ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Implementing Data Security by Representing the Line of a Figure in nth Dimension

Debashis Sar[1],Prof. (Dr.) Pranam Paul.[2]

MCA Final Year Student, Narula Institute of Technology, Agarpara, Kolkata, West Bengal, India[1]

HOD, Department of Computer Application, Narula Institute of Technology, Agarpara, Kolkata, West Bengal, India[2]

ABSTRACT: In recent days, for secure information transmission through internet, Cryptography is used. Here for secure data communication the plain text would be encrypted into cipher text using encryption process. This encrypted text along with the key or information would be send by the sender at receiver's end. Then using the key or information, the receiver would able to decrypt the encrypted text. Using this base idea there exist different algorithm for encryption and decryption and for key generation. Here our basic idea is base on representing the line of a figure in nth dimension .The strength of the technique is analysed in this paper. This is a private key cryptographic technique. From the bit level corresponding decimal value is obtained .The process is later discussed in details in this paper.

KEYWORDS: Cryptography, Cipher Text, Decryption, Encryption, Plain Text, Symmetric Key, nth dimension space.

I. INTRODUCTION

For secure information transmission through internet, as the complexity of the threats increases, so the security measures required to protect networks. In order to protect data from unauthorized intruder data must be transmitted in encrypted form. To achieve this goal, network security and cryptography has now become an emerging research area to develop encryption algorithm, decryption algorithm, key generation algorithm and key matching algorithm for proper secure transaction from sender to receiver, avoiding any middle attacker. To be secured, information needs to hidden from unauthorised access (middle attack), protected from unauthorised change, and available only to the sender and receiver. Cryptography, not only protects data from hacking or alteration, but can also be used for user authentication. The scenario of present day of information security system includes confidentiality, authenticity, integrity, and non-repudiation. Security breaches can often be easily prevented. How? This guide provides you with a general overview of the most common network security threats and the steps you and your organization can take to protect yourselves from threats and ensure that the data travelling across your networks is safe .Each type of data has its own features; therefore different techniques should be used to protect confidential data from unauthorized access. Here the same idea of cryptography is working. After encryption the encrypted file size can be decreases or increases based on some component related to the algorithm and the file on which the encryption process will apply and also for encrypted file size decrease, it results possible lossless compression.

II. RELATED WORKS

We can represent a square in two dimensional space. In a rectangle there is four vertices, four side and two diagonal. If we give direction any line then it will be treated as vector. If we give this vector concept in rectangle then there will be four vertices. Each vertices treated as NULL vector. So there will be 4 NULL vector, 8 SIDE vector and 4 DIAGONAL vector. Now we can represent a cube in three dimensional space. In a cube there is 8 NULL vector, 24 SIDE vector and 32 DIAGONAL vector. Thus for n dimensional figure will be 2^n NULL vector points, n*2 no of SIDE vector and $2^n(2^n$-n -1) no of DIAGONAL vector where n=dimension.

III. PROPOSED ALGORITHM

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.

DOI: 10.15680/IJIRCCE.2016. 0405146

2.1. Key Generation

Our algorithm is based on private key operation. We can choose any number as key of the form 2^k.

2.2.Algorithm of Encryption:-

STEP 1: At first we need to convert the plain text into its binary form thus get a source bit stream.
STEP 2: Here we take n=2^k as a 'key' to generate the block size using the formula as bellow-
Block size = 2^k. In this algorithm we take 4 as key.
STEP 3: We divide the (0-255) ASCII value in three part i.e NULL VECTOR,SIDE VECTOR and DIAGONAL VECTOR. The NULL,SIDE and DIAGONAL VECTOR mention in above.
There will be 16 NULL vector,64 SIDE vector and 176 DIAGONAL vector.
STEP 4: Now decompose the source bit stream into $2*2^k$ bit stream. Now we take first block and checked what vector it is. If we determine the vector then we take the positional value of the bit stream of that vector.
STEP 5: Ifthe determined vector is NULL vector then put 00 at first then we represent the positional value in 4 bit stream, if the determined vector is SIDE vector then put 01 at first then we represent the positional value in 6 bit stream. We decompose the DIAGONAL vector in 3 parts given bellow.
i)If the positional value(pv) of DIAGONAL vector is less 128(2^7) then put 10 at first then we represent the positional value in 7 bit stream.
ii)we do pv-128(2^7) then we put 11 at first and if that positional value is less than 32(2^5) then we append 0, then it represent the position value in 5 bit stream.
iii) Now we do pv-32(2^5) then append 1 at first then it will be represented the positional value of 4 bit stream.
STEP 6: Like this ways source block is encrypted into a target block. This process will be continued for all the source bit block and finally get the source bit stream.
STEP 7: Now the source bit stream is converted into the it's byte code value and finally we get the encrypted text. To do this those bits are not involved to convert the encrypted text it is kept into the 2nd segment of key.

2.3.Algorithm of Decryption:-

STEP 1: Convert the cipher text into its binary form and we get bit stream.
STEP 2: Now we take two bit at a time. If it is 00 then we take next 4 bit as a position of NULL vector.
STEP 3: If it is 01 then we take next 6 bit as a position of SIDE vector.
STEP 4: If it is 10 then we take next 7 bit as a position of DIAGONAL vector. If it is 110 then also we take next 5 bit as a DIAGONAL vector and do pv+128(2^7). If it is 111 then also we take 4 bit as a DIAGONAL vector and do pv+128+32.Here pv is treated as positional value.

IV. EXAMPLES

3.1. Key Generation

Our algorithm is based on private key operation. We can choose any number as key of the form 2^k. For example here we choose the block size as key i.e 4.

3.2. Encryption Process

Consider the text "ENCRYPTION "as plain text.
STEP 1: First each character of the plain text is converted into its corresponding ASCII value.

E → 69 N → 78 C → 67 R → 82 Y → 89 P → 80
T → 84 I → 73 O → 79 N → 78

Now each ASCII value converted into its binary form of 8 numbers of bits. And we get a binary stream for the plain text as below—
01000101 01001110 01000011 01010010 01011001 01010000 01010100 01001001 01001111 01001110
STEP 2: Decompose source bit stream into some blocks with given block size. Here the block size is 4.
0100 → 0101 (SIDE vector)
0100 → 1110 (DIAGONAL vector)
0100 → 0011 (DIAGONAL vector)

0101 → 0010 (DIAGONAL vector)

0101 → 1001 (DIAGONAL vector)

0101 → 0000 (DIAGONAL vector)

0101 → 0100 (SIDE vector)

0100 → 1001 (DIAGONAL vector)

0100 → 1111 (DIAGONAL vector)

0100 → 1110 (DIAGONAL vector)

STEP 3: Now the position of SIDE vector and Diagonal vector are counted with the treating as following two points in the figure of 2^k dimensional space.

0100 → 0101 (pv=17)

0100 → 1110 (pv=53)

0100 → 0011 (pv=46)

0101 → 0010 (pv=56)

0101 → 1001 (pv=60)

0101 → 0000 (pv=55)

0101 → 0100 (pv=21)

0100 → 1001 (pv=49)

0100 → 1111 (pv=54)

0100 → 1110 (pv=53)

Here pv refers to positional value.

STEP 4: Now we represent the source bits into their position values and convert into binary stream. For SIDE vector we append 01 at first and convert it into 6 bit streams. For DIAGONAL vector append 10 at first and convert it into 7 bit streams. Now we get the bit streams as follows:

01 010001

10 0110101

10 0101110

10 0111000

10 0111100

10 0110111

01 010101

10 0110001

10 0110110

10 0110101

STEP 5: Now we decompose the source bit into 8 bit streams as follows.

01010001→81

10011010→154

11001011→203

10100111→167

00010011→19

11001001→201

10111010→186

10101100→172

11000110→198

01101101→109

00110101→53

STEP 6: Now we get the cipher text as follow.

81→ ‘Q’

154→ ‘š’

203→ ‘Ë’

167→ ‘§’

19→ ‘’

201→ ‘É’
186→ ‘º’
172→ ‘¬’
198→ ‘Æ’
109→‘m’
53→ ‘5’

3.3 Decryption process

STEP 1: Consider the“QšË§Éº¬Æm5” as cipher text and convert each character to its corresponding ASCII value.

‘Q’→81
‘š’→154
‘Ë’→203
‘§’ →167
‘’ →19
‘É’→201
‘º’→186
‘¬’→172
‘Æ’→198
‘m’→109
‘5’→53

Now each ASCII value converted into its binary form of 8 numbers of bits. And we get a binary stream for the cipher text as below—

0101000110011010110010111010011100010011110010011011101010101100
1100011001101101001101101

STEP 2: Now we check the 1st two bit of the source bit stream and continuing STEP 2, STEP 3 and STEP 4 of decryption process.

Here we get 01 in front of source bit stream so we take next 6 bits as positional value of SIDE vector. This process will be continued until unless the source bit stream will be end. Sowe get the position value as binary stream as follow-

010001 0110101 0101110 0111000 0111100 0110111 010101 0110001 0110110 0110101.

STEP 3: Now we convert the binary stream into its corresponding decimal value to get the positional value of corresponding vectors as follow-

010001→17 (pv of SIDE vector)
0110101→53 (pv of DIAGONAL vector)
0101110→46 (pv of DIAGONAL vector)
0111000→57 (pv of DIAGONAL vector)
0111100→60 (pv of DIAGONAL vector)
0110111→55 (pv of DIAGONAL vector)
010101→21 (pv of SIDE vector)
0110001→49 (pv of DIAGONAL vector)
0110110→54 (pv of DIAGONAL vector)
0110101→53 (pv of DIAGONAL vector)

STEP 4: From every positional value we get binary stream of 8 bits and convert all 8 bits binary stream into its corresponding decimal value as follow-

01000101→69
01001110→78
01000011→67
01010010→82
01011001→89
01010000→80
01010100→84
01001001→73
01001111→79
01001110→78

STEP 5: Each decimal value is now converted to its corresponding ASCII character and get the plaintext.
PLAIN TEXT →ENCRYPTION

V. RESULT AND ANALYSIS

In this section we are analysing the whole matter in a different kinds of aspects. Section 4.1 is analyzed how to detect the NULL, SIDE and DIAGONAL vectors where as in section 4.2, a comparative report between encryption and decryption time is been shown. Analyzing the strength of the algorithm, depending the key is discussed.

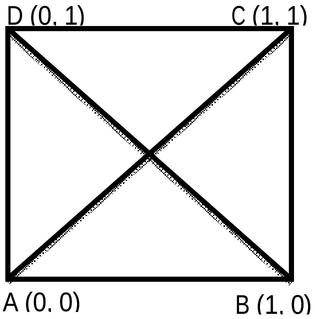
4.1 Analysis for Detecting the different Vector:

We are considering consecutive two source bit blocks as a two points of a figure of 2^kdimension. Now we are XORingbetween two points and whatever the result is come depending on that we can detect NULL vector, SIDE vector and DIAGONAL vector as follow.

NULL VECTOR: If we get all are 0 after XORing between two points then it is treated as NULL vector.
EXAMPLE: i) 11⊕11→00 (2 dimensional)
ii)1011⊕1011→0000 (4 dimensional)

SIDE VECTOR: If we get only single 1 after XORing between two points then it is treated as SIDE vector.
EXAMPLE: i)10⊕11→01 (2 dimensional)
ii)1011⊕1111→0100 (4 dimensional)

DIAGONAL VECTOR: If we get minimum two 1 after XORing between two points then it is treated as DIAGONAL vector.
EXAMPLE: i) 11⊕00→11 (2 dimensional)
ii) 1001⊕0010→1011 (4 dimensional)



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. Therefore that encryption technique can be used fortext encryption, image encryption i.e., multimedia encryption process.

4.2 Size and Time Comparative Report

This algorithm has been implemented on number of data files varying types of content and sizes of wide range, shown in 4.2.1.

Table: 4.2.1
Size and Time Comparative Table of encryption

SL. No.	Original File Size(byte)	Encrypted File Size(byte)	Encryption Time	Encryption Time/byte
1	10	11	0	0
2	70	77	0	0
3	210	231	0.21978	0.000951429
4	400	440	0.384615	0.000874125
5	610	671	0.659341	0.000982624
6	880	968	1.263736	0.001305512
7	1044	1147	1.153846	0.001005969
8	1239	1,362	1.428571	0.001048877
9	1710	1884	1.923077	0.001020742
10	2028	2232	2.252747	0.001009295

Table: 4.2.1 shows time, taken for encryption for different file size i.e. Original file size and time taken for encryption for each byte and encrypted file size. From the above table data we draw the following figure.

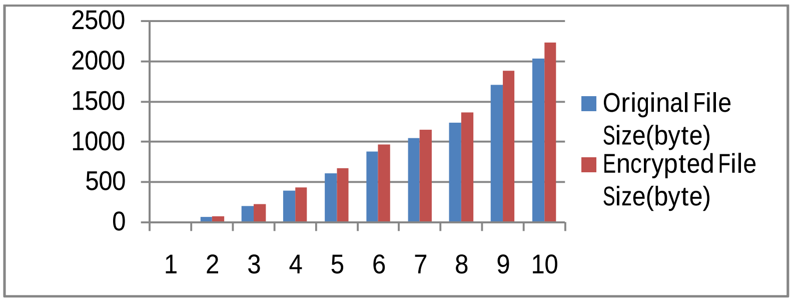


Fig: 4.2.1
Original file size vs. Encrypted file size.

Figure 4.2.1 shows the size of original file and encrypted file with two different pillars, through which relation between two file size.

Table: 4.2.2
Size and Time Comparative Table of Decryption

SL. No.	Encrypted File Size(byte)	Decrypted File Size(byte)	Decryption Time	Decryption Time /byte
1	11	10	0	0
2	77	70	0	0
3	231	210	0.164835	0.000784929
4	440	400	0.384615	0.000961538
5	671	610	0.714286	0.001170961
6	968	880	1.098901	0.001248751
7	1147	1044	1.208791	0.001157846
8	1,362	1239	1.318681	0.001064311
9	1884	1710	1.923077	0.001124606
10	2232	2028	2.197802	0.001083729

Table: 4.2.2 shows time, taken for decryption for different file size i.e. Encrypted file size and time taken for decryption for each byte and decrypted file size. From the above table data we draw the following figure.

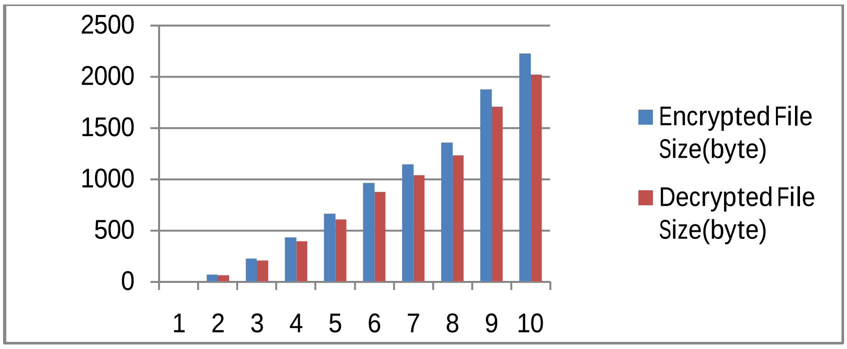


Fig: 4.2.2
Encrypted file size vs. Decrypted file size.

In the Figure 4.2.2 it is clearly seen that encrypted file size is decreased in corresponding decrypted file which is actually nothing but original file.

DOI: 10.15680/IJIRCCE.2016. 0405146

4.3 Strength of the Algorithm

If we conceder n^th^ dimension space along with 0 and 1 coordinator value for all axes, the maximum, 2^n points can be generate. In 2 dimension, there are 4 points (0,0) , (0,1), (1,0) and (1,1). With these 4 points, there are only 4 NULL vectors. There are 4 sides of the figure (Square) generated by 4 points in 2D. So numbers of side vector is 8. Similarly there are 2 diagonals as well as 4 diagonals vectors. Similarly there are 8 points in 3D. In the figure (Cube) in 3D, there are 8 NULL, 12 X 2 = 24 side and 16 X 2 = 32 diagonal vectors. So in n^th^ dimension space there are 2^n numbers of points as well as 2^n number of NULL vectors. n side vectors is being generated from each point. As there are 2^n points, number of total side vector is n X 2^n. From each point, $2^n - 1$ number of points can be connected with $(2^n - 1)$ numbers of vectors. So the total $2^n(2^n - 1)$ vectors can be generated with n number of points. Among them, n number of side vectors are present, the rest are diagonal vectors. Total numbers of diagonals vectors are $2^n(2^n - 1) - n\,2^n = 2^n(2^n - n - 1)$.

So Total number of Null vector = 2^n

Total number of Side vector = $n*2^n$

Total number of Diagonal vector = $2^n(2^n - n - 1)$

If we represent positions of all 3 kinds of vectors with minimum numbers of bits that is encrypted bit stream, we try to represent maximum positional value of the vectors with exactly particular number of bits.

Let $n = 2^K$,

Total number of Null vector = 2^{2^K}

Total number of Side vector = $2^K \text{ X } 2^{2^K}$

Total number of Diagonal vector = $2^{2^K}(2^{2^K} - 2^K - 1)$

Now all NULL and Side vectors can be represented respectively 2^K and K bits.

Here, key has 3 segments. In 1st segment, the key contains value of K when block size is 2^K. So, possible combination of the range of the block will be 0 to 2^{2^K} -1. If we increase the value of K, then block size will be increased exponentially of 2. Moreover maximum value of the blocks will be increased exponentially of the exponent of 2. So for a little change of value of K, a huge amount of difference will be reflected in the algorithm.

VI. Conclusion and Future Work

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 various type and size of file. It will be secured.

References

[1] A. Kahate,"Cryptography and Network Security", (2nd ed.). New Delhi: Tata McGraw Hill, 2008.

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

[3]India2Zirra Peter Buba, Gregory MakshaWajiga– "Cryptographic Algorithms for Secure Data Communication" ,International Journal of Computer Science and Security,
Vol. 5, Issue 2, 2011.

[4]Pranam Paul, SaurabhDutta, 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.

[5]SanjitMazumdar, SujayDasgupta, 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.

[6]SujayDasgupta, SanjitMazumdar, Prof.(Dr) Pranam Paul, "Implementation of Information Security based on Common Division", Internationaljournal of Computer Science and Network Security, Vol. 11, No.2,pp. 51-53, 2011.

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

[8]AsokeNath, SaimaGhosh, MeheboobAlamMallik, "Symmetric Key Cryptography using Random key Generator", Proceeding of Internationalconference on security and management (SAM10" held at Las Vegas, USA Jull 12-15,2010), P-Vol-2, pp. 239-244,2010.

[9] Pranam Paul, SaurabhDutta, "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 CalcuttaSection,Computer Science & Engineering Department, CMATER &SRUVMProject-JadavpurUniversity and Computer Jagat.

[10]OdedGoldreich, “Foundation of Cryptography (A primer)”, July 2004.

[11] Bruce Schneier, “Applied Cryptography”, ISBN 0-471-12845-7

[12] John Talbot, Dominic Welsh; “Complexity and Cryptography An introduction”. ISBN-10: 0521852315

[13] Denise Sutherland, Mark Koltko-Rivera “Cracking Codes and Cryptograms For Dummies”; ISBN: 978-0-470-59100-0;October 2009

[14] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone “Handbook of Applied Cryptography”; CRC Press; ISBN: 0-8493-8523-7

[15]WILLIAM F. FRIEDMAN; “MILITARY CRYPTANALYSIS, Part I, MONOALPHABETIC SUBSTITUTION SYSTEMS”

[16]Henk C.A. van Tilborg, SushilJajodia; “Encyclopedia of Cryptography and Security”, 2nd edition; 2011; ISBN: 144195905X

[17]Wenbo Mao; “Modern Cryptograph”..

[18]Wels Chenbach; “Cryptography in C and C++”.

[19]Koblitz, N.,"A Course in Number Theory and Cryptography, 2nd ed. New York: Springer-Verlag, 1994.

[20] A. Menezes, P. Van Oorschot, S. Vanstone,"Handbook of Applied Cryptography", CRC Press, 1996.

[21] Mark Adler, Jean-Loup Gailly,"An Introduction to Cryptography", released June 8, 2004. [Online] Available: http://www.pgp.com.

[22]AyanBanrjee, Prof. Dr.Pranam Paul, “Bock Based Encryption and Decryption”, International journal of Computer Science and Network Security, ISSN: 0974 – 9616 vol-7,No.2,2015.

[23]Shibaranjan Bhattacharyya, Prof. Dr.Pranam Paul, “An Approach to Block Ciphering using Root of Perfect Square Number”, International journal of Computer Science and Network Security,ISSN: 0974 – 9616 vol-7,No.2,2015.

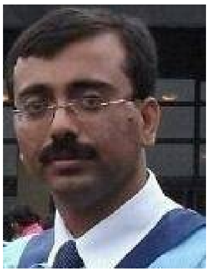
[24]Moinakchowdhury and Prof. Dr.Pranam Paul, **“**BLOCK BASED DATA ENCRYPTION AND DECRYPTION USING THE DISTENCE BETWEEN PRIME NUMBERS”

[25] AnupamMondal, Prof. Dr Pranam Paul,Implementing Cryptography on the Concept of Returning Back Its Own Nest of a Bird, IJIRCCE

[26] SukanyaChakravarty, Prof. Dr.Pranam Paul, Approach Based on Finding the Difference between Consecutive Numbers, IJIRCCE.

BIOGRAPHY

Debashis Sar, he is a student of MCA from Narula Institute of Technology, and formar student of BSc from RamakrishnaMissionVivekanandaCentenaryCollege(Rahara)under WBSU(Barasat).



Author Dr Pranam Paul, ***Assistant Professor and Departmental Head, CA Department, Narula Institute of Technology (NIT), Agarpara*** had completed MCA in 2005. Then his carrier had been started as an academician from MCKV Institute of Technology, Liluah. Parallel, At the same time, he continued his research work. At October, 2006,National Institute of Technology (NIT), Durgapur had agreed to enroll his name as a registered Ph.D. scholar.Then he had joined Bengal College of Engineering and Technology, Durgapur. After that Dr. B. C. RoyEngineering College hired him in the MCA department at 2007. At the age of 30, he had got Ph.D. from NationalInstitute of Technology, Durgapur, and West Bengal. He had submitted his Ph.D. thesis only within 2 Years and 5Months. After completing the Ph.D., he had joined Narula Institute of Technology in Computer ApplicationDepartment. Parallel he continues his research work. For that, he has 39 International Journal Publications among54 accepted papers in different areas. He also reviewer of International Journal of Network Security (IJNS), Taiwan and International Journalof Computer Science Issue (IJCSI); Republic of Mauritius**.**

DOI: 10.15680/IJIRCCE.2016. 0405146