A Novel Approach for Block Based Data Encryption Using LSB and MSB

[1]V.Lakshma Reddy, [2]T.Venkata Satya Vivek

[1]Department of Computer Science & Engineering, PACE Institute of Technology & Sciences, Ongole, India.
[2]Department of Computer Science & Engineering, Grandhi Varalakshmi Venkatarao Institute of Technology, Bhimavaram, India.

Abstract:- With the development of information correspondence over the web, the requirement for security has gotten most extreme significance. Information should be kept avoided all the web clients separated from the approved ones. Information ought to be encoded before sending it through the web. Here, a calculation taking into account Block-Based information encryption is presented, where the encryption and unscrambling procedure is done on paired information, so it will be appropriate to all information in the field of software engineering.

INTRODUCTION:-

The current configuration of block ciphers depends on the idea of an iterated product cipher. Product ciphers were proposed and investigated by Claude Shannon in his seminal 1949 publication Communication Theory of Secrecy Systems as a way to adequately enhance security by joining basic operations, for example, substitutions and permutations.[1] Iterated product ciphers carry out complete encryption in various rounds, each of which uses an alternate sub key got from the first key. One far reaching execution of such ciphers is known as a Feistel system, named after Horst Feistel, and outstandingly actualized in the DES cipher.[2] Many different acknowledge of block ciphers, for example, the AES, are named substitution-change networks[3]. The publication of the DES cipher by the U.S. National Bureau of Standards (now National Institute of Standards and Technology, NIST) in 1977 was key in general society comprehension of cutting edge block cipher outline. In the same way, it affected the scholarly advancement of cryptanalytic assaults. Both differential and straight cryptanalysis emerged out of studies on the DES outline. Today, there is a palette of assault strategies against which a block cipher must be secure, in addition to being hearty against brute force attack. Indeed, even a guarded block cipher is suitable just for the encryption of an individual block under an established key. A huge number of methods of operation have been intended to permit their rehashed use secury, usually to accomplish the security objectives of privacy and credibility. In any case, block ciphers might likewise be utilized as building blocks as a part of other cryptographic conventions, for example, all inclusive hash capacities and pseudo-irregular number generators.

With the fast development of information correspondence, the need to safely exchange information starting with one PC then onto the next has increased most extreme noteworthiness.

Here I attempt and present an encryption and unscrambling calculation in view of Block-based information encryption method. The procedure of encryption is started by producing a key. Key length and information length can be arbitrary. We then create optional key A from the key. Structure auxiliary key A, we produce optional key B, C and D. The plain content is then separated into four sections of equivalent lengths. We attempt and segmentize the torment content into lengths that is equivalent to that of the optional keys, while the remaining bits staying unaltered. Every sub blocks are then XNORed with the optional keys to create the cipher text. The decoding procedure is the polar opposite of the encryption process. We partition the figure content into four portions. Every section id then XNORed with the auxiliary keys to get the plain content.

Proposed Algorithm:-

A. Key Generation Algorithm: In the novel approach of this algorithm we are going to generate four keys.

1. Consider a Random Primary Key ‘K’; and Reverse this key, and name it as RSK1.
2. The secondary key RSK2 is generated by taking the LSB as it is and by XORing the i^{th} term with the $(i+1)^{th}$ term.
3. To generate the key 3 i.e RSK3, perform the ‘OR’ in the i^{th} term of SK1 with the ith term of SK2 from the Most Significant Bit (MSB) Side.
4. To generate the key 4 i.e RSK4, perform the ‘EXOR’ from the LSB Bit Side.

B. Encryption Algorithm

1. Let us consider the Plain Text in Binary form.

2. Divide the Plain Text into Two (2) Equal Parts as Left Side(L) and Right Side(R).
3. Reverse both the sides individually, and after that combine the sides.
4. Divide the combined list into Four (4) Equal Parts, and name the parts as PT1,PT2,PT3,PT4.
5. From PT1 to PT4, Firstly remove the First Two bits from the Most Significant Bit Side, after from Least Significant Bit Side as Vice-Versa.
6. Now apply the XNOR Function from Least Significant Bit Side between PT1 and RSK1.The result of this is named as Cipher Text1(CT1).
7. Now apply the XNOR Function from Most Significant Bit Side between PT2 and RSK2.The result of this is named as Cipher Text2 (CT2).
8. Now apply the XNOR Function from Least Significant Bit Side between PT3 and RSK3.The result of this is named as Cipher Text3 (CT3).
9. Now apply the XNOR Function from Most Significant Bit Side between PT4 and RSK4.The result of this is named as Cipher Text4 (CT4).
10. Combine all the Cipher Text parts(CT1,CT2,CT3.CT4) to form a single text.

C. Decryption Algorithm:- The decryption process is the Reverse of the Encryption Algorithm.

Implementation:- The above algorithm is implemented on the below example:

The Plain Text is : 011101010110111101110011.

Now, Divide the Plain Text into Two(2) Equal Halfs, as L0 and R0.

L0=011101010110, R0=111101110011.

Take a Random Primary Key 'K' as 1011, Reversing the Key we get: 1101. So, RSK1=1101.

The secondary key RSK2 is generated by taking the LSB as it is and by XORing the i^{th} term with the $(i+1)^{th}$ term.

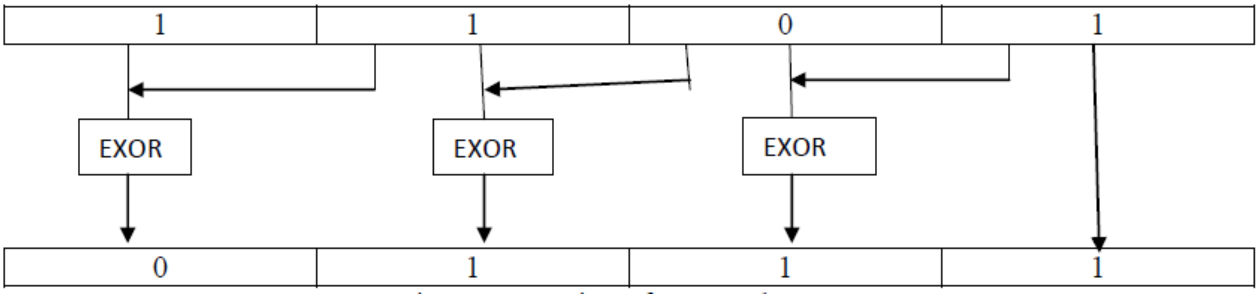

Fig 1. Generation of SK2 and RSK2.

From the above Figure: SK2=0111 , Reverse of SK2 i.e RSK2 is 1110.

The Next Step is to generate the Key3, i.e SK3 and RSK3 by performing the "OR" Operation with the ith term of SK1 and with the ith term of SK2 from the Most Significant Bit side.

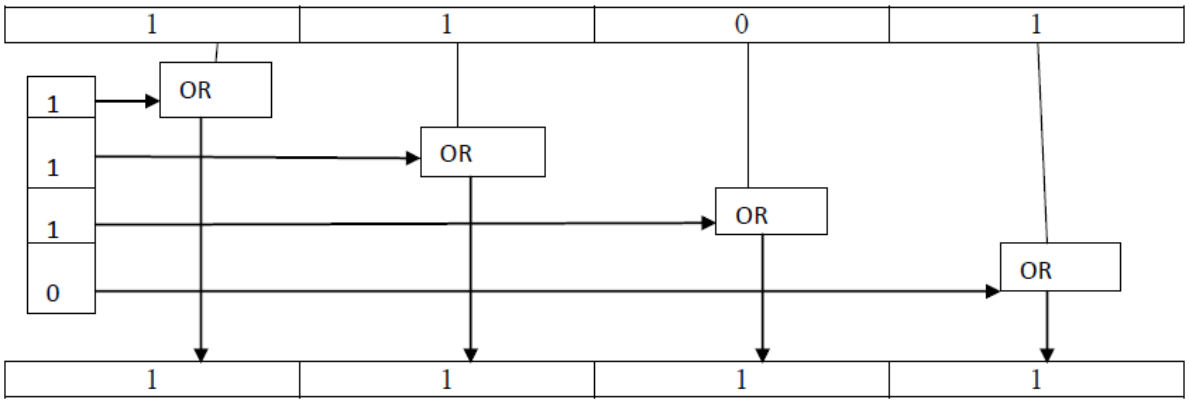

Fig 2.: Generation of SK3 and RSK3.

SK3=1111

Reverse SK3 i.e RSK3= 1111

By taking the above figure into consideration we need to generate the Key-4 i.e SK4 by applying EXOR from the Least Significant Bit Side.

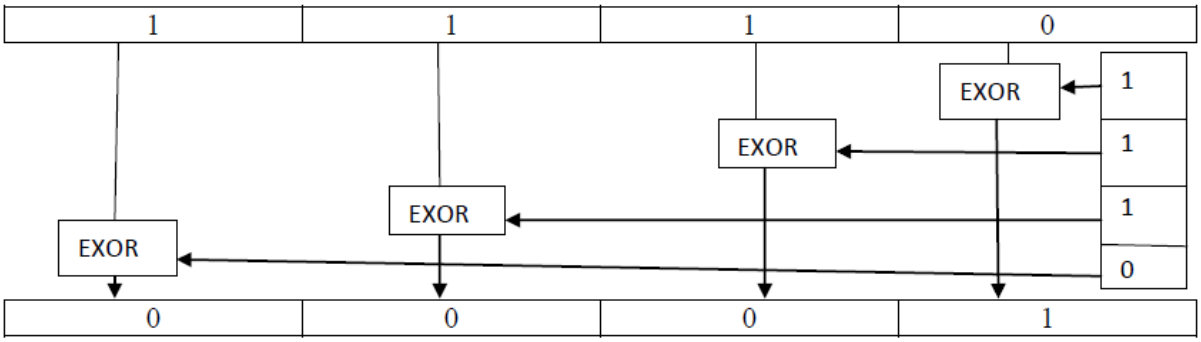

Fig 3.Generation of SK4 and RSK4.

Encryption Process:-

Let us consider the plain text in the binary format. Divide the plain text into Two(2) Equal Parts as 'L' and 'R'. Reverse 'L' and 'R' individually and combine the whole list.

Now, divide the Combined Plain text into Four(4) Equal Parts, and name them as PT1 to PT4; as shown below:

So, PT1=011010 , PT2=101110 , PT3=110011 , PT4=101111.

Now Reduce the Bits size from 6bits to 4bits, by taking and deleting the MSB and LSB bits vice versa, as shown below:

PT1= 1 0 1 0

PT2= 1 0 1 1

PT3= 0 0 1 1

PT4= 1 0 1 1

Now we need to apply the XNOR Function between PT1 to PT4 and RSK1 to RSK4.

Firstly the Least Significant Bit (LSB) is applied on the first step followed by the Most Significant Bit (MSB) as Vice versa. The below are the diagrams which are applied to get the Final Cipher Text.

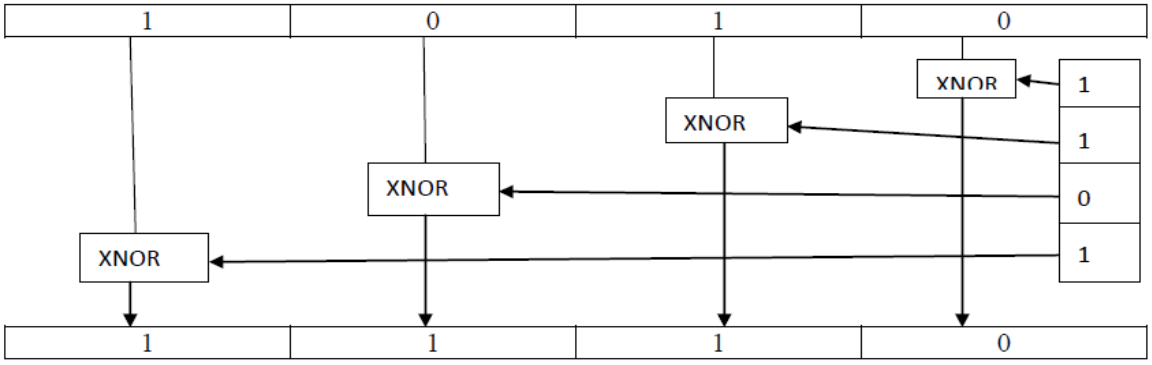

Fig 4. Generation of Cipher Text1 from PT1 and RSK1

In the next Succedding step of Cipher Text we are going to generate Cipher Text2 from PT2 with the help of RSK2. We are taking the RSK2 from the Most Signifcant Bit Side, as specified in the proposed algorithm.

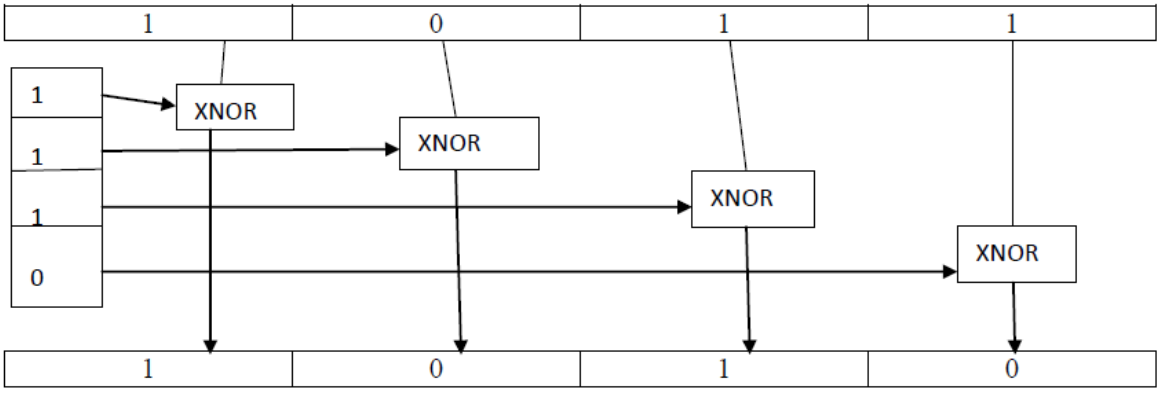

Fig 5. Generation of Cipher Text2 from PT2 and RSK2

In the next Succedding step of Cipher Text we are going to generate Cipher Text3 from PT3 with the help of RSK3. We are taking the RSK3 from the Least Significant Bit Side, as specified in the proposed algorithm.

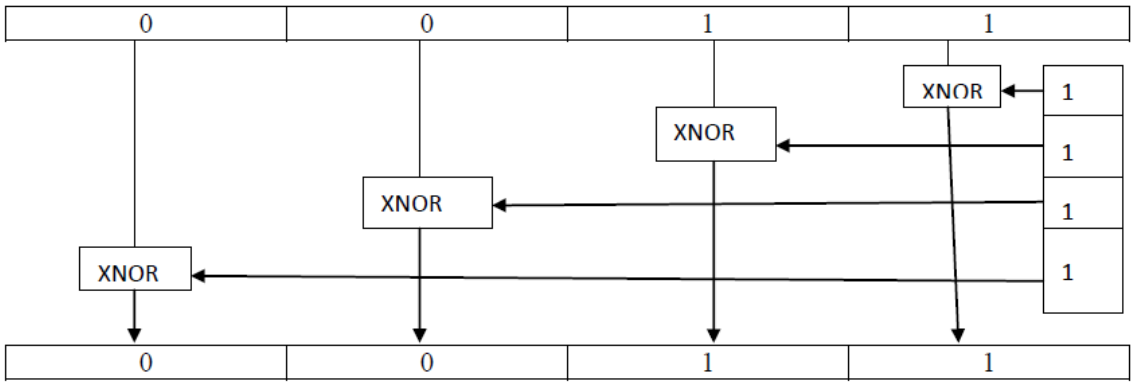

Fig 6. Generation of Cipher Text3 from PT3 and RSK3

In the last step of generating the Cipher Text, we are going to consider the Most Significant Bit side to encrypt the binary data. The below is the diagram showing the Generation of Cipher Text4.

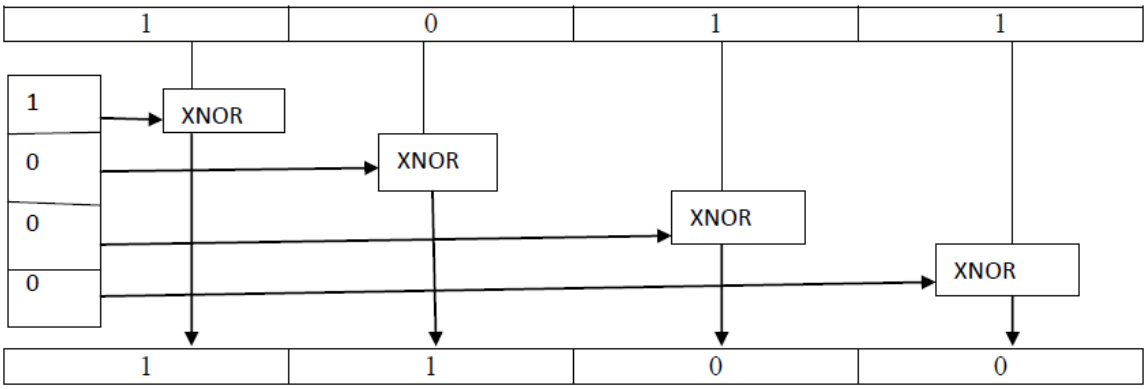

Fig 7. Generation of Cipher Text4 from PT4 and RSK4

The Cipher Text for the above proposed algorithm is produced as shown below:

CT1=1110; CT2=1010; CT3=0011; CT4=1100.

Now, Combine the above Cipher Text into Single Binary Data with the left over bits also placing in the respective format.

The Final Cipher Text is : 011110101010110011110011.

Algorithm Analysis:- In this algorithm the encryption is performed on the binary information. All information which is under stable by the PC is at long last changed over into binary bits. So it can be executed for any information encryption process. For example, text encryption, picture encryption or sound encryption process. Not just that, the key length is not fixed in this algorithm, so we can take expansive key for making it more unpredictable. On the off chance that the key length is expect "n" then 2n-2 numbers of combination can be conceivable. So if one bit is expanded the conceivable blend will be 2n-1 i.e. the unpredictability expanded exponentially. For this situation if one bit is expansion the many-sided quality is expansion in exponentially. In this algorithm the length of the plain content is not confined so it can be relevant for any vast document. We can rearrange the fragments after EXOR ing which deliver more challenges for unapproved access.

CONCLUSION:-

The algorithm has been executed and composed on binary information. Any information, be it picture, sound or content, that is perceived by the PC can be changed over to binary information. So this algorithm applies to a wide range of information. As said before, that the key length and the information length is arbitrary. So we can take substantial documents and scramble them. In this Proposed Technique we are encrypting the data with the help of taking MSB and LSB into the consideration. It is very difficult for the attacker to decrypt the data; until the proposed technique is known.

REFERENCES:-

[1] Shannon, Claude (1949). "Communication Theory of Secrecy Systems" (PDF). Bell System Technical Journal 28 (4): 656–715.

[2] Van Tilborg, Henk C. A.; Jajodia, Sushil, eds. (2011). Encyclopedia of Cryptography and Security. Springer. ISBN 978-1-4419-5905-8., p.455

[3] R. Anderson and E. Biham, "Two practical and provably secure block ciphers: BEAR and LION," in Fast Software Encryption, Third Int. Workshop Proc. Berlin, Germany: Springer-Verlag, 1996, pp. 113–120.

[4] X. Lai and J. L. Massey, "A proposal for a new block encryption standard," in Advances in Cryptology—EUROCRYPT'90. Berlin: Springer-Verlag, 1991, pp. 389–404.

[5] J. L. Massey, "SAFER K-64: A byte oriented block-ciphering algorithm," in Fast Software Encryption, R. Anderson, Ed. Berlin, Germany: Springer, 1993, (LNCS 809), pp. 1–17.

[6] R. Impagliazzo, L. Levin, and M. Luby, "Pseudo-random number generation from one-way functions," in Proc. 21st Annu. Symp. Theory Computing, 1989, pp. 12–24.

[7] M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions," SIAM J. Comput., vol. 17, pp. 373–386, 1988.

[8] R. Impagliazzo and M. Luby, "One-way functions are essential for complexity-based cryptography," in Proc. 30th Annu. Symp. Foundations Computer Science, 1989, pp. 230–235.

[9] A. Yao, "Theory and applications of trapdoor functions," in IEEE 23rd Symp. Foundations Computer Science, 1982, pp. 80–91.

[10] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," SIAM J. Comput., vol. 15, pp. 364–383, 1986.

[11] J. C. Largaris, "Pseudo-random numbers," in Probability and Algorithms. Washington, DC: National Academy, 1992, pp. 65–85.

[12] M. Matsui, "Linear cryptanalysis method for DES ciphers," in Advances in Cryptology—EUROCRYPT'93. Berlin, Germany: Springer-Verlag, 1994, pp. 386–397.

[13] X. Lai, "Higher order derivations and differential cryptanalysis," in Communication and Cryptography: Two Sides of One Tapestry. Norwell, MA: Kluwer, 1994, pp. 227–233.

[14] B. Kaliski, Jr. and M. Robshaw, "Linear cryptoanalysis using multiple approximations," in Advances in Cryptology—CRYPTO '94. Berlin, Germany: Springer-Verlag, 1994, pp. 26–39.

[15] L. Knudsen and M. Robshaw, "Non-linear approximations in linear cryptanalysis," in Advances in Cryptology—EUROCRYPT '96. Berlin, Germany: Springer-Verlag, 1996, pp. 224–236.