Lesson 6

Classical Cryptosystems:

Knowledge is the basis of action

Wholeness Statement

In a classical cryptosystem, the sender and receiver both use the same key. Therefore key management is very important for secure communication. One of the most important steps in the management of the mantra is keeping it private.

Overview

1. `WKLVOHFWXUHLVDQLQWURGXFWLRQWRFUBSWRJUDSKB` (ciphertext)
2. `THISLECTUREISANINTRODUCTIONTOCRYPTOGRAPHY` (plaintext)
3. A key was used with an enciphering function to encipher the plaintext.
4. The same key was used with a deciphering function to decipher the ciphertext.
5. Cryptanalysis
   a. ciphertext only
   b. known plaintext
   c. chosen plaintext

Overview

6. Two fundamental types of ciphers
   a. Transposition
   b. Substitution
7. Ciphers
   a. Rail fence
   b. Caesar
   c. Vigenere
   d. One-time pad
   e. DES

Classical Cryptography

Chapter 8.1 and 8.2

Cryptosystem

Five components (M, C, K, E, D)

1. (M) set of plaintexts
   (messages in plain English)
2. (C) set of ciphertexts
   (plaintexts that have been enciphered)
3. (K) set of keys
   (used to encipher and decipher messages)
4. (E) set of enciphering functions
   (use keys to encipher plaintext)
5. (D) set of deciphering functions
   (use keys to decipher ciphertext)

Attacks

- The subject attempting to break the cryptosystem is the *adversary*
  - Always assume the adversary knows the encryption and decryption algorithm but not the key
  - What is the security principle?

Three Types of Attacks

1. Ciphertext only

   (adversary only has ciphertext, wants to find plaintext or key)
2. Known plaintext

   (adversary has the ciphertext for a known plaintext, wants to find the key)
3. Chosen plaintext

   (adversary can generate ciphertext for any plaintext, wants to find the key)

Attacks are usually specific to the plaintext language

- Attacks are usually based on mathematics and statistical analysis
- Statistical attacks
  - Make assumptions about the plaintext language
    - E.g., distribution of letters, pairs of letters (such as ‘q’ is always followed by ‘u’), triplets of letters, *etc.*
    - Called *model of the language*
  - Examine ciphertext, correlate properties with the assumptions

Classical Cryptography

- Single-key (or symmetric) cryptosystem
  - Same key for both sender (encrypting the message) and the receiver (decrypting)
- Three types of cipher
  - Transposition
  - Substitution
  - Product (combinations of the above)

Main Point

1. A cryptosystem has an encipher function, a decipher function, and a key used to convert plaintext into ciphertext. One could view the outward stroke of meditation as an encipher function and the inward stroke as a decipher function. The key is the mantra, the field of relative existance is ciphertext and the field of pure consciousness the plaintext.

Transposition Ciphers

Chapter 8.2.1

Transposition Cipher

- A transposition cipher rearranges the characters in the plaintext to form the cipher text.
  - The letters are not changed.
- Can be cracked by using a table of n-gram frequencies to identify common n-grams.

Rail Fence Cipher

- A *rail fence cipher* is composed by writing the plaintext in two rows, proceeding down, then across, and reading the ciphertext across, then down.

Example:
Encrypt `FAMILY`

```
FML
AIY
```

to get `FMLAIY`

Substitution Ciphers

Chapter 8.2.2

Substitution Cipher

- A substitution cipher changes characters in plaintext to produce the ciphertext.
- Can be cracked
  - By comparing character frequencies with a model of the language
    - statistical frequency analysis
    - cipher text statistically looks too much like plaintext
  - Through exhaustive search (if key is too short)

Caesar Cipher

A *Caesar Cipher* shifts the letters of the plaintext right k characters with wraparound

For example, if k = 4

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
 B             O        X    (plaintext)
 B   F             S         (ciphertext)
```

Substitution Cipher Java

```
static public void caesar()
 {
        // m is a member of M
      String m = "abcdefghijklmnopqrstuvwxyz“;
        // k is a member of K
      int k = 4;
        // e is a member of E, c is a member of C
      String c = e(m, k);
       // d is a member of D
      String r = d(c, k);
      System.out.println(c);
      System.out.println(r);
}
```

```
// e is an encryption function
static String e(String m, int k)
{
    StringBuilder sb
        = new StringBuilder();
    m = m.toUpperCase();
    for (int i=0; i<m.length(); i++)
    {
      // convert to 0..26
      int n = m.charAt(i) - 'A';
    // shift right with wraparound
      n = (n + k) % 26;
      // convert back to ASCII
      sb.append((char)('A' + n));
    }
    return sb.toString();
}
```

```
// d is a decryption function
static String d(String c, int k)
{
    StringBuilder sb
        = new StringBuilder();
    for (int i=0; i<c.length(); i++)
    {
      // convert to 0..26
      int n = c.charAt(i) - 'A';
      // shift left with wraparound
      // (add 26 to prevent neg #)
      n = (26 + n - k) % 26;
      // convert back to ASCII
      sb.append((char)('A' + n));
    }
    return sb.toString();
}
```

Substitution Cipher C#

```
static public void Caesar()
{
     // m is a member of M
    string m = "abcdefghijklmnopqrstuvwxyz“;
     // k is a member of K
    int k = 4;
     // e is a member of E, c is a member of C
    string c = E(m, k);
    // d is a member of D
    string r = D(c, k);
    Console.WriteLine(c);
    Console.WriteLine(r);
}
```

```
// e is an encryption function
static string E(string m, int k)
{
    StringBuilder sb
        = new StringBuilder();
    m = m.ToUpperCase();
    foreach (char ch in m)
    {
      // convert to 0..26
      int n = ch - 'A';
    // shift right with wraparound
      n = (n + k) % 26;
      // convert back to ASCII
      sb.Append((char)('A' + n));
    }
    return sb.ToString();
}
```

```
// d is a decryption function
static string D(string c, int k)
{
    StringBuilder sb
      = new StringBuilder();
    foreach (char ch in c)
    {
      // convert to 0..26
      int n = ch - 'A';
      // shift left with wraparound
      // (add 26 to prevent neg #)
      n = (26 + n - k) % 26;
      // convert back to ASCII
      sb.Append((char)('A' + n));
    }
    return sb.ToString();
}
```

Vigenere Cipher

- Uses a key to map characters of plaintext to cipher text.
- Terms
  - *Period*: length of the key
  - *Tableaux*: table used to encipher and decipher messages
- The key mapping is represented as a tableaux (see below).
- The key repeats as necessary

Vigenere Tableaux

```
  ABCDEFGHIJKLMNOPQRSTUVWXYZ      ABCDEFGHIJKLMNOPQRSTUVWXYZ
  --------------------------      --------------------------
A ABCDEFGHIJKLMNOPQRSTUVWXYZ    N NOPQRSTUVWXYZABCDEFGHIJKLM
B BCDEFGHIJKLMNOPQRSTUVWXYZA    O OPQRSTUVWXYZABCDEFGHIJKLMN
C CDEFGHIJKLMNOPQRSTUVWXYZAB    P PQRSTUVWXYZABCDEFGHIJKLMNO
D DEFGHIJKLMNOPQRSTUVWXYZABC    Q QRSTUVWXYZABCDEFGHIJKLMNOP
E EFGHIJKLMNOPQRSTUVWXYZABCD    R RSTUVWXYZABCDEFGHIJKLMNOPQ
F FGHIJKLMNOPQRSTUVWXYZABCDE    S STUVWXYZABCDEFGHIJKLMNOPQR
G GHIJKLMNOPQRSTUVWXYZABCDEF    T TUVWXYZABCDEFGHIJKLMNOPQRS
H HIJKLMNOPQRSTUVWXYZABCDEFG    U UVWXYZABCDEFGHIJKLMNOPQRST
I IJKLMNOPQRSTUVWXYZABCDEFGH    V VWXYZABCDEFGHIJKLMNOPQRSTU
J JKLMNOPQRSTUVWXYZABCDEFGHI    W WXYZABCDEFGHIJKLMNOPQRSTUV
K KLMNOPQRSTUVWXYZABCDEFGHIJ    X XYZABCDEFGHIJKLMNOPQRSTUVW
L LMNOPQRSTUVWXYZABCDEFGHIJK    Y YZABCDEFGHIJKLMNOPQRSTUVWX
M MNOPQRSTUVWXYZABCDEFGHIJKL    Z ZABCDEFGHIJKLMNOPQRSTUVWXY
```

Enciphering Example

Encrypt ABACUS using the key CPU

Key:	`CPUCPU`
Plaintext:	`ABACUS`
Ciphertext:	`CQUEJM`

This is because
  column C, row A is C
  column P, row B is Q
  column U, row A is U
  column C, row C is E
  column P, row U is J
  column U, row S is M

Deciphering Example

- To decrypt CQUEJM we can either use the above tableau and search down each column for the ciphertext character and then look to the far left to find the plaintext character, or we could create an inverse tableaux.

Key:	`CPUCPU`
Ciphertext:	`CQUEJM`
Plaintext:	`ABACUS`

Cracking the Vigenere Cipher

- For many years the Vigenere Cipher was considered unbreakable.
  - Until it was noticed that repetitions occur when characters of a key appear over the same characters in the ciphertext
  - It was noted that the number of characters between the repetitions is a multiple of the period

Plan of Attack

- Determine the period p
- Divide the message into segments of length p
  - Each character would be enciphered by the same character
- Then do a statistical or mathematical analysis of each segment
- These principles of attack are useful for attacking more complex ciphers
  - Patterns are the enemy of cryptosystems

Attacking the Vigenere Cipher

Encrypt THE BOY HAS THE BAG using the key VIG

Key:	`VIGVIGVIGVIGVIG`
Plaintext:	`THEBOYHASTHEBAG`
Ciphertext:	`OPKWWECIYOPKWIM`

- Note that OPK appears twice in the ciphertext
- Both are caused by VIG enciphering the same plaintext, THE
- The ciphertext repetitions are 9 characters apart, so 9 is a multiple of the period

Cracking the Vigenere Cipher

- Examine the ciphertext for multiple repetitions and tabulate their length and the number of characters between successive repetitions
- The period is likely to be a factor of the number of characters between repetitions
- From this we establish the probable period
- Then tabulate the characters for each key letter separately and solve each as a Caesar cipher
- The book gives an example showing how the key can be determined (4 pages)

One-Time Pad

- A variant of the Vigenere cipher
- Provably unbreakable
- The key must be at least as long as the message
  - So it does not repeat
- The key is chosen at random
  - If not random, one can attack by trying to regenerate the key
  - Pseudorandom number generators do NOT generate random numbers!

Binary Vigenere Cipher

- Consider the following tableaux which is binary rather than ASCII

```
     0  1
0    0  1
1    1  0
```

- Note that this is the truth table for the binary exclusive-or operator
- Thus whenever a cipher applies the xor operation on plaintext and a key, the result is, in effect, ciphertext from a binary Vigenere cipher
- DES, which is the next topic, uses this enciphering technique in several places

Example One-Time Pad

- Let's say that someone wants to send an AYE (yes) or a NAY(no) indicating whether to buy some stock.
- It is first necessary to get the binary of the plaintext:

```
AYE 41 59 45   01000001 01011001 01000101
NAY 4E 41 59   01001101 01000001 01011001
```

Do the following to send an "AYE"

```
Key:        00110011 10010000 01100111 (Key 1)
Plaintext:  01000001 01011001 01000101 (AYE)
Ciphertext: 01110010 11001001 00100010 (Not ASCII)
```

- Note that the cipher is just applying an exclusive-or to obtain the ciphertext

Example (cont.)

- Let's say that an attacker gets the above ciphertext and randomly generates keys to decipher it.
- A key exists that will decipher the ciphertext to "NAY" and the attacker has now no way of knowing that this is wrong since it is one of the expected messages.

```
Key:        00111111 10001000 01111011 (Key 2)
Ciphertext: 01110010 11001001 00100010 (AYE)
Plaintext:  01001101 01000001 01011001 (NAY)
```

- In fact, the ciphertext is equally likely to belong to any three characters (assuming the key is random)
  - i.e, there are as many keys that produce the actual plaintext as there are for any other three letter sequence

One-Time Pad is Unbreakable

- Unbreakable since it is possible to generate a key that will convert a ciphertext to any given plaintext
- Since the key is only used once, it is not possible to try a key on more than one ciphertext
- Since the key is random and the same length as the message, there will be no detectable patterns in the ciphertext

DES

Chapter 8.2.3

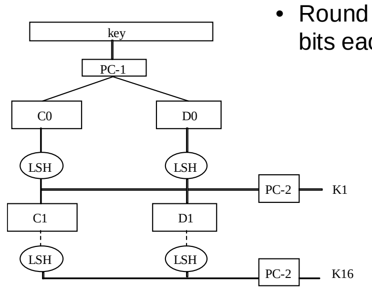
Data Encryption Standard

DES was designed to encipher sensitive but unclassified data

Overview

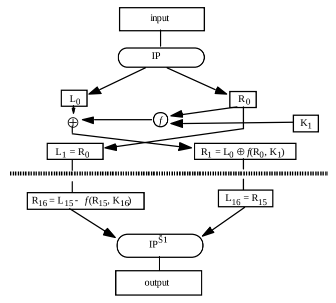
1. It is a product cipher because it uses both transposition and substitution
2. It is binary (like the one-time pad)
3. It breaks the plaintext into 64 bit blocks and enciphers each block independently of the others.
4. The encryption of a 64 bit input block produces a 64 bit output block
5. The output blocks are concatenated together to produce the ciphertext.
6. The key block used to encrypt a block is 64 bits long.
7. The cipher consists of 16 rounds (iterations). The output of one round is input to the next round.
8. The key used by a round is generated from the key block.
9. In a round, each 64 bit block is processed as follows:

Generation of Round Keys

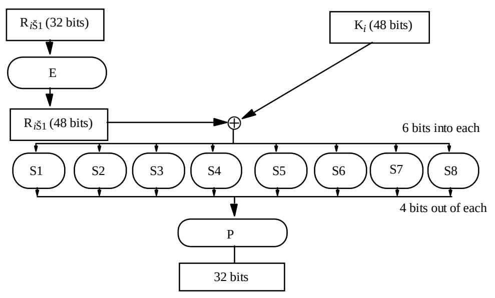


- Round keys are 48 bits each

Encipherment



The *f* Function



The *f* Function

- The f function provides the strength of the DES
- The right half of the input is expanded to 48 bits
- This is xor'ed with the round key
- The resulting 48 bits are split into eight sets of six bits each
- Each six bit set is put through a substitution table called the S-box
- Each S-box produces four bits of output
- These four bit outputs are concatenated into a 32-bit output which is permuted
- This permuted result is the output of the f function

Main Point

2. S- and P-boxes are used by DES to encipher a block during a round. The extreme simplicity of these two entities when combined intelligently can give rise to a seemingly random (complex) result. A basic tenet of SCI is that diversity arises from a singularity, the field of pure consciosness, the simplest form of awareness.

DES Modes

- Electronic Code Book Mode (ECB)
  - Encipher each block independently
  - Rarely used
- Cipher Block Chaining Mode (CBC)
  - Xor each block with previous ciphertext block
  - Requires an initialization vector for the first one
  - Most commonly used mode
  - Each block depends on the input and the previous block
  - *Self-healing*: errors (corrupted blocks) are propagated for at most two blocks

Conclusion and the Future of Cryptosystems

- ECB: Since each 64 bit block is enciphered independently, two identical 8 byte plaintext blocks will be enciphered to the same ciphertext.
- This is essentially a glorified substitution cipher.
- An improvement on ECB is Cipher Block Chaining (CBC) mode in which the ciphertext of the previous block is used in the encipherment of the next block.
- However, DES can be cracked with a distributed attack and faster hardware.
- The successor to DES is the Advanced Encryption Standard (AES).
- AES can use keys of 128, 192, or 256 bits and operate on blocks of 128 bits.

Main Point

3. DES had a relatively short heyday. New cryptanalytic techniques were developed trying to crack it. Also faster and faster machines were developed. The nature of life is progressive.

CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

1. Julius Caesar substituted each letter of a message with another letter that was a fixed number of places away from it in the alphabet to create the encoded message.

2. The Caesar cipher is a symmetric substitution cipher.

3. <u>Transcendental Consciousness</u> is the simplest form of awareness.

4. <u>Wholeness moving within itself</u>: in Unity Consciousness one has "deciphered" the relative field of existence and found it to be coextensive with the absolute unchanging field of pure consciousness.