Encryption and Forensics/Data Hiding
Cryptography Background

See:
http://www.cacr.math.uwaterloo.ca/hac/
For more information
Security Objectives

Confidentiality (Secrecy):
Prevent/Detect/Deter improper disclosure of information

Integrity:
Prevent/Detect/Deter improper modification of information

Availability:
Prevent/Detect/Deter improper denial of access to services provided by the system
Security Services

- **Confidentiality**: protection of any information from being exposed to unintended entities.
  - Information content
  - Parties involved
  - Where they are, how they communicate, how often, etc.
- **Authentication**: assurance that an entity of concern or the origin of a communication is authentic - it’s what it claims to be or from
- **Integrity**: assurance that the information has not been tampered with
Encryption/Decryption

- Plaintext: a message in its original form
- Ciphertext: a message in the transformed, unrecognized form
- Encryption: the process for producing ciphertext from plaintext
- Decryption: the reverse of encryption
- Key: a secret value used to control encryption/decryption
Cryptanalysis: Break an Encryption Scheme

• Ciphertext only
  – Analyze only with the ciphertext
  – Exhaustive search until “recognizable plaintext”
  – Need enough ciphertext

• Known Plaintext
  – \langle plaintext, ciphertext \rangle is obtained
  – Great for monoalphabetic cipher

• Chosen Plaintext:
  – Choose plaintext, get the ciphertext
  – Useful if limited set of messages
Methods for Attacking Encrypted Text

- Table 4-1 of the textbook
- Cryptanalysis
  - Ciphertext only
    - Analyze only with the ciphertext
    - Exhaustive search until “recognizable plaintext”
    - Need enough ciphertext
  - Known Plaintext
    - \(<\text{plaintext}, \text{ciphertext}>\) is obtained
  - Chosen Plaintext:
    - Choose plaintext, get the ciphertext
    - Useful if limited set of messages
- Password Guess (Similar to known plaintext)
  - Dictionary
  - Educated Guess
  - Brute Force
Methods for Attacking Encrypted Text
– Con’t

• Scavenge Password
  – Physical Search
  – Logical Search
  – Network Sniff

• …
Computationally Difficult

- Cryptographic algorithms need to be reasonably efficient
- Cryptographic algorithms are not impossible to break with the key
  - e.g. try all the keys – brute-force cryptanalysis
  - Time can be saved by spending money on more computers.
- A scheme can be made more secure by making the key longer
  - Increase the length of the key by one bit
    - The good guy’s job just a little bit harder
    - The bad guy’s job up to twice as hard.
Types of Cryptographic functions

• Secret Key Cryptography
  – One key
• Public Key Cryptography
  – Two keys: public, private
• Hash function
  – No key
Secret Key Cryptography

plaintext $\xrightarrow{\text{encryption}}$ ciphertext $\xleftarrow{\text{decryption}}$ plaintext

key $\leftarrow$ same key $\rightarrow$ key

- Same key is used for both encryption and decryption
  - Symmetric cryptography
  - Conventional cryptography
- Ciphertext is about the same length as the plaintext
- Examples: DES, IDEA, AES…
Public Key Cryptography

- Invented/published in 1975
- Each individual has two keys:
  - Private key is kept secret
  - Public key is publicly known
- Much slower than secret key cryptography
- Also known as
  - Asymmetric cryptography
Public Key Cryptography cont’d

plaintext \xrightarrow{\text{signing}} \text{Signed message} \xrightarrow{\text{verification}} \text{plaintext}

private key \uparrow \quad \text{public key}

- **Digital Signature**
  - Only the party with the private key can generate a digital signature
  - Verification of the signature only requires the knowledge of the public key
  - The signer cannot deny he/she has done so.
  - Example illustrated in Fig. 4-4 and 4-5
Applications of Public Key Cryptography

• Security uses of public key cryptography
  – Known public key cryptography is orders of magnitude slower than the best known secret key cryptographic algo.

• Transmitting over an Insecure Channel
  Alice Encrypt \( m_A \) using \( e_B \) → Decrypt to \( m_A \) using \( d_B \)
  Bob

  Bob Encrypt \( m_B \) using \( e_A \) ← Decrypt to \( m_B \) using \( d_A \)

• \( e \): public key, \( d \): private key

• Secure Storage on Insecure Media
  – Because of performance issues, you can randomly generate a secret key, encrypt the data with that secret key, and encrypt the secret key with the public key
  – Using public key of a trusted person
Hash Algorithms

• Message digests, one-way transformations

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

• Easy to compute $h(m)$
• Given $h(m)$, no easy way to find $m$
• Computationally infeasible to find $m_1$ and $m_2$, so that $h(m_1) = h(m_2)$
Trusted Intermediaries

• Cannot do pair-wise authentication with secret key technology
  – Each computer needs to know n-1 keys
• Key Distribution Center (KDC)
• Certification Authorities (CAs)
• Certificate
Key Distribution Center

• Use a trusted node known as Key Distribution Center (KDC)
  – Secret key cryptography

• The KDC knows keys for all nodes
  – $\alpha$ asks KDC for secret (securely) to talk to $\beta$
  – KDC encrypts $R_{\alpha\beta}$ with the key shared between $\alpha$ and KDC, send to $\alpha$
  – KDC encrypts $R_{\alpha\beta}$ with the key shared between $\beta$ and KDC, send to $\beta$ : ticket
Certification Authorities (CAs)

• Public key cryptography
  – Problem: How can you be sure that the public keys are correct?

• CA: ensure validity of public keys

• Certificates
  – Signed messages specifying a name (Alice) and the corresponding public key
  – All nodes need to be preconfigured with the CA’s public key
Certificate Authorities Trusted by IE

Certification Practice Statement

• Certification Practice Statement (CPS)
  – How certificate authorities operate, maintain the security of their infrastructures.
  – Certificate Revocation List

• One example:
  – Verisign CPS
    • http://www.verisign.com/repository/CPS/
Codes and Compression

• **uuencode**
  – Uuencoding obscures binary data, but not ASCII text
  – Winzip can open and extract uuencoded files

• **Compression**
  – Recognizable patterns
  – Lossless data compression
    • Zip, gzip
    • GIF, TIFF..
  – Lossy data compression
    • JPEG, MPEG...

• **Data is often compressed before it is encrypted**
Challenges

• Any transformation performed on text data make it difficult or impossible to do a batch search for keywords!
• How to identify encrypted data
  – To see if it can be compressed
Password recovery tool for Windows

- Cain:
  - Uncovering cached password
  - Recovering password by sniffing the network
  - Cracking encrypted password using Dictionary
  - Brute-force and Cryptanalysis attacks
  - ...
Cain – uncover password from protected storage

<table>
<thead>
<tr>
<th>Resource</th>
<th>Username</th>
<th>Password</th>
<th>Type</th>
<th>Identity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ldap</td>
<td>ldapuser</td>
<td>ldappass</td>
<td>Outlook Express LDAP Account</td>
<td>Main Identity</td>
</tr>
<tr>
<td>pop.test.test</td>
<td>popuser</td>
<td>poppass</td>
<td>MS Outlook 2002 POP3 Account</td>
<td></td>
</tr>
<tr>
<td><a href="http://oe.msn.msnnmail.hotmail.com/cgi-bin/hmdata">http://oe.msn.msnnmail.hotmail.com/cgi-bin/hmdata</a></td>
<td>msnuser</td>
<td>msnpass</td>
<td>MS Outlook 2002 HTTP Account</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.test.test">http://www.test.test</a></td>
<td>webuser</td>
<td>webpass</td>
<td>MS Outlook 2002 HTTP Account</td>
<td></td>
</tr>
<tr>
<td>IdentitiesPass</td>
<td></td>
<td></td>
<td>pass1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Outlook Express Identity</td>
<td>Main Identity</td>
</tr>
</tbody>
</table>

Menu actions:
- Refresh
- Remove
- Delete
- Remove All
- Export
Cain – attack against encrypted password
Password Cracker

- www.lostpassword.com

- L0phCrack
- ZipPassword
Hiding and Finding Data

• Changing a file’s extension
  – Windows uses the filename extension to identify the data type of the file
  – *Quick View Plus*

• Check the file header
  – Contain a hexadecimal value that can be usually be correlated to file type

• File Format Information
Steganography

- Steganos: secret or hidden
- Graphy: drawing or writing
- http://www.stegoarchive.com/
## File Systems

<table>
<thead>
<tr>
<th>Developer</th>
<th>FAT12</th>
<th>FAT16</th>
<th>FAT32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Full Name</th>
<th>FAT12</th>
<th>FAT16</th>
<th>FAT32</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(12-bit version)</td>
<td>(16-bit version)</td>
<td>(32-bit version)</td>
</tr>
<tr>
<td>Introduced</td>
<td>1977 (Microsoft Disk BASIC)</td>
<td>July 1988 (MS-DOS 4.0)</td>
<td>August 1996 (Windows 95 OSR2)</td>
</tr>
<tr>
<td>Partition identifier</td>
<td>0x01 (MBR)</td>
<td>0x04, 0x06, 0x0E (MBR)</td>
<td>0x0B, 0x0C (MBR) EBD0A0A2-B9E5-4433-87C0-68B6B72699C7 (GPT)</td>
</tr>
</tbody>
</table>

- Windows NT and Windows XP support NTFS, FAT16, and FAT 32.
NTFS Alternate Data Streams (ADS)

- NTFS file systems supports multiple data streams
- Allow files to be associated with more than one data stream
- Method of hiding executables or proprietary content
- Uses NTFS file system multiple attributes
- Syntax – {file name}:{stream name}
- Create: type file > visible:hidden
- Reference:
  - http://www.windowsecurity.com/articles/Alternate_Data_S treams.html
ADS Example 1

- `start c:\temp\calc.exe:notepad.exe`
ADS Example 2
ADS Example 2 – Con’t
ADS Example 2– Con’t
LADS – List Alternate Data Streams

- http://www.heysoft.de/nt/ep-lads.htm