So far: Shift, Affine  \[ f(x) = 3x + 2 \]

<table>
<thead>
<tr>
<th>X</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>D</td>
</tr>
<tr>
<td>B</td>
<td>E</td>
</tr>
<tr>
<td>C</td>
<td>F</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>C</td>
</tr>
<tr>
<td>B</td>
<td>F</td>
</tr>
<tr>
<td>Z</td>
<td>Z (fixed point)</td>
</tr>
</tbody>
</table>

Generalize this idea - any valid chart could be a key!

<table>
<thead>
<tr>
<th>X</th>
<th>f(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>P 26</td>
</tr>
<tr>
<td>B</td>
<td>A 25x</td>
</tr>
<tr>
<td>C</td>
<td>M 24</td>
</tr>
<tr>
<td>D</td>
<td>X etc.</td>
</tr>
</tbody>
</table>

(1) Pros
- lots of possible keys!!!  
  \[ 26! \approx 10^{18}\text{ or more} \]

(2) Cons
- harder to store key  
  (bigger)
But, since about the year 1000, substitution ciphers have been decrypted!

**How?**

al-Kindi - different letters appear diff # of times
substitution - set of letter freq
stays the same!

Invariant in substitution cipher \( \Rightarrow \) LETTER FREQ!

\[ \Rightarrow \text{GUESSING, PROCESS OF ELIMINATION} \]
\[ \Rightarrow \text{OTHER LANG FEATURES} \]

Frequencies won't be a perfect map!

in English
\[
12.7\% \rightarrow E \\
9.1\% \rightarrow T
\]

In small samples "error bar" is large.
Some samples might be biased.

Usually most freq letter is either E, T, A, O, S, H, R

Common digraphs, trigraphs "ED", "INO", "AND", "EN"

1) Make guesses, but
2) Be willing to backtrack if you hit something impossible.
3) KEEP TRACK OF WHAT YOU'VE TRIED!
Beginning rough! 2hrs → 6 letters
next 20 letters takes 10 min!

Queen Mary Communication

1) Hide in beer barrels (steganography)

2) Code
   Substitution +
   (1) 36 code words (the, for, it, etc.)
   (2) 4 null characters
   (3) doubleth - next char doubled “rr”

3) loyalists bribed a guard
   - guard double paid by Queen!

4) Sir Francis Walsingham (Queen Eliz.
   also getting messages/cryptanalyst)

Additional bellst whittles

delete char
100 cipher text (00...99)
assign by let freq e → 12 diff codes
t → 9 code
a → 8 codes

Each 100 codes will appear roughly equally!

Most of these were eventually broken!