

15-251

Great Theoretical Ideas in Computer Science

Clarifications of some of the homework Problems.

Problem 1

Problem 5

Problem 6

Raising numbers to powers, Cryptography and RSA,

Lecture 14 (February 25, 2010)



$$a^{p-1} \equiv_p 1$$

It's all done
with splay
trees.

QED.

Class
Dismissed.



Not

How do you compute...

5^8 using few multiplications?

First idea:

$$5 \cdot 5^2 \cdot 5^3 \cdot 5^4 \cdot 5^5 \cdot 5^6 \cdot 5^7 \cdot 5^8 \\ = 5 \cdot 5 \cdot 5^2 \cdot 5^2 \cdot 5$$

How do you compute...

$$5^8$$

Better idea:

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

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

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?

$$5 \quad 5^2 \quad 5^4 \quad 5^8 \quad 5^{16}$$

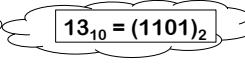
too high! what now?
assume no divisions allowed...

How do you compute...

$$5^{13}$$

Use repeated squaring again?

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

Note that $13 = 8 + 4 + 1$ 
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

How do you compute...

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

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

$$5 \quad 5^2 \quad 5^4 \quad 5^8 \quad 5^{12} \quad 5^{13} = 1 \quad 220 \quad 703 \quad 125$$

$$= 5^{8+5} = 5^{12} \cdot 5$$

then take the answer mod 11

$$1220703125 \pmod{11} = 4$$

How do you compute...

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

Better idea: keep reducing the answer mod 11

$$5 \quad 5^2 \quad 5^4 \quad 5^8 \quad 5^{12} \quad 5^{13}$$

$$\quad \quad \quad \begin{matrix} 25 \\ \cdot_{11} 3 \end{matrix} \quad \begin{matrix} 5^4 \\ \cdot_{11} 9 \end{matrix} \quad \begin{matrix} 81 \\ \cdot_{11} 4 \end{matrix} \quad \begin{matrix} 36 \\ \cdot_{11} 3 \end{matrix} \quad \begin{matrix} 15 \\ \cdot_{11} 4 \end{matrix}$$

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...

$$5^{121242653} \pmod{11}$$

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...

First, recall...

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

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

Fundamental lemmas mod 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) $x - a \equiv_n y - b$
- 4) $cx \equiv_n cy \Rightarrow a \equiv_n b$ i.e., if $c \in \mathbb{Z}_n^*$

Euler Phi Function $\phi(n)$

$\phi(n)$ = size of $\mathbb{Z}_n^*$

p prime $\Rightarrow \phi(p) = p-1$

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

~~Fundamental lemma of powers?~~

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

NO!

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

(Correct) Fundamental lemma of powers.

If $a \in \mathbb{Z}_n^*$ and $x \equiv_{\phi(n)} y$ then $a^x \equiv_n a^y$

Equivalently,

for $a \in \mathbb{Z}_n^*$, $a^x \equiv_n a^{x \bmod \phi(n)}$

How do you compute...

$5^{121242653} \pmod{11}$

$121242653 \pmod{10} = 3$

$5^3 \pmod{11} = 125 \pmod{11} = 4$

Why did we
take mod 10?

for $a \in \mathbb{Z}_n^*$, $a^x \equiv_n a^{x \bmod \phi(n)}$

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

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

$$343281^{327847324} \bmod 39$$

Step 1: reduce the base mod 39

Step 2: reduce the exponent mod $\hat{A}(39) = 24$

NB: you should check that $\gcd(343280, 39) = 1$ to use lemma of powers

Step 3: use repeated squaring to compute 3^4 , taking mods at each step

(Correct) Fundamental lemma of powers.

If $a \in \mathbb{Z}_n^*$ and $x \equiv_{\hat{A}(n)} y$ then $a^x \equiv_n a^y$

Equivalently,

for $a \in \mathbb{Z}_n^*$, $a^x \equiv_n a^{x \bmod \hat{A}(n)}$

How do you prove the lemma for powers?

Use Euler's Theorem

For $a \in \mathbb{Z}_n^*$, $a^{\hat{A}(n)} \equiv_n 1$

Corollary: Fermat's Little Theorem

For p prime, $a \in \mathbb{Z}_p^* \Rightarrow a^{p-1} \equiv_p 1$

Proof of Euler's Theorem: for $a \in \mathbb{Z}_n^*$, $a^{\hat{A}(n)} \equiv_n 1$

Define a $\mathbb{Z}_n^* = \{a \cdot x \mid x \in \mathbb{Z}_n^*\}$ for $a \in \mathbb{Z}_n^*$

By the cancellation property, $\mathbb{Z}_n^* = a\mathbb{Z}_n^*$

$\prod x \equiv_n \prod ax$ [as x ranges over $\mathbb{Z}_n^*$]

$\prod x \equiv_n \prod x$ (a size of $\mathbb{Z}_n^*$) [Commutativity]

$1 \equiv_n a^{\text{size of } \mathbb{Z}_n^*}$ [Cancellation]

$$a^{\hat{A}(n)} \equiv_n 1$$

Please remember

Euler's Theorem

For $a \in \mathbb{Z}_n^*$, $a^{\hat{A}(n)} \equiv_n 1$

Corollary: Fermat's Little Theorem

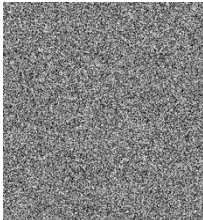
For p prime, $a \in \mathbb{Z}_p^* \Rightarrow a^{p-1} \equiv_p 1$

Basic Cryptography

One Time Pads

The meeting
will be
in the town hall
at
midnight!

Come with your
parole officer!

A square image of a noisy, textured surface, possibly representing a one-time pad or a noisy channel. The texture is dense and granular, with varying shades of gray and black, resembling a close-up of a rough material or a heavily noisy digital image.

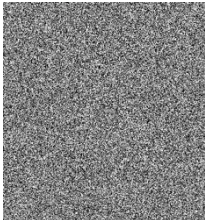
The meeting
will be
in the town hall
at
midnight!

Come with your
parole officer!

**Come with your
parole officer!**

One Time Pads

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Aliquam congue dolor et nunc. Vestibulum varius.
 Vivamus gravida quam et arcu. Sed aceler velit et ma-
 trices. Incommodo nuncup et ante. Aliquam congue. Aliquam
 tristique. In digna velit. Vivamus ante. In hac
 habitant odio, diam. Pellentesque porttitor nunc eget
 tristique. Nullam velit tristique. Ut nulla lacinia. Aenean
 pellentesque in. Vivit. Sed ne ante. Pivis eros. Donec
 dignum lacinia erat. Consequat gravida.



they give perfect security!

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam posuere dolor et neque. Vivamus sed lorem. Vivamus gravida quam et arcu. Sed arcuor velit et tui. Donec venenatis augue et nisi. Aliquam erat. Aliquam uttupta. In ligula velit, scelerisque ac lacina quis. Sed nisi arcu, donec vestibulum. Nulla in nulla quis. Sed nisi. Nullam vel tortor. Cras in tortor, quis sed, pellentesque in, velit. Sed ne sit. Proin orci. Donec dignissim tempor erat. Quisque gravida.

But reuse is bad

XOR =

Comrade officer, I am writing to inform you that I have received your letter of the 15th. I am sorry to hear that you are having trouble with your work. I will try to help you in any way I can. Please let me know if you need any more information. Sincerely,
John Doe
Officer

Can do other attacks as well



==

[illegible]

Can do other attacks as well

Agreeing on a secret

One time pads rely on having a shared secret!

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

How can they do this?

A couple of small things

A value g in Z_n^* “generates” Z_n^* if
 $g, g^2, g^3, g^4, \dots, g^{A(n)}$
contains all elements of Z_n^*

Diffie-Hellman Key Exchange

Alice:

- Picks prime p , and a generator g in Z_p^*
- Picks random a in Z_p^*
- Sends over $p, g, g^a \pmod{p}$

Bob:

- Picks random b in Z_p^* , and sends over $g^b \pmod{p}$

Now both can compute $g^{ab} \pmod{p}$

6

What about Eve?

Alice:
 Picks prime p , and a value g in $\mathbb{Z}_p^*$
 Picks random a in $\mathbb{Z}_p^*$
 Sends over $p, g, g^a \pmod{p}$

Bob:
 Picks random b in $\mathbb{Z}_p^*$, and sends over $g^b \pmod{p}$

Now both can compute $g^{ab} \pmod{p}$

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

It's believed that computing $g^{ab} \pmod{p}$ from just
 this information is not easy...

also, discrete logarithms seem hard

Discrete-Log:

Given $p, g, g^a \pmod{p}$, 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?

Diffie Hellman requires both parties
 to exchange information to share a secret

can we get rid of this assumption?

The RSA Cryptosystem

Our dramatis personae



Rivest



Shamir



Adleman



Euler



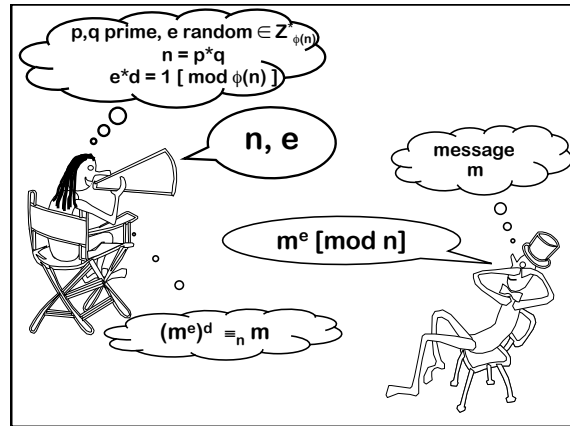
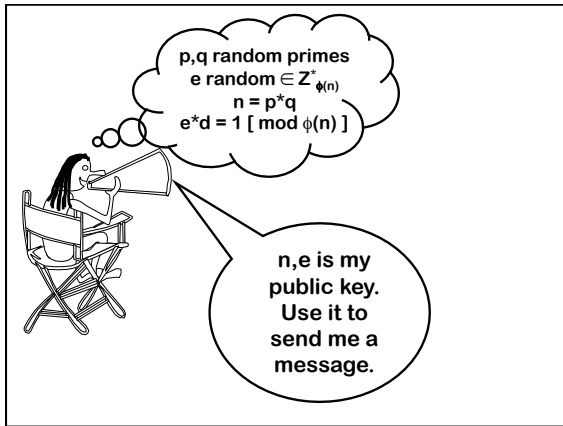
Fermat



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

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

Compute $d = \text{inverse of } e \text{ in } \mathbb{Z}_{\phi(n)}^*$
 Hence, $e \cdot d = 1 \pmod{\phi(n)}$
 "Private Key": d



Alice

$p=11$
 $q=3$
 $e=3$
 $\phi(p)=20$
 $d = 3^{-1} \bmod 20 = 7$

Maxille

$n=33$
 $e=3$

Bob

$m=7$

$7^e \bmod n = 7^3 \equiv_n 13$

$(13)^d \equiv 13^7 \bmod 33$
 $\equiv 13^3 \cdot 13^3 \cdot 13$
 $\equiv (2192) \cdot (2192) \cdot 13 \equiv 19 \cdot 19 \cdot 13 \equiv 4693 \equiv 7$

Extended gcd on ocaml.
 (egcd a b) returns a triple (x,y,g) such
 That g is the gcd(a,b) and $g=ax+by$.

let rec egcd a b =
 if a = 0 then (0,1,b) else
 let (x,y,g) = egcd (b mod a) a in
 (y-(b/a)*x, x, g)

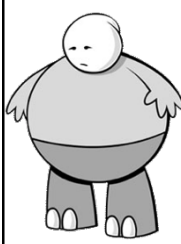
How hard is cracking RSA?

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

If we know n and $\hat{A}(n)$, can we crack RSA?

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

We don't know (yet)...



Here's What
 You Need to
 Know...

- Fast exponentiation
- Fundamental lemma of powers
 - Euler phi function $\phi(n) = |\mathbb{Z}_n^*|$
 - Euler's theorem
 - Fermat's little theorem
- Diffie-Hellman Key Exchange
- RSA algorithm