ISSN 2320 - 2602
Volume 3, No.4, April 2014
International Journal of Advances in Computer Science and Technology
Available Online at http://warse.org/pdfs/2014/ijacst07342014.pdf

BLOCK BASED DATA ENCRYPTION AND DECRYPTION USING A CYCLIC OPERATION

Debasish Patra[1], Sharad Jash[2], Suvajit Dutta[3], Tanumay Das[4], Prof. Dr. Pranam Paul[5]
[1]Student, MCA, Narula Institute of Technology, Agarpara, Kolkata-109, West Bengal, India, debnit90@gmail.com
[2]Student, MCA, Narula Institute of Technology, Agarpara, Kolkata-109, West Bengal, India, sharad99jash@gmail.com
[3]Student, MCA, Narula Institute of Technology, Agarpara, Kolkata-109, West Bengal, India, suvajit.brj@gmail.com
[4]Student, MCA, Narula Institute of Technology, Agarpara, Kolkata-109, West Bengal, India, tanumay.das@gmail.com
[5]Computer Application, Narula Institute of Technology, Agarpara, Kolkata-109, West Bengal, India, pranam.paul@gmail.com

ABSTRACT

In this paper we have developed a new symmetric key algorithm for encryption and decryption of data which uses a cyclic operation. With the growth of internet and network, the need for secure data transmission become more and more essential and important, as security is a major concern in the internet world. Data likely to be kept hide from all people except from the authorized user cannot be sent in plain text. So the plain text should be codified by the process of encryption. Each type of data has its own features; therefore different techniques should be used to protect confidential data from unauthorized access. Here we introduced a new algorithm which is based on block and level data encryption and decryption technique. In this algorithm encryption is done on binary file so it can be applicable for any type of data such as text as well as multimedia data. Here the same idea of cryptography is working (i.e. using key, conversion of plain text into cipher text called encryption and the reverse, means cipher text to plain text called decryption).

Keywords: Cryptography, Encryption, Decryption, Plain Text, Cipher Text, Symmetric Key

1. INTRODUCTION

The rapid growth of computer networks allowed larger files, such as digital image, text to be easily transmitted over the internet. Data encryption is widely used to ensure security of those data. I introduce a block based symmetric key encryption algorithm. For encryption a key is to be generated. Key length and bit stream is chosen at random.

At first, the plain text has been decomposed into some blocks. Maximum defined bit – length among consecutive randomly defined number of blocks has been calculated. From the randomly taken key having the same size with block, with calculating prime factor, square, add, modulo, sorting and finally applying the cyclic operation some effective keys are generated and also cipher test will be produced. To get back the original text the operation is called the decryption. The cipher text will produced the plain text using decryption algorithm.

In section 2, algorithm is defined. While section 3 shows the example of whole process. An analysis has been done in section 4, along with conclusion.

2. ALGORITHMS

In this section, Key generation is discussed in section 2.1. In the section 2.2 and 2.3 discussed about the encryption process and decryption process respectively.

2.1. Key Generation

The Key used to be in Encryption and Decryption process consists of two segments as follows. Figure 1 shows structure of key generation.
Segment 1: Block Size
Segment 2: Level

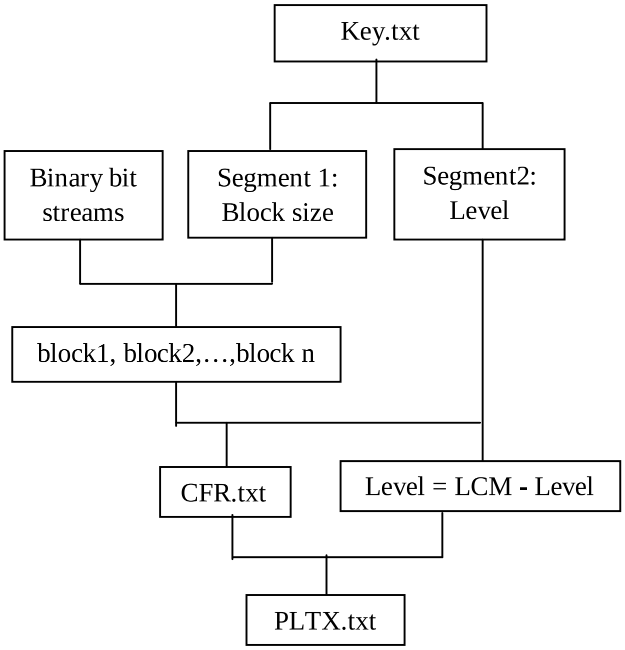


Figure 1. Structure of Key Generation

Step 1: Using block size (blk_sz) the binary bit streams are divided into separate block in file BIN.txt.
Step 2: Level used in circular representation to convert Plain text binary file BIN.txt into Cipher text CFR.txt.
Step 3: In decryption block size (blk_sz) are used same as encryption. But level is used in a different manner like-----

New Level = LCM – Level

New level is used in circular representation to convert cipher text file CFR.txt into decrypted plain text file PLTX.txt

2.2. Encryption Process

Step 1: Convert the file to be encrypted into a binary file.
Step 2: Segment the binary file into binary streams of n bits.

Step 3: Depending on the number of blocks i.e., first find how many numbers are generated from the n bit binary number i.e., 2^n such that, for n=4 the corresponding decimal value is from 0 to 15.
Step 4: From the every decimal number calculate prime factor, then square of every factor and addition and keep it in an array, say S[i]. Then find modulo 2nd of S[i] and keep it into an array, say TEMP[i] for all i=0,1….n-1 in the following manner-----

$\text{Dec\_no} \xrightarrow{\text{prime factor (pf)}} \text{pf1,pf2,…,pf-k} \xrightarrow{\text{Square and Add}}$

$\text{Pf1}^2,\text{pf2}^2,\ldots\text{pf-k}^2 \xrightarrow{\text{put in}} \text{S[i]} \xrightarrow{\%2n} \text{TEMP[i]}$

Step 5: Apply descending order sorting on TEMP[i] and result put in an array, say RES[i] and find corresponding array index value according to sorting value in RES[i] into an array, say RESULT[i]. And each index of RESULT[i] denotes the corresponding value.

Step 6: Apply a circular representation to RESULT[i] such that, for each and every index starting from an index ‘i’ denotes to corresponding value of RESULT[i], then goes to index according to the value of RESULT[i] which also denotes another value of RESULT[i] and so on. Repeat step until getting the starting index, along with need to count steps or level i.e., after how much level reach to starting index.

Step 7: Find LCM of every counted level value. Create a circular representation and repeat Step 6 until N (i.e., result of LCM) levels completed.

Step 8: Using 2nd segment of key i.e., level, say L, for any decimal value in circular representation after computing L level, the computed value will be the Cipher text of corresponding decimal value.

Step 9: We convert the encrypted value into binary streams of n bit Cipher text and write to be the encrypted file.

2.2. Decryption Process

Step 1: Read the encrypted file.

Step 2: Read the binary streams and divided in segments of n number of bits.

Step 3: The decryption process is same as the encryption process. Only one thing is different that is, in this decryption process the level is to be used, is calculated by subtracting the second segment of key i.e., level from LCM, in the following manner----

Level = LCM – Level

After executing the level in the circular representation occurred decimal value will be the final decrypted decimal value.

Step 4: We convert the decrypted decimal value into binary streams of n bit and also in plain text and write to be the decrypted file.

3. EXAMPLES

3.1. Structure of Key:

Consider the text 01000001 01101101 01101001. The length of the plain text = 24

The key text Key.txt consists of two segments as follows-----

1) Segment 1 : Block size
2) Segment 2 : Level

Let the block size = 4 and level = 5.

Step 1: Using block size 4 binary bit streams are divided into 4 bit separate block in BIN.txt.

Step 2: Using level 5 binary file from BIN.txt is converted into cipher text CFR.txt.

Step 3: In decryption level 5 is used as following manner-----

Level = LCM – level

This calculated level is applied for converting the cipher text to decrypted binary text.

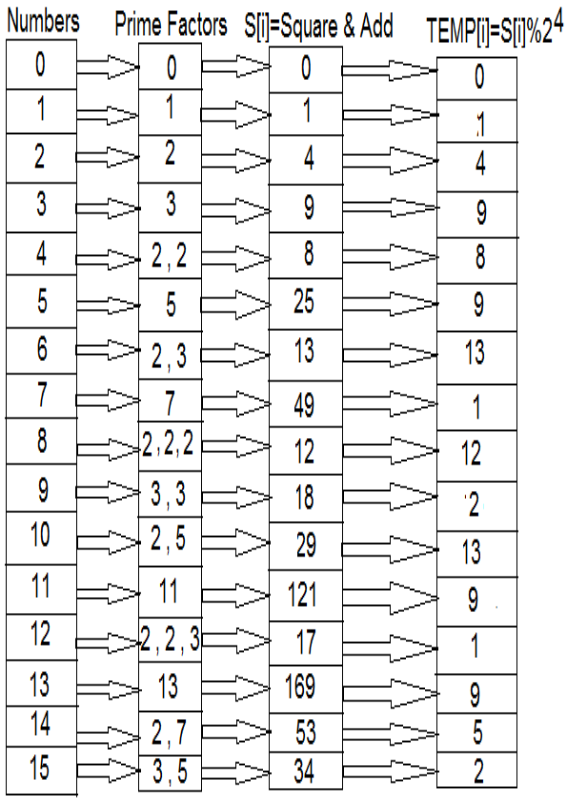
3.2. Encryption Process

Step 1: Fist the plain text is converted into binary form as -----
Plain text = 01000001 01101101 01101001.

Step 2: Binary streams are divided into 4 bit individual blocks as block size is 4.

Step 3: 4 bits block represent the 2^4 i.e., 16 decimal number. This decimal number is from 0 to 15.

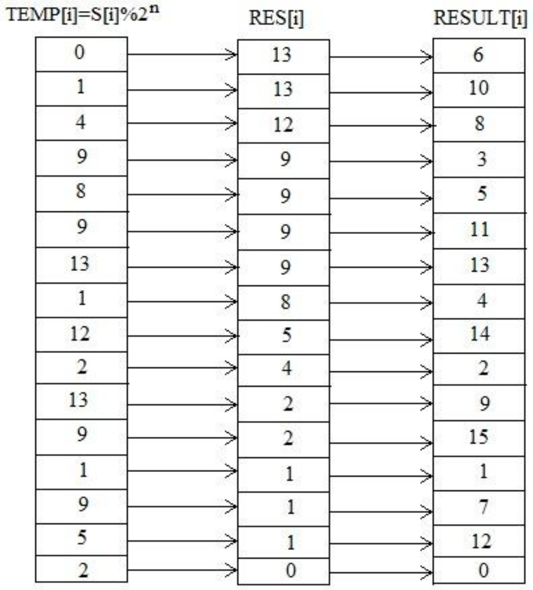
Step 4: Figure 2 shows that the numbers are converted from one form to another form by the different operations---



Numbers	Prime Factors	S[i]=Square & Add	TEMP[i]=S[i]%2^4
0	0	0	0
1	1	1	1
2	2	4	4
3	3	9	9
4	2,2	8	8
5	5	25	9
6	2,3	13	13
7	7	49	1
8	2,2,2	12	12
9	3,3	18	2
10	2,5	29	13
11	11	121	9
12	2,2,3	17	1
13	13	169	9
14	2,7	53	5
15	3,5	34	2

Figure 2. Encryption step 4

Step 5 : In the Figure 3 TEMP[i] all values are sorted in descending order and keep it in RES[i] and the final array RESULT[i] holds the value, which value are denoted by their corresponding index value as follows----



TEMP[i]=S[i]%2^n	RES[i]	RESULT[i]
0	13	6
1	13	10
4	12	8
9	9	3
8	9	5
9	9	11
13	9	13
1	8	4
12	5	14
2	4	2
13	2	9
9	2	15
1	1	1
9	1	7
5	1	12
2	0	0

Figure 3. Encryption step 5

Step 6: In the array RESULT[i], the index is denoting to the corresponding value of the array, one decimal represents another number as follows-----

0→6
1→10
2→8
3→3
4→5
5→11
6→13
7→4
8→14
9→2
10→9
11→15
12→1
13→7
14→12
15→0

And from this representation the circular representation will be created as follows-----

0→6→13→7→4→5→11→15→0 = 8(i.e., after computing 8 level 0 back to the previous form i.e., zero becomes zero).

1→10→9→2→8→14→12→1 = 7 (i.e., after 7 level 1 becomes 1).

3→3 = 1(i.e., after level 1 3 becomes 3).

Step 7: Calculate then LCM.

Therefore the LCM of (8, 7, 1) = 8*7*1 = 56.

Step 8: The binary text is divided into 4 bit blocks. And the text is 010000010110110101101001. Then 1st 4 bit '0100' is equal to '4' decimal value.

In the above circular representation, after 5 level 4 will be decimal '0'= 0000(in binary). 2nd 4 bit '0001' is 1 which represents after 5 levels, decimal 14=1110(in binary). By the same process----

3rd 4 bit '0110' will be '1011'.

4th 4 bit '1101' will be '1111'.

5th 4 bit '0110' will be '1011'.

6th 4 bit '1001' will be '0001'.

Step 9: Thus we will get the encrypted text or cipher binary text.

3.3. Decryption Process

In the decryption process same thing will be done as encryption process. Only one thing is different, that is the level is to be used in decryption will be-----

Level = LCM – level
=56 – 5
=51

Therefore, after 51 levels the converting decimal cipher text denotes another one decimal number. That number will be the decrypted decimal value.

Example: Cipher text 1st 4 bit ‘0000’ will be reached in ‘0100’ after calculating 51 levels. And so on.

And finally converting this decimal into binary then converting into character the final decrypted plain text will be occurred.

4. ANALYSIS

In this algorithm encryption is perform on binary data. All data which is under stable by the computer is finally converted into binary bits. So it can be implemented for any data type encryption process such as text encryption, image encryption i.e., multimedia encryption process.

4.1. In the structure of key

Not only that, the key has two segments and both two segments i.e., block size and level are user defined. That is the specialty of this encryption and decryption algorithm.

The block size is not fixed in this algorithm, so we can take large number of block number for making it more complex. If the block size is assumed ‘n’ then how much numbers are generated from ‘n’ bit number, the calculation will be possible. Also level is user defined but there is a restriction i.e., maximum range of level should be same as LCM.

In this algorithm the length of the plane text is not restricted, so it can be applicable for any large file.

Now after encryption is carried out the encrypted file generated is again a binary file. Now while attempting to decrypt this file will be particularly easy since the algorithm is same as the encryption process. Only thing is that new level is calculated from LCM and key segment level, which can be easily possible.

4.1 Size Comparative Report

This algorithm has been implemented on number of data files varying types of content and sizes of wide range, shown in Table 1.

Table 1

Serial Number	Source File Name	Original Plain Text (bytes)	Decrypted Plain Text (bytes)
1	app.txt	424	424
2	b.txt	2,570	2,570
3	c.txt	5,132	5,132
4	Deb.tif	442,188	442,188
5	Sum.OBj	933	933
6	prime.cpp	530	530
7	Lake.jpg	7,133	7,133
8	Getch.exe	14,773	14,773

Here we can see that before encryption and after decryption the file size is same by the Size analysis.

4.2. Time Comparative Report

In Table 1, a comparison on basis of encryption time and decryption time with their file size has been shown here on the same set of files used in Table 2.

Table 2

Sl. No	File Size(bytes)	Encryption Time	Decryption Time
1	424	0.000000	0.054945
2	2,570	0.219780	0.274725
3	5,132	0.384615	0.109890
4	442,188	0.000000	0.000000
5	933	0.000000	0.000000
6	530	0.000000	0.000000
7	7,133	0.000000	0.000000
8	14,773	0.054945	0.000000

5. CONCLUSION

My conclusion towards this algorithm is that I have tested the implementation of this algorithm and this algorithm worked correctly for the above set of values. From this we can assume that algorithm can correctly be implemented for very large values of block size ‘n’ and level ‘L’. Since very large values can be used for encryption and decryption and large files can be encrypted using the same. Hence we say our algorithm is quite secured.

REFERENCES

[1] William Stallings,”Cryptography and network security principles and practices”, 4th edition, Pearson Education, Inc., publishing as Prentice Hal, 2006.

[2] Pranam Paul, Saurabh Dutta, A K Bhattacharjee,”An Approach to ensure Security through Bit-level Encryption with Possible Lossless Compression”, International Journal of Computer Science and Network Security”, Vol. 08, No. 2, pp. 291 – 299,2008.

[3] Sanjit Mazumdar, Sujay Dasgupta, Prof.(Dr) Pranam Paul,”Implementation of Block based Encryption at Bit-Level”, International journal of Computer Science and Network Security, Vol. 11, No.2, pp. 18-23, 2011.

[4] Sujay Dasgupta, Sanjit Mazumdar, Prof.(Dr) Pranam paul, “Implementation of Information Security based on Common Divison”, International journal of Computer Science and Network Security, Vol. 11, No.2,pp. 51-53, 2011.

[5] Somdip Dey, “Amalgamation of Cyclic Bit Operation in SD-EI Image Encryption Method: An Advance Version of SD-EI method: SD-EI Ver-2”, International journal of Cyber-Security and Digital Forensic,1(3):221-225, 2012.

[6] http://en.wikipedia.org/wiki/Symmetric-key_algorithm

[7] http://searchsecurity.techtarget.com/Understanding-encryption-and-cryptography-basics

[8] Asoke Nath, Saima Ghosh, Meheboob Alam Mallik,”Symmetric Key Cryptography using Random key Generator”, Procedding of International conference on security and management(SAM’10” held at Las Vegas, USA Jull 12-15,2010), P-Vol-2, pp. 239-244,2010.

[9] Pranam Paul, Saurabh Dutta, “An Enhancement of Information Security using Substitution of Bits Through Prime Detection in Blocks”, Proceeding of National Conference on Recent Trends in information Systems(ReTIS-06), Organized by IEEE Gold Affinity Group, IEEE Calcutta Section, Computer Science & Engineering Department, CMATER & SRUVM Project-Jadavpur University and Computer Jagat.

[10] Dripto Chatterjee, Joyshree Nath, Suvadeep Dasgupta, Asoke Nath, “A new Symmetric key Cryptography Algorithm using extended MSA method: DJSA symmetric key algorithm”, International Conference on Communication Systems and Network Technologies, 978-0-7695-4437-3/11,$26.00 © 2011 IEEE.

[11] A.Nath, S.Das, A.Chakraborti, “Data Hiding and Retrieval”, Proceedings of IEEE International conference on Computer Intelligence and Computer Network held at Bhopal, pp. 26-28, Nov, 2010.

[12] Text book Atul kahate, “Cryptography and Network Security”, Tata McGraw-Hill Education, 2003