Integer Factorization

Cryptanalysis

2016.09

Contents

Chapter 3

Section 3.5

POLLARD'S $p-1$ FACTORIZATION ALGORITHM

RSA cryptosystem

- The security of RSA relies on the apparent difficulty of factoring large numbers
  - To make RSA more efficient, we want to use a modulus $N = pq$ that is as small as possible
  - On the other hand, if an opponent can factor N, then our encrypted messages are not secure
- It is thus vital to understand how hard it is to factor large numbers

Pollard's $p-1$ method

- We are presented with a number $N = pq$ and our task is to determine the prime factors p and q
- Suppose we manage to find an integer L with the properties
  - $p-1$ divides L
  - $q-1$ does not divide L
- There are integers i, j, and k with $k \neq 0$ satisfying
  - $L = i(p-1)$
  - $L = j(q-1)+k$

Pollard's $p-1$ method

- Consider what happens if we take a randomly chosen integer a and compute a^L
- Fermat's little theorem tells us that

$$a^L = a^{i(p-1)} = (a^{p-1})^i \equiv 1^i \equiv 1 \pmod{p},$$
$$a^L = a^{j(q-1)+k} = a^k (a^{q-1})^j \equiv a^k \cdot 1^j \equiv a^k \pmod{q}.$$

- The exponent k is not equal to 0, so it is quite unlikely that a^k will be congruent to 1 modulo q

Pollard's $p-1$ method

- Thus with very high probability, i.e., for most choices of a, we find that
  - p divides $a^L - 1$
  - q does not divide $a^L - 1$
- This is wonderful, since it means that we can recover p via the simple gcd computation

$$p = \gcd(a^L - 1, N).$$

Pollard's $p-1$ method

- This is all well and good, but how to find an exponent L?
- Pollard's observation is that if $p-1$ happens to be a product of many small primes, then it will divide $n!$ for some not-too-large value of n
- For each number $n = 2, 3, 4, \ldots$, we choose a value of a and compute

$$\gcd(a^{n!} - 1, N)$$

Pollard's $p-1$ method

- If the gcd is equal to 1, then we go on to the next value of n
- If the gcd ever equals N, then we've been quite unlucky, but a different a value will probably work
- And if we get a number strictly between 1 and N, then we are done with a nontrivial factor of N

Pollard's $p-1$ method

- Remark. There are two important remarks to make before we put Pollard's idea into practice
- The first concerns the quantity $a^{n!} - 1$
  - Even for $a = 2, n = 100$, it is not feasible to compute $a^{n!} - 1$ exactly
  - The number $2^{100!}$ has more than 10^{157} digits, which is larger than the number of elementary particles in the known universe!

Pollard's $p-1$ method

- Luckily, we are interested only in the greatest common divisor of $a^{n!} - 1$ and N, so it suffices to compute

$$a^{n!} - 1 \pmod{N}$$

- Second, we do not even need to compute the exponent $n!$
  - Assuming $a^{n!} \pmod{N}$ is already computed in the previous step, we can compute the next value as

$$a^{(n+1)!} \equiv \left(a^{n!}\right)^{n+1} \pmod{N}$$

Pollard's $p-1$ method

Input. Integer N to be factored.

1. Set $a = 2$ (or some other convenient value).
2. Loop $j = 2, 3, 4, \ldots$ up to a specified bound.
   3. Set $a = a^j \bmod N$.
   4. Compute $d = \gcd(a-1, N)^\dagger$.
   5. If $1 < d < N$ then **success**, return d.
6. Increment j and loop again at Step **2**.

† For added efficiency, choose an appropriate k and compute the gcd in Step **4** only every kth iteration.

Pollard's $p-1$ method

- Example. We use Pollard's $p-1$ method to factor $N = 13927189$
- Starting with $\gcd(2^{9!}-1, N)$, we find that

$$2^{9!}-1 \equiv 13867883 \pmod{13927189}, \quad \gcd(2^{9!}-1, 13927189) = 1,$$
$$2^{10!}-1 \equiv 5129508 \pmod{13927189}, \quad \gcd(2^{10!}-1, 13927189) = 1,$$
$$2^{11!}-1 \equiv 4405233 \pmod{13927189}, \quad \gcd(2^{11!}-1, 13927189) = 1,$$
$$2^{12!}-1 \equiv 6680550 \pmod{13927189}, \quad \gcd(2^{12!}-1, 13927189) = 1,$$
$$2^{13!}-1 \equiv 6161077 \pmod{13927189}, \quad \gcd(2^{13!}-1, 13927189) = 1,$$
$$2^{14!}-1 \equiv 879290 \pmod{13927189}, \quad \gcd(2^{14!}-1, 13927189) = 3823.$$

Pollard's $p-1$ method

- We obtain a nontrivial factor $p = 3823$ of N
- The other factor $q = N/p = 13927189/3823 = 3643$
- The reason that an exponent of 14! worked in this instance is that $p-1$ factors into a product of small primes

$$p - 1 = 3822 = 2 \cdot 3 \cdot 7^2 \cdot 13$$

- The other satisfies $q - 1 = 3642 = 2 \cdot 3 \cdot 607$, which is not a product of small primes

Pollard's $p-1$ method

- Example. Let $N = 168441398857$

$$2^{50!}-1 \equiv 114787431143 \pmod{N}, \quad \gcd(2^{50!}-1, N) = 1,$$
$$2^{51!}-1 \equiv 36475745067 \pmod{N}, \quad \gcd(2^{51!}-1, N) = 1,$$
$$2^{52!}-1 \equiv 67210629098 \pmod{N}, \quad \gcd(2^{52!}-1, N) = 1,$$
$$2^{53!}-1 \equiv 8182353513 \pmod{N}, \quad \gcd(2^{53!}-1, N) = 350437.$$

- We find a prime factor $p = 350437$ of N, and the other factor is $q = 480661$
- Of course, $p - 1 = 2^2 \cdot 3 \cdot 19 \cdot 29 \cdot 53$ is a product of small factors

Pollard's $p-1$ method

- Remark. It is easy for Bob and Alice to avoid the dangers of Pollard's $p-1$ method
- Simply check their chosen secret primes p and q have the property that neither $p-1$ nor $q-1$ factors entirely into small primes
- The moral is that even if we build a cryptosystem based on a seemingly hard problem, we must be wary of special cases of the problem that, for subtle and nonobvious reasons, are easier to solve than the general case

Section 3.6

FACTORIZATION VIA DIFFERENCE OF SQUARES

Factorization via Difference of Squares

- In order to factor a number N, we look for an integer b such that the quantity $N+b^2$ is a perfect square, say equal to a^2, so

$$N = a^2 - b^2 = (a+b)(a-b)$$

and we have effected a factorization of N

Factorization via Difference of Squares

- Example. We factor $N = 25217$ by looking for an integer b making $N+b^2$ a perfect square:

$25217 + 1^2 = 25218$	not a square,
$25217 + 2^2 = 25221$	not a square,
$25217 + 3^2 = 25226$	not a square,
$25217 + 4^2 = 25233$	not a square,
$25217 + 5^2 = 25242$	not a square,
$25217 + 6^2 = 25253$	not a square,
$25217 + 7^2 = 25266$	not a square,
$25217 + 8^2 = 25281 = 159^2$	Eureka! ** square **.

Factorization via Difference of Squares

- Then we compute

$$25217 = 159^2 - 8^2 = (159+8)(159-8) = 167 \cdot 151$$

- If N is large, it is unlikely that a randomly chosen value of b will make $N + b^2$ into a perfect square
- An important observation is that it often suffices to write some multiple kN of N as a difference of two squares

Factorization via Difference of Squares

- If $kN = a^2 - b^2 = (a+b)(a-b)$, then there is a reasonable chance that N has a nontrivial factor in common with each of $a + b$ and $a - b$
- It is then a simple matter to recover the factors by computing $\gcd(N, a+b)$ and $\gcd(N, a-b)$

Factorization via Difference of Squares

- Example. $N = 203299$
- If we make a list of $N + b^2$ for values of $b = 1, 2, 3, \ldots$, say up to $b = 100$, we do not find any square values
- So next we try listing the values of $3N + b^2$ and we find

Factorization via Difference of Squares

$3 \cdot 203299 + 1^2 = 609898$	not a square,
$3 \cdot 203299 + 2^2 = 609901$	not a square,
$3 \cdot 203299 + 3^2 = 609906$	not a square,
$3 \cdot 203299 + 4^2 = 609913$	not a square,
$3 \cdot 203299 + 5^2 = 609922$	not a square,
$3 \cdot 203299 + 6^2 = 609933$	not a square,
$3 \cdot 203299 + 7^2 = 609946$	not a square,
$3 \cdot 203299 + 8^2 = 609961 = 781^2$	Eureka! ** square **.

Factorization via Difference of Squares

- Thus

$$3 \cdot 203299 = 781^2 - 8^2 = (781 + 8)(781 - 8) = 789 \cdot 773$$

- So we compute
  - $\gcd(203299, 789) = 263$
  - $\gcd(203299, 773) = 773$
- The numbers 263 and 773 are prime, so the full factorization of N is $203299 = 263 \cdot 773$

Factorization via Difference of Squares

- The multiples of N are the numbers that are congruent to 0 modulo N, so rather than searching for a difference of squares $a^2 - b^2$ that is a multiple of N, we may instead search for distinct numbers a and b satisfying

$$a^2 \equiv b^2 \pmod{N}$$

Factorization via Difference of Squares

- In practice it is not feasible to search directly for integers a and b satisfying

$$a^2 \equiv b^2 \pmod{N}$$

- Instead we use a three-step process
- This procedure, in one form or another, underlies most modern methods of factorization

Factorization via Difference of Squares

1. **Relation Building**: Find many integers $a_1, a_2, a_3, \ldots, a_r$ with the property that the quantity $c_i \equiv a_i^2 \pmod{N}$ factors as a product of small primes.
2. **Elimination**: Take a product $c_{i_1} c_{i_2} \cdots c_{i_s}$ of some of the c_i's so that every prime appearing in the product appears to an even power. Then $c_{i_1} c_{i_2} \cdots c_{i_s} = b^2$ is a perfect square.
3. **GCD Computation**: Let $a = a_{i_1} a_{i_2} \cdots a_{i_s}$ and compute the greatest common divisor $d = \gcd(N, a - b)$. Since

$$a^2 = (a_{i_1} a_{i_2} \cdots a_{i_s})^2 \equiv a_{i_1}^2 a_{i_2}^2 \cdots a_{i_s}^2 \equiv c_{i_1} c_{i_2} \cdots c_{i_s} \equiv b^2 \pmod{N},$$

there is a reasonable chance that d is a nontrivial factor of N.

Factorization via Difference of Squares

- Example. We factor $N = 914387$ using the procedure
- We first search for integers a with the property that $a^2 \bmod N$ is a product of small primes
- For this example, we ask that each $a^2 \bmod N$ be a product of primes in the set $\{2, 3, 5, 7, 11\}$

Factorization via Difference of Squares

- We observe that

$$1869^2 \equiv 750000 \pmod{914387} \quad \text{and} \quad 750000 = 2^4 \cdot 3 \cdot 5^6,$$
$$1909^2 \equiv 901120 \pmod{914387} \quad \text{and} \quad 901120 = 2^{14} \cdot 5 \cdot 11,$$
$$3387^2 \equiv 499125 \pmod{914387} \quad \text{and} \quad 499125 = 3 \cdot 5^3 \cdot 11^3.$$

- None of the numbers on the right is a square, but if we multiply them together, then we do get a square

$$1869^2 \cdot 1909^2 \cdot 3387^2 \equiv 164255^2 \ (mod\ N)$$

Factorization via Difference of Squares

- Notice that

$$1869 \cdot 1909 \cdot 3387 \equiv 9835 \ (mod\ 914387)$$

- So we compute

$$\gcd(914387, 9835 - 164255)$$
$$= \gcd(914387, 154420) = 1103.$$

- We have factored $914387 = 1103 \cdot 829$

Factorization via Difference of Squares

- Example. We do a second example to illustrate a potential pitfall in this method
- $N = 636683$
- After some searching, we find

$$1387^2 \equiv 13720 \pmod{636683} \quad \text{and} \quad 13720 = 2^3 \cdot 5 \cdot 7^3,$$
$$2774^2 \equiv 54880 \pmod{636683} \quad \text{and} \quad 54880 = 2^5 \cdot 5 \cdot 7^3.$$

- Multiplying these two values gives a square

$$1387^2 \cdot 2774^2 \equiv 27440^2 \ (mod\ N)$$

Factorization via Difference of Squares

- Unfortunately, when we compute the gcd
  $\gcd(636683, 1387 \cdot 2774 - 27440) = 636683$
- After all our work, we have made no progress!
- We can gather more values of a and try to find a different relation
- Extending the above list, we discover that

$3359^2 \equiv 459270 \pmod{636683}$ and $459270 = 2 \cdot 3^8 \cdot 5 \cdot 7.$

Factorization via Difference of Squares

- Thus we have
  $1387^2 \cdot 3359^2 \equiv 79380^2 \ (mod\ N)$
- And
  $\gcd(636683, 1387 \cdot 3359 - 79380) = 787$
- This gives the factorization $N = 787 \cdot 809$

Factorization via Difference of Squares

- Remark. How many solutions to
  $a^2 \equiv b^2 \ (mod\ N)$
  are we likely to try before we find a factor of N?
- We know that
  $(a-b)(a+b) = a^2 - b^2 = kN = kpq$
  for some value of k
- The prime p must divide at least one of $a-b$ and $a+b$, and it has approximately equal probability of dividing each. Similarly for q.

Factorization via Difference of Squares

- We win if $a-b$ is divisible by exactly one of p and q, which happens approximately 50% of the time
- Hence if we can actually generate random a's and b's satisfying $a^2 \equiv b^2 \ (mod\ N)$, then it won't take us long to find a factor of N
- Of course this leaves us with the question of just how hard it is to find these a's and b's

Factorization via Difference of Squares

- The factorization procedure consists of three steps:

1. Relation Building
2. Elimination
3. GCD Computation

- In Step 3, $\gcd(N, a-b)$ can be computed in $\mathcal{O}(\ln N)$ steps using the Euclidean algorithm
- We need more tools to analyze Step 1

Factorization via Difference of Squares

- Suppose each of the numbers $a_1, \ldots, a_r$ found in Step 1 has the property that $c_i \equiv a_i^2 \pmod{N}$ factors into a product of small primes, for example, the first t primes $\{p_1, p_2, \ldots, p_t\}$
- This means that there are exponents e_{ij} such that

$$\begin{aligned} c_1 &= p_1^{e_{11}} p_2^{e_{12}} p_3^{e_{13}} \cdots p_t^{e_{1t}}, \\ c_2 &= p_1^{e_{21}} p_2^{e_{22}} p_3^{e_{23}} \cdots p_t^{e_{2t}}, \\ &\vdots \qquad\qquad \vdots \\ c_r &= p_1^{e_{r1}} p_2^{e_{r2}} p_3^{e_{r3}} \cdots p_t^{e_{rt}}. \end{aligned}$$

Factorization via Difference of Squares

- Our problem is finding $u_1, u_2, \ldots, u_r \in \{0, 1\}$ such that

$$c_1^{u_1} \cdot c_2^{u_2} \cdots c_r^{u_r}$$

is a perfect square

- Using summation and product notation, we may write

$$\prod_{i=1}^{r} c_i^{u_i} = \prod_{j=1}^{t} p_j^{\sum_{i=1}^{r} e_{ij} u_i}$$

our goal is to choose $u_1, u_2, \ldots, u_r$ such that all of the exponents are even

Factorization via Difference of Squares

- To recapitulate, we are given integers $\{e_{ij}\}$, and we are searching for integers $u_1, u_2, \ldots, u_r$ such that

$$\begin{aligned} e_{11}u_1 + e_{21}u_2 + \cdots + e_{r1}u_r &\equiv 0 \pmod 2, \\ e_{12}u_1 + e_{22}u_2 + \cdots + e_{r2}u_r &\equiv 0 \pmod 2, \\ &\vdots \qquad\qquad \vdots \\ e_{1t}u_1 + e_{2t}u_2 + \cdots + e_{rt}u_r &\equiv 0 \pmod 2. \end{aligned}$$

Factorization via Difference of Squares

- This is a system of linear equations over the finite field $\mathbb{F}_2$
- Hence standard techniques from linear algebra, such as Gaussian elimination, can be used to solve these equations
- Remark. In order to factor a large number N, it may be necessary to use a set $\{p_1, p_2, \ldots, p_t\}$ containing hundreds of thousands, or even millions, of primes

Factorization via Difference of Squares

- Then the system contains millions of linear equations, and being very difficult to solve
- However, it turns out that the systems of linear equations used in factorization are quite sparse
- There are special techniques for solving sparse systems of linear equations that are much more efficient than ordinary Gaussian elimination

Section 3.7

SMOOTH NUMBERS, SIEVES, AND BUILDING RELATIONS FOR FACTORIZATION

Smooth Numbers

- **Definition**. An integer n is called B-smooth if all of its prime factors are less than or equal to B
- Example. The first few 5-smooth numbers and the first few numbers that are not 5-smooth:
  - 5-smooth: 2, 3, 4, 5, 6, 8, 9, 10, 12, 15,…
  - Not 5-smooth: 7, 11, 13, 14, 17, 19,…
- **Definition**. The function $\psi(X, B)$ counts B-smooth numbers, $\psi(X, B) =$ Number of B-smooth integers n such that $1 < n \leq X$

Smooth Numbers

- Example. $\psi(25,5) = 15$
  - Since the 5-smooth numbers between 1 and 25 are the 15 numbers
  - 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25
- In order to evaluate the efficiency of the three step factorization method, we need to understand how $\psi(X,B)$ behaves for large values of X and B

Smooth Numbers

- **Theorem** (Canfield, Erdős, Pomerance).
- Fix a number $0 < \epsilon < \frac{1}{2}$, and let X and B increase together while satisfying

$$(\ln X)^{\epsilon} < \ln B < (\ln X)^{1-\epsilon}$$

- Then the number of B-smooth numbers less than X satisfies

$$\psi(X,B) = X \cdot u^{-u(1+o(1))}$$

where $u = \frac{\ln X}{\ln B}$

Smooth Numbers

- Remark (little-o notation).
- The expression $o(1)$ denotes a function that tends to 0 as X tends to infinity
- More generally, we write

$$f(X) = o\big(g(X)\big)$$

if the ratio $f(X)/g(X)$ tends to 0 as X tends to infinity

Smooth Numbers

- The question remains of how we should choose B in terms of X
- It turns out that the following curious-looking function $L(X)$ is what we will need:

$$L(X) = e^{\sqrt{(\ln X)(\ln\ln X)}}$$

- **Corollary**. For any fixed value of c with $0 < c < 1$,

$$\psi\big(X, L(X)^c\big) = X \cdot L(X)^{-(1/2c)(1+o(1))}$$

as $X \to \infty$

Smooth Numbers

- *Proof.* If $B = L(X)^c$ and if we take any $\epsilon < \frac{1}{2}$, then
$$\ln B = c \ln L(X) = c\sqrt{(\ln X)(\ln \ln X)}$$
satisfies $(\ln X)^\epsilon < \ln B < (\ln X)^{1-\epsilon}$
- Apply Theorem with
$$u = \frac{\ln X}{\ln B} = \frac{1}{c} \cdot \sqrt{\frac{\ln X}{\ln \ln X}}$$

Smooth Numbers

- It is easily checked (exercise) that this value of u satisfies
$$u^{-u(1+o(1))} = L(X)^{-(1/2c)(1+o(1))}$$
which completes the proof of the corollary

3.32. This exercise asks you to verify an assertion in the proof of Corollary 3.45. Let $L(X)$ be the usual function $L(X) = e^{\sqrt{(\ln X)(\ln \ln X)}}$.

(a) Prove that there is a value of $\epsilon > 0$ such that
$$(\ln X)^\epsilon < \ln L(X) < (\ln X)^{1-\epsilon} \qquad \text{for all } X > 10.$$

(b) Let $c > 0$, let $Y = L(X)^c$, and let $u = (\ln X)/(\ln Y)$. Prove that
$$u^{-u} = L(X)^{-\frac{1}{2c}(1+o(1))}.$$

Smooth Numbers

- The function $L(X) = e^{\sqrt{(\ln X)(\ln \ln X)}}$ and other similar functions appear prominently in the theory of factorization due to their close relationship to the distribution of smooth numbers
- It is thus important to understand how fast $L(X)$ grows as a function of X
- As a supplement to big-$\mathcal{O}$ notation, it is convenient to introduce two other ways of comparing the rate at which functions grow

Smooth Numbers

- **Definition** (Order Notation).
- Let $f(X)$ and $g(X)$ be functions of X whose values are positive
- We say that f is big-Ω of g and write
$$f(X) = \Omega\big(g(X)\big)$$
if there are positive constants c and C such that
$$f(X) \geq cg(X) \qquad \text{for all } X \geq C$$
- If f is both big-$\mathcal{O}$ and big-Ω of g, we say that f is big-Θ of g, and write $f(X) = \Theta\big(g(X)\big)$

Smooth Numbers

- Remark. In analytic number theory there is an alternative version of order notation that is quite intuitive
- For functions $f(X)$ and $g(X)$, we write

$$f(X) \ll g(X) \quad \text{if } f(X) = \mathcal{O}(g(X)),$$
$$f(X) \gg g(X) \quad \text{if } f(X) = \Omega(g(X)),$$
$$f(X) \gg\ll g(X) \quad \text{if } f(X) = \Theta(g(X)).$$

- The advantage of this notation is that it is transitive: if $f \gg g$ and $g \gg h$, then $f \gg h$

Smooth Numbers

- **Definition**. A function $f(X)$ is said to grow *exponentially* if there are positive constants α and β such that

$$\Omega(X^{\alpha}) = f(X) = \mathcal{O}(X^{\beta})$$

and it is said to grow *polynomially* if there are positive constants α and β such that

$$\Omega\big((\ln X)^{\alpha}\big) = f(X) = \mathcal{O}\big((\ln X)^{\beta}\big)$$

Smooth Numbers

- In the alternative notation, exponential and polynomial growth are written, respectively, as $X^{\alpha} \ll f(X) \ll X^{\beta}$ and $(\ln X)^{\alpha} \ll f(X) \ll (\ln X)^{\beta}$
- A function that falls in between these two categories is called *subexponential*
- Thus $f(X)$ is subexponential if for every positive constant α, no matter how large, and for every positive constant β, no matter how small,

$$\Omega\big((\ln X)^{\alpha}\big) = f(X) = \mathcal{O}(X^{\beta})$$

Smooth Numbers

- Remark. The function $L(X)$ falls into the subexponential category

X	$\ln L(X)$	$L(X)$
2^{100}	17.141	$2^{24.73}$
2^{250}	29.888	$2^{43.12}$
2^{500}	45.020	$2^{64.95}$
2^{1000}	67.335	$2^{97.14}$
2^{2000}	100.145	$2^{144.48}$

Smooth Numbers

- Suppose that we attempt to factor N by searching for values $a^2 \pmod N$ that are B-smooth
- Since the smooth numbers correspond to the variables, while the primes less than B correspond to the equations, we need (at least) as many B-smooth numbers as there are primes less than B
- We thus need at least $\pi(B)$ B-smooth numbers, where $\pi(B)$ is the number of primes up to B

Smooth Numbers

- It will turn out that we can take $B = L(N)^c$ for a suitable value of c
- In the next proposition we use the formula for $\psi(X, L(X)^c)$ and the prime number theorem to choose the smallest value of c
- **Theorem** (The Prime Number Theorem).

$$\lim_{X\to\infty} \frac{\pi(X)}{X/\ln(X)} = 1$$

Smooth Numbers

- **Proposition**. Let $L(X) = e^{\sqrt{(\ln X)(\ln\ln X)}}$, let N be a large integer, and set $B = L(N)^{\frac{1}{\sqrt{2}}}$
  a) We expect to check approximately $L(N)^{\sqrt{2}}$ random numbers modulo N in order to find $\pi(B)$ numbers that are B-smooth
  b) We expect to check approximately $L(N)^{\sqrt{2}}$ random numbers of the form $a^2 \pmod N$ in order to find enough B-smooth numbers to factor N
- Hence the factorization procedure should have a subexponential running time

Smooth Numbers

- *Proof*. We already explained why (a) and (b) are equivalent, assuming that the numbers $a^2 \pmod N$ are sufficiently random. We now prove (a)
- The probability that a randomly chosen number modulo N is B-smooth is $\psi(N, B)/N$
- In order to find $\pi(B)$ numbers that are B-smooth, we need to check approximately $\frac{\pi(B)}{\psi(N,B)/N}$ numbers

Smooth Numbers

- We want to choose B so as to minimize this function, since checking numbers for smoothness is a time-consuming process
- Corollary says that

$$\psi(N, L(N)^c)/N \approx L(N)^{-1/2c}$$

so we set $B = L(N)^c$ and search for the value of c

Smooth Numbers

- The prime number theorem tells us that

$$\pi(B) \approx \frac{B}{\ln(B)}$$

- So that

$$\frac{\pi(L(N)^c)}{\psi(N, L(N)^c)/N} \approx \frac{L(N)^c}{c \ln L(N)} \cdot \frac{1}{L(N)^{-1/2c}} = L(N)^{c+1/2c} \cdot \frac{1}{c \ln L(N)}$$

- The factor $L(N)^{c+\frac{1}{2c}}$ dominates this last expression, so we choose the value of c that minimizes the quantity $c + \frac{1}{2c}$

Smooth Numbers

- It is minimized when $c = \frac{1}{\sqrt{2}}$, and the minimum value is $\sqrt{2}$
- Thus if we choose $B \approx L(N)^{\frac{1}{\sqrt{2}}}$, then we need to check approximately $L(N)^{\sqrt{2}}$ values in order to find $\pi(B)$ numbers that are B-smooth, and hence to find enough relations to factor N

Smooth Numbers

- Remark. Proposition suggests that we need to check approximately $L(N)^{\sqrt{2}}$ *randomly chosen* numbers modulo N in order to find enough smooth numbers to factor N
- In particular, it suffices to check approximately $L(N)$ random numbers of the form $a^2 \ (mod\ N)$ with a close to $\sqrt{N}$

Smooth Numbers

- Remark. When estimating the effort needed to factor N, we have completely ignored the work required to check whether a given number is B-smooth
- Taking this additional effort into account, one finds that it takes approximately $L(N)^{\sqrt{2}}$ trial divisions to find enough smooth numbers to factor N, even using values of $a \approx \sqrt{N}$

Smooth Numbers

- The quadratic sieve uses a more efficient method for generating B-smooth numbers and thereby reduces the running time down to $L(N)$
- The number field sieve, moving beyond the realm of the ordinary integers, achieves running time of $e^{c\sqrt[3]{(\ln N)(\ln \ln N)^2}}$, faster than $L(N)^{\epsilon}\ \forall\, \epsilon > 0$

The Quadratic Sieve

- We know that we need to take $B \approx L(N)^{\frac{1}{\sqrt{2}}}$ in order to have a reasonable chance of factoring N

> How can we efficiently find many numbers $a > \sqrt{N}$ such that each $a^2 \pmod N$ is B-smooth?

- An early approach to finding B-smooth squares modulo N was to look for fractions $\frac{a}{b}$ that are as close as possible to $\sqrt{kN}$ for $k = 1, 2, 3, \ldots$

The Quadratic Sieve

- Then $a^2 \approx b^2 kN$, so c is reasonably small, and thus is more likely to be B-smooth
- The theory of *continued fractions* gives an algorithm for finding such $\frac{a}{b}$
- An alternative approach that turns out to be much faster in practice is to allow slightly larger values of a and to use an efficient cancellation process called a ***sieve*** to simultaneously create a large number of values $a^2\ (mod\ N)$ that are B-smooth

The Quadratic Sieve

- The Pomerance's *quadratic sieve* is still the fastest known method for factoring large numbers $N = pq$ up to about 2^{350}
- For numbers considerably larger than this, say larger than 2^{450}, the more complicated *number field sieve* holds the world record for quickest factorization

The Quadratic Sieve

- What we need is a list of numbers of the form $a^2 \pmod N$ that are B-smooth
- Our strategy for accomplishing this uses the polynomial $F(T) = T^2 - N$
- We want to start with a value of a that is slightly larger than $\sqrt{N}$, so we set $a = \lfloor \sqrt{N} \rfloor + 1$
- We then look at the list of numbers

$$F(a), F(a+1), F(a+2), \ldots, F(b)$$

The Quadratic Sieve

- The idea is to find the B-smooth numbers in this list by sieving away the primes smaller than B and seeing which numbers in the list get sieved all the way down to 1
- We choose B sufficiently large so that, by the end of the sieving process, we are likely to have found enough B-smooth numbers to factor N

The Quadratic Sieve

- **Definition**. The set of primes less than B (or sometimes the set of prime powers less than B) is called the *factor base*
- Suppose that p is a prime in our factor base
- Which numbers t between a and b satisfy

$$t^2 \equiv N \pmod p ?$$

The Quadratic Sieve

- If the congruence has no solutions, then we discard the prime p
- Otherwise the congruence has two solutions, which we denote by
$$t = \alpha_p, \beta_p$$
- If $p = 2$, there is only one solution α_p

The Quadratic Sieve

- It follows that each of the numbers
$$F(\alpha_p), F(\alpha_p + p), F(\alpha_p + 2p), F(\alpha_p + 3p), \ldots$$
$$F(\beta_p), F(\beta_p + p), F(\beta_p + 2p), F(\beta_p + 3p), \ldots$$
is divisible by p
- Thus we can sieve away a factor of p from every p-th entry in the list
$$F(a), F(a+1), F(a+2), \ldots, F(b)$$

The Quadratic Sieve

- Example. We illustrate the quadratic sieve applied to the composite number $N = 221$
- The smallest number whose square is larger than N is $a = \lfloor \sqrt{221} \rfloor + 1 = 15$
- We set $F(T) = T^2 - 221$ and sieve the numbers from $F(15) = 4$ up to $F(30) = 679$ using successively the prime powers from 2 to 7

The Quadratic Sieve

- The initial list of numbers $T^2 - N$ is

4	35	68	103	140	179	220	263	308	355	404	455	508	563	620	679

- We first sieve by $p = 2$, which means that we cancel 2 from every second entry in the list

4	35	68	103	140	179	220	263	308	355	404	455	508	563	620	679
$\downarrow^2$		$\downarrow^2$		$\downarrow^2$		$\downarrow^2$		$\downarrow^2$		$\downarrow^2$		$\downarrow^2$		$\downarrow^2$	
2	35	34	103	70	179	110	263	154	355	202	455	254	563	310	679

- Next we sieve by $p = 3$, but the congruence
$$t^2 \equiv 221 \equiv 2 \ (mod\ 3)$$
has no solution. None of the entries are divisible

The Quadratic Sieve

- We move on to the prime power 2^2
- Every odd number is a solution of the congruence

$$t^2 \equiv 221 \equiv 1 \,(mod\, 4)$$

which means that we can sieve another factor of 2 from every second entry in our list

2	35	34	103	70	179	110	263	154	355	202	455	254	563	310	679
$\downarrow^4$		$\downarrow^4$		$\downarrow^4$		$\downarrow^4$		$\downarrow^4$		$\downarrow^4$		$\downarrow^4$		$\downarrow^4$	
1	35	17	103	35	179	55	263	77	355	101	455	127	563	155	679

- The small 4 is to indicate sieving by 4, although cancel only a factor of 2 from each entry

The Quadratic Sieve

- Next we move on to $p = 5$. The congruence

$$t^2 \equiv 221 \equiv 1 \,(mod\, 5)$$

has two solutions, $\alpha_5 = 1$ and $\beta_5 = 4$ modulo 5

- The first t value in our list that is congruent to 1 modulo 5 is $t = 16$, so starting with $F(16)$, we find that every fifth entry is divisible by 5

1	35	17	103	35	179	55	263	77	355	101	455	127	563	155	679
	$\downarrow^5$					$\downarrow^5$					$\downarrow^5$				
1	7	17	103	35	179	11	263	77	355	101	91	127	563	155	679

The Quadratic Sieve

- Similarly, every fifth entry starting with $F(19)$ is divisible by 5, so we sieve out those factors

1	7	17	103	35	179	11	263	77	355	101	91	127	563	155	679
				$\downarrow^5$					$\downarrow^5$					$\downarrow^5$	
1	7	17	103	7	179	11	263	77	71	101	91	127	563	31	679

- For sieving the prime $p = 7$. The congruence

$$t^2 \equiv 221 \equiv 4 \,(mod\, 7)$$

has the two solutions $\alpha_7 = 2$ and $\beta_7 = 5$

The Quadratic Sieve

- This yields

1	7	17	103	7	179	11	263	77	71	101	91	127	563	31	679
	$\downarrow^7$							$\downarrow^7$							$\downarrow^7$
1	1	17	103	7	179	11	263	11	71	101	91	127	563	31	97
				$\downarrow^7$							$\downarrow^7$				
1	1	17	103	1	179	11	263	11	71	101	13	127	563	31	97

- Notice that the original entries

$$F(15) = 4, F(16) = 35, F(19) = 140$$

have been sieved all the way down to 1

The Quadratic Sieve

- So we have discovered several squares modulo 221 that are products of small primes:

$$15^2 \equiv 2^2 \pmod{221},$$
$$16^2 \equiv 5 \cdot 7 \pmod{221},$$
$$19^2 \equiv 2^2 \cdot 5 \cdot 7 \pmod{221}.$$

- We can obtain relations between squares

$$(16 \cdot 19)^2 \equiv (2 \cdot 5 \cdot 7)^2 \pmod{221}$$

The Quadratic Sieve

- Computing

$$\gcd(221, 16 \cdot 19 - 2 \cdot 5 \cdot 7) = \gcd(221, 234) = 13$$

gives a nontrivial factor of 221

- If we continue sieving up to $B = 11$, we will obtain two additional relations

$$F(21) \equiv 21^2 \equiv 2^2 \cdot 5 \cdot 11 \pmod{221}$$
$$F(23) \equiv 23^2 \equiv 2^2 \cdot 7 \cdot 11 \pmod{221}$$

The Quadratic Sieve

15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
4	35	68	103	140	179	220	263	308	355	404	455	508	563	620	679
$\downarrow 2$		$\downarrow 2$		$\downarrow 2$		$\downarrow 2$		$\downarrow 2$		$\downarrow 2$		$\downarrow 2$		$\downarrow 2$	
2	35	34	103	70	179	110	263	154	355	202	455	254	563	310	679
$\downarrow 4$		$\downarrow 4$		$\downarrow 4$		$\downarrow 4$		$\downarrow 4$		$\downarrow 4$		$\downarrow 4$		$\downarrow 4$	
1	35	17	103	35	179	55	263	77	355	101	455	127	563	155	679
	$\downarrow 5$					$\downarrow 5$					$\downarrow 5$				
1	7	17	103	35	179	11	263	77	355	101	91	127	563	155	679
				$\downarrow 5$					$\downarrow 5$					$\downarrow 5$	
1	7	17	103	7	179	11	263	77	71	101	91	127	563	31	679
	$\downarrow 7$							$\downarrow 7$							$\downarrow 7$
1	1	17	103	7	179	11	263	11	71	101	91	127	563	31	97
				$\downarrow 7$							$\downarrow 7$				
1	1	17	103	1	179	11	263	11	71	101	13	127	563	31	97
								$\downarrow 11$							
1	1	17	103	1	179	11	263	1	71	101	13	127	563	31	97
						$\downarrow 11$									
1	1	17	103	1	179	1	263	1	71	101	13	127	563	31	97

The Quadratic Sieve

- We can combine these relations with the earlier relations to obtain new square equalities:

$$(19 \cdot 21 \cdot 23)^2 \equiv (2^3 \cdot 5 \cdot 7 \cdot 11)^2 \pmod{221}$$

- These give another way to factor 221:

$$\gcd(221, 19 \cdot 21 \cdot 23 - 2^3 \cdot 5 \cdot 7) = \gcd(221, 6097) = 13$$

The Quadratic Sieve

- Remark. If p is an odd prime, then the congruence $t^2 \equiv N \ (mod\ p)$ has either 0 or 2 solutions modulo p
- More generally, congruences $t^2 \equiv N \ (mod\ p^e)$ have either 0 or 2 solutions
- This makes sieving odd prime powers relatively straightforward

The Quadratic Sieve

- Sieving with powers of 2 is a bit trickier, since the number of solutions may be different modulo 2, modulo 4, and modulo higher powers of 2
- So although sieving powers of 2 is not intrinsically difficult, it must be dealt with as a special case

The Quadratic Sieve

- Remark. There are many implementation ideas that can be used to greatly increase the practical speed of the quadratic sieve
- Although the running time of the sieve remains a constant multiple of $L(N)$, the multiple can be significantly reduced
- A time-consuming part of the sieve is the necessity of dividing every p-th entry by p, since if the numbers are large, division by p is moderately complicated

The Quadratic Sieve

- A key idea to speed up this step is to use approximate logarithms, which allows the slower division operations to be replaced by faster subtraction operations
- Instead of using the list of values

$$F(a),\ F(a+1),\ F(a+2),\ \ldots,$$

we use a list of integer values that are approximately equal to

$$\log F(a),\ \log F(a+1),\ \log F(a+2),\ \log F(a+3),\ \ldots$$

The Quadratic Sieve

- In order to sieve p from $F(t)$, we subtract an integer approximation of $\log p$ from the integer approximation to $\log F(t)$, since by the rule of logarithms

$$\log F(t) - \log p = \log \frac{F(t)}{p}$$

- At the end of the sieving process, the entries that are reduced to 0 would be precisely the values of $F(t)$ that are B-smooth

The Quadratic Sieve

- However, since we use only approximate logarithm values, at the end we look for entries that have been reduced to a small number
- Then we use division on only those few entries to find the ones that are actually B-smooth
- A second idea that can be used to speed the quadratic sieve is to use the polynomial

$$F(t) = t^2 - N$$

only until t reaches a certain size, and then replace it with a new polynomial

The Number Field Sieve

- The number field sieve is a factorization method that works in a ring that is larger than the ordinary integers
- In order to factor N, we start by finding a nonzero integer m and an irreducible monic polynomial $f(x) \in \mathbb{Z}[x]$ of small degree satisfying

$$f(m) \equiv 0 \ (mod\ N)$$

The Number Field Sieve

- Example. Suppose that we want to factor the number $N = 2^{2^9} + 1$
- Then we could take $m = 2^{103}$ and $f(x) = x^5 + 8$, since

$$f(m) = f(2^{103}) = 2^{515} + 8$$
$$= 8(2^{512} + 1) \equiv 0 \pmod{2^{2^9} + 1}$$

The Number Field Sieve

- Let d be the degree of $f(x)$ and β be a root of $f(x)$
- Note that β might be a complex number
- We will work in the ring

$$\mathbb{Z}[\beta] = \{c_0 + c_1\beta + c_2\beta^2 + \cdots + c_{d-1}\beta^{d-1} \in \mathbb{C} : c_0, c_1, \ldots, c_{d-1} \in \mathbb{Z}\}$$

- Although we have written $\mathbb{Z}[\beta]$ as a subring of the complex numbers, we can work with $\mathbb{Z}[\beta]$ purely algebraically, since it is equal to the quotient ring

$$\mathbb{Z}[\beta]/(f(x))$$

The Number Field Sieve

- Example. Let $f(x) = 1 + 3x - 2x^3 + x^4$, let β be a root of $f(x)$, and consider the ring $\mathbb{Z}[\beta]$
- In order to add the elements

$$u = 2 - 4\beta + 7\beta^2 + 3\beta^3$$
$$v = 1 + 2\beta - 4\beta^2 - 2\beta^3$$

we simply add their coefficients

$$u + v = 3 - 2\beta + 3\beta^2 + \beta^3$$

The Number Field Sieve

- Multiplication is a bit more complicated
- First we multiply u and v, treating β as if it were a variable

$$uv = 2 - 9\beta^2 + 29\beta^3 - 14\beta^4 - 26\beta^5 - 6\beta^6$$

- Then we divide by $f(\beta) = 1 + 3\beta - 2\beta^3 + \beta^4$, still treating β as a variable, and keep the remainder

$$uv = 92 + 308\beta + 111\beta^2 - 133\beta^3 \in \mathbb{Z}[\beta]$$

The Number Field Sieve

- The next step in the number field sieve is to find a large number of pairs of integers

$$(a_1, b_1), \ldots, (a_k, b_k)$$

that simultaneously satisfy

$$\prod_{i=1}^{k} (a_i - b_i m) \text{ is a square in } \mathbb{Z}$$

and

$$\prod_{i=1}^{k} (a_i - b_i \beta) \text{ is a square in } \mathbb{Z}[\beta]$$

The Number Field Sieve

- There is an integer $A \in \mathbb{Z}$ and an element $\alpha \in \mathbb{Z}[\beta]$ such that

$$\prod_{i=1}^{k}(a_i - b_i m) = A^2$$

and

$$\prod_{i=1}^{k}(a_i - b_i \beta) = \alpha^2$$

The Number Field Sieve

- By definition of $\mathbb{Z}[\beta]$, we can find an expression for α of the form

$$\alpha = c_0 + c_1\beta + c_2\beta^2 + \cdots + c_{d-1}\beta^{d-1}$$

- Recall our original assumption

$$f(m) \equiv 0 \ (mod\ N)$$

- This means that we have

$$m \equiv \beta \pmod{N}$$

in the ring $\mathbb{Z}[\beta]$

The Number Field Sieve

- So on the one hand

$$A^2 \equiv \alpha^2 \pmod{N}$$

in the ring $\mathbb{Z}[\beta]$

- On the other hand

$$\alpha \equiv c_0 + c_1 m + c_2 m^2 + \cdots + c_{d-1} m^{d-1} \pmod{N}$$

in the ring $\mathbb{Z}[\beta]$

The Number Field Sieve

- Hence

$$A^2 \equiv (c_0 + c_1 m + c_2 m^2 + \cdots + c_{d-1} m^{d-1})^2 \pmod{N}$$

- Thus we have created a congruence

$$A^2 \equiv B^2 \ (mod\ N)$$

that is valid in the ring of integers $\mathbb{Z}$

- As usual, there is then a good chance that $\gcd(A - B, N)$ will yield a nontrivial factor of N

The Number Field Sieve

- How do we find the (a_i, b_i) pairs to make both of the products into squares?
- For the first product, we can use a sieve-type algorithm to find values of $a - bm$ that are smooth, then use linear algebra to find a subset with the desired property
- Pollard's idea is to simultaneously do something similar for the second product while working in the ring $\mathbb{Z}[\beta]$

The Number Field Sieve

- Thus we look for pairs of integers (a, b) such that the quantity $a - b\beta$ is "smooth" in $\mathbb{Z}[\beta]$
- There are many serious issues that arise when we try to do this, including the following:

1. The ring $\mathbb{Z}[\beta]$ usually does not have unique factorization of elements into primes or irreducible elements. So instead, we factor the ideal $(a - b\beta)$ into a product of prime ideals. We say that $a - b\beta$ is smooth if the prime ideals appearing in the factorization are small.

The Number Field Sieve

2. Unfortunately, even ideals in the ring $\mathbb{Z}[\beta]$ may not have unique factorization as a product of prime ideals. However, there is a slightly larger ring, called the ring of integers of $\mathbb{Q}(\beta)$, in which unique factorization of ideals is true
3. Suppose that we have managed to make the ideal $(\prod(a_i - b_i\beta))$ into the square of an ideal in $\mathbb{Z}[\beta]$, it need not be the square of an ideal generated by a single element. Even if it is equal to an ideal of the form $(\gamma)^2$, we can conclude only that $\prod(a_i - b_i\beta) = u\gamma^2$ for some unit $u \in \mathbb{Z}[\beta]^*$

The Number Field Sieve

- It would take us too far afield to explain how to deal with these potential difficulties
- Through a number of ingenious ideas due to Adleman, Buhler, H. Lenstra, Pomerance, and others, the obstacles were overcome, leading to a practical factorization method

The Number Field Sieve

- However, we will comment further on the first step in the algorithm. In order to get started, we need an integer m and a monic irreducible polynomial $f(x)$ of small degree such that
$$f(m) \equiv 0 \ (mod\ N)$$
- The trick is first to choose the desired degree d of f, next to choose an integer m satisfying
$$(N/2)^{1/d} < m < N^{1/d}$$

The Number Field Sieve

- Then to write N as a number to the base m
$$N = c_0 + c_1 m + c_2 m^2 + \cdots + c_{d-1} m^{d-1} + c_d m^d$$
with $0 \leq c_i < m$
- The condition on m ensures that $c_d = 1$, so we can take f to be the monic polynomial
$$f(x) = c_0 + c_1 x + c_2 x^2 + \cdots + c_{d-1} x^{d-1} + x^d$$

The Number Field Sieve

- We also need $f(x)$ to be irreducible, but if $f(x)$ factors in $\mathbb{Z}[\beta]$, say $f(x) = g(x)h(x)$, then
$$N = f(m) = g(m)h(m)$$
gives a factorization of N
- So now we have an $f(x)$ and an m, which allows us to get started using the number field sieve

The Number Field Sieve

- There is no denying the fact that the number field sieve is much more complicated than the quadratic sieve. So why is it useful?
- The reason has to do with the size of the numbers that must be considered
- For the quadratic sieve, we sieved to find smooth numbers of the form
$$\left(\lfloor \sqrt{N} \rfloor + k\right)^2 - N$$
for $k = 1,2,...$

The Number Field Sieve

- So we needed to pick out the smooth numbers from a set of numbers whose size is a little larger than $\sqrt{N}$
- For the number field sieve one ends up looking for smooth numbers of the form

$$(a - mb) \cdot b^d f(a/b)$$

- By a judicious choice of m and f, these numbers are much smaller than $\sqrt{N}$

The Number Field Sieve

- In order to describe how much smaller, we use a generalization of the subexponential function
- **Definition**. For any $0 < \epsilon < 1$, we define the function

$$L_\epsilon(X) = e^{(\ln X)^\epsilon (\ln \ln X)^{1-\epsilon}}$$

- **Theorem**. Under some reasonable assumptions, the expected running time of the number field sieve to factor the number N is $L_{1/3}(N)^c$ for a small value of c

The Number Field Sieve

- For general numbers, the best known value of c in Theorem is a bit less than 2, while for special numbers such as $2^{2^9} + 1$ it is closer to 1.5
- As a practical matter, the quadratic sieve is faster for numbers smaller than 10^{100}, while the number field sieve is faster for numbers larger than 10^{130}

Section 3.8

THE INDEX CALCULUS METHOD

The Index Calculus Method

- The index calculus is a method for solving the discrete logarithm problem in a finite field $\mathbb{F}_p$
- The algorithm uses smooth numbers and bears some similarity to the sieve methods that we have studied in this chapter, which is why we cover it here

The Index Calculus Method

- To solve the discrete logarithm problem

$$g^x \equiv h \pmod{p}$$

where p, g, h are given
- For simplicity, we will assume that g is a primitive root modulo p, so its powers give all of $\mathbb{F}_p^*$
- We choose a value B and solve the discrete logarithm problem

$$g^x \equiv l \pmod{p}$$

for all primes $l \leq B$

The Index Calculus Method

- Having done this, we next look at the quantities

$$h \cdot g^{-k} \pmod{p} \qquad \text{for } k = 1, 2, \ldots$$

until we find a value of k such that $h \cdot g^{-k} \pmod{p}$ is B-smooth
- For this value of k we have

$$h \cdot g^{-k} \equiv \prod_{\ell \leq B} \ell^{e_\ell} \pmod{p}$$

for certain exponents e_l

The Index Calculus Method

- We rewrite in terms of discrete logarithms as

$$\log_g(h) \equiv k + \sum_{\ell \leq B} e_\ell \cdot \log_g(\ell) \pmod{p-1}$$

- But we are assuming that we already computed $\log_g(l)$ for all primes $l \leq B$
- Hence the formula gives the value of $\log_g(h)$
- It remains to explain how to find $\log_g(l)$ for small primes l

The Index Calculus Method

- For a random selection of exponents i we compute $g_i \equiv g^i \pmod{p}$ with $0 < g_i < p$
- If g_i is not B-smooth, then we discard it, while if g_i is B-smooth, then we can factor it as

$$g_i = \prod_{\ell \leq B} \ell^{u_\ell(i)}$$

The Index Calculus Method

- In terms of discrete logarithms, this gives the relation

$$i \equiv \log_g(g_i) \equiv \sum_{\ell \leq B} u_\ell(i) \cdot \log_g(\ell) \pmod{p-1}$$

- Notice that the only unknown quantities in the formula are the discrete logarithm values $\log_g(l)$
- So if we can find more than $\pi(B)$ equations, then we can use linear algebra to solve for the $\log_g(l)$ "variables"

The Index Calculus Method

- Remark. A minor issue that we have ignored is the fact that the linear equations are congruences modulo $p-1$
- Standard linear algebra methods such as Gaussian elimination do not work well modulo composite numbers, because there are numbers that do not have multiplicative inverses
- The Chinese remainder theorem solves this problem

The Index Calculus Method

- First we solve the congruences modulo q for each prime q dividing $p-1$
- Then, if q appears in the factorization of $p-1$ to a power q^e, we lift the solution from $\mathbb{Z}/q\mathbb{Z}$ to $\mathbb{Z}/q^e\mathbb{Z}$
- Finally, we use the Chinese remainder theorem to combine solutions modulo prime powers to obtain a solution modulo $p-1$

The Index Calculus Method

- In cryptographic applications one should choose p such that $p-1$ is divisible by a large prime; otherwise, the Pohlig–Hellman algorithm solves the discrete logarithm problem
- For example, if we select $p = 2q+1$ with q prime, then the index calculus requires us to solve simultaneous congruences modulo q and modulo 2

The Index Calculus Method

- Example. We let p be the prime $p = 18443$ and use the index calculus to solve the discrete logarithm problem

$$37^x \equiv 221 \ (mod\ 18443)$$

- We note that $g = 37$ is a primitive root modulo $p = 18443$
- Take $B = 5$, so our factor base is the set of primes $\{2,3,5\}$

The Index Calculus Method

- We start by taking random powers of $g = 37$ modulo 18443 and pick out the ones that are B-smooth
- A couple of hundred attempts gives four equations:

$$g^{12708} \equiv 2^3 \cdot 3^4 \cdot 5 \pmod{18443},$$
$$g^{15400} \equiv 2^3 \cdot 3^3 \cdot 5 \pmod{18443},$$
$$g^{11311} \equiv 2^3 \cdot 5^2 \pmod{18443},$$
$$g^{2731} \equiv 2^3 \cdot 3 \cdot 5^4 \pmod{18443}.$$

The Index Calculus Method

- These in turn give linear relations for the discrete logarithms of 2, 3, and 5 to the base g. For example,

$$12708 = 3 \cdot \log_g(2) + 4 \cdot \log_g(3) + \log_g(5)$$

- To ease notation, we let

$$x_2 = \log_g(2), \quad x_3 = \log_g(3), \quad \text{and} \quad x_5 = \log_g(5)$$

The Index Calculus Method

- Then the four congruences become the following four linear relations:

$$\begin{aligned} 12708 &= 3x_2 + 4x_3 + x_5 \pmod{18442},\\ 11311 &= 3x_2 + 2x_5 \pmod{18442},\\ 15400 &= 3x_2 + 3x_3 + x_5 \pmod{18442},\\ 2731 &= 3x_2 + x_3 + 4x_5 \pmod{18442}. \end{aligned}$$

- Note that the formulas are congruences modulo

$$p - 1 = 18442 = 2 \cdot 9221$$

The Index Calculus Method

- The number 9221 is prime, so we need to solve the system of linear equations modulo 2 and modulo 9221
- This is easily accomplished by Gaussian elimination. The solutions are

$$(x_2, x_3, x_5) \equiv (1, 0, 1) \pmod{2},$$

$$(x_2, x_3, x_5) \equiv (5733, 6529, 6277) \pmod{9221}$$

The Index Calculus Method

- Combining these solutions yields

$$(x_2, x_3, x_5) \equiv (5733, 15750, 6277) \pmod{18442}$$

- We check the solutions by computing

$$\begin{aligned} 37^{5733} &\equiv 2 \pmod{18443},\\ 37^{15750} &\equiv 3 \pmod{18443},\\ 37^{6277} &\equiv 5 \pmod{18443} \end{aligned}$$

The Index Calculus Method

- Recall that our ultimate goal is to solve

$$37^x \equiv 221 \ (mod\ 18443)$$

- We compute the value of

$$211 \cdot 37^{-k} \ (mod\ 18443)$$

for random values of k until we find a value that is B-smooth

- After a few attempts we find that

$$211 \cdot 37^{-9549} \equiv 2^5 \cdot 3^2 \cdot 5^2 \pmod{18443}$$

The Index Calculus Method

- Using the values of the discrete logs of 2, 3, and 5 from above, this yields

$$\log_g(211) = 9549 + 5\log_g(2) + 2\log_g(3) + 2\log_g(5)$$
$$= 9549 + 5 \cdot 5733 + 2 \cdot 15750 + 2 \cdot 6277 \equiv 8500 \pmod{18442}$$

- Finally, we check our answer $\log_g(211) = 8500$ by computing

$$37^{8500} \equiv 211 \pmod{18443}$$

The Index Calculus Method

- Remark. The index calculus is a subexponential algorithm for solving the discrete logarithm problem in $\mathbb{F}_p^*$
- Using ideas based on the number field sieve, the running time can be further reduced to a small power $L_{1/3}(p)$

The Index Calculus Method

- This stands in marked contrast to the discrete logarithm problem in elliptic curve groups
- Currently, the best known algorithms to solve the general discrete logarithm problem in elliptic curve groups are fully exponential