Encryption

What is it and why do we need it?

Admin

Midterm next Thursday
- Covers everything from 9/24 – 10/27 + some minor SML
- Will not have to write any assembly
- 2 pages of notes
- Review sessions next week (TBA)

Assignment 6

4 more assignments:
- Assignment 7, due 11/13 5pm
- Assignment 8, due 11/20 5pm
- Assignments 9 & 10, due 12/9 11:59pm
Encryption

I like bananas

Encryption: a bad attempt

Encryption: the basic idea

encrypt message
send encrypted message
decrypt message
Encryption: a better approach

Encryption uses

Where have you seen encryption used?

Encryption uses

Private key encryption
Private key encryption

Any problems with this?

Private key encryption

Public key encryption

Private key encryption

Two keys, one you make publicly available and one you keep to yourself
Public key encryption

Share your public key with everyone

Public key encryption

I like bananas

send encrypted message

I like bananas

decrypt message

Public key encryption

Only the person with the private key can decrypt!

Modular arithmetic

Normal arithmetic:
\[ a = b \]
\[ a \text{ is equal to } b \text{ or } a-b = 0 \]

Modular arithmetic:
\[ a \equiv b \pmod{n} \]
\[ a-b = n^k \text{ for some integer } k \]
\[ a = b + n^k \text{ for some integer } k \]
\[ a \% n = b \% n \text{ (where } \% \text{ is the mod operator) } \]
Modular arithmetic

Which of these statements are true?

12 \equiv 5 \pmod{7}

52 \equiv 92 \pmod{10}

17 \equiv 12 \pmod{6}

65 \equiv 33 \pmod{32}

12-5 = 7 = 1*7

12 \% 7 = 5 = 5 \% 7

92-52 = 40 = 4*10

92 \% 10 = 2 = 32 \% 20

17-12 = 5

17 \% 6 = 5

12 \% 6 = 0

65-33 = 32 = 1*32

65 \% 32 = 1 = 33 \% 32

Modular arithmetic properties

If:

a \equiv b \pmod{n}

then:

a \mod{n} \equiv b \mod{n} \pmod{n}

More importantly:

(a + b) \mod{n} \equiv (a \mod{n}) + (b \mod{n} \pmod{n)

and

(a * b) \mod{n} \equiv (a \mod{n}) * (b \mod{n} \pmod{n)}

What do these say?
Modular arithmetic

Why talk about modular arithmetic and congruence? How is it useful? Why might it be better than normal arithmetic?

We can limit the size of the numbers we're dealing with to be at most \( n \) (if it gets larger than \( n \) at any point, we can always just take the result \( \% \ n \))

The mod operator can be thought of as mapping a number in the range \( 0 \ldots \text{number}-1 \)

GCD

What does GCD stand for?

Greatest Common Divisor

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

\[
gcd(25, 15) = ?
\]

\[
\begin{array}{c|c|c}
\hline
\text{Divisors:} & 25 & 15 \\
\hline
25 & 5 & 5 \\
5 & 3 & 1 \\
1 & 1 & 1 \\
\hline
\end{array}
\]

\[
gcd(25, 15) = 5
\]
Greatest Common Divisor

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

\[ \text{gcd}(100, 52) = ? \]

\[
\begin{array}{c|c}
100 & 52 \\
20 & 13 \\
10 & 4 \\
5 & 2 \\
2 & 1 \\
1 & 1 \\
\end{array}
\]

Divisors:

\[ \text{gcd}(100, 52) = 4 \]

\[ \text{gcd}(100, 9) = ? \]
\[ \text{gcd}(23, 5) = ? \]
\[ \text{gcd}(7, 56) = ? \]
\[ \text{gcd}(14, 63) = ? \]
\[ \text{gcd}(1, 17) = ? \]

Greatest Common Divisor

\[ 1 \]

\[ 7 \]

\[ 1 \]

\[ 1 \]

Any observations?
Greatest Common Divisor

When the $\gcd = 1$, the two numbers share no factors/divisors in common

If $\gcd(a, b) = 1$ then $a$ is relatively prime to $b$

This a weaker condition than primality, since any two prime numbers are also relatively prime, but not vice versa

A useful property:

If two numbers are relatively prime (i.e. $\gcd(a, b) = 1$), then there exists a $c$ such that

$$a^c \mod b = 1$$

RSA public key encryption

Have you heard of it?

What does it stand for?

RSA is one of the most popular public key encryption algorithms in use

RSA = Ron Rivest, Adi Shamir and Leonard Adleman
RSA public key encryption

1. Choose a bit-length \( k \)
   Security increase with the value of \( k \), though so does computation

2. Choose two primes \( p \) and \( q \) which can be represented
   with at most \( k \) bits

3. Let \( n = pq \) and \( \phi(n) = (p-1)(q-1) \)
   \( \phi \) is called Euler's totient function

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

5. Find \( e \) such that \( ed \mod \phi(n) = 1 \)
   Remember, we know one exists!

Given this setup, you can prove that given a number \( m \):

\[
(m^e)^d = m^{ed} = m \quad \text{(mod } n)\]

What does this do for us, though?

RSA encryption/decryption

<table>
<thead>
<tr>
<th>Private key</th>
<th>Public key</th>
</tr>
</thead>
<tbody>
<tr>
<td>( (d, n) )</td>
<td>( (e, n) )</td>
</tr>
</tbody>
</table>

You have a number \( m \) that you want to send encrypted

\[
\text{crypto}(m) = m^e \mod n \quad \text{(uses the public key)}
\]

How does this encrypt the message?
RSA encryption/decryption

<table>
<thead>
<tr>
<th>private key</th>
<th>public key</th>
</tr>
</thead>
<tbody>
<tr>
<td>(d, n)</td>
<td>(e, n)</td>
</tr>
</tbody>
</table>

You have a number $m$ that you want to send encrypted

$$\text{encrypt}(m) = m^e \mod n$$  \hspace{1cm} (uses the public key)

- Maps $m$ onto some number in the range 0 to $n-1$
- If you vary $e$, it will map to a different number
- Therefore, unless you know $d$, it's hard to know original $m$ was after the transformation

Does this work?

$$\text{decrypt}(z) = z^d \mod n$$  \hspace{1cm} (uses the private key)

$z$ is some encrypted message

$$\text{decrypt}(z) = \text{decrypt}(m^e \mod n) = (m^e \mod n)^d \mod n$$  \hspace{1cm} definition of decrypt

$$= (m^e)^d \mod n$$  \hspace{1cm} modular arithmetic

$$= m \mod n$$  \hspace{1cm} \text{if } 0 \leq m < n, \text{ yes!}$

Did we get the original message?
### RSA encryption: an example

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$: prime number</td>
<td>$q(n) = (p-1)(q-1)$</td>
</tr>
<tr>
<td>$q$: prime number</td>
<td>$d$: $0 &lt; d &lt; n$ and $\gcd(d,q(n)) = 1$</td>
</tr>
<tr>
<td>$n = pq$</td>
<td>$e$: $de \mod q(n) = 1$</td>
</tr>
</tbody>
</table>

- $p = 3$
- $q = 13$
- $n = 39$
RSA encryption: an example

\[
p, q: \text{prime number} \\
\phi(n) = (p-1)(q-1) \\
n = pq \\
de \mod \phi(n) = 1 \\
p = 3 \\
q = 13 \\
n = 39 \\
\phi(n) = 2 \times 12 = 24
\]

RSA encryption: an example

\[
p, q: \text{prime number} \\
\phi(n) = (p-1)(q-1) \\
n = pq \\
de \mod \phi(n) = 1 \\
p = 3 \\
q = 13 \\
n = 39 \\
\phi(n) = 24 \\
d = ? \\
e = ?
\]

RSA encryption: an example

\[
p, q: \text{prime number} \\
\phi(n) = (p-1)(q-1) \\
n = pq \\
de \mod \phi(n) = 1 \\
p = 3 \\
q = 13 \\
n = 39 \\
\phi(n) = 24 \\
d = 5 \\
e = 5
\]

RSA encryption: an example

\[
n = 39 \\
d = 5 \\
e = 5 \\
\text{encrypt}(m) = m^e \mod n \\
\text{decrypt}(z) = z^d \mod n
\]

\[
\text{encrypt}(10) = ?
\]
### RSA encryption: an example

<table>
<thead>
<tr>
<th>( n = 39 )</th>
<th>encrypt(m) = ( m^e \mod n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d = 5 )</td>
<td>decrypt(z) = ( z^d \mod n )</td>
</tr>
</tbody>
</table>

encrypt(10) = \( 10^5 \mod 39 = 4 \)

decrypt(4) = \( ? \)

\[ \text{encrypt}(2) = ? \]
RSA encryption: an example

\[
\begin{align*}
n &= 39 & \text{encrypt}(m) &= m^e \mod n \\
d &= 5 & \text{decrypt}(z) &= z^d \mod n \\
e &= 5 & \\
\text{encrypt}(2) &= 2^e \mod 39 = 32 \mod 39 = 32 \\
\text{decrypt}(32) &= ?
\end{align*}
\]

RSA encryption in practice

For RSA to work: \(0 \leq m < n\)

- What if our message isn’t a number?
- What if our message is a number that’s larger than \(n\)?

We can always convert the message into a number (remember everything is stored in binary already somewhere!)

What if our message is a number that’s larger than \(n\)?

Break it into \(m\) sized chunks and encrypt/decrypt those chunks
### RSA encryption in practice

**encrypt(“I like bananas”) =**

- `0101100101011100 ...` encode as a binary string (i.e. number)
- `4, 15, 6, 2, 22, ...` break into multiple < n size numbers
- `17, 1, 43, 15, 12, ...` encrypt each number

**decrypt((17, 1, 43, 15, 12, …)) =**

- `0101100101011100 ...` decrypt each number
- `4, 15, 6, 2, 22, ...` put back together
- “I like bananas” turn back into a string (or whatever the original message was)

Often encrypt and decrypt just assume sequences of bits and the interpretation is done outside.