RSA

Public-key cryptography

- Everyone has a public + private key.
  - public + advertised.
  - completely secret
- To send a message, look up public key of recipient.
- Encode message using public key.
- Recipient decodes message using private key.

Some facts

- Multiplying large #s can be done quickly
- Factoring large #s (as far as we know!) cannot.
- BUT testing whether a large # is prime can be done quickly!

RSA — 1977 Rivest, Shamir, Adleman (Clifford Cocks)

- Pick two large primes p and q. (Generate random #s and test.)
- Compute $n = pq$.
- Choose e such that $1 < e < (p-1)(q-1)$ and $\gcd(e, (p-1)(q-1)) = 1$.
- Find d such that $de \equiv_{(p-1)(q-1)} 1$ i.e. d is multiplicative modular inverse of e, mod $(p-1)(q-1)$.
- p, q, and d are the secret key.
- n and e are the public key.

To encode a message M:

1. Turn M into a number $< n$. (Make multiple messages if too large.)

2. Compute $C = M^e \bmod n$. ← 1. Can be done efficiently
   ← 2. Hard to undo (as far as we know).

   C: encoded message

To decode:

Compute $C^d \bmod n$.

Claim: $(M^e \bmod n)^d \bmod n = M$.

i.e. $(M^e)^d \equiv_n M$.

Proof. Since $de \equiv_{(p-1)(q-1)} 1$, there exists k such that

$de = 1 + k(p-1)(q-1)$. Also, by Fermat's Little Theorem, if $p \nmid M$,

$$M^{p-1} \equiv_p 1.$$

So,

$$(M^e)^d = M^{ed} = M^{1+k(p-1)(q-1)} = M \cdot (M^{p-1})^{k(q-1)}$$

$$\equiv_p M \cdot 1^{k(q-1)} = M.$$

Hence $(M^e)^d \equiv_p M$.

By the same argument, $(M^e)^d \equiv_q M$.

By the Chinese Remainder Theorem, $(M^e)^d \equiv_{pq} M$.