RSA Encryption

Symmetric Vs Asymmetric Encryption

- Symmetric:
  - Same key used in encryption and decryption
- Asymmetric:
  - One key used for encryption, and another for decryption

RSA Keys

- RSA is an asymmetric encryption method.
  - It has 2 keys that are both integers.
  - Encryption key is called e.
  - Decryption key is called d.

Key Generation

- Choose two very large prime numbers, call them p and q.
  - The encryption key $n = pq$
  - Compute $\varphi(n)$ it will be used later.
    - $\varphi(n) = \varphi(p)\varphi(q) = (p - 1)(q - 1)$

Encryption Key

- Choose some integer e where $1 < e < \varphi(n)$ and where e and $\varphi(n)$ are coprime.
  - This is called the public key.

Decryption Key

- Find the modular multiplicative inverse of $e\ (mod\ \varphi(n))$
  - We say that $d = e^{-1}(mod\ \varphi(n))$
  - d is the private key.

Encryption

- Suppose that we have some information characterized by the integer m.
  - m is the plaintext
- To Encrypt:
  - Compute $c = m^e \ (mod\ n)$
  - c is the ciphertext

Decryption

- If we have an RSA ciphertext encrypted with the key e we decrypt it as follows:
  - $m = c^d (mod\ n)$

Justification

- Why does the decryption work?
- $c \equiv m^e \ (mod\ n) \rightarrow c^d \equiv (m^e)^d \ (mod\ n)$
- We want to show that $m \equiv m^{ed} (mod\ n)$
- Recall how we chose e and d.
- $ed \equiv 1 (mod\ \varphi(n))$
- For some natural number h, we get $ed = 1 + h\ \varphi(n)$
- $m^{ed} = m^{1+h\ \varphi(n)} = m\left(m^{\varphi(n)}\right)^h \equiv m(1)^h \equiv m\ (mod\ n)$
    - This is just Euler's theorem in disguise

Security

- Public key components: n and e
- Private key components: $p, q, \varphi(n)$ and d.
- If $n = pq$, why is n public, but p and q private?
  - The n's used in practice have hundreds of digits.
  - It is very hard to factor large semiprime numbers (product of 2 primes)

Security

- What are some security considerations for RSA?

Example RSA key

- RSA-640 (640 bits) =
  3107418240490043721350750035888567930037346022842727545720161948823206440518081504
  5563468296717232867824379162728380334154710731085019195485290073377248227835257423
  8645401469173660247765234660 9
- RSA-640 =
  1634733645809253848443133883865090859841783670033092312181110852389333100104508151
  212118167511579
  ×
  1900871281664822113126851573935413975471896789968515493666638539088027103802104498
  957191261465571

- This took a 5 moths on 80 2.2 GHz AMD Opteron CPU to figure out. (2005)
- Windows uses primes that are 1024 bits in length which means that n is 2048 bits long (over 3 times longer than this).

Intractable Problems

- Prime factorization of very large numbers is an example of an intractable problem.
- It is technically possible to solve, but with computational capabilities it can be hard in any useful time.