ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

Providing Data Security by Hidding Large Amount of Data Using Steganography

Kingsly.P, Akila.R

PG Scholar, Dept. of Computer Technology, Sri Krishna Arts and Science College, Coimbatore, Tamil Nadu, India

Assistant professor, Dept. of Computer Technology, Sri Krishna Arts and Science College, Coimbatore, Tamil Nadu, India

ABSTRACT: This paper deals with comparison of network security algorithms like , RSA (Rivest-Shamir-Adleman) algorithm and DES (Data Encryption Standard) algorithm, TRIPLE DES algorithm, for Steganography, based on hiding large amount of data (image, audio, text etc.,) file into colour BMP and TIF images. Here image segmentation is used with bit replacement on the appropriate pixel. These Pixels are selected randomly rather than sequentially. According to the step of design, two types of images formats are used as a cover all aspects of the input data into colour Bitmap and TIF image. High security layer have been proposed through two layers to make it difficult to break through the encryption of the input data. This comparison shows the efficiency of these network security algorithms to hide large amount of data with high security.

KEYWORDS: RSA (Rivest-Shamir-Adleman) algorithm, DES (Data Encryption Standard) algorithm and TRIPLE DES Steganography.

I.INTRODUCTION

Steganography is an art and science of information hiding and invisible communication. Hiding information inside images is a popular technique. The goal is to secure communications from an eavesdropper [4]. Sending encrypted information will arouse suspicion while invisible information will not work [10]. When the steganography 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 in “noisy” areas with many colour variations. Like most other forms of cryptography and the secret writing, steganography has thrived in the digital era most digital documents contain useless or insignificant areas of data, or involve enough redundancy some of their information can be altered without obvious effect [7].

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][12].

DOI: 10.15680/IJIRCCE.2017. 0507042

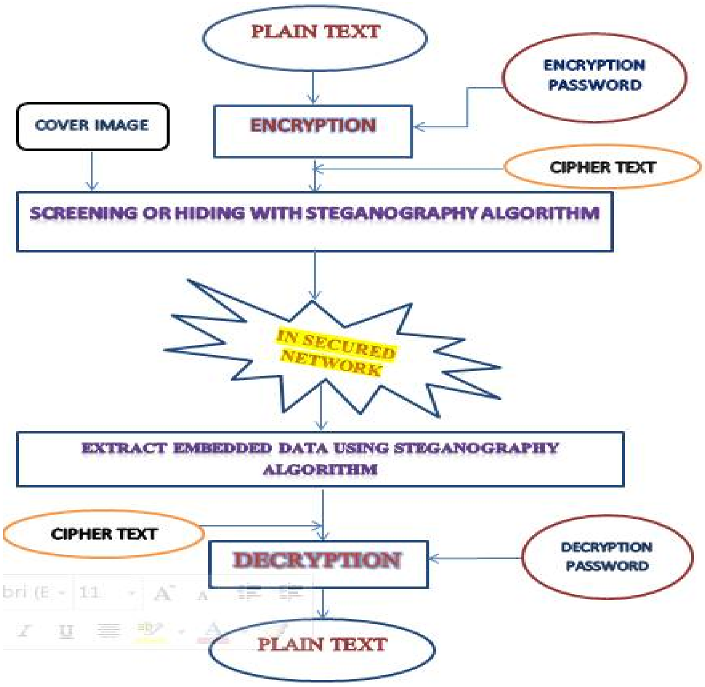

Fig 1: Processing flow to send secure Data using Steganography algorithm.

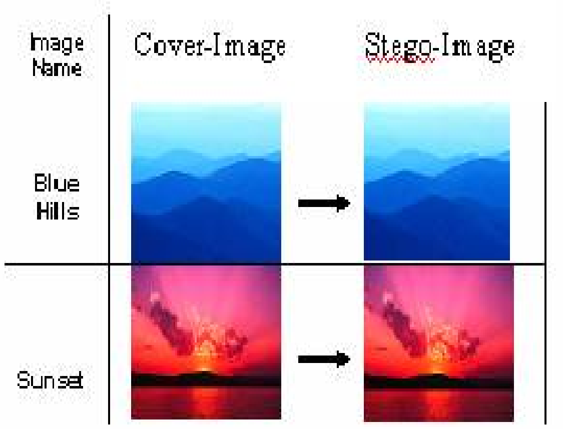

Fig 2: Bitmap image bfore and after data hidding.[7]

The above algorithm, changing 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.

II.RELATED WORK

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 eavesdroppe;The CMS describes an encapsulation syntax for data protection. Itsupports digital signatures and encryption. The syntax allows multiple encapsulations;An encryption method is presented with the novel property that publicly re-vealing an encryption key does not thereby reveal the corresponding decryptionkey.

III.RSA ALGORITHM

RSA provides security by setting two keys, a public key and a private key [1]. The public key can be known to everyone and is used for encrypting messages [7]. Messages encrypted with the public key can only be decrypted using the isolated key. The keys for the RSA algorithm,

1. Generate two large random primes, p and q, of around equal size such that their product n = pq is of the required bit length, e.g. 1024 bits.

2. Compute n = pq and (φ) phi = (p-1)(q-1).

3. Choose an integer e, 1 < e < phi, such that gcd(e, phi) = 1.

4. Compute the secret exponent d, 1 < d < phi, such that ed ≡ 1 (mod phi).

5. The public key is (n, e) and the private key is (n, d). Keep all the values d, p, q and phi secret. To produce 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 likelihood 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 while generating and testing the primes in step 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} mod phi, also written d = (1/e) mod phi. [8] This is known as modular inversion. The modular inverse d is defined as the integer value such that ed = 1 mod 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,[8]. 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.

Encryption:

Sender A does the following:-

1. Obtains the recipient B’s public key (n, e).
2. Represents the plaintext message as a positive integer m .
3. Computes the cipher text c = m^e mod n.
4. Sends the cipher text c to B.

Decryption:

Recipient B does the following:-

1. Uses his private key (n, d) to compute m = c^d mod n.
2. Extracts the plaintext from the message representative m.

An encryption primitive produces a cipher text representative from a message representative under the control of a public key, and a decryption primitive recovers the message representative from the cipher text representative under the control of the corresponding private key [5,8,9]. Fig 4 shows the image of water lilies and winter before and after hiding the data.

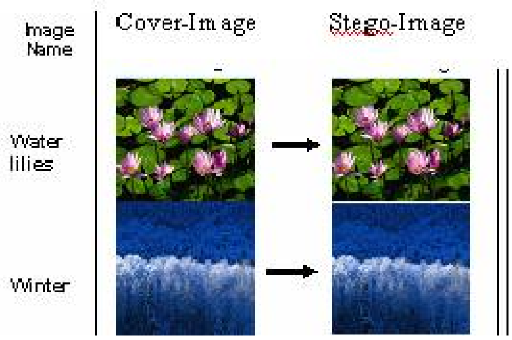

Fig 3. Water lilies and Winter bitmap images before and after data hiding [7]

VI. DES ALGORITHM

The DES algorithm is the most extensively used encryption algorithm. It is a block cipher and it functions on plaintext blocks of a given size (64-bits) and returns cipher text blocks of the same size. It was highly influential in the development of modern cryptography in. Data Encryption Standard (DES) is a widely-used method of data encryption using a private (secret) key. For each given message, the key is chosen at random from among the massive number of keys. Like other private key cryptographic methods, both the sender and the receiver must know and use the same private key.

DES applies a 56-bit key to each 64-bit block of data. The process can run in numerous modes and involves 16 rounds or operations. Although this is measured "strong" encryption, many companies use "triple DES", which applies three keys in succession. The workings of DES are based on a cipher known as the Feistel block cipher. This was a block cipher. [11]

It consists of a number of rounds where each round contains:

- Bit-shuffling,
- Non-linear substitutions (S-boxes) and
- Exclusive OR operations.

DES expects two inputs:

- The Plaintext to be encrypted and
- The Secret Key.

The manner in which the plaintext is accepted, and the key procedure used for encryption and decryption, both determine the type of cipher it is.

DES is therefore a symmetric, 64 bit block cipher as it uses the same key for both encryption and decryption and only operates on 64 bit blocks of data at a time5 (be they plaintext or cipher text). The key size used is 56 bits, however a 64 bit (or eight-byte) key is actually input. The least significant bit of each byte is either used for parity (odd for DES) or set arbitrarily and does not increase the security in any way. All blocks are numbered from left to right which makes the eight bit of each byte the parity bit.

Once a plain-text message is acknowledged to be encrypted, it is arranged into 64 bit blocks required for input. If the number of bits in the message is not evenly divisible by 64, then the last block will be padded. Multiple permutations and substitutions are incorporated throughout in order to increase the difficulty of performing a cryptanalysis on the cipher.

V.TRIPLE DES

In cryptography, Triple DES (3DES), officially the Triple Data Encryption Algorithm (TDEA or Triple DEA), is a symmetric-key block cipher, which applies the Data Encryption Standard (DES) cipher algorithm three times to each data block

ALGORITHM :

A native approach to increase strength of a block encryption algorithm with short key length (like DES) would be to use two keys (K1, K2) instead of one, and encrypt each block twice: EK2(EK1(plaintext)). If the original key length is n bits, one would hope this system provides security corresponding to using key 2n bits long. Unfortunately, this approach is susceptible to meet-in-the-middle attack given a known plaintext pair (x, y), such that y = EK2(EK1(x)), one can improve the key pair (K1, K2) in ~2n steps, instead of ~22n steps one would expect from algorithm with 2n bits of key.
Therefore, Triple DES uses a "key bundle" that comprises three DES keys, K1, K2 and K3, each of 56 bits (excluding parity bits). The encryption algorithm is:

ciphertext = EK3(DK2(EK1(plaintext)))

I.e., DES encrypt with K1, DES decrypt with K2, then DES encrypt with K3.

Decryption is the reverse:

plaintext = DK1(EK2(DK3(ciphertext)))
I.e., decrypt with K3, encrypt with K2, then decrypt with K1.

Each triple encryption encrypts one block of 64 bits of data.

In each case the middle operation is the reverse of the first and last. This improves the strength of the algorithm when using keying option 2, and provides backward compatibility with DES with keying option 3.

1.RESULT AND DISCUSSION:

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

1.1Comparison 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 inserting is satisfied. In this work, it shows the efficiency of embedding data is very high when it is associated to the related work. We have been shown the comparison result of BMP images. The result describes the level of effectiveness according to the amount of hidden data.

Fig 5: compares the amount of hidden data on selected images from Fig.3-4.

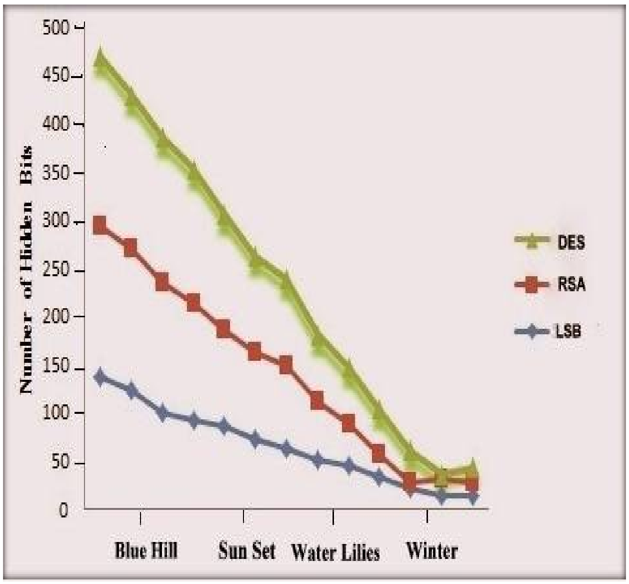

Fig 4. The amount of hidden data for bitmap images using the steganography algorithms

This comparison shows that the Triple DES algorithm is competently hide large amount of data and exceeds the abilities of the related algorithms [LSB and RSA].

VI.CONCLUSION AND FUTURE WORK

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 which is described by pseudo-code is presented and it 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 enhance 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. Cryptographic Message Syntax (CMS), R. Housley, 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.