MINISTRY OF EDUCATION AND SCIENCE OF UKRAINE

NATIONAL TECHNICAL UNIVERSITY

"KHARKIV POLYTECHNICAL INSTITUTE"

Department of Computer Engineering and Programming

«Software Means of Information Protection »

*Laboratory work report No 5-6*

*Topic: «* *Encryption and decryption of data using block and* streaming *algorithms»*

Student:

Group. KH 919 i.e.

Ayoub EL-HADDADI

Verified by:

Lecturer Viktor CHELAK

Kharkiv – 2022

***Purpose of work:***

Creating programs to encrypt and decrypt information using substitution ciphers.

***Individual task:***

* Information compression program (archiving). As information, use a copy of the file with the developed program.
* Program to return the archive file to its original form (unzip).

Variant 8:

Block algorithm: RC2.

Stream algorithm: Additive generators.

***Algorithm of the program:***

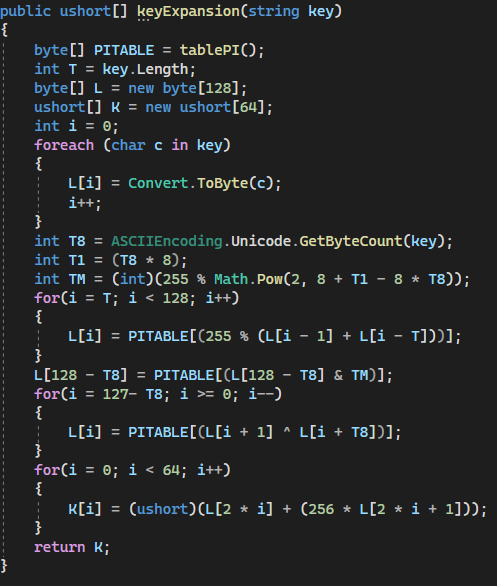
*#Block encryption / decryption (RC2):*

RC2 is a block cipher, and the block size is 8 bytes (64 bits). This means that the input data is first divided into blocks of 8 bytes and then each of them is processed separately. Each data block is treated as four words, each word has 16 bits (2 bytes).

***Key Expansion:***

Apart from the data, the RC2 cipher takes as input a secret user key. The key provided by the user may be of size from one byte up to 128 bytes. The first operation which RC2 then performs is to expand the key, to receive new 128 key bytes which will be used for encryption of decryption of all data bytes.

Then, the following steps are performed on the key data. Table PI mentioned below is a fixed array of size 256, filled randomly with values from 0 to 255:



***Encryption:***

For encryption process, we create the “Blocks”, we separate data from input file into word (16-bit). Every block of data will be encrypted by using the same 64 words of expanded secret key: K [0] K [1] ... K [63].

The following encryption steps are performed on every data block:

1. Perform five Mixing Rounds.
2. Perform one Mashing Round.
3. Perform six Mixing Rounds.
4. Perform one Mashing Round.
5. Perform five Mixing Rounds.



***Decryption:***

The decryption procedure takes as input four cipher text from encrypted file and separate it into words, which form one block of encrypted data. Every block of data will be decrypted by using the same 64 words of expanded secret key: K [0] K [1] ... K [63].

The following decryption steps are performed on every cipher text block:

1. Perform five R-Mixing Rounds.
2. Perform one R-Mashing Round.
3. Perform six R-Mixing Rounds.
4. Perform one R-Mashing Round.
5. Perform five R-Mixing Rounds.



*#Stream encryption / decryption (Additive generators):*

A stream cipher is an encryption algorithm that uses a symmetric key to encrypt and decrypt a given amount of data. A symmetric cipher key, as opposed to an asymmetric cipher key, is an encryption tool that is used in both encryption and decryption.

***Key generator:***

A Lagged Fibonacci generator (LFG) is an example of a pseudorandom number generator. This class of random number generator is aimed at being an improvement on the 'standard' linear congruential generator. These are based on a generalization of the Fibonacci sequence.

The relation used for generation of key is:

Key (n) = Key input [i] XOR Key input [i + 1]

Where i: 0 < i < file content length; and do not forget to add generated byte into Key input

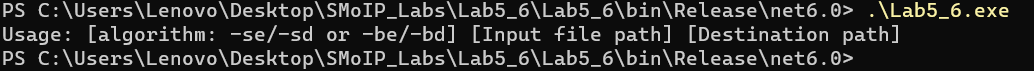
***Encryption:***

Since we have a generated key with the same length as file content (bytes), we just make an exclusive-or operator (XOR) for each bit of key and input content to create the unreadable encrypted data and store it into a binary file.

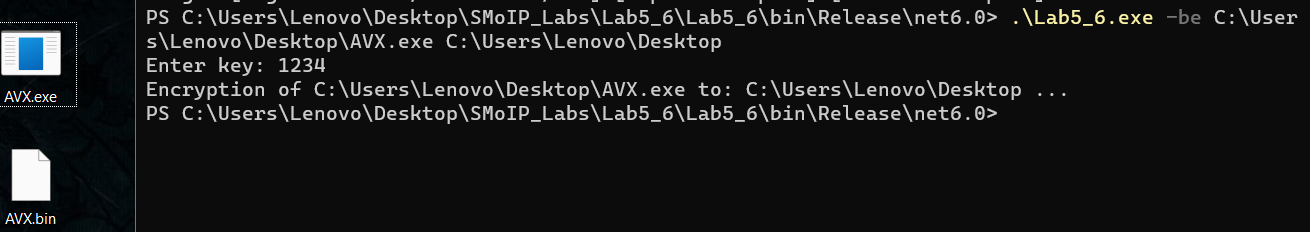
***Decryption:***

The program request from user a key and it generates key with the same key generator function. And, for decryption we perform the same function of encryption. An exclusive-or operator of every bit of encrypted data and the generated key. Finally we store decrypted data in file destination given by user.

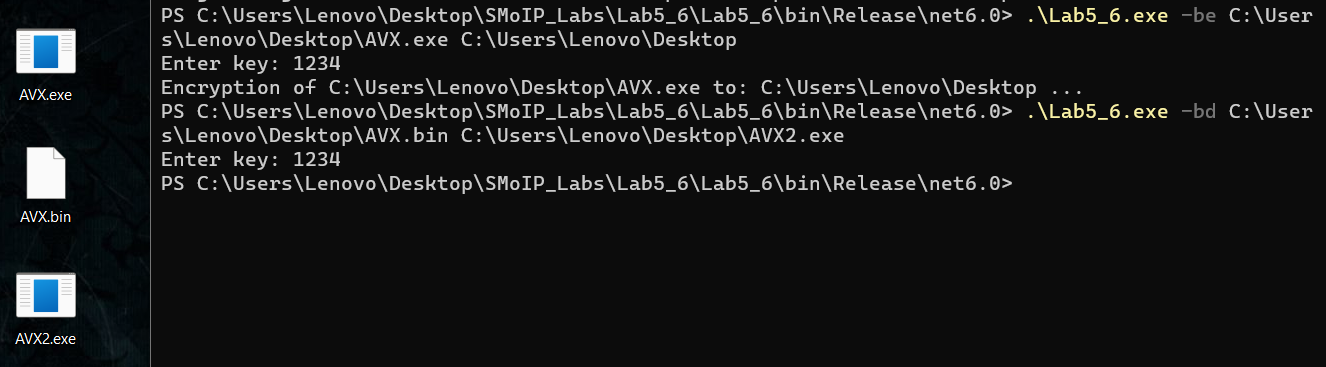
***Screenshots of the program***:



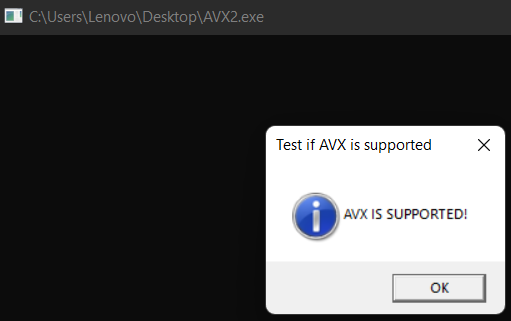
*Figure 1 – Usage of the program*

**

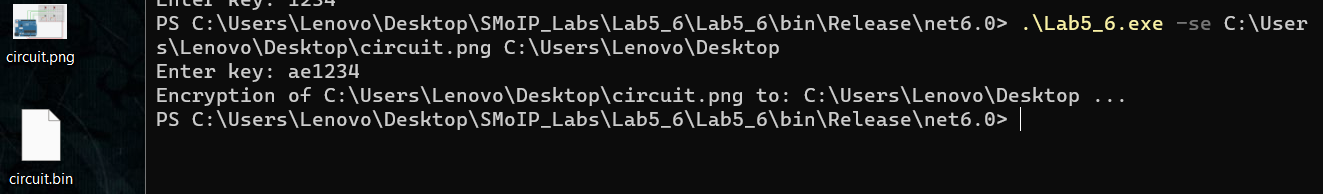
*Figure 2 – Encrypt executable file (AVX) with block algorithm*

**

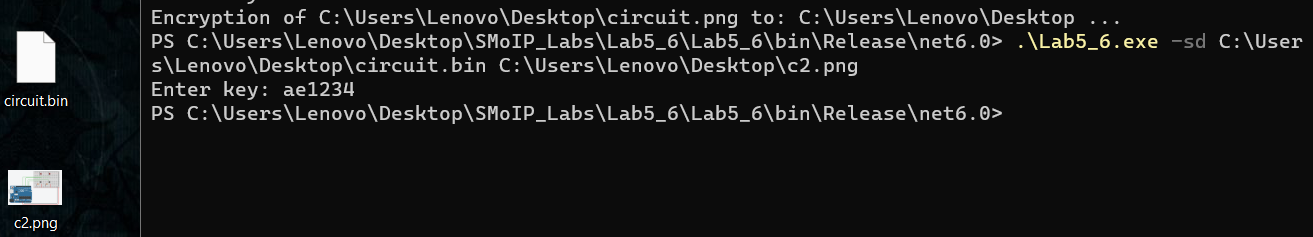
*Figure 3 – Decrypt AVX.bin file with correct key*

**

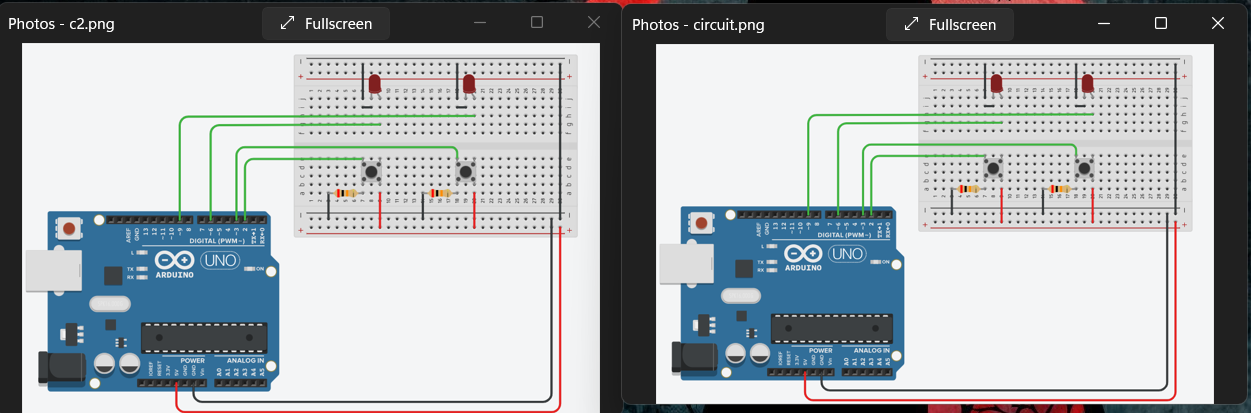
*Figure 4 – Test decrypted file*

**

*Figure 5 – Stream encryption of an image*

**

*Figure 6 – Stream decryption of the image*

****

*Figure 7 – Original image with decrypted image*

**Source Code:**

[*https://github.com/Elh-Ayoub/SMoIP\_Labs/tree/main/Lab5\_6*](https://github.com/Elh-Ayoub/SMoIP_Labs/tree/main/Lab5_6)

**Conclusions:**

For this laboratory work, I have gained principles of developing a program for encrypting and decrypting data to its original format using block and streaming algorithms.