

8. modular arithmetic and PKC

tkkwon@snu.ac.kr

some slides from course15-251@cmu
course15-853@duke
Jeremy Johnson@Drexel

modular arithmetic

- we covered addition/multiplication and their inverses
- first, we dive into exponentiation in modular arithmetic

$$a^b \pmod{n}$$

a: base

b: exponent

n: modulus

let's forget modular arithmetic for a while

How do you compute...

5^8 using few multiplications?

First idea:

$$5 \quad 5^2 \quad 5^3 \quad 5^4 \quad 5^5 \quad 5^6 \quad 5^7 \quad 5^8$$
$$= 5 * 5 \quad 5^2 * 5$$

How do you compute...

5^8

Better idea:

$$5 \quad 5^2 \quad 5^4 \quad 5^8$$
$$= 5 * 5 \quad 5^2 * 5^2 * 5^2 * 5^2$$

Used only 3 mults
instead of 7 !!!

**Repeated squaring calculates
 a^{2^k}
in k multiply operations**

**compare with
 $(2^k - 1)$ multiply
operations
used by the naïve method**

How do you compute...

5^{13}

Use repeated squaring again?

5 5^2 5^4 5^8 ~~5^{16}~~

too high! what now?

assume no divisions allowed...

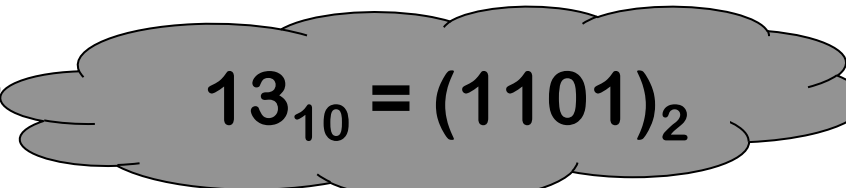
How do you compute...

$$5^{13}$$

Use repeated squaring again?

$$5 \quad 5^2 \quad 5^4 \quad 5^8$$

Note that $13 = 8 + 4 + 1$...


$$13_{10} = (1101)_2$$

$$\text{So } a^{13} = a^8 * a^4 * a^1$$

Two more multiplies!

To compute a^m

Suppose $2^k \leq m < 2^{k+1}$

$$a \quad a^2 \quad a^4 \quad a^8 \quad \dots \quad a^{2^k}$$

This takes k multiplies

Now write m as a sum of distinct powers of 2

$$\text{say, } m = 2^k + 2^{i_1} + 2^{i_2} \dots + 2^{i_t}$$

$$a^m = a^{2^k} * a^{2^{i_1}} * \dots * a^{2^{i_t}}$$

at most k more multiplies

**Hence, we can compute
 a^m
while performing at most
 $2 \lfloor \log_2 m \rfloor$ multiplies**

Now come back to modular arithmetic

How do you compute...

$$5^{13} \pmod{11}$$

First idea: Compute 5^{13} using 5 multiplies

$$\begin{aligned} 5 \quad 5^2 \quad 5^4 \quad 5^8 \quad 5^{12} \quad 5^{13} &= 1,220,703,125 \\ &= 5^8 * 5^5 * 5^{12} * 5 \end{aligned}$$

then take the answer mod 11

$$1,220,703,125 \pmod{11} = 4$$

How do you compute...

$$5^{13} \pmod{11}$$

Better idea: keep reducing the answer mod 11

5	5^2	5^4	5^8	5^{12}	5^{13}
		$9\%11=9$		$36\%11=3$	
$25\%11 = 3$			$81\%11=4$		$15\%11=4$

Hence, we can compute
 $a^m \pmod n$
while performing at most
 $2 \lfloor \log_2 m \rfloor$ multiplies

where each time we multiply
together numbers
with $\lfloor \log_2 n \rfloor + 1$ bits

How do you compute...

$$5^{121,242,653} \pmod{11}$$

The current best idea would still
need about 54 calculations

$$\text{answer} = 4$$

Can we exponentiate any faster?

OK, need a little more number theory for this one...

an example of modular exponentiation

- mod 7 case

Look at columns and rows!

$1^1 \equiv 1$	$1^2 \equiv 1$	$1^3 \equiv 1$	$1^4 \equiv 1$	$1^5 \equiv 1$	$1^6 \equiv 1$
$2^1 \equiv 2$	$2^2 \equiv 4$	$2^3 \equiv 1$	$2^4 \equiv 2$	$2^5 \equiv 4$	$2^6 \equiv 1$
$3^1 \equiv 3$	$3^2 \equiv 2$	$3^3 \equiv 6$	$3^4 \equiv 4$	$3^5 \equiv 5$	$3^6 \equiv 1$
$4^1 \equiv 4$	$4^2 \equiv 2$	$4^3 \equiv 1$	$4^4 \equiv 4$	$4^5 \equiv 2$	$4^6 \equiv 1$
$5^1 \equiv 5$	$5^2 \equiv 4$	$5^3 \equiv 6$	$5^4 \equiv 2$	$5^5 \equiv 3$	$5^6 \equiv 1$
$6^1 \equiv 6$	$6^2 \equiv 1$	$6^3 \equiv 6$	$6^4 \equiv 1$	$6^5 \equiv 6$	$6^6 \equiv 1$

Table 1.9: Powers of numbers modulo 7

Fermat's Little Theorem (FLT)

- $a^{p-1} = 1 \pmod{p}$
 - where p is prime and $\gcd(a, p) = 1$
- **also** $a^p = a \pmod{p}$
- useful in public key and primality testing

if modulus is denoted by p , that implies p is a prime number

Fermat's Little Theorem (FLT)

(Fermat's Little Theorem) If p is a prime and $p \nmid a$, then

$$a^{p-1} \equiv 1 \pmod{p}.$$

Proof. Let $S = \{1, 2, 3, \dots, p-1\}$. Consider the map $S \rightarrow S$:
 $m(x) \equiv ax \pmod{p}$. Clearly $m(x) \not\equiv 0 \pmod{p}$. Now, suppose
 $x \neq y \in S$. We have $ax \not\equiv ay \pmod{p}$. Therefore, $m(1), m(2), \dots,$
 $m(p-1)$ are distinct elements of S . It follows that

$$1 \cdot 2 \cdot 3 \cdots (p-1) \equiv m(1)m(2)\cdots m(p-1) \equiv (a \cdot 1) \cdot (a \cdot 2) \cdot (a \cdot 3) \cdots (a \cdot (p-1)) \equiv a^{p-1}(1 \cdot 2 \cdot 3 \cdots (p-1)) \pmod{p}.$$

Since $\gcd(j, p) = 1$ for $j \in S$, we can divide this congruence
by $1 \cdot 2 \cdot 3 \cdots (p-1)$. What remains is $1 \equiv a^{p-1} \pmod{p}$.

FLT: examples

- $2^{10} \pmod{11}$?
 - $2^{10} = 1024 = 1 \pmod{11}$
- $2^{53} \pmod{11}$?
 - $(2^{10})^5 2^3 = 2^3 = 8 \pmod{11}$

[Go back to page 13](#)

what if modulus is not a prime?

- Euler's Theorem
 - A general version of FLT
- start with Euler's totient/phi function $\phi(n)$
 - the number of the positive integers less than or equal to n that are relatively prime to n .

Euler's Totient Function $\phi(n)$

- when doing arithmetic modulo n
- **complete set of residues** is $Z_n = \{0, 1, \dots, n-1\}$
- **reduced set of residues** is those numbers (residues) which are relatively prime to n
 - e.g. for $n=10$,
 - complete set of residues is $Z_n = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
 - reduced set of residues is $Z_n^* = \{1, 3, 7, 9\}$
- number of elements in reduced set of residues is called the Euler's Totient Function $\phi(n)$

Euler's Totient Function: $\phi(n)$

- $\phi(4) = 2$ (1, 3 are relatively prime to 4)
- $\phi(5) = 4$ (1, 2, 3, 4 are relatively prime to 5)
- $\phi(6) = 2$ (1, 5 are relatively prime to 6)
- $\phi(7) = 6$ (1, 2, 3, 4, 5, 6 are relatively prime to 7)

Euler's Totient Function $\phi(n)$

- to compute $\phi(n)$, we need to count number of residues to be excluded
- in general we need prime factorization, but
 - for p (p : prime) $\phi(p) = p-1$
 - for $p \cdot q$ (p, q : prime) $\phi(pq) = (p-1)(q-1)$
 - for p^k (p : prime) $\phi(p^k) = p^{k-1}(p-1)$
- e.g.
 - $\phi(37) = 36$
 - $\phi(21) = (3-1)(7-1) = 2 \times 6 = 12$

Euler's Theorem

- a generalisation of Fermat's Little Theorem
- $a^{\phi(n)} = 1 \pmod{n}$
 - for any a, n where $\gcd(a, n) = 1$
 - if n is prime, this becomes FLT
- e.g.
 - $a=3; n=10; \phi(10)=4;$
hence $3^4 = 81 = 1 \pmod{10}$
 - $a=2; n=11; \phi(11)=10;$
hence $2^{10} = 1024 = 1 \pmod{11}$

Proof of Euler's theorem

Consider $x_1, x_2, \dots, x_{\phi(n)} < n$ and coprime to n

Since a is also coprime to n , from previous result

$$ax_1 \equiv x_i \pmod{n}, ax_2 \equiv x_j \pmod{n}, \dots \text{etc.}$$

$$\Rightarrow a^{\phi(n)} x_1 x_2 x_3 \dots x_{\phi(n)} \equiv x_1 x_2 x_3 \dots x_{\phi(n)} \pmod{n}$$

$$\Rightarrow a^{\phi(n)} x \equiv x \pmod{n} \text{ where } x = x_1 x_2 x_3 \dots x_{\phi(n)}$$

$$\Rightarrow n \mid x(a^{\phi(n)} - 1)$$

But n doesn't divide x

$$\Rightarrow n \mid (a^{\phi(n)} - 1)$$

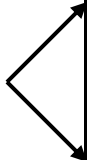
$$\Rightarrow a^{\phi(n)} \equiv 1 \pmod{n}$$

Groups based on modular arithmetic

- group: a set of elements with a binary operation
 - the outcome of the operation should satisfy four properties below
- The group of positive integers modulo a prime p
 $Z_p^* \equiv \{1, 2, 3, \dots, p-1\}$
 $*_p \equiv$ multiplication modulo p
Denoted as: $(Z_p^*, *_p)$
- **Required properties**
 1. Closure. Yes.
 2. Associativity. Yes.
 3. Identity. 1.
 4. Inverse. Yes.
- **Example:** $Z_7^* = \{1, 2, 3, 4, 5, 6\}$
 $1^{-1} = 1, 2^{-1} = 4, 3^{-1} = 5, 6^{-1} = 6$

Other properties

- $|Z_p^*| = (p-1)$
- By Fermat's little theorem: $a^{(p-1)} = 1 \pmod{p}$
- Example of Z_7^*

Generators 

x	x^2	x^3	x^4	x^5	x^6
1	1	1	1	1	1
2	4	1	2	4	1
<u>3</u>	2	6	4	5	1
4	2	1	4	2	1
<u>5</u>	4	6	2	3	1
6	1	6	1	6	1

For all p , the group is cyclic. A cyclic group is a group that can be generated by a single element (the generator)

What if n is not a prime?

- The group of positive integers modulo a non-prime n
 $Z_n \equiv \{1, 2, 3, \dots, n-1\}$, n not prime
 $*_p \equiv$ multiplication modulo n
- **Required properties?**
 1. Closure. ?
 2. Associativity. ?
 3. Identity. ?
 4. Inverse. ?
- How do we fix this?

Groups based on modular arithmetic

- The **multiplicative group modulo n**

$$\mathbb{Z}_n^* \equiv \{m : 1 \leq m < n, \gcd(n,m) = 1\}$$

* \equiv multiplication modulo n

Denoted as $(\mathbb{Z}_n^*, *_n)$

- **Required properties:**

- Closure. Yes.
- Associativity. Yes.
- Identity. 1.
- Inverse. Yes.

- **Example:** $\mathbb{Z}_{15}^* = \{1,2,4,7,8,11,13,14\}$

- $1^{-1} = 1, 2^{-1} = 8, 4^{-1} = 4, 7^{-1} = 13, 11^{-1} = 11, 14^{-1} = 14$

Recap: Euler's Phi Function

$$\phi(n) = |Z_n^*| = n \prod_{p|n} (1 - 1/p)$$

If n is a product of two primes p and q , then

$$\phi(n) = pq(1 - 1/p)(1 - 1/q) = (p - 1)(q - 1)$$

Euler's Theorem:

$$a^{\phi(n)} = 1 \pmod{n} \text{ for } a \in Z_n^*$$

Or for $n = pq$

$$a^{(p-1)(q-1)} = 1 \pmod{n} \text{ for } a \in Z_{pq}^*$$

Law of exponentiation:

$$\text{if } a = b \pmod{\phi(n)} \text{ then } x^a = x^b \pmod{n} \quad x \in Z_n^*$$

This will be very important in RSA!

Exponential Inverses

Finding modular inverses is good enough for decoding simple modular cryptography.

However, in RSA, an encryption consists of *exponentiating* modulo N , i.e. $m^e \bmod N$.

We want to find a different exponent d based on e and N which will give us back m , i.e. we want $m^{ed} \bmod N = m$. In other words, we want an ***exponential inverse*** for e modulo N .

Generators in Groups

- Example of Z_{10}^* : $\{1, 3, 7, 9\}$

Generator
= primitive
element

x	x^2	x^3	x^4
1	1	1	1
<u>3</u>	9	7	1
<u>7</u>	9	3	1
9	1	9	1

Exponential Inverses

To tackle the general problem, start first with the case of N a prime number.

Exponentiation modulo a prime number is well understood.

e.g. exponentiating 3 modulo 7: (3 is generator)

1. $3^1 \bmod 7 = 3$

7. $3^7 \bmod 7 = 3$

2. $3^2 \bmod 7 = 2$

8. $3^8 \bmod 7 = 2$

3. $3^3 \bmod 7 = 6$

9. $3^9 \bmod 7 = 6$

4. $3^4 \bmod 7 = 4$

10. $3^{10} \bmod 7 = 4$

5. $3^5 \bmod 7 = 5$

11. $3^{11} \bmod 7 = 5$

6. $3^6 \bmod 7 = 1$

12. $3^{12} \bmod 7 = 1$

Exponential Inverses

Exponentiating to the $p - 1$ power results in 1 (by FLT).

Therefore, any further exponentiation results in a cycling, with repetitions occurring every 6 exponentiations.

Fermat's Little Theorem says that this effect happens for all relative-prime numbers under prime modulus:

1. $3^1 \bmod 7 = 3$

2. $3^2 \bmod 7 = 2$

3. $3^3 \bmod 7 = 6$

4. $3^4 \bmod 7 = 4$

5. $3^5 \bmod 7 = 5$

6. $3^6 \bmod 7 = 1$

7. $3^7 \bmod 7 = 3$

8. $3^8 \bmod 7 = 2$

9. $3^9 \bmod 7 = 6$

10. $3^{10} \bmod 7 = 4$

11. $3^{11} \bmod 7 = 5$

12. $3^{12} \bmod 7 = 1$

Exponential Inverses

Corollary: If e is relatively prime to $p - 1$, where p is prime, then its *exponential* inverse modulo p exists and is the multiplicative inverse of e modulo $p-1$.

Proof. Supposing $ed \equiv 1 \pmod{p-1}$. Then for some k , $ed = 1 + k(p-1)$. So if a is any number not divisible by p , FLT implies:

$$a^{ed} \equiv a^{1+k(p-1)} \pmod{p} \equiv a \pmod{p}$$

In other words, exponentiating by ed doesn't change numbers, modulo p , so by definition, d and e are exponential inverses. •

Exponential Inverses

E.g. Find the exponential inverse of 3 modulo 11.

$p = 11$, so $p-1 = 10$. The inverse of 3 modulo 10 is **7**, which is the answer.

Exponential Inverses: Next Step

Q: Why don't we just use a prime number as our base N since it's so easy to find the decryptor d ?

Exponential Inverses: Next Step

A: *Because* it's so **easy** to find the decryptor d !

Recall, this is a *public cryptosystem*. The key (N, e) is available to all users including attackers. If a prime N were used, anybody can find the inverse of e modulo $N-1$.

RSA uses next simplest case: $N = pq$ where N is a product of two (different) primes.

Exponential Inverses: Next Step

If we know what p and q are, then we'll be able to find the exponential inverse.

But that's a big **if**.

Factoring large prime numbers is a surprisingly difficult problem.

No one knows how to do this in polynomial time.

RSA: one way function

- Multiplication of two prime numbers is **believed** to be a one-way function.
- We say “**believed**” because nobody has been able to **prove** that it is hard to factorise.

RSA Cryptosystem

Proof of Decryption

Proof that d is inverse of e mod $(p-1)(q-1)$:

We can therefore find k such that $ed = 1 + k(p-1)(q-1)$.

Does m^{ed} equal itself modulo $N = pq$?

$$\begin{aligned} m^{ed} &\equiv m^{1+k(p-1)(q-1)} \pmod{pq}. \\ &\equiv m^1 \cdot m^{k(p-1)(q-1)} \pmod{pq} \\ &\equiv m \cdot m^{k(p-1)(q-1)} \pmod{pq} \\ &\equiv m \cdot (m^{(p-1)(q-1)})^k \pmod{pq} \\ &\equiv m \cdot (1)^k \pmod{pq} \\ &\equiv m \pmod{pq} \end{aligned}$$

RSA Cryptosystem: example

1. Select primes: $p=17$ & $q=11$
2. Compute $N = pq = 17 \times 11 = 187$
3. Compute $\phi(N) = (p-1)(q-1) = 16 \times 10 = 160$
4. Select e : $\gcd(e, 160) = 1$; choose $e=7$
5. Determine d : $ed=1 \pmod{160}$ and $d < 160$
Value is $d=23$ since $23 \times 7 = 161 = 160 + 1$
6. Publish public key $K_U = \{7, 187\}$
7. Keep secret private key $K_R = \{23\}$

RSA Cryptosystem: example

- sample RSA encryption/decryption is:
- given message $M = 88$ (nb. $88 < 187$)

- encryption:

$$C = 88^7 \bmod 187 = 11$$

- decryption:

$$M = 11^{23} \bmod 187 = 88$$

In reality, e is small, d is large!

Chinese Remainder Theorem (CRT)

For each $x \in Z_{15}$, write $x \bmod 3$ and $x \bmod 5$.

Each $x \in Z_{15}$ has a different $x \bmod 3, x \bmod 5$ pair.

Thus, the function $f(x) = (x \bmod 3, x \bmod 5)$ from Z_{15} to the 15 pairs (i, j) with $0 \leq i < 3$ and $0 \leq j < 5$ is one-to-one.

$\Rightarrow x$ is uniquely determined by its pair of remainders.

x	$x \bmod 3$	$x \bmod 5$
0	0	0
1	1	1
2	2	2
3	0	3
4	1	4
5	2	0
6	0	1
7	1	2
8	2	3
9	0	4
10	1	0
11	2	1
12	0	2
13	1	3
14	2	4

CRT

Let $m_1, \dots, m_n > 0$ be relative prime. Then the system of equations $x \equiv a_i \pmod{m_i}$ (for $i=1$ to n) has a **unique solution modulo $m = m_1 \cdot m_2 \cdot \dots \cdot m_n$.**

Proof: Let $M_i = m/m_i$. Thus $\gcd(m_i, M_i)=1$.

as multi. inverse exists, $\exists y_i$ such that $y_i M_i \equiv 1 \pmod{m_i}$.

Now let $x = \sum_i a_i y_i M_i$.

Since $m_i \mid M_k$ for $k \neq i$, we have $M_k \equiv 0 \pmod{m_i}$.

$\therefore x \pmod{m_i} = \sum_l a_l y_l M_l \pmod{m_i} = a_i y_i M_i \pmod{m_i} = a_i \pmod{m_i}$.

Thus, $x \equiv a_i \pmod{m_i}$ holds for each i .

Thus, the congruences hold.

What's x such that:

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

$$x \equiv 2 \pmod{7}?$$

(So, $a_1 = 2$, etc.
and $m_1 = 3$ etc.)

$$m = m_1 \cdot \dots \cdot m_n$$

$$M_i = m/m_i$$

$$y_i M_i \equiv 1 \pmod{m_i}$$

$$x = \sum_i a_i y_i M_i$$

$$m = 3 \times 5 \times 7 = 105$$

$$M_1 = m/3 = 105/3 = 35$$

2 is an inverse of $M_1 = 35 \pmod{3}$ since $35 \times 2 \equiv 1 \pmod{3}$

$$M_2 = m/5 = 105/5 = 21$$

1 is an inverse of $M_2 = 21 \pmod{5}$ since $21 \times 1 \equiv 1 \pmod{5}$

$$M_3 = m/7 = 15$$

1 is an inverse of $M_3 = 15 \pmod{7}$ since $15 \times 1 \equiv 1 \pmod{7}$

$$x \equiv 2 \times 2 \times 35 + 3 \times 1 \times 21 + 2 \times 1 \times 15 = 233 \equiv 23 \pmod{105}$$

So answer: $x \equiv 23 \pmod{105}$

RSA with CRT

- Assuming that M is not divisible by either p or q ,
 - Fermat's Little Theorem tells us that
$$M^{p-1} \equiv 1 \pmod{p} \text{ and } M^{q-1} \equiv 1 \pmod{q}$$
- Thus, we have that the following two congruences hold:

First:
$$C^d \equiv M \cdot (M^{p-1})^{k(q-1)} \equiv M \cdot 1^{k(q-1)} \equiv M \pmod{p}$$

Second:
$$C^d \equiv M \cdot (M^{q-1})^{k(p-1)} \equiv M \cdot 1^{k(p-1)} \equiv M \pmod{q}$$

RSA with CRT

- For faster decryption, we use CRT

$$n = pq$$

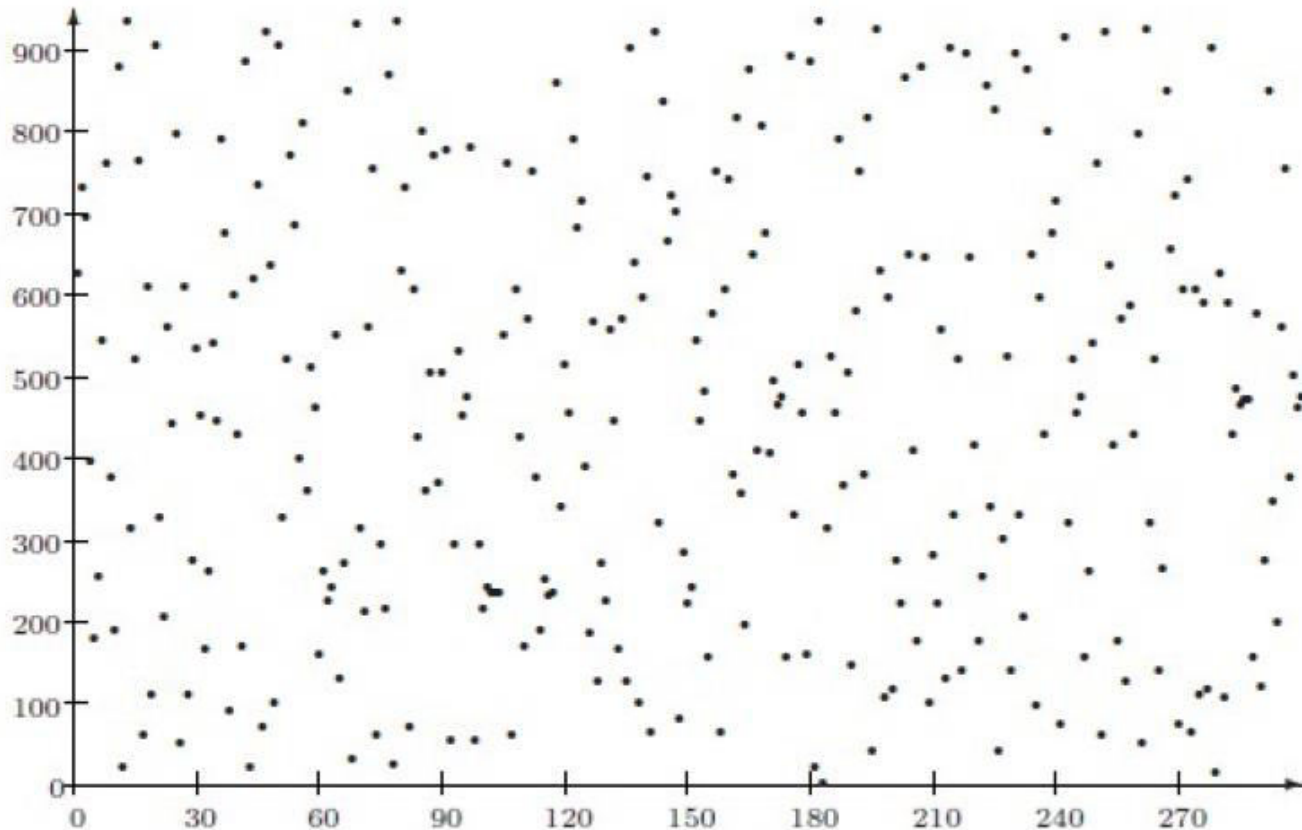
- Decryption: $c^d \bmod n$.
- Instead of computing $c^d \bmod n$ directly, we
 - compute $c_1 = c \bmod p$ and $c_2 = c \bmod q$
 - compute $m_1 = (c_1)^d \bmod p$ and $m_2 = (c_2)^d \bmod q$
 - recover the plaintext by solving
$$\begin{cases} x \equiv m_1 \pmod{p} \\ x \equiv m_2 \pmod{q} \end{cases}$$

Discrete Logarithms

- the inverse problem to exponentiation is to find the **discrete logarithm** of a number modulo p
- that is to find x such that $y = g^x \pmod{p}$
- this is written as $x = \log_g y \pmod{p}$
- if g is a generator then it always exists, otherwise it may not, e.g.
 - $x = \log_3 4 \pmod{13}$ has no answer
 - $x = \log_2 3 \pmod{13} = 4$ by trying successive powers
- whilst exponentiation is relatively easy, finding discrete logarithms is generally a **hard** problem

Discrete logarithm: plotting

A graph of $f(x) = 627^x \bmod 941$ for $x = 1, 2, 3, \dots$



Source: Kaafarani@Oxford Univ.

Primitive Element Theorem

- $Z_p^* = \langle \alpha \rangle$, i.e. $\text{ord}(\alpha) = p-1$.
- Example
 - $Z_7^* = \langle 3 \rangle$ $3^1=3, 3^2=2, 3^3=6, 3^4=4, 3^5=5, 3^6=1$
 - $Z_{13}^* = \langle 2 \rangle$ $2^1=2, 2^2=4, 2^3=8, 2^4=3, 2^5=6, 2^6=12, 2^7=11, 2^8=9, 2^9=5, 2^{10}=10, 2^{11}=7, 2^{12}=1$
- Note. $\text{ord}(\alpha) = p-1 \Rightarrow \{\alpha, \alpha^2, \dots, \alpha^{p-1}\}$ distinct.

* Primitive element = generator

Discrete Logarithms

- Discrete log problem
 - Given $Z_p^* = \langle \alpha \rangle$
 - $\text{Log}_\alpha(y) = x$, if $y = \alpha^x$.
- Example
 - $Z_{13}^* = \langle 2 \rangle$; $2^1=2, 2^2=4, 2^3=8, 2^4=3, 2^5=6, 2^6=12, 2^7=11, 2^8=9, 2^9=5, 2^{10}=10, 2^{11}=7, 2^{12}=1$
 - $\text{Log}_2(5) = 9$.

ElGamal algorithm

- One way function: modular exponentiation
 - Discrete logarithm is hard
- Solving $\text{Log}_\alpha(y)$ has some solutions, each of which is of high complexity
 - None of them run in polynomial time

Setting up ElGamal

- Let **p** be a large prime
 - By “large” we mean here a prime rather typical in length to that of an RSA modulus
- Select a special number **g**
 - The number **g** must be a **primitive element** modulo **p**.
- Choose a private key **x**
 - This can be any number bigger than 1 and smaller than **p-1**
- Compute public key **y** (from **x**, **p** and **g**)
 - The public key **y** is **g** raised to the power of the private key **x** modulo **p**. In other words:

$$y = g^x \text{ mod } p$$

- Publicize **p**, **g**, **y**

Setting up ElGamal: example

Step 1: Let $p = 23$

Step 2: Select a primitive element $g = 11$

Step 3: Choose a private key $x = 6$

Step 4: Compute $y = 11^6 \pmod{23}$
 $= 9$

Public key is 9

Private key is 6

ElGamal encryption

The first job is to represent the plaintext M as a series of numbers modulo p . Then:

1. Generate a random number k (ephemeral key)
2. Compute two values C_1 and C_2 , where

$$\mathbf{C_1 = g^k \text{ mod } p} \quad \text{and} \quad \mathbf{C_2 = My^k \text{ mod } p}$$

3. Send the ciphertext C , which consists of the two separate values C_1 and C_2 .

ElGamal decryption

$$\mathbf{C}_1 = g^k \bmod p \quad \mathbf{C}_2 = My^k \bmod p$$

1 - The receiver begins by using their private key \mathbf{x} to transform \mathbf{C}_1 into something more useful:

$$\mathbf{C}_1^{\mathbf{x}} = (g^k)^{\mathbf{x}} \bmod p$$

NOTE: $\mathbf{C}_1^{\mathbf{x}} = (g^k)^{\mathbf{x}} = (g^{\mathbf{x}})^k = (\mathbf{y})^k = \mathbf{y}^k \bmod p$

2 - This is a very useful quantity because if you divide \mathbf{C}_2 by it you get \mathbf{M} . In other words:

$$\mathbf{C}_2 / \mathbf{y}^k = (\mathbf{M}\mathbf{y}^k) / \mathbf{y}^k = \mathbf{M} \bmod p$$

ElGamal encryption: example

To encrypt $M = 10$ using Public key **9**

1 - Generate a random number $k = 3$

2 - Compute $C_1 = 11^3 \bmod 23 = 20$

$$C_2 = 10 \times 9^3 \bmod 23$$

$$= 10 \times 16 = 160 \bmod 23 = 22$$

3 - Ciphertext $C = (20, 22)$

ElGamal decryption: example

To decrypt $C = (20, 22)$

1 - Compute $20^6 = 16 \pmod{23}$

2 - Compute $22 / 16 = 10 \pmod{23}$

3 - Plaintext = 10