
Cryptography

- Cryptography is a collection of mathematical techniques for protecting information;

- Most important cryptographic technique is encryption/decryption

Cryptography for information protection

<table>
<thead>
<tr>
<th>Level</th>
<th>What to protect</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Existence of message</td>
<td>Steganography</td>
</tr>
<tr>
<td>2</td>
<td>Metadata of message</td>
<td>Privacy-enhancing technologies</td>
</tr>
<tr>
<td>1</td>
<td>Content of message</td>
<td>Encryption</td>
</tr>
<tr>
<td>0</td>
<td>Nothing</td>
<td>None</td>
</tr>
</tbody>
</table>

Table by I.A. Goldberg

Encryption is used
- Directly at the level 1
- As an important ingredient at the levels 2 and 3

Two categories of encryption algorithms

- Symmetric encryption (or symmetric key encryption):
  - to encrypt and decrypt a message the same key (a piece of information; sequence of bits) is used

- Asymmetric encryption (or asymmetric key encryption):
  - One key is used for encryption (usually publicly known, public key);
  - Another key is used for decryption (usually private, or secret key)
Symmetric (conventional) encryption

Components of Symmetric Encryption
- Plaintext
- Encryption algorithm
- Secret key
- Ciphertext (encrypted text)
- Decryption algorithm

Security of symmetric encryption
- **Important principle:** security of symmetric encryption depends on
  - the secrecy of the key,
  - Not the secrecy of the algorithm

Why?
- It is difficult to invent new algorithms and keep them in a secret;
- Producing keys is much easier;

Requirements for symmetric encryption
- **Strong** encryption algorithm:
  - The adversary (opponent) should be unable to decrypt encrypted text, even if he/she knows several pairs (plaintext, encrypted plaintext)
  - Sender and receiver must have obtained copies of the secret key in a secure way and **must keep the key secure**
Two more classifications of cryptosystems

- **Type of operations used**
  - Substitutions;
  - Transpositions;

- **The way in which plaintext is processed**
  - Block cipher: input block of elements (e.g., characters) is transformed to the output block at once;
  - Stream cipher: processes the input elements continuously, one element at a time.

### Classics: substitutions

- Each element of the plaintext (bit, letter, group of bits) is mapped to another element
- **Example:**
  - A -> B
  - B -> C
  - C -> D
  - ...
  - Z -> A

  Plaintext “Knowledge is power”
  
  Is transformed into
  
  “Lopxmfehf jt rpxfs”

### Classics: transposition

- Elements of the plaintext are re-arranged.
- Example: “Knowledge is power”

  Is transformed into
  
  “Keiwndseog weprl o”

### Two remarks

- Most modern algorithms include multiple stages of *interleaving* substitutions and transpositions;
- The encryption uses a *key* (unlike simple examples on the previous slides)
Cryptanalysis and computationally secure schemes

- **Cryptanalysis**: The process of attempting to discover the plaintext or key;
- Depends very much on the information available;
- An encryption scheme is **computationally secure** if
  - The cost of breaking the scheme exceeds the value of the encrypted information;
  - The time required to break the scheme is more than lifetime of the information;

---

Types of Attacks (Cryptanalysis)

<table>
<thead>
<tr>
<th>Type of attack</th>
<th>Known to Cryptanalyst</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selected info</td>
<td>Encryption algorithm</td>
</tr>
<tr>
<td>Known plaintext</td>
<td>Encryption algorithm</td>
</tr>
<tr>
<td>Cipher text</td>
<td>Encryption algorithm</td>
</tr>
<tr>
<td>Known plaintext</td>
<td>Encryption algorithm</td>
</tr>
<tr>
<td>Cipher text</td>
<td>Encryption algorithm</td>
</tr>
</tbody>
</table>

---

Brute-Force Approach in Cryptanalysis

- If nothing else helps and there is no weakness in the encryption algorithms, brute-force approach may be applied;
- Try every possible key until correct translation of the encrypted text into plaintext is obtained;
- **Possible issue**: how does cryptanalyst recognize correct plaintext? Imagine it has been compressed before encryption;
- **Main issue**: time !!!

---

Time required for brute-force search

<table>
<thead>
<tr>
<th>Key Size (bits)</th>
<th>Number of Alternatives</th>
<th>Time required at 1 encryption/second</th>
<th>Time required at 10^9 encryptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>2^50 - 1.2 x 10^15</td>
<td>2^49 ps = 6.04 years</td>
<td>2.5 x 10^15 years</td>
</tr>
<tr>
<td>90</td>
<td>2^90 - 2.2 x 10^27</td>
<td>2^89 ps + 1114 years</td>
<td>1.6 x 10^19 years</td>
</tr>
<tr>
<td>128</td>
<td>2^128 - 4.4 x 10^38</td>
<td>2^127 ps + 5.3 x 10^9 years</td>
<td>3.4 x 10^20 years</td>
</tr>
<tr>
<td>192</td>
<td>2^192 - 6.6 x 10^46</td>
<td>2^191 ps + 2.0 x 10^16 years</td>
<td>3.3 x 10^26 years</td>
</tr>
<tr>
<td>256</td>
<td>2^256 - 1.1 x 10^54</td>
<td>2 x 10^25 ps + 6.6 x 10^17 years</td>
<td>6.6 x 10^27 years</td>
</tr>
</tbody>
</table>