A CRYPTO NERD’S IMAGINATION:

His laptop’s encrypted. Let’s build a million-dollar cluster to crack it.

No good! It’s 4096-bit RSA!

Blast! Our evil plan is foiled!

WHAT WOULD ACTUALLY HAPPEN:

His laptop’s encrypted. Drug him and hit him with this $5 wrench until he tells us the password.

Got it.

XKCD: http://xkcd.com/538/
Cryptography

• Cryptography: An ancient art
  • 500BC – 20\textsuperscript{th} century: Design -> break -> repair -> break -> repair ->.....

• Modern Cryptography: Cryptography as a \textit{science}
  • relies on rigorous threat models
  • firm theoretical foundations and proofs!
Design, analysis and implementation of mathematical techniques for securing information, systems and computation against adversarial attacks.
Modern Cryptography: How many of the following actions involve cryptography?

1. Git cloning your lab repo
2. Connecting to Swarthmore’s WiFi
3. Updating software on your device
4. Making online purchases

A. One of these
B. Two of these
C. Three of these
D. Four of these
If you don’t understand what you want to achieve, how can you possibly know when (or if) you have achieved it?
Modern Cryptography

Importance of clear assumptions:
• allows researchers to validate assumptions
• comparison between schemes based on different assumptions
• re-adjust for weaknesses in assumptions

*any new cryptographic construction should be proven secure with respect to a specific definition, and a set of clearly stated assumptions*
Where Does the Attacker Live?

Client
- Browser: renders the webpage
- Private data
- Malware attacker

Server
- Web Server hosts the web page
- Database
- Network attacker
- Web server attacker
Scenarios and Goals

Alice

Public network

Bob

Disk
Scenarios and Goals

Confidentiality
Keep others from reading Alice’s messages/data

Integrity
Keep others from undetectably tampering with Alice’s messages/data

Authenticity
Keep others from undetectably impersonating Alice (keep her to her word too!)
Recall the Bigger Picture

• Cryptography: small piece of a larger system
• Protect the entire system (recall: the weakest link)
  • physical security
  • OS security
  • Network security
  • Users
  • Cryptography
• Cryptography is a crucial part of this toolbox
Cryptography: Terms

Encryption (E): The process of transforming a message so that its meaning is not obvious
Decryption (D): The process of transforming an encrypted message back into its original form.
Plaintext (P): Original, unencrypted form of a message
Ciphertext (C): The encrypted form of a message

Formal Notation: We seek a cryptosystem for which P = D (E (P))
Historical Ciphers

• Substitution Cipher
  • Monoalphabetic - Ceasar’s Cipher – fixed subst. over the entire message
  • Polyalphabetic – a number of substitutions at different positions in the message

• Transposition Ciphers

• Codebooks

• Machines

Recommended Reading: The Codebreakers by David Kahn, The Code Book by Simon Singh
Ceasar Cipher: Substitution Cipher

Plaintext letters replaced with letters fixed shift way in the alphabet.

Example:

• Plaintext: HEY BRUTUS BRING A KNIFE TO THE PARTY.
• Ciphertext: KHB EUXWXV EULQJ D NQLIH WR WKH SDUWB
• Key Shift 3:
  • ABCDEFGHIJKLMNOPQRSTUVWXYZ
  • DEFGHIJKLMNOPQRSTUVWXYZABC
Ceasar Cipher: Substitution Cipher

Plaintext letters replaced with letters fixed shift way in the alphabet.

Example:

- Plaintext: HEY BRUTUS BRING A KNIFE TO THE PARTY.
- Ciphertext: KHB EUXWXV EULQJ D NQLIH WR WKH SDUWB
- Key Shift 3:
  - ABCDEFGHIJKLMNOPQRSTUVWXYZ
  - DEFHIJKLMNOPQRSTUVWXYZABC

- Encryption and Decryption are symmetric.
- Key space?
  - 26
- Attack shift ciphers?
  - brute force
Substitution Cipher

• **Superset of shift ciphers:** each letter is substituted for another one.
• One implementation: Add a secret key
• Example
  • Plaintext: ABCDEFGHIJKLMNOPQRSTUVWXYZ
  • Cipher: ZEBRASCDFGHIJKLMNOPQTUVWXY
• “state-of-the-art” for thousands of years
Monoalphabetic Substitution Cipher

• What is the key space?
  • $26! \approx 2^{88}$

• Launching an attack?
  • frequency analysis: the study of frequency of letters or groups of letters (grams).
  • Common letters: T, A, E, I, O
  • Common 2-letter combinations (bi-grams): TH, HE, IN, ER
  • Common 3-letter combinations (tri-grams): THE, AND, ING.
Cryptanalysis of Monoalphabetic Substitution

• Dominates cryptography through the first millennium

• Frequency analysis (Al-Kindi from 800 AD)

• Lessons?
  • Use large blocks: instead of replacing ~6 bits at a time, replace 64 or 128 bits
    • Leads to block ciphers like DES and AES
  • Use different substitutions to prevent frequency analysis
    • Leads to polyalphabetic substitution ciphers and stream ciphers
Vigenère Cipher (1596)

- Main weakness of monoalphabetic substitution ciphers:
  - Each letter in the ciphertext corresponds to only one letter in the plaintext

- Polyalphabetic substitution cipher
  - Given a key $K = (k_1, k_2, \ldots, k_m)$
  - Shift each letter $p$ in the plaintext by $k_i$, where $i$ is modulo $m$

| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Plaintext  CRYPTOGRAPHY
Key        LUCK LUCKLUCK  (Shift 11 20 2 10 11 20 2 11 ...)
Ciphertext NLAZEIIBLJJI
Kasisk Test

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>THE SUN AND THEM A N IN THE MOON</th>
</tr>
</thead>
<tbody>
<tr>
<td>Key</td>
<td>KING KING KING KING KING KING KING</td>
</tr>
<tr>
<td>Ciphertext</td>
<td>D P R Y E V N T N BUK WIAOX BUK WWB T</td>
</tr>
</tbody>
</table>

Distance = 8

- Repeating patterns (of length >2) in ciphertext are a tell
  - Likely due to repeated plaintext encrypted under repeated key characters
  - The distance is likely to be a multiple of the key length
Cryptanalysis of Vigenère Cipher

• Cracking Vigenère (1854 or 1863)
  1. Guess the key length \( x \) using Kasiski test of index of coincidence
  2. Divide the ciphertext into \( x \) shift cipher encryptions
  3. Use frequency analysis on each shift cipher

• Lessons?
  • As key length increases, letter frequency becomes more random
  • If key never repeated, Vigenère wouldn’t be breakable!
- WW2 German Enigma machine
  - Polyalphabetic substitution cipher
    - Substitution table changes from character to character
    - Rotors control substitutions

- Allies broke Enigma (even before the war), significant intelligence impact

- Computers were built to break WW2 ciphers, by Alan Turing and others
Enigma Machine

- Use rotors that change position after each key
- Key: initial setting of the rotors
- Key space?
  - $26^n$ for $n$ rotors
- KeyGen:
  - Choose rotors, rotor orders, rotor positions, and plugboard settings
  - 158,962,555,217,826,360,000 possible keys!
Cryptanalysis: Enigma

- Polish and British cryptographers built BOMBE, a machine to brute-force Enigma keys

- Why was Enigma breakable?
  - **Kerckhoff’s principle**: The Allies stole Enigma machines, so they knew the algorithm
  - **Known plaintext attacks**: the Germans often sent predictable messages (e.g. the weather report every morning)
  - **Chosen plaintext attacks**: the Allies could trick the Germans into sending a message (e.g. “soldiers at Normandy”)
  - **Brute-force**: BOMBE would try many keys until the correct one was found
Legacy of Enigma

• Alan Turing, one of the cryptographers who broke Enigma, would go on to become one of the founding fathers of computer science

• Most experts agree that the Allies breaking Enigma shortened the war in Europe by about a year

Alan Turing
Kerckhoffs’ principle

Encryption and Decryption and Key Generation Algorithm are publicly known. *The only unknown is the shared secret key*
One Time Pad (1920s)

- Fix vulnerability in Vigenère cipher: use very long keys
- Key is a random string: at least as long as the plaintext
- Plaintext: Message that is $n$ bits long
- Key: $\{0, 1\}^n$ sequence of $n$ bits chosen uniformly at random.
Review: XOR

The XOR operator takes two bits and outputs one bit:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0 ⊕ 0 = 0</td>
<td>0 ⊕ 1 = 1</td>
<td>1 ⊕ 0 = 1</td>
</tr>
</tbody>
</table>

Useful properties of XOR:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x ⊕ 0 = x</td>
<td>x ⊕ x = 0</td>
<td>x ⊕ y = y ⊕ x</td>
</tr>
<tr>
<td>(x ⊕ y) ⊕ z = x ⊕ (y ⊕ z)</td>
<td>(x ⊕ y) ⊕ x = y</td>
<td></td>
</tr>
</tbody>
</table>
Review: XOR Algebra

Algebra works on XOR too

<table>
<thead>
<tr>
<th>( y \oplus 1 = 0 )</th>
<th>Goal: Solve for ( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y \oplus 1 \oplus 1 = 0 \oplus 1 )</td>
<td>XOR both sides by 1</td>
</tr>
<tr>
<td>( y = 1 )</td>
<td>Simplify with identities</td>
</tr>
</tbody>
</table>
One-Time Pads: Key Generation

The key $K$ is a randomly-chosen bitstring.

Recall: We are in the symmetric-key setting, so we’ll assume Alice and Bob both know this key.
One-Time Pads: Encryption

The plaintext $M$ is the bitstring that Alice wants to encrypt.

<table>
<thead>
<tr>
<th>Alice</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K$</td>
</tr>
<tr>
<td>$M$</td>
</tr>
</tbody>
</table>

Idea: Use XOR to scramble up $M$ with the bits of $K$. 

## One-Time Pads: Encryption

### Encryption algorithm

Encryption algorithm: XOR each bit of $K$ with the matching bit in $M$.

The ciphertext $C$ is the encrypted bitstring that Alice sends to Bob over the insecure channel.
One-Time Pads: Decryption

Bob

<table>
<thead>
<tr>
<th>K</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>0</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>0</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Bob receives the ciphertext $C$. Bob knows the key $K$. How does Bob recover $M$?
**One-Time Pads: Decryption**

<table>
<thead>
<tr>
<th>Bob</th>
<th>( K )</th>
<th>( C )</th>
<th>( M )</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1 1 0 0 1 0 1 0 1 1 1</td>
<td>1 1 1 1 1 1 1 0 0 0 0 1 1</td>
<td>1 0 0 1 1 0 0 1 0 1 0 0</td>
</tr>
</tbody>
</table>

Decryption algorithm: XOR each bit of \( K \) with the matching bit in \( C \).
Cryptography by Computers

• The modern era of cryptography started after WWII, with the work of Claude Shannon

• “New Directions in Cryptography” (1976) showed how number theory can be used in cryptography
  • Its authors, Whitfield Diffie and Martin Hellman, won the Turing Award in 2015 for this paper

• This is the era of cryptography we’ll be focusing on.
How Cryptosystems work today

• Layered approach:
  • Cryptographic protocols (e.g., CBC mode encryption)
  • Built on: Cryptographic primitives (block ciphers)

• Flavors of cryptography:
  • Symmetric: private key
  • Asymmetric: public key

• Public algorithms: Kerckhoff’s principle

• Security proofs based on assumptions (not this course)

• Warning! Here be dragons!
  • careful about inventing your own
  • Use vetted libraries to apply crypto algorithms
Cryptosystem Stack

• Primitives:
  • AES/DES
  • ESA / ElGamal / Elliptic Curve

• Modes
  • Block mode (CBC, ECV, CTR, GCM..)
  • Padding structures

• Protocols:
  • TLS, SSL, SSH

• Usage of protocols:
  • Browser security
  • Secure remote logins
Flavors of Cryptography

• Symmetric cryptography
  • Both communicating parties have access to a shared random string $K$, called the key.

• Asymmetric cryptography
  • Each party creates a public key $p_k$ and a secret key $s_k$
  • Inventors won Turing Award!
Kerckhoff’s Principle

• Security of a cryptographic object should depend only on the secrecy of the secret (private) key.
• Security should not depend on the secrecy of the algorithm itself.
• Foreshadow: Need for randomness – the key to keep private