Cryptography
Symmetric Encryption

Class 2

Stallings: Ch 3 & 6
Stallings: Ch 4
Symmetric Cryptosystems

Encryption Key

Plaintext

Encryption Algorithm

Decryption Key

Decryption Algorithm

Plaintext
Outline

- Encryption Techniques
  - Classical Techniques
  - Modern Times: DES
Classic Techniques

- Substitution Ciphers
  - Cesar Cipher
  - Monoalphabetic Ciphers
  - Polyalphabetic Ciphers
    - Vigenere
    - Vernam
    - One Time Pad
- Transposition Ciphers
- Product Ciphers
Classic Substitution Techniques

- *Letters of plaintext are replaced by other letters or by numbers or symbols*
- If plaintext is viewed as a sequence of bits
  - Replace plaintext bits with ciphertext bits
Caesar Cipher

- Earliest known substitution cipher
- By Julius Caesar
- First attested use in military affairs

**Example:**
meet me after the toga party
PHHW PH DIWHU WKH WRJD SDUWB

**Guess how?**
- Replaces each letter by 3rd letter on
Caesar Cipher (cont’d)

- Define transformation as:

  a b c d e f g h i j k l m n o p q r s t u v w x y z
  D E F G H I J K L M N O P Q R S T U V W X Y Z A B C

- Give each letter a number

  a b c d e f g h i j k l m n o p q r s t u v w x y z
  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

- Math -- Caesar cipher is:

  \[ c = E(k, p) = (p + k) \mod (26) \]
  \[ p = D(k, c) = (c - k) \mod (26) \]
Cryptanalysis of Caesar Cipher

- Only have 25 possible ciphers
  - A maps to B, C, ..., Z
- Brute force search
  - Given ciphertext, just try all shifts of letters
  - Do need to recognize when have plaintext
  - E.g. break ciphertext
    - QIIIX QI MR QMEQM
Monoalphabetic Cipher

- Shuffle the letters arbitrarily
  - Random permutation – how many in total?
- Each plaintext letter maps to a different random ciphertext letter
- Key is 26 letters long
- Example:

  Plain: abcdefghijklmnopqrstuvwxyz
  Cipher: DKVQFIBJWPESCTMYAUOLRGZN

  Plaintext: ifwewishtoreplaceletters
  Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
Monoalphabetic Cipher Security

- Key space increases from 25 (Caesar) to \(26! = 4 \times 10^{26}\) (English)
- With so many keys, might think is secure
- But would be **WRONG**
- Problem is language characteristics
Language and Cryptanalysis

- **Letters are not equally used**
  - In English E is by far the most common letter
  - Other letters like Z, J, K, Q, X are fairly rare

- Build digrams
- Trigrams ...


**Monoalphabetic substitution ciphers do not change relative letter frequencies**

- Discovered by Arabian scientists in 9\textsuperscript{th} century

1. Calculate letter frequencies for ciphertext
2. Compare counts against known values
   - peaks at: A-E-I triple, NO pair, RST triple
   - valleys at: JK, X-Z

---

Use In Cryptanalysis
Playfair Cipher

- Not even the large number of keys in a monoalphabetic cipher provides security
- Idea: *encrypt multiple letters at a time*
- **Example:** the **Playfair Cipher**
  - Invented by Charles Wheatstone in 1854, but named after his friend Baron Playfair
Playfair Key Matrix

- 5X5 matrix of letters based on a **keyword**
- Fill in letters of keyword (without duplicates)
- Fill rest of matrix with other letters
- **Example:** using the keyword **MONARCHY**

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

Plaintext is encrypted two letters at a time

1. If a pair is a repeated letter, insert filler like 'X’
2. If both letters fall in the same row, replace each with letter to right (wrapping back to start from end)
3. If both letters fall in the same column, replace each with the letter below it (wrapping to top from bottom)
4. Otherwise each letter is replaced by the letter in the same row and in the column of the other letter of the pair
Security of Playfair

- Security much improved over monoalphabetic
- Since we have $26 \times 26 = 676$ digrams
  - Would need a 676 entry frequency table to analyse
  - Versus 26 for a monoalphabetic
  - And correspondingly more ciphertext
- Was widely used for many years
  - E.g. by US & British military in WW1
- It *can* be broken, given a few hundred letters
  - Since still has much of plaintext structure
Classic Techniques

- Substitution Ciphers
  - Cesar Cipher
  - Monoalphabetic Ciphers
  - Polyalphabetic Ciphers
    - Vigenere
    - Vernam
    - One Time Pad
- Transposition Ciphers
- Product Ciphers
Polyalphabetic Ciphers

- Improve security using multiple cipher alphabets
- Flatten frequency distribution
- Use key to select which alphabet is used for each letter of the message
  - Use each alphabet in turn
  - Repeat from start after end of key is reached
Vigenere Ciphers

- Simplest polyalphabetic substitution cipher
- Effectively multiple Caesar ciphers
- Key is multiple letters long \( K = k_1 k_2 \ldots k_m \)
  - \( i^{th} \) letter of key specifies \( i^{th} \) alphabet to use

\[
C_i = (P_i + k_1) \mod 26 \\
\vdots \\
C_m = (P_m + k_m) \mod 26 \\
C_{m+1} = (P_{m+1} + k_1) \mod 26 \\
\vdots
\]
Example: Vigenere Cipher

- Write the plaintext out
- Write the keyword repeated above it
- Use each key letter as a Caesar cipher key
- Encrypt the corresponding plaintext letter

key #: 342415198214342415198214342415198214
key: deceptive deceptive deceptive
plaintext: wearediscoveredsaveyourself
ciphertext: ZICVTWQNGRZGVTVAVZHCQYGLMGJ
Security of Vigenere Cipher

- Have multiple ciphertext letters for each plaintext letter
- Hence letter frequencies are obscured
- *But not totally lost!*


Kasiski Method

- Repetitions in ciphertext give clues to period
  - Find same plaintext an exact period apart
  - Which results in the same ciphertext

  key: deceptive deceptive deceptive
  plaintext: wearediscovered saveyourself
  ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ

- Suggests key size of 3 or 9
- Then attack each monoalphabetic cipher individually using same techniques as before
Vernam Cipher

- Ultimate defense is to use a key as long as the plaintext
  - With no statistical relationship to it
- Invented by AT&T engineer Gilbert Vernam in 1918
- *Originally proposed using a very long but eventually repeating key*
Vernam Cipher

- Generalization of Vigenere
- Represent messages in binary format
  - E.g., ASCII (google it): A = 10 (decimal) = 1010 (binary)
- Key has the same length as the ciphertext

Key stream generator

Encryption

Decryption
One Time Pad

- Vernam: the key was very long, but …
  - Repeatable

- One Time Pad:
  - Generalization of Vernam
  - The key is used ONLY ONCE!
  - The only “Perfect security” cryptosystem

- Problems:
  - Production of many random keys
  - Distribution of random keys – as long as the message
Classic Techniques

- Substitution Ciphers
  - Cesar Cipher
  - Monoalphabetic Ciphers
  - Polyalphabetic Ciphers
    - Vigenere
    - Vernam
    - One Time Pad
- Transposition Ciphers
- Product Ciphers
Transposition Ciphers

- Transposition or permutation ciphers
- Rearrange the letter order
  - Without altering the actual letters used
- Easily recognizable
  - Have the same frequency distribution as the original text
Rail Fence Cipher

- Write message letters over a number of columns
- Then read off cipher by rows
- **Example:** write message out as:
  
  mematrhtgpry
eftefeteoaat

- Resulting ciphertext
  
  MEMATRHTGPRYETEFETEOAAT
Row Transposition Ciphers

- Write letters of message out in rows over a specified number of columns
- Then reorder the columns according to some key before reading off the rows

Key: 4 3 1 2 5 6 7

attack postpone until w o a m x y z

Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ
Product Ciphers

- Ciphers using substitutions or transpositions are not secure because of language characteristics
- Use several ciphers in succession to make harder:
  - Two substitutions make a more complex substitution
  - Two transpositions make more complex transposition
  - **Product cipher**: a substitution followed by a transposition makes a new much harder cipher
- This is bridge from classical to modern ciphers
Rotor Machines

- Before modern ciphers, rotor machines were most common complex ciphers in use
- Widely used in WW II
  - German Enigma, Allied Hagelin, Japanese Purple
- Implemented a very complex, varying substitution cipher
- Used a series of cylinders, each giving one substitution, which rotated and changed after each letter was encrypted
- With 3 cylinders have $26^3 = 17576$ alphabets
Hagelin Rotor Machine
Enigma Rotor Machine
Symmetric Ciphers

- Stream Ciphers
- Block Ciphers
Stream Ciphers

- Encrypt one bit (byte) at a time
  - Example: Vigenere, Vernam
- Length of key = length of (clear/cipher) text
  - Hard to share between sender and receiver
Block Ciphers

- Encrypt one block of text at a time
  - 64-128 bit long
- Encryption key = Decryption key
  - Shared by sender and receiver

<table>
<thead>
<tr>
<th>Key (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaintext</td>
</tr>
</tbody>
</table>

```
64 bits
```

```
Encryption Algorithm
```

```
64 bits
```

Block Cipher Principles

- n bit input to n bit output
- $2^n$ possible inputs
- Each must produce a unique ciphertext
  - Otherwise encryption is not reversible
  - No decryption possible
Ideal Block Cipher

4-Bit Input

4 to 16 Decoder

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

16 to 4 Encoder

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

4-Bit Output

Need $2^n$ table to encrypt!
Data Encryption Standard (DES)

- Most widely used block cipher in world
- Adopted in 1977 by NBS (now NIST)
  - As FIPS PUB 46
- Encrypts 64-bit data using 56-bit key
- Has seen considerable controversy over its security
IBM developed Lucifer cipher
- Team led by Feistel in late 60’s
- Used 64-bit data blocks with 128-bit key
- Redeveloped as a commercial cipher with input from NSA and others

1973: National Bureau of Standards (NBS) issued request for proposals for a national cipher standard
- IBM submitted their revised Lucifer which was eventually accepted as the DES
DES Controversy

- DES standard is public
- Considerable controversy over design
  - Choice of 56-bit key (vs Lucifer 128-bit)
  - Design criteria were classified
- Subsequent events and public analysis show in fact design was appropriate
- Use of DES has flourished
  - Especially in financial applications
  - Still standardised for legacy application use
- Replaced by AES
DES Encryption
Initial Permutation (IP)

- First step of the data computation
  - IP reorders the input data bits
  - Even bits to LH half, odd bits to RH half
  - Quite regular in structure (easy in h/w)

- Example:

\[
\text{IP}(675a6967 \ 5e5a6b5a) = (ffb2194d \ 004df6fb)
\]
Uses two 32-bit L & R halves

Feistel cipher:
\[ L_i = R_{i-1} \]
\[ R_i = L_{i-1} \oplus F(R_{i-1}, K_i) \]
Feistel Cipher

- Introduced by Horst Feistel
- 16 + 1 rounds

$F$ defined soon

$L_i = R_{i-1}$

$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
Feistel Cipher Structure
DES Structure: Function F

- Expands R to 48-bits using perm E
- Adds to subkey using XOR
- 8 S-boxes to get 32-bit result
- Finally permutes using 32-bit perm P
Symmetric Key Crypto

- Symmetric Ciphers
- Multiple Encryption
- Modes of Operation
Multiple Encryption and DES

- Uses 56-bit keys to encrypt 64 bit blocks
- Differential cryptanalysis – $O(2^{47})$ encryptions
- Linear cryptanalysis – $O(2^{43})$ encryptions

Can we make DES withstand attacks without changing its structure? 

Yes!
Double DES

2 DES with keys $K_1$ and $K_2$: $C = E_{K_2}(E_{K_1}(P))$
2 DES: Meet-in-the-Middle

- 2 DES uses two keys: $56+56=112$ bits
- Is the strength $2^{56}$ of DES?  
  - NO !!!!
- Given $P$ and $C$
  - Encrypt $P$ for all possible $2^{56}$ values of $K_1$
  - Store in table $T$: pairs $(K_1, E_{K_1}(P))$
  - Decrypt $C$ for all possible $2^{56}$ values of $K_2$
  - Search $D_{K_2}(C)$ in table $T$
  - Success when $E_{K_1}(P) = D_{K_2}(C)$
- Attack takes $O(2^{56})$ steps – similar to DES
Triple DES: Two Keys

- Must use 3 encryptions
- But can use 2 keys with E-D-E sequence
  - $C = E_{K_1} (D_{K_2} (E_{K_1}(P)))$
  - If $K_1 = K_2$ then equivalent with single DES
- Standardized in ANSI X9.17 & ISO8732
- No current known practical attacks
  - Several proposed impractical attacks might become basis of future attacks
Triple DES: Three Keys

- Can use Triple-DES with Three-Keys to avoid even these
  - $C = E_{K3}(D_{K2}(E_{K1}(P)))$
- Has been adopted by some Internet applications
  - PGP, S/MIME
Symmetric Key Crypto

- Symmetric Ciphers
- Multiple Encryption
- Modes of Operation
Modes of Operation

- Block ciphers encrypt fixed size blocks
  - DES encrypts 64-bit blocks with 56-bit key

- Need to encrypt and decrypt arbitrary amounts of data in practice

- NIST SP 800-38A defines 5 modes
  - Electronic Code Book: ECB
  - Cipher Block Chaining: CBC
  - Cipher Feedback: CFB
  - Output Feedback: OFB
  - Counter Mode: CTR

- Can be used with any block cipher
Electronic Code Book (ECB)

- Split message into blocks of length $b$ (e.g., 64 bits)
- Use the same key to encrypt each block
  - Each block is mapped into a unique value like a codebook

\[ P_1 \xrightarrow{K} \text{DES Encrypt} \xrightarrow{} C_1 \]

\[ \cdots \]

\[ P_s \xrightarrow{K} \text{DES Encrypt} \xrightarrow{} C_s \]
ECB Decryption

- Weakness due to independent encryptions
  - Same bit repeated each b positions
- Main use is sending a few blocks of data
  - E.g., shared keys
Cipher Block Chaining (CBC)

- Use Initial Vector (IV) to start process
- Chain current cipher block into next encryption

```
P_1 \oplus IV \rightarrow DES Encrypt \rightarrow C_1

K \rightarrow DES Encrypt

P_2 \oplus C_1 \rightarrow DES Encrypt \rightarrow C_2

\ldots

\ldots (s blocks)
```
CBC: Decryption

\[ K \rightarrow \text{DES Decrypt} \rightarrow C_1 \]

\[ IV \rightarrow \oplus \rightarrow P_1 \]

\[ K \rightarrow \text{DES Decrypt} \rightarrow C_2 \]

\[ \oplus \rightarrow P_2 \]

\[ (s \text{ blocks}) \]

CBC: Decryption
CBC Discussion

- Padding:
  - Message length may not be divisible by \( b \)
  - End of message must handle a possible last short block
  - Random padding
  - *May require an extra entire block over those in message*

- Need **Initialization Vector (IV)**
  - Must be known to sender & receiver
  - May be sent encrypted in ECB mode before rest of message
Stream Modes of Operation

- Block modes (ECB, CBC) encrypt entire block
  - *May need to operate on smaller units: Why?*
    - Real time data
- Convert block cipher into stream cipher
  - Cipher feedback (CFB) mode
  - Output feedback (OFB) mode
  - Counter (CTR) mode
Cipher Feedback Mode (CFB)

- Message is treated as a stream of bits
- Take $s$ bits at a time; $s < b$

$IV \ (b \ bits)$

$K$

DES Encrypt

$s \ bits$ Discard

$P_1 \ (s)$

$C_1$

$K$

DES Encrypt

$s \ bits$ Discard

$P_2 \ (s)$

$C_2$

IV Shift $s$ bits

... (so on)
More on CFB

- Decryption similar ...
- Appropriate when data arrives in bits/bytes
Counter Mode (CTR)

- **b** is block size

\[ \text{Counter}_1, \text{Counter}_2, \ldots \]

- \( C_{1} = P_{1}(b) \oplus \text{Encrypt}(K, \text{Counter}_1) \)
- \( C_{2} = P_{2}(b) \oplus \text{Encrypt}(K, \text{Counter}_2) \)
- \( \text{Counter}_2 = \text{Counter}_1 + 1, \ldots , \text{Counter}_n = \text{Counter}_{n-1} + 1 \)
CTR (cont’d)

- The initial $Counter_1$ is random
- Decryption is identical to encryption
  - $Counter_1$ must be known
- Counters should not be reused
  - This includes across multiple messages
CTR Advantages

- Hardware/software efficient
  - Can process blocks in parallel
- Preprocessing
  - Precompute encryptions of counters
- Random access
  - Can encrypt/decrypt any block
CTR Advantages (cont’d)

- Provable security
  - At least as secure as the other modes
- Simplicity
  - Encryption = Decryption
Summary

- Symmetric Ciphers
- Multiple Encryption
- Cipher Modes of Operation