Bit Reversal Encryption (BiRE)

Rupesh Wagh, Rishabh Jain, Pradeep Dalal, Yudeepkumar Janbandhu, Abhilash Borkute

Information Technology, Dr. Babasaheb Ambedkar College of Engineering and Research, Nagpur, India

Abstract - This project aims to create a new encryption algorithm that changes the value of the bits containing information within a byte by regular linear reversal of bits. The idea is to provide a fast and efficient algorithm that can encrypt or decrypt any file of any format using the same amount of time as per its size, in other words the larger the file the greater the cost to encrypt but with better security. This encryption algorithm works in lower order of bits so there is no set way to determine or calculate the original plaintext from the cipher text without the key.

Keywords: Encryption, Decryption, key, Bit reversal, BiRE.

I. INTRODUCTION

We created an encryption program which reads a file of any format as an input along with a loop of input for a Key. The file is read one byte (character) at a time and then this byte is encrypted by one of the key value. The key is an array that has one value to co-relate with one byte of the file and this pair of key and byte is used for encryption of the file. The key is input one value at a time and the key value is calculated to one digit integer. Functions are called to encrypt these bytes and to decrypt the encrypted file all we need to do is take the encrypted file as input and insert it with same key in the same functions.

This algorithm can encrypt any file stored digitally on a system.

II. WORKING

The encryption is performed by rotation or reversal of bits taken 'k' numbers at a time till all bits in a byte are repositioned at least once.

Supposing, the key is k_i=3 and the 8 bits of the byte is B_i: 1010 1110 now each 3 bits is taken at a time and is reversed there after the next three bits i.e., from second position to forth is taken and reversed. This process continues till 8^{th} bit is revered at least once.

B_i = 1010 1110 = > 1011 1010 = C_i'

This result is obtained as follows:

0*101*0 1110 -> *101*0 1110

1*010* 1110 -> 1*010* 1110

10*10 1*110 -> 10*10 1*110

101*0 11*10 -> *1011 1010*

1011 *101*0 -> 1011 *101*0

1011 1010 -> 1011 1*010*

Now this encrypted byte is completely reversed to get C_i as follows: C_i' = 1011 1010 => 0101 1101 = C_i

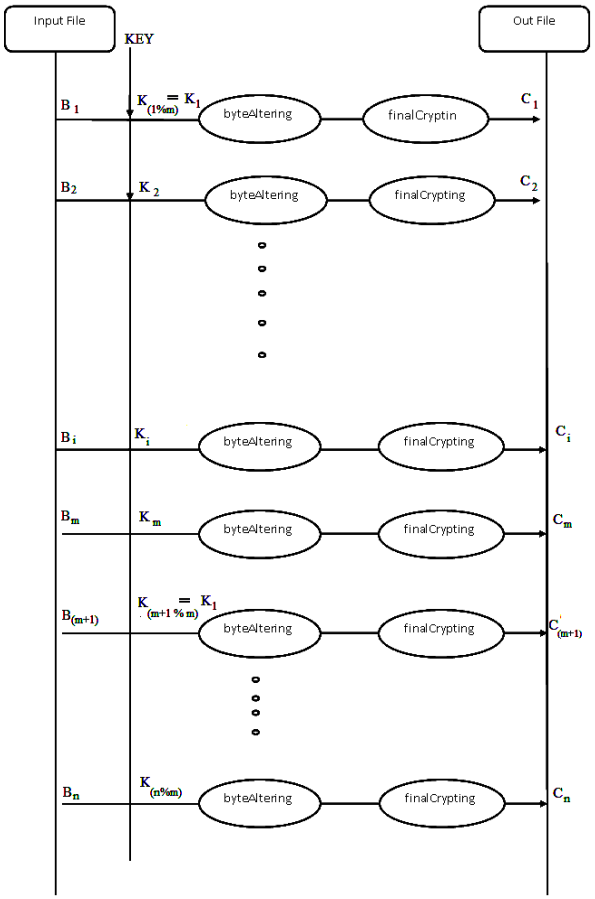
Now this encrypted byte is stored in output file and next key-byte pair (i+1) is considered. Now for a key of k_{i+1}=5 and same initial byte B_{i+1} = B_i we get a result of:

1010 1110 (= B_i) => 1110 1010 (= C_i') => 0101 0111 (= C_i)

To break the code for a small file with key of 50 numbers, we will need 10^{50} attempts or for a key of size n 10^n attempts are required.

To make the rotation easier, we use ch<<(k-1)|(ch>>(9-k)) for all odd keys and for 2. The keys 0, 1, 8, and 9 are all not very useful as you can't really rotate a bit string of their length in a byte hence we use these keys as escape keys if the need shall arise in future, presently when these keys are used, the byte is copied unaltered into the output file. Again the key 4 and 6 do not get the same result hence we use a loop to get the result for these keys.

III. MODEL



IV. ADVANTAGES

For a file of 16 byte we have a possibility of 10^{16} keys thus ensuring that the encrypted file is very secure.

Unlike previous model we can encrypt special symbols and can encrypt a complete file.

We need 2^{16} keys for any file using BREA but in our case we provide better security, i.e., for a file of n byte with key of m byte the possible keys are 10^m.

V. CONCLUSION AND FUTURE SCOPE

The system can be modified easily by changing a single line of code in the main to call more methods and adding them accordingly. Like triple DES we can create a Triple-Byte Reversal Encryption (TBRE) by simply using another array of key and calling it using a new variable.

The algorithm cold also be modifies to work with multi-threads or by using parallel computing to increase the speed and decrease the time complexity. The key can also be used to generate a new random number which will be then given to user to be used for decryption.

Again, a file could be used as a key which converts each of its byte into a single digit integer to be used as the key.

REFERENCES

[1] Sunita Bhati, Anita Bhati and S. K. Sharma, A new Approach towards Encryption Schemes: Byte-Rotation Encryption Algorithm, Proceedings of the world Congress on Engineering and Computer Science 2012, Vol II, WCECS 2012

[2] Donald Fraser, Incrementing a Bit-Reversed Integer, Computers, IEEE Transactions on (Volume:C-18, Issue: 1)

[3] Nidhi Gouttam, Implementation of Simulation of Byte Rotation Encryption Algorithm, International Journal of Technology Enhancements and Emerging Engineering Research, Vol 2, Issue 5