Factoring of Large Composite Numbers

AK Bhateja

Integer Factorization

- The security of RSA public-key, Rabin public-key cryptographic techniques depends upon the intractability of the integer factorization problem
- Some factoring techniques
  - Trial Division
  - Pollard's rho
  - Pollard's p – 1
  - Quadratic Sieve

Trial division

- For each integer d greater than 1 and not larger than $\sqrt{n}$, check if d divides n.
- It takes roughly $\sqrt{n}$ divisions in the worst case when n is a product of two primes of the same size.
- Good if n is less than 240
- The density of primes is $n/\ln n$, and since primes go up to $\sqrt{n}$ the complexity is $O(\sqrt{n}/\ln\sqrt{n})$.

Pollard's p -1 factoring

- Due to J. M. Pollard 1974.
- B-smooth: Let B be a positive integer. An integer n is B-smooth, or smooth with respect to a bound B, if all its prime factors are $\leq B$.

  Ex. 153 is 17-smooth, because $153 = 3^2 \times 17$
- Pollard's $p - 1$ algorithm with bound B will succeed to find a factor of n if n has a factor p such that $p - 1$ is B-smooth.

Pollard's p -1 factoring

- Let B be a smoothness bound. Let Rbe the least common multiple of all powers of primes $\leq B$.
- If $q^l \leq B$, then $l \ln q \leq \ln n$ *i.e.* $l \leq \lfloor \ln B / \ln q \rfloor$
- Therefore$R = \prod_{q \leq B} q^{\left\lfloor \frac{\ln B}{\ln q} \right\rfloor}$
- If p is a prime factor of n such that $p - 1$ is B-smooth, then p - 1 divides R
- For any a satisfying $\gcd(a, p) = 1$, $a^{p-1} \equiv 1 \pmod{p}$.

$\therefore$ $a^R \equiv 1 \pmod{p} \Rightarrow p \mid \gcd(a^R - 1, n)$.

- $\gcd(a^R - 1, n)$ is a non-trivial factor of n

Pollard's *p*-1 Algorithm

1. Choose a smoothness bound B, usually about $10^5 - 10^6$
2. Select a random integer a, $2 < a < n - 1$,
   compute $d = \gcd(a, n)$.
   If $d \geq 2$ then return(d).
3. for each prime $q \leq B$

do compute $l = \lfloor \ln B / \ln q \rfloor$

compute $a = a^{q^l} \bmod n$

4. Compute $d = \gcd(a - 1, n)$.
   If $d = 1$, go to 1 and increase B
   If $d = n$, go to 2 and change a

$d \neq 1$ & $d \neq n$, return d (It is a non-trivial factor of n)

Example: Pollard's p –1 method

To factor the number $n = 299$.
Select $B = 5$.
Thus $R = 2^2 \times 3^1 \times 5^1$
Select $a = 2$

$q = 2, l = 2, \ q^l = 4, \ a = 2^4 \mod 299 = 16$
$q = 3, l = 1, \ q^l = 3, \ a = 16^3 \mod 299 = 209$
$q = 5, l = 1, \ q^l = 5, \ a = 209^5 \mod 299 = 170$

$g = \gcd(a - 1, n) = \gcd(169, 299) = 13$
Since $1 < 13 < 299$, thus return 13
Other factor of 299 is $299 / 13 = 23$ is prime
Thus it is fully factored: $299 = 13 \times 23$

Conditions of success of the algorithm

- Pollard's $p-1$ algorithm with bound B will succeed to find a factor of n if n has a factor p such that $p - 1$ is B-smooth
- If this B is too big, then Pollard's $p - 1$ algorithm will not be faster than trial division.

Pollard’s rho factoring algorithm

- Pollard’s rho algorithm is a special-purpose factoring algorithm for finding small factors of a composite integer.
- It was invented by John Pollard in 1975.
- It uses only a small amount of space, and its expected running time is proportional to the square root of the size of the smallest prime factor of the composite number being factorized.

Pollard's rho algorithm

- Let p be a prime factor of a composite integer n.
- Consider the following thought experiment.

  Pick some numbers $x_0, x_1, x_2, \ldots, x_l$ uniformly at random in Z_n .

  Assume that all the numbers are distinct .

  suppose that there exists some $0 \le i < j \le l$ such that

  $$x_i \equiv x_j \pmod{p}.$$

  Then $p \mid (x_i - x_j)$, and since $p \mid n$ also, $\therefore\ p \mid \gcd(x_i - x_j, n)$.
- Since $-n < x_i - x_j < n$ and $x_i \neq x_j$, $\gcd(x_i - x_j, n) < n$.
  Thus $\gcd(x_i - x_j, n)$ provides a nontrivial factor of n.

Pollard's rho factoring: requirements

- Generation of $x_0 , x_1 , x_2 , \ldots , x_l$ uniformly at random in Z_n
- Finding x_i and x_j ; $0 \leq i < j \leq l$ such that

$$x_i \equiv x_j \pmod{p}.$$

Since p divides n but is unknown, this can be done by computing the terms $x_i \bmod n$, instead of $x_i \bmod p$ and searching x_i and x_j s.t. $\gcd(x_i - x_j , n) > 1$.

- If $\gcd(x_i - x_j , n) > 1$, then a non-trivial factor of n is $\gcd(x_i - x_j , n)$.
- Pollard's idea was to choose $x_0 , x_1 , x_2 , \ldots , x_l$ in a recursive manner, choosing $x_0 \in Z_n$ at random and then computing $x_m = f(x_{m-1}) \bmod n$ for some appropriate function f.

Pollard's rho factoring

- Generation of random numbers: The clever trick here is to not pick the x_i 's randomly, but instead in a way that "looks" random.
- Let $f : Z_n \rightarrow Z_n$ be defined by $f(x) = x^2 + 1 \bmod n$. The sequence

  $x_0 , \ x_1 = f(x_0), \ x_2 = f(x_1), \ldots , x_i = f(x_{i-1}), \ldots .$
  looks random
- Finding x_i and x_j ; $0 \leq i < j \leq l$ s.t. $x_i \equiv x_j \pmod n$ can be done by Floyd's cycle detection algorithm.

Floyd's Cycle Detection

- Finding a cycle in a sequence of iterated function values.
- For any function f that maps a finite set S to itself, and any initial value x_0 in S, the sequence of iterated function values

  $$x_0,\ x_1 = f(x_0),\ x_2 = f(x_1),\ \ldots,\ x_i = f(x_{i-1})$$

  there must be some pair of distinct indices i and j such that $x_i = x_j$.

Robert W. Floyd's cycle detection Algorithm

- Floyd’s Cycle-Finding Algorithm is similar to the Tortoise Hair Story.
- Floyd’s Cycle-Finding Algorithm uses two pointers that move at different speeds. If there is a cycle, both of the pointers would point to the same value at some point in the future.
- The Tortoise and the Hare are both pointers, and both start at the top of the list. For each iteration, the Tortoise takes one step and the Hare takes two. If there is a loop, the hare will go around that loop (possibly more than once) and eventually meet up with the turtle. If there is no loop, the hare will get to the end of the list without meeting up with the turtle.
- This algorithm takes O(N) time.

Pollard's rho factoring

- Let n, the integer to be factored
- Consider polynomial $f(x) = x^2 + 1 \bmod n$ and find

  x_m and x_{2m} s.t. $\gcd(x_m - x_{2m}, n) \neq 1$
- If $\gcd(x_m - x_{2m}, n) > 1$, then a non-trivial factor of n is $\gcd(x_i - x_j, n)$.
- The situation $\gcd(x_m - x_{2m}, n) = n$ occurs with negligible probability.

Pollard's rho algorithm for factoring integers

INPUT: a composite integer n that is not a prime power.

Set $a \leftarrow 2$, $b \leftarrow 2$, $d \leftarrow 1$

while $d = 1$

$a \leftarrow f(a) \bmod n$

$b \leftarrow f(f(b)) \bmod n$

$d \leftarrow \gcd(|a - b|, n)$

If $d = n$ then terminate the algorithm with failure

else return(d) and terminate with success

Note: (options upon termination with failure) try again using a starting value other than 2 or with a different polynomial instead of $f(x) = x^2 + 1$. i.e. $f(x) = x^2 + c$, $c \neq 1$.

Pollard's rho algorithm for factoring

- The expected time for Pollard's rho algorithm to find a factor p of n is $O(p^{1/2})$ modular multiplications. i.e. the expected time to find a non-trivial factor of n is $O(n^{1/4})$ modular multiplications.
- Example: Let $n = 8051$ and $f(x) = x^2 + 1 \bmod 8051$.

a	b	$\lvert a - b \rvert$	$\gcd(\lvert a - b \rvert, 8051)$
2	2	0	1
5	26	21	1
26	7474	7450	1
677	871	194	97

97 is a non-trivial factor of 8051.

Fermat's factorization method

- Let n be an odd integer as the difference of two squares:

  $$n = a^2 - b^2$$

  then $n = (a - b)(a + b)$ if neither factor equals one, it is a proper factorization of n.
- Each odd number (not prime) has such a representation.

  If $n = cd$ is a factorization of n, then

  $n = ((c + d)/2)^2 - ((c - d)/2)^2$

  Since n is odd, then c and d are also odd, so those halves are integers.

Algorithm: Fermat's factoring

$a \leftarrow \text{ceiling}(\text{sqrt}(n))$

$b \leftarrow a \times a - n$

repeat until b is a square:

$a \leftarrow a + 1$

$b \leftarrow a \times a - n$

return $(a - \text{sqrt}(b))$

Dixon's factorization method

- It is a general-purpose integer factorization algorithm; it is the prototypical factor base method.
- It was designed by John D. Dixon, a mathematician and published in 1981.
- (Basic idea) find a congruence of squares modulo the integer n, i.e. select random or pseudorandom x values, s.t. $x^2 \bmod n$ is a perfect square.

  i.e. If $x^2 \bmod n \equiv y^2$, then $\gcd(x - y, n)$ is a factor of n.
- Approach: start with $x = \lceil \sqrt{n} \rceil$ and counting up so that $x^2 \bmod n$ is a perfect square.

Theorem: Let n be a positive integer. Suppose there exist integers x, y such that $x^2 \equiv y^2 \pmod{n}$ but $x \neq \pm y \pmod{n}$. Then $\gcd(x - y, n)$ gives a non-trivial factor of n.

Proof: Let $x^2 \equiv y^2 \pmod{n}$

$n \mid (x - y)(x + y)$

Since $x \neq \pm y \pmod{n}$

implies neither $n \mid (x - y)$ nor $n \mid (x + y)$

Therefore $\gcd(x - y, n)$ is a non-trivial factor of n.

Practicality of Dixon's factorization method

- Selecting random x values will take long time to find a congruence of squares, since there are $\sqrt{n}$ squares less than n.
- Better approach: To factorize n,
  - choose an integer B (bound) and a set

    $P = \{ p;$ prime $\leq B\}$, called factor base
  - Search for integer z s.t. z^2 mod n is B-smooth i.e.

$$z^2 \bmod n = \prod_{p_i \in P} p_i^{e_i} \pmod n$$

  - Generate enough such relations so that the exponents of the primes on RHS are all even i.e.

$$z_1^2 \cdot z_2^2 \cdots z_k^2 \equiv \prod_{p_i \in P} p_i^{e_{i,1}+e_{i,2}+..,+e_{i,k}}$$

    where $e_{i,1} + e_{i,2} + \ldots + e_{i,k}$ is even

  This gives $x^2 \bmod n \equiv y^2$

Algorithm: Dixon's factorization

choose bound B

repeat

for $i = 1$ to $k+1$

choose z_i between 1 and n s.t. $z_i^2 \bmod n$ is B-smooth

i.e. $z_i^2 \bmod n \equiv \prod_{p_j \in P} p_i^{e_{ij}}$

find non empty set S s.t. $\sum_{z_i \in S} e_{ij}$ is even

$$x = \prod_{z_i \in S} z_i \bmod n, \quad y = \prod_{p_j \in P} p_j^{\left(\sum_{z_i \in S} e_{ij}\right)/2} \bmod n$$

while $x \equiv \pm y \bmod n$

return $\gcd(x + y, n)$

Example: Dixon's factorization

$n = 23449$ and factor base P = {2, 3, 5, 7}

$[\sqrt{n}] = 154$. Starting here for z_i

The first related squares are:

$970^2 \bmod (23449) = 2940 = 2^2 \times 3 \times 5 \times 7^2$

$8621^2 \bmod (23449) = 11760 = 2^4 \times 3 \times 5 \times 7^2$

So, $(970 \times 8621)^2 \equiv (2^3 \times 3 \times 5 \times 7^2)^2 \pmod{23449}$

i.e. $14526^2 \equiv 5880^2 \pmod{23449}$

Now, find: gcd(14526 -5880, 23449) = 131

gcd(14526 + 5880, 23449) = 179

Factors are $n = 131 \times 179$

Quadratic sieve factoring

- Invented by Carl Pomerance in 1981
- Fastest algorithm for factoring a numbers up to 110 digits long.
- An improvement to Dixon's factorization method.
- Basic idea
  - Tries to set up a congruence of squares modulo *n* to find a factor

Sieve of Eratosthenes for Prime Number

1. Start with all numbers greater than 1
2. Divide all by the first number
3. Repeat until no numbers are left to divide by
4. What remains are the prime numbers.

Quadratic sieve factoring

- Fact: Let x and y be integers. If $x^2 \equiv y^2 \bmod n$ but $x \neq \pm y \bmod n$, then $\gcd(x - y, n)$ is a nontrivial factor of n.
- Fact: Let n be an odd composite integer that is divisible by k distinct odd primes. If $a \in Z^*_n$, then the congruence $x^2 = a^2 \pmod n$ has exactly 2^k solutions modulo n, two of which are $x = a$ and $x = -a$.
- If x and y are two randomly selected integers, s.t. $x^2 \equiv y^2 \bmod n$, then with probability at least 1/2 it is the case that $x \neq \pm y \bmod n$.

Finding x & y at random, satisfying $x^2 \equiv y^2 \bmod n$

- (Factor Base) A set consisting of the first t primes S = $\{p_1, p_2, \ldots, p_t\}$ is chosen.
- Find pairs of integers (a_i, b_i) satisfying the following

  (i) $a_i^2 \equiv b_i \pmod{n}$

  (ii) $b_i = \prod_{j=1}^{t} p_j^{e_{ij}}$, $e_{ij} \geq 0$, i.e. b_i is p_t smooth

- Find a subset of the b_i's whose product is a perfect square.

Quadratic sieve factoring

- Let integer n is to be factorized
- Let $m = \lfloor \sqrt{n} \rfloor$, define polynomial $Q(x) = (x + m)^2 - n$

  $Q(x)$ is small, if x is small in absolute value.

  $x = \pm 1, \pm 2, \ldots$
- If $a_i = (x + m)$ & $b_i = (x + m)^2 - n$, then
  $a_i^2 = (x + m)^2 \equiv b_i \pmod{n}$
- Therefore, aim is to selects $a_i = (x + m)$ and tests whether $b_i = (x + m)^2 - n$ is p_t smooth.

Factor base

- A prime p divides b_i then $(x + m)^2 \equiv n \pmod{p}$ i.e. n is a quadratic residue modulo p.
- Thus the factor base need only contain those primes p for which the Legendre symbol (n/p) is 1.
- Since b_i may be negative, - 1 should be included in the factor base.
- Example: For $n = 24961$, $t = 6$

  Factor base = {-1, 2, 3, 5, 13, 23}

Checking smoothness of b_i

- For each i, associate the binary vector $v_i = (v_{i1}, v_{i2}, \ldots, v_{it},)$ with the integer exponent vector $(e_{i1}, e_{i2}, \ldots, e_{it})$ such that $v_{ij} = e_{ij} \bmod 2$.
- If $t + 1$ pairs (a_i, b_i) are obtained, then the t-dim vectors $v_1, v_2, \ldots, v_{t+1}$ must be linearly dependent over Z_2.

  i.e. there must exist a non-empty subset $T \subseteq \{ 1, 2, \ldots, t + 1)$ s.t. $\sum_{i \in T} v_i = 0$, and hence $\prod_{i \in T} b_i$ is a perfect square.

Finding x & y

- $\prod_{i \in T} a_i^2$ is a perfect square.
- Thus $x = \prod_{i \in T} a_i$ and y square root of $y = \prod_{i \in T} b_i$ satisfies $x^2 \equiv y^2 \bmod n$.
- If this pair also satisfies $x \neq \pm y \bmod n$, then $\gcd(x - y, n)$ is a nontrivial factor of n.

  Otherwise, some of the (a_i, b_i) pairs may be replaced by some new such pairs.
- In practice, there is high probability that at least one (x, y) pair satisfying $x \neq \pm y \bmod n$

Example

- $n = 24961$
- Select the factor base S = {-1, 2, 3, 5, 13, 23} of size $t = 6$.
- 7, 11, 17 and 19 are omitted from S since $(n/p) = -1$ for these primes.
- Compute $m = \lfloor \sqrt{24961} \rfloor = 157$.
- collected for the first $t + 1$ values of x for which $Q(x)$ is 23-smooth.

i	x	$Q(x)$	Factors of $Q(x)$	a_i	v_i
1	0	-312	$-2^3 \cdot 3 \cdot 13$	157	(1, 1, 1, 0, 1, 0)
2	1	3	3	158	(0, 0, 1, 0, 0, 0)
3	-1	-625	-5^4	156	(1, 0, 0, 0, 0, 0)
4	2	320	$2^6 \cdot 5$	159	(0, 0, 0, 1, 0, 0)
5	-2	-936	$-2^3 \cdot 3^2 \cdot 13$	155	(1, 1, 0, 0, 1, 0)
6	4	960	$2^6 \cdot 3 \cdot 5$	161	(0, 0, 1, 1, 0, 0)
7	-6	-2160	$-2^4 \cdot 3^3 \cdot 5$	151	(1, 0, 1, 1, 0, 0)

- Linearly dependence: $v_1 + v_2 + v_5 = 0$
- Compute $x = (a_1 a_2 a_5 \bmod n) = 936$.
- Exponents of primes in y :

  $l_1 = 1, l_2 = 3, l_3 = 2, l_4 = 0, l_5 = 1, l_6 = 0$.
- Compute $y = -2^3 \cdot 3^2 \cdot 13 \bmod n = -936 \bmod 24961 = 24025$.
- Since $936 \equiv -24025 \pmod n$, consider another linear dependency.
- By inspection, $v_3 + v_6 + v_7 = 0$
- Compute $x = (a_3 a_6 a_7 \bmod n) = 23405$.
- Exponents $l_1 = 1, l_2 = 5, l_3 = 2, l_4 = 3, l_5 = 0, l_6 = 0$.
- Compute $y = (-2^5 \cdot 3^2 \cdot 5^3 \bmod n) = 13922$.
- Since $23405 \neq \pm 13922 \pmod n$, compute $\gcd(x - y, n)$

  $\gcd(9483, 24961) = 109$.
- Hence, two non-trivial factors of 24961 are 109 and 229.