
# Intro. Number Theory

## Notation

# Background

We will use a bit of number theory to construct:

- Key exchange protocols
- Digital signatures
- Public-key encryption

This module: crash course on relevant concepts

More info: read parts of Shoup's book referenced at end of module

# Notation

From here on:

- N denotes a positive integer.
- p denote a prime.

Notation: $\mathbb{Z}_N = \{0, 1, 2, \ldots, N-1\}$

Can do addition and multiplication modulo N

# Modular arithmetic

Examples: let $N = 12$

$9 + 8 = 5$ in $\mathbb{Z}_{12}$ $\{0, 1, \ldots\ldots 11\}$

$5 \times 7 = 11$ in $\mathbb{Z}_{12}$ $35 = 11 \bmod 12$

$5 - 7 = 10$ in $\mathbb{Z}_{12}$

Arithmetic in $\mathbb{Z}_N$ works as you expect, e.g $x \cdot (y+z) = x \cdot y + x \cdot z$ in $\mathbb{Z}_N$

# Greatest common divisor

**Def**: For ints. x,y: **gcd(x, y)** is the greatest common divisor of x,y

Example: gcd( 12, 18 ) = 6

$\underbrace{\boxed{2} \times 12 \; \boxed{-1} \times 18 = 6}$

a b

**Fact**: for all ints. x,y there exist ints. a,b such that

$$\mathbf{a \cdot x + b \cdot y = gcd(x,y)}$$

a,b can be found efficiently using the extended Euclid alg.

互素

If gcd(x,y)=1 we say that x and y are **relatively prime**

# Modular inversion

Over the rationals, inverse of 2 is ½ .   What about $\mathbb{Z}_N$ ?

**Def**: The **inverse** of x in $\mathbb{Z}_N$ is an element y in $\mathbb{Z}_N$ s.t. $x \cdot y = 1$ in $\mathbb{Z}_N$

y is denoted $x^{-1}$ .

Example: let N be an odd integer. The inverse of 2 in $\mathbb{Z}_N$ is $\frac{N+1}{2}$

$$2 \cdot \left(\frac{N+1}{2}\right) = N+1 = 1 \text{ in } \mathbb{Z}_N$$

# Modular inversion

Which elements have an inverse in $\mathbb{Z}_N$ ?

$x \cdot y = 1 \bmod N$ ?

**Lemma**: x in $\mathbb{Z}_N$ has an inverse $\Leftarrow$ if and only if $\Rightarrow$ gcd(x,N) = 1

Proof:

$\gcd(x,N)=1 \Rightarrow \exists\, a,b:\ a\cdot x + \cancel{b\cdot N} = 1 \Longrightarrow a\cdot x = 1 \text{ in } \mathbb{Z}_N$

$\Longrightarrow x^{-1} = a \text{ in } \mathbb{Z}_N$

$\gcd(x,N) > 1 \Rightarrow \forall a:\ \gcd(a\cdot x, N) > 1 \Rightarrow a\cdot x \neq 1 \text{ in } \mathbb{Z}_N$

$\gcd(x,N)=2 \Longrightarrow \forall a:\ a\cdot x \text{ is even} \Longrightarrow \overbrace{a\cdot x}^{\text{even}} \neq \overbrace{b\cdot N + 1}^{\text{odd}}$

# More notation

**Def:** $\mathbb{Z}_N^*$ = (set of invertible elements in $\mathbb{Z}_N$) =

= $\{ \ x \in \mathbb{Z}_N \ : \ \gcd(x,N) = 1 \ \}$

Examples:

1. for prime p, $\mathbb{Z}_p^* = \underline{\mathbb{Z}_p \setminus \{0\}} = \{1, 2, \ldots, p-1\}$
2. $\mathbb{Z}_{12}^*$ = { 1, 5, 7, 11}

2, 3, 4, 6, 8, 9, 10

$x, N \Rightarrow ax + bN = 1$

mod N

For x in $\mathbb{Z}_N^*$, can find $x^{-1}$ using extended Euclid algorithm.

# Solving modular linear equations

Solve: $a \cdot x + b = 0$ in $\mathbb{Z}_N$

Solution: $x = -b \cdot a^{-1}$ in $\mathbb{Z}_N$

Find $a^{-1}$ in $\mathbb{Z}_N$ using extended Euclid. Run time: $O(\log^2 N)$

What about modular quadratic equations?

next segments


# Intro. Number Theory

## Fermat and Euler

# Review

N denotes an n-bit positive integer. p denotes a prime.

- $Z_N$ = { 0, 1, …, N-1 }
- $(Z_N)^*$ = (set of invertible elements in $Z_N$) = $\{ \ x \in Z_N : \gcd(x,N) = 1 \ \}$

$\log^2 N$

$n = O(\log N)$

Can find inverses efficiently using Euclid alg.: time = $O(n^2)$

# Fermat's theorem (1640)

**Thm:** Let p be a prime

$$\forall\, x \in (Z_p)^* : \quad x^{p-1} = 1 \text{ in } Z_p \quad \Rightarrow x \cdot x^{p-2}$$

Example: p=5. $3^4 = 81 = 1$ in $Z_5$

$Z_5^* = \{1, 2, 3, 4\}$

$Z_p^*$

So: $x \in (Z_p)^* \quad \Rightarrow \quad x \cdot x^{p-2} = 1 \quad \Rightarrow \quad x^{-1} = x^{p-2}$ in $Z_p$

another way to compute inverses, but less efficient than Euclid

# Application: generating random primes

Suppose we want to generate a large random prime

say, prime p of length 1024 bits ( i.e. $p \approx 2^{1024}$ )

$\mathbb{Z}_p^* \Rightarrow \forall x.\quad x^{p-1} = 1 \mod p$

Step 1: choose a random integer $p \in [\ 2^{1024}\ ,\ 2^{1025}-1\ ]$

Step 2: test if $2^{p-1} = 1$ in $\mathbb{Z}_p$ $\qquad 2^{p-1} = 1 \mod p.$

If so, output p and stop. If not, goto step 1 .

Simple algorithm (not the best). **Pr[ p not prime ] < $2^{-60}$**

# The structure of $(Z_p)^*$

循环群

**Thm** (Euler): $(Z_p)^*$ is a **cyclic group**, that is

$$\exists\, g \in (Z_p)^* \quad \text{such that} \quad \{1, g, g^2, g^3, \ldots, g^{p-2}\} = (Z_p)^*$$

$\{1, g, g^2, \ldots g^{p-2}\}$ $Z_p^*$

g is called a **generator** of $(Z_p)^*$

Example: p=7. $\{1, 3, 3^2, 3^3, 3^4, 3^5\} = \{1, 3, 2, 6, 4, 5\} = (Z_7)^*$

Not every elem. is a generator: $\{1, 2, 2^2, 2^3, 2^4, 2^5\} = \{1, 2, 4\}$

# Order

$2 \quad 2^2 \quad 2^3$

For $g \in (Z_p)^*$ the set $\{1, g, g^2, g^3, \ldots\}$ is called $\{1, 2, 4\}$

the **group generated by g**, denoted <g>

生成群

**Def**: the **order** of $g \in (Z_p)^*$ is the size of <g>

$$\mathbf{ord_p(g) \quad = \quad |\langle g \rangle| \quad = \quad (smallest\ a>0\ s.t.\ \ g^a = 1\ in\ Z_p)}$$

Examples: $ord_7(3) = 6$ ; $ord_7(2) = 3$ ; $ord_7(1) = 1$

$\{1, 2, 4\}$

**Thm** (Lagrange): $\forall g \in (Z_p)^*$ : $\mathbf{ord_p(g)}$ divides $p-1$

# Euler's generalization of Fermat (1736)

**Def**: For an integer N define $\varphi(N) = |(Z_N)^*|$ (Euler's φ func.)

Examples: $\varphi(12) = |\{1,5,7,11\}| = 4$ ; $\varphi(p) = p-1$

For $N=p \cdot q$: $\varphi(N) = N-p-q+1 = (p-1)(q-1)$

**Thm** (Euler): $\forall\, x \in (Z_N)^*$ : $x^{\varphi(N)} = 1$ in $Z_N$   $Z_p^*$   $x^{p-1} = 1$ in $Z_p^*$

N is prime.

Example: $5^{\varphi(12)} = 5^4 = 625 = 1$ in $Z_{12}$

Generalization of Fermat. Basis of the RSA cryptosystem


# 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 .

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

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

$\mathbb{Z}_{11}^*$

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

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

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

$\nexists\, x \in \mathbb{Z}_{11}$ s.t. $x^2 = 2$ mod 11

# The easy case

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

$\varphi(p) = p-1$

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

$c^{p-1} = 1 \mod p$

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

gcd(e, p-1)

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

$\Rightarrow \left(\boxed{c^d}\right)^e = c^{d \cdot e} = c^{k \cdot (p-1)+1} = \left[c^{p-1}\right]^k \cdot c = \boxed{c}$ 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 $\mathbb{Z}_{11}^*$ :

| 1 10 | 2 9 | 3 8 | 4 7 | 5 6 |
|---|---|---|---|---|
| 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 $\Rightarrow$ 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. $\iff$ $x^{(p-1)/2} = 1$ in $Z_p$ (p odd prime)

$x^{\frac{(p-1)}{2}}$

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

$Z_p^*$

勒让德符号

**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 ≈ $\underline{\underline{O(\log^3 p)}}$ $\log^2 N$

# Solving quadratic equations mod p

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

Solution: $x = (-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}\,) \;/\; 2a$ 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** $\mathbb{Z}_N$ exist? Can we compute it efficiently?

$\varphi(N)$

$c^{\frac{1}{e}} = c^{d}$. gcd.$(e, \varphi(N))$

$d = e^{-1} \bmod (\varphi(N))$.

Answering these questions requires the factorization of N

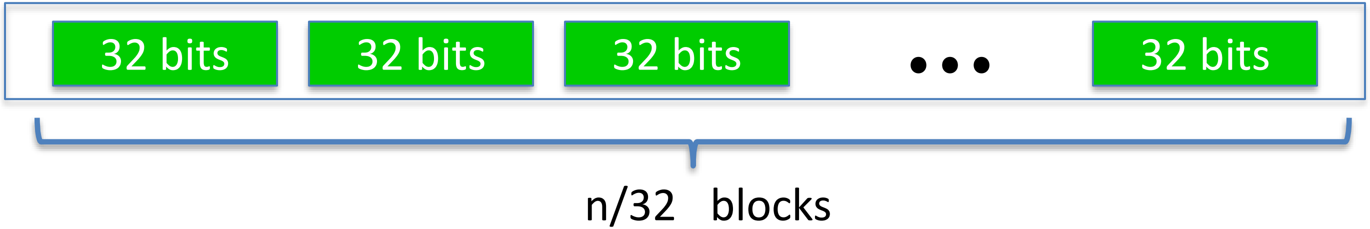
(as far as we know)


# 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): $O(n^{1.585})$ $\log_2 3$

  Basic idea: $(2^b x_2 + x_1) \times (2^b y_2 + y_1)$ with 3 mults.

  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 $g^x$ $\quad g \cdot g \cdot \cdots g$

**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 ⟵ g ,   z ⟵ 1
for i = 0 to n do:
      if (x[i] == 1):    z ⟵ z·y
      y ⟵ y²
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 $Z_N$:** linear time $T_+ = O(n)$
- **Modular multiplication in $Z_N$:** naively $T_\times = O(n^2)$
- **Modular exponentiation in $Z_N$ ( $g^x$ ):**

$$n \cdot n^2$$

$$O\Big( (\log x)\cdot T_\times \Big) \;\le\; O\Big( (\log x)\cdot n^2 \Big) \;\le\; O( n^3 )$$


# 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: $x \mapsto g^x$ in $Z_p$

Now, consider the inverse function:

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

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:

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

Example candidates:

(1) $(Z_p)^*$ for large p, (2) Elliptic curve groups mod p

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

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

$e^{\sqrt[3]{n}}$

| cipher key size | modulus size ($Z_p^*$) | 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 $x,y \in \{1,\dots,q\}$ define $\boxed{H(x,y) = g^x \cdot h^y}$ in G

↳ 解决 Dlog.

**Lemma:** finding collision for H(.,.) is as hard as computing $\boxed{\mathrm{Dlog}_g(h)}$

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

≠0

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

# 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-829** (250 digits)

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

2048

$\Rightarrow$ likely possible this decade

https://listserv.nodak.edu/cgi-bin/wa.exe?A2=NMBRTHRY;dc42ccd1.2002

# 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