Agenda

1. Crypto Usage & Goals

2. Classical Crypto

3. One-time Pad & Perfect Secrecy

4. Modern Crypto
Crypto Use Is Ubiquitous

HTTPS

Bank of America

Facebook

Secure messaging

WhatsApp

Bitcoin

Tor
A Classical Crypto Goal: Privacy

Transfer $5 to account 12345

Privacy: Adversary can’t learn anything from the content that it eavesdrops.
A Classical Crypto Goal: Privacy

Private-key setting: $K_e = K_d$

Public-key setting: $K_e \neq K_d$

secret

Public
secret
But Privacy Alone Is **Not** Enough

Transfer $5 to account 12345

Transfer $1000 to account 99999

**Authenticity:** Adversary can’t forge valid ciphertexts
Four Fundamental Cryptographic Problems

- **Privacy**
  - Private-key Enc
  - Public-key Enc

- **Authenticity**
  - MAC
  - Digital Signature
1. Crypto Usage & Goals

2. Classical Crypto

3. One-time Pad & Perfect Secrecy

4. Modern Crypto
Caesar Cipher

No key
Broken once scheme is known
Caesar Cipher In The Wild

Mafia boss undone by clumsy crypto

Little Caesar

By John Leyden

Clues left in the clumsily encrypted notes of a Mafia don have helped Italian investigators to track his associates and ultimately contributed to his capture after years on the run.

BA jihadist relied on Jesus-era encryption

By Team Register

An IT worker from British Airways jailed for 30 years for terrorism offences used encryption techniques that pre-date the birth of Jesus.
Shift Cipher

Use a secret key $K \in \{0, \ldots, 25\}$

Same as Caesar cipher, but shift $K$ positions, instead of 3.
Substitution Cipher

Key: a permutation $\pi : \Sigma \rightarrow \Sigma$

Example: $\Sigma = \{A, B, C, \ldots, Z\}$
26! \approx 2^{88}

keys for the English alphabet
Sherlock Holmes’s Example

The Adventure of the Dancing Men
Problem: DM code is not a substitution cipher, as its output is not in English alphabet.
Exercise

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>I</td>
<td>J</td>
<td>K</td>
<td>L</td>
<td>M</td>
<td>N</td>
</tr>
<tr>
<td>O</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
</tr>
<tr>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Given a machine $\Pi$ to break substitution cipher, find a way to use $\Pi$ to break DM code
### Break Substitution Cipher: Frequency Analysis

#### Input text letter frequencies:

<table>
<thead>
<tr>
<th>Letter</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>11.63%</td>
</tr>
<tr>
<td>X</td>
<td>11.63%</td>
</tr>
<tr>
<td>I</td>
<td>9.30%</td>
</tr>
<tr>
<td>Y</td>
<td>9.30%</td>
</tr>
<tr>
<td>L</td>
<td>6.98%</td>
</tr>
<tr>
<td>M</td>
<td>6.98%</td>
</tr>
<tr>
<td>R</td>
<td>6.98%</td>
</tr>
<tr>
<td>S</td>
<td>6.98%</td>
</tr>
<tr>
<td>Z</td>
<td>6.98%</td>
</tr>
<tr>
<td>D</td>
<td>4.65%</td>
</tr>
<tr>
<td>T</td>
<td>4.65%</td>
</tr>
<tr>
<td>U</td>
<td>4.65%</td>
</tr>
<tr>
<td>A</td>
<td>2.33%</td>
</tr>
<tr>
<td>E</td>
<td>2.33%</td>
</tr>
<tr>
<td>N</td>
<td>2.33%</td>
</tr>
<tr>
<td>V</td>
<td>2.33%</td>
</tr>
<tr>
<td>B</td>
<td>2.33%</td>
</tr>
<tr>
<td>C</td>
<td>2.33%</td>
</tr>
<tr>
<td>G</td>
<td>2.33%</td>
</tr>
<tr>
<td>P</td>
<td>2.33%</td>
</tr>
<tr>
<td>H</td>
<td>2.33%</td>
</tr>
</tbody>
</table>

#### English text letter frequencies:

<table>
<thead>
<tr>
<th>Letter</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>E</td>
<td>12.72%</td>
</tr>
<tr>
<td>T</td>
<td>9.06%</td>
</tr>
<tr>
<td>A</td>
<td>8.17%</td>
</tr>
<tr>
<td>O</td>
<td>7.51%</td>
</tr>
<tr>
<td>I</td>
<td>6.97%</td>
</tr>
<tr>
<td>N</td>
<td>6.75%</td>
</tr>
<tr>
<td>S</td>
<td>6.33%</td>
</tr>
<tr>
<td>H</td>
<td>6.10%</td>
</tr>
<tr>
<td>R</td>
<td>5.99%</td>
</tr>
<tr>
<td>D</td>
<td>4.26%</td>
</tr>
<tr>
<td>L</td>
<td>4.03%</td>
</tr>
<tr>
<td>C</td>
<td>2.78%</td>
</tr>
<tr>
<td>U</td>
<td>2.75%</td>
</tr>
<tr>
<td>M</td>
<td>2.40%</td>
</tr>
<tr>
<td>W</td>
<td>2.36%</td>
</tr>
<tr>
<td>F</td>
<td>2.22%</td>
</tr>
<tr>
<td>G</td>
<td>2.01%</td>
</tr>
<tr>
<td>Y</td>
<td>1.97%</td>
</tr>
<tr>
<td>P</td>
<td>1.92%</td>
</tr>
<tr>
<td>B</td>
<td>1.49%</td>
</tr>
<tr>
<td>V</td>
<td>0.97%</td>
</tr>
<tr>
<td>K</td>
<td>0.77%</td>
</tr>
<tr>
<td>J</td>
<td>0.15%</td>
</tr>
<tr>
<td>X</td>
<td>0.15%</td>
</tr>
<tr>
<td>Q</td>
<td>0.09%</td>
</tr>
<tr>
<td>W</td>
<td>0.07%</td>
</tr>
</tbody>
</table>

Only work for **large** ciphertexts.
Real-world Issues: Many-to-one Mapping
to bat-rb. con todo mi respeto. I was sitting down playing chess with danny de emf and boxer de el centro was sitting next to us. Boxer was making loud and loud voices so I tell him por favor can you kick back homie

A decrypted jail letter
A Better Attack via Monte Carlo Markov Chain

View keys as nodes in a complex graph

Initial key

True key
Our Goal Is To Find A Path To The Destination

Problem: Can’t store the graph in memory
Claim: By properly defining how nodes are connected, the distance will be short.
Define Connectivity: Swap Two Entries

A node

A \rightarrow B \rightarrow H \rightarrow \ldots \rightarrow S

A neighbor

H \rightarrow B \rightarrow A \rightarrow \ldots \rightarrow S
Quiz: Find The Distance Between These Nodes
How Far Is The Distance For This Huge Graph?
25 steps to go from source to destination
How To Find A Way When Can’t Store Graph?
Estimate scores for current node and a random neighbor

Higher score → Closer to destination → Move
Graph Traversing: Hill Climbing

- Look for a random neighbor
- Move if higher score

Initial key

True key

0.03

0.07

0.1

0.3

0.5

0.8
Implementing Hill Climbing
Get Around Local Maximum in Hill Climbing

Occasionally, one may have to go **down**
Implementing Hill Climbing

To Go or Not To Go

Always go

Go with probability \( \frac{0.4}{0.5 + 0.4} \)
Estimating Score

Candidate key $K$

Ciphertext

Decrypted message

Need to tell how decrypted message looks like an English text
Estimating Score: Overview

Training phase

Testing phase

Tentative decrypted message

Frequency histogram

Score

Frequency histogram of all bigrams
Training Phase: Bigram Analysis

“the cat is chasing the mouse”

Bigrams: th he e □ c ca at t □ ...

must consider white space
Frequency of Bigrams in English

27 $\times$ 27 iterations

For each bigram $x$ do:

A large English text, say a novel

Frequency that $x$ appears in the text

Store the frequencies in an array $F$
An Example

“ca” appears 1235 times out of total 437907 bigrams

\[ F[“ca”] = \frac{1235}{437907} : \text{frequency of bigram “ca” in English} \]

Some bigrams will have frequency 0, as they don’t appear in English
Testing Phase: Assess Englishness

Candidate key $K$

Ciphertext

Decrypted message $M$

\[ \text{Score}[K] = \prod F[x] \]

bigram $x$ in $M$ (repetition included)
An Example

Decrypted message = “ale lens”

Bigrams: al le e□ l le en ns

Score = $F[al] \cdot F[le] \cdot F[e□] \cdot F[l] \cdot F[le] \cdot F[en] \cdot F[ns]$
Programming Issues: When To Stop?

**Loop forever**, but keep track of the best node so far

User decide when to terminate, typically after 5000 steps
Programming Issues: Numerical Errors

\[
\text{Score}[K] = \prod_{\text{bigram } x \text{ in } M} F[x]
\]

Example: 1000 bigrams, and each of frequency \( \sim 0.01 \)

\[
\text{Score} \approx 2^{-6643}
\]

Impossible to represent score via floating point
Maneuvering Scores: Keep Track of Score-Log

\[
\log(\text{Score}[K]) = \sum \log(F[x])
\]

**bigram** \(x\) **in** \(M\)

**Example**: 1000 bigrams, and each of frequency \(\sim 0.01\)

\[
\text{Score-log} \approx -6643
\]

Represent score-log with reasonable accuracy
Maneuvering Scores: Dealing with Zero

\[ F[r] = 0 \quad \rightarrow \quad \log(F[r]) = ? \]

Use a \textbf{negative} constant of large magnitude
Maneuvering Scores: Graph Traversing

Score-log = \( y > x \)

Go with prob = \( \frac{2^z}{2y + 2^z} = \frac{1}{2^{y-z} + 1} \)

Always go

Score-log = \( z \leq y \)

Score-log = \( x \)
The Enigma
Broken by British
in an effort led by Turing
1. Crypto Usage & Goals

2. Classical Crypto

3. One-time Pad & Perfect Secrecy

4. Modern Crypto
Encryption Syntax

Key Gen

$K \xrightarrow{$} K$

Encrypt

$M \xrightarrow{E} C$

Decrypt

$C \xrightarrow{D} M$

or $\perp$
Define security?
Perfect Secrecy

**Intuition:** Ciphertext should reveal *no additional info* about plaintext

For every $m$ and $c$:

$$\Pr_{K \leftarrow \mathcal{K}}[\text{Msg} = m \mid \mathcal{E}_K(\text{Msg}) = c] = \Pr[\text{Msg} = m]$$

**Common case:** Ciphertext is uniformly random, independent of msg
An Example

\[ \text{Msg} = \ast o \ast \]

\{cat, dog, hot\}

Ctx \quad \text{cracked} \quad \rightarrow \quad \text{Msg} = \ast o \ast \\

Not perfectly secure

\[
\Pr[\text{Msg} = \text{dog} | \text{Ctx}] = \frac{1}{2} \neq \Pr[\text{Msg} = \text{dog}] = \frac{1}{3}
\]
Substitution Cipher Is Not Perfectly Secret

\[
\begin{align*}
\text{Msg} & \quad \{\text{hot day, bad boy}\} \\
\text{Ctx} & \quad \text{“tic tok”}
\end{align*}
\]

\[
\Pr[\text{Msg} = \text{bad boy} \mid \text{Ctx}] = 1 \neq \Pr[\text{Msg} = \text{bad boy}] = 1/2
\]
An Application of Perfect Secrecy
Bob’s Privacy in the 5-card Trick for The Dating Problem

Assume that Alice disagrees

Want: prove 5-card trick reveals nothing about Bob’s input

Otherwise, Bob’s privacy is moot
5-card Trick As Encryption

Initial configuration = Bob’s message

Assume nothing about message distribution
Cutting as Cyclic Shift

Alice’s cut

Alice’s cyclic shift by $a \in \{0, 1, 2, 3, 4\}$ positions

known by Alice
Cutting as Cyclic Shift

Bob’s cut

Bob’s cyclic shift by $b \leftrightarrow \{0, 1, 2, 3, 4\}$ positions

uniformly random in Alice’s view
Composition of Cuts

Cyclic shift by \((a + b \text{ mod } 5)\) positions

uniformly random in Alice’s view

Ctx is uniformly random, independent of msg \(\rightarrow\) Perfect secrecy
Achieving Perfect Secrecy: One-time Pad

Key Gen

$K \xleftarrow{\$} \{0, 1\}^m$

Message space

Encrypt

$M \rightarrow K \rightarrow C$

Decrypt

$C \rightarrow K \rightarrow M$
Behind Every Notion, There Is An Assumption

For every $m$ and $c$:

$$\Pr_{K \leftarrow \mathcal{K}}[\text{Msg} = m \mid \mathcal{E}_K(\text{Msg}) = c] = \Pr[\text{Msg} = m]$$

It’s **assumed** that you pick a fresh key for each encryption.
Reusing One-time Pad Breaks Security

One can obtain $M \oplus M'$ via $C \oplus C'$

Can recover both $M$ and $M'$ if the messages are English texts and long enough
Bad Usage of One-time Pad: USSR’s reusing of one-time pads led to the decryption of 2900 messages.
Bad Usage of One-time Pads:

PPTP protocol in Windows NT
Fortinet’s blunder led them to reuse a one-time pad several times
Limitation of Perfect Secrecy

If $|\mathcal{M}| > |\mathcal{K}|$ then no scheme is perfectly secret

Impractical
1. Crypto Usage & Goals

2. Classical Crypto

3. One-time Pad & Perfect Secrecy

4. Modern Crypto
Modern Crypto: A Lego Approach

Primitives: AES  SHA-2  Factoring  ...

Applications: Encryption  MAC  Digital Signature

Transformers
Modern Crypto: A Computational Science

- Assume **computational** hardness of **a few** primitives

  AES  SHA-2  Factoring  ...

- Confidence by cryptanalysis
Modern Crypto: Provable Security

- Define security notions for applications
- Prove the transformer meets the notions