Cryptography

- Cryptography
  - Original meaning: The art of secret writing
  - Becoming a science that relies on mathematics (number theory, algebra)
  - Process data into unintelligible form, reversible, without data loss
  - Usually one-to-one (not compression)
Encryption/Decryption

- Plaintext: a message in its original form
- Ciphertext: a message in the transformed, unrecognized form
- Encryption: the process that transforms a plaintext into a ciphertext; also known as encode and encipher
- Decryption: the process that transforms a ciphertext to the corresponding plaintext; also known as decode and decipher
- Key: the value used to control encryption/decryption
- Cryptosystem: a system for encryption and decryption

Cryptanalysis

- Ciphertext only:
  - Analyze only with the ciphertext
  - Example: Exhaustive search until “recognizable plaintext”
  - Smarter ways available
- Known plaintext:
  - Secret may be revealed (by spy, time), thus <ciphertext, plaintext> pair is obtained
  - Great for mono-alphabetic ciphers
Cryptanalysis (Cont’d)

- Chosen plaintext:
  - Choose text, get encrypted
  - Useful if limited set of messages
- Chosen ciphertext:
  - Choose ciphertext
  - Get feedback from decryption, etc.

Simple Forms of Encryption

- Substitutions
  - One letter is replaced with another
- Transpositions
  - Also called permutations
  - The order of the letters is rearranged
- Building blocks of modern cryptographic algorithms
Substitution Ciphers

- Monoalphabetic cipher (simple substitution)
  - Use a correspondence table
  - Substitute a character or symbol for each character of the original message
  - Example: Caesar cipher
    - Replace each letter with the one 3 letters later

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>b</td>
<td>c</td>
<td>d</td>
<td>e</td>
<td>f</td>
<td>g</td>
<td>h</td>
<td>i</td>
<td>j</td>
<td>k</td>
<td>l</td>
<td>m</td>
</tr>
<tr>
<td>n</td>
<td>o</td>
<td>p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>r</td>
<td>s</td>
<td>t</td>
<td>u</td>
<td>v</td>
<td>w</td>
<td>x</td>
<td>y</td>
<td>z</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Exercise
  - E ("COMPUTER SCIENCE") →
  - D ("qf vwdwh") →

Caesar Cipher

- Cryptanalysis of Caesar cipher
  - Can be done by guessing

- Clues
  - Break between two words is preserved
    - You can try common letters starting or ending a word
  - Double letters are preserved
  - Always use the same mapping
  - Exercise:
    - wklv phv vdjh lv qr wrr kdu g wr euhdn
Other Substitutions

- In general
  - The alphabet is scrambled
  - Each plaintext letter maps to a unique ciphertext letter
  - A substitution table can be defined using a permutation
    • A permutation is a reordering of the elements of a sequence

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>x</td>
<td>f</td>
<td>t</td>
<td>h</td>
<td>i</td>
<td>w</td>
<td>k</td>
<td>y</td>
<td>m</td>
<td>n</td>
<td>o</td>
<td>p</td>
</tr>
<tr>
<td>N</td>
<td>O</td>
<td>P</td>
<td>Q</td>
<td>R</td>
<td>S</td>
<td>T</td>
<td>U</td>
<td>V</td>
<td>W</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

Cryptanalysis of Substitution Ciphers

- Ad hoc clues
  - Short words, words with repeated patterns, common initial and final letters

- Language specific knowledge
  - Frequency of letters
    • E, T, O, and A occur far more often than J, Q, X, and Z
  - Letter patterns
    • th, er, en, ss, st, …
One-Time Pads

- Encrypt plaintext with a large, non-repeating set of keys
  - Absolute synchronization between sender and receiver
  - Unlimited number of keys

Vernam cipher

Book Cipher

- Use book, piece of music, or other object with which structure can be analyzed
  - Both sender and receiver need access to identical objects
  - Example: book cipher with Vigenère tableau
    - Key: I am, I exist, that is certain.
    - Plaintext: MACHINES CANNOT THINK

\[
\begin{array}{c}
\text{iamie xistt hatis cert} \\
\text{MACHI NESCA NNOTT HINK} \\
\text{Uaopm kmkvt unbhl jmed}
\end{array}
\]
Cryptanalysis of Book Cipher

- Flaw of book cipher
  - Distributions of both key and message cluster around high frequency letters
  - Example
    - A, E, O, T, N, I account for 50% of all letters
    - Probability of both key and plaintext letters are one of them: 0.25

- Cryptanalysis
  - Look for intersections of the above six letters
  - For each cipher text letter, identify the possible plain text letter from those intersections

```
uaopm kmkvt unbhl jmed
0 I I T NTT IE
T T T T
```

Correct prediction underlined
Transpositions (Permutations)

- Letters of the message are rearranged
  - Aim to break established patterns
- Confusion and diffusion
  - Confusion
    - Make it difficult to determine how message and key are transformed into cipher text
    - Complex relationship between plaintext, key, and ciphertext
    - Done through substitution
  - Diffusion
    - Widely spread the information from the message or key across the cipher text
    - Done through transposition (permutation)

Columnar Transpositions

- Rearrange characters of the plain text into columns

<table>
<thead>
<tr>
<th>Key:</th>
<th>4</th>
<th>3</th>
<th>1</th>
<th>2</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plaintext:</td>
<td>A</td>
<td>T</td>
<td>T</td>
<td>A</td>
<td>C</td>
<td>K</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>O</td>
<td>S</td>
<td>T</td>
<td>P</td>
<td>O</td>
<td>N</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>D</td>
<td>U</td>
<td>N</td>
<td>T</td>
<td>I</td>
<td>L</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>O</td>
<td>A</td>
<td>M</td>
<td>X</td>
<td>Y</td>
<td>Z</td>
</tr>
</tbody>
</table>

Cipher text: ______________________________________
Cryptanalysis of Transpositions

- Diagram analysis
  - Frequent diagram
    - Patterns of pairs of adjacent letters
    - RE, EN, ER, NT, ...
  - Frequent trigrams
    - Groups of three letters
    - ENT, ION, AND, ING, ...
  - Infrequent diagrams and trigrams
    - VK and QP

Cryptanalysis by Diagram Analysis

- Confirms it is a transposition
  - Compute the letter frequencies
- Find adjacent columns
  - Try different column sizes
  - Look for common diagrams
  - Verify possible matches for different positions
- Rely heavily on a human’s judgment of what “looks right”
Product Cipher

- Product cipher
  - Combination of two ciphers
  - Modern ciphers: interleaved substitutions and transpositions (permutations)
  - $S \rightarrow P \rightarrow S \rightarrow P \rightarrow \ldots$
- But
  - How about $S \rightarrow P \rightarrow S \rightarrow S \rightarrow P \rightarrow \ldots$
  - How about $S \rightarrow P \rightarrow P \rightarrow S \rightarrow \ldots$

“Good” Encryption algorithms

- What does it mean for a cipher to be “good”?
  - Meaning of “good” depends on intended use of the cipher
  - Commercial applications
  - Military applications
Characteristics of “Good” Ciphers

- Shannon’s principles
  - The amount of secrecy needed should determine the amount of labor appropriate for the encryption and decryption
  - The set of keys and the enciphering algorithm should be free from complexity
    - No restrictions on keys or plain text; keys should be short
  - The implementation of the process should be as simple as possible
    - Formulated with hand encryption in mind
    - Implementation on a computer need not be simple, as long as the time complexity is tolerable

- Shannon’s Principles (Cont’d)
  - Errors in ciphering should not propagate and cause corruption of further information in the message
    - No error propagation
  - The size of the enciphered text should be no larger than the text of the original message
    - Dramatic cipher expansion in size does not carry more information, but
    - It gives the cryptanalyst more data to infer patterns
Security of An Encryption Algorithm

• Unconditionally secure
  – It is impossible to decrypt the ciphertext
  – One-time pad (the key is as long as the plaintext)
    \[ C_i = P_i \oplus K_i \]

• Computationally secure
  – The cost of breaking the cipher exceeds the value of the encrypted information
  – The time required to break the cipher exceeds the useful lifetime of the information

Secret Keys v.s. Secret Algorithms

• Security by obscurity
  – We can achieve better security if we keep the algorithms secret
  – Hard to keep secret if used widely
  – Reverse engineering, social engineering

• Publish the algorithms
  – Security of the algorithms depends on the secrecy of the keys
  – Less unknown vulnerability if all the smart (good) people in the world are examine the algorithms
Secret Keys v.s. Secret Algorithms (Cont’d)

• Commercial world
  – Published
  – Wide review, trust

• Military
  – Keep algorithms secret
  – Avoid giving enemy good ideas
  – Military has access to the public domain knowledge anyway.

Types of Cryptography

• Number of keys
  – Hash functions: no key
  – Secret key cryptography: one key
  – Public key cryptography: two keys - public, private

• The way in which the plaintext is processed
  – Block cipher: divides input elements into blocks
  – Stream cipher: process one element (e.g., bit) a time
Secret Key Cryptography

- Same key is used for encryption and decryption
- Also known as
  - Symmetric cryptography
  - Conventional cryptography

Secret Key Cryptography (Cont’d)

- Basic technique
  - Product cipher
  - Multiple applications of interleaved substitutions and permutations
- Cipher text approximately the same length as plaintext
Stream and Block Ciphers

• Stream ciphers
  – Convert one symbol of plaintext immediately into a symbol of ciphertext
    • A symbol: a character, a bit
  – Examples
    • Substitution ciphers discussed earlier
    • Modern example: RC4

Stream and Block Ciphers (Cont’d)

• Block cipher
  – Encrypt a group of plaintext symbols as on block
  – Examples
    • Columnar transposition
    • Modern examples: DES, AES
Applications of Secret Key Cryptography

- Transmitting over an insecure channel
  - Challenge: How to share the key?
- Secure Storage on insecure media
- Authentication
  - Challenge-response
  - To prove the other party knows the secret key
  - Must be secure against chosen plaintext attack
- Integrity check
  - Message Integrity Code (MIC)
  - Also called Message Authentication Code (MAC)

Public Key Cryptography

plaintext → encryption → ciphertext → decryption → plaintext

Public key

Private key

- Invented/published in 1975
- A public/private key pair is used
  - Public key can be publicly known
  - Private key is kept secret by the owner of the key
- Much slower than secret key cryptography
- Also known as
  - Asymmetric cryptography
Public Key Cryptography (Cont’d)

- Another mode: digital signature
  - Only the party with the private key can create a digital signature.
  - The digital signature is verifiable by anyone who knows the public key.
  - The signer cannot deny that he/she has done so.

Applications of Public Key Cryptography

- Data transmission
  - Alice encrypts $m_a$ using Bob’s public key $e_B$, Bob decrypts $m_a$ using his private key $d_B$.
- Storage
  - Can create a safety copy: using public key of trusted person.
- Authentication
  - No need to store secrets, only need public keys.
  - Secret key cryptography: need to share secret key for every person to communicate with.
Applications of Public Key Cryptography (Cont’d)

- Digital signatures
  - Sign hash $H(m)$ with the private key
    - Authorship
    - Integrity
    - Non-repudiation: can’t do with secret key cryptography

- Key exchange
  - Establish a common session key between two parties

Hash Algorithms

- Message of arbitrary length $\rightarrow$ Hash $H$ $\rightarrow$ A fixed-length short message

- Also known as
  - Message digests
  - One-way transformations
  - One-way functions
  - Hash functions

- Length of $H(m)$ much shorter than length of $m$
- Usually fixed lengths: 128 or 160 bits
Hash Algorithms (Cont’d)

• Desirable properties of hash functions
  – **Performance**: Easy to compute $H(m)$
  – **One-way property**: Given $H(m)$ but not $m$, it’s difficult to find $m$
  – **Weak collision free**: Given $H(m)$, it’s difficult to find $m'$ such that $H(m') = H(m)$.
  – **Strong collision free**: Computationally infeasible to find $m_1, m_2$ such that $H(m_1) = H(m_2)$.

Applications of Hash Functions

• Primary application
  – Generate/verify digital signature

```
Message m
\[ H \rightarrow H(m) \rightarrow \text{Sign} \rightarrow \text{Signature Sig(H(m))} \]
Private key

Message m
\[ H \rightarrow H(m) \rightarrow \text{Verify} \rightarrow \text{Yes/No} \]
Public key
```
Applications of Hash Functions (Cont’d)

• Password hashing
  – Doesn’t need to know password to verify it
  – Store $H(password + salt)$ and salt, and compare it with the user-entered password
  – Salt makes dictionary attack more difficult

• Message integrity
  – Agree on a secret key $k$
  – Compute $H(m|k)$ and send with $m$
  – Doesn’t require encryption algorithm, so the technology is exportable

DES (Data Encryption Standard)

• Officially adopted in 1976
• Expired in 1998
• Key: 64 bit quantity=8-bit parity+56-bit key
  – Every 8th bit is a parity bit.
• 64 bit input, 64 bit output.
DES Top View

- 64-bit Input
- Permutation
  - Initial Permutation
  - Round 1 (48-bit K1)
  - Round 2 (48-bit K2)
  - Round 16 (48-bit K16)
- Swap 32-bit halves
- Permutation
- Final Permutation
- 64-bit Output
- 56-bit Key
- Generate keys

Bit Permutation (1-to-1)

Input:
0 0 1 0 ...

Output:
1 0 1 1 ...

1 bit
Initial and Final Permutations

- **Initial permutation (IP)**
- View the input as M: 8 X 8 bit matrix
- Transform M into M’ in two steps
  - Transpose row x into column (9-x), 0<x<9
  - Apply permutation on the rows:
    • For even row y, it becomes row y/2
    • For odd row y, it becomes row (5+y/2)
- **Final permutation FP = IP⁻¹**

Per-Round Key Generation

- Initial Permutation of DES key
  - $C_{i-1}$ 28 bits
  - $D_{i-1}$ 28 bits
  - Circular Left Shift
  - Permutation with Discard
  - Final result: $C_i$, $D_i$ 28 bits

Key: $K_i$ 48 bits

- Round 1, 2, 9, 16: single shift
- Others: two bits
A DES Round

One Round Encryption

Mangler Function

E-Boxes

S-Boxes

P

32 bits

32 bits

48 bits

K_i

32 bits

32 bits

E Box of DES (Expansion Permutation)

- How is the E box defined
- Each row expands from 4 bits to 6 bits

<table>
<thead>
<tr>
<th>32</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
<td>17</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>18</td>
<td>19</td>
<td>20</td>
<td>21</td>
</tr>
<tr>
<td>20</td>
<td>21</td>
<td>22</td>
<td>23</td>
<td>24</td>
<td>25</td>
</tr>
<tr>
<td>24</td>
<td>25</td>
<td>26</td>
<td>27</td>
<td>28</td>
<td>29</td>
</tr>
<tr>
<td>28</td>
<td>29</td>
<td>30</td>
<td>31</td>
<td>32</td>
<td>1</td>
</tr>
</tbody>
</table>
Another View of the Mangler Function

The permutation produces “spread” among the chunks/S-boxes!

S-Box (Substitute and Shrink)

- 48 bits ==> 32 bits. (8*6 ==> 8 *4)
- 2 bits used to select amongst 4 permutations for the rest of the 4-bit quantity

\[ S_i \]

2 bits row

4 bits column

\[ i = 1, \ldots, 8. \]

an integer between 0 and 15.
The First S Box S1

Each row and column contain different numbers.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>…</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>14</td>
<td>4</td>
<td>13</td>
<td>1</td>
<td>2</td>
<td>15</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>15</td>
<td>7</td>
<td>4</td>
<td>14</td>
<td>2</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1</td>
<td>14</td>
<td>8</td>
<td>13</td>
<td>6</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>15</td>
<td>12</td>
<td>8</td>
<td>2</td>
<td>4</td>
<td>9</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Example: input: 100110 output: ???

DES Standard

- Cipher Iterative Action
  - Input: 64 bits
  - Key: 48 bits
  - Output: 64 bits

- Key Generation Box
  - Input: 56 bits
  - Output: 48 bits

One round (Total 16 rounds)
Avalanche Effect

- A small change in either the plaintext or the key should produce a significant change in the ciphertext
- DES has a strong avalanche effect
- Example
  - Plaintexts: 0X0000000000000000 and 0X8000000000000000
  - Same key: 0X016B24621C181C32
  - 34 bits difference in cipher-texts
  - Similar result with same plaintext and slightly different keys

Concerns about DES

- Key space problem: 56 bit key ($2^{56}$)
  - DESCHALL recovered RSA challenge I key on June 17, 1997 (6 month into the contest)
  - $.25m (total cost), July 15, 1998, RSA DES challenge II key recovered in 56 hours
- Cryptanalysis
  - Sixteen weak and semi-weak keys:
    - Differential cryptanalysis require less tries using chosen plaintext/ciphertext [Biham, 1993]
      - Effective up to 15 rounds
      - DES is well designed to defeat differential analysis
    - Linear cryptanalysis requires only known plaintext/ciphertext [Matsui, 1993]