BILL
RECORD THIS
LECTURE
Gen Sub Cipher and Random-Looking Ciphers
General Substitution Cipher
The Problem with Shift and Affine

Shift and Affine both have small keyspaces.

Shift and Affine both use some math—hence math can be used against them.

We present the General Substitution Cipher which:

Has a large keyspace.

Does not use any math.
The Problem with Shift and Affine

- Shift and Affine both have small keyspaces.
The Problem with Shift and Affine

- Shift and Affine both have small keyspaces.
- Shift and Affine both use some math—hence math can be used against them.
The Problem with Shift and Affine

- Shift and Affine both have small keyspaces.
- Shift and Affine both use some math—hence math can be used against them.

We present the **General Substitution Cipher** which:
The Problem with Shift and Affine

- Shift and Affine both have small keyspaces.
- Shift and Affine both use some math—hence math can be used against them.

We present the **General Substitution Cipher** which:
- Has a large keyspace.
The Problem with Shift and Affine

- Shift and Affine both have small keyspaces.
- Shift and Affine both use some math—hence math can be used against them.

We present the **General Substitution Cipher** which:

- Has a large keyspace.
- Does not use any math.
Def Gen Sub Cipher with perm $f$ on $\{0, \ldots, 25\}$.

1. Encrypt via $x \rightarrow f(x)$.
2. Decrypt via $x \rightarrow f^{-1}(x)$. 
General Substitution Cipher: Example

Assume Alphabet is just \( \{a, \ldots, i\} \).

If the message is FBI it will encrypt to GIH.
General Substitution Cipher: Example

Assume Alphabet is just \{a, \ldots, i\}.
Encrypt Using:

\[
\begin{array}{cccccccc}
  a & b & c & d & e & f & g & h & i \\
  d & i & a & b & e & g & f & c & h \\
\end{array}
\]

If the message is FBI it will encrypt to GIH.
General Substitution Cipher: Example

Assume Alphabet is just \{a, \ldots, i\}.

Encrypt Using:

\[
\begin{array}{cccccccc}
 a & b & c & d & e & f & g & h & i \\
 d & i & a & b & e & g & f & c & h \\
\end{array}
\]

Decrypt Using:

\[
\begin{array}{cccccccc}
 a & b & c & d & e & f & g & h & i \\
 c & d & h & a & e & g & f & i & b \\
\end{array}
\]

If the message is FBI it will encrypt to GIH.
General Substitution Cipher: Example

Assume Alphabet is just \{a, \ldots, i\}.
Encrypt Using:

\[
\begin{array}{cccccccccc}
  a & b & c & d & e & f & g & h & i \\
  d & i & a & b & e & g & f & c & h \\
\end{array}
\]

Decrypt Using:

\[
\begin{array}{cccccccccc}
  a & b & c & d & e & f & g & h & i \\
  c & d & h & a & e & g & f & i & b \\
\end{array}
\]

If the message is **FBI** it will encrypt to **GIH**.
Theorem:  The Gen Sub Cipher is Uncrackable in reasonable time.

Proof: Eve sees a text \( T \). There are \( 26! \) possible permutations that could have been used. Eve has to look at all of them. This takes roughly \( 26! \) steps which is unreasonable.

End of Proof

Why is this proof incorrect? Discuss.

The proof assumes that Eve uses brute force. Our model of what Eve can do is too limited.

Okay, the proof is wrong, but is Gen Sub crackable? Yes. Eve can use Freq Analysis.
**Theorem:** The Gen Sub Cipher is Uncrackable in reasonable time.

**Proof:** Eve sees a text $T$. There are $26!$ possible permutations that could have been used. Eve has to look at all of them. This takes roughly $26!$ steps which is unreasonable.
The Gen Sub Cipher is Uncrackable (a False Proof)

**Theorem:** The Gen Sub Cipher is Uncrackable in reasonable time.

**Proof:** Eve sees a text $T$. There are 26! possible permutations that could have been used. Eve has to look at all of them. This takes roughly 26! steps which is unreasonable.

**End of Proof**

Why is this proof incorrect? Discuss.

The proof assumes that Eve uses brute force. Our model of what Eve can do is too limited.

Okay, the proof is wrong, but is Gen Sub crackable?

Yes. Eve can use Freq Analysis.
The Gen Sub Cipher is Uncrackable (a False Proof)

**Theorem:** The Gen Sub Cipher is Uncrackable in reasonable time.

**Proof:** Eve sees a text $T$. There are $26!$ possible permutations that could have been used. Eve has to look at all of them. This takes roughly $26!$ steps which is unreasonable.

**End of Proof**

Why is this proof incorrect? Discuss.
The Gen Sub Cipher is Uncrackable (a False Proof)

**Theorem:** The Gen Sub Cipher is Uncrackable in reasonable time.

**Proof:** Eve sees a text $T$. There are $26!$ possible permutations that could have been used. Eve has to look at all of them. This takes roughly $26!$ steps which is unreasonable.

**End of Proof**

Why is this proof incorrect? Discuss.

The proof assumes that Eve uses brute force. Our model of what Eve can do is too limited.
The Gen Sub Cipher is Uncrackable (a False Proof)

**Theorem:** The Gen Sub Cipher is Uncrackable in reasonable time.

**Proof:** Eve sees a text $T$. There are $26!$ possible permutations that could have been used. Eve has to look at all of them. This takes roughly $26!$ steps which is unreasonable.

**End of Proof**

Why is this proof incorrect? Discuss.

The proof assumes that Eve uses brute force. Our model of what Eve can do is too limited.

Okay, the proof is wrong, but is Gen Sub crackable?
The Gen Sub Cipher is Un Crackable (a False Proof)

Theorem: The Gen Sub Cipher is Un crackable in reasonable time.
Proof: Eve sees a text $T$. There are $26!$ possible permutations that could have been used. Eve has to look at all of them. This takes roughly $26!$ steps which is unreasonable.

End of Proof

Why is this proof incorrect? Discuss.

The proof assumes that Eve uses brute force. Our model of what Eve can do is too limited.

Okay, the proof is wrong, but is Gen Sub crackable?

Yes Eve can use Freq Analysis
Alice sends Bob a LONG text encrypted by Gen Sub Cipher. Eve finds freq of letters, pairs, triples, ....

Text in English.

1. Can use known freq: e is most common letter, th is most common pair.

2. Depending on topic may need to adjust frequencies. For example, if message is about the Mid East then q is more common (Iraq, Qatar).
Counter Example – Pangrams

Pangrams:
Sentence where each letter occurs at least once.

Short Pangrams ruin Freq analysis. Here are some:
1. The quick brown fox jumps over the lazy dog.
2. Pack my box with five dozen liquor jugs.
3. Amazingly few discotheques provide jukeboxes.
4. Watch Jeopardy! Alex Trebek's fun TV quiz game.

That should have been the ad slogan for watching Jeopardy. And now it can’t be :-(

Counter Example – Pangrams

**Pangrams**: Sentence where each letter occurs at least once.

1. The quick brown fox jumps over the lazy dog.
2. Pack my box with five dozen liquor jugs.
3. Amazingly few discotheques provide jukeboxes.
4. Watch Jeopardy! Alex Trebek’s fun TV quiz game.

That should have been the ad slogan for watching Jeopardy. And now it can't be :-(

Counter Example – Pangrams

**Pangrams:** Sentence where each letter occurs at least once.

Short Pangrams ruin Freq analysis. Here are some:

1. The quick brown fox jumps over the lazy dog.
2. Pack my box with five dozen liquor jugs.
3. Amazingly few discotheques provide jukeboxes.
4. Watch Jeopardy! Alex Trebek's fun TV quiz game. That should have been the ad slogan for watching Jeopardy. And now it can't be :-(.
Counter Example – Pangrams

**Pangrams**: Sentence where each letter occurs at least once.

Short Pangrams ruin Freq analysis. Here are some:

1. The quick brown fox jumps over the lazy dog.
Counter Example – Pangrams

**Pangrams:** Sentence where each letter occurs at least once.

Short Pangrams ruin Freq analysis. Here are some:

1. The quick brown fox jumps over the lazy dog.
2. Pack my box with five dozen liquor jugs.
Counter Example – Pangrams

**Pangrams:** Sentence where each letter occurs at least once.

Short Pangrams ruin Freq analysis. Here are some:

1. The quick brown fox jumps over the lazy dog.
2. Pack my box with five dozen liquor jugs.
3. Amazingly few discotheques provide jukeboxes.
Counter Example – Pangrams

**Pangrams:** Sentence where each letter occurs at least once.

Short Pangrams ruin Freq analysis. Here are some:

1. The quick brown fox jumps over the lazy dog.
2. Pack my box with five dozen liquor jugs.
3. Amazingly few discotheques provide jukeboxes.
4. Watch Jeopardy! Alex Trebek’s fun TV quiz game.
Counter Example – Pangrams

**Pangrams:** Sentence where each letter occurs at least once.

Short Pangrams ruin Freq analysis. Here are some:

1. The quick brown fox jumps over the lazy dog.
2. Pack my box with five dozen liquor jugs.
3. Amazingly few discotheques provide jukeboxes.
4. Watch Jeopardy! Alex Trebek’s fun TV quiz game. That should have been the ad slogan for watching Jeopardy. And now it can’t be :-(
Counter Example – Lipograms

Lipograms: A work that omits one letter.

1. Gadsby is a 50,000-word novel with no e’s in English. This inspired a French novel, A Void that also has no e’s.

2. Many book reviews of Gadsby and A Void used no e’s.

3. Eunoia is a 5-chapter novel, indexed by vowels. Chapter A only uses the vowel A, etc.

4. How I met your mother, Season 9, Episode 9: Lily and Robin challenge Barney to get a girl’s phone number without using the letter e. We are not going to deal with this silliness!

We assume long normal texts!
Counter Example – Lipograms

**Lipograms**: A work that omits one letter.

1. *Gadsby* is a 50,000-word novel with no *e*'s in English. This inspired a French novel, *A Void* that also has no *e*'s.

2. Many book reviews of *Gadsby* and *A Void* used no *e*'s.

3. *Eunoia* is a 5-chapter novel, indexed by vowels. Chapter A only uses the vowel A, etc.

4. *How I met your mother*, Season 9, Episode 9: Lily and Robin challenge Barney to get a girl's phone number without using the letter *e*.

We are not going to deal with this silliness!

We assume long normal texts!
Counter Example – Lipograms

Lipograms: A work that omits one letter.

1. **Gadsby** is a 50,000-word novel with no e’s in English. This inspired a French novel, **A Void** that also has no e’s.
Counter Example – Lipograms

**Lipograms:** A work that omits one letter.

1. **Gadsby** is a 50,000-word novel with no e’s in English. This inspired a French novel, **A Void** that also has no e’s.
2. Many book reviews of **Gadsby** and **A Void** used no e’s.
Counter Example – Lipograms

Lipograms: A work that omits one letter.

1. **Gadsby** is a 50,000-word novel with no e’s in English. This inspired a French novel, **A Void** that also has no e’s.
2. Many book reviews of **Gadsby** and **A Void** used no e’s.
3. **Eunoia** is a 5-chapter novel, indexed by vowels. Chapter A only use the vowel A, etc.
Lipograms: A work that omits one letter.

1. **Gadsby** is a 50,000-word novel with no e’s in English. This inspired a French novel, **A Void** that also has no e’s.
2. Many book reviews of **Gadsby** and **A Void** used no e’s.
3. **Eunoia** is a 5-chapter novel, indexed by vowels. Chapter A only use the vowel A, etc.
4. **How I met your mother, Season 9, Episode 9**: Lily and Robin challenge Barney to get a girl’s phone number without using the letter e.
Counter Example – Lipograms

**Lipograms:** A work that omits one letter.

1. **Gadsby** is a 50,000-word novel with no e’s in English. This inspired a French novel, **A Void** that also has no e’s.
2. Many book reviews of **Gadsby** and **A Void** used no e’s.
3. **Eunoia** is a 5-chapter novel, indexed by vowels. Chapter A only use the vowel A, etc.
4. **How I met your mother, Season 9, Episode 9:** Lily and Robin challenge Barney to get a girl’s phone number without using the letter e.

**We are not going to deal with this sillyness!**
Lipograms: A work that omits one letter.

1. **Gadsby** is a 50,000-word novel with no e’s in English. This inspired a French novel, **A Void** that also has no e’s.
2. Many book reviews of **Gadsby** and **A Void** used no e’s.
3. **Eunoia** is a 5-chapter novel, indexed by vowels. Chapter A only use the vowel A, etc.
4. **How I met your mother, Season 9, Episode 9**: Lily and Robin challenge Barney to get a girl’s phone number without using the letter e.

*We are not going to deal with this sillyness!*

*We assume long normal texts!*
“Just Use Freq Analysis”- Yeah, Right

All of the textbooks I have looked at say

The Gen Sub Cipher is crackable using Freq Analysis
“Just Use Freq Analysis”- Yeah, Right

All of the textbooks I have looked at say

The Gen Sub Cipher is crackable using Freq Analysis

But they do not actually say quite how to really do that.
All of the textbooks I have looked at say

The Gen Sub Cipher is crackable using Freq Analysis

But they do not actually say quite how to really do that.

1. They can’t tell me– its classified.
“Just Use Freq Analysis”- Yeah, Right

All of the textbooks I have looked at say

The Gen Sub Cipher is crackable using Freq Analysis

But they do not actually say quite how to really do that.

1. They can’t tell me– its classified. Unlikely.
All of the textbooks I have looked at say

The Gen Sub Cipher is crackable using Freq Analysis

But they do not actually say quite how to really do that.

1. They can’t tell me– its classified. Unlikely.
2. It’s complicated so people haven’t bothered writing it down.
"Just Use Freq Analysis"- Yeah, Right

All of the textbooks I have looked at say
The Gen Sub Cipher is crackable using Freq Analysis
But they do not actually say quite how to really do that.

1. They can’t tell me– its classified. Unlikely.
2. It’s complicated so people haven’t bothered writing it down. Likely.
All of the textbooks I have looked at say

The Gen Sub Cipher is crackable using Freq Analysis

But they do not actually say quite how to really do that.

1. They can’t tell me— its classified. Unlikely.

2. It’s complicated so people haven’t bothered writing it down. Likely.

3. In Summer 2019 I had a student, David Zhen, work on cracking gen sub cipher. We will later present what he did.
“Just Use Freq Analysis”- Yeah, Right

All of the textbooks I have looked at say

The Gen Sub Cipher is crackable using Freq Analysis

But they do not actually say quite how to really do that.

1. They can’t tell me– its classified. Unlikely.
2. It’s complicated so people haven’t bothered writing it down. Likely.
3. In Summer 2019 I had a student, David Zhen, work on cracking gen sub cipher. We will later present what he did.
4. Spoiler Alert:
All of the textbooks I have looked at say

The Gen Sub Cipher is crackable using Freq Analysis

But they do not actually say quite how to really do that.

1. They can’t tell me— its classified. Unlikely.
2. It’s complicated so people haven’t bothered writing it down. Likely.
3. In Summer 2019 I had a student, David Zhen, work on cracking gen sub cipher. We will later present what he did.
4. Spoiler Alert: 
   David Zhen has a program that cracks the gen sub cipher.
Random-Looking Ciphers
Alternatives to Gen Sub (History)

**In the Year 2020** Alice can easily generate a *random* permutation of \{a, \ldots, z\} and send it to Bob. Key length is not a problem.
In the Year 2020 Alice can easily generate a random permutation of \{a, \ldots, z\} and send it to Bob. Key length is not a problem.

In the Year 1020 it was hard for Alice to generate a random perm and impossible to give it a short description. Hence she generates a random-looking permutation of \{a, \ldots, z\} with a short key.
In the Year 2020 Alice can easily generate a random permutation of \{a, \ldots, z\} and send it to Bob. Key length is not a problem.

In the Year 1020 it was hard for Alice to generate a random perm and impossible to give it a short description. Hence she generates a random-looking permutation of \{a, \ldots, z\} with a short key.

1. We show one such methods.
In the Year 2020 Alice can easily generate a random permutation of \{a, \ldots, z\} and send it to Bob. Key length is not a problem.

In the Year 1020 it was hard for Alice to generate a random perm and impossible to give it a short description. Hence she generates a random-looking permutation of \{a, \ldots, z\} with a short key.

1. We show one such methods.
2. These methods are primitive examples of psuedo-random generators which take a short string and make a random-looking much longer string. These are important in crypto. We will encounter them again.
Keyword-Shift Cipher. Key is (Phrase, Shift)

\[ \Sigma = \{a, \ldots, k\}. \textbf{Key:} \ (\text{jack}, \ 4). \]
Keyword-Shift Cipher. Key is (Phrase, Shift)

\[ \Sigma = \{a, \ldots, k\} \]. \textbf{Key:} (jack, 4).

Alice then does the following:
Keyword-Shift Cipher. Key is (Phrase, Shift)

$\Sigma = \{a, \ldots, k\}$. **Key:** (jack, 4).

Alice then does the following:

1. List out the key word and then the remaining letters:

   $$
   \begin{array}{cccccccccc}
   j & a & c & k & b & d & e & f & g & h & i \\
   \end{array}
   $$
Keyword-Shift Cipher. Key is (Phrase, Shift)

$\Sigma = \{a, \ldots, k\}$. **Key:** (jack, 4).

Alice then does the following:

1. List out the key word and then the remaining letters:

   $\begin{array}{cccccccccc}
   j & a & c & k & b & d & e & f & g & h & i \\
   \end{array}$

2. Now do Shift 4 on this:

   $\begin{array}{cccccccccc}
   f & g & h & i & j & a & c & k & b & d & e \\
   \end{array}$

This is where $a, b, c, \ldots$ go, so:

$\begin{array}{cccccccccc}
   a & b & c & d & e & f & g & h & i & j & k \\
   f & g & h & i & j & a & c & k & b & d & e \\
   f & g & h & i & j & a & c & k & b & d & e \\
\end{array}$
Keyword-Shift Cipher. Key is (Phrase, Shift) (cont)

To encrypt use:

```
  a b c d e f g h i j k
  f g h i j a c k b d e
```
Keyword-Shift Cipher. Key is (Phrase, Shift) (cont)

To encrypt use:

```
<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>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>a</td>
<td>c</td>
<td>k</td>
<td>b</td>
<td>d</td>
<td>e</td>
</tr>
</tbody>
</table>
```

To decrypt you invert the table:

```
<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>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>i</td>
<td>g</td>
<td>j</td>
<td>k</td>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>h</td>
</tr>
</tbody>
</table>
```
From (jack,4) (which is short) we got

\[
\begin{array}{cccccccccc}
 a & b & c & d & e & f & g & h & i & j & k \\
 f & g & h & i & j & a & c & k & b & d & e \\
\end{array}
\]

Does this cipher look like it was generated randomly? Discuss.

1. No - Note the f-g-h-i-j all in order.
2. The f-g-h-i-j is not an accident. The keyword-Shift cipher tends to have streaks like that.
3. With 4-letter keywords, prob of 5-in-a-row is large.
4. Truly random perm, prob of 5-in-a-row is small.
5. With 4-letter keywords, not that rand looking.
From (jack,4) (which is short) we got

<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>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>a</td>
<td>c</td>
<td>k</td>
<td>b</td>
<td>d</td>
<td>e</td>
</tr>
</tbody>
</table>

Does this cipher look like it was generated randomly? Discuss.
From Short Key Got Rand-Looking Perm(?)

From (jack,4) (which is short) we got

<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>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>a</td>
<td>c</td>
<td>k</td>
<td>b</td>
<td>d</td>
<td>e</td>
</tr>
</tbody>
</table>

Does this cipher look like it was generated randomly? Discuss.

1. No- Note the f-g-h-i-j all in order.
From (jack,4) (which is short) we got

<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>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>a</td>
<td>c</td>
<td>k</td>
<td>b</td>
<td>d</td>
<td>e</td>
</tr>
</tbody>
</table>

Does this cipher look like it was generated randomly? Discuss.

1. No- Note the f-g-h-i-j all in order.
2. The f-g-h-i-j is not an accident. The keyword-Shift cipher tends to have streaks like that.
From Short Key Got Rand-Looking Perm(?)

From (jack,4) (which is short) we got

\[
\begin{array}{cccccccc}
  a & b & c & d & e & f & g & h & i & j & k \\
  f & g & h & i & j & a & c & k & b & d & e \\
\end{array}
\]

Does this cipher look like it was generated randomly? Discuss.

1. No- Note the f-g-h-i-j all in order.
2. The f-g-h-i-j is not an accident. The keyword-Shift cipher tends to have streaks like that.
3. With 4-let keywords, prob of 5-in-a-row is large.
From Short Key Got Rand-Looking Perm(?)

From (jack,4) (which is short) we got

\[
\begin{array}{cccccccc}
    a & b & c & d & e & f & g & h & i & j & k \\
    f & g & h & i & j & a & c & k & b & d & e \\
\end{array}
\]

Does this cipher look like it was generated randomly? Discuss.

1. No- Note the f-g-h-i-j all in order.
2. The f-g-h-i-j is not an accident. The keyword-Shift cipher tends to have streaks like that.
3. With 4-let keywords, prob of 5-in-a-row is large.
4. Truly random perm, prob of 5-in-a-row is small.
From Short Key Got Rand-Looking Perm(?)

From (jack,4) (which is short) we got

<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>
</tr>
</thead>
<tbody>
<tr>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>a</td>
<td>c</td>
<td>k</td>
<td>b</td>
<td>d</td>
<td>e</td>
</tr>
</tbody>
</table>

Does this cipher look like it was generated randomly? Discuss.

1. No- Note the f-g-h-i-j all in order.
2. The f-g-h-i-j is not an accident. The keyword-Shift cipher tends to have streaks like that.
3. With 4-let keywords, prob of 5-in-a-row is **large**.
4. Truly random perm, prob of 5-in-a-row is **small**.
5. With 4-let keywords, not that rand looking.
What about Longer Keywords?

Longer keywords would help
What about Longer Keywords?

Longer keywords would help
We assume normal 26 letter alphabet.
What about Longer Keywords?

Longer keywords would help
We assume normal 26 letter alphabet.

If you use

Garey and Johnson: A guide to NP-completeness
with shift 4.
What about Longer Keywords?

Longer keywords would help
We assume normal 26 letter alphabet.

If you use

Garey and Johnson: A guide to NP-completeness
with shift 4.

First eliminate spaces and repeats:

`gareyndjohsuitpcml` (18 letters)
What about Longer Keywords?

Longer keywords would help
We assume normal 26 letter alphabet.

If you use

**Garey and Johnson: A guide to NP-completeness**

with shift 4.

First eliminate spaces and repeats:

```
gareyndjohsuitpcml
```

(18 letters)

I leave the rest to you. Find the encode and decode tables and see if they look random.