# Development of a Novel Algorithm to Provide Efficiency and Security

Shriya Pandey
Researcher
LNCT Bhopal, M.P
India

Anshul Sarawagi
Associate Proffessor
LNCT, Bhopal M.P
India

## ABSTRACT

To achieve high security, steganography process is combined with cryptographic encryption decryption algorithm such that if any one detect the presence of secret information still it is not understandable. Also there are many encryption/ decryption and steganography algorithm but there is always a competition to develop such an algorithm which is not only fast but also secure too. The proposed architecture first encrypts the secret text with proposed encryption algorithm and then hides it behind another text file through steganography algorithm**.**

## General Terms

Security, Algorithm , Efficiency, Hiding.

## Keywords

Computer Security, Steganography, RJDA Algorithm, Encryption Decryption Algorithm, Network , Cryptography, Symmetric Key.

## 1. INTRODUCTION

Cryptography is the study of shuffling information from in such a way that no one can understand the original meaning of message without knowing the secret key which make it again original text. Cryptography encrypts the original message that is being sent. Steganography is the form of convert communication in which a secret message is hided with a carrier data.

Encryption encodes the data so that an unintended recipient cannot determine its intended meaning. Steganography in contrast attempts to prevent an unintended recipient from suspecting that the data is there .The authors studied both the algorithms and studied the techniques that use both the algorithm to provide high degree of security and also compare the result on the basis of timing and avalanche effect..

## 2. LITREATURE SURVEY

Nearly everyone these days in steganographic systems uses image as cover object, Authors have investigated many such algorithms, many techniques have been proposed to conceal the secret data behind the cover image without leaving any mark. As described by Neil Johnson and sushil Jagodia [4], least significant bit (LSB) incorporation is an ordinary, straightforward and uncomplicated approach to implant secret data in a cover object. To hide the secret data with image as a covering file, LSB bit of a pixel is substituted with the message bit. In a 24-bit image cover file, each pixel is made of three colours (R, G, and B) and each colour is of 8 bit. Each pixel is modified by 3 bits of message by modifying LSB of each colour with the message bit. To the human eye, the resulting stego image will look indistinguishable to the cover image.

Rishav Ray, Jeeyan Sanyal, Debanjan Das, Asoke Nath [1] used a text file/MS word file as a cover file to hide the secret data. To increase the degree of security authors have first encrypted the secret data by using Modified Generalized Vernam Cipher Method (MGVCM) and then hide the encrypted text behind the text file. Authors proposed two new algorithms one is Encryption and Decryption Algorithm and second algorithm explained the method to hide the secret message behind the text file. Authors named the proposed algorithm “RJDA” which is architecture of joining these proposed two algorithms. To hide secret message inside the text file authors proposed a new method in which the bits of each character of secret message file is inserted in place of eight randomly selected blank space characters of the cover files. In this method author first converted the secret file in binary format where every character is represented by 8 bits. To hide each bit, authors use blank spaces, to hide bit–0 authors left the blank space as it is, and for bit-1 authors replace the blank space to a character having ASCII value 160 and this character also appear like a blank space on the screen or while printing

## 3. PROPOSED WORK

The proposed architecture is a combination of two algorithms first encryption/ decryption algorithm which shuffles the data in such a way that unauthorized person cannot determine the original meaning of the text and then hide the data behind any other text file.

The proposed algorithm is a block cipher that divides data into blocks of equal length and then encrypts each block using a special logical operation and key. This algorithm uses symmetric key technique for encoding and decoding of data i.e. it uses the same key at both ends. The attractive feature of proposed algorithm is its time complexity. It is simple and converts any plaintext into cipher text very fast. Another plus point of proposed algorithm is that it protects the cipher text from Brute-force attacks as the large variable key length used in the encryption process. This algorithm generates 8 different key to encrypt the plaintext into cipher text. Thus without knowing the exact sequence of key, it will be very hard to gain plaintext from cipher text.

## 3.1 Encryption

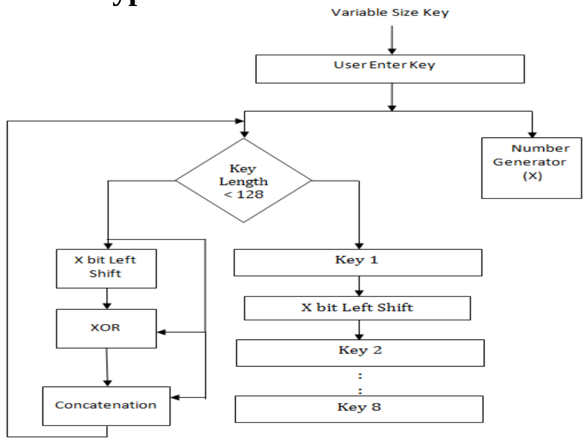

**Figure 1 (a): Generation of eight different keys**

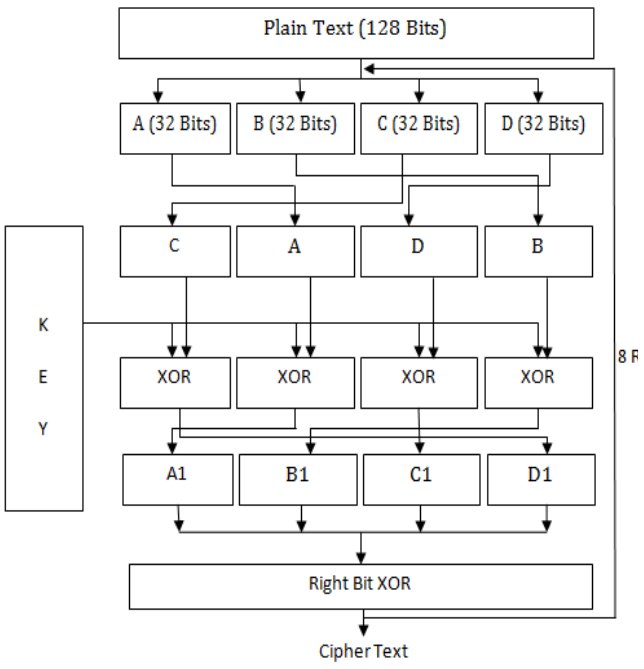

**Figure 1 (b): Proposed Encryption Block**

## 3.2 Proposed Encryption Algorithm

Proposed Encryption Algorithm is divided in to two part, first key generation and second encryption block. Steps to convert a plaintext into cipher text are as follows:

1. **Key Generation**: Figure 1 (a) shows the block diagram of key generation. Proposed key generation algorithm takes a variable length key. This key is used to generate a random number X and different 8 keys. Steps of creating 8 different keys are as follows:

   a. Key entered by user is used to generate a random number in between 0 to 7. This number is a sum of all the 1's bit mod 8.

   b. Now, if the length of key in bits is less than 128 then i. Perform left shift of operation on a key by X bits.

      i. Now, the result of step i xor with the key

      ii. Result of step ii is concatenated with the key.

      iii. Repeat step i to ii, while the key length is less than 128.

   c. If key length is greater than equal to 128 than the first 128 bits are used as first key. First key is left shifted by X bits and the result is key 2 repeat the same process to calculate all the eight keys.

2. Encryption Block: Encryption block converts the plaintext into cipher text. Steps of converting plaintext into cipher text are as follows:

   a. First the complete plaintext is divided in to a block of 128 bits. If last block contains less bits then padding of zero bits make them equal to 128 bits block.

   b. Now for each 128 bit plaintext block do the following steps:

   i. Divide the 128 bits block in to four 32 bits block.

   ii. Now, shuffle each block as shown in figure 1 (b).

   iii. Now the result of step ii is xor with key 1. (Keys are also divided in to four blocks of 32 bits.)

   iv. Result of step iii is again shuffled as mentioned in figure 1 (b).

   v. Now, all the block are concatenated and perform right bit xor operation on it. Right bit xor is xor of each bits with is right bits.

   vi. Repeat step i to step v eight times, but each time different keys are used from the key set generated at step 1.

   c. After eight round the result comes is the cipher text of first 128 bits. Repeat step b for 128 bit plaintext block.

   d. Exit.

## 3.3 Proposed Decryption Algorithm

Proposed decryption is exactly reverse of encryptionprocess. Decryption process again divided into two parts. First key generation and second is decryption block.

1. **Key Generation**: Key generation of decryption block is exactly same as mentioned in encryption process.

2. **Decryption Block:** Decryption block is a reverse of encryption block. Steps of converting cipher text into plaintext are as follows:

   *a.* First the complete cipher text is divided in to a block of 128 bits.

   *b.* Now for each 128 bit cipher text block do the following steps:

   i. Perform reverse right bit xor operation on 128 bit cipher text.

   ii. Divide the 128 bits block in to four 32 bits block.

   iii. Now, shuffle each block as shown in figure 2 (a).

   iv. Now the result of step ii is xor with key 8. (Keys are also divided in to four blocks of 32 bits.)

v. Result of step iii is again shuffled as mentioned in figure 2 (a).

vi. Repeat step i to step v eight times, but each time with different keys in reverse order are used from the key set generated at step 1.

c. After eight round the result comes is the plaintext of first 128 bits. Repeat step b for 128 bit cipher text block.

d. Exit.

### 3.4. Decryption Block Diagram

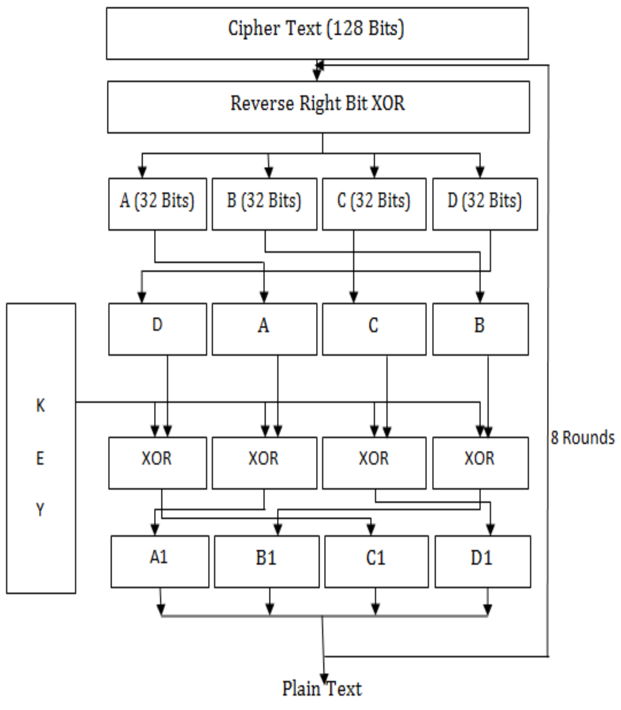

**Figure 2 (a): Proposed Decryption Block**

### 3.5 Steganography

After encrypting the entire message, cipher text is hiding behind the text file. For hiding the text behind the text, two characters are used first whose ASCII value is 32 and otherwhose ASCII value is 160. Both look like a space but there ASCII value are different.

Steps of hiding cipher text are as follows:

1. Convert the cipher text again in to binary form.
2. Select a cover text file having number of spaces equal or more then number of bits in cipher text.
3. Now, to hide the cipher text, find the first space in cover file, if the bit in cipher text is 0 leave the blank space as it is but if the cipher bit is 1 then replace the space with character whose ASCII value is equal to 160.
4. Repeat step 3 for all the bits in the cipher bit stream.At the time of unhiding, reverse process will be performed, means if space encounter whose ASCII value is 32 means 0 and space encounter whose ASCII value is 160 means 1.

## 4. PERFORMANCE ANALYSIS

This section gives the performance analysis of proposed algorithm. For proving the efficiency of proposed algorithm, authors have implemented the proposed algorithm and compare it with well known AES algorithm. The comparison of proposed algorithm is done different parameters. To compare the internal strength of proposed algorithm, avalanche effect is calculated. To compare its efficiency, time analysis is done and compares it with AES algorithm.

**A. Time Analysis:** Time analysis is one of the important features to compare any cryptographic algorithm. Encryption/ Decryption also used to provide confidentiality on real time communication. It is very important for any algorithm to be time efficient when it is used with real time communication. To check time efficiency of proposed algorithm, implementation analysis is done Table 1 and Table 2 shows the analysis result.

**Table 1: Comparison of Encryption Time of Proposed Algorithm on Various File Size in seconds**

| **File Size in KB** | **Algorithm** | |
|---|---|---|
| | Execution Time in Second | |
| | Proposed Algorithm | AES |
| **5 KB** | 0.152 | 0.741 |
| **10 KB** | 0.507 | 1.762 |
| **20 KB** | 2.222 | 3.887 |

Figure 3 and Figure 4 shows the graphical representation of Table 1 and Table 2. It is very clear from the result that proposed algorithm is time efficient than AES algorithm. Also time taken by proposed algorithm shows that it can be used for any real time communication.

**Avalanche Effect**: Avalanche effect is another parameter used to show the strength of proposed algorithm. An algorithm that is time efficient but not enough secure is waste. Therefore to show the internal strength of proposed algorithm avalanche effect is calculated. According to avalanche effect, little change in the key will makes 50% of bits change in the result. This is an ideal state. An algorithm closed to this condition considered more secure than other.

Table 3 shows the avalanche effect of proposed algorithm Analysis result is taken after experimenting it on 50 different sample files and average of all comes out to be a result.It is clear from the result that avalanche effect of proposed algorithm is high and much closed to the ideal state.

**.Table 2: Comparison of Decryption Time of Proposed Algorithm on Various File Size**

| File Size in KB | Algorithm | |
|---|---|---|
| | Execution Time (in Second) | |
| | Proposed Algorithm | AES |
| **5 KB** | 0.155 | 0.740 |
| **10 KB** | 0.510 | 1.777 |
| **20 KB** | 2.231 | 3.814 |

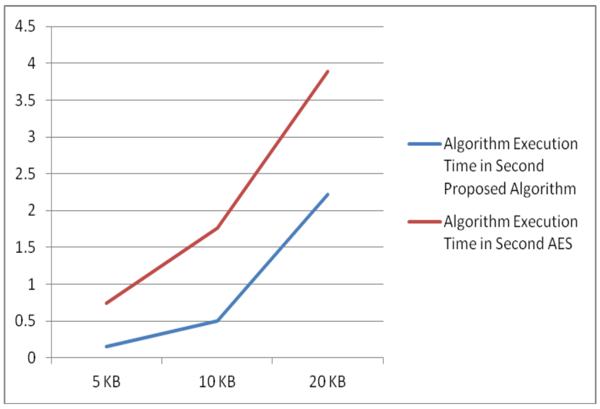

**Figure 3: Encryption time of the proposed algorithm**

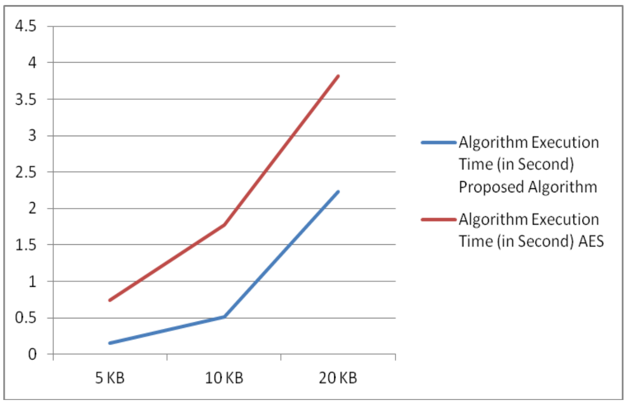

**Figure 4: Decryption time of the proposed algorithm**

**Table 3 Avalanche Effect of Proposed Algorithm**

| File Size in KB | Proposed Algorithm |
|---|---|
| | **Avalanche Effect** |
| Key-1 (ABCD) | 49.4475% |
| Key-2 (ABCE) | |

**C. Memory Requirements***:* Table 4 shows the memory requirement of proposed algorithm and memory requirement of AES algorithm. It is clearly understood from Table 4 that memory requirement of AES and proposed is same.

**Table 4: Memory comparison between Proposed Algorithm and AES**

| Algorithms Name | Key Length in Bits | Plain Text Length in Bits (Input Block) | Cipher Text Length in Bits (Output Block) |
|---|---|---|---|
| **Proposed Algorithm** | 1-128 | 128 | 128 |
| **AES** | 128 | 128 | 128 |

**D. Key Analysis***:* Security of any encryption algorithm is on key, if intruder get the access on this key, no matter how strong the algorithm is, they can easily get the secret text. Proposed algorithm uses variable length key which is used to generate eight 128 bits key. Hence the complexity to break the proposed key is 2128 which is near to impossible to solve even with a super computer.

## 5. FUTURE WORK

The proposed architecture is an efficient approach of providing security and confidentiality over the transmitted data or stored data. This architecture proposed a new encryption/decryption algorithm which not only secure but also time efficient. Further in order to improve the security, the resultant cipher text is hiding behind the text file which improves the confidentiality of proposed architecture. Efficiency of algorithm makes it suitable for AD-HOC network and also makes it suitable for real time communication.

## 6. ACKNOWLEDGMENTS

This project is the result of guidence and support of faculty of my college .I express my ardent and earnest gratitude to my guide Mr.Anshul Sarawagi and my Head of Department Dr. Manish Shrivastva.

## 7. REFERENCES

[1] Rishav Ray, Jeeyan Sanyal, Debanjan Das, Asoke Nath."A new Challenge of hiding any encrypted secret message inside any Text/ASCII file or in MS word file: RJDA Algorithm". 2012 IEEE International Conference on Communication Systems and Network Technologies

[2] Clair, Bryan. "Steganography: How to Send a Secret Message." 8 Nov. 2001.

[3] Westfeld, A., and G. Wolf, Steganography in a Video conferencing system, in proceedings of the second international workshop on information hiding, vol. 1525 of lecture notes in computer science, Springer, 1998.pp. 32-47.

[4] Johnson, N. F. and Jajodia, S, "Exploring steganography: Seeing the unseen", IEEE Computer Magazine, pp. 26-34, February 1998.

[5] William Stallings, Cryptography and Network Security, Principles and Practice, Third edition, Pearson Education, Singapore, 2003.

[6] Symmetric Key Cryptography using Random Key generator: Asoke Nath, Saima Ghosh, Meheboob Alam Mallik: "Proceedings of International conference on security and management(SAM'10" held at Las Vegas, USA Jul 12-15, 2010), P-Vol-2, 239-244(2010).

[7] An Integrated Symmetric key Cryptography Algorithm using Generalised modified Vernam Cipher method and DJSA method: DJMNA symmetric key algorithm : Debanjan Das, Joyshree Nath, Megholova Mukherjee, Neha Choudhary, Asoke Nath: Communicated for publication in IEEE International conference WICT 2011 to be held at Mumbai Dec 11-14, 2011.

[8] Data Hiding and Retrieval: Asoke Nath, Sankar Das, Amlan Chakraborty, published in IEEE "Proceedings of International Conference on Computational Intelligence and Communication Networks (CICN 2010)" held from 26-28 NOV' 2010 at Bhopal.

[9] Advanced Steganographic approach for hiding encrypted secret message in LSB, LSB+1, LSB+2, LSB+3 bits in non standard cover files : Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath, International Journal of Computer Applications, Vol- 14, No. 7, Page-31-35, Feb (2011).

[10] Advanced Steganography Algorithm using encrypted secret message: Joyshree Nath and Asoke Nath, International Journal of Computer Science and Applications, Vol-2, No. 3, Page- 19-24, Mar (2010).

[11] A Challenge in hiding encrypted message in LSB and LSB+1 bit positions in any cover files: executable files, Microsoft Office files and database files, image files, audio and video files : Joyshree Nath, Sankar Das, Shalabh Agarwal and Asoke Nath : JGRCS, Vol-2, No. 4, Page- 180-185, Apr (2011).

[12] New Data Hiding Algorithm in MATLAB using Encrypted secret message: Agniswar Dutta, Abhirup Kumar Sen, Sankar Das, Shalabh Agarwal and Asoke Nath : Proceedings of IEEE CSNT- 2011 held at SMVDU (Jammu), 03-06 Jun, 2011, Page 262-267.

[13] New Steganography algorithm using encrypted secret message: Joyshree Nath, Meheboob Alam Mallik, Saima Ghosh and Asoke Nath : Proceedings of Worldcomp 2011 held at Las Vegas (USA), 18-21 Jul, 2011.

[14] Steganography In Digital Media: Principles, Algorithms and Applications by Jessica Fridrich : Cambridge University Press.

[15] Cryptography and Network Security, William Stallings, Prentice Hall of India.

[16] Cryptography & Network Security, Behrouz A. Forouzan, Tata McGraw Hill Book Company.

[17] Cryptography and Information Security, V. K. Pachghare, Prentice Hall of India.

[18] SANS Security Essentials, (volume 1.4, chapter 4) Encryption and Exploits, 2001.

[19] Petitcolas, Fabien A.P., "Information Hiding: Techniques for Steganography and Digital Watermarking.", 2000.

[20] StegoArchive, "Steganography Information, Software and News to enhance your Privacy", 2001

[21] Petitcolas, Fabien A.P., "The Information Hiding Homepage: Digital Watermarking and Steganography"

[22] Johnson, Neil F., "Steganography", 2000.

[23] The WEPIN Store, "Steganography (Hidden Writing)", 1995

[24] Sellars, D., "An Introduction to Steganography"

[25] Bender, W., "Techniques for Data Hiding", IBM Systems Journal, Vol. 35, Nos 3+4, Pgs 313-336, 1996.

[26] Krinn, J., "Introduction to Steganography", 2000

[27] Noto, M., "MP3Stego: Hiding Text in MP3 files", 2001

[28] "Chameleon", Image Steganography by Mark David Gan

[29] Translation-Based Steganography: Christian Grothoff, Krista Grothoff, Ludmila Alkhutova, Ryan Stutsman, Mikhail Atallah

[30] Hide and Seek: An introduction to Steganography: Niels Provos, Peter Honeyman.

[31] Modren Steganography: Miroslav Dobší_ek