C Y B E R S E C U R I T Y

LAB 3 /\ Solutions

1. Introduction.

The laboratory covers topics related to symmetric encryption - streaming and block ciphers. Furthermore, the algorithms have been analyzed in terms of how the data is encrypted, both block and stream ciphers are investigated.

Symmetric Key Cryptography is known under various terms, such as Secret Key Cryptography or Private Key Cryptography. As already mentioned, symmetric key algorithms have only one key, which is used for both encryption and decryption of messages. Therefore, when receiving a message, it is not possible to decrypt it unless the key is known to the recipient.

However, there are several problems with symmetric key algorithms. The first one is the key distribution problem. It is about how to get the key from the sender. Either the recipient has to meet the sender personally to receive the key or the key has to be sent to the recipient and thus can be accessed by unauthorized persons. This is the main problem of symmetric key cryptography and is called *the problem of distribution* or *exchange of keys*, and is inherent in symmetric key cryptography.

The second problem is also very serious. Let's assume that Alice wants to communicate with Bob as well as with John. There should be one key for all communication between Alice and Bob, and another key for all communication between Alice and John. The same key that is used between Alice and Bob cannot be used for communication be
between Alice and John, because in such a situation there is a chance that John can interpret messages passing between Alice and Bob. In a large amount of data exchange, this becomes impractical as each pair of senders and recipients would require a separate key.

In symmetric-key cryptography, the same algorithm is used by the sender and recipient. However, the key is changed from time to time. The same plain text encrypted with different keys leads to different ciphertext. Since the encryption algorithm is publicly available, it should be strong enough to prevent an attacker from deciphering the encrypted text.

Stream Cipher—A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. The decryption also happens one bit at a time. An example of a stream cipher is the RC4 cipher.

Block Cipher—In a block cipher, a block of plain text is treated as a whole and used to produce a ciphertext block of equal length. Typically, a block size of 64, 128, or 256 bits is used. Decryption also takes one block of encrypted data at a time. A problem occurs with block ciphers when there are repeating text patterns. Since the same cipher is generated for the same plain texts, it gives a clue to the cryptanalyst about the original plain text. The cryptanalyst can look for repeating strings and try to break them and thus there lies a chance that a large portion of the original message may be broken. To avoid this problem, block ciphers are used in chaining mode. In this mode, the previous block cipher is mixed with the current block, to obscure the ciphertext. This allows us to avoid repeating patterns of blocks with the same plain texts.

There are several methods to encrypt data. Such methods are referred to as *block cipher modes* of operations. The following are the standard block cipher modes of operations:

- electronic-codebook mode (ECB),
- cipher-block-chaining mode (CBC),
- cipher-feedback mode (CFB),
- output-feedback mode (OFB),
- counter mode (CTR).

ECB - In this mode, the message is split into blocks, and the blocks are sequentially encrypted. This mode is vulnerable to attack using the frequency analysis, the same sort used in simple substitution. Identical blocks would get encrypted to the same blocks, thus exposing the key.

CBC - A logical operation is performed on the first block with what is known as an *initial vector* using the secret key to randomize the first block. The output of this step is logically combined with the second block and the key to generating encrypted text, which is then used with the third block and so on.

CFB –It is a stream cipher where bits are encrypted at one time. Therefore CFB does not divide data into blocks. The mode overcomes the drawbacks of ECB and CBC modes, the receiver must not wait for the entire ciphertext block to arrive before decryption can be started.

OFB - operates in the same way as that of CFB mode. The main difference between OFB mode and CFB mode is that the selected bits are supplied as input to the next round in OFB mode. Because, if any bit error occurs in a particular ciphertext, it will be propagated to all the remaining blocks in CFB. To avoid this problem, only the selected bits of the output produced by the encryption of the previous round will be given as input to the next round in OFB.

CTR - It is hardly a block cipher mode: it turns a block cipher into a stream cipher. In this mode, the block cipher algorithm won't transform plaintext data. Instead, it will encrypt blocks composed of a *counter (Ctr)* and a *nonce (N)*, which starts from an initial value and increases by 1 each time. A counter is an integer that is incremented for each block. No two blocks should use the same counter within a message, but different messages can use the same sequence of counter values (1, 2, 3, . . .). A nonce is a number used only once. It is the same for all blocks in a single message, but no two messages should use the same nonce. CTR is simple, and it overcomes the drawbacks of ECB. It is commonly used in applications that require faster encryption speed.

2. Toolset.

Only two modes of operation are included in Cryptool (CBC and ECB). OFB and CFB are available on the Online Encryption Tool. A detailed description below:

Cryptool – choose 'Encrypt-Decrypt'/ 'Symmetric (Modern)'/…

- AES (CBC)
- DES (CBC, ECB),
- 3DES (CBC,ECB),
- RC4

Online Encryption Tool - http://www.txtwizard.net/crypto:

- AES (CBC,ECB, OFB, CFB),
- DES (CBC,ECB, OFB, CFB),

3. Block ciphers and their modes of operation.

See the demonstration of the DES algorithm in Cryptool.

Prepare three texts with different entropy, each at least 1000 characters:

- ***homogenous*** *text (e.g. only one character - "hhhhhhhhhhhhhhhhhhhhh…")*
- ***medium-diversified*** *text (e.g. repeated strings of characters/ words - "agree all use city with model agree all use city with model agree all use city with model …."),*
- ***highly-diversified*** *text (e.g. normal text - "*"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis …"),

For the following algorithms IDEA, DES, AES and complete the following tasks.

I. Tasks

1. Compare the entropy value for each type of plaintext with ciphertext.

For AES(CBC)

- **Homogenous , Plain-text 0.00 & Cipher-text 7.81**
- **Medium-diversified, Plain-text 4.04 & Cipher-text 7.82**
- **Highly-diversified, Plain-text 3.97 & Cipher-text 7.83**

For DES(CBC)

- **Homogenous , Plain-text 0.00 & Cipher-text 7.80**
- **Medium-diversified, Plain-text 4.04 & Cipher-text 7.81**
- **Highly-diversified, Plain-text 3.97 & Cipher-text 7.78**

2. Compare the impact of a block and key length on the entropy of ciphertext. If applicable, try to use different available lengths and key values of selected algorithms.

- **AES(CBC)**

> Homogenous, 128 bit - 7.81 && 256 bit - 7.80

> Medium-diversified, 128 bit - 7.82 && 256 bit - 7.81

> Highly-diversified, 128 bit - 7.78 && 256 bit - 7.81

- **DES(CBC)**

  There was no option to change key length for DES.

II. Questions:

1. Which algorithms are most popular?

- **Upon lookup, AES, DES, TripleDES etc.**

2. Which parameter values (block length, key length) are nowadays considered standard (safe)?

- **64, 128, 256 bit (above 64)**

3. What can we say about the observed changes in histograms and entropy values during the realization of points 1 and 2?

- **With different algorithms entropy values differ yet not so much & with key length there is a slight drop in entropy value.**

4. Does the length of the block affect the entropy of the ciphertext?

- **No, What changes entropy is different elements in the text.**

5. Does the length of the key influence the entropy of the ciphertext?

- **Yes.**

6. Does the observed entropy of a ciphertext depend on the entropy of plain text?

- **Yes.**

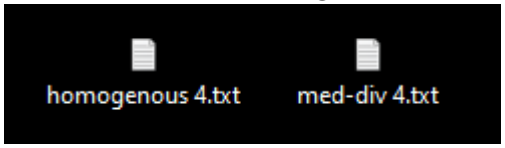
7. Does the observed ciphertext entropy depend on the algorithm used?

- **Yes yet some return similar entropy value.**

4. Modes of operation.

I. Tasks:

1. Generate a homogenous text (only one letter) and a medium-diversified text - a recurring sequence of characters e.g. "abcd," etc.

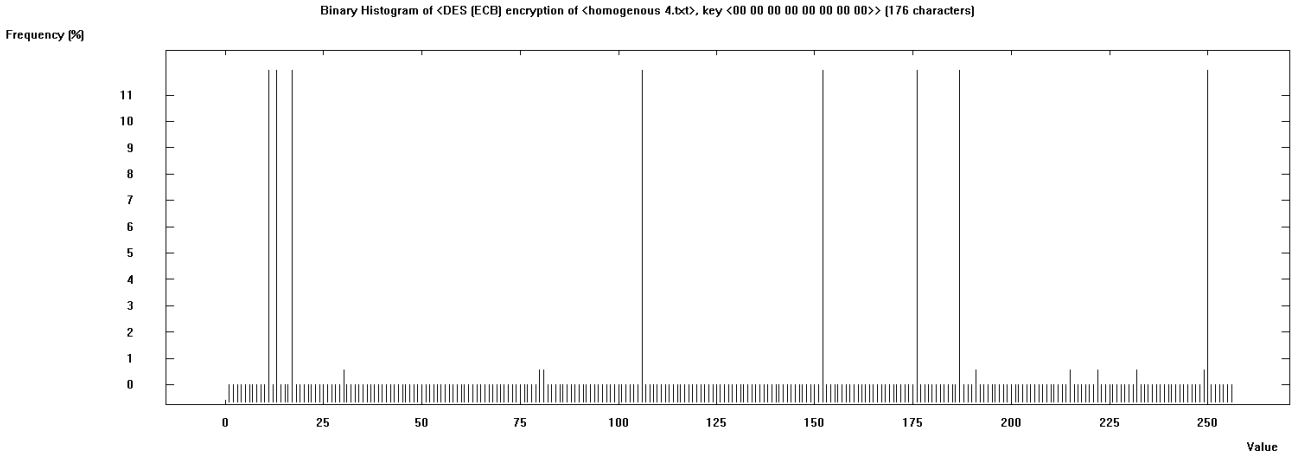


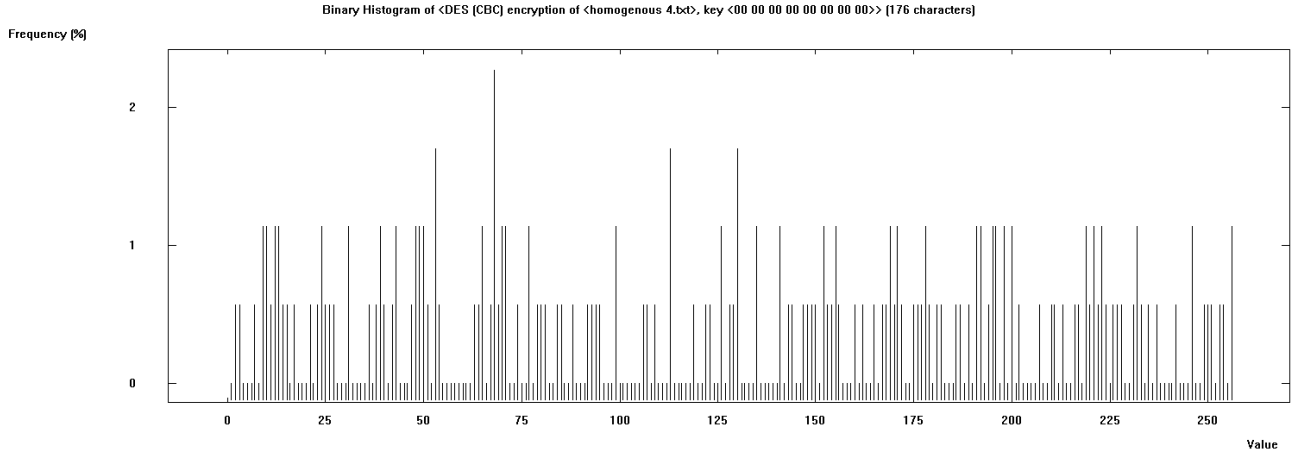
2. Choose one of the available encryption algorithms and encrypt a plain text using a different mode of operations, i.e. create ciphertexts for the modes: ECB, CBC, OFB, CFB.

- **For this i am choosing DES(ECB)**

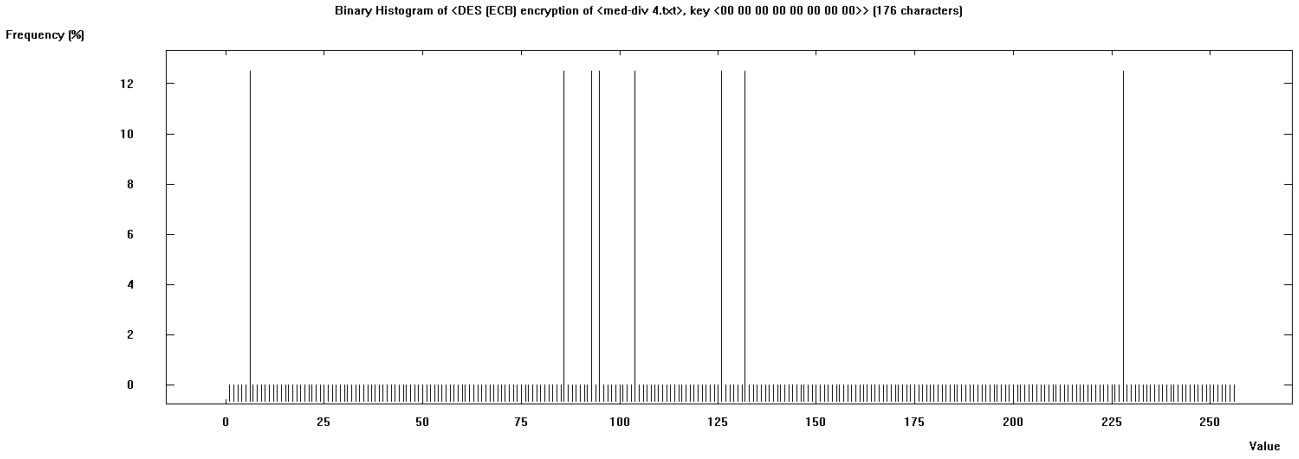
3. Check the histograms and calculate the entropy for the previously created ciphertexts.

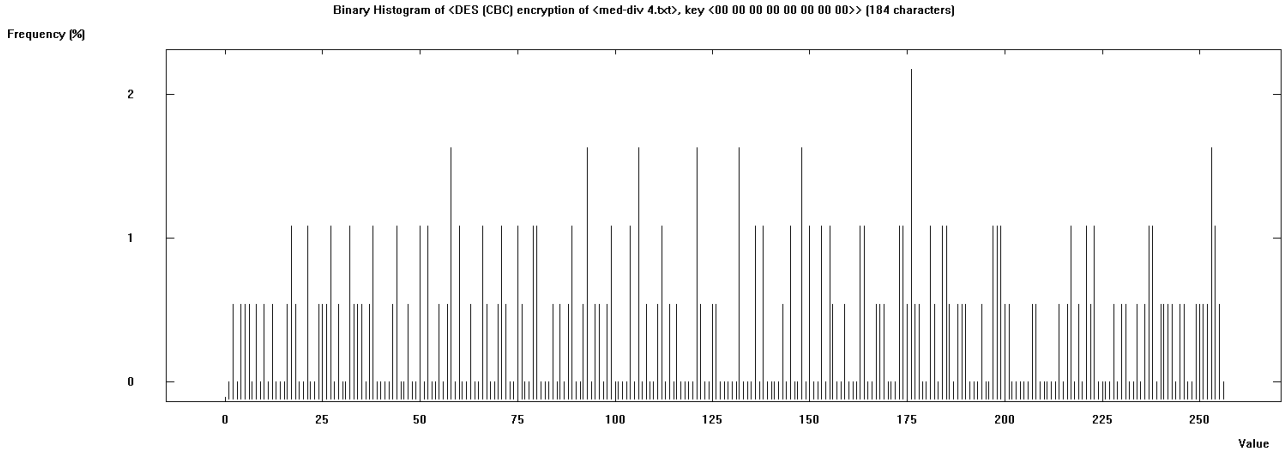
- **Homogenous ECB vs CBC**





- **Medium-diversified, ECB vs CBC**





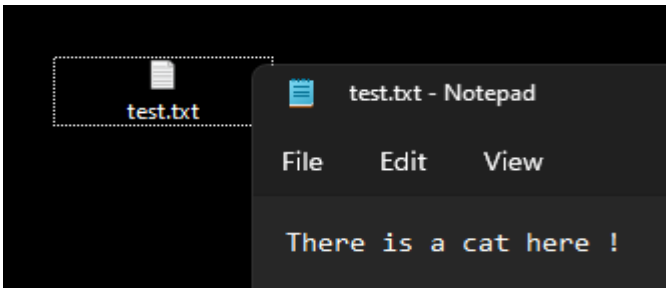
4. Check the contents of the ciphertext and evaluate the suitability of each mode of operation for encrypting files consisting of repeating blocks of bytes.

Introducing errors in the ciphertext:

- **ECB — Identical blocks would get encrypted to the same blocks.**
- **CBC — Any particular error will result in causing error in remaining blocks.**

5. Encrypt any text file using one selected algorithm operating in all modes of operation.

-

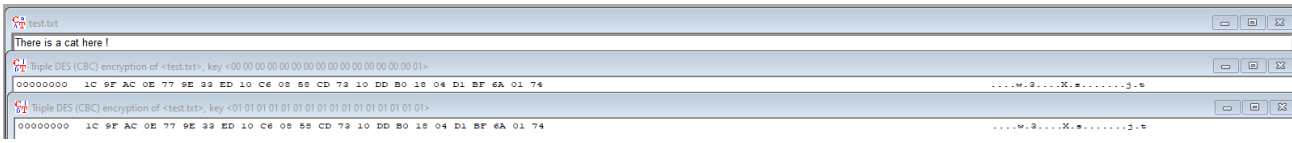


I will use TripleDES to encrypt this file.

6. Afterward, make the following changes to the ciphertext:

- change one bit,

- change one each or several bits in different bytes (close or far apart), Decrypt the ciphertexts and check their content.



- **From top Original > 1 changed bit > Multiple changed bits**

7. What will cause the following changes for AES encryption?

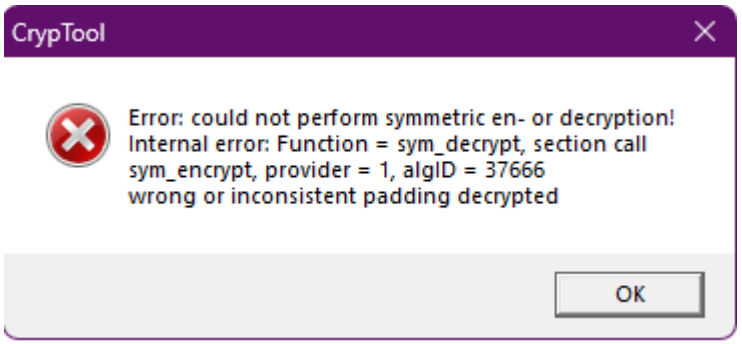
   • one byte added,

   • one byte removed,

   • a few bytes added,

   • a few bytes removed,

- **If you add or remove more bytes the AES encryption cannot decipher back to the original text.**



8. Please try for the following cryptograms to identify and motivate your decision on:

   • Mode of operation used

   • Length of algorithm block

a.

685c281585d8c0eeb59022def8488a78
685c281585d8c0eeb59022def8488a78
685c281585d8c0eeb59022def8488a78
685c281585d8c0eeb59022def8488a78
685c281585d8c0eeb59022def8488a78
685c281585d8c0eeb59022def8488a78
685c281585d8c0eeb59022def8488a78

b.

43d9f1d44ebea3e411ad70dd41b2c24e
af20e9491c13f9d419862d7f1bceca14
43d9f1d44ebea3e411ad70dd41b2c24e
af20e9491c13f9d419862d7f1bceca14
43d9f1d44ebea3e411ad70dd41b2c24e
af20e9491c13f9d419862d7f1bceca14
43d9f1d44ebea3e411ad70dd41b2c24e
af20e9491c13f9d419862d7f1bceca14
43d9f1d44ebea3e411ad70dd41b2c24e
af20e9491c13f9d419862d7f1bceca14

c.

6a239123a19647032e3e637ab0b02d56
26b593edf8445c07e9e79e408ec89e56
16d9e89bef28c92d61ab6d82d39921ac
8109e9d03dafd2fd6416a0ba97513ede
0615a2688a785319d75d95a4a9aed55e
e6c221cfa51d7e89f93a5dfbc4a2bc22
368ecba0bd42c8f56c2710723b883e5

a>inconclusive ,b>RC6 ,c>RSA-AES

II. Questions:

1. What does a ciphertext and its entropy look like for plain text with the homogeneous structure depending on the chosen encryption mode?

- **AES and DES hold almost similar values.**

2. What is the impact of errors introduced into the ciphertext after decryption, what does the plain text with changed one bit and multiple bits look like?

- **The original text will be changed. The more characters are changed the more will be changed in the original text.**

3. What does each mode of operation do with the loss of a part of the cipher text? Is it possible to retrieve plain text after removing some parts from the ciphertext?

- **With loss in ciphertext the keys will not match, that being said if there are multiple losses in ciphertext data - it will be impossible to retrieve the original text.**

4. Which of the modes of operation allows for encryption and decryption processes in parallel? Which mode of operation allows us to divide plaintext/ ciphertext into several parts to perform encryption or decryption independently (on different threads)?

- **CTR – encrypts and decrypts in parallel**
- **ECB – encrypts and decrypts independently, on multiple threads**
- **CBC – decrypts on different threads**

Lab 3 module we learned some common cipher and decipher algorithms and their uses based on key length encryption and their values comparison in different order.