DEVELOPMENT OF SECURE COMMUNICATION USING ENCRYPTION AND STEGANOGRAPHY

Ayush Vats[1], Akshay Bajetha[2], Ishant Kushwaha[3], Aryan Ujalyan[4], Ms. Anshul Khanna[5]

[1]Student, Dept. of Information Technology, Inderprastha Engineering College, Uttar Pradesh, India
[2]Student, Dept. of Information Technology, Inderprastha Engineering College, Uttar Pradesh, India
[3]Student, Dept. of Information Technology, Inderprastha Engineering College, Uttar Pradesh, India
[4]Student, Dept. of Information Technology, Inderprastha Engineering College, Uttar Pradesh, India
[5]Assistant professor, Dept. of Information Technology, Inderprastha Engineering College, Uttar Pradesh, India

Abstract - *Data security is of paramount importance in this technological world where many people interact with the Internet to perform their normal tasks, such as online transactions containing confidential data. It's a big deal in one's life because data plays an important role, whether it's business, online billing or online payment, where information should be included. Security data is required, as there is always someone present when reading those confidential data. The paper is about security using steganography. Steganography is the art of hiding information in ways to prevent the discovery of hidden messages. Secure data transfer through a software network, trying to convert your own data files to a specific code by using "Tiny Encryption Algorithm". Data encryption plays a very important role in the real-time environment to keep data inaccessible to unauthorized people, so that it can be changed and interrupted. After encryption, files can be transferred securely using steganography. The request should be a back-up process as it should be in a position to decrypt the data in its original format at the right request by the user.*

***Key Words*: Tiny encryption algorithm, Steganography, LSB, Decryption, Round function**

1. INTRODUCTION

Traditionally, where cryptography was a military province too, and more recently, of the banking community, ciphers are usually launched in Hardware. Everyone is familiar with the Enigma machinery used in the trick the Germans in World War II, and other excellent articles on them appear on these pages. Anyone who has worked in the back office has seen SWIFT facilities used for bank transfer. These are Hardware devices are also typical of how cryptography is used the first community made up of the advent of the electronic computer. Nowadays, almost all cipher algorithms are needed in software to work with it applications running on PCs.[2] In this paper, we explain the Tiny Encryption Algorithm (known for its simple TEA dictionary), which is very likely efficient - and very fast - built-in software writing algorithm. Code too small can be memorized, and easily edited in a few minutes in virtually any computer language of the user's choice TEA was developed by David Wheeler and Roger needham at the Computer Laboratory of Cambridge University and presented for the first time at a Fast Software Encryption workshop in Cambridge 1994. [1]

2. LITERATURE SURVEY

2.1 Tiny Encryption Algorithm

The Tiny Encryption Algorithm (TEA) is a block cipher encryption algorithm that is very easy to use, has fast execution time, and takes up minimal storage space. The embedded example is compiled in cryptography, the Tiny Encryption Algorithm (TEA) is a block cipher that is easily identifiable with its interpretation and functionality, a few strings of code. It (TEA) is much safer. The Tiny Encryption Algorithm (TEA) is a specific way of hiding information. Encryption is the process of converting information from one form (usually readable), to another form (not usually human-readable). The algorithm breaks the data into pieces, called blocks, 64 in size. It uses the 128-bit key, which is part of the external information, previously known, to perform this conversion [2]. The Tiny Encryption Algorithm (TEA) is one of the fastest and most effective algorithms available. The (TEA) is a symmetric (private) encryption key for the algorithm.[1][7]

2.2 Steganography

The same general terms required to understand the following steganography system-

Original data: Works as a media page for encrypting data.

Private message: It is the data from which we will hide the original data.

Keys: A key is a number or number. The embedding process and the extraction process are both key.

In video steganography, video signals are used to hide secret information. The objective is to hide large amount of secret data in video files.[1]

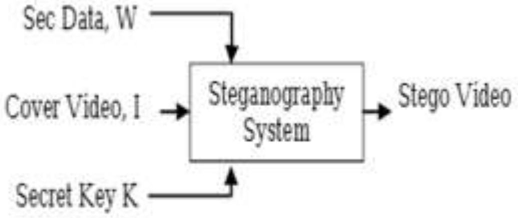

Fig-1: General block diagram of video steganography embedding algorithm

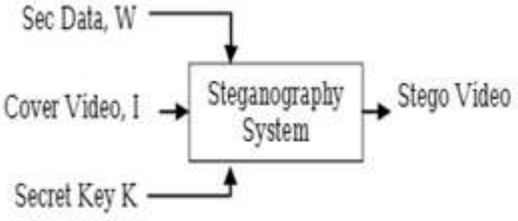

Fig-2: General block diagram of video steganography extraction algorithm

2.3 Embedding Process

Video Steganography for embedding imagery is the art and science of hiding images by embedding images inside a video file, seemingly harmless images. A hidden image or files may continue to hide information using steganography, so even if the hidden file is broken, the hidden message is not recognized. LSB method is used along with Masking Filtering and Transformations techniques to hide private imagery or other files.[4]

3. PROPOSED WORK

This paper proposed the techniques which provide security during data transmission across the network. In this case the sender writes specific information to a specific form using the "Tiny Encryption Algorithm". This algorithm was used because it requires less memory. It uses simple functions only, so it's easy to use. While encrypting specific information into a specific form, the key file is inserted by the sender. The purpose of the key file is to provide security to the system as it is known only to the sender and the receiver. Hidden data will embed a video file using the concept of steganography. Using the input / output packages steganography will read the video file and the encrypted data and take it as a video file. So, whenever a hacker tries to open a file, only the video file is visible to them. After that the video file is sent to the network. The receiver will receive a video file from the network. The recipient will then embed the encrypted data into the video file. The decryption is only performed when the data recipient enters the correct key. The information is then transmitted from sender to receiver in a secure way.

4. METHODOLOGY

4.1 Tiny Encryption Algorithm

Encrypted data using TEA is embedded in a video file using Steganography and Input / Input Packages. This file can be is transmitted over a network with high security to another user. The recipient can embed the video file and separate the original data using the same key used during encryption.

The following notifications are required here:

- Hexadecimal numbers will be written as “h,” e.g., 10 = 16. h
- Logical Change: The left shift of x in bits is determined by x << y. The logical shift of x's right y denoted by x >> y.

- Bitwise Equations: Left rotation of x in bits is denoted by x <<< y. The right rotation of x in bits is calculated by x>>> y.
- Exclusive-OR: The performance of the installation of n-tuples in the field (also known as 2F special-or) is indicated by x⊕y.

The Enhanced Tiny Encryption Algorithm is a Feistel type cipher that uses functions from mixed algebraic groups. Two Switching causes all data volumes and keys to be merged more often [7].

An important algorithm for a simple schedule; the 128-bit K key is divided into four 32-bit bits each with K = (K [0], K [1], K [2], K [3]). In Feistel cipher, encrypted is divided into two parts. The round function, F, is applied to one part using a small key and the F output is (special-or-ed (XORed)) and the other part. The two halves are then replaced. Each cycle follows the following the same pattern except for the last round where it is usually unchanged.[5]

- For each round I enter the upper left [1] and the Right [1], based on the previous round, as well as the lower left K [i] key from 128-bit global K.
- The sub-keys [K] are different from K and from each other.
- Permanent delta = (51 / 2-1) * 231 = 9E3779B h, taken from a gold numerical measure to ensure that the keys are unique and its exact value has no cryptographic value.
- The circular function differs slightly from Fiestel's classical structure in the number 2 addition used instead of special-or as a combining operator.

The cipher text as output is compressed into a video file using the input / output of Java packages [6].

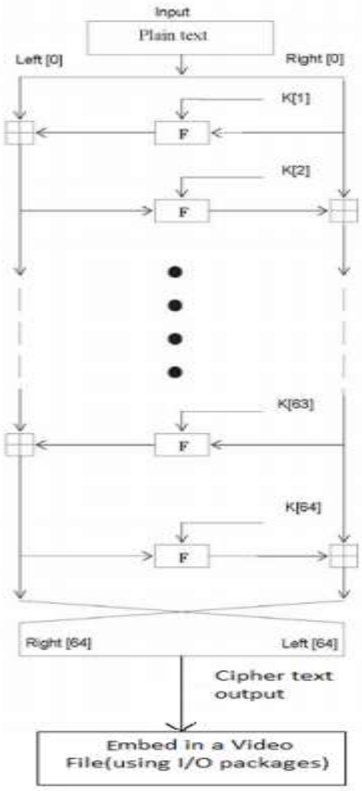

Fig-1: Diagram for encryption

Above The diagram provides the internal details of the ETEA cycle. The round function, F, consists of adding a key, slowly XOR and left-handed operation. We can define the output (Left [i +1], Right [i +1]) of the ETEA cycle input (Left [i], Right [i]) as follows

Left [i + 1] = Left [i] F (right [i], K [0, 1], delta [i]),

Right [i +1] = Right [i] F (right [i +1], K [2, 3], delta [i]),

delta [i] = (i +1) / 2 * delta,

The round function, F, is defined by, F (M, K [j, k], delta [i]) = ((M << 4) K [j]) ⊕ (M delta [i]) ⊕ ((M >> 5) K [k]) .

An important algorithm for a simple schedule; the 128-bit K key is divided into four blocks of 32 K = (K [0], K [1], K [2], K [3]). Buttons K [0] and K [1] are used in the undesired cycle and the K [2] and K [3] buttons are used even in rounds.

An embedded message is embedded in a video file and the Cipher text is considered to be a Decryption process.

Decryption is almost the same as the encryption process; in the process allowed for the use of cipher text as input in algorithm, but the lower K [i] buttons are used in reverse order.

The value at the center of the undo process is equal to the corresponding value of the hiding process the halves of the value have been changed.

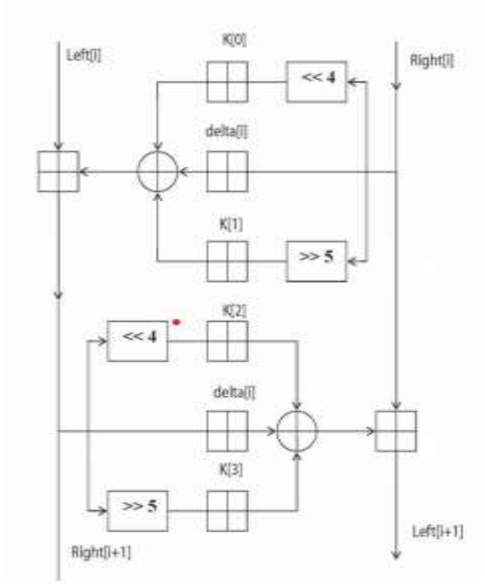

Fig-2: Diagram for round function

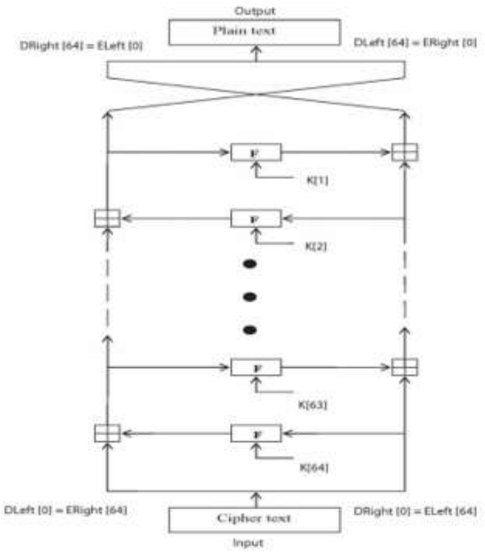

Fig-3: Diagram for decryption

4.2 LSB (LEAST SIGNIFICANT BYTES)

The least important least significant algorithm (LSB) was used in hiding the data in video file[4]. The advantage of LSB encoding is a very high watermark channel quality and low portability difficulty. An embedded watermark overlay is used to enter the LSB code method, increasing with increasing depth of LSB is used hiding information. In this way, modifications are made to for at least a few key pixels of the carrier file, thus encrypting information [8] Here each pixel has a location with three private data trees, one for each RGB price. Using a 24-bit image, it is possible to hide three pieces of data inside pixel value for each color using a 1024x768 pixel image; and it is possible to hide up to 2,359,296 pieces. The human eye cannot easily distinguish 21-bit color from 24-bit color [9]. Like for a simple example of LSB replacement, consider "hiding" I character 'A' for all of the following eight Bytes of host file:

(00100111 11101001 11001000)
(00100111 11001000 11101001)
(11001000 00100111 11101001)

Letter 'A' is represented in ASCII format as the binary string 10000011. The eight fragments can be "labeled" for each of the LSBs for the following eight bombs as follows (LSBs are orally labeled and strong):

(00100111 11101000 11001000)

(00100110 11001000 11101000)

(11001001 00100111 11101001).

With such small variations in the colors of the video image it can be very difficult for the human eye to detect the differences and thus give greater power to the system.[2]

5. LIMITATIONS

- Provides data storage in unsecured mode.
- Password leaks may also result in unavailable data access.
- Entrants will touch stegos.

6. CONCLUSIONS

There are various types of steganography techniques are available to hide the data in the video however LSB replacement is an easy option. The methodology is based on a survey of hiding message in video images (AVI) giving a robust and secure method of data transfer. The proposed embedded steganography has many benefits like user-friendliness, an easy and efficient way to encrypt a private message for extra security.

REFERENCES

[1] http://www.google.com

[2] http://www.wikipedia.com

[3] Christian Cachin.‖ An information-theoretic model for steganography‖. Lecture Notes in Computer Science, 1525:306.318, 1998.

[4] K. Steffy Jenifer , G. Yogaraj K. Rajalakshmi, LSB Approach for Video Steganography to Embed Images , vol-5, 2015.

[5] Hernández, Julio César; Isasi, Pedro; Ribagorda, Arturo. "An application of genetic algorithms to the cryptoanalysis of one round TEA". Proceedings of the 2002 Symposium on Artificial Intelligence and its Application, 2002.

[6] Kelsey, J., Schneier, B., & Wagner, D. "Related key cryptanalysis of 3-WAY", Biham-DES, CAST, DES-X, NewDES, RC2, and TEA. In Information and Communications Security—Proceedings of ICICS, 1334, 1997.

[7] Wheeler, D.J., & Needham, R.J. "TEA, a tiny encryption algorithm". In Fast Software Encryption – Proceedings of the 2nd International Workshop, 1008, 1998.

[8 Neeta Deshpande, Kamalapur Sneha, Daisy Jacobs, ―Implementation of LSB Steganography and Its Evaluation for various Bits Digital Information Management, 2006 1st International Conference on. 06/01/2007; DOI: 10.1109/ICDIM.2007.369349

[9] Mamta Juneja, Parvinder S. Sandhu, and Ekta Walia,"Application of LSB Based Steganographic Technique for 8-bit Color Images", WASET 2009

[10] HweeHwa Pang, Kian-Lee Tan, and Xuan Zhou. "Steganographic schemes for file system". IEEE Transactions on Knowledge and Data Engineering, 16(6):701.713, June 2004.