MACQUARIE UNIVERSITY

Department of Computing

COMP343 CRYPTOGRAPHY AND INFORMATION SECURITY

ASSIGNMENT 1 - Implementation of Block Cipher

Worth: 150 marks scaled to 15%

Includes:

- program for encryption and decryption
- program that implements the birthday attack on a compression function built from the cipher
- program that implements an attack on double cipher (bonus)

Learning outcomes assessed: are as follows

- Understanding of basic principles of modern cryptography (100 marks, i.e. 10% overall);
- Implementation skills (50 marks, i.e. 5% overall).

Submission:

- Final Submission Deadline – 10pm Sunday, April 28, 2013

1 Overview

1.1 Implementation of Block Cipher

Block ciphers are basic cryptographic algorithms that provide confidentiality for messages transmitted over insecure channels or stored in insecure databases. In this assignment, you need to implement a simple cryptographic system that encrypts and decrypts files using the key provided. Use C++ as your programming language. Call your program

cryptalg.cc

The program must be designed in such a way that it can be used for both: encryption and decryption. The program should read the file name to be encrypted/decrypted from the first command line parameter and write the ciphertext/plaintext to the file specified as the second command line parameter. The cryptographic key is provided as a 16-bit hexadecimal number in the third command line parameter. The fourth parameter specifies whether to encrypt (E or e) or to decrypt (D or d). In other words, your program will be run as

$*cryptalg plain.txt cipher.txt 0xfc39 E*

for encryption and

$*cryptalg cipher.txt plain.txt 0xfc39 D*

for decryption, where *plain.txt* and *cipher.txt* are example file names (for plain and encrypted content, respectively) and *0xfc39* is an example key[1].

[1]Note that *0xfc* is high-order and *0x39* is low-order byte of that key.

1.2 Compression Hash Function and Birthday Attack

In this part we consider the cipher as a compression function that accepts a 16-bit message block m and 16-bit chaining variable h and produces 16-bit digest H or

$$E_h(m) = H,$$

where $E_k(\cdot)$ is the encryption function of the cipher with the key k. As the compression function takes 32-bit input (m, h) and translates it to a shorter 16-bit output, there are must be colliding inputs or two inputs (m_1, h_1) and (m_2, h_2) such that

$$E_{h_1}(m_1) = E_{h_2}(m_2)$$

Implement the birthday paradox attack so your program `birthday_attack.cc` finds collisions.

1.3 Cryptanalysis of Double Cipher

The security level of the cipher is normally equivalent to the length of the cryptographic key. For our cipher you would expect that the cipher provides 16-bit security. At the lectures we discussed that you could increase the security level by concatenation of encryptions. For instance the security level of DES can be improved by using the triple DES. At the lectures, it was shown that double DES fails to provide an expected level of security of 112 bits.

Implement the attack on double cipher in which

$$c = E_{k_2}(E_{k_1}(m))$$

where k_1, k_2 are two 16-bit keys.

2 Assignment Tasks

Consider the following Feistel-type cipher that takes a plaintext $m \in \{0,1\}^{16}$ and a primary cryptographic key $k \in \{0,1\}^{16}$ and generates a ciphertext $c \in \{0,1\}^{16}$. The cipher consists of 8 identical rounds. In each round, we apply the Feistel permutation that takes the input

$$(L_i, R_i)$$

and converts it into

$$L_{i+1} = R_i \text{ and } R_{i+1} = L_i \oplus P(z)$$

where (L_{i+1}, R_{i+1}) is output of the ith round. P is the permutation defined below and the value of byte z is computed in the following way:

$$(z_7, z_6, z_5, z_4) = S(y_7, y_6, y_5, y_4) \quad \text{and} \quad (z_3, z_3, z_1, z_0) = S(y_3, y_2, y_1, y_0),$$

where $y = R_i \oplus K_i$ and K_i is the ith round key (see the key scheduling algorithm). The function $S(\cdot)$ is the S-box function defined as shown in the table below

Input	Output
0000	0000
0001	0001
0010	1011
0011	1101
0100	1001
0101	1110
0110	0110
0111	0111
1000	1100
1001	0101
1010	1000
1011	0011
1100	1111
1101	0010
1110	0100
1111	1010

The permutation P is a rotation of bits of the argument by 2 positions left, i.e.

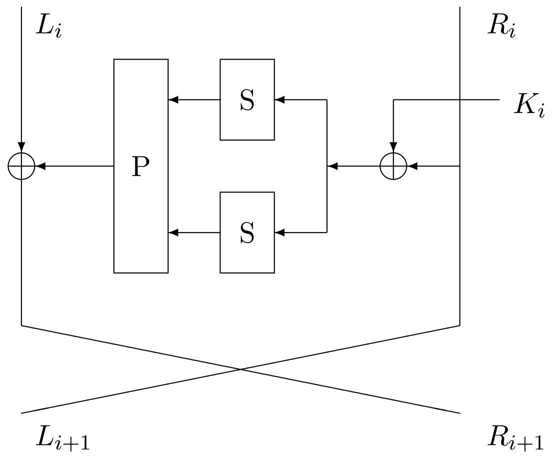
$$P((z_7, z_6, z_5, \ldots, z_2, z_1, z_0)) = z \lll 2 = (z_5, z_4, \ldots, z_0, z_7, z_6).$$

Key scheduling algorithm produces 8 round keys $K_0, \ldots, K_7$ from 16-bit primary key k in the following way. $K_0 = (k_7, k_6, \ldots, k_1, k_0)$ is the low-order byte of k, $K_1 = (k_{15}, k_{14}, \ldots, k_9, k_8)$ is the high-order byte of k and the rest of working keys is generated according to the following recurrence relation

$$K_i = ROTL^3(K_{i-1}) \oplus ROTL^5(K_{i-2}), \quad i = 2, \ldots, 7,$$

where $ROTL^i(a)$ means rotation of bits of byte a by i positions left.

Figure 1: One round of the cipher



You are expected to complete the following tasks.

1. Implement the encryption and decryption transformations by writing a single program called *cryptalg.cc* that can be used to encrypt and decrypt input files in the ECB mode. If the input plaintext file has an odd number of bytes, append one byte of value 0 at the end of the file. You can assume that ciphertext input files are always correct, i.e. they always

have even number of bytes. Your implementation should be efficient and in particular, you should NOT declare single boolean variables as arrays or strings (this is simply waste of memory).
[100 marks]

2. Write a program `birthday_attack.cc`, which prints on the standard output the pair (m_i, h_i) of colliding inputs in hexadecimal as shown below

   ```
   message1   chaining_value1
   message2   chaining_value2
   collision
   ```

   [50marks]

3. **This is a bonus task ! - you can get extra marks**. Implement an attack on the double cipher that is able to extract the 32-bit secret key assuming that you are able to observe the inputs and outputs of your cipher. If the cipher encrypts a 16-bit message m with a 32-bit key (k_1, k_2), then the cryptogram is

   $$c = E_{k_2}(E_{k_1}(m))$$

   Your program called `double_cipher_attack.cc` should

   - first implement the double cipher - you should call it `double_cryptalg.cc` program (the program accepts a 16-bit message and 32-bit key and outputs 16-bit cryptogram,
   - generate two 16-bit keys (you can use a pseudorandom generator available),
   - use the 32-bit key to obtain two observations of pairs (message cryptogram) for your double cipher. Let the pairs be (m_1, c_1) and (m_2, c_2),
   - take the first observation (m_1, c_1) and create two tables T_m and T_c. First one T_m stores all (single) encryptions of m_1 and m_2 for all possible keys k_1. The second T_c stores all (single) decryptions of c_1 and c_2 for all possible keys k_2,
   - now you would like to identify entries (values of the keys $k_1 = i$ and $k_2 = j$) for which

     $$\begin{aligned} E_i(m_1) &= D_j(c_1) \\ E_i(m_2) &= D_j(c_2) \end{aligned}$$

     You may sort (using some off-shelf algorithm) both tables and identify entries that have identical entries . Make sure that your algorithm is as efficient as possible.
   - print all entries that have identical entries in the both tables and values for the keys.

   [50marks]

3 Submission of Assignment

Your submission must must be done via `iLearn` and must include

- a copy of your program `cryptalg.cc`,
- a copy of your program `birthday_attack.cc` that accepts no arguments but produces two colliding inputs on the standard output,

Overall you must submit 2 files

- **electronically** via your `iLearn` account

You MAY also submit solution to the bonus task

- `double_cipher_attack.cc`

in this case you need to submit three files.

4 Plagiarism

Your work must be your own. Please take a look at the Academic Honesty Policy at

http://www.mq.edu.au/policy/docs/academic_honesty/policy.html