Lecture Outline

• What did we cover last time?

• What is ahead?
• Goal: secrecy of communication

• Basic terminology
  – plaintext or message
  – ciphertext
  – cryptographic key

• Encryption scheme is defined by algorithms
  – Gen: setup public parameters and key(s)
  – Enc: given a message $m$ and encryption key, output ciphertext $c$
  – Dec: given a ciphertext $c$ and decryption key, output plaintext $m$ or fail
• Gen can be configurable and takes a parameter \( n \in \mathbb{N} \) called security parameter

• Encryption scheme \( \mathcal{E} = (\text{Gen}, \text{Enc}, \text{Dec}) \) has associated
  – message space \( \mathcal{M} \)
  – ciphertext space \( \mathcal{C} \)
  – key space \( \mathcal{K} \)

• We obtain:
  – \( \text{Gen} : \mathbb{N} \rightarrow \mathcal{K} \)
  – \( \text{Enc} : \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C} \)
  – \( \text{Dec} : \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M} \)
• What do we want from an encryption scheme?
  – correctness
  – security
Types of Encryption

• Symmetric key encryption

• Public-key encryption

• How about cryptography beyond encryption?
History of Ciphers

- Date back to 2500+ years
- An ongoing battle between codemakers and codebreakers
- Driven by current communication and computation technology
  - paper and ink
  - radio, cryptographic engines
  - computers and digital communication
Caesar Cipher

- Caesar cipher works on individual letters
  - associates each letter with a number between 0 and 25, i.e., $A = 0$, $B = 1$, etc.
  - message space is $\mathcal{M} = \{0, \ldots, 25\}$ and ciphertext space is $\mathcal{C} = \{0, \ldots, 25\}$

- **Encryption**: shift the letter right by 3 positions, i.e.,
  $$\text{Enc}(m) = (m + 3) \mod 26$$

- **Decryption**: shift the letter left by 3 positions, i.e.,
  $$\text{Dec}(c) = (c - 3) \mod 26$$
Caesar Cipher

- Example

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

- Message $M = \text{CIPHER}$
- Ciphertext $C = ?$

- Assuming Kerckhoffs’ principle, how do you break shift cipher?
Shift Cipher

- Shift cipher is generalization of Caesar cipher
  - uses a key with key space $\mathcal{K} = \{1, \ldots, 25\}$
- Gen: choose $k \xleftarrow{R} \mathcal{K}$
- Enc: given key $k$, shift the letter right by $k$ positions, i.e.,
  $\text{Enc}_k(m) = (m + k) \mod 26$
- Dec: given key $k$, shift the letter left by $k$ positions, i.e.,
  $\text{Dec}_k(c) = (c - k) \mod 26$
- How hard is this one to break? What does it tell us?
Substitution Cipher

- Similarly, operates on one letter at a time ($\mathcal{M} = \mathcal{C} = \mathbb{Z}_{26}$)

- The key space consists of all possible permutations of the 26 symbols 0, …, 25

- Gen: choose a random permutation $\pi : \mathbb{Z}_{26} \rightarrow \mathbb{Z}_{26}$

- Enc: permute using $\pi$, i.e., $\text{Enc}_\pi(m) = \pi(m)$

- Dec: reverse permutation, i.e., $\text{Dec}_\pi(c) = \pi^{-1}(c)$, where $\pi^{-1}$ is the inverse permutation to $\pi$

- Example

\[
\begin{array}{cccccccccccccccccccc}
\end{array}
\]
Substitution Cipher

- **Key space** is $26! \approx 4 \cdot 10^{26}$
  - exhaustive (or brute-force) search is no longer possible
  - the cipher thought to be unbreakable at the time it was used

- The key to breaking the cipher lies in *frequency analysis*

- The **fact:** each language has certain features such as frequency of letters and frequency of groups of letters

- **Substitution cipher** preserves such features
Substitution Cipher: Cryptanalysis

- Probabilities of occurrence of English language letters:

<table>
<thead>
<tr>
<th>letter</th>
<th>prob</th>
<th>letter</th>
<th>prob</th>
<th>letter</th>
<th>prob</th>
<th>letter</th>
<th>prob</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.082</td>
<td>H</td>
<td>0.061</td>
<td>O</td>
<td>0.075</td>
<td>V</td>
<td>0.010</td>
</tr>
<tr>
<td>B</td>
<td>0.015</td>
<td>I</td>
<td>0.070</td>
<td>P</td>
<td>0.019</td>
<td>W</td>
<td>0.023</td>
</tr>
<tr>
<td>C</td>
<td>0.028</td>
<td>J</td>
<td>0.002</td>
<td>Q</td>
<td>0.001</td>
<td>X</td>
<td>0.001</td>
</tr>
<tr>
<td>D</td>
<td>0.043</td>
<td>K</td>
<td>0.008</td>
<td>R</td>
<td>0.060</td>
<td>Y</td>
<td>0.020</td>
</tr>
<tr>
<td>E</td>
<td>0.127</td>
<td>L</td>
<td>0.040</td>
<td>S</td>
<td>0.063</td>
<td>Z</td>
<td>0.001</td>
</tr>
<tr>
<td>F</td>
<td>0.022</td>
<td>M</td>
<td>0.024</td>
<td>T</td>
<td>0.091</td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>0.020</td>
<td>N</td>
<td>0.067</td>
<td>U</td>
<td>0.028</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- The common sequences of two or three consecutive letters (diagrams and trigrams, resp.) are also known.

- **Other language features**: vowels constitute 40% of plaintext, letter Q is always followed by U, etc.
Substitution Cipher: Cryptanalysis

• Given a ciphertext, **count different characters and their combinations** to determine the frequency of usage.

• **Examine the ciphertext for patterns**, repeated series, etc.

• **Replace ciphertext characters** with possible plaintext equivalents using known language characteristics.

• Example:
  
  YIFQFMZRQWQFYZVECFCMDZPCVMRZWNMDZVEJBTXCDDUMJ
  NDIFEFMDZCDMZKCEYFCJMYRNCWJCSZREXCHZUNMXZ
  NZUCDRJXYYSMRTMEYIFZWDYVZVYFZUMRZCRWNZDZJJ
  XZWGCHSMRNMHDHNCMFQCHZJMXJZWIEJYUCFWDJNZDIR
Another Attack on Shift Ciphers

- Using probabilities we can also automate cryptanalysis of shift cipher
  - why is previous approach harder to automate?

- How this attack works
  - let $p_i$ denote the probability of $i$th letter, $0 \leq i \leq 25$, in English text
  - using known values for $p_i$’s, we get
    $$\sum_{i=0}^{25} p_i^2 \approx 0.065$$
  - let $q_i$ denote the probability of $i$th letter in a ciphertext
    - how is it computed?
Another Attack on Shift Ciphers

- How this attack works (cont.)
  - if the key was $k$, then we expect $q_i + k \approx p_i$
  - so test each value of $k$ using
    \[ I_j = \sum_{i=0}^{25} p_i \cdot q_i + j \]
    for $0 \leq j \leq 25$
  - output $k$ for which $I_k$ is closest to 0.065
The security of the substitution cipher can be improved if each letter is mapped to different letters. Such ciphers are called polyalphabetic. Shift and substitution ciphers are both monoalphabetic.

In Vigenère cipher, the key is a string of length $\ell$ and is called a keyword.

Encryption is performed on $\ell$ characters at a time similar to the shift cipher.
Vigenère Cipher

- Gen: choose $\ell \leftarrow \mathbb{N}$ and random key $k \stackrel{R}{\leftarrow} \mathbb{Z}_{26}^\ell$

- Enc: given key $k = (k_1, k_2, \ldots, k_\ell)$, encrypt $\ell$-character message $m$ as
  \[
  \text{Enc}_k(m_1, \ldots, m_\ell) = ((m_1 + k_1) \mod 26, \ldots, (m_\ell + k_\ell) \mod 26)
  \]

- To decrypt $c$ using $k$:
  \[
  \text{Dec}_k(c_1, \ldots, c_\ell) = ((c_1 - k_1) \mod 26, \ldots, (c_\ell - k_\ell) \mod 26)
  \]
Example:

- using $\ell = 4$ and the keyword $k = \text{LUCK}$, encrypt the plaintext $m = \text{CRYPTOGRAPHY}$

- rewrite the key as $k = (11, 20, 2, 10)$ and compute the ciphertext as:

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>17</td>
<td>24</td>
<td>15</td>
<td>19</td>
<td>14</td>
<td>6</td>
<td>17</td>
<td>0</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>11</td>
<td>20</td>
<td>2</td>
<td>10</td>
<td>11</td>
<td>20</td>
<td>2</td>
<td>10</td>
<td>11</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>0</td>
<td>25</td>
<td>4</td>
<td>8</td>
<td>8</td>
<td>1</td>
<td>11</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

- the ciphertext is $c = \text{NLAZEIIBLJJI}$
Vigenère Cipher: Cryptanalysis

- Shift ciphers are vulnerable to frequency analysis attacks, but what about the Vigenère cipher?

- As the length of the keyword increases, usage of letters no longer follows language structure

- Think of this cipher as a collection of several shift ciphers

- Now the first task is to find the length of the key $\ell$

- Then we can divide the message into $\ell$ parts and use frequency analysis on each
Vigenère Cipher: Cryptanalysis

- There are two methods to find the key length: Kasisky test and index of coincidence

- Kasisky test:
  - two identical segments of plaintext will be encrypted to the same ciphertext if they are δ positions apart where δ ≡ 0 (mod ℓ)
  - search for identical segments (of length ≥ 3) and record the distances between them (δ₁, δ₂, . . .)
  - ℓ divides the δᵢ’s ⇒ ℓ divides gcd(δ₁, δ₂, . . .)
Vigenère Cipher: Cryptanalysis

- **Index of coincidence:**
  - assume we are given a string $x = x_1 x_2 \cdots x_n$ of $n$ characters
  - index of coincidence of $x$, $I_c(x)$, is measures the likelihood that two randomly drawn elements of $x$ are identical
  - as before, let $q_i$ denote probability of $i$th letter in $x$
  - index of coincidence is computed (in simplified form) as
    \[
    I_c(x) \approx \sum_{i=0}^{25} q_i^2
    \]
  - for English text, we get 0.065
  - for random strings, each $q_i$ has roughly the same probability
Vigenère Cipher: Cryptanalysis

- Index of coincidence:
  - for $q_i = 1/26$, we get
    \[
    I_c(x) = \sum_{i=0}^{25} \left( \frac{1}{26} \right)^2 = \frac{1}{26} \approx 0.038
    \]

- Thus we can test for various key lengths to see whether $I_c$ of the ciphertext is close to that of English

- We first divide the ciphertext string $c = c_1 \ldots c_n$ into $\ell$ substrings $s_1, \ldots, s_\ell$ and write them in a matrix
Vigenère Cipher: Cryptanalysis

- **Guessing key length:**

  \[
  \begin{bmatrix}
  c_1 & c_{\ell+1} & \cdots & c_{n-\ell+1} \\
  c_2 & c_{\ell+2} & \cdots & c_{n-\ell+2} \\
  \vdots & \vdots & \ddots & \vdots \\
  c_{\ell} & c_{2\ell} & \cdots & c_n
  \end{bmatrix}
  \begin{align*}
  &= C_1 \\
  &= C_2 \\
  &\vdots \\
  &= C_\ell
  \end{align*}
  \]

  - compute \( I_c(C_i) \) for \( i = 1, \ldots, \ell \)
  
  - if the values are not close to 0.065, try a different key length \( \ell \)

- Once the key size is determined, use frequency analysis on each \( C_i \)
How index of coincidence is derived

- denote the frequency of $i$th letter in $x$ by $f_i$
- so we have $q_i = f_i/n$ for $n$-character $x$
- we can choose two elements in $x$ in $\binom{n}{2}$ ways
  - recall that the binomial coefficient $\binom{n}{k} = \frac{n!}{k!(n-k)!}$
  - for each letter $i$, there are $\binom{f_i}{2}$ ways of choosing both elements to be $i$

$$I_c(x) = \frac{\sum_{i=0}^{25} \binom{f_i}{2}}{\binom{n}{2}} = \frac{\sum_{i=0}^{25} f_i(f_i - 1)}{n(n-1)} \approx \frac{\sum_{i=0}^{25} f_i^2}{n^2} = \sum_{i=0}^{25} q_i^2$$
Types of attacks on encryption:

- ciphertext only attack: the cryptanalyst knows a number of ciphertexts
- known plaintext attack: the cryptanalyst knows a number of ciphertexts and the corresponding plaintexts
- chosen plaintext attack: the cryptanalyst can obtain encryptions of chosen plaintext messages
- chosen ciphertext attack: the cryptanalyst can obtain decryptions of chosen ciphertexts

- Which did we use so far? what about others?

- How realistic are they?
Summary

- **Encryption**: definitions, types, properties

- **Shift ciphers** have small key space and are easy to break using brute force search

- **Substitution ciphers** preserve language features and are vulnerable to frequency analysis attacks

- **Vigenère ciphertexts** can be decrypted as well
  - once the key length is found, frequency analysis can be applied