Overview

- What is cryptography?
- Classic cryptosystems
  - The Caesar cipher
  - Monoalphabetic replacement cipher
  - The one-time pad
- Types of cryptosystems
  - Codes vs. ciphers
  - Symmetric-key vs. asymmetric-key
  - Hybrid cryptosystems
What is Cryptography?

- **Text defines Cryptography** as the science of designing and analyzing cryptosystems which are used to disguise messages so that only certain people can see through the disguise.
- A classic cryptosystem: the Caesar cipher
  - Replace every ‘A’ in the message with a ‘D’
  - Replace every ‘B’ in the message with a ‘E’
  - Replace every ‘C’ in the message with a ‘F’
  - Etc.
The Caesar Cipher

- Camouflage the message “ATTACK AT DAWN” by writing “DWWDFN DW GDZQ”
- “ATTACK AT DAWN” is plaintext
- “DWWDFN DW GDZQ” is ciphertext

- Encryption is the process used to convert plaintext into ciphertext
- Decryption is the process used to convert ciphertext into plaintext
The Key to a Cryptosystem

- Assumptions:
  - Encryption and decryption algorithms are public
  - Their results depend on some value known as a **key**
  - Protection is based solely on the secrecy of the key
  - Encryption for the Caesar cipher = “shift forward by \( n \)”
  - Decryption for the Caesar cipher = “shift backwards by \( n \)”
  - The key for the Caesar cipher is \( n \)
    - Encryption: \( C_i = (P_i + n) \mod 26 \)
    - Decryption: \( P_i = (C_i - n) \mod 26 \)
The Keypspace for a Cryptosystem

• For the Caesar cipher, any value from the set \{1, 2, ..., 25\} can be a key
• The set of usable keys is referred to as a cryptosystem’s **keyspace**
• Cryptosystems with a small keyspace are vulnerable to a **brute-force search** for the proper key
What is Cryptanalysis?

• **Cryptanalysis** the science of attacking cryptosystems
  – Deduce the key and/or
  – recover plaintext

• Assume adversary knows the ciphertext and encryption algorithm
Cryptanalysis of the Caesar Cipher

- Ciphertext = “GRR MGAR OY JOBOJKJ OT ZNXKK VGXZY”
- Perform decryption with each possible key:
  - Plaintext (if key is 1): FQQ LFZQ NX INANIIJ NS YMWJJ UFWYX
  - Plaintext (if key is 2): EPP KEYP MW HMZMHII MR XLVII TEVXW
  - Plaintext (if key is 3): DOO JDXO LV GLYLGHG LQ WKUHII SDUWV
  - Plaintext (if key is 4): CNN ICWN KU FKXKFGF KP VJITGG
  - Plaintext (if key is 5): BMM HBVM JT EJWJEFE JO UISFF QBSUT
  - Plaintext (if key is 6): ALL GAUL IS DIVIDED IN THREE PARTS
  - Plaintext (if key is 7): ZKK FZTK HR CHUHCDC HM SGQDD OZQSR
    ...
  - Plaintext (if key is 26): GRR MGAR OY JOBOJKJ OT ZNXKK
    VGXZY
- Only one of the plaintexts above (the one corresponding to a key of 6) makes sense.
The Monoalphabetic Replacement Cipher

• Similar to the Caesar cipher but with much larger keyspace
• A key is any permutation of the 26 letters of the alphabet
  – Example:
    JQPLMZKOWHANXIEURYTGSDVCB
• Defines a cipher alphabet:

| 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 |
| J | Q | P | L | M | Z | K | O | W | H | A | N | X | I | E | U | R | Y | T | G | S | F | D | V | C | B |
The Monoalphabetic Replacement Cipher - Encryption

- Plaintext (by Thomas Jefferson):
  - “I prefer freedom with danger to slavery with ease.”
- Cipher alphabet

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

- Encryption: replace each plaintext letter with the corresponding cipher letter from the cipher alphabet
- Examples:
  - Replace every “A” in the plaintext with a “J”
  - Replace every “B” in the plaintext with a “Q”
  - Replace every “C” in the plaintext with a “P”
  - Etc.
The Monoalphabetic Replacement Cipher – Encryption (cont)

- Plaintext:
  - “I prefer freedom with danger to slavery with ease.”

- Cipher alphabet:

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

- Ciphertext:
  - “W uymzmy zymmlex dwgo ljkmy ge tnjfmyc dwgo mjtm.”
The Monoalphabetic Replacement Cipher - Decryption

• Ciphertext:
  – “W uymzmy zymmlex dwgo ljikmy ge tnjfmcy dwgo mjtm.”

• Cipher alphabet

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

• Decryption: replace each plaintext letter with the corresponding cipher letter from the cipher alphabet

• Plaintext:
  – “I prefer freedom with danger to slavery with ease.”
The Monoalphabetic Replacement Cipher - Keyspace

• Key = any permutation of the 26 letters of the alphabet

• Keyspace contains 26! elements
  – 403,291,461,126,605,635,584,000,000

• Exhaustive search at one trillion keys per second takes:
  – 400 trillion seconds
  – More than 12 million years

• It is fairly easy to perform cryptanalysis on this cipher, but not using exhaustive search
The Monoalphabetic Replacement Cipher – Weak Keys

• Some keys result in better-disguised ciphertext than others:
  – Using JQPLMZKOWHANXIEURYTGSFDVCB as a key gives:
    “W uymzmy zymmlex dwgo ljikmy ge tnjfmyc dwgo mjtm.”
  – Using ABCDEFGHIJKLMNOPQRSTUVWXYZ as a key gives:
    “I prefer freedom with danger to slavery with ease.”
  – Using ABCDEFGHIJKLMNOPQRSTUVWXZY as a key gives:
    “I prefer freedom with danger to slaverz with ease.”

• Keys that produce weak ciphertext are called **weak** keys

• Weak keys need not be a problem so long as:
  – They are not used
  – The vast majority of the keys are not weak
One-Time Pads

• Unbreakable
• Sender and receiver must generate a large, non-repeating set of truly random key letters
  – E.g. IPKLPFSFHGQYPWKSVCX…
• Sender uses each key letter on the pad to encrypt one letter of plaintext
  – \( C_i = (P_i + K_i) \mod 26 \)
• Receiver uses each key letter on the pad to decrypt one letter of ciphertext
  – \( P_i = (C_i - K_i) \mod 26 \)
One-Time Pad Encryption - Example

• One time pad:
  IPKLPFHSFQYPWPKQMSVCK...

• Plaintext:
  “ATTACKATDAWN”

• Ciphertext:
  “JJEMSDGBKRVD”

  A (1) + I (9) mod 26 = J (10)  A (1) + F (6) mod 26 = G (7)
  T (20) + P (16) mod 26 = J (10)  T (20) + H (8) mod 26 = B (2)
  T (20) + K (11) mod 26 = E (5)  D (4) + G (7) mod 26 = K (11)
  A (1) + L (12) mod 26 = M (13)  A (1) + Q (17) mod 26 = R (18)
  C (3) + P (16) mod 26 = S (19)  W (23) + Y (25) mod 26 = V (22)
  K (11) + S (19) mod 26 = D (4)  N (14) + P (16) mod 26 = D (4)
One-Time Pad Decryption - Example

• One time pad:
  IPKLPSFHGQYPWKQMSVCX…

• Ciphertext:
  “JJEMSDGBKRVD”

• Plaintext:
  “ATTACKATDAWN”
  J (10) - I (9) mod 26 = A (1)
  J (10) - P (16) mod 26 = T (20)
  E (5) - K (11) mod 26 = T (20)
  .
  .
  .
One-Time Pad - Security

• Why is it an unbreakable encryption algorithm?
  – Assume the adversary doesn’t know any of the key letters on the one-time pad
  – If they were generated truly randomly then all key letters are equally likely in each position
  – So when the adversary sees the ciphertext, “JJEMSDGBKVRD”
  – **All plaintexts are equally possible:**
    JJEMSDGBKVRD = ATTACKATDAWN for IPKLPSFHGQYP
    JJEMSDGBKVRD = ELVISISALIVE for EXIDZUNAYIZY
    .
    .
One-Time Pad - Security (cont)

- Every plaintext message is equally possible
- No way for an adversary to determine which plaintext is correct
- A truly random key sequence added to a nonrandom plaintext produces a truly random ciphertext
- No algorithm will enable the adversary to choose the proper plaintext with better than random probability
One-Time Pads - Drawbacks

- Key must be as long as the message
- Security depends on adversary never obtaining a copy of the pad
  - Pad must be distributed securely to sender and receiver
  - Pad must be destroyed immediately after use to lessen the likelihood that old messages will be compromised
- Security depends on using the cryptosystem properly
  - Pad must be generated truly randomly (pseudo-random won’t due)
  - No part of the pad can ever be reused
Types of Cryptosystems

• Codes, ciphers, or a combination of the two
• Ciphers (e.g. the Caesar cipher)
  – Transform each plaintext block into a ciphertext block
  – **Block** is a fixed-size unit on which a cryptosystem operates
    • Single character (e.g. Caesar cipher)
    • Two or more characters
Ciphers

- **Substitution** ciphers apply some function to the plaintext block and key to produce a block of ciphertext which replaces the plaintext (e.g. the Caesar cipher)
- **Transposition** ciphers shuffle the blocks into a new order that depends on the plaintext block and key

<table>
<thead>
<tr>
<th>A</th>
<th>T</th>
<th>T</th>
<th>A</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>A</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>A</td>
<td>W</td>
<td>N</td>
<td></td>
</tr>
</tbody>
</table>

= “AKDT ATAWATNC”

<table>
<thead>
<tr>
<th>A</th>
<th>K</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>T</td>
<td>A</td>
<td>W</td>
</tr>
<tr>
<td>A</td>
<td>T</td>
<td>N</td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

= “ATTACK AT DAWN”
Codes

- Sender and receiver each have a copy of a **codebook** which specifies one or more **codewords** for each word that might be used in a message:

<table>
<thead>
<tr>
<th>Word</th>
<th>Codeword</th>
</tr>
</thead>
<tbody>
<tr>
<td>AT</td>
<td>September</td>
</tr>
<tr>
<td>ATTACK</td>
<td>March</td>
</tr>
<tr>
<td>ATTACK</td>
<td>December</td>
</tr>
<tr>
<td>DAWN</td>
<td>April</td>
</tr>
<tr>
<td>DAWN</td>
<td>October</td>
</tr>
<tr>
<td>(null)</td>
<td>July</td>
</tr>
<tr>
<td>(null)</td>
<td>January</td>
</tr>
</tbody>
</table>
Codes – Encryption and Decryption

• Plaintext:
  – “ATTACK AT DAWN”

• Ciphertext:
  – “March September October” or
  – “March September April” or
  – “July December January September April July” or …

• Codewords can be random numbers, strings of characters, or other symbols
Types of Cryptosystems (cont)

• Symmetric-key
  – Same key used for encryption and decryption
  – Typically used for bulk encryption

• Asymmetric-key (or public-key)
  – Different key used for encryption and decryption
  – Usually not used for bulk encryption

• Hybrid cryptosystems
Symmetric-key Cryptosystems

• Standard use of a symmetric-key cryptosystem:
  – Sender and receiver agree on a secret key
    • Must be done securely!
  – Messages are encrypted by the sender with the shared key and decrypted by the receiver with the shared key
    – Note: Users need to have a previously-established shared secret to communicate securely
Public-Key Cryptosystems

- Standard use of a public-key cryptosystem:
  - Generate a public-key/private-key pair
    - Disseminate your public key widely
    - Keep your private key secret
  - Anybody can encrypt a message to you using your public key
  - Only you can decrypt the message using your private key

- Note: unlike symmetric-key cryptosystems, users don’t need to have a previously-established shared secret to communicate securely
Public-Key Cryptosystems (cont)

• Standard use of a public-key cryptosystem:
  – Digital signatures - proof of authorship of a document or agreement with its contents
    • User encrypts a document with his private key to create a digital signature
    • Anybody can verify the digital signature by using the signer’s public key
    • Only the signer can produce his signature, and he can’t reasonably claim he didn’t sign a document bearing his signature
  – Note: unlike symmetric-key cryptosystems, users can create authentic, unforgeable, nonreusable, nonrepudiable digital signatures
Public-Key Cryptosystems (cont)

• In order for a public-key cryptosystem to work:
  – For every message, $M$, $\text{Decrypt}(\text{Encrypt}(M, A_{\text{Public}}), A_{\text{Private}}) = M$
  – For every pair of users, $A$ and $B$, $(A_{\text{Public}}, A_{\text{Private}})$ and $(B_{\text{Public}}, B_{\text{Private}})$ must be distinct
  – Deriving $A_{\text{private}}$ from $A_{\text{Public}}$ or the plaintext from the ciphertext is difficult
  – Key generation, encryption, and decryption routines must be relatively fast
Public-Key Cryptosystems - Problems

- Problem #1 - Man in the Middle:
  - Recall - everybody should know A’s public key
  - So if B wants to send a message, M, to A then B needs to encrypt M with $A_{Public}$
  - What if an adversary, C, is able to trick B into thinking that $C_{Public}$ is $A_{Public}$?

  $B \xrightarrow{\text{Encrypt}(M,C_{Public})} C \xrightarrow{\text{Encrypt}(M,A_{Public})} A$

  - A and B think their messages are secure, but C can read them
  - Public-key cryptography depends heavily on knowing to whom a public key belongs
Public-Key Cryptosystems - Problems

• Problem #2 - Known Ciphertext:
  – Recall - everybody should know A’s public key
  – So if C sees an encrypted message, Encrypt(M, A_{Public}) from B to A
    • C can choose a message, M’
    • Encrypt M’ with A’s public key to get Encrypt(M’, A_{Public})
    • Compare Encrypt(M’, A_{Public}) with Encrypt(M, A_{Public}), if they match then C knows the message B sent to A
  – This is a serious problem if the number of possible plaintext messages is small enough to allow exhaustive search
Hybrid Cryptosystems

- Symmetric-key cryptosystems:
  - Good for bulk data, but require shared secrets
- Public-key cryptosystems:
  - Don’t require any shared secrets, but too slow for bulk encryption
- Hybrid cryptosystems:
  - Given a message $M$
  - Choose a key, $K$, at random to be used with a symmetric-key algorithm
  - Encrypt $K$ with the recipient’s public key
  - Encrypt $M$ with $K$
  - Send to recipient:

  $\text{Encrypt}(K, A_{\text{Public}}) \quad \text{Encrypt}(M, K)$
## Hybrid Cryptosystems (cont)

- Hybrid cryptosystems:

<table>
<thead>
<tr>
<th>Encrypt($K, A_{Public}$)</th>
<th>Encrypt($M, K$)</th>
</tr>
</thead>
</table>

- Recipient decrypts first part of the message with his/her private key to learn $K$
- Recipient uses $K$ to decrypt the remainder of the message
- Result: Doesn’t require any shared secrets, and good for bulk encryption
Summary

- **Cryptography** is the science of designing and analyzing cryptosystems which are used to disguise messages so that only certain people can see through the disguise.
- **Cryptanalysis** is the science of attacking cryptosystems.
- Classic cryptosystems include the **Caesar cipher**, **monoalphabetic replacement cipher**, and **one-time pad**.
- **Symmetric-key** cryptosystems are useful for bulk data encryption but required a shared secret.
- **Public-key** cryptosystems are much slower but don’t require shared secrets and support digital signatures.
- **Hybrid** cryptosystems are good for bulk encryption and don’t require any shared secrets.