Introduction to Number Theory

This slide is made based the online course of Cryptography by Dan Boneh

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

一些符号定义

From here on:

- N denotes a positive integer. 正整数
- p denote a prime. 正质数

Notation: $Z_N = \{0,1,2,\ldots,N-1\}$ 表示 0 到 N-1

Can do addition and multiplication modulo N

加 乘 取模N 下

Modular arithmetic

Examples: let N = 12

$9 + 8 = 5$ in $\mathbb{Z}_{12}$ (17 mod 12 = 5)

$5 \times 7 = 11$ in $\mathbb{Z}_{12}$ (35 mod 12 = 11)

$5 - 7 = 10$ in $\mathbb{Z}_{12}$ (−2 mod 12 = 10)

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

最大公约数

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

Example: gcd(12, 18) = 6 $\quad 2\times 12 - 1\times 18 = 6$

存在这样的两个数

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

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

拓展的欧几里得算法

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

a和b不仅存在，还可用一个算法找到

不用3部

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

互质的

最大公约数是1

Modular inversion

模的逆

Over the rationals, inverse of 2 is ½ .

以前学过 2的逆是 1/2

What about $\mathbb{Z}_N$?

在$\mathbb{Z}_N$上呢?

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 $\frac{N+1}{2}$

奇

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

这个的模在$\mathbb{Z}_N$中=除以的余数

Modular inversion

Which elements have an inverse in $\mathbb{Z}_N$?
在Zn中哪些元素有逆?

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

Proof:

gcd(x,N)=1 ⇒ ∃ a,b: a·x + b·N = 1 ⇒ $a \cdot x = 1$ *in* $\mathbb{Z}_N$

⇒ $x^{-1} = a$ *in* $\mathbb{Z}_N$

x in $\mathbb{Z}_N$ has an inverse ⇒ gives us a number y s.t. $xy \equiv 1 \bmod n$.

It means that $xy = kn+1$, or $xy - kn = 1$.

For any common divisor, c, of x and n we have $c|(xy-kn)$ which gives $c|1$, that is, $c=1$.

More notation

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

= { x∈ $\mathbb{Z}_N$: gcd(x,N) = 1 }

Examples:

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

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

线性同余方程

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

End of Segment

Intro. Number Theory

Fermat and Euler

Review

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

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

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

Fermat's theorem (1640)

费马

Thm: Let p be a prime

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

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

$O(\log^3 p)$

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

another way to compute inverses, but less efficient than Euclid

Application: generating random primes

产生随机数算法

Suppose we want to generate a large random prime

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

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

Step 2: test if $2^{p-1} = 1$ in Z_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 $(Z_p)^*$

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

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

因为 $g^{p-1} = 1$

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

生成元

Example: p=7 , please give a generator

$(Z_7)^* = \{1, 2, 3, 4, 5, 6\}$ 到 $N-1$

let $g = 3 \Rightarrow \{3^0\ 3^1\ 3^2\ 3^3\ 3^4\ 3^5\}\ 3^6 = 1$ 到 g^{N-2}

in $(Z_7)^*$ = 1 3 2 6 4 5

The structure of $(Z_p)^*$

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

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

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

1 2 3 4 5 6 都有3. ⇒ 3是生成元.

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

⇒ 2不是生成元

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

Order

这个元组的大小，有多少元素

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

the **group generated by g**, denoted $\langle g \rangle$

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

所 这个元组有多少元素

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

$2^3 = 8$ in Z_7 = 1 即 $a=3$

Examples: $\text{ord}_7(3) = 6$; $\text{ord}_7(2) = 3$; $\text{ord}_7(1) = 1$

$\{1, 2, 4\}$ $\{1,\}$

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

这个阶始终整除 $p-1$

Euler's generalization of Fermat (1736)

欧拉定理

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

若N刚好是两个质数p，q的乘积

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

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

Generalization of Fermat. Basis of the RSA cryptosystem

End of Segment

Intro. Number Theory

Modular e’th roots

Modular e'th roots

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

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

What about higher degree polynomials?

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

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

Modular e'th roots

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

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

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

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

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

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

$5^2 = 25 = 3$ in $\mathbb{Z}_{11}$

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

$\Rightarrow 1^3 = 1$ in Z_{11}

The easy case

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

The easy case: 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 $c^{1/e} = c^d$ in Z_p

$d \cdot e = 1 \bmod p-1$ 商在 k

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

$c^{k(p-1)+1} = (c^{p-1})^k \cdot c = c$ in Z_p

2° The case e=2: square roots

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

奇质数 偶

不互质时(只有一半元素有平方根)

x −x

x^2

求平方根

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

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

1	10^{-1}	2	9^{-2}	3	8^{-3}	4	7	5	6
1		4		9		5		3	

其他元素无平方根

有平方根

4的平方根是 2、9

9的平方根3、8

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

x有平方根：二次剩余，无平方根：非二次剩余

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

奇质数的二次剩余个数?

也是二次剩余

⇒ Euler's theorem 判断x是否有平方根呢？

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

x是一个二次剩余，

p=11 (p-1)/2 = 5

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

是二次剩余 非二次剩余

取非0的数x

由费马定理，一定是1 or -1

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

勒让德符号 值1就是-1

Computing square roots mod p

计算质数模的平方根

Suppose $p = 3 \pmod 4$

$p+1 = 0 \bmod 4$

$\frac{p+1}{4}$ is int

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

Proof: $\left[c^{(p+1)/4}\right]^2 = c^{(p+1)/2} = c^{(p-1)/2} \cdot c = c$ in Z_p

c是二次剩余，必成得1

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

当p=1时不好算，用随机算法

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: $x = (-b \pm \sqrt{b^2 - 4 \cdot a \cdot c}) / 2a$ 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** 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

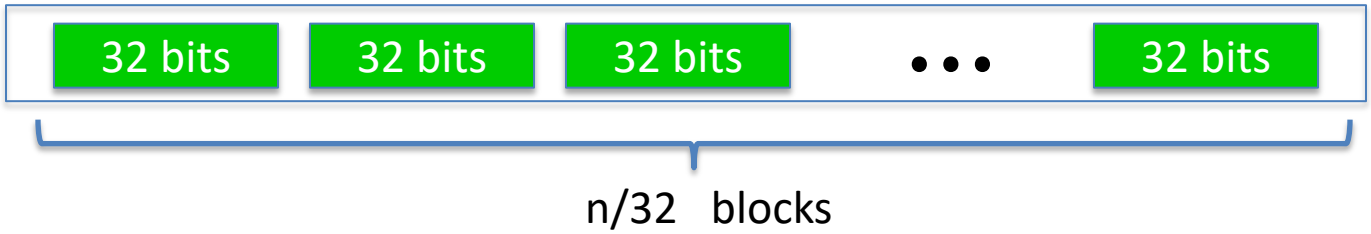
计算大整数的模

Representing bignums

计算机中放大整数

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

32个为一组 共 n/32个组



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): $O(n^{1.585})$

  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 求幂问题

有限循环生成群 G，这个群由 g 的幂生成

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

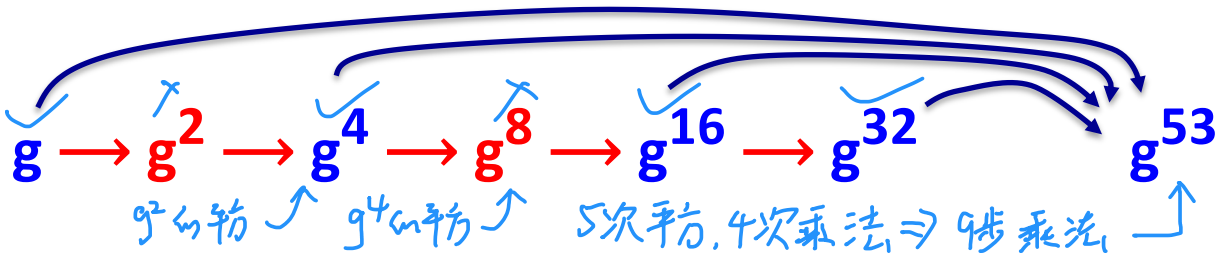
Goal: given g in G and x compute g^x 普通办法：$g \cdot g \cdot g = g^3$

$x = 500$？ $gg \cdots g$（500） 垃圾！

重复平方法，get！

Example: suppose $x = 53 = (110101)_2 = 32+16+4+1$ （32 16 8 4 2 1）

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



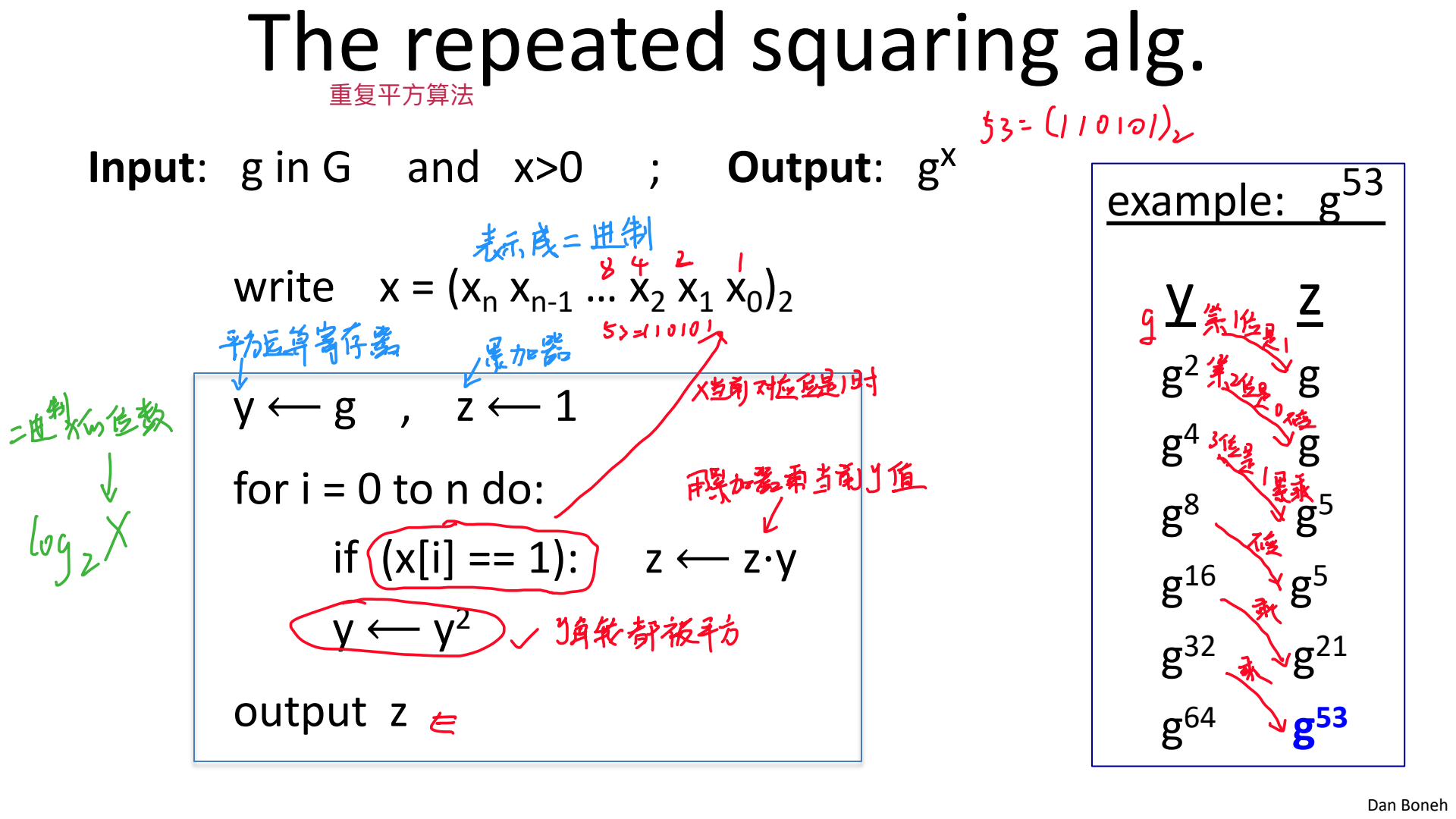
The repeated squaring alg.

重复平方算法

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 ← y^2
output z
```



example: g^{53}

y	z
g	第1位是1
g^2	g
g^4	第2位是0跳
g^8	g
g^{16}	第3位是1累乘
g^{32}	g^5
g^{64}	跳
	g^{21}
	乘
	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)**:

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

End of Segment

Intro. Number Theory

Intractable problems

依数模，多数模难题

Easy problems

- Given composite N and x in Z_N find x^{-1} in Z_N 在Z_N中找x的逆元

- Given prime p and polynomial f(x) in $Z_p[x]$ 某个多项式f(x)

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

  找x是这个多项式的根容易

  Running time is linear in deg(f) .

  耗时与多项式次数是线性的

… but many problems are difficult 难 ⇒ 困难问题组成了许多公钥密码系统

Intractable problems with primes

质数模的棘手问题

固定某个大质数p（p可能600位质数） 固定$(\mathbb{Z}_p)^*$中某个元素g，这个元素g的阶正好是数q

Fix a prime p>2 and g in $(\mathbb{Z}_p)^*$ of order q.

一个指数函数（重复平方，容易）

Consider the function: $x \mapsto g^x$ in $\mathbb{Z}_p$

我它的对数为，困难

Now, consider the inverse function:

离散对数问题

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

Example:

$2^0=1$ $2^1=2$ $2^2=4$ $2^4=16=5$ in $\mathbb{Z}_{11}$ $2^3=8$

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

1的离散对数为0

底数

DLOG: more generally

小质数好算，大质数困难

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

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

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

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

随机选元素g　随机选指数x　order 阶数　给群描述　给g　给g^x　计算出离散对数概率可忽略

这两个问题离散对数问题被认为是困难的

Example candidates:

椭圆曲线上可选更小的参数，攻击难度与$(Z_p)^*$上的更大质数相当

可选的：椭圆曲线群上的离散对数问题

(1) $(Z_p)^*$ for large p,

群上离散对数是困难的

(2) Elliptic curve groups mod p

离散椭圆曲线群(实用)

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

离散对数问题的直接应用　　抗碰撞哈希函数

G上的离散对数问题

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

阶为q的子群

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

群G有质数阶q(个数)

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

离散对数问题

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

碰撞也不用信息对

有个函数H的碰撞

Proof: Suppose we are given a collision $H(x_0,y_0) = H(x_1,y_1)$ $\neq 0$

模q的逆

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

若$y_0=y_1$ $x_0=x_1$ 非碰撞

计算出g的h的离散对数

Intractable problems with composites

和数的棘手问题

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

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

刚好两质数乘积、 位

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

一个随机整数，如何分解它？

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

非线性多项式，次数大于1

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

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

一个随机合数 找个多项式从根x ⇐ RSA

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