使用不同數量的 core 所需時間比較表

core 數量	秒數	進步比例			
		和前一項比較		和 4 個 cores 比較	
		理論	實際	理論	實際
4	30724			1	1
8	17267	2	1.779348	2	1.779348
16	9139	2	1.889375	4	3.361856
24	6745	1.5	1.35493	6	4.555078

Parallelization of *matsolve*

- 當核心 (CPU) 增加一倍，速度分別增加為 1.78 與 1.89 倍，和理論值 (完美加速值) 的誤差只有 11% 與 5.5%，是非常有效的平行化，表示我們已經成功找出並平行化 matsolve 程式中的瓶頸
- 但若使用 32 核心，由於某些硬體因素，執行效率甚差
- MultB_T64 和 MultB64 是兩個關鍵函式，平行化前者較容易，後者較麻煩；我們解決 race condition 的方式是先讓每個 thread 都有自己一份 Product_threads_specific 陣列，每個 thread 先把自己被分配到的部分乘完

GNFS 四大步驟

4. 「開平方根」Square Root

- 計算於第三步驟所得的 $\prod_{(a,b)\in S}(a-b\alpha)$ 形式代數整數之平方根，再以同態函數對應至分解所需整數
- 此步驟所佔時間比例相當低，並非瓶頸
- 但是即使如此，瞭解並實作任何一套適用於此處的平方根計算方法，皆需要大量代數知識
- GGNFS 的開平方根原始碼亦長達數千行

Square Root

- At the HP cluster:

Statistics Comparison between Ours and [4]

	Ours	[4]
Square Root	2 hours 59 minutes	45 hours 33 minutes(1)

- Factorization of RSA-155:

109417386415705274218097073220403576120037329454492059909138421314763499842889347847179972578912673324976257528997818337970765372440271467435315933543338975 =

10263959282974110577205419657399167590071656780803806680334193352179071130777 9 ×

10660348838016845482092722036001287867920795857598929152227060823719306280864 3

Comment and Conclusion

- We have shown that …
  - RSA-512 can be cracked within 3 days at NTU
  - Using RSA-512 is very dangerous today
    - It is still an option of SSL (secure socket layer)
- Experts in cryptography community believe that factoring RSA-1024 in one year is feasible now
  - Say, by National Security Agency of USA?
- If RSA-1024 must be used, better change keys every six months
- Time to switch to Elliptic Curve Cryptosystems?

- 主旨: 關於 HP cluster 主機的磁碟空間使用

陳教授，

抱歉有個雜事要麻煩您，最近我們在清查系統磁碟空間使用，發現您在 HP cluster 主機使用量超過 1 TB, 可否麻煩請貴團隊幫忙清理一下。

Thanks.

計中張傑生敬上

- 讓我紀錄一下　以後應該也沒什麼機會這樣了..

jerry@HPC118:~$ du -sh

1.1T

我開始刪了