BILL
RECORD THIS
LECTURE

September 20, 2020

Gen Sub Cipher and Random-Looking Ciphers

September 20, 2020

General Substitution Cipher

September 20, 2020

The Problem with Shift and Affine

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 Substation 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 Substation 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 Substation Cipher** which:

- Has a large keyspace.
- Does not use any math.

General Substitution Cipher

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 $\{a, \ldots, i\}$.

General Substitution Cipher: Example

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

a	b	c	d	e	f	g	h	i
d	i	a	b	e	g	f	c	h

General Substitution Cipher: Example

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

a	b	c	d	e	f	g	h	i
d	i	a	b	e	g	f	c	h

Decrypt Using:

a	b	c	d	e	f	g	h	i
c	d	h	a	e	g	f	i	b

General Substitution Cipher: Example

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

a	b	c	d	e	f	g	h	i
d	i	a	b	e	g	f	c	h

Decrypt Using:

a	b	c	d	e	f	g	h	i
c	d	h	a	e	g	f	i	b

If the message is **FBI** it will encrypt to **GIH**.

The Gen Sub Cipher is Uncrackable (a False Proof)

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

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.

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

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.

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

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

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. If Alice is telling Bob about Mid East Politics than may need to adjust: *q* is more common (Iraq, Qatar) and some words more common.

Counter Example – Pangrams

Counter Example – Pangrams

Pangrams: Sentence where each letter occurs at least once.

Counter Example – Pangrams

Pangrams: Sentence where each letter occurs at least once.

Short Pangrams ruin Freq analysis. Here are some:

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 – Lipograms

Counter Example – Lipograms

Lipograms: A work that omits one letter.

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 Review 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 Review 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.

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 Review 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 Review 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!

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 Review 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**.

“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.

“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.

“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.

"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.

“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. I had a summer student, David Zhen, work on this over the summer and will be presenting what we came up with later.

“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. I had a summer student, David Zhen, work on this over the summer and will be presenting what we came up with later.
4. Spoiler Alert:

“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. I had a summer student, David Zhen, work on this over the summer and will be presenting what we came up with later.
4. Spoiler Alert: David Zhen has a program that cracks the gen sub cipher.

Random-Looking Ciphers

September 20, 2020

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.

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 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.

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 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 two such methods.

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 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 two 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 (Word,Shift)

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

Keyword-Shift Cipher. Key is (Word,Shift)

$\Sigma = \{a, \ldots, k\}$. **Key:** (jack, 4).
Alice then does the following:

Keyword-Shift Cipher. Key is (Word,Shift)

$\Sigma = \{a, \ldots, k\}$. **Key:** (jack, 4).
Alice then does the following:

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

j	a	c	k	b	d	e	f	g	h	i

Keyword-Shift Cipher. Key is (Word,Shift)

$\Sigma = \{a, \ldots, k\}$. **Key:** (jack, 4).
Alice then does the following:

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

j	a	c	k	b	d	e	f	g	h	i

2. Now do Shift 4 on this:

f	g	h	i	j	a	c	k	b	d	e

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

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 (Word,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 (Word,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

To decrypt you invert the table:

a	b	c	d	e	f	g	h	i	j	k
f	i	g	j	k	a	b	c	d	e	h

From Short Key Got Rand-Looking Perm(?)

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

a	b	c	d	e	f	g	h	i	j	k
f	g	h	i	j	a	c	k	b	d	e

From Short Key Got Rand-Looking Perm(?)

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

a	b	c	d	e	f	g	h	i	j	k
f	g	h	i	j	a	c	k	b	d	e

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

a	b	c	d	e	f	g	h	i	j	k
f	g	h	i	j	a	c	k	b	d	e

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

1. No- Note the f-g-h-i-j all in order.

From Short Key Got Rand-Looking Perm(?)

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

a	b	c	d	e	f	g	h	i	j	k
f	g	h	i	j	a	c	k	b	d	e

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

a	b	c	d	e	f	g	h	i	j	k
f	g	h	i	j	a	c	k	b	d	e

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. Keyword-Shift Cipher, 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

a	b	c	d	e	f	g	h	i	j	k
f	g	h	i	j	a	c	k	b	d	e

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. Keyword-Shift Cipher, 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

a	b	c	d	e	f	g	h	i	j	k
f	g	h	i	j	a	c	k	b	d	e

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. Keyword-Shift Cipher, 4-let keywords, prob of 5-in-a-row is **large.**
4. Truly random perm, prob of 5-in-a-row is **small.**
5. Keyword-Shift Cipher, 4-let keywords, not that rand looking.

What about Longer Keywords?

Longer keywords would help

What about Longer Keywords?

Longer keywords would help

If you use

Garey and Johnson: A guide to NP-completeness

and eliminate repeats, get:

gareyndjohsuitpcml

What about Longer Keywords?

Longer keywords would help

If you use

Garey and Johnson: A guide to NP-completeness

and eliminate repeats, get:

gareyndjohsuitpcml

I suspect this would not leave a tell-tale sign of not being random.

Keyword-Mixed Cipher. $\Sigma = \{a, \ldots, k\}$. Key:jack

Keyword-Mixed Cipher. $\Sigma = \{a, \ldots, k\}$. Key:jack

1. Write **jack**. Under it write the rest of Σ in blocks of size $|$**jack**$|$.

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

Keyword-Mixed Cipher. $\Sigma = \{a, \ldots, k\}$. Key:jack

1. Write **jack**. Under it write the rest of Σ in blocks of size $|$**jack**$|$.

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

2. Write down these letters column by column:

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

Keyword-Mixed Cipher. $\Sigma = \{a, \ldots, k\}$. Key:jack

1. Write **jack**. Under it write the rest of Σ in blocks of size $|$**jack**$|$.

j	a	c	k
b	d	e	f
g	h	i	

2. Write down these letters column by column:

j	b	g	a	d	h	c	e	i	k	f

3. Put the letters in order under it:

a	b	c	d	e	f	g	h	i	j	k
j	b	g	a	d	h	c	e	i	k	f

Keyword-Shift vs Keyword-Mixed

Both Keyword-Shift, Keyword-Mixed take a short seed and produce a **Random-Looking** permutation. Which one is better?

Keyword-Shift vs Keyword-Mixed

Both Keyword-Shift, Keyword-Mixed take a short seed and produce a **Random-Looking** permutation. Which one is better?

We won't answer that question, but we will show how to ask it.

Keyword-Shift vs Keyword-Mixed

Both Keyword-Shift, Keyword-Mixed take a short seed and produce a **Random-Looking** permutation. Which one is better?

We won't answer that question, but we will show how to ask it.

We will use a **game**!

Keyword-Shift vs Keyword-Mixed

Both Keyword-Shift, Keyword-Mixed take a short seed and produce a **Random-Looking** permutation. Which one is better?

We won't answer that question, but we will show how to ask it.

We will use a **game**!

My wife say that when mathematicians use the word **game**, such games are not fun.

Keyword-Shift vs Keyword-Mixed

Both Keyword-Shift, Keyword-Mixed take a short seed and produce a **Random-Looking** permutation. Which one is better?

We won't answer that question, but we will show how to ask it.

We will use a **game**!

My wife say that when mathematicians use the word **game**, such games are not fun.

Today's lecture will **support** her viewpoint.

Keyword-Shift vs Truly Random

Alice and Eve play the following game:

Keyword-Shift vs Truly Random

Alice and Eve play the following game:
Game: $\Sigma = \{a, b, \ldots, z\}$. L is length of keyword, $L = 6$.

Keyword-Shift vs Truly Random

Alice and Eve play the following game:

Game: $\Sigma = \{a, b, \ldots, z\}$. L is length of keyword, $L = 6$.

1. Alice flips a fair coin.

Keyword-Shift vs Truly Random

Alice and Eve play the following game:

Game: $\Sigma = \{a, b, \ldots, z\}$. L is length of keyword, $L = 6$.

1. Alice flips a fair coin.

   If T then Alice gen rand perm of Σ and sends to Eve.

Keyword-Shift vs Truly Random

Alice and Eve play the following game:

Game: $\Sigma = \{a, b, \ldots, z\}$. L is length of keyword, $L = 6$.

1. Alice flips a fair coin.

   If T then Alice gen rand perm of Σ and sends to Eve.
   If H then Alice gen rand word $w \in \Sigma^6$, with 6 **diff** letters, rand $s \in \mathbb{Z}_{25}$, creates a perm using Keyword-Shift with w, s, and sends to Eve.

Keyword-Shift vs Truly Random

Alice and Eve play the following game:

Game: $\Sigma = \{a, b, \ldots, z\}$. L is length of keyword, $L = 6$.

1. Alice flips a fair coin.

   If T then Alice gen rand perm of Σ and sends to Eve.
   If H then Alice gen rand word $w \in \Sigma^6$, with 6 **diff** letters, rand $s \in \mathbb{Z}_{25}$, creates a perm using Keyword-Shift with w, s, and sends to Eve.

2. Eve says RP (Rand Perm) if she thinks Alice flipped T, KS (Keyword-Shift) if she thinks Alice flipped H. If Eve is correct she wins! If not then Alice wins!

Alice has no strategy in this game.

Keyword-Shift vs Truly Random

Alice and Eve play the following game:

Game: $\Sigma = \{a, b, \ldots, z\}$. L is length of keyword, $L = 6$.

1. Alice flips a fair coin.

   If T then Alice gen rand perm of Σ and sends to Eve.
   If H then Alice gen rand word $w \in \Sigma^6$, with 6 **diff** letters, rand $s \in \mathbb{Z}_{25}$, creates a perm using Keyword-Shift with w, s, and sends to Eve.

2. Eve says RP (Rand Perm) if she thinks Alice flipped T, KS (Keyword-Shift) if she thinks Alice flipped H. If Eve is correct she wins! If not then Alice wins!

Alice has no strategy in this game.
Eve can have a strategy.

Keyword-Shift vs Truly Random

Alice and Eve play the following game:

Game: $\Sigma = \{a, b, \ldots, z\}$. L is length of keyword, $L = 6$.

1. Alice flips a fair coin.

   If T then Alice gen rand perm of Σ and sends to Eve.
   If H then Alice gen rand word $w \in \Sigma^6$, with 6 **diff** letters, rand $s \in \mathbb{Z}_{25}$, creates a perm using Keyword-Shift with w, s, and sends to Eve.

2. Eve says RP (Rand Perm) if she thinks Alice flipped T, KS (Keyword-Shift) if she thinks Alice flipped H. If Eve is correct she wins! If not then Alice wins!

Alice has no strategy in this game.
Eve can have a strategy.
We measure how good the Keyword-Shift is by the probability that an optimal Eve can win.

Keyword-Shift vs Truly Random

Alice and Eve play the following game:

Game: $\Sigma = \{a, b, \ldots, z\}$. L is length of keyword, $L = 6$.

1. Alice flips a fair coin.

   If T then Alice gen rand perm of Σ and sends to Eve.
   If H then Alice gen rand word $w \in \Sigma^6$, with 6 **diff** letters, rand $s \in \mathbb{Z}_{25}$, creates a perm using Keyword-Shift with w, s, and sends to Eve.

2. Eve says RP (Rand Perm) if she thinks Alice flipped T, KS (Keyword-Shift) if she thinks Alice flipped H. If Eve is correct she wins! If not then Alice wins!

Alice has no strategy in this game.

Eve can have a strategy.

We measure how good the Keyword-Shift is by the probability that an optimal Eve can win.

How well can Eve do? Discuss.

Game Needs Clarification

We have not specified how powerful Eve is. Two options:

Game Needs Clarification

We have not specified how powerful Eve is. Two options:

1. Eve has unlimited computational power. She is only limited by how much information she has.

Game Needs Clarification

We have not specified how powerful Eve is. Two options:

1. Eve has unlimited computational power. She is only limited by how much information she has.
2. Eve is limited computationally. We will clarify later.

Unlimited Eve Strategy

Assume Eve has unlimited computational power.

Unlimited Eve Strategy

Assume Eve has unlimited computational power.
Before Eve plays the game she does the following:

Unlimited Eve Strategy

Assume Eve has unlimited computational power.
Before Eve plays the game she does the following:

- For every word $w \in \Sigma^6$ (all diff letters) and shift $s \in \{0, \ldots, 25\}$ find the perm generated by keyword-Shift.

Unlimited Eve Strategy

Assume Eve has unlimited computational power.
Before Eve plays the game she does the following:

- For every word $w \in \Sigma^6$ (all diff letters) and shift $s \in \{0, \ldots, 25\}$ find the perm generated by keyword-Shift.
- Store all $L = 26 \times 25 \times 24 \times 23 \times 22 \times 21 \times 26$ perms: $\sigma_1, \sigma_2, \ldots, \sigma_L$.

Unlimited Eve Strategy

Assume Eve has unlimited computational power.
Before Eve plays the game she does the following:

- For every word $w \in \Sigma^6$ (all diff letters) and shift $s \in \{0, \ldots, 25\}$ find the perm generated by keyword-Shift.
- Store all $L = 26 \times 25 \times 24 \times 23 \times 22 \times 21 \times 26$ perms: $\sigma_1, \sigma_2, \ldots, \sigma_L$.
- Note that the number of perms is $\sim 10^9$.

Unlimited Eve Strategy

Assume Eve has unlimited computational power.
Before Eve plays the game she does the following:

- For every word $w \in \Sigma^6$ (all diff letters) and shift $s \in \{0, \ldots, 25\}$ find the perm generated by keyword-Shift.
- Store all $L = 26 \times 25 \times 24 \times 23 \times 22 \times 21 \times 26$ perms: $\sigma_1, \sigma_2, \ldots, \sigma_L$.
- Note that the number of perms is $\sim 10^9$.
- Note that $26! \sim 10^{26}$.

Unlimited Eve Strategy

Assume Eve has unlimited computational power.
Before Eve plays the game she does the following:

- For every word $w \in \Sigma^6$ (all diff letters) and shift $s \in \{0,\ldots,25\}$ find the perm generated by keyword-Shift.
- Store all $L = 26 \times 25 \times 24 \times 23 \times 22 \times 21 \times 26$ perms: $\sigma_1, \sigma_2, \ldots, \sigma_L$.
- Note that the number of perms is $\sim 10^9$.
- Note that $26! \sim 10^{26}$.

Eve's strategy:

Unlimited Eve Strategy

Assume Eve has unlimited computational power.
Before Eve plays the game she does the following:

- For every word $w \in \Sigma^6$ (all diff letters) and shift $s \in \{0, \ldots, 25\}$ find the perm generated by keyword-Shift.
- Store all $L = 26 \times 25 \times 24 \times 23 \times 22 \times 21 \times 26$ perms: $\sigma_1, \sigma_2, \ldots, \sigma_L$.
- Note that the number of perms is $\sim 10^9$.
- Note that $26! \sim 10^{26}$.

Eve's strategy:
Alice gives Eve perm τ. If τ is one of the σ_i then Eve says KS, otherwise Eve says RP.

Unlimited Eve Analysis

Unlimited Eve Analysis

- If KS then Eve will guess it correctly.

Unlimited Eve Analysis

- If KS then Eve will guess it correctly.
- If RP then the prob Eve gets it wrong is the prob that perm just happens to be one of the σ_i:

$$\sim \frac{10^9}{10^{26}} = \frac{1}{10^{17}}$$

Unlimited Eve Analysis

- If KS then Eve will guess it correctly.
- If RP then the prob Eve gets it wrong is the prob that perm just happens to be one of the σ_i:

$$\sim \frac{10^9}{10^{26}} = \frac{1}{10^{17}}$$

Prob Eve right is $1 - \frac{1}{10^{17}} = 0.99999999999999999 = L$.

Unlimited Eve Analysis

- If KS then Eve will guess it correctly.
- If RP then the prob Eve gets it wrong is the prob that perm just happens to be one of the σ_i:

$$\sim \frac{10^9}{10^{26}} = \frac{1}{10^{17}}$$

Prob Eve right is $1 - \frac{1}{10^{17}} = 0.99999999999999999 = L$.

Prob Eve wins is
$\Pr(KS) \times 1 + \Pr(RP) \times L = \frac{1}{2} \times 1 + \frac{1}{2} \times L = \frac{1}{2}(1 + L) = L'$
which is very close to 1.
Upshot Unlimited Eve wins most of the time.

Comp Limited Eve

How much do we want to limit Eve? We want that she cannot look at all the perms.

Comp Limited Eve

How much do we want to limit Eve? We want that she cannot look at all the perms.

Number of perms is $|\Sigma|! \sim (\frac{|\Sigma|}{e})^{|\Sigma|}$.

Eve is **Comp Limited** if she only has $|\Sigma|^a$ time for some $a \in \mathbb{N}$.

Comp Limited Eve

How much do we want to limit Eve? We want that she cannot look at all the perms.

Number of perms is $|\Sigma|! \sim (\frac{|\Sigma|}{e})^{|\Sigma|}$.

Eve is **Comp Limited** if she only has $|\Sigma|^a$ time for some $a \in \mathbb{N}$.

This is Poly vs Exp, like P vs NP.

Comp Limited Eve

How much do we want to limit Eve? We want that she cannot look at all the perms.

Number of perms is $|\Sigma|! \sim (\frac{|\Sigma|}{e})^{|\Sigma|}$.

Eve is **Comp Limited** if she only has $|\Sigma|^a$ time for some $a \in \mathbb{N}$.

This is Poly vs Exp, like P vs NP.

Poly time is a way of saying **NOT exhaustive search**

Comp Limited Eve

How much do we want to limit Eve? We want that she cannot look at all the perms.

Number of perms is $|\Sigma|! \sim (\frac{|\Sigma|}{e})^{|\Sigma|}$.

Eve is **Comp Limited** if she only has $|\Sigma|^a$ time for some $a \in \mathbb{N}$.

This is Poly vs Exp, like P vs NP.

Poly time is a way of saying **NOT exhaustive search**

Note this is not really rigorous since we are thinking of $|\Sigma|$ as 26, but the idea is sound.

Strategy for Comp Limited Eve (Motivation)

We will do a few more Keyword-Shift Ciphers and see if there are hints that they **are** Keyword-Shift.

Recall Keyword-Shift Cipher with (jack,4)

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

Recall Keyword-Shift Cipher with (jack,4)

$\Sigma = \{a, \ldots, k\}$. **Key:** (jack,4).
Alice then does the following:

Recall Keyword-Shift Cipher with (jack,4)

$\Sigma = \{a, \ldots, k\}$. **Key:** (jack,4).
Alice then does the following:

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

j	a	c	k	b	d	e	f	g	h	i

Recall Keyword-Shift Cipher with (jack,4)

$\Sigma = \{a, \ldots, k\}$. **Key:** (jack,4).
Alice then does the following:

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

j	a	c	k	b	d	e	f	g	h	i

2. Now do Shift 4 on this:

f	g	h	i	j	a	c	k	b	d	e

Recall Keyword-Shift Cipher with (jack,4)

$\Sigma = \{a, \ldots, k\}$. **Key:** (jack,4).
Alice then does the following:

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

j	a	c	k	b	d	e	f	g	h	i

2. Now do Shift 4 on this:

f	g	h	i	j	a	c	k	b	d	e

3. This is where $a, b, c, \ldots,$ goes to. Put the table in order to get how to encode.

a	b	c	d	e	f	g	h	i	j	k
f	**g**	**h**	i	j	a	c	k	b	d	e

Much of the bottom row is in alpha-order.

Keyword-Shift Cipher with (fbia,1)

$\Sigma = \{a, \ldots, k\}$. **Key:** (fbia,1).

Keyword-Shift Cipher with (fbia,1)

$\Sigma = \{a, \ldots, k\}$. **Key:** (fbia,1).

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

f	b	i	a	c	d	e	g	h	j	k

Keyword-Shift Cipher with (fbia,1)

$\Sigma = \{a, \ldots, k\}$. **Key:** (fbia,1).

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

f	b	i	a	c	d	e	g	h	j	k

2. Now do Shift 1 on this:

g	c	j	b	d	e	f	h	i	k	a

Keyword-Shift Cipher with (fbia,1)

$\Sigma = \{a, \ldots, k\}$. **Key:** (fbia,1).

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

f	b	i	a	c	d	e	g	h	j	k

2. Now do Shift 1 on this:

g	c	j	b	d	e	f	h	i	k	a

3. This is where $a, b, c, \ldots,$ goes to. Put the table in order to get how to encode.

a	b	c	d	e	f	g	h	i	j	k
g	c	j	b	**d**	**e**	**f**	h	i	k	a

Again we get three letters in a row!

Strategy for Comp Limited Eve

Strategy for Comp Limited Eve

1. Eve gets τ.

Strategy for Comp Limited Eve

1. Eve gets τ.
2. If τ has 3 consecutive letters (e.g., p, q, r) then say KS, else say RP. (We do not count wrap around.)

Prob that Limited Eve Wins

If KS then Eve is correct (we omit this part).

Prob that Limited Eve Wins

If KS then Eve is correct (we omit this part).
If RP then prob Eve wrong is prob a rand perm has 3 cons lets.

- Number of perms: 26!
- Number of perms with 3 consecutive letters:

Prob that Limited Eve Wins

If KS then Eve is correct (we omit this part).
If RP then prob Eve wrong is prob a rand perm has 3 cons lets.

- Number of perms: 26!
- Number of perms with 3 consecutive letters:

  Pick the space to begin the 3 cons lets: 24 $(a, \ldots, x)$

Prob that Limited Eve Wins

If KS then Eve is correct (we omit this part).
If RP then prob Eve wrong is prob a rand perm has 3 cons lets.

- Number of perms: 26!
- Number of perms with 3 consecutive letters:

  Pick the space to begin the 3 cons lets: 24 ($a, \ldots, x$)

  Pick the let to put there (also determinesthe next 2 lets): 26

Prob that Limited Eve Wins

If KS then Eve is correct (we omit this part).
If RP then prob Eve wrong is prob a rand perm has 3 cons lets.

- Number of perms: 26!
- Number of perms with 3 consecutive letters:

  Pick the space to begin the 3 cons lets: 24 $(a, \ldots, x)$

  Pick the let to put there (also determinesthe next 2 lets): 26

  Permute remaining 23 letters in remaining 23 places: 23!

Prob that Limited Eve Wins

If KS then Eve is correct (we omit this part).
If RP then prob Eve wrong is prob a rand perm has 3 cons lets.

- Number of perms: 26!
- Number of perms with 3 consecutive letters:

  Pick the space to begin the 3 cons lets: 24 $(a, \ldots, x)$

  Pick the let to put there (also determinesthe next 2 lets): 26

  Permute remaining 23 letters in remaining 23 places: 23!

  We have counted some perms ≥ 2 times. So
  Numb of perms with 3 cons lets is $\leq 24 \times 26 \times 23!$.

Prob that Limited Eve Wins

If KS then Eve is correct (we omit this part).
If RP then prob Eve wrong is prob a rand perm has 3 cons lets.

- Number of perms: 26!
- Number of perms with 3 consecutive letters:

  Pick the space to begin the 3 cons lets: 24 $(a, \ldots, x)$

  Pick the let to put there (also determinesthe next 2 lets): 26

  Permute remaining 23 letters in remaining 23 places: 23!

  We have counted some perms ≥ 2 times. So
  Numb of perms with 3 cons lets is $\leq 24 \times 26 \times 23!$.

  Prob that Alice picks perm with 3 cons lets is

$$\leq \frac{24 \times 26 \times 23!}{26!} = \frac{1}{25} = 0.04$$

Prob that Limited Eve Wins

If KS then Eve is correct (we omit this part).
If RP then prob Eve wrong is prob a rand perm has 3 cons lets.

- Number of perms: 26!
- Number of perms with 3 consecutive letters:

  Pick the space to begin the 3 cons lets: 24 $(a, \ldots, x)$

  Pick the let to put there (also determinesthe next 2 lets): 26

  Permute remaining 23 letters in remaining 23 places: 23!

  We have counted some perms ≥ 2 times. So
  Numb of perms with 3 cons lets is $\leq 24 \times 26 \times 23!$.

  Prob that Alice picks perm with 3 cons lets is

$$\leq \frac{24 \times 26 \times 23!}{26!} = \frac{1}{25} = 0.04$$

  Prob that Eve wins is $\geq 1 - 0.04 = 0.96$.

Prob Eve wins is $\frac{1}{2} \times 1 + \frac{1}{2} \times 0.096 = 0.98$

Definition of Random-Looking

We do this informally.

Definition of Random-Looking

We do this informally.
Let C be a crypto-system. Let $|C|$ be the number of perms. (For Shift $|C| = 26$, for keyword-shift with 6-letter words, $|C| = 26 \times \cdots 21 \times 26$).

Definition of Random-Looking

We do this informally.
Let C be a crypto-system. Let $|C|$ be the number of perms. (For Shift $|C| = 26$, for keyword-shift with 6-letter words, $|C| = 26 \times \cdots 21 \times 26$).

Assume Eve is limited in time by $\log |C|$. (The idea is that Eve REALLY cannot look at anything close to $|C|$ perms.)

Definition of **Random-Looking**

We do this informally.
Let C be a crypto-system. Let $|C|$ be the number of perms. (For Shift $|C| = 26$, for keyword-shift with 6-letter words, $|C| = 26 \times \cdots 21 \times 26$).

Assume Eve is limited in time by $\log |C|$. (The idea is that Eve REALLY cannot look at anything close to $|C|$ perms.)

C generates perms that **look random** if when Eve plays the game the prob that she wins is $\leq \frac{1}{2}$.

Why this is all Silly and Why this is Not all Silly

Why this is all Silly and Why this is Not all Silly

1. **Silly** We can measure how good a cipher C is much more easily by looking at how many different permutations it can generate. For example, Shift leads to 26 perms, Affine to 312 perms.

Why this is all Silly and Why this is Not all Silly

1. **Silly** We can measure how good a cipher C is much more easily by looking at how many different permutations it can generate. For example, Shift leads to 26 perms, Affine to 312 perms.
2. **Not Silly** The above objection holds for **unlimited Eve** but not for **limited Eve**.

Why this is all Silly and Why this is Not all Silly

1. **Silly** We can measure how good a cipher C is much more easily by looking at how many different permutations it can generate. For example, Shift leads to 26 perms, Affine to 312 perms.
2. **Not Silly** The above objection holds for **unlimited Eve** but not for **limited Eve**.
3. **Not Silly** We have restated Keyword-Shift and Keyword-Mixed as ways to take a short seed and get a **Random-Looking** permutation.

Why this is all Silly and Why this is Not all Silly

1. **Silly** We can measure how good a cipher C is much more easily by looking at how many different permutations it can generate. For example, Shift leads to 26 perms, Affine to 312 perms.
2. **Not Silly** The above objection holds for **unlimited Eve** but not for **limited Eve**.
3. **Not Silly** We have restated Keyword-Shift and Keyword-Mixed as ways to take a short seed and get a **Random-Looking** permutation.
   This is an example of a **psuedo-random generator**.

Why this is all Silly and Why this is Not all Silly

1. **Silly** We can measure how good a cipher C is much more easily by looking at how many different permutations it can generate. For example, Shift leads to 26 perms, Affine to 312 perms.
2. **Not Silly** The above objection holds for **unlimited Eve** but not for **limited Eve**.
3. **Not Silly** We have restated Keyword-Shift and Keyword-Mixed as ways to take a short seed and get a **Random-Looking** permutation.
   This is an example of a **psuedo-random generator**.
   We will visit that concept later and use a similar game.

Which is Better Keyword-Mixed or Keyword-Shift?

1. Have given you a way to find out for yourself.
2. Might make into a HW.