DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4278

Volume 8, No. 7, July – August 2017

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

ENHANCE THE DATA SECURITY BY CHANGINGTHE ENCRYPTION TECHNIQUE BASED ON DATA PATTERN IN BLOCK BASED PRIVATE KEY DATA ENCRYPTION

Sanjit Mazumder
Assistant Professor (CSE),
Modern Institute of Engineering & Technology, Bandel, WestBengal.

Neha Kumari Shaw, Bidisha Dey, and Farhin Mahmuda Laskar
CSE 4th Year student of Modern Institute of Engineering & Technology, Bandel, West Bengal.

Abstract: Cryptography is the process of data hiding from unauthorized users. Historically, the term “Cryptography” has been associated with the problem of designing and analyzing encryptionschemes (i.e., scheme that provide secret communication over insecure communication media). With the growth of internet secure data transmission is more and more essential and important. Sometime single encryption technique is not sufficient to protect the valuable data. We developed a new algorithm which can change the encryptiontechnique with change of plane textpattern to enhance the confusion and as well as diffusion.

Keyword: Encryption, Decryption, Cipher Text, Private Key Data Encryption

1. INTRODUCTION

With the increase of internet users, the secure data transmission through internet is more and more essential and important. There are many private key[1]data encryption[1] algorithm are available. Some of them are time consuming and some of them are power consuming. We develop a new algorithm which can change the encryption[1] process with change of data pattern to improve the data security.
In section 2, Algorithm is defined. While section 3 shows the example of whole process, section 4 is result and section 5 is discuss the analysis and conclusion.

2. ALGORITHM

Here we use symmetric key[2]block based encryption technique. We choose block cipher approach as it is more secure and convenient to use. In our program we use a single key(n x n matrix), as private key, to encrypt and decrypt data. In this algorithm the plain text is converted into cipher text[2] or encrypted text using the private key[2] and this cipher text is decrypted into plain text using the same key. The key should be shared by both sender and receiver using a secure channel. Basicin this section we discuss the key generation process in section 2.1, concept is shown in fig:1 and fig:2

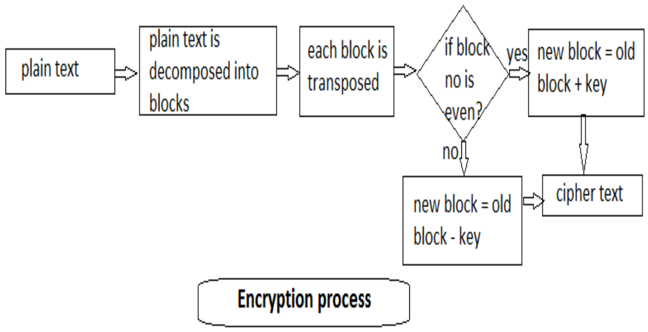

Fig 1

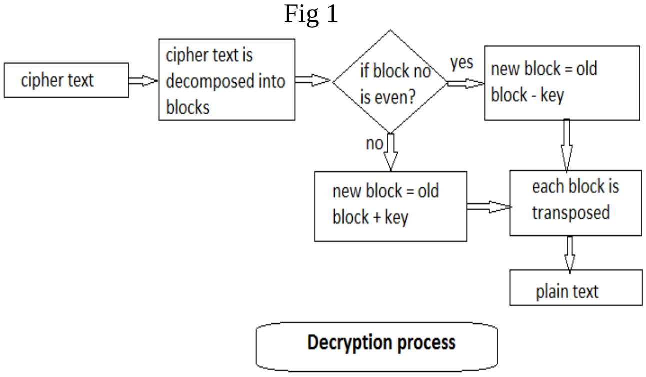

Fig 2

2.1Key Generation Process

Step 1: Here we generated a randomly generated m*m matrix as a key.

Step 2: The key should contain information about the length of UB (unchanged bits) which will be append with the sub key to generate the original key.

2.2 Encryption Process[2]

Step 1: DecomposetheL bits plain text into n no of blocks(they are B_1, B_2, ..., B_n) so that each blocks will contain m x m no of bits. After splitting, no of blocks will be [n/(m*m)].Let's say it is x.
Step 2: After splitting the text the remaining bits will be [n – (n/(m*m))] = UB. Those bits will append with the cipher text.
Step 3: then each block are transposed. After transpose the blocks are B^T_1, B^T_2,....,B^T_n.

Step 4: Each block are checked as

If ([n/m*m] = = 0)
{
Loop: 1 to x
$C_i = B^T_i + Key$; [where B^T_irepresent the blocks and i = 1 to x]
`}
else
{
Loop: 1 to x
$C_i = B^T_i – Key$; [where B^T_i represent the blocks and i = 1 to x]
}

Step 5: C_i UB is appended to produce the encrypted text. [where i = 1 to x]
Step 8: Exit.

2.3. Decryption Process[2]

Step 1: We take the previous encrypted text and decompose it into n no of blocks so that each block contains m*m no of bits. After splitting the no of blocks will be [n / m*m] . Lets say it is x.
Step 2: After splitting the remaining bits will be [n –(n / m*m)]. This will be treated as unchanged blocks (UB).
Step 4: Check
If([n / m*m] = = 0)
{
Loop : 1 to x. [where x is the no of blocks]
$C_i = B_i – Key$ [where Bi represent the blocks and i = 1 to x.]
}
else
{
Loop : 1 to x. [where x is no of blocks]
$C_i = B_i + Key$ [where Bi represent the blocks and i = 1 to x.]
}
Step 5: Transpose each blocks(say C^T_1,C^T_2,.....,C^T_n).
Step 6: Append C^T_i UB to produce the decrypted text. [where i = 1 to x]
Step 7: Exit.

3. EXAMPLE

3.1 Key generation: Randomly generate an 3 x 3 matrix whichis

3	3	3
3	3	3
3	3	3

3.2 Encryption[3]: Let assume the plane text is: "It is an ex of encryption".
3.2.1 Decompose the plain text into no of blocks, where the block size is same as the key size, which are
B_1 =

I	t	*
i	s	*
a	n	*

and

B_2 =

e	x	*
o	f	*
e	n	c

[where * represent the blank space]
After decomposing, the remaining bits are – ryption (unchanged bits)
3.2.2 Each blocks now transposed
B^T_1 =

I	i	a
t	s	n
*	*	*

and

B^T_2 =

e	o	e
x	f	n
*	*	c

3.2.3 if([n / m*m]==0) then $C_1 = B^T_1 + key$
Else $C_1 = B^T_1 – key$
So, $B^T_1 – key = C_1$ =

F	f	^
q	p	k
*	*	*

and

$B^T_2 + key = C_2$ =

h	r	h
{	i	q
#	#	f

So, encrypted text will be Ff^qpkhrh{iq##f

3.3 Decryption[3]
3.3.1 Decompose the cipher text[3] into blocks

B_1 =

F	f	^
q	p	k
*	*	*

and

B_2 =

h	r	h
{	i	q
#	#	f

3.3.2 check if([n / m*m]==0) then $C_i = B_i$ – Key
Else $C_i = B_i$ + Key

So, B_1 + Key = C_1 =

I	i	a
t	s	n
*	*	*

and

B_2 – Key = C_2 =

e	o	e
x	f	n
*	*	c

3.3.3 Each block then transposed.

C^T_1 =

I	t	*
i	s	*
a	n	*

and

C^T_2 =

E	x	*
O	f	*
E	n	c

So, the decrypted text[3] is – It is an ex of enc.

3.3.4 Unchanged bits are now appended with the encrypted text to generate the encrypted text – It is an ex of encryption.

Key structure:

Segment	Decryption	Maximum no of bits required(size)
Segment-1	Main part of the key	m*m
Segment-2	Un changed block	[L%(m*m)]
Segment-3	Data information	1bit

Figure Details:

Sl. No.	Figure No.	Description
1	1	Block diagram of encryptionprocess[3].
2	2	Block diagram of decryption process[3].
3	3	Comparison between standard algorithm decryption time and new algorithm decryption time.
4	4	Comparison between standard algorithm encryption time and new algorithm encryption time.
5	5	Comparison between encryption time and decryption time with file size of new algorithm.

4 RESULT

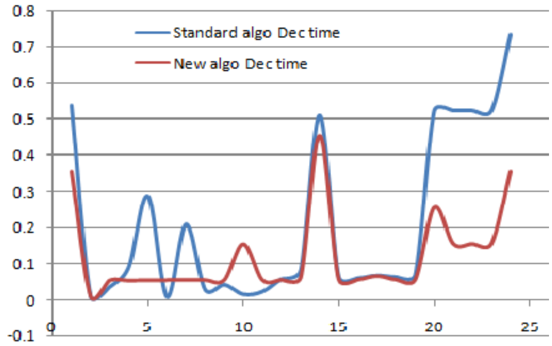

Fig- 3

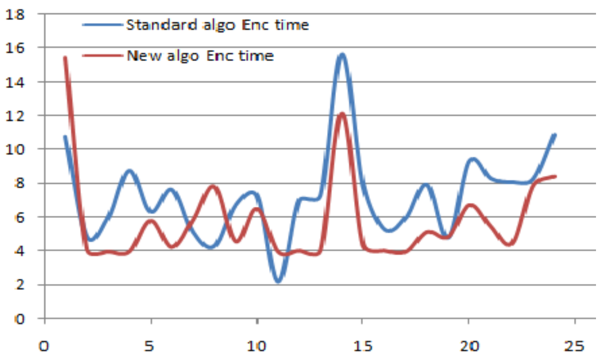

Fig- 4

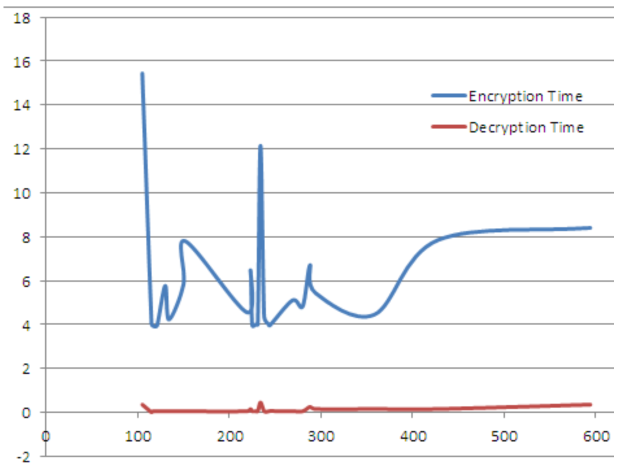

Fig- 5

Table 1

File Size (Byte)	Encryption Time (RSA)	Encryption Time (Our Algorithm)

105	10.764501	15.43956
115	4.876532	4.065934
116	5.984565	3.956044
121	8.764543	3.956044
130	6.345298	5.769231
134	7.639087	4.230769
150	5.109312	5.846154
152	4.298222	7.802198
219	6.729874	4.56044
223	7.297845	6.483516
225	2.198376	3.956044
229	7.029801	4.010989
231	7.280012	4.010989
234	15.638902	12.142857
238	7.892892	4.340659
243	5.298732	4.010989
244	5.982302	3.956044
269	7.92831	5.10989
280	4.801223	4.835165
288	9.30928	6.703297
294	8.29845	5.43956
358	8.092287	4.450549
428	8.27892	7.857143
594	10.876533	8.406593

Table 2

File Size (Byte)	Decryption Time (RSA)	Decryption Time (Our Algorithm)
105	0.537822	0.354333
115	0.010438	0.008965
116	0.036542	0.052784
121	0.091837	0.052267
130	0.290187	0.053345
134	0.009382	0.053696
150	0.209901	0.053699
152	0.029101	0.053742
219	0.042611	0.053781
223	0.016389	0.153856
225	0.023541	0.054231
229	0.058191	0.054256
231	0.082333	0.060533
234	0.510578	0.454689
238	0.063201	0.054767
243	0.061013	0.054863
244	0.068299	0.064879
269	0.064302	0.054902
280	0.069222	0.054943
288	0.523781	0.254863
294	0.523711	0.154945
358	0.523711	0.154745
428	0.523202	0.154945
594	0.732901	0.354645

Table 3

File Size (Byte)	Encryption Time (Our Algorithm)	Decryption Time (Our Algorithm)
105	15.43956	0.354333
115	4.065934	0.008965
116	3.956044	0.052784
121	3.956044	0.052267
130	5.769231	0.053345
134	4.230769	0.053696
150	5.846154	0.053699
152	7.802198	0.053742
219	4.56044	0.053781
223	6.483516	0.153856
225	3.956044	0.054231
229	4.010989	0.054256
231	4.010989	0.060533
234	12.142857	0.454689
238	4.340659	0.054767
243	4.010989	0.054863
244	3.956044	0.064879
269	5.10989	0.054902
280	4.835165	0.054943
288	6.703297	0.254863
294	5.43956	0.154945
358	4.450549	0.154745
428	7.857143	0.154945
594	8.406593	0.354645

5 ANALYSIS AND CONCLUSION

In our encryption process[5] the following advantage are provided: the 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[5].

As the key length is not fixed in this algorithm, we can take large key length for making it more complex. If the key length is assumed as m*mmatrix the complexity of guessing is 2^{m*m}. Hencethe complexity of the key is increase exponentially with respect to the increase of key length. In this algorithm the length of the plane text is not restricted so it can be applicable for any large file.

6 REFERENCES

[1] J. K. Mandal, S. Dutta, “A 256-bit recursive pair parityencoder for encryption”, Advances D -2004, Vol. 9 nº1,Association for the Advancement of Modelling andSimulation Techniques in Enterprises (AMSE, France), www.AMSE-Modeling.org, pp. 1-14

[2] Pranam Paul, SaurabhDutta, “A Private-Key Storage-Efficient Ciphering Protocol for Information CommunicationTechnology”, National Seminar on Research Issues inTechnical Education (RITE), March 08-09, 2006, NationalInstitute of Technical Teachers’ Training and Research,Kolkata, India

[3] Pranam Paul, SaurabhDutta, “An Enhancement ofInformation Security Using Substitution of Bits ThroughPrime Detection in Blocks”, Proceedings of NationalConference on Recent Trends in Information Systems(ReTIS-06), July 14-15, 2006, Organized by IEEE GoldAffinity Group, IEEE Calcutta Section, Computer Science &Engineering Department, CMATER & SRUVM Project-Jadavpur University and Computer Jagat

[4] Dutta S. and Mandal J. K., “A Space-Efficient UniversalEncoder for Secured Transmission”, InternationalConference on Modelling and Simulation (MS’ 2000 –Egypt, Cairo, April 11-14, 2000

[5] Mandal J. K., Mal S., Dutta S., A 256 Bit Recursive PairParity Encoder for Encryption, accepted for publication inAMSE Journal, France, 2003

[6] Dutta S., Mal S., “A Multiplexing Triangular EncryptionTechnique – A move towards enhancing security inECommerce”, Proceedings of IT Conference (organized byComputer Association of Nepal), 26 and 27 January, 2002,BICC, Kathmandu.

SanjitMazumder is an assistant professor of CSE department of Modern Institute of Engineering &Technology, Bandel, WestBengal. He has total six journals publication in different international journals. His field of interest is Cryptography, Network security and Automata & Formal Language.

BidishaDeyis 4th yearstudent ofCSE Department of Modern Institute of Engineering &Technology(Bandel). She has a great interest in networks security, software development, web application.

Neha Kumari Shaw,student of 4th year Computer Science & Engineering of Modern Institute of Engineering &Technology (Bandel).Highly interested in working with network security and computer networks, java programming, android application.

FarhinMahmudaLaskar,student of 4th year Computer Science & Engineering of Modern Institute of Engineering &Technology (Bandel). Interested in networking and security, web application.

