PRACTICE 2: MODES OF OPERATION

INTRODUCTION

Symmetric key encryption schemes based on block ciphers take a certain number of bits as input (block size) and generate an output with the same number of bits. In the well-known algorithm DES (*Data Encryption Standard*, FIPS PUB 46), 64 bits are taken as input to the encryption algorithm together with 56 bits of the DES key resulting in a ciphered output of 64 bits. This encryption algorithm makes use of a series of iterations.

If a message shall be encrypted, which is longer than the block size, the usage of a certain **mode of operation** is required: **ECB** (*Electronic Codebook Mode)*, **CBC** (*Cipher Block Chaining Mode*), **CFB** (*Cipher Feedback Mode*), **OFB** (*Output Feedback Mode*), and **PCBC** (*Propagating Cipher Block Chaining*, see the final notes on the last page). These modes allow encrypting messages, independently from the message size. Studying the properties of these modes of operations constitutes the objective of this practice.

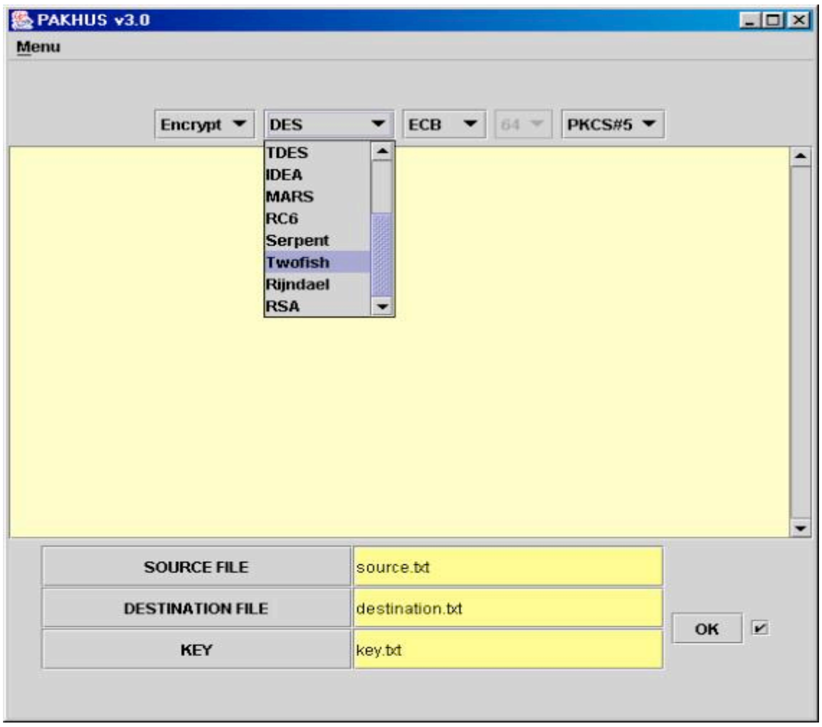
PRINCIPLES

For the preparation of this practice, you will use a Java application which has been especially programmed for this practice (*Pakhus v3.0*). and which has a graphical interface. It uses the *IAIK-JCE* libraries of the IAIK (*Institute for Applied Information Processing and Communications*, *http://jcewww.iaik.tu-graz.ac.at/*), which implement different encryption functions. Version 3.0 of *Pakhus* provides graphical support for the hash functions MD2, MD5, SHA, and RIPEMD-160, for the symmetric encryption schemes IDEA, DES, Triple-DES, MARS, RC6, Serpent, Twofish, and Rijndael, and for the RSA public key scheme. For the symmetric encryption schemes, the different modes of operation ECB, CBC, CFB, OFB, and PCBC are supported. Moreover, in case of DES it is possible to select the block length. Furthermore, it is allowed to select the padding mechanism to use (*PKCS#5*, '1', or no padding).

The handling of the application is quite intuitive, for example, with the possibility of selecting the algorithm to use (and if necessary, the mode of operation, block length, and the padding mechanism). Furthermore, there is a configuration option for using on algorithm for encryption or decryption, for the input file, the output file, the key, and a text console, where the results can be viewed optionally (cf. the figure below). In any case, you can consult the manual, which is provided together with the application, to solve questions about using the application *Pakhus*.

In this practice, a modified version of *Pakhus* is used, where only nine options are available (‘*Algorithm 0*’ to ‘*Algorithm 8*’), together with the possibility of using them to cipher or to decipher. Each algorithm can also be associated with a certain padding mechanism. The objective of the practice is to recognize each algorithm and which block modes and block lengths correspond to one of the available options.

As editing tool you can use the program Hedit, which allows you to edit a file in hexadecimal format.



SOURCE FILE: source.txt
DESTINATION FILE: destination.txt
KEY: key.txt

PRACTICE AND QUESTIONS

- Create a text file that contains the name, the last name(s), and the DNI of each member of your practice group, together with the date and hour in which the practice has been prepared. Encrypt this file with each of the five first available algorithms (these algorithms are all DES algorithms working in some mode of operation and with some block length).

1. **Based exclusively on the results obtained from the first encrypted texts and on the theoretical knowledge on modes of operation (and the data on PCBC in the notes at the end of this document), indicate well-founded if it is possible to know which mode of operation is in use for each of the first five algorithms presented by the application. If this can not be done, indicate what can at least be deduced from the obtained results.**

- In case that the previous test is not enough, make whichever tests that seem necessary to you (on the appropriate text files), until you can determine the mode of operation and block length for each of the five first algorithms labelled (‘Algorithm 0’ to ‘Algorithm 4’).

2. **Describe in detail the tests that you have made and the reason why you have made them. Describe also the obtained conclusions, the results that have led to these conclusions and explain why these results have led to the conclusions.**

3. **Indicate well-founded for each of the previous algorithms and modes of operation if it is possible to determine the corresponding initialization vector (IV). Obtain the initialization vectors for those cases, where it is possible. Show in detail for each mode of operations, the set of tests you have made. Furthermore, give reasons why you have made these tests.**

4. **DES algorithm has some keys that are denominated "weak" and other ones that are denominated "semi-weak". Locate these passwords in the recommended bibliography and try them in the Pakhus application.**

5. **Suppose that a user A wishes to transmit to another user B the following hexadecimal sequence:**

   4c 61 24 67 26 35 2b 64 48 79 3d 6a 54 6b (14 bytes)

   Therefore, A encrypts the sequence using the scheme labelled as “Algorithm 5” and transmits the result towards B by a communication network. B receives the ciphertext and decrypts it using the same scheme. Suppose that during the process of transmitting the value of the bits 104, 107, and 109 are flipped. Knowing that the used scheme corresponds to the coding scheme IDEA, using the OFB mode of operation and a block length of 64 bits, answer to the following questions:

   a. **What is the hexadecimal sequence, B can gain as output from the decryption?**
   b. **What can be observed comparing this sequence to the original one?**
   c. **What is the reason for this?**

   NOTES:
   - The initial hexadecimal sequence can be obtained by creating a text file containing the following sequence of characters:

     La$g&5+dHy=jTk

   - Suppose that the enumeration of bits begins with 0 and is increasing from left to right.

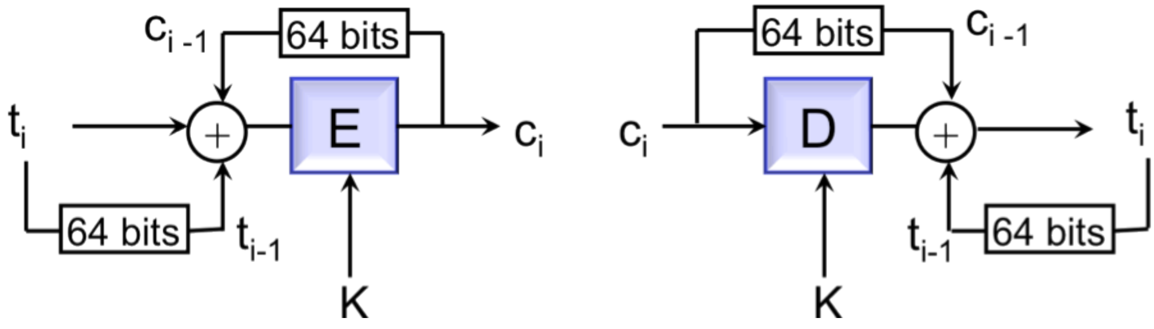
6. **Finally identify the remaining algorithms offered by the application and indicate how you have identified them.**

- The application shows in some cases an error message. This can occur if a file is encrypted, then some of the bits of the ciphertext are modified, and this file it finally decrypted again. However, this error message does not occur always. Apparently, it depends on the bit which has been modified in the ciphertext, whether an error occurs after the decryption or not.

7. **Indicate with reasons, if it is possible to calculate approximately the probability that an error occurs when decrypting a specific file for which a bit has been inverted after the encryption.**

NOTES:

- ✓ A Zip file is provided with this practice (2P_MO_pack.zip), containing all the files and programs that are necessary to do the practice.
- ✓ The application has been programmed using JDK 1.4.0_02. You may use a later version if you wish.
- ✓ In spite of being written in Java, the application is not portable to an operating system different from Windows, because the IAIK classes use native calls which prevent an execution on a different operating system.
- ✓ Like all applications written in Java, it is possible to obtain the source code by decompiling the corresponding classes. Although countermeasures have been taken to make the analysis of the source code difficult, everybody who wishes this is invited to respond to the questions of the practice using reverse engineering of the compiled code (explaining in detail the steps to deduce the answers).
- ✓ You can also use the application HEdit32 that is provided with Pakhus to edit and modify files.
- ✓ The PCBC mode of operation has the following structure:


c_{i-1}
64 bits
t_i
+
E
c_i
64 bits
t_{i-1}
K
64 bits
c_{i-1}
c_i
D
+
t_i
t_{i-1}
64 bits
K