15-251

Great Theoretical Ideas in Computer Science

Number Theory, Cryptography and RSA

Lecture 15, October 13, 2009)

$\text{p-1} \equiv_p 1$

The reduced system modulo n:

$Z_n = \{0, 1, 2, \ldots, n-1\}$

Define operations $+_n$ and $*_n$:

$a +_n b = (a+b \bmod n)$
$a *_n b = (a*b \bmod n)$

$Z_n = \{0, 1, 2, \ldots, n-1\}$

$a +_n b = (a+b \bmod n)$
$a *_n b = (a*b \bmod n)$

$+_n$ and $*_n$ are commutative and associative binary operators from $Z_n * Z_n \rightarrow Z_n$

The reduced system $Z_6 = \{0,1,2,3,4,5\}$

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

An operator has the permutation property if each row and each column has a permutation of the elements.

For every n, $+_n$ on Z_n has the permutation property

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

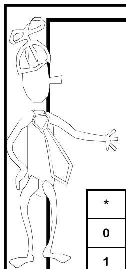
An operator has the permutation property if each row and each column has a permutation of the elements.

What about multiplication? Does $*_6$ on Z_6 have the permutation property?

No

*	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

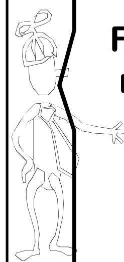
An operator has the permutation property if each row and each column has a permutation of the elements.



Fundamental lemma of plus, minus, and times modulo n:

If ($x \equiv_n y$) and ($a \equiv_n b$). Then

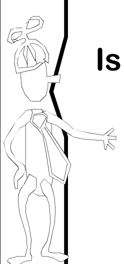
1) $x + a \equiv_n y + b$
2) $x - a \equiv_n y - b$
3) $x * a \equiv_n y * b$



Is there a fundamental lemma of division modulo n?

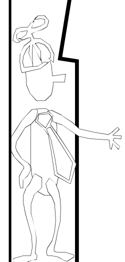
$cx \equiv_n cy \Rightarrow x \equiv_n y$?

No!



When <u>can't</u> I divide by c?

If GCD(c,n) > 1 then you can't always divide by c.



Fundamental lemma of division modulo n.

If GCD(c,n)=1, then $ca \equiv_n cb \Rightarrow a \equiv_n b$

So

Consider the set

$Z_n^* = \{x \in Z_n \mid GCD(x,n) = 1\}$

Multiplication over this set Z_n^* will have the cancellation property.

Euler Phi Function $\phi(n)$

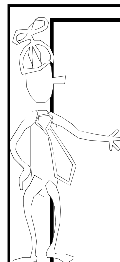
Define $\phi(n)$

=

size of Z_n^*

=

number of $1 \leq k < n$ that are relatively prime to n.



$Z_6 = \{0, 1,2,3,4,5\}$
$Z_6^* = \{1,5\}$

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

*	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

$Z_{12}^* = \{0 \leq x < 12 \mid \gcd(x,12) = 1\}$
$= \{1,5,7,11\}$

$\phi(12) = 4$

$*_{12}$	1	5	7	11
1	1	5	7	11
5	5	1	11	7
7	7	11	1	5
11	11	7	5	1

$Z_5^* = \{1,2,3,4\}$ $= Z_5 \setminus \{0\}$

$*_5$	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

For all primes p, $Z_p^* = Z_p \setminus \{0\}$,
since all $0 < x < p$ satisfy $\gcd(x,p) = 1$

If p prime
then $\phi(p) = (p-1)$

If p,q distinct primes
then $\phi(pq) = (p-1)(q-1)$

If p prime
then $\phi(p^2) = ?$

What are the properties of Z_n^*

For $*_n$ on Z_n^* the following properties hold

[Closure]
$x, y \in Z_n \Rightarrow x *_n y \in Z_n$

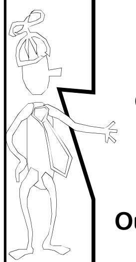
[Associativity]
$x, y, z \in Z_n \Rightarrow (x *_n y) *_n z = x *_n (y *_n z)$

[Commutativity]
$x, y \in Z_n \Rightarrow x *_n y = y *_n x$

The additive inverse of $a \in Z_n$
is the unique $b \in Z_n$ such that
$a +_n b \equiv_n 0$.

We denote this inverse by “–a”.

It is trivial to calculate:
“-a” = (n-a).



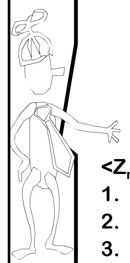
Efficient algorithm to find multiplicative inverse a^{-1} from a and n.

Extended Euclidean Algorithm(a, n)

Get r,s such that $ra + sn = gcd(a,n) = 1$

So $ra \equiv_n 1$

Output: r is the multiplicative inverse of a



$Z_n = \{0, 1, 2, \ldots, n-1\}$
$Z_n^* = \{x \in Z_n \mid GCD(x,n) = 1\}$

Define $+_n$ and $*_n$:
$a +_n b = (a+b \bmod n)$
$a *_n b = (a*b \bmod n)$

$\langle Z_n, +_n \rangle$
1. Closed
2. Associative
3. 0 is identity
4. Additive Inverses
5. Cancellation
6. Commutative

$\langle Z_n^*, *_n \rangle$
1. Closed
2. Associative
3. 1 is identity
4. Multiplicative Inverses
5. Cancellation
6. Commutative

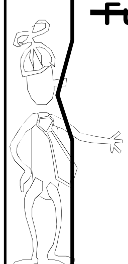
$c *_n (a +_n b) \equiv_n (c *_n a) +_n (c *_n b)$

new stuff starts here…

Fundamental Lemmas until now

For x, y, a, b in Z_n, ($x \equiv_n y$) and ($a \equiv_n b$).
Then
1) $x + a \equiv_n y + b$
2) $x - a \equiv_n y - b$
3) $x * a \equiv_n y * b$

For a,b,c in Z_n^*
then $ca \equiv_n cb \Rightarrow a \equiv_n b$

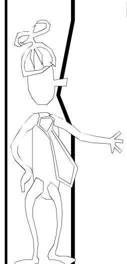


~~Fundamental lemma of powers?~~

If ($a \equiv_n b$)
Then $x^a \equiv_n x^b$?

NO!

($2 \equiv_3 5$) , but it is not the case that:
$2^2 \equiv_3 2^5$



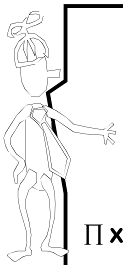
By the permutation property, two names for the same set:

$Z_n^* = aZ_n^*$

where
$aZ_n^* = \{a *_n x \mid x \in Z_n^*\}, a \in Z_n^*$

Example: Z_5^*

*	1	2	3	4
1	1	2	3	4
2	2	4	1	3
a	3	1	4	2
4	4	3	2	1



Two products on the same set:

$Z_n^* = aZ_n^*$

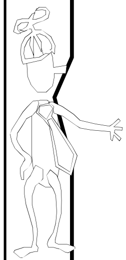
$aZ_n^* = \{a *_n x \mid x \in Z_n^*\}, a \in Z_n^*$

$\prod x \equiv_n \prod ax$ [as x ranges over Z_n^*]

$\prod x \equiv_n \prod x$ ($a^{\text{size of } Zn^*}$) [Commutativity]

$1 =_n a^{\text{size of } Zn^*}$ [Cancellation]

$$a^{\Phi(n)} =_n 1$$

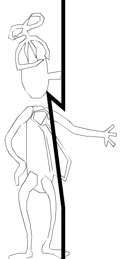


Euler's Theorem

$a \in Z_n^*, a^{\Phi(n)} \equiv_n 1$

Fermat's Little Theorem

p prime, $a \in Z_p^* \Rightarrow a^{p-1} \equiv_p 1$



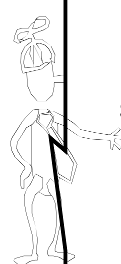
(Correct) Fundamental lemma of powers.

Suppose $x \in Z_n^*$, and a,b,n are naturals.

If $a \equiv_{\Phi(n)} b$ Then $x^a \equiv_n x^b$

Equivalently,

$x^a \equiv_n x^{a \bmod \Phi(n)}$

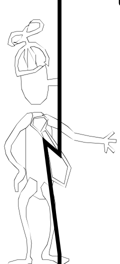


Defining negative powers

Suppose $x \in Z_n^*$, and a,n are naturals.

x^{-a} is defined to be the multiplicative inverse of x^a

$x^{-a} = (x^a)^{-1}$



Rule of integer exponents

Suppose $x,y \in Z_n^*$, and a,b are integers.

$(xy)^{-1} \equiv_n x^{-1} y^{-1}$

$X^a X^b \equiv_n X^{a+b}$

A note about exponentiation

How do you calculate

$2^{6666666666661}$ mod 7

Fundamental lemma of powers.

Suppose $x \in Z_n^*$, and a,n are naturals.

$x^a \equiv_n x^{a \bmod \Phi(n)}$

For $x \in Z_x^*$, $x^a \pmod n = x^{a \bmod \Phi(n)} \pmod n$

Time to compute

To compute $a^x \pmod n$ for $a \in Z_n^*$
first, get $x' = x \bmod \Phi(n)$

By Euler's theorem: $a^x = a^{x'} \pmod n$

Hence, we can calculate $a^{x'}$ where $x' \cdot n$.

But still that might take $x'-1 \approx n$ steps
if we calculate a, a^2, a^3, a^4, …, $a^{x'}$

Faster exponentiation

How do you compute $a^{x'}$ fast?

Suppose $x' = 2^k$	Suppose $2^k \leq x' < 2^{k+1}$
a	a
$\to a^2 \pmod n$	$\to a^2 \pmod n$
$\to a^4 \pmod n$	$\to a^4 \pmod n$
…	…
$\to a^{2^{k-1}} \pmod n$	$\to a^{2^{k-1}} \pmod n$
$\to a^{2^k} \pmod n$	$\to a^{2^k} \pmod n$

multiply together the appropriate powers

How much time did this take?

Only 2 log x' multiplications

Instead of (x'-1) multiplications

Ok, back to number theory

Agreeing on a secret

Alice and Bob have never talked before
but they want to agree on a secret…

How can they do this?

Diffie-Hellman Key Exchange

Alice:
- Picks prime p, and a value g in Z_p^*
- Picks random a in Z_p^*
- Sends over p, g, g^a (mod p)

Bob:
- Picks random b in Z_p^*, and sends over g^b (mod p)

Now both can compute g^{ab} (mod p)

What about Eve?

Alice:
- Picks prime p, and a value g in Z_p^*
- Picks random a in Z_p^*
- Sends over p, g, g^a (mod p)

Bob:
- Picks random b in Z_p^*, and sends over g^b (mod p)

Now both can compute g^{ab} (mod p)

If Eve's just listening in,
she sees p, g, g^a, g^b

It's believed that computing g^{ab} (mod p) from just this information is not easy…

btw, discrete logarithms seem hard

Discrete-Log:
Given p, g, g^a (mod p), compute a

How fast can you do this?

If you can do discrete-logs fast,
you can solve the Diffie-Hellman problem fast.

How about the other way? If you can break the DH key exchange protocol, do discrete logs fast?

The RSA Cryptosystem

Our dramatis personae

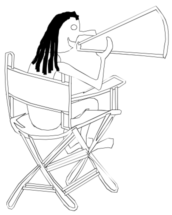
Rivest

Shamir

Adleman

Euler

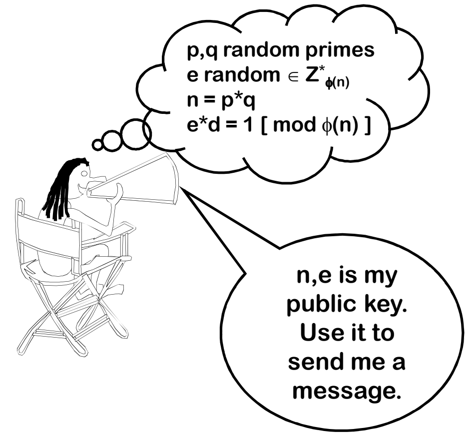
Fermat

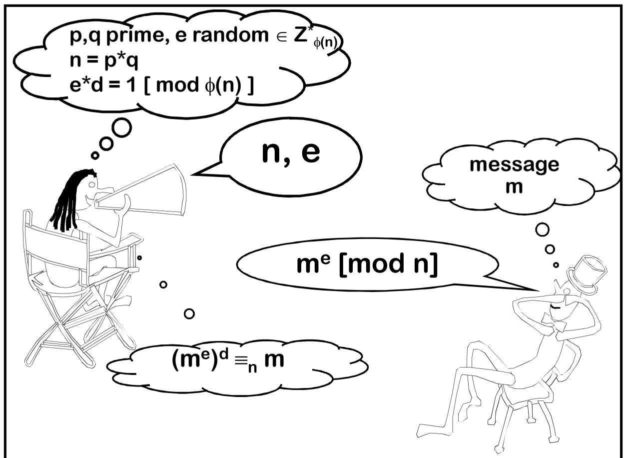


Pick secret, random large primes: p,q
Multiply $n = p*q$
"Publish": n

$\phi(n) = \phi(p)\,\phi(q) = (p-1)*(q-1)$
Pick random $e \in Z^*_{\phi(n)}$
"Publish": e

Compute d = inverse of e in $Z^*_{\phi(n)}$
Hence, $e*d = 1$ [mod $\phi(n)$]
"Private Key": d





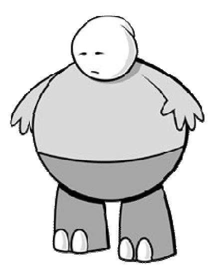
How hard is cracking RSA?

If we can factor products of two large primes, can we crack RSA?

If we know \phi(n), can we crack RSA?

How about the other way? Does cracking RSA mean we must do one of these two?

We don't know..



Here's What You Need to Know…

Fundamental lemma of powers
- Euler phi function $\phi(n) = |Z_n^*|$
- Euler's theorem
- Fermat's little theorem

Fast exponentiation

Diffie-Hellman Key Exchange

RSA algorithm