DTTF/NB479: Dszquphsbqiz Day 19

- Announcements:
  - DES due now
  - Chapter 3 Exam tomorrow
    - No cheat sheets allowed
  - Term project groups and topics due end of next week
    - Use ch 10 – 19 as inspiration
- Today
  - Using RSA: practical considerations
- Questions?

RSA (Rivest – Shamir – Adelman)

For Alice to send a message to Bob.

- Bob chooses primes p,q (large, ~100 digits each)
- He publishes his public key (n,e):
  - $n = pq$
  - e, a large number such that $\gcd(e, (p-1)(q-1)) = 1$

- Alice has a message $m < n$.
  - Otherwise (if $m > n$), break message into chunks $< n$
- Alice sends $c = m^e (\text{mod } n)$
- Bob computes $c^d \ (\text{mod } n) = (m^e)^d = m \ (\text{mod } n)$.
- What does he use for d?

The security of RSA lies in the difficulty of factoring products of large primes

Are there any shortcuts to decryption?

Consider:

1. Can we find the e^{th} root of $c = m^e$ quickly?

   No, since mod n

2. Is $\phi(n)$ as hard to find as factors p and q?

   Yesterday: yes, using $n - \phi(n) + 1$ and the quadratic formula

3. Is finding d directly is as hard to do as finding p and q?

   Next week: yes!

Toy example

- Alice – (m) → Bob
- Bob's key:
  - $n = pq = (13)(17) = 221$
  - $e = 35$: $\gcd(e, (p-1)(q-1)) = 1$
  - $d=e^{-1} \bmod 192$ exists:
    d = __11__
- m = 20 (letter t)
  - 1-based, so leading 'a' = 1 not ignored
- $c = m^e (\bmod n)$ = _197___
- $c^d (\bmod n)$ = _20__

Issues:

How to compute $20^{35} (\bmod 221)$?

Efficiency is O(log e)

How to compute d?

Extended Euclidean alg.

- And why is this secure?
  - Why can't Eve calculate d herself?

Example with larger numbers

- Maple's worksheet mode
- For some reason, inert power (&^) only works for me when entering in the red (single-line exponents) entry-mode; press F5 (or |> button) to toggle.
- myConcatenator is a lambda expression.

> $msg := convert(\text{"hello"}, bytes);$

$$msg := [104, 101, 108, 108, 111]$$

> $myConcatenator := arr \to sum(arr[i] \cdot 1000^{(i-1)}, i = 1 .. nops(arr));$

$$myConcatenator := arr \to \sum_{i=1}^{nops(arr)} arr_i\, 1000^{(i-1)}$$

> $m := myConcatenator(msg);$

$$m := 111108108101104$$

> $p := nextprime(10^{20});$

$$p := 100000000000000000039$$

> $q := nextprime(10^{21});$

$$q := 1000000000000000000117$$

>

> $n := p \cdot q;$

$$n := 100000000000000000050700000000000000004563$$

> $e := 65537;$

$$e := 65537$$

> $\phi := (p-1) \cdot (q-1);$

$$\phi := 100000000000000000049600000000000000004408$$

```
> c := m &^ e mod n;
```

$$c := 41172530747560554631603662398453570506594$$

>

```
> d:= e&^(-1) mod phi;
```

$$d := 45366739399118055472095262218288905505761$$

```
> dec := c&^d mod n;
```

$$dec := 111108108101104$$

> $convert(dec, base, 1000);$

$$[104, 101, 108, 108, 111]$$

> $convert(\%, bytes);$

"hello"

>