Implementation of Information Security based on Common Division

Sujoy Dasgupta [1], Sanjit Mazumder [1], Prof. (Dr) Pranam Paul [2]
[1] Student, M. Tech. (CSE), Narula Institute of Technology, Agarpara, West Bengal, INDIA
[2] Computer Application, Narula Institute of Technology, Agarpara, West Bengal, INDIA

ABSTRACT
In this paper we are developed a new encryption technique. 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 substitution based block cipher encryption technique. In this algorithm encryption is done on binary file so it can be applicable for any type of file.

Key Word:
Cryptography, Encryption, Decryption, Plain Text, Cipher Text, Network Security

1. 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 use to ensure security of those data. I introduce a block based symmetric key encryption algorithm. For encryption a key is to generate. Key length and bit stream is chosen at random.
At first, the plain text has been decomposed into some blocks. Maximum common divisor of 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 circular left shifting, some effective keys are generated for XOR-ing with quotient of division between taken consecutive block and the calculated common divisor of them.
To get back the original text the operation is called the decryption. In this process the encrypted text first divided into segments. Getting desirable information and generated effective keys from key reverse process of encryption has been applied on each segment
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. 2 ALGORITHM:

In this section, encryption process is discussed in section 2.1. In the section 2.2 and 2.3 discussed about the structure of key and decryption respectively.

2.1 Encryption Process:

Binary stream s is decomposed into l number of n bit blocks. Last n1 bit block, say ub ,(when 0<=n1<n) should be an unchanged block which does not involve into any encryption process.
Step 1: p number of consecutive n bit blocks are taken such that l is divisible by p, say b1,b2, bp.
Step 2: Calculate k bit maximum (1<=k<=n) Common divisor, say d, of those p bit block and integer part of n/p, say r.
Step 3: Randomly taken n bit temporary key.
Step 4: Every r bit circular left shift of temporary key, p number of effective keys, say e1,e2,……..,ep are generated.
Step 5: XOR operation has been done between ei and quotient of bi /d when i=1 to p and append then consecutively.
Step 6: Appending the result of Step 5 with k bit representation of d, is considered as part of encrypted text.
Step 7: Continue the process from Step 1 to Step 6 on next p number of consecutive blocks and append the recently generated encrypted text with previous one, until all l number of blocks have been involved.
Step 8: Lastly concatenating the result of Step 7 with unchanged block, ub , encrypted text is generated.

Manuscript received February 5, 2011
Manuscript revised February 20, 2011

2.2 Structure of Key

Structure of Key

Segment	*Description*	*Maximum number of bit required (size)*
1	To present block size n	m
2	To represent k bit	m1
3	To represent p bit	q
4	To represent n1 bit	m
5	Temporary Key	n
Total Key Size		***2m+q+n+m1***

2.3 Decryption Process:

From the key we can get the size of each block and size of temporary key n and number of consecutive blocks used at a time p and number of bits to represent the maximum common divisor k and length of unchanged block n1.

Step1: By n and p, we can get the number of bits circularly left shifted, say r which produces effective keys e1,e2,……ep from temporary key.

Step 2: First n1 number of bits from the encrypted text is stored in ub as unchanged block.

Step 3: Next k – bit produces common divisor, say d of next p number of consecutive n – bit blocks, say d1,d2,…..,dp.

Step 4: XOR operation has been done between ei and di and blocki is generated, where i=1 to p.

Step 5: Multiply each of the p number of n bit blocks by d, like (block1*d), (block2*d),……. (blockp*d) and arrange the results of multiplication consecutively as blocks occur in the encrypted text. In this way decrypted text for first p number of n – bit blocks is generated.

Step 6: Continue the process from Step 3 to Step 5 on next p number of consecutive blocks and append the next part of decrypted text with the previous one until all blocks are involved in the process.

Step 7: Lastly concatenate the content of ub at the end of decrypted text and in this way the original plain text can be recovered.

3. Example:

To illustrate the algorithm, an example has been shown. The algorithm can be operated on a binary stream s act as plain text for an example. S is given below:-

01101101110111000100101110001101 0011

Let us start with the encryption process first. In this process, we divide this stream into four 8 bit blocks. so we can say n=8,l=4.In this example we are taking two consecutive blocks, say b1,b2,p=2. Last 4 bit block remains unchanged block which doesn't involve into any encryption process. so n1=4. Now let us calculate the number of bit to represent maximum common divisor d, is k. In this example k=2.

Now we can think about key generation. In this example we take the integer part of n/p, say r. so r=4.Let us generate a random number of length=8. Let temporary key is 11011011.After 4 bit circular left shifting we get two keys which are effective keys e1,e2 .

Therefore e1 =10111101, e2 =11011011.After getting keys we perform division operations between bp and d and this results are XORed with effective keys ep .

b1/d=01101101/01=01101101,

b2/d=11011100/01=11011100

(b1/d) XOR e1 = 11010000

(b2/d) XOR e2= 00000111

Now we have to append the results of XOR operation and that will be the cipher text of first two block of plain text. It will be look like as given:- 011101000000000111

For next two consecutive blocks we perform (bp/d) operation and XOR ep with results in the same way and get following binary stream:-

111010010011001101

Last 4 bits will be unchanged block and it have to append at the front of two concatenated binary stream. This is the final encrypted text of whole plain text as given below:-

0011011101000000000111111010010011001101

Now we have to start the decryption process. From the key structure we can get size of each block n=8, number of consecutive blocks used at a time p=2, number of bits to represent maximum common divisor (d) k=2, length of unchanged block (ub) n1=4 and temporary key.

By n and p we can get the number of bits circularly left shifted, r=2, the temporary key to get the effective keys ep.

If random number=11011011, then effective keys e1=10111101,e2=11011011.

Now we take the first 4 bits from cipher text and store it in ub. So content of ub=0011.Then by n and k we can get the value of d for next p number of n bit blocks and consider (n*p) number of bits from the rest and differ two 8 bit blocks. The value of d for first two 8 bit blocks is 01 and the blocks are 11010000, 00000111.

Then we XOR the keys e1,e2 with two 8 bit blocks consecutively. The XOR operation and results are shown below:-

(11010000)XOR(10111101)= 01101101

(00000111)XOR(11011011)= 11011100

The results of two XOR operations are individually multiplied by value of d. This multiplication can be shown below:- 01101101 *01=01101101

11011100*01=11011100

The resultant blocks are first two 8 bit blocks b1, b2 of original binary stream. So b1=01101101, b2=11011100. It will be:- 0110110111011100
Through this process we get a part of decrypted text. Following the above process for rest of the cipher text block, say 111010010011001101, we first get value of d, say 11, for two 8 bit blocks and then XOR the keys with two blocks and then multiply the results by d and get rest two blocks of the original binary stream, say b3=01001011 and b4=10001101.This will be:- 0100101110001101
Then we append the new results (b3,b4) with the old one (b1,b2)content of unchanged block (ub) at the end. In this way we can recover the whole plain text which is as follows:-
01101101110111000100101110001101001 1

4. ANALYSIS AND CONCLUSION

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 or sound encryption process.
In this algorithm the length of the plain text is not restricted so it can be applicable for any large file.
We can shuffle the segments after EXOR ing which produce more difficulties for unauthorized access.
Since the random number is shifted randomly so the key is more complex for attacker to compute.
Section 2.2 shows the a formula for key, through which key size can be known. If the block size (n), number of consecutive blocks, considered at a time during encryption (p) and number of taken bits for calculating the common divisor (k) are changed, then the key size will be changed. During every encryption we can choose different size of key by considering the different values of above mentioned variables.

REFERENCES

[1] J. K. Mandal, S. Dutta, “A 256-bit recursive pair parity encoder for encryption”, Advances D -2004, Vol. 9 nº1, Association for the Advancement of Modelling and Simulation Techniques in Enterprises (AMSE, France), www. AMSE-Modeling.org, pp. 1-14
[2] Pranam Paul, Saurabh Dutta, “A Private-Key Storage-Efficient Ciphering Protocol for Information Communication Technology”, National Seminar on Research Issues in Technical Education (RITE), March 08-09, 2006, National Institute of Technical Teachers’ Training and Research, Kolkata, India
[3] Pranam Paul, Saurabh Dutta, “An Enhancement of Information Security Using Substitution of Bits Through Prime Detection in Blocks”, Proceedings of National Conference on Recent Trends in Information Systems (ReTIS-06), July 14-15, 2006, Organized by IEEE Gold Affinity 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 Universal Encoder for Secured Transmission”, International Conference on Modelling and Simulation (MS’ 2000 – Egypt, Cairo, April 11-14, 2000
[5] Mandal J. K., Mal S., Dutta S., A 256 Bit Recursive Pair Parity Encoder for Encryption, accepted for publication in AMSE Journal, France, 2003
[6] Dutta S., Mal S., “A Multiplexing Triangular Encryption Technique – A move towards enhancing security in ECommerce”, Proceedings of IT Conference (organized by Computer Association of Nepal), 26 and 27 January, 2002, BICC, Kathmandu.

Sujoy Dasgupta is MTech (CSE) final year student of Narula Institute of Technology, Agarpara. He had complete his BTech in Information Technology.

Sanjit Mazumder is MTech (CSE) final semester student of Narula Institute of Technology, Agarpara. He had completed his MSc. in Information Technology. He has one international journal publication.

Dr Pranam Paul is an Assistance Professor cturer of Narula Institute Technology, Agarpara. He had completed his Ph.D from Electronic and Communication Engineering department of National Institute of Technology, Durgapur in the field of Cryptography and Network Security and master degree in Computer Application in 2005 under West Bengal University of Technology, INDIA. He has total 19 International Journal publications among total 32 publications, except this one.