csci54 – discrete math & functional programming
RSA
Transmitting information - cryptography

- goal is to keep someone with access to the channel from finding out information about the message.
- assumptions (for now)
  - message = message'
  - codeword = codeword'
- why?
- how?
Private key cryptography

- Symmetric-key algorithms
- The communicating parties share a piece of secret information (the key k)
Public key cryptography

- asymmetric-key algorithm
- Everyone who wants to receive messages generates a public/private key pair and publishes their public key.
- To send a message to someone, you encrypt it with their public key.
- When you receive a message you decrypt it with your private key.
RSA algorithm

- A very widely used public key encryption algorithm

- Three algorithmic components
  - key generation
  - encryption
  - decryption

- Our plan
  - What is the algorithm?
  - Why does it work?
  - How to implement it efficiently?
Greatest common divisor (gcd)

- gcd(a,b) is the largest positive integer that divides both a and b without a remainder.

Practice:
- gcd(14, 63)
- gcd(23, 5)
- gcd(100, 9)

if gcd (a,b) = 1 then:
- a and b have no factors in common
- we say that a and b are relatively prime
- there exists an integer x such that ax = 1 (mod b)
RSA algorithm: key generation

1. Choose a bit-length $k$

2. Choose two primes $p$ and $q$ which can be represented with $k$ bits

3. Let $n = pq$. This means $\phi(n) = (p-1)(q-1)$

4. Find $e$ such that $0 < e < n$ and $\gcd(e, \phi(n)) = 1$

5. Find $d$ such that $(d*e) \mod \phi(n) = 1$
RSA encryption: example (part 1)

\[ p: \text{prime number} \quad \phi(n) = (p-1)(q-1) \]
\[ q: \text{prime number} \quad e: \quad 0 < e < n \text{ and } \gcd(e, \phi(n)) = 1 \]
\[ n = pq \quad d: \quad (d*e) \mod \phi(n) = 1 \]

\[ p = 3 \]
\[ q = 13 \]
\[ n = \]
\[ \phi(n) = \]
\[ e = \]
\[ d = \]
RSA algorithm: encryption, decryption

- You now have your
  - public key: $(e,n)$
  - private key: $(d,n)$

- If someone wants to send you a message (number) $m$, they:
  - compute and send: $\text{encrypt}(m) = m^e \mod n$

- When you get a message $z$, you:
  - compute and read: $\text{decrypt}(z) = z^d \mod n$
Figure 7.27 A schematic of the RSA cryptosystem, where $n = pq$ and $de \equiv_{(p-1)(q-1)} 1$, for prime numbers $p$ and $q$. 
RSA encryption: example (part 2)

p: prime number
q: prime number
n = pq

ϕ(n) = (p-1)(q-1)
e: 0 < e < n and gcd(e,ϕ(n)) = 1
d: (d*e) mod ϕ(n) = 1

p = 3
q = 13
n = 39
ϕ(n) = 24
e = 5
d = 29

What is the public key?
What is the private key?
What do you get if you encrypt 10?
RSA encryption: an example

\[ \begin{align*}
\text{p: prime number} & \quad \phi(n) = (p-1)(q-1) \\
\text{q: prime number} & \quad e: \ 0 < e < n \text{ and } \gcd(e, \phi(n)) = 1 \\
n = pq & \quad d: \ (d*e) \mod \phi(n) = 1
\end{align*} \]

\[ \begin{align*}
p & = 3 \\
q & = 13 \\
n & = 39 \\
\phi(n) & = 24 \\
e & = 5 \\
d & = 29
\end{align*} \]

What is the public key?
(5, 39)

What is the private key?
(29, 39)

What do you get if you encrypt 10?
\[ 10^5 \mod 39 = 4 \]
Why does the RSA algorithm work?

Figure 7.27 A schematic of the RSA cryptosystem, where \( n = pq \) and \( de \equiv (p-1)(q-1) \) 1, for prime numbers \( p \) and \( q \).
RSA: correctness

- Claim: $\text{decrypt(encrypt}(m)) = m$
- Proof:
  $\text{decrypt(encrypt}(m)) = ...$

\begin{itemize}
  \item $p$: prime number
  \item $q$: prime number
  \item $n = pq$
  \item $\phi(n) = (p-1)(q-1)$
  \item $e$: $\gcd(e,\phi(n)) = 1$
  \item $d$: $(d*e) \mod \phi(n) = 1$
  \item $\text{encrypt}(m) = m^e \mod n$
  \item $\text{decrypt}(z) = z^d \mod n$
\end{itemize}
RSA: correctness

- Claim: $\text{decrypt(encrypt(m))} = m$
- Proof:
  $\text{decrypt(encrypt(m))} = \text{decrypt(m}^e \mod n\text{)}$
  $= (m^e \mod n)^d \mod n$
  $= (m^e)^d \mod n$
  $= (m^{ed}) \mod n$
  $= (m^{k\phi(n)+1}) \mod n$
  $= (m^{k\phi(n)}) \mod n$
  $= (m \mod n) \times (m^{k\phi(n)} \mod n)$
  ... now what?

$p$: prime number
$q$: prime number
$n = pq$
$\phi(n) = (p-1)(q-1)$
e: $\gcd(e, \phi(n)) = 1$
d: $(d\times e) \mod \phi(n) = 1$
$\text{encrypt(m)} = m^e \mod n$
$\text{decrypt(z)} = z^d \mod n$
Fermat and Euler

- **Fermat's Little Theorem:**
  - If $p$ is prime and $\gcd(a, p) = 1$, then $a^{p-1} = 1 \mod p$
  - Equivalently, $a^p = a \mod p$

- **Euler:**
  - Euler's totient function: $\phi(n) = | \{ x : x < n \text{ and } \gcd(n, x) = 1 \} |$
  - What is $\phi(n)$ if $n$ is prime?
  - Theorem: If $\gcd(a, n) = 1$, then $a^{\phi(n)} = 1 \mod n$
RSA: correctness

- Claim: decrypt(encrypt(m)) = m
- Proof:

  decrypt(encrypt(m)) = decrypt(m^e \mod n)
  = (m^e \mod n)^d \mod n
  = (m^e)^d \mod n
  = (m^{ed}) \mod n
  = (m^{k\phi(n)+1}) \mod n
  = (mm^{k\phi(n)}) \mod n
  = (m \mod n) \times (m^{k\phi(n)} \mod n)

Euler: If \( \gcd(a,n) = 1 \), then \( a^{\phi(n)} \equiv 1 \mod n \)

\( p, q \): prime numbers
\( n = pq \)
\( \phi(n) = (p-1)(q-1) \)
\( e \): \( \gcd(e,\phi(n)) = 1 \)
\( d \): \( (d*e) \mod \phi(n) = 1 \)
\( encrypt(m) = m^e \mod n \)
\( decrypt(z) = z^d \mod n \)
RSA: correctness

- Claim: decrypt(encrypt(m)) = m
- Proof:

  decrypt(encrypt(m)) = decrypt(m^e mod n)
  = ((m^e mod n)^d mod n)
  = (m^e d mod n)
  = (m^{ed} mod n)
  = (m^{k\phi(n)+1} mod n)
  = (m^{k\phi(n)} mod n)
  = (m^{k\phi(n)\phi(n)+1} mod n)
  = (m^{k\phi(n)} \mod n)
  = (m \mod n) \times (m^{k\phi(n)} \mod n)
  = (m \mod n) \times ((m^{k\phi(n)})^\phi(n) \mod n)
  = (m \mod n), as long as gcd(m,n) = 1
  = m, as long as m < n

Euler: If gcd(a,n) = 1, then a^{\phi(n)} = 1 \mod n

p: prime number
q: prime number
n = pq
φ(n) = (p-1)(q-1)
e: gcd(e,φ(n)) = 1
d: (d*e) mod φ(n) = 1

encrypt(m) = m^e mod n
decrypt(z) = z^d mod n
RSA in practice

- What if the message isn't a number?
  - Everything is a number

- What if the message isn't a number less than n?
  - Divide it into chunks

- Would you ever flip? Encrypt with private key and decrypt with public key?
  - Digital signature
Why is RSA algorithm good?

Figure 7.27 A schematic of the RSA cryptosystem, where $n = pq$ and $de \equiv (p-1)(q-1) 1$, for prime numbers $p$ and $q$.

How secure is this?
Security of RSA

- **Given encrypt(m), can you figure out m?**
  - given $m^e \mod n$ can you figure out $m$?
  - issue is that many, many messages $m$ will map to the same encrypted value.

- **Given (e,n), can you figure out (d,n)?**
  - know: $(d\times e) \mod \phi(n) = 1$
  - but you don't know $\phi(n)$ and there isn't a good way to get it unless you can figure out $p$ and $q$ from $n$
  - how expensive is this?

---

$p$: prime number
$q$: prime number
$n = pq$

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

$e$: $\gcd(e,\phi(n)) = 1$

$d$: $(d\times e) \mod \phi(n) = 1$

encrypt($m$) = $m^e \mod n$

decrypt($z$) = $z^d \mod n$
>>> pow(2, 1024)
179769313486231590772930519078902473361797697
894230657273430081157732675805500963132708477
322407536021120113879871393357658789768814416
622492847430639474124377767893424865485276302
219601246094119453082952085005768838150682342
462881473913110540827237163350510684586298239
9472459384797163048353563296242242137216

https://crypto.stackexchange.com/questions/1978/how-big-an-rsa-key-is-considered-secure-today