INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Symmetric Key Encryption of Text Documents

Shubham
B. Tech Student, Electronics & Communication Engineering,
J. K. Institute of Applied Physics & Technology, University of Allahabad, India
Jayati Singh
B. Tech Student, Electronics & Communication Engineering,
J. K. Institute of Applied Physics & Technology, University of Allahabad, India

Abstract:
The issue of data security has manifested itself over the last two decades. This paper orates the implementation of private-key or symmetric-key encryption for equipment against security invasion and confidential conveyance of information from one location to another.
Presented in this paper are methodologies to encrypt any plain-text file as cipher-text and to decrypt the cipher-text to recover original text file using Exclusive-or based symmetric key encryption. The encryption algorithm explained in the paper, while extremely simple, is nearly unbreakable. This methodology has promising applications in a wide variety of fields like cryptography, digital signature etc.

Keywords: *Cryptography, Symmetric Key Encryption, Exclusive-or encryption, Text encryption, Encipher, Decipher*

1. Introduction

The art of writing or solving codes is termed as cryptography. Military, government, health and multi-national corporations are all in dire need of data privacy; hence the elevation in the field of cryptography has been immense in order to keep up with the demands of the industry. Symmetric key cryptography is a discipline of cryptography, which makes use of the same key to encrypt and decrypt data. This technique of data privacy has undergone major technological advances since the days of World War II.
In its most primitive form, this method employs a substitution algorithm of the original sequence of alphabets to be replaced by a new succession which differs from the former by a fixed pattern.
Three fundamental characteristics are associated with any algorithm of symmetric key cryptography. These include

- The same key is used for encryption and decryption.
- The key must be essentially kept private between the users; else the method might prove nugatory.
- Both the involved parties must be in possession of the key.

Other methods of implementing symmetric key cryptography include the usage of “One-time pad”, stream cipher and block cipher etc. The algorithm of symmetric key encryption and decryption discussed in this paper are based on the principle of Exclusive OR gate.

2. Principle

This method requires that the encryptor and the decryptor have access to the encryption key. The fundamental idea used in the encryption and decryption algorithms described below is the implementation of the Exclusive OR gate (XOR GATE). The Exclusive OR encryption works by using the Boolean algebra function XOR. XOR is a binary operator which works on two arguments.
Algebraically, it can be expressed as follows:
If A and B are the two input arguments, then the truth table for XOR gate is

A	B	OUTPUT
0	0	0
0	1	1
1	0	1
1	1	0

Table 1

The output is true if the inputs are different, else the output is false.
Consider the equations

- A XOR B =C
- C XOR B =A
- (A XOR B) XORB=A

It can be observed that when the output of A and B is XORed with one of the inputs, it results in the remaining input. If A is the data to be encrypted and B is the encryption key, then the data can be encrypted using this key on one side to produce some unintelligible data C. For the decryption to take place, the user on the other side will employ the same encryption key B to get the decoded data A. The idea behind Exclusive OR encryption is that it is impossible to reverse the operation without knowing the initial value of one of the arguments.

3. Methodology
Described in the succeeding text is the procedure of encryption of data. The algorithm of encryption and decryption will remain the same.

3.1. Encryption
Consider a text file which is to be encrypted. This text file is a collection of characters where each character can be represented by an 8-bit ASCII value. This text file is accepted as input in the form of a byte array. The key which will be used to encrypt the data of the text file is input as a string type byte array. Essentially, each character of the text file is then XORed with each character of the encryption key in a sequence. The resulting output is stored in an output file which is also a byte array and is of the same size as the input file array. This is used to display the unintelligible encrypted message as output.

3.2. Decryption
For decryption, the input text file array will store the encrypted message and will XOR each character of the array with the encryption key stored in the key byte array in the same sequence to produce the decrypted, readable and original data.
The equation describing the above procedure is:
A XOR B =C
C XORB=A
Where the first equation describes the encryption and the second equation describes decryption.

3.3. Encryption Algorithm
Here, a function Encode (file [], key []) will be defined with the
following parameters:

3.3.1. Input
file []: It refers to the byte array used to store the characters of the input text file. (file. Length denotes the number of bytes in the file[] byte array) Assume the input text file to be of size n.
Key []: It refers to the byte array which is used to store the characters of the encryption key. (key. Length denotes the number of bytes in the file [] byte array)

3.3.2 Output-
file2[]: It is the byte array pertaining to the output file which will be used to display the encrypted message. The output file will be of the same size as the input file.

3.4. Procedure
Three variables- “message Code”, “key Code” and “new Code” are initialized, each of which can store 8-bit ASCII values.
A variable i is initialized which is used to read the characters of the input text file one by one and the ASCII value of each variable is stored in the variable “message Code”.
The modulo operation is performed on the current value of I by the size of the input file and incremented by one. The value obtained refers to the position of the character in the input file array whose ASCII is stored in the variable “keyCode”.

The values obtained in “messageCode” and “keyCode” are then XORed, and the corresponding value obtained is stored in the variable “newCode”.
The character pertaining to the ASCII value stored in “newCode” is then read in to the output file array file2[].
This procedure is repeated for the entire length of the input file, and finally the output file is ready to display the encrypted message.
Once this procedure has been performed on all the characters of the input file, the output is placed into the file2[], for display of the output.
Here, a function Encrypt (file [], key []) will be defined with the following parameters-

Algorithm –

Encrypt (file [], key [])

- m ← file.Length
- n ← key.Length
- Initialize new byte array “file2[]” with length = n
- Initialize three variable “messageCode” , “keyCode” and “newCode” to store 8-bit ASCII values
- For i from n
  - messageCode ← getASCII_value(file[i])
  - keyCode ← getASCII_value(key[(i%n) + 1])
  - newCode ← messageCode XOR keyCode
  - file2[i] ← getCharacter(newCode)
- End Loop
- Return file2[]

4. Implementation and Observations

A JAVA Swing application was developed using Java Development Kit 8.0 to implement this methodology. The results were found to be commendable. The user interface of the application is illustrated in Figure 1.

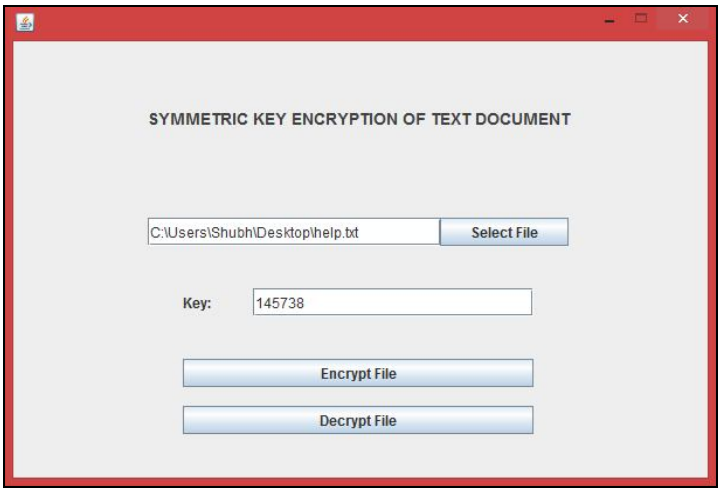

Figure 1: Application User Interface

To encrypt a text-file, it is loaded in the application using the button “Select File” and the “Encrypt File” button is clicked after entering the string to be used as the key for the encryption. Subsequently the Encoding algorithm is implemented on the loaded file and a new encrypted-file is generated. Similar is the procedure to decrypt the file. A file so generated is illustrated in Figure 2 and Figure 3.

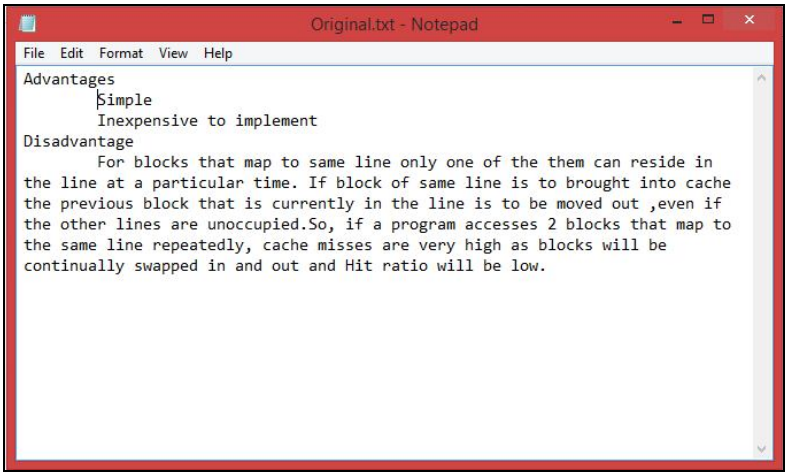

Figure 2: Typical Image:Original Text-document

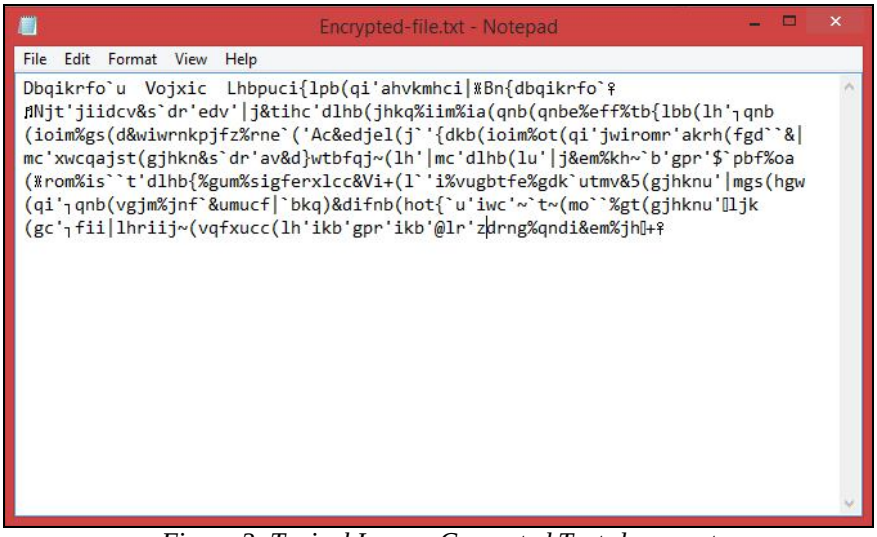

Figure 3: Typical Image: Generated Text-document

5. Conclusion

Symmetric key cryptography has undergone many stages of technological advancements since its advent. The applications of private key cryptography include ATM machines, computer passwords etc. A wide number of algorithms are available for the implementation of the same, and the scope for the development of better and faster working algorithms continues to exist in this field. Despite the straightforward technique behind this application of cryptography, its effectiveness in the protection against private data invasion is the foremost basis of the continued usage of symmetric key encryption for the purpose of classified communication.

6. References

1. Cryptography Engineering: Design Principles and Practical Applications by Niels Ferguson (Author), Bruce Schneier (Author),Tadayoshi Kohno (Author)
2. Everyday Cryptography: Fundamental Principles and Applications by Keith M. Martin (Author) Thakur, Jawahar and Kumar , Nagesh.
3. "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis". International Journal of Emerging Technology and Advanced Engineering, ISSN 2250-2459, Volume 1, Issue 2, December 2011
4. Ayushi, "A Symmetric Key Cryptographic Algorithm". International Journal of Computer Applications (0975 -8887), Vol. 1, No. 15, 2010