Volume 3, Issue 11, November 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Comparison of Novel Network Security Algorithms for Securing and Screening Data Using Steganography

K. Devika Rani Dhivya
M.Sc.(CS)., M.Phil.,
Assistant Professor, Dept. of CA and SS,
Sri Krishna Arts and Science College,
Coimbatore, Tamil Nadu, India

Abstract- ***This paper deals with comparison of network security algorithms like LSB (Least Significant Bit) algorithm, RSA (Rivest-Shamir-Adleman) algorithm and DES (Data Encryption Standard) algorithm for Steganography, which based on hiding large amount of data (image, audio, text etc.,) file into color BMP and TIF images. So image segmentation is used with bit replacement on the appropriate pixel. Here the Pixels are selected randomly rather than sequentially. According to the step of design, two types of images formats are used for as a cover for the input data into color Bitmap and TIF image. This comparison shows the efficiency of these network security and steganography algorithms to hide large amount of data with high security.***

Key words: ***LSB (Least Significant Bit) algorithm, RSA (Rivest-Shamir-Adleman) algorithm and DES (Data Encryption Standard) algorithm, Steganography.***

I. INTRODUCTION

Steganography is process hiding information into multimedia content like audio, video and image etc for secure communication over insecure network. Information hiding into images is a popular technique. The goal is to secure the communications from an eavesdropper [4]. Steganography comes from the Greek words Steganós (Covered). Most of the time it is confused the both terms Steganography and cryptology because these are similar in the way that they are used to secure the information from the hackers. The difference between the two is that Steganography involves hiding information. But the cryptography is the process of encrypting the information, which is in the form that is not understood to anyone. If anyone sees the object that the information is hidden inside, so that will have no aim that there is any hidden information, thus they will not attempt to decrypt the information.

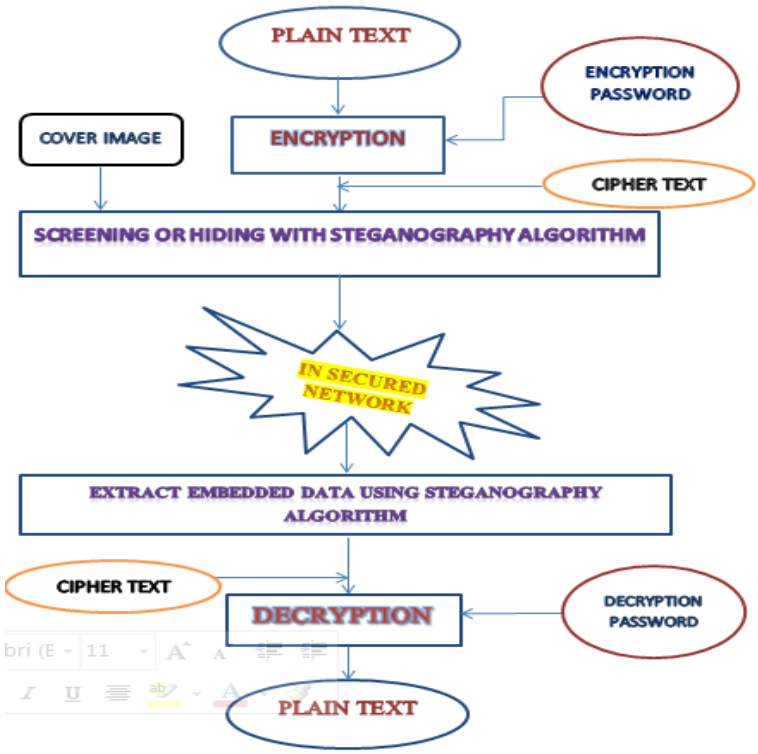

Fig. 1 Processing flow to send secure Data using Steganography algorithm.

The common use of steganography is to hide a file inside another file. When information or a file is hidden inside a carrier file, the data is usually encrypted with a password. Many steganography systems are weak against visual and statistical attacks. Sending encrypted information will create intuition when invisible information is not working [10]. Whenever the steganography become fails and the message cannot be detected if a cryptography technique is used. To hide a message inside an image without changing its visible properties, the cover source can be altered with many numbers variations. Like most other forms of cryptography and the secret writing, steganography has developed in the digital period of time most the documents contain useless or unwanted parts of data, some of their information can be altered without obvious effect [7].

The above figure shows the process of the flow to send information in a secure manner. The process starts with plain text (the original information). The plain text then converted into encrypted form (cipher text) by using network security algorithms with its key. The next process is that hide or screen the cipher text into the image. Here image is act as a cover. Then the covered image with cipher text is sent through insecure network. At the other side the cipher text is extracted by steganography algorithm and the plain text is retrieved by using network security algorithm with its key. Thus there will be a secure communication of information that cannot retrieve by the secret listener (eavesdroppers).This paper describes public key cryptography. For the security purpose the algorithm uses two types of keys. One is public key another one is private key. The RSA algorithm uses public key and DES algorithm uses the Secret Key for encoding and decoding the data [3][11]. Here the section 2 gives the overview of LSB algorithm. The section 3 contains RSA algorithm overview. The section 3 describes the overview of DES algorithm.

II. LSB ALGORITHM

LSB (Least Significant Bit) algorithm is a very popular methodology, which replaces the least significant bit in some bytes of the cover file to hide a sequence of bytes containing the hidden data [4]. It is an effective technique, where the LSB substitution doesn't cause significant quality degradation, such as in 24-bit bitmaps. [4]

For illustration, to hide the letter "D" (ASCII code 68 that is 01000100) inside 8 bytes of a cover image, it will be the LSB of each byte like the following form:

10010010
01010011
10011010
11010010
10001010
00000011
01110010
00101010

The above decoding contain 8 Least Significant Bits of those bytes to rearranges the hidden byte, that is 01000100—the letter "D". Thus using this technique it will hide a byte of every eight bytes of the cover.

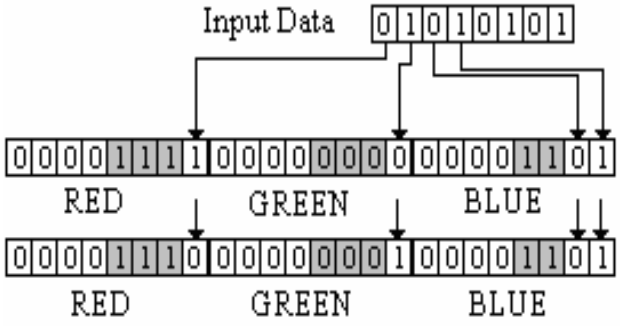

Fig 2 Data Hidding using random bit allocation

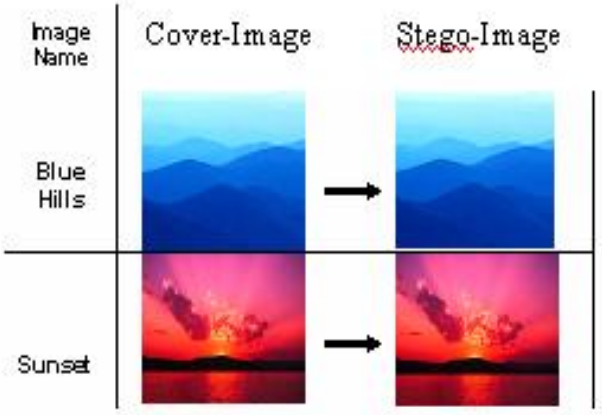

Fig. 3 Bitmap image bfore and after data hidding.[7]

The above algorithm changes the least significant bit of the bitmap image to encrypt the data. Even though it is very effective, rarely if one least significant bit changed then the whole information can be changed**.** Further to this we can also provide public key cryptography in BMP and TIF images by using RSA algorithm for security purpose.

III. RSA ALGORITHM

RSA is a cryptographic algorithm provides security for the information. Which initialize two types of keys, first one is public key and the other one is private key [1]. The public key is known to all and it is used for encrypting the messages [7]. Messages, which are encrypted with the public key can only be decrypted using the private key.

The key steps for the RSA algorithm,

- Two large random primes, p and q.
- Compute n = pq and (φ) phi = (p-1)(q-1).
- Choose an integer e, $1 < e < phi$, such that gcd(e, phi) = 1.
- Compute the secret exponent d, $1 < d < phi$, such that $ed \equiv 1 \pmod{phi}$.
- The public key - (n, e) and
- The private key - (n, d).
- Keep all the values d, p, q and phi secret.

To generate the primes *p* and *q*, generate a random number of bit length *b* is the required bit length of *n*, incrementing by 2, just generates another random number each time. There are stricter rules in ANSI X9.31 [2] to produce *strong primes* and other restrictions on *p* and *q* to minimize the possibility of known techniques being used against the algorithm. There is much argument about this topic. It is probably better just to use a longer key length. Also, having chosen *e*, it is simpler to test whether gcd (e, p-1) =1 and gcd(e, q-1)=1.

Values of *p* or *q* that fail this test can be rejected there and then. (Even better: if *e* is prime and greater than 2 then you can do the less-expensive test (p mod e)! =1 instead of gcd (p-1,e)==1.) Use the Extended *Euclidean Algorithm* to calculate $d = e^{-1} \bmod phi$, also written $d = (1/e) \bmod phi$. This is known as *modular inversion*. [8] The modular inverse *d* is defined as the integer value such that $ed = 1 \bmod phi$. It only exists if *e* and *phi* have no common factors and the integer *m*, If m = 0 or 1 or n-1 there is no security as the cipher text has the same value. For more details on how to represent the plaintext octets as a suitable representative integer *m*, It is important to make sure that m < n otherwise the algorithm will fail. This is usually done by making sure the first octet of m is equal to 0. [8]

Encryption:

"A" the sender does the following:-

- Needs the recipient B's public key (n, e).
- The plaintext message is represented as a positive integer *m*.
- Computes the cipher text $c = m^e \bmod n$.
- Sends the cipher text *c* to B.

Decryption:

"B" the recipient does the following:-

- Needs his private key (n, d) to compute $m = c^d \bmod n$.
- Get the plaintext from the message representative *m*.

After encryption it produces a cipher text from a message which is under the control of a public key, and a decryption recovers the message from the cipher text under the control of the appropriate private key [5,8,9]. Fig 3 shows the image of water lilies and winter before and after hiding the data.

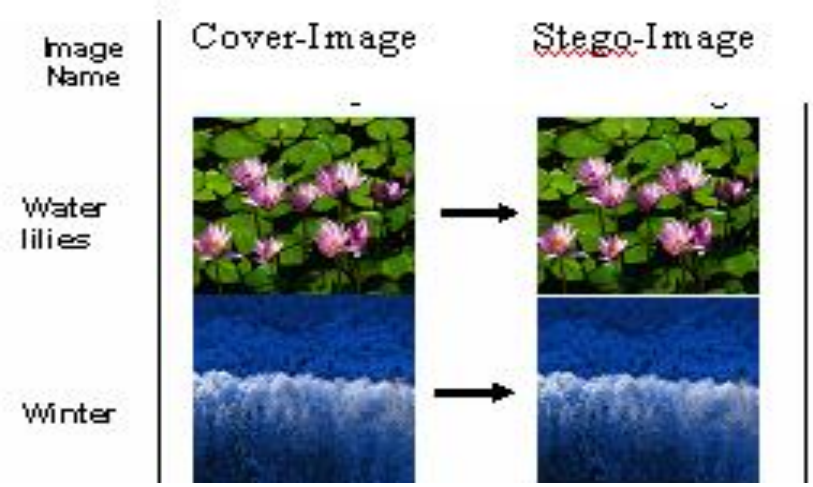

Fig. 4 Water lilies and Winter bitmap images before and after data hiding. [7]

The above algorithm encrypts and decrypts the information by using two effective keys. Even though it is very effective, rarely the information may leak by the eavesdroppers. Further to this we can also provide secret key cryptography with block cipher in BMP and TIF images by using DES algorithm for security purpose.

IV. DES ALGORITHM

The DES (Data Encryption Standard) algorithm is the most widely used encryption algorithm. It is a block cipher and it works on plaintext and returns cipher text. It was highly influential in the advancement of modern cryptography in. It is a popularly method of data encryption which uses a private or secret key. For given message, the key is chosen at random from the number of keys. Here both the end users (the sender and the receiver) have to know and they must use the same private key. The workings of DES are based on a cipher known as the block cipher. [11]

DES expects inputs and gives the outcome as:

- The Plaintext is to be encrypted and
- The Secret Key
- Decrypted then the plain test is convert into cipher text by using secret key.

Whenever the plaintext is accepted and then the key used for encryption and decryption, here both text is consider as the type of ciphers. DES is therefore a symmetric, whenever the original information is received to be encrypted, and then it is formed into 64 bit blocks as input. If the number of bits in the message is not evenly divisible by 64, then the last block will be removed. Like this ever information is encrypted and using the secret key the encrypted text is converted into original form. Thus this algorithm is more efficient than the other two cryptographic algorithms.

V. RESULT AND DISCUSSION

Confidence in the present result is gained by comparing of the result obtained from the above steganography algorithms (LSB, RSA, and DES algorithms).

A. Comparison With Related Work

We use more than 50 BMP images to test the present algorithm and to be sure that the aim of the data embedding is satisfied. In this work, it shows the efficiency of embedding data is very high when it is compared to the related work. We have been shown the comparison result of BMP images. The result describes the level of efficiency according to the amount of hidden data.

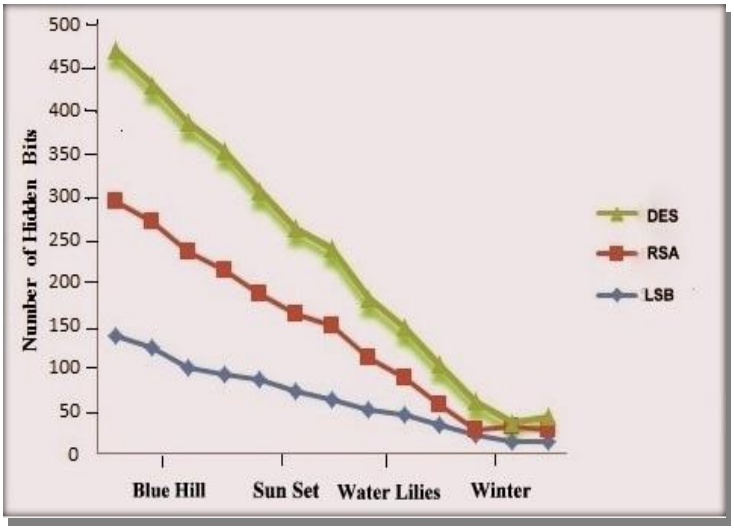

Fig. 5. The amount of hidden data for bitmap images using the steganography algorithms

The above figure gives the compares the amount of hidden data on selected images from Fig.3-4. This comparison shows that the DES algorithm is efficiently secure and hide large amount of data and exceeds the capabilities of the related algorithms [LSB and RSA].

VI. CONCLUSION AND FUTURE ENHANCEMENT

This paper satisfies the aim. Steganography is an effective way to obscure data and hide sensitive information. The present algorithms allows an individual to hide data inside other data with hopes that the transfer medium will be so obscure that no one would ever think to examine the contents of the file. The algorithm is possible to implement a steganography algorithm to hide a large amount of data. Two layers of security to secured data by obscuring the context in which it was transferred. Future this work can be enhancing with algorithms like TMA, S-tools etc with other image format like JPEG, PING etc.

REFERENCES:

1. Ajtai .M and C. Dwork, *A public-key cryptosystem with worst-case/average-case equivalence*, Proc. 29th ACM STOC *(1997), 284-297.*
2. ANSI X9.31-1998, *Digital Signatures using Reversible Public Key Cryptography for the Financial Services Industry (rdsa)*, Appendix A, American National Standards Institute, 1998.
3. *A Method for Obtaining Digital signature and Public-keycryptosystems.* Communications of the ACM, 21 (2), pp. 120-126, February 1978.
4. Chandramouli, R. And Memon. N , 2001. *Analysis of LSB based image steganography techniques*. Proc. Of ICIP, Thessaloniki, Greece.
5. R. Housley, *Cryptographic Message Syntax (CMS)*, September 2009 (obsoletes RFC3852, RFC3369, RFC2630).
6. Dumitrescu, S, W. Xiaolin and Z. Wang, 2003, *Detection of LSB steganography via sample pair analysis*. In: LNCS, Vol. 2578, Springer-Verlag, New York, pp: 355–372.
7. Nameer N. EL-Emam ,"*Hiding a Large Amount of Data with High Security Using Steganography Algorithm*", Journal of Computer Science 3 (4): 223-232, 2007 SSN 1549-3636 2007 Science Publications
8. Menezes, van Oorschot and Vanstone, *Handbook of Applied Cryptography*, CRC Press LLC, 1997.
9. PKCS #1 v2.1: *RSA Cryptography Standard*, RSA Laboratories june 14, 2002.(ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf).
10. S-Tools (http:// digitalforensics. Champlain. Edu/download/ s-tools 4.zip).
11. www.facweb.iitkgp.ernet.in/~sourav/DES.pdf

