Outline

1. Cryptosystems
2. One-Time Pad
3. Public key Cryptography
4. FLT + RSA
5. Digital Signatures
6. Attacks

Reminder

Midterm on Monday July 12th, 8pm PDT

More announcements in the upcoming week.
Alice wants to send a message (bitstring) to Bob.

She encrypts it and sends it as $E(m)$.

Eve can see $E(m)$.

Bob uses decryption function $D$ to recover $m$.

Note: $E$ and $D$ often depend on some key $k$. $E_k$ and $D_k$.

Goal: Make sure Eve cannot recover $m$, but Bob can.
One-Time Pad

**XOR**: Exclusive OR, denoted \( \oplus \)

<table>
<thead>
<tr>
<th></th>
<th>( X )</th>
<th>( Y )</th>
<th>( X \oplus Y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\( k \) is a bit string which is as long as \( m \)

Choose \( E_k(m) = m \oplus k \)

\( D_k(m) = m \oplus k \)

\[ D_k(E_k(m)) = ((m \oplus k) \oplus k) = m \oplus (k \oplus k) = m \]

Only Alice and Bob can know \( k \).

**Pro:**
It works if Eve does not know \( k \)

**Cons:**
Cannot reuse the pad \( k \)

Alice and Bob need to decide on \( k \) beforehand.
Why can’t we reuse?

\[ E(m_1) = m_1 \oplus k \]
\[ E(m_2) = m_2 \oplus k \]

Eve can see these.

Eve can compute:

\[
(m_1 \oplus k) \oplus (k \oplus m_2)
\]

\[
= m_1 \oplus m_2
\]

Info leaked for \( m_1, m_2 \).
Public key Cryptography

Can send messages securely without having to meet privately first!
Fermat's Little Theorem (FLT)

For any prime $p$ and any $a \in \mathbb{Z} \setminus \{0\}, \ldots, p-1$, we have $a^{p-1} \equiv 1 \pmod{p}$

**Proof**

Observing that $f(x) = ax \pmod{p}$ is a bijection from $S$ to $S$

This is because $\gcd(a, p) = 1$, so

$$a^i = a^j \pmod{p} \implies i \equiv j \pmod{p}$$

So, $f$ maps each element of $S$ to a distinct value in $S$

$$\implies \prod_{i \in S} i \equiv \prod_{i \in S} a \cdot i \pmod{p}$$

$$\Rightarrow (p-1)! \equiv a^{p-1} (p-1)! \pmod{p}$$

$$1 \equiv a^{p-1} \pmod{p}$$
RSA \textcircled{Rivest, Shamir, Adleman}

Start of with two primes \( p \) and \( q \).

\textbf{Public key (Treasure Box)}

\((N, e)\)

\(N := pq\)

\(e\) is a number such that \(\text{gcd}(e, (p-1)(q-1)) = 1\)

\textbf{Private key}

\(d := e^{-1} \pmod{(p-1)(q-1)}\)

\(E(x) = x^e \pmod{N}\)

\(D(y) = y^d \pmod{N}\)

Correctness: \(D(E(x)) = x\) ?

\((xe)^d \equiv x \pmod{N}\)

\(x^{ed} - x \equiv 0 \pmod{N}\)

Note: \(ed \equiv 1 \pmod{(p-1)(q-1)} \Rightarrow ed = 1 + k(p-1)(q-1)\)
\[
x - x \equiv 0 \pmod{N}
\]

\[
x \left( x^{k(p-1)(q-1)} - 1 \right) \equiv 0 \pmod{N}
\]

**Approach:** show divisibility by \( p \) and by \( q \) separately.

**Case 1:** \( x \) is divisible by \( p \)

**Case 2:**

\[
x \left( x^{k(p-1)(q-1)} - 1 \right) \pmod{p}
\]

\[
\equiv x \left( \left( x^{(p-1)(q-1)} \right)^k - 1 \right) \pmod{p}
\]

\[
\uparrow \text{FLT}
\]

\[
\equiv x \left( 1 - 1 \right) \pmod{p}
\]

\[
\equiv 0 \pmod{p}
\]

Similarly, the expression is also divisible by \( q \).

So, it is divisible by \( N = pq \)

\[
\Rightarrow x \left( x^{k(p-1)(q-1)} - 1 \right) \equiv 0 \pmod{N}
\]
Why does RSA work?

1. Assumes $N$ is too large to brute force $x^e$ for each $x$ and check if the encoded message matches.

2. Assumes $d$ can’t be computed without extracting $p$ and $q$ from $N$ (factoring $N$ is hard).
RSA Example (from Notes)

\[ p = 5 \]
\[ q = 11 \]
\[ N = 5 \cdot 11 = 55 \]

Say \( e = 3 \), \( \gcd(e, 40) = 1 \)

Bob:

Public key: \((N, e) = (55, 3)\)

Private key: \(3^{-1} \mod 40\)

\[ 40 = 3 \cdot 13 + 1 \]

\[ 40 \cdot 1 - 13 \cdot 3 = 1 \]

\[ d = -13 = 27 \mod 40 \]

Alice can then send \( x \) as \( E(x) = x^3 \mod 55 \)

Bob will decrypt this as \( D(y) = y^{27} \mod 55 \)

Example:

\[ x = 13 \]
\[ E(x) = 13^3 \mod 55 = 82 \]

\[ D(82) = 82^{27} \equiv 13 \mod 55 \]
Digital Signatures

Trusted

You

Certificate Authority (CA) Amazon.com

(N, e)

1. \( m = \text{"This is Amazon"} \)

2. Signed by CA: \( CA = \frac{m^d}{s} \)

3. You can check using \((N, e)\):

\[
se = m^d e \equiv m \pmod{N}
\]

Checks out, CA did confirm/sign.
RSA Attack

Replay Attack Example

I send $E(m)$ to Amazon to make purchase.

Eve reads $E(m)$, and sends it to Amazon again.

Now I got charged twice :(

Solution

Send $E(\text{concatenate}(m, s))$ where

$s$ is a random string.

If Amazon gets the same message twice, it will just reject the second one.
RSA Sampling Primes

Prime Number Theorem states that

\[ \# \text{ of primes} \leq N \text{ is at least } \frac{N}{\ln(N)} \]

Go through all numbers less than \( N \) and check if they are prime.

There exists an efficient algorithm that tests if \( N \) is prime

(polynomial time in the number of bits)

Note: Want \( p \) and \( q \) to be very large \( \rightarrow 512 \text{ bits each} \).
\[ x \equiv 20 \pmod{30} \]
\[ x = 20 + 30k \]

\[ ed \equiv 1 \pmod{(p-1)(q-1)} \]

\[ \Rightarrow ed = 1 + k(p-1)(q-1) \]