Mathematics for Computer Science
MIT 6.042J/18.062J

Euler's Theorem
RSA encryption

Inverses mod n

Thm. If k is relatively prime to n, there is an inverse k'

$$k \cdot k' \equiv 1 \pmod{n}$$

Cor.

OK to cancel (mod n)

The interval from 0 to n

$$[0,n) ::= \{0,1,\ldots,n-1\}$$
$$[0,n] ::= \{0,1,\ldots,n\}$$

Euler ϕ function

$\phi(n) ::=$ # $k \in [0,n)$ s.t.
k rel. prime to n

$\phi(7) = 6$ 1,2,3,4,5,6

$\phi(12) = 4$
1,2,3,4,5,6,7,8,9,10,11

Calculating ϕ

If p prime, everything from 1 to $p-1$ is rel. prime to p, so

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

Euler ϕ function

$\phi(49)$?

0,1,2,3,4,5,6,7,8,9,…,13,14,15,…,21,…

every 7th number is divisible by 7

so, $\phi(49) = 49-7$

Calculating ϕ

For $[0,p^k)$ every pth element is not rel. prime to p^k:

$0,1,\ldots,p-1,p,\ldots,2p,\ldots,(p^{k-2})p,\ldots,p^k-1$

$(1/p)p^k$ elements not rel. prime to p^k

Calculating ϕ

$$\phi(p^k) = p^k - p^{k-1}$$

Calculating ϕ

For $1,2,\ldots,p-1,p,\ldots,2p,\ldots,p^k-1,p^k$ every pth is not rel. prime to p^k

$$\phi(p^k) = p^k - p^{k-1}$$

Calculating ϕ

Lemma:

For a,b relatively prime,

$$\phi(a\cdot b) = \phi(a)\cdot\phi(b)$$

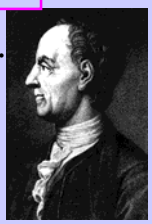
pf: Pset 7 now;
another way in 3 weeks

Euler's Theorem

For k relatively prime to n,

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

Fermat Thm a special case.
Euler proof essentially same as Fermat:



Proof of Euler's Thm

For k relatively prime to n, let $r ::= \phi(n)$ and

$k_1,\ldots,k_r$

the integers in $[0,n)$ relatively prime to n. Then $\text{rem}(k_1k,n), \text{rem}(k_2k,n),\ldots, \text{rem}(k_rk,n)$ is a permutation of $k_1,\ldots,k_r$.

pf: cancel $k \pmod n$.

Proof of Euler's Thm

So

$k_1 \cdots k_r = \text{rem}(k_1 k, n) \cdots \text{rem}(k_r k, n)$

$\equiv k_1 k \cdots k_r k \pmod{n}$

$= k^r \cdot k_1 \cdots k_r \pmod{n}$

But OK to cancel $k_1, \cdots, k_r$, so

$1 \equiv k^r \pmod{n}$.

RSA Public Key Encryption

Beforehand

- receiver generates primes p,q
- n ::= pq
- selects e rel. prime to (p-1)(q-1)
- (e, n) ::= public key, publishes it
- finds d , inverse mod (p-1)(q-1) of e
- d is secret key, keeps hidden

Receiver's abilities

- find two large primes p, q
  - ok because: lots of primes
  - fast test for primality
- find e rel. prime to (p-1)(q-1)
  - ok: lots of rel. prime nums
  - gcd easy to compute
- find inverse of e
  - easy using Pulverizer or Euler

RSA

- Encoding message m:
  send $m' ::= \text{rem}(m^e, n)$
- Decoding m':
  receiver computes
  $\text{rem}((m')^d, n) = m$

Why does this work?

...explained in Team Problem

Why is it secure?

- easy to break *if* can factor n (find d same way receiver did)
- conversely, from d can factor n
- but factoring appears hard
- has withstood 25 years of attacks

Team Problems

Problems 1&2