Number Theory

Intro. Number Theory

Modular e'th roots

Modular e'th roots

We know how to solve modular **linear** equations:

$\mathbf{a \cdot x + b = 0}$ in Z_N Solution: $\mathbf{x = -b \cdot a^{-1}}$ in Z_N

What about higher degree polynomials?

Example: let p be a prime and $c \in Z_p$. Can we solve:

$$x^2 - c = 0 \quad , \quad y^3 - c = 0 \quad , \quad z^{37} - c = 0 \quad \text{in} \quad Z_p$$

Modular e'th roots

Let p be a prime and $c \in Z_p$.

<u>Def</u>: $x \in Z_p$ s.t. $x^e = c$ in Z_p is called an **<u>e'th root</u>** of c .

$6^3 = 216 = 7$ in $\mathbb{Z}_{11}$

Examples: $7^{1/3} = 6$ in $\mathbb{Z}_{11}$

$3^{1/2} = 5$ in $\mathbb{Z}_{11}$

$1^{1/3} = 1$ in $\mathbb{Z}_{11}$

$2^{1/2}$ does not exist in $\mathbb{Z}_{11}$

The easy case

When does $c^{1/e}$ in Z_p exist? Can we compute it efficiently?

<u>The easy case</u>: suppose $\gcd(e, p-1) = 1$

Then for all c in $(Z_p)^*$: $c^{1/e}$ exists in Z_p and is easy to find.

Proof: let $d = e^{-1}$ in Z_{p-1}. Then $\boxed{c^{1/e} = c^d \text{ in } \mathbb{Z}_p}$

$d \cdot e = 1$ in Z_{p-1} $\Rightarrow$ $\exists k \in \mathbb{Z}: d \cdot e = k \cdot (p-1) + 1 \Longrightarrow$

$\Longrightarrow (c^d)^e = c^{d \cdot e} = c^{k \cdot (p-1) + 1} = [c^{p-1}]^k \cdot c = c$ in $\mathbb{Z}_p$

The case e=2: square roots

If p is an odd prime then gcd(2, p-1) ≠ 1

$$x \quad -x \longrightarrow x^2$$

Fact: in $\mathbb{Z}_p^*$, $x \longrightarrow x^2$ is a 2-to-1 function

Example: in $\mathbb{Z}_{11}^*$:

x, $-x$	1, 10	2, 9	3, 8	4, 7	5, 6
x^2	1	4	9	5	3

Def: x in $\mathbb{Z}_p$ is a **quadratic residue** (Q.R.) if it has a square root in $\mathbb{Z}_p$

p odd prime ⇒ the # of Q.R. in $\mathbb{Z}_p$ is (p-1)/2 + 1

Euler's theorem

Thm: x in $(Z_p)^*$ is a Q.R. $\Longleftrightarrow$ $x^{(p-1)/2} = 1$ in Z_p (p odd prime)

Example:

in $\mathbb{Z}_{11}$:	1^5,	2^5,	3^5,	4^5,	5^5,	6^5,	7^5,	8^5,	9^5,	10^5
=	1	-1	1	1	1,	-1,	-1,	-1,	1,	-1

Note: $x \neq 0 \Rightarrow x^{(p-1)/2} = \left(x^{p-1}\right)^{1/2} = 1^{1/2} \in \{1, -1\}$ in Z_p

Def: $x^{(p-1)/2}$ is called the **Legendre Symbol** of x over p (1798)

Computing square roots mod p

Suppose $p = 3 \pmod 4$

Lemma: if $c \in (Z_p)^*$ is Q.R. then $\sqrt{c} = c^{(p+1)/4}$ in Z_p

Proof: $\left[c^{\frac{p+1}{4}}\right]^2 = c^{\frac{p+1}{2}} = \underbrace{c^{\frac{p-1}{2}}}_{=1} \cdot c = c$ in $\mathbb{Z}_p$

When $p = 1 \pmod 4$, can also be done efficiently, but a bit harder

run time $\approx O(\log^3 p)$

Solving quadratic equations mod p

Solve: $a \cdot x^2 + b \cdot x + c = 0$ in Z_p

Solution: $x = \left(-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}\right) / 2a$ in Z_p

- Find $(2a)^{-1}$ in Z_p using extended Euclid.

- Find square root of $b^2 - 4 \cdot a \cdot c$ in Z_p (if one exists)

  using a square root algorithm

Computing e'th roots mod N ??

Let N be a composite number and e>1

When does $c^{1/e}$ **in** Z_N exist? Can we compute it efficiently?

Answering these questions requires the factorization of N

(as far as we know)

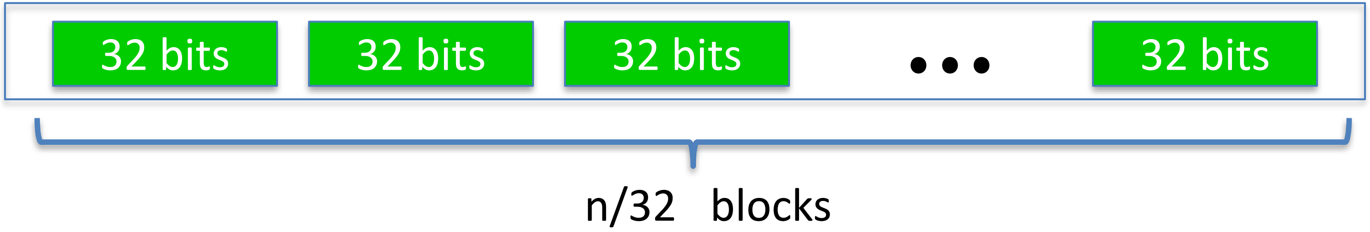
End of Segment

Intro. Number Theory

Arithmetic algorithms

Representing bignums

Representing an n-bit integer (e.g. n=2048) on a 64-bit machine



Note: some processors have 128-bit registers (or more)
and support multiplication on them

Arithmetic

Given: two n-bit integers

- **Addition and subtraction**: linear time $O(n)$
- **Multiplication**: naively $O(n^2)$. Karatsuba (1960): $O(n^{1.585})$ $\log_2 3$

  Best (asymptotic) algorithm: about $O(n \cdot \log n)$.
- **Division with remainder**: $O(n^2)$.

Exponentiation

Finite cyclic group G (for example G = $\mathbb{Z}_p^*$)

Goal: given g in G and x compute g^x

Example: suppose $x = 53 = (110101)_2 = 32+16+4+1$

Then: $g^{53} = g^{32+16+4+1} = g^{32} \cdot g^{16} \cdot g^{4} \cdot g^{1}$

$$g \longrightarrow g^2 \longrightarrow g^4 \longrightarrow g^8 \longrightarrow g^{16} \longrightarrow g^{32} \qquad g^{53}$$

The repeated squaring alg.

Input: g in G and x>0 ; **Output**: g^x

write $x = (x_n\, x_{n-1} \ldots x_2\, x_1\, x_0)_2$

$y \longleftarrow g \quad , \quad z \longleftarrow 1$

for i = 0 to n do:

  if (x[i] == 1): $z \longleftarrow z \cdot y$

  $y \longleftarrow y^2$

output z

example: g^{53}

y	z
g^2	g
g^4	g
g^8	g^5
g^{16}	g^5
g^{32}	g^{21}
g^{64}	$\mathbf{g^{53}}$

Running times

Given n-bit int. N:

- **Addition and subtraction in $\mathbf{Z_N}$**: linear time $T_+ = O(n)$
- **Modular multiplication in $\mathbf{Z_N}$**: naively $T_\times = O(n^2)$
- **Modular exponentiation in $\mathbf{Z_N}$ ($\mathbf{g^x}$)**:

$$O\big((\log x)\cdot T_\times\big) \leq O\big((\log x)\cdot n^2\big) \leq O(n^3)$$

End of Segment

Intro. Number Theory

Intractable problems

Easy problems

- Given composite N and x in Z_N find x^{-1} in Z_N

- Given prime p and polynomial $f(x)$ in $Z_p[x]$

  find x in Z_p s.t. $f(x) = 0$ in Z_p (if one exists)

  Running time is linear in deg(f) .

… but many problems are difficult

Intractable problems with primes

Fix a prime $p>2$ and g in $(Z_p)^*$ of order q.

Consider the function: $x \longmapsto g^x$ in Z_p

Now, consider the inverse function:

$\mathrm{Dlog}_g\,(g^x) = x$ **where x in $\{0, \ldots, q-2\}$**

Example:

in $\mathbb{Z}_{11}$:	1,	2,	3,	4,	5,	6,	7,	8,	9,	10
$\mathrm{Dlog}_2(\cdot)$:	0,	1,	8,	2,	4,	9,	7,	3,	6,	5

DLOG: more generally

Let **G** be a finite cyclic group and **g** a generator of G

$$G = \{ 1, g, g^2, g^3, \ldots, g^{q-1} \}$$ (q is called the order of G)

Def: We say that **DLOG is hard in G** if for all efficient alg. A:

$$\Pr_{g \leftarrow G,\ x \leftarrow \mathbb{Z}_q} \left[A(G, q, g, g^x) = x \right] < \text{negligible}$$

Example candidates:

(1) $(\mathbb{Z}_p)^*$ for large p, (2) Elliptic curve groups mod p

Computing Dlog in $(Z_p)^*$ (n-bit prime p)

Best known algorithm (GNFS): run time exp($\tilde{O}(\sqrt[3]{n})$)

cipher key size	modulus size	Elliptic Curve group size
80 bits	1024 bits	160 bits
128 bits	3072 bits	256 bits
256 bits (AES)	**15360** bits	512 bits

As a result: slow transition away from (mod p) to elliptic curves

An application: collision resistance

Choose a group G where Dlog is hard (e.g. $(Z_p)^*$ for large p)

Let $q = |G|$ be a prime. Choose generators g, h of G

For $x,y \in \{1,\ldots,q\}$ define $\boxed{H(x,y) = g^x \cdot h^y}$ in G

Lemma: finding collision for H(.,.) is as hard as computing $Dlog_g(h)$

Proof: Suppose we are given a collision $H(x_0,y_0) = H(x_1,y_1)$

then $g^{x_0} \cdot h^{y_0} = g^{x_1} \cdot h^{y_1} \Rightarrow g^{x_0-x_1} = h^{y_1-y_0} \Rightarrow h = g^{x_0-x_1/y_1-y_0}$

$\neq 0$

Intractable problems with composites

Consider the set of integers: (e.g. for n=1024)

$$\mathbb{Z}_{(2)}(n) \ := \ \{ N = p\cdot q \ \text{ where } p,q \text{ are n-bit primes} \}$$

Problem 1: Factor a random N in $\mathbb{Z}_{(2)}(n)$ (e.g. for n=1024)

Problem 2: Given a polynomial **f(x)** where degree(f) > 1

and a random N in $\mathbb{Z}_{(2)}(n)$

find x in $\mathbb{Z}_N$ s.t. f(x) = 0 in $\mathbb{Z}_N$

The factoring problem

Gauss (1805): *"The problem of distinguishing prime numbers from composite numbers and of resolving the latter into their prime factors is known to be one of the most important and useful in arithmetic."*

Best known alg. (NFS): run time exp($\tilde{O}(\sqrt[3]{n})$) for n-bit integer

Current world record: **RSA-768** (232 digits)

- Work: two years on hundreds of machines
- Factoring a 1024-bit integer: about 1000 times harder

$\Rightarrow$ likely possible this decade

Further reading

- A Computational Introduction to Number Theory and Algebra, V. Shoup, 2008 (V2), Chapter 1-4, 11, 12

Available at **//shoup.net/ntb/ntb-v2.pdf**

End of Segment