Mathematics for Computer Science
MIT 6.042J/18.062J

RSA encryption

Public Key Cryptosystem

Anyone can send a secret (encrypted) message to the receiver, without any prior contact, using publicly available info.

Public Key Cryptosystem

This sounds paradoxical: how can secrecy be possible using only public info?
Actually has paradoxical consequences.

Mental Chess

Chess masters can play without having a chess board:
"mental chess."
OK, how about "mental poker"?
--I'll deal.

No joke! It's possible.

One-way functions

The paradoxical assumption is that there are one-way functions that are easy to compute but hard to invert.

In particular,

- it is easy to compute the product n of two (large) primes p and q.
- But given n, it is generally very hard to factor n to recover p and q

The RSA Protocol

RSA Public Key Encryption

Photograph removed due to copyright restrictions.
See here: http://ams.org/samplings/feature-column/fcarc-internet (under Public Key Systems)

Shamir Rivest Adleman

Beforehand

receiver generates primes p, q

$n ::= p \cdot q$

selects e rel. prime to (p-1)(q-1)

(e, n) ::= public key, publishes it

finds $d ::= e^{-1} \quad (\mathbb{Z}^*_{(p-1)(q-1)})$

d is private key, keeps hidden

RSA

Encoding message $m \in [1,n)$

send $m\hat{} ::= m^e \quad (\mathbb{Z}_n)$

Decoding $m\hat{}$:

receiver computes

$$m = (m\hat{})^d \quad (\mathbb{Z}_n)$$

Why does this work?

follows easily from Euler's Theorem when

$$m \in \mathbb{Z}_n^*$$

Why does this work?

actually works for all m ... explained in Class Problem

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 $e^{-1} \; (\mathbb{Z}^*_{(p-1)(q-1)})$

- easy using Pulverizer

lots of primes

Prime Number Thm:

$\pi(n) ::= |\text{primes} \leq n|$

$\sim n/\ln n$ (deep thm)

Chebyshev's bound:

$\pi(n) > n/4 \log n$

"elementary" proof

lots of primes

so for 200 digit #'s,
at least 1/1000 is prime

Chebyshev's bound:

$\pi(n) > n/4 \log n$

"elementary" proof

Fermat Primality Test

check if

$$a^{n-1} = 1 \quad (\mathbb{Z}_n)$$

if fails, not prime (Fermat)

choose random a in [1,n).

if not prime, Pr(fails)>1/2

(with rare exceptions)

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
  so finding d must also be hard)
- RSA has withstood 35 years of attacks

6.042J / 18.062J Mathematics for Computer Science
Spring 2015