CS326 – Systems Security

Lecture 2

Introduction – Simple Ciphers

Elias Athanasopoulos
eliasathan@cs.ucy.ac.cy
Sections of this lecture

• Introduction to Cryptography
• Simple Ciphers
INTRODUCTION TO CRYPTOGRAPHY
Cryptography Roadmap

Cryptology

Cryptography

Symmetric Ciphers

Asymmetric Ciphers

Utilities (Hash Functions, MACs, etc.)

Apps/Protocols (TLS, ToR, etc.)

Cryptanalysis (Attacks)
The Need for Cryptography

- People had always secrets
- Ordinary applications are based on secrecy
  - e.g., elections (or e-voting)
- Machines need to verify information
  - detect errors
- Unforgeable information
  - ordinary signatures vs digital signatures
- Many new applications
  - From car keys to smartcards, and cellphones
Basic Problem

Oscar can see the message (confidentiality)
Oscar can modify the message (integrity)
Cryptographic Approach

- **m**: plaintext
- **c**: ciphertext
- **k**: key

**Cryptosystem**
- Encryption algorithm
- Decryption algorithm
- Key(s) involvement

**Diagram**

Alice → Encrypt → Internet → Decrypt → Bob

- Secure Channel
  - Secure Channel
  - **K**
  - **K**
Kerchoff’s Principle

A cryptosystem should be secure even if everything about the system, except the key, is public knowledge
Security via Obscurity

• All crypto algorithms are assumed to be known

• Security is based on
  – Secrecy of the key
  – Hard to infer the plaintext via the ciphertext

• Cryptanalysis
  – Infer the plaintext from ciphertext without knowing the key
SIMPLE CIPHERS
Simple Substitution Cipher

• Assume an alphabet
  – abcdefghijklmnopqrstuvwxyz

• Index the letters
  – a is 1, b is 2, c is 3, ..., z is 26

• Select a key (secret), which shifts the order
  – Assuming the key is 3, then a is shifted three letters and becomes d, and z becomes c (wraps around the alphabet)
Caesar Cipher

Encryption
\[ C = P + K \mod 26 \]

Decryption
\[ P = C - K \mod 26 \]

assuming an alphabet of 26 letters

A simple message
\[ X \rightarrow X + \text{key} \]
(i.e., ‘a’ becomes ‘d’)

\[ \text{dcwlpsohcp} \]
\[ \text{hwwdjh} \]
Security Analysis

- Brute force attack
  - Key space is too small, 26 options
  - You need to just try 25 different keys
- All ciphers are vulnerable to brute force attack
  - If key space is too large, then attack is not practical
  - The cipher is then *Computationally Secure*
Multiple and Running Keys

• Vigenère Cipher

Key = r, u, n (three Caesar’s keys)

tobeornottobethatisthequestion
runrunrunrunrunrunrunrunrunrunrunrunrunrunrunrunrunrunrunrun
KIOVIEEIGKIOVNURNVJNUVKHVMGZIA
Secure Enough?

• Vigenère Cipher
  – **Problem 1**: repeated patterns in ciphertext

Key = r, u, n (three Caesar’s keys)

tobeornottobethatisthequestion
runrunrunrunrunrunrunrunrunrunrunrunrunrunrunrunrun

`KIOVIEEIGKIOV` NURNVJNUVKHVMGZIA
Ideal Substitution Cipher

Lookup Table

a becomes z
b becomes d
c becomes x
...
z becomes b

Key space: $26 \times 25 \times 24 \times \ldots \times 1 = 26! \approx 2^{88}$
Frequency Analysis

• Simple substitution leaves the statistics of the plain message in the ciphertext
• A message of a known origin (e.g., English text) has no uniform letter distribution
• Letters e, t, and a, are more popular than x, z, and v
Letter Distribution in English Text

<table>
<thead>
<tr>
<th>Letter</th>
<th>Frequency</th>
<th>Letter</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>8.167%</td>
<td>n</td>
<td>6.749%</td>
</tr>
<tr>
<td>b</td>
<td>1.492%</td>
<td>o</td>
<td>7.507%</td>
</tr>
<tr>
<td>c</td>
<td>2.782%</td>
<td>p</td>
<td>1.929%</td>
</tr>
<tr>
<td>d</td>
<td>4.253%</td>
<td>q</td>
<td>0.095%</td>
</tr>
<tr>
<td>e</td>
<td>12.702%</td>
<td>r</td>
<td>5.987%</td>
</tr>
<tr>
<td>f</td>
<td>2.228%</td>
<td>s</td>
<td>6.327%</td>
</tr>
<tr>
<td>g</td>
<td>2.015%</td>
<td>t</td>
<td>9.056%</td>
</tr>
<tr>
<td>h</td>
<td>6.094%</td>
<td>u</td>
<td>2.758%</td>
</tr>
<tr>
<td>i</td>
<td>6.966%</td>
<td>v</td>
<td>0.978%</td>
</tr>
<tr>
<td>j</td>
<td>0.153%</td>
<td>w</td>
<td>2.360%</td>
</tr>
<tr>
<td>k</td>
<td>0.772%</td>
<td>x</td>
<td>0.150%</td>
</tr>
<tr>
<td>l</td>
<td>4.025%</td>
<td>y</td>
<td>1.974%</td>
</tr>
<tr>
<td>m</td>
<td>2.406%</td>
<td>z</td>
<td>0.074%</td>
</tr>
</tbody>
</table>
Example

Cipher

iq ifcc vqqr fb rdq vfllcq na rdq cfjwhwz hr bnnb hcc hwwhbsqvqbre hwq vhlp

• q := E, h := A, r := T

iq ifcc vEEr fb TdE vfllcE na TdE cfjwAwz AT bnnb Acc AwwAbsEvEbTe AwE vAle

• After more iterations/trials

WE WILL MEET IN THE MIDDLE OF THE LIBRARY AT NOON ALL ARRANGEMENTS ARE MADE
Modular Arithmetic

• In cryptography we use integers to express messages and then perform actions on them
  – For instance, Caesar cipher shifts each letter in the alphabet

• Modern ciphers are more complicated but they also work on finite sets of integers
  – Example one byte can take an integer value between 0 and 255
Modular Arithmetic

• Map the product of any computation (addition, multiplication) to a bounded set of integers
  – The bound is defined by the **modulus** (or base)

• Consider the analog clock, you can add several hours to a particular time, but the result will be always below 12h
  \[4h + 10h = 2h\]
Modular Arithmetic

Let $a$, $r$, $m$ integers and $m > 0$, then

$$a \equiv r \mod m, \text{ if } m \text{ divides } a-r$$

• Examples
  
  $44 \equiv 2 \mod 7$
  
  $-9 \equiv 3 \mod 6$
  
  $11 \equiv 1 \mod 5$
  
  $18 \equiv 8 \mod 10$
Equivalent Numbers

• Consider the set of integers for modulus 7
  – \{0, 1, 2, 3, 4, 5, 6\}
• All integers with reminder 1 form one set
  – \{…, -13, -6, 1, 8, 15, 22, … \}
• All integers with reminder 2 form one set
  – \{…, -12, -5, 2, 9, 16, 23, … \}
• For each set, all numbers are equivalent in modulus 7 arithmetic
• **Example**: 529 mod 7?
  529 = 23*23 mod 7 = 2*2 mod 7 = 4 mod 7
Transposition Cipher

• Instead of substituting letters, re-arrange them

<table>
<thead>
<tr>
<th>t</th>
<th>h</th>
<th>i</th>
<th>s</th>
<th>i</th>
<th>s</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>f</td>
<td>u</td>
<td>n</td>
<td>k</td>
<td>y</td>
</tr>
<tr>
<td>m</td>
<td>e</td>
<td>s</td>
<td>s</td>
<td>a</td>
<td>g</td>
</tr>
<tr>
<td>p</td>
<td>e</td>
<td>r</td>
<td>m</td>
<td>u</td>
<td>t</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>t</th>
<th>a</th>
<th>m</th>
<th>p</th>
<th>h</th>
<th>e</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>f</td>
<td>s</td>
<td>r</td>
<td>s</td>
<td>u</td>
<td>s</td>
</tr>
<tr>
<td>n</td>
<td>a</td>
<td>u</td>
<td>i</td>
<td>k</td>
<td>g</td>
<td>t</td>
</tr>
<tr>
<td>s</td>
<td>y</td>
<td>e</td>
<td>e</td>
<td></td>
<td></td>
<td>d</td>
</tr>
</tbody>
</table>
Scytale
Resources

• This lecture was built using material that can be found at
  – Chapter 1, Understanding Cryptography, http://www.crypto-textbook.com