Chapter 2 Classical Cryptosystems
Note

We will use the convention that plaintext will be lowercase and ciphertext will be in all capitals.
The idea of the Caesar cipher:

- To encrypt, shift the letters to the right by 3 and wrap around.

\[
\begin{align*}
a & \mapsto D \\
b & \mapsto E \\
x & \mapsto A
\end{align*}
\]
The idea of the Caesar cipher:

- To encrypt, shift the letters to the right by 3 and wrap around.
  
  \[
  \begin{align*}
  a &\mapsto D \\
  b &\mapsto E \\
  x &\mapsto A
  \end{align*}
  \]

- To decrypt, shift the letters to the left by 3 and wrap around.
The idea of the Caesar cipher:

- To encrypt, shift the letters to the right by 3 and wrap around.
  
  \[
  a \mapsto D \\
  b \mapsto E \\
  x \mapsto A
  \]

- To decrypt, shift the letters to the left by 3 and wrap around.

Useful to hide message from ‘friendly’ agents.
We could choose any integer between 1 and 25 to shift, with the number being secret.
We could choose any integer between 1 and 25 to shift, with the number being secret.

Example

Suppose you were given

```
Z UVKVJK KYV PREBVVJ
```

with no key. How would you decrypt?
We could choose any integer between 1 and 25 to shift, with the number being secret.

Example

Suppose you were given

Z UVKVJK KYV PREBVVJ

with no key. How would you decrypt?

A less naive attack:

- The first letter is single, so A or I would be good to start with.
We could choose any integer between 1 and 25 to shift, with the number being secret.

Example

Suppose you were given

Z UVKVJK KYV PREBVVJ

with no key. How would you decrypt?

A less naive attack:

- The first letter is single, so A or I would be good to start with.
- Start with just 2-3 letters to see if it is the start of a word
Attacks

No matter the technique, this is easy to break using a ciphertext attack.
Attacks

No matter the technique, this is easy to break using a ciphertext attack.

**Definition**
A ciphertext attack is performed when all that is had is a copy of the ciphertext.
Attacks

No matter the technique, this is easy to break using a **ciphertext attack**.

**Definition**

A **ciphertext attack** is performed when all that is had is a copy of the ciphertext.

Even worse, if the attacker was performing a known-plaintext attack, they would have the decryption key while only needing a single letter.
No matter the technique, this is easy to break using a ciphertext attack.

**Definition**

A ciphertext attack is performed when all that is had is a copy of the ciphertext.

Even worse, if the attacker was performing a known-plaintext attack, they would have the decryption key while only needing a single letter.

**Definition**

A known plaintext attack is when an attacker has a ciphertext and the corresponding plaintext. If the key is not changed, they can decrypt future ciphertexts.
This is a method for encrypting with numbers.
This is a method for encrypting with numbers.

**Procedure**

Take \( \alpha, \beta \) with \( 0 \leq \alpha, \beta \leq 25 \) such that \( (\alpha, 26) = 1 \). Then, the affine function

\[
x \mapsto \alpha x + \beta \pmod{26}
\]

It is a linear transformation followed by a translation.

**Alternate Notation**

\[
E_{\alpha, \beta}(x) \equiv (\alpha x + \beta) \pmod{26}
\]
Encrypting Using Affine Ciphers

Example

\[ E_{3,11}(hello) = GXSSB \]
Encrypting Using Affine Ciphers

Example

\[ E_{3,11}(hello) = GXSSB \]

\[ h = 7 \Rightarrow 3 \cdot 7 + 11 = 32 \equiv 6 \pmod{26} \]
Encrypting Using Affine Ciphers

Example

\[ E_{3,11}(hello) = GXSSB \]

\[ h = 7 \Rightarrow 3 \cdot 7 + 11 = 32 \equiv 6 \pmod{26} \]
\[ e = 4 \Rightarrow 3 \cdot 4 + 11 = 23 \equiv 23 \pmod{26} \]
Encrypting Using Affine Ciphers

Example

\[ E_{3,11}(hello) = GXSSB \]

\[
\begin{align*}
h &= 7 \Rightarrow 3 \cdot 7 + 11 = 32 \equiv 6 \pmod{26} \\
e &= 4 \Rightarrow 3 \cdot 4 + 11 = 23 \equiv 23 \pmod{26} \\
l &= 11 \Rightarrow 3 \cdot 11 + 11 = 44 \equiv 18 \pmod{26}
\end{align*}
\]
Encrypting Using Affine Ciphers

Example

\[ E_{3,11}(hello) = GXSSB \]

\[
\begin{align*}
  h &= 7 \Rightarrow 3 \cdot 7 + 11 = 32 \equiv 6 \pmod{26} \\
  e &= 4 \Rightarrow 3 \cdot 4 + 11 = 23 \equiv 23 \pmod{26} \\
  l &= 11 \Rightarrow 3 \cdot 11 + 11 = 44 \equiv 18 \pmod{26} \\
  o &= 14 \Rightarrow 3 \cdot 14 + 11 = 53 \equiv 1 \pmod{26}
\end{align*}
\]
Decryption with the Affine Cipher

What do we need?
Decryption with the Affine Cipher

What do we need? Inverses ...

\[ y = 3x + 11 \Rightarrow x = \frac{1}{3}(y - 11) \]
Decryption with the Affine Cipher

What do we need? Inverses ...

\[ y = 3x + 11 \implies x = \frac{1}{3}(y - 11) \]

Problem?
Decryption with the Affine Cipher

What do we need? Inverses ...

\[ y = 3x + 11 \Rightarrow x = \frac{1}{3}(y - 11) \]

Problem?
We need to represent \( \frac{1}{3} \) in terms of (mod 26). Since \((3, 26) = 1\), the multiplicative inverse of 3 exists modulo 26.
Decryption with the Affine Cipher

What do we need? Inverses ...

\[ y = 3x + 11 \Rightarrow x = \frac{1}{3}(y - 11) \]

Problem?
We need to represent \( \frac{1}{3} \) in terms of (mod 26). Since \((3, 26) = 1\), the multiplicative inverse of 3 exists modulo 26.

\[ 9 \cdot 3 = 27 \equiv 1\pmod{26} \]
Decryption with the Affine Cipher

What do we need? Inverses ...

\[ y = 3x + 11 \Rightarrow x = \frac{1}{3}(y - 11) \]

Problem?
We need to represent \( \frac{1}{3} \) in terms of (mod 26). Since \((3, 26) = 1\), the multiplicative inverse of 3 exists modulo 26.

\[ 9 \cdot 3 = 27 \equiv 1 \pmod{26} \]

So, we can replace \( \frac{1}{3} \) with 9 in our mapping.

\[ x = 9(y - 11) \]
Decryption with the Affine Cipher

What do we need? Inverses ...

\[ y = 3x + 11 \Rightarrow x = \frac{1}{3} (y - 11) \]

Problem?
We need to represent \( \frac{1}{3} \) in terms of (mod 26). Since \((3, 26) = 1\), the multiplicative inverse of 3 exists modulo 26.

\[ 9 \cdot 3 = 27 \equiv 1 \pmod{26} \]

So, we can replace \( \frac{1}{3} \) with 9 in our mapping.

\[ x = 9(y - 11) \]

So, we see for \( G = 6 \), we have

\[ 9(6 - 11) = -45 \pmod{26} \equiv 7 \pmod{26} \]
Known Plaintext Attack

Example

Suppose we had the following plaintext-ciphertext pairs:

\[ f \mapsto K \]
\[ l \mapsto Z \]

Can we find the key?
Known Plaintext Attack

Example

Suppose we had the following plaintext-ciphertext pairs:

\[ f \mapsto K \]
\[ l \mapsto Z \]

Can we find the key?

First, we convert this to numerical values.

\[ f \mapsto K \Rightarrow 5 \mapsto 10 \]
\[ l \mapsto Z \Rightarrow 11 \mapsto 25 \]
Known Plaintext Attack

Example

Suppose we had the following plaintext-ciphertext pairs:

$$f \mapsto K$$
$$l \mapsto Z$$

Can we find the key?

First, we convert this to numerical values.

$$f \mapsto K \Rightarrow 5 \mapsto 10$$
$$l \mapsto Z \Rightarrow 11 \mapsto 25$$

This gives

$$5\alpha + \beta \equiv 10 \pmod{26}$$
$$11\alpha + \beta \equiv 25 \pmod{26}$$
**Example**

Suppose we had the following plaintext-ciphertext pairs:

\[
\begin{align*}
 f & \mapsto K \\
 l & \mapsto Z
\end{align*}
\]

Can we find the key?

First, we convert this to numerical values.

\[
\begin{align*}
 f & \mapsto K \Rightarrow 5 \mapsto 10 \\
 l & \mapsto Z \Rightarrow 11 \mapsto 25
\end{align*}
\]

This gives

\[
\begin{align*}
 5\alpha + \beta & \equiv 10 \pmod{26} \\
 11\alpha + \beta & \equiv 25 \pmod{26}
\end{align*}
\]

Combining gives

\[
6\alpha \equiv 15 \pmod{26}
\]
Known Plaintext Attack

Example

Suppose we had the following plaintext-ciphertext pairs:

\[ f \mapsto K \]
\[ l \mapsto Z \]

Can we find the key?

First, we convert this to numerical values.

\[ f \mapsto K \Rightarrow 5 \mapsto 10 \]
\[ l \mapsto Z \Rightarrow 11 \mapsto 25 \]

This gives

\[ 5\alpha + \beta \equiv 10 \pmod{26} \]
\[ 11\alpha + \beta \equiv 25 \pmod{26} \]

Combining gives

\[ 6\alpha \equiv 15 \pmod{26} \]

This will not work. Every choice for \( \alpha \) will produce an even number and an even taken modulo 26 will always be even.
Another Known Plaintext Attack

Example

Suppose we had the following plaintext-ciphertext pairs:

\[ f \mapsto K \]
\[ i \mapsto Z \]

Can we find the key?
Another Known Plaintext Attack

Example

Suppose we had the following plaintext-ciphertext pairs:

\[ f \mapsto K \]
\[ i \mapsto Z \]

Can we find the key?

We begin as before.

\[ f \mapsto K \Rightarrow 5 \mapsto 10 \]
\[ i \mapsto Z \Rightarrow 8 \mapsto 25 \]
Example

Suppose we had the following plaintext-ciphertext pairs:

\[ f \mapsto K \]
\[ i \mapsto Z \]

Can we find the key?

We begin as before.

\[ f \mapsto K \Rightarrow 5 \mapsto 10 \]
\[ i \mapsto Z \Rightarrow 8 \mapsto 25 \]

This gives

\[ 5\alpha + \beta \equiv 10 \pmod{26} \]
\[ 8\alpha + \beta \equiv 25 \pmod{26} \]
Another Known Plaintext Attack

Example

Suppose we had the following plaintext-ciphertext pairs:

\[ f \mapsto K \]
\[ i \mapsto Z \]

Can we find the key?

We begin as before.

\[ f \mapsto K \Rightarrow 5 \mapsto 10 \]
\[ i \mapsto Z \Rightarrow 8 \mapsto 25 \]

This gives

\[ 5\alpha + \beta \equiv 10 \pmod{26} \]
\[ 8\alpha + \beta \equiv 25 \pmod{26} \]

Combining gives

\[ 3\alpha \equiv 15 \pmod{26} \]
We can divide because \((3, 26) = 1\), so this becomes

\[ \alpha \equiv 5 \pmod{26} \]
Finishing the Decryption Algorithm

We can divide because \( (3, 26) = 1 \), so this becomes

\[
\alpha \equiv 5 \pmod{26}
\]

And knowing \( \alpha = 5 \) gives that \( \beta = 11 \).
We can divide because \((3, 26) = 1\), so this becomes

\[
\alpha \equiv 5 \pmod{26}
\]

And knowing \(\alpha = 5\) gives that \(\beta = 11\).

Therefore the cipher would be \(5x + 11 \pmod{26}\).
We have to be careful here with our choice as we need the mapping to be 1-1 modulo 26 and this only happens when $(\alpha, 26) = 1$. Here's what happens if we aren't careful: Suppose we try to decrypt, we get $1^{2}$ to find the inverse of. Using the other technique, we cannot find $a^\ast \not\equiv 1 \pmod{26}$, so no multiplicative inverse exists. This would give a non-unique deciphering for the same ciphertext.
We have to be careful here with our choice as we need the mapping to be 1-1 modulo 26 and this only happens when \((\alpha, 26) = 1\).

Here’s what happens if we aren’t careful:

Suppose we try to decrypt, we get \(\frac{1}{2}\) to find the inverse of. Using the other technique, we cannot find

\[ a^* \not\equiv 2a^* \equiv 1 \pmod{26} \]

So, no multiplicative inverse exists. This would give a non-unique deciphering for the same ciphertext.
How many keys are there for a shift cipher?
How many keys are there for a shift cipher? 25

How many keys are there for an affine cipher?
Improvement Over Shifts

How many keys are there for a shift cipher? 25

How many keys are there for an affine cipher?

We could have

- possible values for \( \alpha \)?
Improvement Over Shifts

How many keys are there for a shift cipher? 25

How many keys are there for an affine cipher?

We could have
  - possible values for $\alpha$? 12
  - possible values for $\beta$?
How many keys are there for a shift cipher? 25

How many keys are there for an affine cipher?

We could have

- possible values for $\alpha$? 12
- possible values for $\beta$? 26
How many keys are there for a shift cipher? 25

How many keys are there for an affine cipher?

We could have

- possible values for $\alpha$? 12
- possible values for $\beta$? 26

So the total would be 312 different ciphers.

Another improvement: And, we would need two plaintext-ciphertext pairs to deduce $\alpha$ and $\beta$ instead of one pair for a shift cipher.
One of the main problems with simple substitution ciphers is that they are so vulnerable to frequency analysis. Given a sufficiently large ciphertext, it can easily be broken by mapping the frequency of its letters to the known frequencies of, say, English text.
Vigenere Ciphers

One of the main problems with simple substitution ciphers is that they are so vulnerable to frequency analysis. Given a sufficiently large ciphertext, it can easily be broken by mapping the frequency of its letters to the known frequencies of, say, English text.

Therefore, to make ciphers more secure, cryptographers have long been interested in developing enciphering techniques that are immune to frequency analysis. One of the most common approaches is to suppress the normal frequency data by using more than one alphabet to encrypt the message.
One of the main problems with simple substitution ciphers is that they are so vulnerable to frequency analysis. Given a sufficiently large ciphertext, it can easily be broken by mapping the frequency of its letters to the known frequencies of, say, English text.

Therefore, to make ciphers more secure, cryptographers have long been interested in developing enciphering techniques that are immune to frequency analysis. One of the most common approaches is to suppress the normal frequency data by using more than one alphabet to encrypt the message.

A polyalphabetic substitution cipher involves the use of two or more cipher alphabets. Instead of there being a one-to-one relationship between each letter and its substitute, there is a one-to-many relationship between each letter and its substitutes.
One such cipher is attributed to Blaise de Vigenere from the court of Henry the III of France in the 16\textsuperscript{th} century.

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

So how does this work? We have to have plaintext to encrypt.

the red sox will make the playoffs

and we need a keyword.

boston
So how does this work? We have to have plaintext to encrypt.

the red sox will make the playoffs

and we need a keyword.

boston

Now, we line up the key word, as many times as needed, with the message to encrypt.

b o s t o n b o s t o n
t h e r e d s o x w i l l
b o s t o n b o s t o n
l m a k e t h e p l a y
b o s t o n b o s t o n
o f f f s
Now, find in the table the intersection of the row for “b” and the column for ”t” and that is the letter for the cipher text. We see that the cipher text would be “U”
Now, find in the table the intersection of the row for “b” and the column for ”t” and that is the letter for the cipher text. We see that the cipher text would be “U”

To decrypt, we can use the row for the appropriate letter from the keyword and search for the cipher text character. So, for the letter “U”, we scroll down the row (or column) for “b” and when we find “U”, we see that the intersecting column (or row) is for the letter “t”.

Vigenere Ciphers

Example

Cipher text: VMFF JSFIRVW VF QR SINAEV XH KHB EG NLXR BWM FVGHHH BZKJ XH XHTD OV MMW DJGWK XAEW BAOV
Keyword: discrete

Send someone to my office when you decode this to tell me it's about that time.
Vigenere Ciphers

Example

Cipher text: VMFF JSFIRVW VF QR SINAEV XH KHB EG NLXR BWM FVGHHH BZKJ XH XHTD OV MMW DJGWK XAEW BAOV
Keyword: discrete

Send someone to my office when you decode this to tell me it’s about that time.
Shift and affine ciphers are substitution ciphers - permute one letter at a time.
Shift and affine ciphers are substitution ciphers - permute one letter at a time

Playfair ciphers permute two at a time
§2.6 Playfair and ADFGX Ciphers

- Shift and affine ciphers are substitution ciphers - permute one letter at a time
- Playfair ciphers permute two at a time
- Invented in 1854 by Charles Wheatstone
Shift and affine ciphers are substitution ciphers - permute one letter at a time

Playfair ciphers permute two at a time

Invented in 1854 by Charles Wheatstone

Named after Lord Playfair, who promoted the usage
Shift and affine ciphers are substitution ciphers - permute one letter at a time

Playfair ciphers permute two at a time

Invented in 1854 by Charles Wheatstone

Named after Lord Playfair, who promoted the usage

Used by British in Second Boer War and WWI
§2.6 Playfair and ADFGX Ciphers

- Shift and affine ciphers are substitution ciphers - permute one letter at a time
- Playfair ciphers permute two at a time
- Invented in 1854 by Charles Wheatstone
- Named after Lord Playfair, who promoted the usage
- Used by British in Second Boer War and WWI
- Used by Australians and Germans in WWII
How It Works

We first need a keyword: we’ll use Boston.
We first need a keyword: we’ll use Boston. Letters non-distinct, we delete all copies after first of any repeated letter. So, our keyword becomes bostn.
We first need a keyword: we’ll use Boston. Letters non-distinct, we delete all copies after first of any repeated letter. So, our keyword becomes bostn. We construct a $5 \times 5$ grid with the keyword first and then the rest of the alphabet that is unused (with the convention $i = j$).

```
b o s t n
a c d e f
g h i k l
m p q r u
v w x y z
```
Now we need a message to encode.

*patriots will beat the jets*
Now we need a message to encode.

*patriots will beat the jets*

We break up the text into pairs of letters

```
pa  tr  io  ts  wi  ll
```
Now we need a message to encode.

*patriots will beat the jets*

We break up the text into pairs of letters

```
pa  tr  io  ts  wi  ll
```

Problem ...
Now we need a message to encode.

*patriots will beat the jets*

We break up the text into pairs of letters

```
    pa  tr  io  ts  wi  ll
```

Problem ...

```
    pa  tr  io  ts  wi  lx
    lb  ea  tx  th  ej  et
    sx
```
Rules for Encoding

1. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row cyclically treated.
Rules for Encoding

1. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row cyclically treated.

2. Two plaintext letters that fall in the same column of the matrix are each replaced by the letter beneath them, with the first element of the column cyclically treated.
Rules for Encoding

1. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row cyclically treated.

2. Two plaintext letters that fall in the same column of the matrix are each replaced by the letter beneath them, with the first element of the column cyclically treated.

3. Otherwise each plaintext letter is replaced by the letter that lies in its row and column occupied by the other plaintext letter.
pa is a pair from different rows and columns.

<table>
<thead>
<tr>
<th>b</th>
<th>o</th>
<th>s</th>
<th>t</th>
<th>n</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
</tr>
<tr>
<td>g</td>
<td>h</td>
<td>i</td>
<td>k</td>
<td>l</td>
</tr>
<tr>
<td>m</td>
<td>p</td>
<td>q</td>
<td>r</td>
<td>u</td>
</tr>
<tr>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
</tr>
</tbody>
</table>

Each plaintext letter is replaced but the letter that lies in its row and column occupied by the other plaintext letter.
Example

pa is a pair from different rows and columns.

\[
\begin{array}{cccccc}
  b & o & s & t & n \\
  a & c & d & e & f \\
  g & h & i & k & l \\
  m & p & q & r & u \\
  v & w & x & y & z \\
\end{array}
\]

Each plaintext letter is replaced but the letter that lies in its row and column occupied by the other plaintext letter.

\[
p \mapsto M \\
a \mapsto C
\]
tr is a pair in the same column.

Two plaintext letters that fall in the same column of the matrix are each replaced by the letter beneath them, with the first element of the column cyclically treated.
Example

tr is a pair in the same column.

\[
\begin{array}{cccccc}
\text{b} & \text{o} & \text{s} & \text{t} & \text{n} \\
\text{a} & \text{c} & \text{d} & \text{e} & \text{f} \\
\text{g} & \text{h} & \text{i} & \text{k} & \text{l} \\
\text{m} & \text{p} & \text{q} & \text{r} & \text{u} \\
\text{v} & \text{w} & \text{x} & \text{y} & \text{z}
\end{array}
\]

Two plaintext letters that fall in the same column of the matrix are each replaced by the letter beneath them, with the first element of the column cyclically treated.

\[
t \mapsto E
\]
\[
r \mapsto Y
\]

Finish the ciphertext.
The Ciphertext

Now, we remove the spaces to finish the encryption.

MCEYHSNTXHIZGNFCSYOKDKKEDS

To decrypt, we essentially reverse the process, provided we know the key.
Now, we remove the spaces to finish the encryption.

MCEYHSNTXHIZGNFCSYOKDKKEDS

to decrypt, we essentially reverse the process, provided we know the key.
The Ciphertext

Now, we remove the spaces to finish the encryption.

MCEYHSNTXHIZGNFCSYOKDKKEDS
Now, we remove the spaces to finish the encryption.

$MCEYHSNTXHIZGNFCSYOKDKKEDS$

To decrypt, we essentially reverse the process, provided we know the key.
Weaknesses

- If the key is not known, once broken into pairs, common pairs are searched by looking for repetitive pairs and trial and error to see which common pairs it is.
  Does anyone know what pairs have the highest probability?
Weaknesses

- If the key is not known, once broken into pairs, common pairs are searched by looking for repetitive pairs and trial and error to see which common pairs it is. Does anyone know what pairs have the highest probability? Probability dictates that the common pairs are th, he, an, in, re, es.

- Another weakness is that there are only 5 possible letters for each ciphertext letter.
Weaknesses

- If the key is not known, once broken into pairs, common pairs are searched by looking for repetitive pairs and trial and error to see which common pairs it is.
  Does anyone know what pairs have the highest probability? Probability dictates that the common pairs are th, he, an, in, re, es.

- Another weakness is that there are only 5 possible letters for each ciphertext letter.
  This is an improvement as there are $26^2$ digrams (2 letter pairs) v. only 26 letters in prior methods.
Block ciphers use symmetric keys to encrypt a string of consecutive characters through the use of matrices.
Block ciphers use symmetric keys to encrypt a string of consecutive characters through the use of matrices.

To work with these, we need a little linear algebra.
Block ciphers use symmetric keys to encrypt a string of consecutive characters through the use of matrices.

To work with these, we need a little linear algebra.

We define the inverse of a square matrix $M$, denoted $M^{-1}$, by the equation

$$MM^{-1} = M^{-1}M = I_n$$

The inverse does not always exist, but when it does this equation is satisfied.
Is the Matrix Invertible?

Easiest way to determine if $M$ is invertible is by taking determinants.

Example

Determine if $A$ is invertible.

\[
\begin{vmatrix}
5 & 8 \\
17 & 3
\end{vmatrix}
\]

\[
\begin{vmatrix}
5 & 8 \\
17 & 3
\end{vmatrix}
= 5(3) - 8(17) = -121 \neq 0
\]

Since $\det(A) \neq 0$, $A$ is invertible.
Easiest way to determine if $M$ is invertible is by taking determinants.
Easiest way to determine if $M$ is invertible is by taking determinants.

Example
Determine if $A$ is invertible.

\[
\begin{bmatrix}
5 & 8 \\
17 & 3 \\
\end{bmatrix}
\]
Is the Matrix Invertible?

Easiest way to determine if $M$ is invertible is by taking determinants.

**Example**

Determine if $A$ is invertible.

\[
\begin{vmatrix}
5 & 8 \\
17 & 3 \\
\end{vmatrix}
\]

\[
\begin{vmatrix}
5 & 8 \\
17 & 3 \\
\end{vmatrix} = 5(3) - 8(17) = -121 \neq 0
\]

Since $det(A) \neq 0$, $A$ is invertible.
Example

Find the inverse of $A$. 

Using the algorithm for an $2 \times 2$ matrix, we get

\[
\begin{bmatrix}
-1 & 21 \\
3 & -8 \\
-17 & 5 \\
\end{bmatrix}
\]

\((\text{mod } 26)\)

Problem?

\[-121 \equiv 9 \pmod{26}\]

so we have

\[
\begin{bmatrix}
1 & 9 \\
3 & -8 \\
-17 & 5 \\
\end{bmatrix}
\]

\((\text{mod } 26)\)
Example

Find the inverse of $A$.

Using the algorithm for a $2 \times 2$ matrix, we get

\[-\frac{1}{121} \begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \pmod{26}\]
Example

Find the inverse of $A$.

Using the algorithm for a $2 \times 2$ matrix, we get

$$\frac{-1}{121} \begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \pmod{26}$$

Problem?
Finding the Inverse

Example

Find the inverse of $A$.

Using the algorithm for an $2 \times 2$ matrix, we get

$$-\frac{1}{121} \begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \pmod{26}$$

Problem?

$$-121 \equiv 9 \pmod{26}$$

so we have

$$\frac{1}{9} \begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \pmod{26}$$
Still a problem?
Finding the Inverse

Still a problem?

\[ 3 \cdot 9 \equiv 1 \pmod{26} \]

we can replace \( \frac{1}{9} \) by 3 to get

\[ 3 \left[ \begin{array}{cc} 3 & -8 \\ -17 & 5 \end{array} \right] \pmod{26} \]
Finding the Inverse

Still a problem?

\[ 3 \cdot 9 \equiv 1 \pmod{26} \]

we can replace \( \frac{1}{9} \) by 3 to get

\[ 3 \begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \pmod{26} \]

Done?
Finding the Inverse

Still a problem?

$$3 \cdot 9 \equiv 1 \pmod{26}$$

we can replace $\frac{1}{9}$ by 3 to get

$$3 \begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \pmod{26}$$

Done?

$$\begin{bmatrix} 9 & -24 \\ -51 & 15 \end{bmatrix} \pmod{26}$$
Finding the Inverse

Still a problem?

\[ 3 \cdot 9 \equiv 1 \pmod{26} \]

we can replace \( \frac{1}{9} \) by 3 to get

\[ 3 \begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \pmod{26} \]

Done?

\[ \begin{bmatrix} 9 & -24 \\ -51 & 15 \end{bmatrix} \pmod{26} \]

Now?
Finding the Inverse

Still a problem?

\[ 3 \cdot 9 \equiv 1 \pmod{26} \]

we can replace \( \frac{1}{9} \) by 3 to get

\[
3 \begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} \pmod{26}
\]

Done?

\[
\begin{bmatrix} 9 & -24 \\ -51 & 15 \end{bmatrix} \pmod{26}
\]

Now?

\[
A^{-1} = \begin{bmatrix} 9 & 2 \\ 1 & 15 \end{bmatrix}
\]
This is named after Lester Hill, who produced work in 1929.

The encryption algorithm takes \( m \) successive plaintext letters and substitutes them for \( m \) ciphertext letters. This substitution is determined by \( m \) linear equations in which each character is assigned a numerical value (\( a = 0, b = 1, \ldots \)).
For $m = 3$, the system of equations can be described as

\[
\begin{align*}
c_1 &= (k_{11}p_1 + k_{21}p_2 + k_{31}p_3)(\text{mod } 26) \\
c_2 &= (k_{12}p_1 + k_{22}p_2 + k_{32}p_3)(\text{mod } 26) \\
c_3 &= (k_{13}p_1 + k_{23}p_2 + k_{33}p_3)(\text{mod } 26)
\end{align*}
\]
How the Hill Cipher Works

For \( m = 3 \), the system of equations can be described as

\[
\begin{align*}
  c_1 &= (k_{11}p_1 + k_{21}p_2 + k_{31}p_3)(\text{mod } 26) \\
  c_2 &= (k_{12}p_1 + k_{22}p_2 + k_{32}p_3)(\text{mod } 26) \\
  c_3 &= (k_{13}p_1 + k_{23}p_2 + k_{33}p_3)(\text{mod } 26)
\end{align*}
\]

This can now be expressed in terms of vectors and matrices

\[
\begin{bmatrix}
  c_1 \\
  c_2 \\
  c_3
\end{bmatrix} =
\begin{bmatrix}
  p_1 \\
  p_2 \\
  p_3
\end{bmatrix} 
\begin{bmatrix}
  k_{11} & k_{12} & k_{13} \\
  k_{21} & k_{22} & k_{23} \\
  k_{31} & k_{32} & k_{33}
\end{bmatrix} (\text{mod } 26)
\]

where \( c \) and \( p \) are vectors of length 3 representing the plaintext and associated ciphertext.
Using the Hill Algorithm

Example

Using the encryption key

\[
\begin{bmatrix}
17 & 17 & 5 \\
21 & 18 & 21 \\
2 & 2 & 19
\end{bmatrix}
\]

encrypt ‘pay more money’.
Using the Hill Algorithm

Example

Using the encryption key

\[
\begin{bmatrix}
17 & 17 & 5 \\
21 & 18 & 21 \\
2 & 2 & 19
\end{bmatrix}
\]

encrypt ‘pay more money’.

The first 3 letters of the plaintext are represented by the vector

\[
\begin{bmatrix}
15 & 0 & 24
\end{bmatrix}
\]
Then,

\[
\begin{bmatrix}
15 & 0 & 24 \\
\end{bmatrix}
K = 
\begin{bmatrix}
303 & 303 & 531 \\
\end{bmatrix}
\pmod{26}
\]
Then,

$$\begin{bmatrix} 15 & 0 & 24 \end{bmatrix} K = \begin{bmatrix} 303 & 303 & 531 \end{bmatrix} \pmod{26}$$

Which, when take modulo 26, becomes

$$\begin{bmatrix} 17 & 17 & 11 \end{bmatrix}$$

and this corresponds to RRL.
Then,

\[
\begin{bmatrix}
15 & 0 & 24
\end{bmatrix} \mathbf{K} = \begin{bmatrix}
303 & 303 & 531
\end{bmatrix} \pmod{26}
\]

Which, when take modulo 26, becomes

\[
\begin{bmatrix}
17 & 17 & 11
\end{bmatrix}
\]

and this corresponds to RRL.
Continuing in this way, the ciphertext for the whole plaintext is

*RRLMWBKASPDH*
Decryption with the Hill Cipher

Decryption requires using the inverse of the matrix $K$. First,

$$\begin{vmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{vmatrix} = 23 \neq 0$$

When $K^{-1}$ is applied to the ciphertext, the plaintext is easily recovered.

$$\begin{bmatrix} 17 & 17 & 11 \end{bmatrix} K^{-1} = \begin{bmatrix} 587 & 442 & 544 \end{bmatrix} \pmod{26} = \begin{bmatrix} 15 & 0 & 24 \end{bmatrix}$$
Decryption with the Hill Cipher

Decryption requires using the inverse of the matrix $K$. First,

\[
\begin{vmatrix}
17 & 17 & 5 \\
21 & 18 & 21 \\
2 & 2 & 19
\end{vmatrix} = 23 \neq 0
\]

\[
K^{-1} = \begin{bmatrix}
4 & 9 & 15 \\
15 & 17 & 6 \\
24 & 0 & 17
\end{bmatrix}
\]

when taken modulo 26.
Decryption with the Hill Cipher

Decryption requires using the inverse of the matrix $K$. First,

$$\begin{vmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{vmatrix} = 23 \neq 0$$

$$K^{-1} = \begin{bmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{bmatrix}$$

when taken modulo 26.

If $K^{-1}$ is applied to the ciphertext, then the plaintext is easily recovered.

$$\begin{bmatrix} 17 & 17 & 11 \end{bmatrix} K^{-1} = \begin{bmatrix} 587 & 442 & 544 \end{bmatrix} \pmod{26}$$

$$= \begin{bmatrix} 15 & 0 & 24 \end{bmatrix}$$
What’s Good and Bad

- As with the Playfair cipher, the strength of the Hill cipher is that it completely hides single letter frequencies.
What’s Good and Bad

- As with the Playfair cipher, the strength of the Hill cipher is that it completely hides single letter frequencies.
- And, with Hill, the use of larger matrices hides more frequency information.
What’s Good and Bad

- As with the Playfair cipher, the strength of the Hill cipher is that it completely hides single letter frequencies.
- And, with Hill, the use of larger matrices hides more frequency information.
- For example, $K \in M_3$ hides 2 letter frequencies.
What’s Good and Bad

- As with the Playfair cipher, the strength of the Hill cipher is that it completely hides single letter frequencies.
- And, with Hill, the use of larger matrices hides more frequency information.
- For example, $K \in M_3$ hides 2 letter frequencies.
- Although the Hill cipher is strong against the cipher-only attack, it is easily broken down with a known plaintext attack.
As with the Playfair cipher, the strength of the Hill cipher is that it completely hides single letter frequencies.

And, with Hill, the use of larger matrices hides more frequency information.

For example, $K \in \mathcal{M}_3$ hides 2 letter frequencies.

Although the Hill cipher is strong against the cipher-only attack, it is easily broken down with a known plaintext attack.

If we have an $m \times m$ Hill cipher, suppose we have $m$ plaintext-ciphertext pairs, each of length $m$. We label the pairs $\vec{P}_j = (p_{1j}, p_{2j}, \ldots, p_{mj})$ and $\vec{C}_j = (c_{1j}, c_{2j}, \ldots, c_{mj})$ such that $\vec{C}_j = \vec{P}_j K$ for $1 \leq j \leq m$ and for some unknown key matrix $K$. 
What’s Good and Bad

- As with the Playfair cipher, the strength of the Hill cipher is that it completely hides single letter frequencies.
- And, with Hill, the use of larger matrices hides more frequency information.
- For example, $K \in \mathcal{M}_3$ hides 2 letter frequencies.
- Although the Hill cipher is strong against the cipher-only attack, it is easily broken down with a known plaintext attack.
- If we have an $m \times m$ Hill cipher, suppose we have $m$ plaintext-ciphertext pairs, each of length $m$. We label the pairs $\vec{P}_j = (p_{1j}, p_{2j}, \ldots, p_{mj})$ and $\vec{C}_j = (c_{1j}, c_{2j}, \ldots, c_{mj})$ such that $\vec{C}_j = \vec{P}_j K$ for $1 \leq j \leq m$ and for some unknown key matrix $K$.
- Now, define $X, Y \in \mathcal{M}_m$ such that $X = (p_{ij})$ and $Y = (c_{ij})$. Then we can form the equation $Y = XK$. If $X$ is invertible, $K = X^{-1}Y$ and we are done. If not, then a new version of $X$ can be formed with additional plaintext-ciphertext pairs until and invertible $X$ is obtained.
Example

Suppose the plaintext ‘hill cipher’ is encrypted using a $2 \times 2$ Hill cipher to yield the ciphertext HCRZSSXNSP. Find the encryption key.

Based on the plaintext-cipher text pairs we have, we can set up the following:

$$
\begin{bmatrix}
7 & 8 \\
11 & 11
\end{bmatrix}
\equiv
\begin{bmatrix}
7 & 2 \\
17 & 25
\end{bmatrix}
\pmod{26}
$$

and so forth.
Example

Suppose the plaintext ‘hill cipher’ is encrypted using a $2 \times 2$ Hill cipher to yield the ciphertext HCRZSSXNSP. Find the encryption key.

Based on the plaintext-cipher text pairs we have, we can set up the following:

\[
\begin{bmatrix} 7 & 8 \\ 11 & 11 \end{bmatrix} K(\text{mod } 26) = \begin{bmatrix} 7 & 2 \\ 17 & 25 \end{bmatrix}
\]

and so forth.
Using the first two plaintext-ciphertext pairs, we have

\[
\begin{bmatrix}
7 & 2 \\
17 & 25
\end{bmatrix} = \begin{bmatrix}
7 & 8 \\
11 & 11
\end{bmatrix} K \pmod{26}
\]
Using the first two plaintext-ciphertext pairs, we have

\[
\begin{bmatrix}
7 & 2 \\
17 & 25
\end{bmatrix} = \begin{bmatrix}
7 & 8 \\
11 & 11
\end{bmatrix} K \pmod{26}
\]

Now, we need to find the inverse of \( P \). Since this is a 2 \times 2 system, we have the following algorithm:

\[
P^{-1} = \begin{bmatrix}
a & b \\
d & d
\end{bmatrix}^{-1} = \frac{1}{\det(P)} \begin{bmatrix}
d & -b \\
-c & a
\end{bmatrix}
\]
If $P^{-1}$ exists, then $|P| \neq 0$. Here,

\[
\begin{vmatrix}
7 & 8 \\
11 & 11
\end{vmatrix} = -11 \neq 0
\]
If $P^{-1}$ exists, then $|P| \neq 0$. Here,

$$\begin{vmatrix} 7 & 8 \\ 11 & 11 \end{vmatrix} = -11 \neq 0$$

So, we have that

$$P^{-1} = \frac{1}{-11} \begin{bmatrix} 11 & -8 \\ -11 & 7 \end{bmatrix} \pmod{26}$$
We need to rewrite modulo 26, which means no negatives and no fractions.
We need to rewrite modulo 26, which means no negatives and no fractions.
First, $-11 \equiv 15 \pmod{26}$, so we can replace $\frac{1}{-11}$ with $\frac{1}{15}$. Next, we need to represent $\frac{1}{15}$ as an integer.
We need to rewrite modulo 26, which means no negatives and no fractions.
First, $-11 \equiv 15 \pmod{26}$, so we can replace $\frac{1}{-11}$ with $\frac{1}{15}$. Next, we need to represent $\frac{1}{15}$ as an integer.
To do so, consider that $15 \cdot \frac{1}{15} \equiv 1 \pmod{26}$. So what we need is an integer such that the product of that integer and 15 is congruent to 1 modulo 26. Here, the integer we seek is 7.
Cracking the Hill Algorithm

At this point we now have

\[ P^{-1} = 7 \begin{bmatrix} 11 & -8 \\ -11 & 7 \end{bmatrix} \pmod{26} \]
At this point we now have

\[ P^{-1} = 7 \begin{bmatrix} 11 & -8 \\ -11 & 7 \end{bmatrix} \pmod{26} \]

We multiply through to get

\[ P^{-1} = \begin{bmatrix} 77 & -56 \\ -77 & 49 \end{bmatrix} \pmod{26} \]
At this point we now have

\[ P^{-1} = 7 \begin{bmatrix} 11 & -8 \\ -11 & 7 \end{bmatrix} \pmod{26} \]

We multiply through to get

\[ P^{-1} = \begin{bmatrix} 77 & -56 \\ -77 & 49 \end{bmatrix} \pmod{26} \]

And finally, when we take this modulo 26, we get

\[ \begin{bmatrix} 7 & 8 \\ 11 & 11 \end{bmatrix}^{-1} = \begin{bmatrix} 25 & 22 \\ 1 & 23 \end{bmatrix} \]
Cracking the Hill Algorithm

So,

\[
K = \begin{bmatrix}
25 & 22 \\
1 & 23
\end{bmatrix}
\begin{bmatrix}
7 & 2 \\
17 & 25
\end{bmatrix} = \begin{bmatrix}
549 & 600 \\
398 & 577
\end{bmatrix} \pmod{26}
\]
So,

\[
K = \begin{bmatrix} 25 & 22 \\ 1 & 23 \end{bmatrix} \begin{bmatrix} 7 & 2 \\ 17 & 25 \end{bmatrix} = \begin{bmatrix} 549 & 600 \\ 398 & 577 \end{bmatrix} \pmod{26}
\]

Which reduces to

\[
K = \begin{bmatrix} 3 & 2 \\ 8 & 5 \end{bmatrix}
\]
How many of you remember what a permutation is from abstract algebra?
How many of you remember what a permutation is from abstract algebra?

**Definition**

A permutation $\pi : S \rightarrow S$ such that $\pi$ is a bijection.
How many of you remember what a permutation is from abstract algebra?

**Definition**

A permutation $\pi : S \to S$ such that $\pi$ is a bijection.

We can use permutations to encrypt plaintext by arranging the letters in blocks of an appropriate length and then permuting within each one.
Example

If our plaintext is ‘lets go black and gold’, use the permutation $\pi = (13)(245)$ to encrypt.
Example

If our plaintext is ‘lets go black and gold’, use the permutation $\pi = (13)(245)$ to encrypt.

The length of the permutation tells us that we want to break the plaintext into blocks of length 5.

letsg  oblac  kandg  oldxx
Example

If our plaintext is ‘lets go black and gold’, use the permutation \( \pi = (13)(245) \) to encrypt.

The length of the permutation tells us that we want to break the plaintext into blocks of length 5.

\[
\text{letsg oblac kandg oldxx}
\]

Next, apply \( \pi \) to each block.

\[
\begin{align*}
\text{letsg oblac kandg oldxx} \\
\text{tgles lcoba ngkad dxolx}
\end{align*}
\]
Permutation Cipher Example

Example

If our plaintext is ‘lets go black and gold’, use the permutation \( \pi = (13)(245) \) to encrypt.

The length of the permutation tells us that we want to break the plaintext into blocks of length 5.

\[
\begin{align*}
\text{letsg} & \quad \text{oblac} & \quad \text{kandg} & \quad \text{oldxx} \\
\end{align*}
\]

Next, apply \( \pi \) to each block.

\[
\begin{align*}
\text{letsg} & \quad \text{oblac} & \quad \text{kandg} & \quad \text{oldxx} \\
\text{tgles} & \quad \text{lcoba} & \quad \text{ngkad} & \quad \text{dxolx} \\
\end{align*}
\]

Finally, we have

\( TGLESLOCOBANGKADDXOLX \)
Decryption of the Permutation Cipher

To decrypt, we need to find the inverse of the permutation and apply it to the cipher text.

Example

Find $\pi^{-1}$ for $\pi = (13)(245)$.

We reverse the cycles and invert the order.

$\pi^{-1} = (452)(31) = (452)(13)$. 
To decrypt, we need to find the inverse of the permutation and apply it to the cipher text.
Do we remember how to find the inverse of a permutation?

**Example**

Find $\pi^{-1}$ for $\pi = (13)(245)$. 

To decrypt, we need to find the inverse of the permutation and apply it to the cipher text. Do we remember how to find the inverse of a permutation?

**Example**

Find $\pi^{-1}$ for $\pi = (13)(245)$.

We reverse the cycles and invert the order.

$$\pi^{-1} = (452)(31) = (452)(13)$$
Attacks on the Permutation Cipher

- This cipher is easily beatable with a known plaintext attack
• This cipher is easily beatable with a known plaintext attack
• We could also use brute force
Attacks on the Permutation Cipher

- This cipher is easily beatable with a known plaintext attack
- We could also use brute force

**Definition**

A brute force attack involves trying all possible arrangements of the letters in a ciphertext to find the associated plaintext
Example

Decrypt

\[ TGLESLOBANGKADXXOLX \]

with only the knowledge that it was encrypted with a permutation cipher.
Example

Decrypt

\textit{TGLESLOCOBANGKADDXOLX}

with only the knowledge that it was encrypted with a permutation cipher.

Where do we start?
Example

Decrypt

$TGLESLOBANGKADDXOLX$

with only the knowledge that it was encrypted with a permutation cipher.

Where do we start?
Size of blocks must be divisor of 20.
Decryption with the Permutation Cipher

Example
 Decrypt

    TGLESLOCOBANGKADDXOLX

with only the knowledge that it was encrypted with a permutation cipher.

Where do we start?
Size of blocks must be divisor of 20.
Why can we rule out key size 2?
Decryption with the Permutation Cipher

Example

Decrypt

TGLESLOBANGKADDXOLX

with only the knowledge that it was encrypted with a permutation cipher.

Where do we start?
Size of blocks must be divisor of 20.
Why can we rule out key size 2?
No two letter words made up of T and G.
Next we try key length 4. Can we instantly rule this out?
Next we try key length 4. Can we instantly rule this out? No, since the first 4 letters are TGLE, which could give ‘GET L’
Next we try key length 4. Can we instantly rule this out? No, since the first 4 letters are TGLE, which could give ‘GET L’

\[
\begin{align*}
T & \quad G & \quad L & \quad E \\
S & \quad L & \quad C & \quad O \\
B & \quad A & \quad N & \quad G \\
K & \quad A & \quad D & \quad D \\
X & \quad O & \quad L & \quad X \end{align*}
\]
Next we try key length 4. Can we instantly rule this out? No, since the first 4 letters are TGLE, which could give ‘GET L’

\[
\begin{array}{cccc}
T & G & L & E \\
S & L & C & O \\
B & A & N & G \\
K & A & D & D \\
X & O & L & X \\
\end{array}
\]

Notice the last row and the X’s …
Next we try key length 4. Can we instantly rule this out?
No, since the first 4 letters are TGLE, which could give ‘GET L’

T  G  L  E
S  L  C  O
B  A  N  G
K  A  D  D
X  O  L  X

Notice the last row and the X’s ...
Possible last rows: OLXX and LOXX
Next we try key length 4. Can we instantly rule this out? No, since the first 4 letters are TGLE, which could give ‘GET L’

T   G   L   E
S   L   C   O
B   A   N   G
K   A   D   D
X   O   L   X

Notice the last row and the X’s ...
Possible last rows: OLXX and LOXX
Possible 4<sup>th</sup> rows: ADKD, ADDK, DAKD and DADK
Decryption with the Permutation Cipher

We look at the first 5 letters and we see that we have LETS G, so we cannot rule out 5 for a length. We again set up an array.

\[
\begin{array}{cccc}
T & G & L & E \\
L & C & O & B \\
N & G & K & A \\
D & X & O & L \\
\end{array}
\]

And if we rewrite with appropriate spacing, we'd have our plaintext.
Decryption with the Permutation Cipher

We look at the first 5 letters and we see that we have LETS G, so we cannot rule out 5 for a length. We again set up an array.

\[
\begin{align*}
T & \quad G & \quad L & \quad E & \quad S \\
L & \quad C & \quad O & \quad B & \quad A \\
N & \quad G & \quad K & \quad A & \quad D \\
D & \quad X & \quad O & \quad L & \quad X
\end{align*}
\]

If we begin with LETS G, we arrange the other rows using the same permutation.

\[
\begin{align*}
le & \quad t & \quad s & \quad g \\
ob & \quad l & \quad a & \quad c \\
k & \quad a & \quad n & \quad d & \quad g \\
o & \quad l & \quad d & \quad x & \quad x
\end{align*}
\]

And if we rewrite with appropriate spacing, we’d have our plaintext.
We again are using a permutation, and the initial encoding is not that unlike the permutation cipher. The difference is, instead of just permuting a block, we permute all of them simultaneously and then write the ciphertext by taking the columns in the same orientation as the permutation.

**Example**

Using the permutation \( \pi = (13)(24) \), encrypt the message

\[
\text{this is a sample plaintext}
\]
Encrypting with a Column Permutation Cipher

First we arrange the plaintext into an array with rows of length 4.

```
  t h i s
  i s a s
  a m p l
  e p l a
  i n t e
  x t x x
```

with padding at the end to make all rows the same length.
Then we permute based on $\pi$.

\[
\begin{array}{cccc}
3 & 4 & 1 & 2 \\
\text{t} & \text{h} & \text{i} & \text{s} \\
\text{i} & \text{s} & \text{a} & \text{s} \\
\text{a} & \text{m} & \text{p} & \text{l} \\
\text{e} & \text{p} & \text{l} & \text{a} \\
\text{i} & \text{n} & \text{t} & \text{e} \\
\text{x} & \text{t} & \text{x} & \text{x}
\end{array}
\]
Encrypting with a Column Permutation Cipher

Then we permute based on $\pi$.

$$\begin{array}{cccc}
3 & 4 & 1 & 2 \\
t & h & i & s \\
i & s & a & s \\
am & p & l \\
e & p & l & a \\
i & n & t & e \\
x & t & x & x \\
\end{array}$$

The corresponding ciphertext is

$IAPLTXSSLAEEXTIAEIXHSMPNT$
Here again, we know the key length must be a divisor of the number of characters in the ciphertext. In our last example, the length is 24, so we know there are 2, 3, 4, 6, 8, 12 or 24 columns.
Here again, we know the key length must be a divisor of the number of characters in the ciphertext. In our last example, the length is 24, so we know there are 2, 3, 4, 6, 8, 12 or 24 columns.

Here is a trick to guess this key length - 40% of letters in any stretch of English text are vowels. So we can use probability to help us. We can arrange into a number of columns and do a frequency analysis to see if it makes sense.
Decryption Example with a Column Permutation Cipher

Example

If we know the following Ciphertext was encrypted using a column Permutation cipher, decode the ciphertext.

\begin{center}
\text{WEDENODTURTKRHNSUKUXNSOSOIJQQR} \\
\text{HYGWGRHTTTEARATHEOEHEGIFISOAX}
\end{center}
Example

If we know the following Ciphertext was encrypted using a column Permutation cipher, decode the ciphertext.

\[
\begin{align*}
\text{WEDENODTURTKRHNSUKUXNSOSO} & \text{IJOQR} \\
\text{HYGWGRHTTTEARATHEOAEHE} & \text{GIFISOAX}
\end{align*}
\]

There are 60 characters here, so we know the number of columns is a divisor of 60.
If we know the following Ciphertext was encrypted using a column Permutation cipher, decode the ciphertext.

```
WEDENODTURTKRHNSUKUXNSOSOIJOQR
HYGWGRHITTEARATHEOAEHEGIFISOAX
```

There are 60 characters here, so we know the number of columns is a divisor of 60.
60 is a small number, so frequency analysis may be tough since there is such a small sample size.
The 20th and 60th letter are both X, so we may guess that those were null letters at the bottom of columns.

Good choices for the number of columns?

The number of rows is a divisor of 60 and a multiple of 10. That would leave us with 10, 20 or 30 rows.
Be Clever with Attacks

The 20th and 60th letter are both X, so we may guess that those were null letters at the bottom of columns.
The 20th and 60th letter are both X, so we may guess that those were null letters at the bottom of columns. Good choices for the number of columns?
The 20\textsuperscript{th} and 60\textsuperscript{th} letter are both X, so we may guess that those were null letters at the bottom of columns.

Good choices for the number of columns? The number of rows is a divisor of 60 and a multiple of 10. That would leave us with 10, 20 or 30 rows.
Decryption with a Column Permutation Cipher

So let’s start with 10, which would mean there are 6 columns.

W T N H E H
E K S Y A E
D R O G E G
E H S W A I
N N O G T F
O S I R H I
D U J H E S
T K O T O O
U U Q T A A
R X R T E X
Then, we reorder them with the two columns that end in X as the last two. From there, we want to permute the first four columns until we find an arrangement that makes sense.
Decryption with a Column Permutation Cipher

Then, we reorder them with the two columns that end in X as the last two. From there, we want to permute the first four columns until we find an arrangement that makes sense.

```
when they asked george washington for his id, he just took out a quarter.
```

```
whenth
eyeasked

dgrog

ewashih

ngtonf
orhisii
dhejus
ntooko
utraqua
rterxx
```
Decryption with a Column Permutation Cipher

Then, we reorder them with the two columns that end in X as the last two. From there, we want to permute the first four columns until we find an arrangement that makes sense.

```
when they asked george washington for his id, he just took out a quarter.
```