FIWE: Factoring Integers with ECM

Asena Durukan
Aslı Altıparmak
Elif Özbay
Hasan Ozan Soğukpınar
Nuri Furkan Pala
Advisor: Dr. Hüseyin Hışıl

Yaşar University

Table of Contents

Elliptic Curve Method

Figure: Arjen K. Lenstra [1]

- Arjen K. Lenstra
- One of the fastest special purpose factoring algorithms
- Pollard's p - 1 method
- B-smooth numbers
- Elliptic curves

YAŞAR UNIVERSITY

Elliptic Curve Method

- B-smooth numbers

Notation

$$p_1^{n_1} * p_2^{n_2} * p_3^{n_3} * ... * B^n$$

Example

$$3^2 * 5^4 * 7^2 * 11^3$$

- Number Field Sieve Algorithm
- Factorization of multiple B-smooth numbers

Elliptic Curve Method

Algorithm 1: ECM Algorithm

input : n: composite number to be factorized

output: d: factor of n

```
1  Generate a random projective curve C and a point P on it
2  Choose an integer k that is composed of small prime factors
3  Compute k * P on C
4  if k * P failed due to the lack of inverse of an element then
5      return d = gcd(k, n);
6  else
7      if Threshold for k has not been exceeded then
8          Go back to Step 2.
9      else
10         if Threshold for C has not been exceeded then
11             Go back to Step 1.
12         else
13             Failed. Stop.
14         end
15     end
16 end
```

Elliptic Curve Method

Requirements

- Multi-precision arithmetic
- Elliptic curve arithmetic
- GCD operation
- Modular inversion operation
- Factorizing integers fast
- Factorizing multiple integers simultaneously
- User interface

YASAR UNIVERSITY

Elliptic Curve Method

Algorithm 2 Montgomery's binary algorithm in the group $\mathcal{E}_{(A,B)}(\mathbb{F}_q)$

Input: $k = \sum_{i=0}^{\ell-1} k_i 2^i$ with $k_{\ell-1} = 1$, and $P \in \mathcal{E}(\mathbb{F}_q)$
Output: $[k]P$
Cost: $\ell - 1$ calls to $\oplus$ and ℓ calls to $[2]$

$(\mathsf{R}_0, \mathsf{R}_1) \leftarrow (P, [2]P)$
for $i = \ell - 2$ **down to** 0 **do**
  if $k_i = 0$ **then**
    $(\mathsf{R}_0, \mathsf{R}_1) \leftarrow ([2]\mathsf{R}_0, \mathsf{R}_0 \oplus \mathsf{R}_1)$
  else
    $(\mathsf{R}_0, \mathsf{R}_1) \leftarrow (\mathsf{R}_0 \oplus \mathsf{R}_1, [2]\mathsf{R}_1)$
return R_0

Safegcd

Properties of Safegcd

- Binary GCD
- Finds both GCD & modular inversion
- Can be easily transformed constant-time
- Recent

Safegcd

Algorithm 3: Safegcd for Modular Inverse

input : f, g, $precomp$, t, n

output: z

$f = p$

$\delta,\ U,\ R = 1$

$V,\ Q,\ i = 0$

while $i < n$ **do**

  δ, u, v, q, r=divsteps(t,t,δ,f%(2^t),g%(2^t))

  $f,\ g = (u \times f + v \times g)//(2^t),\ (q \times f + r \times g)//(2^t)$

  $U,\ Q = (u \times U + v \times Q),\ (q \times U + r \times Q)$

  $V,\ R = (u \times V + v \times R),\ (q \times V + r \times R)$

  $i = i + 1$

end

$z = (sign(f) \times V \times precomp)\%p$

return z

Safegcd

Algorithm 4: Safegcd for GCD

input : f, g, t, n

$\delta,\ U,\ R = 1$

$V,\ Q,\ i = 0$

while $i < n$ **do**

  δ, u, v, q, r=divsteps(t,t,δ,f%(2^t),g%(2^t))

  $f,\ g = (u \times f + v \times g)//(2^t),\ (q \times f + r \times g)//(2^t)$

  $i = i + 1$

end

$z = f$

return z

Safegcd

Safegcd in ECM

- Fast enough
- Multi-precision input
- Can be parallelized

Parallelization

- Factorization of multiple numbers simultaneously
- Computationally large problems

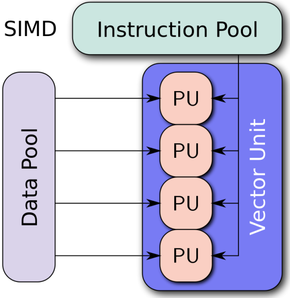

Figure: SIMD Architecture [2]

Parallelization

Parallelization Assumptions

- NVIDIA Graphic Card
- Linux Operating System
- CUDA Language
- Nsight IDE

Parallelization Restriction

- Thread Memory

Figure: Nvidia 1050 Ti

Test & Implementation

Table: Test Statistics

Function Name	Is Tested?	Size of Inputs	# of Trial	# of Success	# of Fail	Time Spent in C (sec)
proCurvePointGMP	Yes	1-100	10000	10000	0	15
proAddGMP	Yes	1-100	10000	10000	0	12
proAddMagma	Yes	1-100	10000	10000	0	45
proDblMagma	Yes	1-100	10000	10000	0	33
proLadderGMP	Yes	1-100	10000	10000	0	27
proLadderMagma	Yes	1-100	10000	10000	0	143
safegcdPyMul	Yes	1-100	10000	10000	0	37
ecmGmpTest	Yes	1-10	1000	1000	0	34

Operation

- External library
- Bash script

Summary & Conclusion

- What we have done?
- Benefits
- Future work

	work hour per member				
month	meeting	report	design	implementation	total
2019-09	8	8	6	0	22
2019-10	10	12	10	0	32
2019-11	10	14	10	0	34
2019-12	12	14	6	0	32
2020-01	9	3	6	18	36
2020-02	8	16	5	25	54
2020-03	9	15	3	25	52
2020-04	10	21	2	29	62
2020-05	8	26	3	27	64
total	84	129	51	124	388

388	total hour per member
1940	total hour for group
242	total man-days for group

References

Wikipedia contributors.
Arjen lenstra — Wikipedia, the free encyclopedia.
https://en.wikipedia.org/w/index.php?title=Arjen_Lenstra&oldid=879950835, 2019.

Wikipedia contributors.
Simd — Wikipedia, the free encyclopedia, 2020.