REVIEW FOR MIDTERM PART TWO

October 12, 2020

The Vigenère Cipher

Key: $k = (k_1, k_2, \ldots, k_n)$.
Encrypt (all arithmetic is mod 26)

$$Enc(m_1, m_2, \ldots, m_N) =$$

$$m_1 + k_1, m_2 + k_2, \ldots, m_n + k_n,$$

$$m_{n+1} + k_1, m_{n+2} + k_2, \ldots, m_{n+n} + k_n,$$

$$\ldots$$

Decrypt Decryption just reverses the process

Three Kinds of Vigenère Ciphers

1. Standard Vig: Use a longish-sentence. Key is Sentence.
2. Book Cipher: Use a book. Key is name of book and edition.
3. one-time pad: Key is random gen sequence.

Cracking Vig cipher: Step One-find Keylength

One have keylength can crack, so okay to try many of them.
Two ways to guess keylengths:

1. Spot (say) a 4-letter sequence that appears 5 times and use differences of appearred to narrow down key length.
2. Try all keylengths of length 1,2,3,... until you hit it.

Cracking the Vig cipher: Step Two-Freq Anal

After Step One we have the key length L. Note:

Cracking the Vig cipher: Step Two-Freq Anal

After Step One we have the key length L. Note:

- Every L^{th} character is "encrypted" using the same shift.

Cracking the Vig cipher: Step Two-Freq Anal

After Step One we have the key length L. Note:

- Every L^{th} character is "encrypted" using the same shift.
- **Important:** Letter Freq still holds if you look at every Lth letter!

Cracking the Vig cipher: Step Two-Freq Anal

After Step One we have the key length L. Note:

- Every L^{th} character is "encrypted" using the same shift.
- **Important:** Letter Freq still holds if you look at every Lth letter!

Step Two:

Cracking the Vig cipher: Step Two-Freq Anal

After Step One we have the key length L. Note:

- Every L^{th} character is "encrypted" using the same shift.
- **Important:** Letter Freq still holds if you look at every Lth letter!

Step Two:

1. Separate text T into L streams depending on position mod L.

Cracking the Vig cipher: Step Two-Freq Anal

After Step One we have the key length L. Note:

- Every L^{th} character is "encrypted" using the same shift.
- **Important:** Letter Freq still holds if you look at every Lth letter!

Step Two:

1. Separate text T into L streams depending on position mod L.
2. For each steam try every shift and use **Is English** to determine which shift is correct.

Cracking the Vig cipher: Step Two-Freq Anal

After Step One we have the key length L. Note:

- Every L^{th} character is "encrypted" using the same shift.
- **Important:** Letter Freq still holds if you look at every Lth letter!

Step Two:

1. Separate text T into L streams depending on position mod L.
2. For each steam try every shift and use **Is English** to determine which shift is correct.
3. You now know all shifts for all positions. Decrypt!

Getting More Out of Your Phrase

If the key was

Corn Flake

You would get a key of length 9. We want **More**.

Getting More Out of Your Phrase

If the key was

Corn Flake

You would get a key of length 9. We want **More**.

Corn is 4 letters long. **Flake** is 5 letters long.
We form a key of length $LCM(4,5) = 20$. (Won't fit on line! Oh Well.)

C	O	R	N	C	O	R	N	C	O	R	N	C	O	R	N	C
F	L	A	K	E	F	L	A	K	E	F	L	A	K	E	F	L
7	25	17	23	6	19	2	13	12	18	22	24	2	24	21	18	1

ADD it up to get new 20-long key.

Book Cipher

A student said:

Let's use Vig cipher with a book for the key

Is it a good idea? **Discuss**

Book Cipher

A student said:

Let's use Vig cipher with a book for the key

Is it a good idea? **Discuss**

1. Before modern computer era: YES.
2. Now. NO.

How to Crack the Vig Book Cipher

Eve sees a d. (Recall that $d = 3$.) What does Eve know? **Discuss**

How to Crack the Vig Book Cipher

Eve sees a d. (Recall that $d = 3$.) What does Eve know? **Discuss**

Eve knows that (First Letter in Key) + (First Letter in Text) = 3.
Hence the following are the only possibilities for
(Letter in Key, Letter in Text) are:

(a, d), (z, e), (y, f), (w, g), $\ldots$, (b, c)

Only 26 possibilities. What of it? **Discuss**

How to Crack the Vig Book Cipher

Eve sees a d. (Recall that $d = 3$.) What does Eve know? **Discuss**

Eve knows that (First Letter in Key) + (First Letter in Text) = 3.
Hence the following are the only possibilities for
(Letter in Key, Letter in Text) are:

$(a, d), (z, e), (y, f), (w, g), \ldots, (b, c)$

Only 26 possibilities. What of it? **Discuss**
Some of the pairs are more likely than others.

1. **Both** the key **and** the text are in English.
2. (z, e): Hmm, z is unlikely but e is likely.
3. (a, d): Hmm, seems more likely than (z, e).
4. Can rank which are more likely (e.g., add or mult the freqs).
5. Can then use adjacent letters and freq of adjacent pairs, and rank them.
6. Triples. Etc.

One-Time Pad

One-Time Pad

- Let $\mathcal{M} = \{0,1\}^n$, the set of all messages.

One-Time Pad

- Let $\mathcal{M} = \{0,1\}^n$, the set of all messages.
- *Gen*: choose a uniform key $k \in \{0,1\}^n$.

One-Time Pad

- Let $\mathcal{M} = \{0,1\}^n$, the set of all messages.
- *Gen*: choose a uniform key $k \in \{0,1\}^n$.
- $Enc_k(m) = k \oplus m$.

One-Time Pad

- Let $\mathcal{M} = \{0, 1\}^n$, the set of all messages.
- *Gen*: choose a uniform key $k \in \{0, 1\}^n$.
- $Enc_k(m) = k \oplus m$.
- $Dec_k(c) = k \oplus c$.

One-Time Pad

- Let $\mathcal{M} = \{0,1\}^n$, the set of all messages.
- *Gen*: choose a uniform key $k \in \{0,1\}^n$.
- $Enc_k(m) = k \oplus m$.
- $Dec_k(c) = k \oplus c$.
- Correctness:

$$\begin{aligned} Dec_k(Enc_k(m)) &= k \oplus (k \oplus m) \\ &= (k \oplus k) \oplus m \\ &= m \end{aligned}$$

One-Time Pad

1. **PRO** $\oplus$ is FAST!
2. **CON** If Key is N bits long can only send N bits.
3. **PRO** Uncrackable if use truly random bits.
4. **CON** Hard to get truly random bits.

Ways to Get Random-Looking Bits

1. **Linear Cong Gen** Pick x_0, A, B, M at random and then use:
   x_0
   $x_{i+1} = Ax_i + B \pmod{M}$
   We summarize how to crack VERY BRIEFLY after this slide.
2. **Merseen Twister** Also a recurrence, also crackable.
3. **VN method** if can generate bits with prob(0)=p, prob(1)=$1-p$ can use to get truly random bits without knowing p.
   Takes a long time to get the bits.
   Generating bits with prob p, $1-p$, still hard.
4. **Elias method** Did not do in class, but
   Better than VN for time to get bits.
   Generating bits with prob p, $1-p$, still hard.
5. **We will see better methods later in the course.**

Cracking Linear Cong Gen

1. Have some word or phrase that you think is there. E.g., **PAKISTAN.** Say its 8 letters.
2. For EVERY 8-letter block (until you succeed) do the thought experliement: What if its **PAKISTAN**?
   2.1 Based on that guess find equations that relate A, B, M.
   2.2 Try to solve those equations. If no solution goto next block-of-8.
   2.3 There is ≥ 1 solution (A, B, M). Use it to find x_0 and the entire plaintext T.
   2.4 Test if T IS-English. If so then DONE. If not then goto next block-of-8.

The Matrix Cipher

Def Matrix Cipher. Pick M an $n \times n$ invertible over mod 26 matrix.

1. Encrypt via $xy \to M(xy)$.
2. Decrypt via $xy \to M^{-1}(xy)$.

Encode: Break text T into blocks of 2, apply M to each pair.

Decode: Do the same only with M^{-1}.

The Matrix Cipher: Good and Bad

Good News:

1. Can test if M^{-1} exists, and if so find it, easily.
2. M small, so Key small.
3. Applying M or M^{-1} to a vector is easy computationally.
4. Not clear if Eve can crack using Ciphertext Only Attack.
5. If n is large enough Eve cannot use brute force, but see next slide.

The Matrix Cipher: Good and Bad

Good News:

1. Can test if M^{-1} exists, and if so find it, easily.
2. M small, so Key small.
3. Applying M or M^{-1} to a vector is easy computationally.
4. Not clear if Eve can crack using Ciphertext Only Attack.
5. If n is large enough Eve cannot use brute force, but see next slide.

Bad News:

1. Eve CAN crack using Known Plaintext Attack Using Linear Algebra.

Lets Try Brute Force Even if Slow

1. Input T, a coded text.
2. For EVERY 8×8 invertible matrix M over mod 26,
   2.1 Decode T into T' using M.
   2.2 IF LOOKS-LIKE-ENGLISH(T')=YES then STOP and output T', else goto next matrix M.

Takes roughly 26^{64} steps.

Can Crack in 8×26^8

The attack in the last slide went through every **Matrix**.

Can Crack in 8×26^8

The attack in the last slide went through every **Matrix**.

Better Idea: We take life **one row at a time**.

Can Crack in 8×26^8

The attack in the last slide went through every **Matrix**.
Better Idea: We take life **one row at a time**.
Example: 3×3 matrix cipher. Decode Matrix M.

$$T = t_1 t_2 \cdots t_N \text{ each } t_i \text{ is 3-long}$$

Can Crack in 8×26^8

The attack in the last slide went through every **Matrix**.
Better Idea: We take life **one row at a time**.
Example: 3×3 matrix cipher. Decode Matrix M.

$$T = t_1 t_2 \cdots t_N \text{ each } t_i \text{ is 3-long}$$

Guess the first row of M. Say:

$$\begin{pmatrix} 1 & 1 & 7 \\ * & * & * \\ * & * & * \end{pmatrix}$$

Can Crack in 8×26^8

The attack in the last slide went through every **Matrix**.
Better Idea: We take life **one row at a time**.
Example: 3×3 matrix cipher. Decode Matrix M.

$$T = t_1 t_2 \cdots t_N \text{ each } t_i \text{ is 3-long}$$

Guess the first row of M. Say:

$$\begin{pmatrix} 1 & 1 & 7 \\ * & * & * \\ * & * & * \end{pmatrix}$$

Let $Mt_i = m_i$. Then $(1,1,7) \cdot t_i = m_i^1$ is first letter of m_i.

$$(m_1^1, m_2^1, m_3^1, \ldots, m_N^1)$$

is every third letter. Can do IS-ENGLISH on it.

Can Crack in 8×26^8

Eve knows that Alice and Bob decode with 8×8 Matrix M.
Ciphertext is

$$T = t_1 t_2 \cdots t_N \qquad t_i = t_i^1 \cdots t_i^8$$

For $i = 1$ to 8

For all $r \in \mathbb{Z}_{26}^8$ (guess that r is ith row of B).

$T' = (r \cdot t_1, \ldots, r \cdot t_N)$ (Is every 8th letter.)

IF IS-ENGLISH(T')=YES then $r_i = r$ and goto next i. Else goto the next r.

M is

$$\begin{pmatrix} \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots \\ r_1 & \cdots & r_n \\ \vdots & \vdots & \vdots \\ \cdots & \cdots & \cdots \end{pmatrix}$$

Takes 8×26^8 steps.

More General n

If M is $n \times n$ matrix.

More General n

If M is $n \times n$ matrix.

Brute force takes $O(26^{n^2})$.

More General n

If M is $n \times n$ matrix.

Brute force takes $O(26^{n^2})$.

The row-by-row method takes $O(n26^n)$.

Important Lesson

Assume: 26^{64} time is big enough to thwart Eve.

1. If we think that best Eve can do is $O(26^{n^2})$ then we take $n = 8$, so Eve needs $O(26^{64})$.
2. If we think that best Eve can do is $O(n26^n)$ then we take $n = 80$, so Eve needs $O(80 \times 26^{80})$.

The $O(n \times 26^n)$ cracking **does not** show that Matrix Cipher is insecure, but it still is very important: Alice and Bob must increase their parameters. That is already a win since it makes life harder for Alice and Bob.

The History of Cryptography in One Slide

The History of Cryptography in One Slide

1. Alice and Bob come up with a Crypto system (e.g., Matrix Cipher with $n = 8$).

The History of Cryptography in One Slide

1. Alice and Bob come up with a Crypto system (e.g., Matrix Cipher with $n = 8$).
2. Alice and Bob think its uncrackable and have a "proof" that it is uncrackable (e.g., Eve HAS to go through all 26^{64} matrices).

The History of Cryptography in One Slide

1. Alice and Bob come up with a Crypto system (e.g., Matrix Cipher with $n = 8$).
2. Alice and Bob think its uncrackable and have a "proof" that it is uncrackable (e.g., Eve HAS to go through all 26^{64} matrices).
3. Eve Cracks it. (The trick above- only about 8×26^8.)

The History of Cryptography in One Slide

1. Alice and Bob come up with a Crypto system (e.g., Matrix Cipher with $n = 8$).
2. Alice and Bob think its uncrackable and have a "proof" that it is uncrackable (e.g., Eve HAS to go through all 26^{64} matrices).
3. Eve Cracks it. (The trick above- only about 8×26^{8}.)
4. Lather, Rinse, Repeat.

The History of Cryptography in One Slide

1. Alice and Bob come up with a Crypto system (e.g., Matrix Cipher with $n = 8$).
2. Alice and Bob think its uncrackable and have a "proof" that it is uncrackable (e.g., Eve HAS to go through all 26^{64} matrices).
3. Eve Cracks it. (The trick above- only about 8×26^8.)
4. Lather, Rinse, Repeat.

Above attack on Matrix Cipher is a microcosm of this history.

The History of Cryptography in One Slide

1. Alice and Bob come up with a Crypto system (e.g., Matrix Cipher with $n = 8$).
2. Alice and Bob think its uncrackable and have a "proof" that it is uncrackable (e.g., Eve HAS to go through all 26^{64} matrices).
3. Eve Cracks it. (The trick above- only about 8×26^8.)
4. Lather, Rinse, Repeat.

Above attack on Matrix Cipher is a microcosm of this history.

Proofs rely on limiting what Eve can do, and hence do not work if Eve does something else.

Cracking Matrix Cipher With Known Ciphertext Attack

Example using 2×2 Matrix Cipher.
Eve learns that (13,24) encrypts to $(3, 9)$. Hence:

Cracking Matrix Cipher With Known Ciphertext Attack

Example using 2×2 Matrix Cipher.
Eve learns that (13,24) encrypts to $(3, 9)$. Hence:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 13 \\ 24 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$

Cracking Matrix Cipher With Known Ciphertext Attack

Example using 2×2 Matrix Cipher.
Eve learns that (13,24) encrypts to $(3, 9)$. Hence:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 13 \\ 24 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$

So

$$\begin{aligned} 13a + 24b &= 3 \\ 13c + 24d &= 9 \end{aligned}$$

Two linear equations, Four variables

Cracking Matrix Cipher With Known Ciphertext Attack

Example using 2×2 Matrix Cipher.
Eve learns that (13,24) encrypts to $(3, 9)$. Hence:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 13 \\ 24 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$

So

$$\begin{aligned} 13a + 24b &= 3 \\ 13c + 24d &= 9 \end{aligned}$$

Two linear equations, Four variables

If Eve learns one more 2-letter message decoding then she will have

Four linear equations, Four variables

which she can solve!

Cracking Matrix Cipher With Known Ciphertext Attack

Example using 2×2 Matrix Cipher.
Eve learns that (13,24) encrypts to $(3, 9)$. Hence:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 13 \\ 24 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$

So

$$\begin{aligned} 13a + 24b &= 3 \\ 13c + 24d &= 9 \end{aligned}$$

Two linear equations, Four variables

If Eve learns one more 2-letter message decoding then she will have

Four linear equations, Four variables

which she can solve! Yeah?

Cracking Matrix Cipher With Known Ciphertext Attack

Example using 2×2 Matrix Cipher.
Eve learns that (13,24) encrypts to $(3, 9)$. Hence:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 13 \\ 24 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$

So

$$\begin{aligned} 13a + 24b &= 3 \\ 13c + 24d &= 9 \end{aligned}$$

Two linear equations, Four variables

If Eve learns one more 2-letter message decoding then she will have

Four linear equations, Four variables

which she can solve! Yeah?Boo?

Cracking Matrix Cipher With Known Ciphertext Attack

Example using 2×2 Matrix Cipher.
Eve learns that (13,24) encrypts to $(3, 9)$. Hence:

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 13 \\ 24 \end{pmatrix} = \begin{pmatrix} 3 \\ 9 \end{pmatrix}$$

So

$$\begin{aligned} 13a + 24b &= 3 \\ 13c + 24d &= 9 \end{aligned}$$

Two linear equations, Four variables

If Eve learns one more 2-letter message decoding then she will have

Four linear equations, Four variables

which she can solve! Yeah?Boo?Depends whose side you are on.

Upshot

Upshot

1. Matrix Cipher with ciphertext only might be hard to crack.

Upshot

1. Matrix Cipher with ciphertext only might be hard to crack.
2. Matrix Cipher where Eve has access to prior messages is easy to crack.

Upshot

1. Matrix Cipher with ciphertext only might be hard to crack.
2. Matrix Cipher where Eve has access to prior messages is easy to crack.
3. We need to better refine our notion of **attack**.

Upshot

1. Matrix Cipher with ciphertext only might be hard to crack.
2. Matrix Cipher where Eve has access to prior messages is easy to crack.
3. We need to better refine our notion of **attack**.
4. We will do this in the next set of slides.

Ciphertext and Plaintext

Plaintext The message Alice really wants to send to Bob, e.g.

Meet me after class.

Ciphertext and Plaintext

Plaintext The message Alice really wants to send to Bob, e.g.

Meet me after class.

Ciphertext What Alice sends Bob. The hope is that if Eve sees it she will **not** learn the plaintext. E.g.

PHHWP HDIWH UFODV V

Types of Attacks

We will describe several different types of attacks Eve can use. They depend on:

1. What information Eve has access to.
2. What computing power Eve has.

Types of Attacks

We will describe several different types of attacks Eve can use. They depend on:

1. What information Eve has access to.
2. What computing power Eve has.

Eve's goal is to find out something about the plaintext she did not already know.

Ciphertext Only Attack (COA)

Ciphertext Only Attacks (COA) All Eve has is the ciphertext.

Ciphertext Only Attack (COA)

Ciphertext Only Attacks (COA) All Eve has is the ciphertext.

Eve cracked shift, affine, general sub, Vig with a COA.

Known Plaintext Attack (KPA)

Known Plaintext Attack (KPA) Eve knows the plaintext for **some of** the ciphertext.

Known Plaintext Attack (KPA)

Known Plaintext Attack (KPA) Eve knows the plaintext for **some of** the ciphertext.

Eve can crack Matrix and Linear-Cong-Gen ciphers with a KPA.

Known Plaintext Attack (KPA)

Known Plaintext Attack (KPA) Eve knows the plaintext for **some of** the ciphertext.

Eve can crack Matrix and Linear-Cong-Gen ciphers with a KPA. How can Eve obtain plaintext for some of the ciphertext?

Known Plaintext Attack (KPA)

Known Plaintext Attack (KPA) Eve knows the plaintext for **some of** the ciphertext.

Eve can crack Matrix and Linear-Cong-Gen ciphers with a KPA. How can Eve obtain plaintext for some of the ciphertext?

1. If yesterday the message **ABC DEFG** was where the spy would be, and today Eve found the spy in **New York**, then **ABC DEFG** decodes to **New York**.

Known Plaintext Attack (KPA)

Known Plaintext Attack (KPA) Eve knows the plaintext for **some of** the ciphertext.

Eve can crack Matrix and Linear-Cong-Gen ciphers with a KPA. How can Eve obtain plaintext for some of the ciphertext?

1. If yesterday the message **ABC DEFG** was where the spy would be, and today Eve found the spy in **New York**, then **ABC DEFG** decodes to **New York**.
2. **WWII History** A German solider in an area where nothing was happening sent **nothing to report** every day.

Known Plaintext Attack (KPA)

Known Plaintext Attack (KPA) Eve knows the plaintext for **some of** the ciphertext.

Eve can crack Matrix and Linear-Cong-Gen ciphers with a KPA. How can Eve obtain plaintext for some of the ciphertext?

1. If yesterday the message **ABC DEFG** was where the spy would be, and today Eve found the spy in **New York**, then **ABC DEFG** decodes to **New York**.
2. **WWII History** A German solider in an area where nothing was happening sent **nothing to report** every day.
3. Guess a word that you think appears in the document. For Linear-Cong-Gen our example was **Pakistan**.

Known Plaintext Attack (KPA)

Known Plaintext Attack (KPA) Eve knows the plaintext for **some of** the ciphertext.

Eve can crack Matrix and Linear-Cong-Gen ciphers with a KPA. How can Eve obtain plaintext for some of the ciphertext?

1. If yesterday the message **ABC DEFG** was where the spy would be, and today Eve found the spy in **New York**, then **ABC DEFG** decodes to **New York**.
2. **WWII History** A German solider in an area where nothing was happening sent **nothing to report** every day.
3. Guess a word that you think appears in the document. For Linear-Cong-Gen our example was **Pakistan**.
4. **More WWII History** Turing and his gang cracked the German Enigma Code, guessing that **ein** (German for **one**) would be in messages.

Chosen Plaintext Attack (CPA)

Chosen Plaintext Attack (CPA) Eve tricks Alice into encoding a particular plaintext.

Chosen Plaintext Attack (CPA)

Chosen Plaintext Attack (CPA) Eve tricks Alice into encoding a particular plaintext.
Later in the course we will see a CPA attack on RSA.

Chosen Plaintext Attack (CPA)

Chosen Plaintext Attack (CPA) Eve tricks Alice into encoding a particular plaintext.
Later in the course we will see a CPA attack on RSA.

1. **WWII History** America thought that the Japanese code for Midway was **AF**. So the Americans send the message to a ship **Midway is low on supplies**. The Americans observe that the next Japanese message has **AF** in it, so their suspicions are confirmed.

Chosen Plaintext Attack (CPA)

Chosen Plaintext Attack (CPA) Eve tricks Alice into encoding a particular plaintext.
Later in the course we will see a CPA attack on RSA.

1. **WWII History** America thought that the Japanese code for Midway was **AF**. So the Americans send the message to a ship **Midway is low on supplies**. The Americans observe that the next Japanese message has **AF** in it, so their suspicions are confirmed.
2. **WWII History** England put mines in places that the Germans had no abbreviations for. The Germans cleared those mines and send **NAME OF PLACE, all clear**, transmitted in code.

Chosen Ciphertext Attack (CCA)

Chosen Ciphertext Attack (CCA) Eve tricks Alice into decoding a particular ciphertext.

Chosen Ciphertext Attack (CCA)

Chosen Ciphertext Attack (CCA) Eve tricks Alice into decoding a particular ciphertext.
Later in the course we may see a CCA attack on RSA.

Dictionary Attack

Dictionary Attack Many variants, we give two:

1. Have database of **X** decodes to **Y** and pattern match.
2. Use Dictionary to guess passwords.

Dictionary Attack

Dictionary Attack Many variants, we give two:

1. Have database of **X** decodes to **Y** and pattern match.
2. Use Dictionary to guess passwords. Or guess on of the most common passwords:

Dictionary Attack

Dictionary Attack Many variants, we give two:

1. Have database of **X** decodes to **Y** and pattern match.
2. Use Dictionary to guess passwords. Or guess on of the most common passwords:

123456	*Password*	12345678	*qwerty*	12345
12345678	*letmein*	1234567	*football*	*iloveyou*
admin	*welcome*	*monkey*	*login*	*abc*123
starwars	123123	*dragon*	*passwOrd*	*master*
hello	*freedom*	*whatever*	*qazwsx*	*trusno*1

Dictionary Attack

Dictionary Attack Many variants, we give two:

1. Have database of **X** decodes to **Y** and pattern match.
2. Use Dictionary to guess passwords. Or guess on of the most common passwords:

123456	*Password*	12345678	*qwerty*	12345
12345678	*letmein*	1234567	*football*	*iloveyou*
admin	*welcome*	*monkey*	*login*	*abc*123
starwars	123123	*dragon*	*passwOrd*	*master*
hello	*freedom*	*whatever*	*qazwsx*	*trusno*1

qwerty: 1st 6 letters on 3rd line of a keyboard.
qazwsz: Similar keyboard shenanigans.

Dictionary Attack

Dictionary Attack Many variants, we give two:

1. Have database of **X** decodes to **Y** and pattern match.
2. Use Dictionary to guess passwords. Or guess on of the most common passwords:

123456	*Password*	12345678	*qwerty*	12345
12345678	*letmein*	1234567	*football*	*iloveyou*
admin	*welcome*	*monkey*	*login*	*abc*123
starwars	123123	*dragon*	*passwOrd*	*master*
hello	*freedom*	*whatever*	*qazwsx*	*trusno*1

qwerty: 1st 6 letters on 3rd line of a keyboard.
qazwsz: Similar keyboard shenanigans.

trusno1: I like that one, I think I'll use it :-)

Brute Force Attacks (BFA)

Brute Force Attacks (BFA) Guess all possible keys.

Brute Force Attacks (BFA)

Brute Force Attacks (BFA) Guess all possible keys.
We cracked shift, affine, Vig, this way.

Brute Force Attacks (BFA)

Brute Force Attacks (BFA) Guess all possible keys.
We cracked shift, affine, Vig, this way.
Only effective if either:

Brute Force Attacks (BFA)

Brute Force Attacks (BFA) Guess all possible keys.
We cracked shift, affine, Vig, this way.
Only effective if either:

1. The key space is small enough for Eve's computing power.

Brute Force Attacks (BFA)

Brute Force Attacks (BFA) Guess all possible keys.
We cracked shift, affine, Vig, this way.
Only effective if either:

1. The key space is small enough for Eve's computing power.
2. Clever way to reduce key space before you do BFA.

Brute Force Attacks (BFA)

Brute Force Attacks (BFA) Guess all possible keys.
We cracked shift, affine, Vig, this way.
Only effective if either:

1. The key space is small enough for Eve's computing power.
2. Clever way to reduce key space before you do BFA.

Easy to thwart Use a bigger key space!

Timing and Power Attacks

Use how much time or power a cryptosystem is using to figure out information about the key and narrow down the keyspace.

Timing and Power Attacks

Use how much time or power a cryptosystem is using to figure out information about the key and narrow down the keyspace.

1. Scary!

Timing and Power Attacks

Use how much time or power a cryptosystem is using to figure out information about the key and narrow down the keyspace.

1. Scary! Alice and Bob are using cryptosystem C such that Eve can crack C with a CPA attack iff Eve can Solve the ERIC problem.

Timing and Power Attacks

Use how much time or power a cryptosystem is using to figure out information about the key and narrow down the keyspace.

1. Scary! Alice and Bob are using cryptosystem C such that Eve can crack C with a CPA attack iff Eve can Solve the ERIC problem. The mathematical proof of this does not take timing attacks into account.

Timing and Power Attacks

Use how much time or power a cryptosystem is using to figure out information about the key and narrow down the keyspace.

1. Scary! Alice and Bob are using cryptosystem C such that Eve can crack C with a CPA attack iff Eve can Solve the ERIC problem. The mathematical proof of this does not take timing attacks into account.
   Alice and Bob can easily thwart timing/power attacks by padding. But there was a time before this attack was known when it may have been effective.

Timing and Power Attacks

Use how much time or power a cryptosystem is using to figure out information about the key and narrow down the keyspace.

1. Scary! Alice and Bob are using cryptosystem C such that Eve can crack C with a CPA attack iff Eve can Solve the ERIC problem. The mathematical proof of this does not take timing attacks into account.
   Alice and Bob can easily thwart timing/power attacks by padding. But there was a time before this attack was known when it may have been effective.
2. There may be other attacks that we do not know about like Timing/Power was. It may be hard (impossible?) to prove that there is no such attack that works. This involves issues outside of the Mathematical realm.

Timing and Power Attacks

Use how much time or power a cryptosystem is using to figure out information about the key and narrow down the keyspace.

1. Scary! Alice and Bob are using cryptosystem C such that Eve can crack C with a CPA attack iff Eve can Solve the ERIC problem. The mathematical proof of this does not take timing attacks into account.
   Alice and Bob can easily thwart timing/power attacks by padding. But there was a time before this attack was known when it may have been effective.
2. There may be other attacks that we do not know about like Timing/Power was. It may be hard (impossible?) to prove that there is no such attack that works. This involves issues outside of the Mathematical realm.
3. Look up the Maginot Line.

The Playfair Cipher: The Key

We use $\Sigma = \{a, \ldots, z\} - \{j\}$. Need a square. If need to use j use an i.

Key is a word or phrase. Delete all repeats from it. We will use **Bill Gasarch** which becomes BILGASRCH. Use the key to start a 5×5 array of all of the letters

B	I	L	G	A
S	R	C	H	D
E	F	K	M	N
O	P	Q	T	U
V	W	X	Y	Z

The Playfair Cipher: The First Case

B	I	L	G	A
S	R	C	H	D
E	F	K	M	N
O	P	Q	T	U
V	W	X	Y	Z

Given a pair, what do you map it to? Start by finding the pair in the grid.

1) If the pair are NOT in the same row or column then look at rectangle formed and take other corners. EXAMPLE: Map *RA*:

I	L	G	A
R	C	H	D

RA maps to *ID*.

The Playfair Cipher: The Other Cases

We skip them for the review.

The Rail Fence Cipher

October 12, 2020

Rail Fence Cipher as Understood In Past

Key is 3. Message is **Marina is a TA**.

Write it in three rows as such:

```
M           N           A
   A     I     A     S     T
      R           I           A
```

Write each row: **MNAAIASTRIA**

Rail Fence Cipher as Understood In Past

Key is 3. Message is **Marina is a TA**.

Write it in three rows as such:

```
M       N       A
  A   I   A   S   T
    R       I       A
```

Write each row: **MNAAIASTRIA**

How would you describe this cipher in modern terminology?

Discuss

Rail Fence Cipher as Understood In Past

Key is 3. Message is **Marina is a TA**.

Write it in three rows as such:

```
M       N       A
  A   I   A   S   T
    R       I       A
```

Write each row: **MNAAIASTRIA**

How would you describe this cipher in modern terminology?

Discuss

In current case of 3 rows and message of length 11 we did

Rail Fence Cipher as Understood In Past

Key is 3. Message is **Marina is a TA**.

Write it in three rows as such:

```
M           N           A
   A     I     A     S     T
      R           I           A
```

Write each row: **MNAAIASTRIA**

How would you describe this cipher in modern terminology?

Discuss

In current case of 3 rows and message of length 11 we did

First list out the letters in positions $\equiv 1 \pmod 4$.

Rail Fence Cipher as Understood In Past

Key is 3. Message is **Marina is a TA**.

Write it in three rows as such:

```
M           N           A
   A     I     A     S     T
      R           I           A
```

Write each row: **MNAAIASTRIA**

How would you describe this cipher in modern terminology?

Discuss

In current case of 3 rows and message of length 11 we did

First list out the letters in positions $\equiv 1 \pmod 4$.

Second list out the letters in positions $\equiv 0, 2 \pmod 4$.

Rail Fence Cipher as Understood In Past

Key is 3. Message is **Marina is a TA**.

Write it in three rows as such:

```
M       N       A
  A   I   A   S   T
    R       I       A
```

Write each row: **MNAAIASTRIA**

How would you describe this cipher in modern terminology?

Discuss

In current case of 3 rows and message of length 11 we did

First list out the letters in positions $\equiv 1 \pmod 4$.

Second list out the letters in positions $\equiv 0, 2 \pmod 4$.

Third list out letters in positions $\equiv 3 \pmod 4$).

Rail Fence Cipher as Understood In Past

Key is 3. Message is **Marina is a TA**.
Write it in three rows as such:

M				N				A		
	A		I		A		S		T	
		R				I				A

Write each row: **MNAAIASTRIA**
How would you describe this cipher in modern terminology?
Discuss
In current case of 3 rows and message of length 11 we did
First list out the letters in positions $\equiv 1 \pmod 4$.
Second list out the letters in positions $\equiv 0, 2 \pmod 4$.
Third list out letters in positions $\equiv 3 \pmod 4$).
Leave as an exercise what happens if k rows, n letter message.

The Autokey Cipher

October 12, 2020

The AutoKey Cipher: A Variant of Vig

IDEA: Use the plaintext as a Key after a start.

The AutoKey Cipher: A Variant of Vig

IDEA: Use the plaintext as a Key after a start.

1. There is a key, a short word or phrase. We'll use **Metz**.

The AutoKey Cipher: A Variant of Vig

IDEA: Use the plaintext as a Key after a start.

1. There is a key, a short word or phrase. We'll use **Metz**.
2. **Metz** is $(12, 4, 19, 25)$. We shift the first letter by 12, the second by 4, the third by 19, he fourth by 25.

The AutoKey Cipher: A Variant of Vig

IDEA: Use the plaintext as a Key after a start.

1. There is a key, a short word or phrase. We'll use **Metz**.
2. **Metz** is $(12, 4, 19, 25)$. We shift the first letter by 12, the second by 4, the third by 19, he fourth by 25.
3. After first four use plaintext lagged by 4.

The AutoKey Cipher: A Variant of Vig

IDEA: Use the plaintext as a Key after a start.

1. There is a key, a short word or phrase. We'll use **Metz**.
2. **Metz** is $(12, 4, 19, 25)$. We shift the first letter by 12, the second by 4, the third by 19, he fourth by 25.
3. After first four use plaintext lagged by 4.

Example Key is **Metz** and I want to encode **Joe Biden is running.** So Key is metzjoebidenisrunning

The AutoKey Cipher: A Variant of Vig

IDEA: Use the plaintext as a Key after a start.

1. There is a key, a short word or phrase. We'll use **Metz**.
2. **Metz** is $(12, 4, 19, 25)$. We shift the first letter by 12, the second by 4, the third by 19, he fourth by 25.
3. After first four use plaintext lagged by 4.

Example Key is **Metz** and I want to encode **Joe Biden is running.** So Key is metzjoebidenisrunning

1. Encode (j,o,e,b) by shifting by $(12, 4, 19, 25)$.

The AutoKey Cipher: A Variant of Vig

IDEA: Use the plaintext as a Key after a start.

1. There is a key, a short word or phrase. We'll use **Metz**.
2. **Metz** is $(12, 4, 19, 25)$. We shift the first letter by 12, the second by 4, the third by 19, he fourth by 25.
3. After first four use plaintext lagged by 4.

Example Key is **Metz** and I want to encode **Joe Biden is running.** So Key is metzjoebidenisrunning

1. Encode (j,o,e,b) by shifting by $(12, 4, 19, 25)$.
2. Encode

   $$(i, d, e, n, i, s, r, u, n, n, i, n, g)$$

   by the shift induced by

   $$(j, o, e, b, i, d, e, n, i, s, r, u, n)$$

To Decode will need to do this four letters at a time.

AutoKey Pros and Cons

PROS: The techniques for cracking Vig do not work.
PROS: If Eve does not know you are using it, seems uncrackable.

CON: Complicated to use (more on that next slide).

Question: How would you crack it?

AutoKey Pros and Cons

PROS: The techniques for cracking Vig do not work.
PROS: If Eve does not know you are using it, seems uncrackable.

CON: Complicated to use (more on that next slide).

Question: How would you crack it?

Similar to Book Cipher in that the key and the message are **both** in English so can use freq somewhat.

If guess the key word then rest is EASY!

(Another) Book Cipher

The term **Book Cipher** was used both for the Vig cipher where key is a book and for what I will discuss now.

(Another) Book Cipher

The term **Book Cipher** was used both for the Vig cipher where key is a book and for what I will discuss now.

Def Book Cipher:

(Another) Book Cipher

The term **Book Cipher** was used both for the Vig cipher where key is a book and for what I will discuss now.

Def Book Cipher:

1. Alice and Bob agree on a book to be the key.

(Another) Book Cipher

The term **Book Cipher** was used both for the Vig cipher where key is a book and for what I will discuss now.

Def Book Cipher:

1. Alice and Bob agree on a book to be the key.
2. To send a mesage Alice specifies, for each word,
   - A page number. E.g., Page 19.
   - A line number. E.g., Line 24 (On Page 19).
   - A word number. E.g., Word 4 (On Page 19, Line 24).

(Another) Book Cipher

The term **Book Cipher** was used both for the Vig cipher where key is a book and for what I will discuss now.

Def Book Cipher:

1. Alice and Bob agree on a book to be the key.
2. To send a mesage Alice specifies, for each word,
   - A page number. E.g., Page 19.
   - A line number. E.g., Line 24 (On Page 19).
   - A word number. E.g., Word 4 (On Page 19, Line 24).
3. Alice will try to use different triples for the same word.

(Another) Book Cipher

The term **Book Cipher** was used both for the Vig cipher where key is a book and for what I will discuss now.

Def Book Cipher:

1. Alice and Bob agree on a book to be the key.
2. To send a mesage Alice specifies, for each word,
   - A page number. E.g., Page 19.
   - A line number. E.g., Line 24 (On Page 19).
   - A word number. E.g., Word 4 (On Page 19, Line 24).
3. Alice will try to use different triples for the same word.
4. Bob has same book so can decode.

Security Known to be crackable, but won't go into that here.

A Problem with MOST of our Ciphers/Terminology

1. Most of our ciphers are deterministic so always code *m* the same way. This leaks information.
2. One-Time Pad and Book Ciphers avoid this, but have very long keys.
3. The problem of the same message leading to the same ciphertext is called

The NY,NY Problem.

How to Fix This Without a Long Key

Randomized Shift Key is a **function** $f : S \to S$.

1. To send message $(m_1, \ldots, m_L)$ (each m_i is a character):
   1.1 Pick random $r_1, \ldots, r_L \in S$.
   1.2 Send $((r_1; m_1 + f(r_1)), \ldots, (r_L; m_L + f(r_L)))$.

How to Fix This Without a Long Key

Randomized Shift Key is a **function** $f : S \to S$.

1. To send message $(m_1, \ldots, m_L)$ (each m_i is a character):
   1.1 Pick random $r_1, \ldots, r_L \in S$.
   1.2 Send $((r_1; m_1 + f(r_1)), \ldots, (r_L; m_L + f(r_L)))$.
2. To decode message $((r_1; c_1), \ldots, (r_L; c_L))$:
   2.1 Find $(c_1 - f(r_1), \ldots, c_L - f(r_L))$.

Cracking Randomized Shift

With a long text Rand Shift **is** crackable.
If N is long and Eve sees:

$$(r_1; \sigma_1)(r_2; \sigma_2) \cdots (r_N; \sigma_N).$$

1. If N is long enough we will see EVERY r MANY times.
2. From our study of Vig we know that taking every mth letter in a text has the same distribution of letters as a normal text.

Cracking Randomized Shift

With a long text Rand Shift **is** crackable.
If N is long and Eve sees:

$$(r_1; \sigma_1)(r_2; \sigma_2) \cdots (r_N; \sigma_N).$$

1. If N is long enough we will see EVERY r MANY times.
2. From our study of Vig we know that taking every mth letter in a text has the same distribution of letters as a normal text.
3. It turns out that taking a **random** set of letters also has the same distribution as a normal text.

Cracking Randomized Shift Final Algorithm

Cracking Randomized Shift Final Algorithm

1. Input $(r_1; \sigma_1)(r_2; \sigma_2) \cdots (r_N; \sigma_N)$

Cracking Randomized Shift Final Algorithm

1. Input $(r_1;\sigma_1)(r_2;\sigma_2)\cdots(r_N;\sigma_N)$
2. For each $r \in \{1,\ldots,26\}$:

Cracking Randomized Shift Final Algorithm

1. Input $(r_1; \sigma_1)(r_2; \sigma_2) \cdots (r_N; \sigma_N)$
2. For each $r \in \{1, \ldots, 26\}$:
    2.1 Look at the spots (r, σ), so

$$(r, \sigma_1) \cdots (r, \sigma_2) \cdots (r, \sigma_L).$$

Cracking Randomized Shift Final Algorithm

1. Input $(r_1; \sigma_1)(r_2; \sigma_2) \cdots (r_N; \sigma_N)$
2. For each $r \in \{1, \ldots, 26\}$:
   2.1 Look at the spots (r, σ), so

   $$(r, \sigma_1) \cdots (r, \sigma_2) \cdots (r, \sigma_L).$$

   2.2 All of these σ_{i_j}'s used same shift.

Cracking Randomized Shift Final Algorithm

1. Input $(r_1; \sigma_1)(r_2; \sigma_2) \cdots (r_N; \sigma_N)$
2. For each $r \in \{1, \ldots, 26\}$:
    2.1 Look at the spots (r, σ), so

    $$(r, \sigma_1) \cdots (r, \sigma_2) \cdots (r, \sigma_L).$$

    2.2 All of these σ_{i_j}'s used same shift.
    2.3 Guess each shift and use IS-ENGLISH to find out which shift is correct.

Cracking Randomized Shift Final Algorithm

1. Input $(r_1; \sigma_1)(r_2; \sigma_2) \cdots (r_N; \sigma_N)$
2. For each $r \in \{1, \ldots, 26\}$:
   2.1 Look at the spots (r, σ), so

   $$(r, \sigma_1) \cdots (r, \sigma_2) \cdots (r, \sigma_L).$$

   2.2 All of these σ_{i_j}'s used same shift.
   2.3 Guess each shift and use IS-ENGLISH to find out which shift is correct.
3. We now have the mapping of r's to shifts. r maps to shift s_r.

Cracking Randomized Shift Final Algorithm

1. Input $(r_1; \sigma_1)(r_2; \sigma_2) \cdots (r_N; \sigma_N)$
2. For each $r \in \{1, \ldots, 26\}$:
   2.1 Look at the spots (r, σ), so
   $$(r, \sigma_1) \cdots (r, \sigma_2) \cdots (r, \sigma_L).$$
   2.2 All of these σ_{i_j}'s used same shift.
   2.3 Guess each shift and use IS-ENGLISH to find out which shift is correct.
3. We now have the mapping of r's to shifts. r maps to shift s_r.
4. Can use the s_r's to decode entire message.

Upshot

Upshot

1. Det. Ciphers: Message M always maps to the same thing. Boo!

Upshot

1. Det. Ciphers: Message M always maps to the same thing. Boo!
2. We can turn any Det. Cipher into a randomized one. Will use this later in the course.

Upshot

1. Det. Ciphers: Message M always maps to the same thing. Boo!
2. We can turn any Det. Cipher into a randomized one. Will use this later in the course.
3. If turn a weak Det. Cipher (like Shift) into a randomized one, still crackable.

Upshot

1. Det. Ciphers: Message M always maps to the same thing. Boo!
2. We can turn any Det. Cipher into a randomized one. Will use this later in the course.
3. If turn a weak Det. Cipher (like Shift) into a randomized one, still crackable.
4. Cracking it takes a much longer text.

BILL, STOP RECORDING LECTURE!!!!

BILL STOP RECORDING LECTURE!!!