REVIEW FOR MIDTERM PART TWO

Vig and One-Time Pad

The Vigenère Cipher

Key: $k = (k_1, k_2, \ldots, k_n)$.

The Vigenère Cipher

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

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

The Vigenère Cipher

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

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

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

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

$$\ldots$$

The Vigenère Cipher

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

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

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

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

$$\cdots$$

Decrypt Decryption just reverses the process

Three Kinds of Vigenère Ciphers

Three Kinds of Vigenère Ciphers

1. Standard Vig: Use a longish-sentence. Key is Sentence.

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.

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

Cracking Vig cipher: Step One-find Keylength

Two ways to guess key lengths:

Cracking Vig cipher: Step One-find Keylength

Two ways to guess key lengths:

1. Spot (say) a 4-letter sequence that appears 5 times and use differences of appeared to narrow down key length.

Cracking Vig cipher: Step One-find Keylength

Two ways to guess key lengths:

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

Cracking the Vig cipher: Step Two-Freq Anal

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!

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

One-Time Pad

1. **PRO**$\oplus$ is FAST!

One-Time Pad

1. **PRO** $\oplus$ is FAST!
2. **CON** If Key is N bits long can only send N bits.

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.

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

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.

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. **Mersenne Twister** Also a recurrence, also crackable, harder.

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. **Mersenne Twister** Also a recurrence, also crackable, harder.
3. **VN method** (1950's) and **Elias Method** (1980's) not covered in this course, but ended up not being that good.

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. **Mersenne Twister** Also a recurrence, also crackable, harder.
3. **VN method** (1950's) and **Elias Method** (1980's) not covered in this course, but ended up not being that good.
4. **We will see better methods later in the course.**

Cracking Linear Cong Gen

Cracking Linear Cong Gen

1. Have some word or phrase that you think is there. E.g., **PAKISTAN.** Say its 8 letters.

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 experiments: What if its **PAKISTAN**?

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 experiments: What if its **PAKISTAN**?
   2.1 Based on that guess find equations that relate A, B, M.

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 experiments: 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.

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 experiments: 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.

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 experiments: 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.

The Matrix Cipher

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

1. Encrypt via $xy \to M(xy)$.

The Matrix Cipher

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

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

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.

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:

The Matrix Cipher: Good and Bad

Good News:

1. Can test if M^{-1} exists, and if so find it, easily.

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.

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.

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.

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, 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, Eve cannot use brute force. But see next slide.

Bad News:

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, Eve cannot use brute force. But see next slide.

Bad News:

1. Eve CAN crack if she has pairs if plaintext-ciphertext, using linear algebra.

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, Eve cannot use brute force. But see next slide.

Bad News:

1. Eve CAN crack if she has pairs if plaintext-ciphertext, using linear algebra.
2. Caveat: the linear algebra is over mod 26.

Lets Try Brute Force Even if Slow

Lets Try Brute Force Even if Slow

1. Input T, a coded text.

Lets Try Brute Force Even if Slow

1. Input T, a coded text.
2. For EVERY 8×8 invertible matrix M over mod 26,

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.

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.

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.

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 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

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

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

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).

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.)

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.

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

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

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.

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 $O(26^{64})$.

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.

Another Lesson

I had Kunal code up the row-by-row algorithm.

Another Lesson

I had Kunal code up the row-by-row algorithm.

1. Didn't really work. The **IS-ENGLISH** program needs to have a **gap** between a correct and incorrect guess. Didn't seem to have that gap.

Another Lesson

I had Kunal code up the row-by-row algorithm.

1. Didn't really work. The **IS-ENGLISH** program needs to have a **gap** between a correct and incorrect guess. Didn't seem to have that gap.
2. I will later have Kunal or someone else look into more sophisticated **IS-ENGLISH** programs to see if we can make this work.

Another Lesson

I had Kunal code up the row-by-row algorithm.

1. Didn't really work. The **IS-ENGLISH** program needs to have a **gap** between a correct and incorrect guess. Didn't seem to have that gap.
2. I will later have Kunal or someone else look into more sophisticated **IS-ENGLISH** programs to see if we can make this work.
3. **Lesson Learned** A method to crack a code that looks good on paper may run into difficulties when really tried.

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. We'll assume she got around the **IS-ENGLISH** program issue.)

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}. We'll assume she got around the **IS-ENGLISH** program issue.)
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}. We'll assume she got around the **IS-ENGLISH** program issue.)
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. We'll assume she got around the **IS-ENGLISH** program issue.)
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 Pairs

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

Cracking Matrix Cipher With Pairs

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 Pairs

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

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

Cracking Matrix Cipher With Pairs

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 Pairs

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.

Eve can solve that!

Cracking Matrix Cipher With Pairs

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.

Eve can solve that! Yeah?

Cracking Matrix Cipher With Pairs

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.

Eve can solve that! Yeah? Boo?

Cracking Matrix Cipher With Pairs

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.

Eve can solve that! 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** (we do this later, after midterm).

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** (we do this later, after midterm).
4. We will do this in the next set of slides.

Other Ciphers

1. **The AutoKey Cipher** Use the message itself as the key. We skip details here, but note that could be good if If Eve does not know you are using it. So might be good if **Kerckhoff's Principle** does not hold.

Other Ciphers

1. **The AutoKey Cipher** Use the message itself as the key. We skip details here, but note that could be good if If Eve does not know you are using it. So might be good if **Kerckhoff's Principle** does not hold.
2. **(Another) Book Cipher** Alice and Bob agree on a book to be the key. Specify words by page/line/word.

Security Both are 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** $f : S \to S$.

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):

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 $r_1, \ldots, r_L \in S$.

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 $((r_1; c_1), \ldots, (r_L; c_L))$:

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 $(c_1 - f(r_1), \ldots, c_L - f(r_L))$.

Security Randomized Shift is crackable, but needs a longer text than ordinary shift. We won't get into that here—Our point is that adding randomization to shift (and other encoding systems) solves the NY,NY problem.

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!!!