International Journal of Research (IJR) Vol-1, Issue-5, June 2014 **ISSN 2348-6848**

Fortified Multimedia Application

Akhand Pratap Singh

Abstract

The project entitled **Fortified Multimedia Application** is the application developed to embed an video file in another video signal. It is concerned with embedding information in an innocuous cover Speech in a secure and robust manner. This system makes the Files more secure by using the concepts Steganography and Cryptography.

Steganography, poor cousin of Cryptography is the art of hiding messages inside other messages such that the very existence of the message is unknown to third party. The goal of cryptography is to make data unreadable by a third party, the goal of Steganography is to hide the data from a third party Through the use of advanced computer software, authors of images and software can place a hidden trademark in their product, allowing them to keep a check on piracy. This is commonly known as watermarking. Hiding serial numbers or a set of characters that distinguishes an object from a similar object is known as finger printing. Together, these two are intended to fight piracy. The latter is used to detect copyright violators and the former is used to prosecute them. But these are only examples of the much wider field of Steganography.

The cover data should not be significantly degraded by the embedded data, and the embedded data should be as imperceptible as possible. The embedded data should be as immune as possible to modifications from intelligent attacks or anticipated manipulations. Thus it is necessary that the hidden message should be encrypted.

KEYWORDS:-Cryptography , Steganography , Encryption.

Introduction

Encryption of data plays a vital role in the real time environment to keep the data out of reach of unauthorized people, such that it is not altered and tampered and sending the in video format is most secured way to transfer the data through the network. The Video Stegnography is software, which tries to alter the originality of the file into some encrypted form and embed the file into an video file. The major task of the Video Stegnography is to provide the user the flexibility of passing the information implementing the encryption standards as per the specification and algorithms proposed and store the information in a form that is unreadable. The Application should have a reversal process as of which should be in a position to de embed the data file from video file and decrypt the data to its original format upon the proper request by the user. While the Encryption and Decryption is done the application should confirm the standards of authentication and authorization of the user.

The Entire application should strive to achieve a user friendly Graphical User Interface, which need to be in a self-learning mode for the end user. The System Should provide all the functional standards of proper navigation with in the environment, which makes it possible for the users to have a smooth flow while working under the environment. The Overall system should provide proper menu based navigation for easier navigation and operation. The Application should be designed in such a way that, as soon as it starts create a Buffer and associate this buffer to some homogeneous data environment, the application should ask the user for the Encryption Key details and should start its functionality upon the logistics that are provided with in this key. The key should be designed in such a way that it prevents the unauthorized persons from stealing the information at any point of time. This is some part of securing the data from third party people. And the other way of securing the data is using Steganography in which embedding the encrypted file in to a video file. If any one track that file they only see the video file not the data.

The application of De-embedding, Decryption should be a reverse process at the other end and should be translated only when the receiver of the data applies the proper reversal key. The Decryption process should have a log-based methodology that will take care of any errors that may be encountered while the system is under utilization and should record all those events, which are above the general standards of security.

This system basically uses the Tiny Encryption Algorithm to encrypt the passwords. This algorithm is a 64-bit block cipher with a variable length key. This algorithm has been used because it requires less memory. It uses only simple operations, therefore it is easy to implement.

Modules:

1) Encryption Module
2) Stegnography Module
3) Send File
4) Application Module

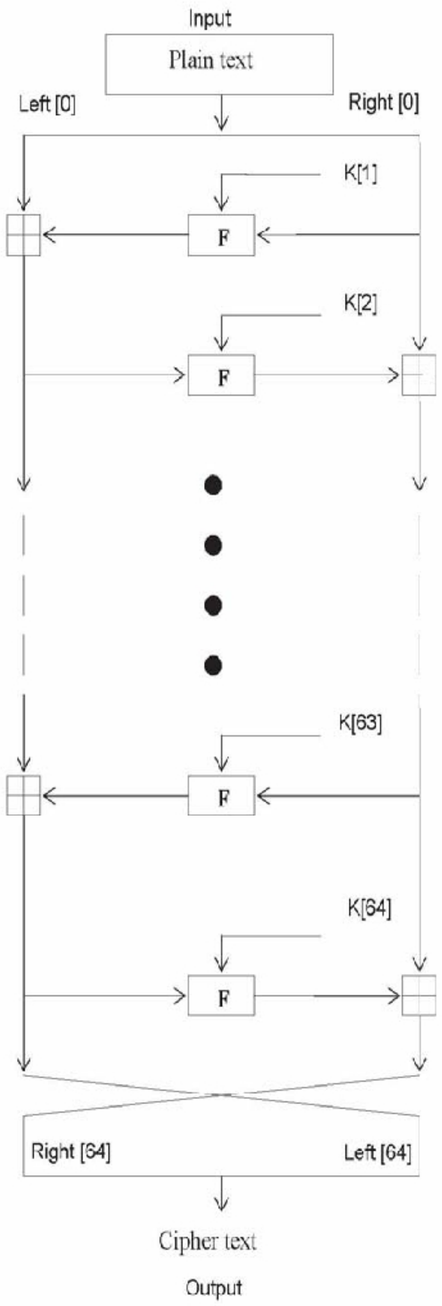
Material And Methods

1.Tiny Encryption Algorithm

The Tiny Encryption Algorithm (TEA) is a cryptographic algorithm designed to minimize memory footprint and maximize speed. It is a Feistel type cipher that uses operations from mixed (orthogonal) algebraic groups. This research presents the cryptanalysis of the Tiny Encryption Algorithm. In this research we inspected the most common methods in the cryptanalysis of a block cipher algorithm. TEA seems to be highly resistant to differential cryptanalysis, and achieves complete diffusion (where a one bit difference in the plaintext will cause approximately 32 bit differences in the cipher text) after only six rounds. Time performance on a modern desktop computer or workstation is very impressive.

As computer systems become more pervasive and complex, security is increasingly important. Cryptographic algorithms and protocols constitute the central component of systems that protect network transmissions and store data. The security of such systems greatly depends on the methods used to manage, establish, and distribute the keys employed by the cryptographic techniques. Even if a cryptographic algorithm is ideal in both theory and implementation, the strength of the algorithm will be rendered useless if the relevant keys are poorly managed.

The inputs to the encryption algorithm are a plaintext block and a key K .The plaintext is P = (Left[0], Right[0]) and the cipher text is C = (Left[64], Right[64]). The plaintext block is split into two halves, Left[0] and Right[0]. Each half is used to encrypt the other half over 64 rounds of processing and then combine to produce the cipher text block.

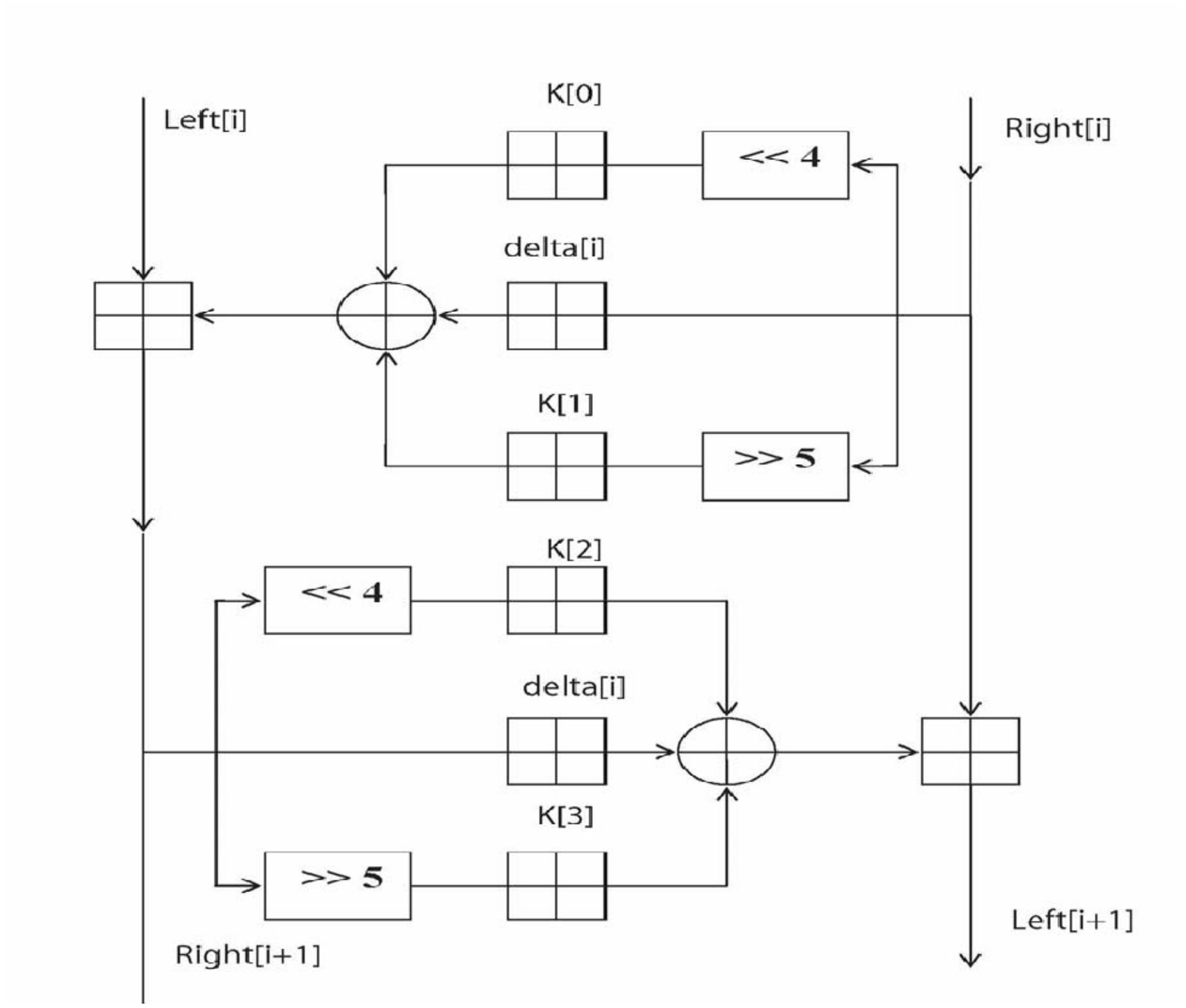


• Each round *i* has inputs Left[*i*-1] and Right[*i*-1], derived from the previous round, as well as a sub key K[*i*] derived from the 128 bit overall K.

• The sub keys K[*i*] are different from K and from each other.

• The constant delta $=(5^{1/2}-1)*2^{31}$ =9E3779B h , is derived from the golden number ratio to ensure that the sub keys are distinct and its precise value has no cryptographic significance.

• The round function differs slightly from a classical Fiestel cipher structure in that integer addition modulo 2^{32} is used instead of exclusive-or as the combining operator.



Above Figure presents the internal details of the *i*th cycle of TEA. The round function, F, consists of the key addition, bitwise XOR and left and right shift operation. We can describe the output (Left[*i* +1] , Right[*i* +1]) of the *i*th cycle of TEA with the input (Left[*i*] ,Right[*i*]) as follows

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

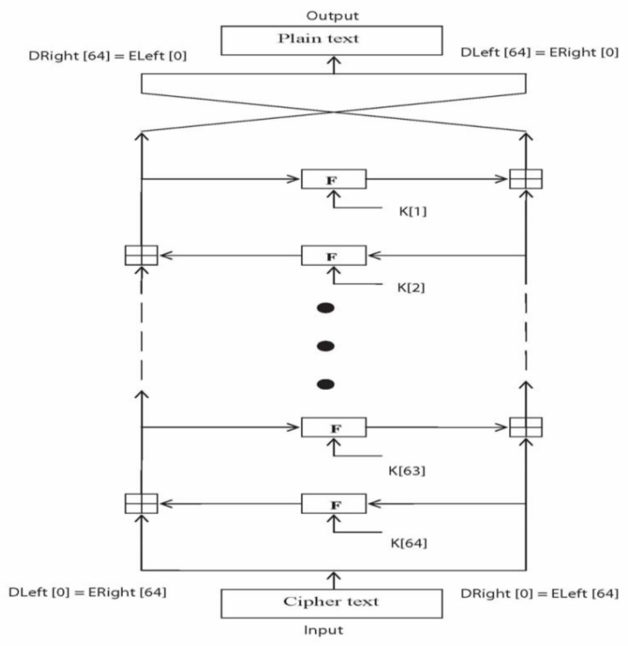
Right [*i* +1] = Right[*i*] F (Right[*i* +1], K [2, 3], delta[*i*]),

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*]).

The round function has the same general structure for each round but is parameterized by the round sub key K[*i*]. The key schedule algorithm is simple; the 128-bit key K is split into four 32-bit blocks K = (K[0], K[1], K[2], K[3]). The keys K[0] and K[1] are used in the odd rounds and the keys K[2] and K[3] are used in even rounds.



Decryption is essentially the same as the encryption process; in the decode routine the cipher text is used as input to the algorithm, but the sub keys K[*i*] are used in the reverse order.

Figure presents the structure of the TEA decryption routine. The intermediate value of the decryption process is equal to the corresponding value of the encryption process with the two halves of the value swapped. For example, if the output of the nth encryption round is

ELeft[*i*] || ERight[*i*] (ELeft[*i*] concatenated with ERight[*i*]).

Then the corresponding input to the (64-*i*)th decryption round is

DRight[*i*] || DLeft[*i*] (DRight[*i*] concatenated with DLeft[*i*]).

After the last iteration of the encryption process, the two halves of the output are swapped, so that the cipher text is ERight[64] || ELeft[64], the output of that round is the final cipher text C. Now this cipher text is used as the input to the decryption algorithm. The input to the first round is ERight[64] || ELeft[64], which is equal to the 32-bit swap of the output of the 64th round of the encryption process.

2.Steganography

Stegnography is art of hiding information in ways that prevent the detection of hidden messages. Stegnography derived from Greek, literally means "Covered Writing". It includes a vast array of secret communications methods that conceal the message's very existence. Theses methods are including invisible inks, microdots, character arrangement, digital signature, and covert channels and spread spectrum communications.

In this technology, the end user identifies an video file, which is going to act as the carrier of data. The data file is also selected and then to achieve greater speed of transmission the data file and video file are sent. Prior to this the data is embedded into the video and then sent. The image if hacked or interpreted by a third party user will open up in any video player but not displaying the data. This protects the data from being invisible and hence is secure during transmission. The user in the receiving end uses another piece of code to retrieve the data from the video file.

The module deals with identifying the hidden data in the video file. The module receives the video file that is then browsed to remove the associated data. The data is then removed from the video file.

Conclusion:

The entire project has been developed and deployed as per the requirements stated by the user, it is found to be bug free as per the testing standards that is implemented. Any specification-untraced errors will be concentrated in the coming versions, which are planned to be developed in near future. The system at present does not take care of lower level check constraints in accessing the file types in distributed environments, which is to be considered in the future up gradations.As per the present status the project developed is well equipped to handle the Central file system of an organization in a server and provide access to the users with various privileges as prescribed by the higher authorities in the password file.

Reference

[1] R.C. Merkle and M. Hellman, Hiding Information and Signatures in Trap Door Knapsacks, IEEE Trans. Inform. Theory, vol 24 1978,pp 525-530.

[2] R. L. Rivest, A Shamir and L. Adleman, A method for obtaining digital signatures and publickey cryptosystems, Communications of the Association for Computing Machinery, vol 21,no.2, pp 120-126.

[3] W. Diffie and M. E. Hellman, New direction in cryptography, IEEE Transactions on Information Theory, vol. IT- 22 ,no. 6,pp.644-654.K. Elissa,

[4] Souvik Bhattacharyya. andGautamSanyal. Study of secure Steganography model In Proceedings of International Conference on Advanced Computing and Communication Technologies (ICACCT-2008), Panipath , India 2008.

[5] G. Simmons, The prisoners problem and the subliminal channel, CRYPTO, 1983

[6] Souvik Bhattacharyya. andGautamSanyal. An Image Based Steganography Model for Promoting Global Cyber Security. In Proceedings of International Conference on Systemics, Cybernetics and Informatics, Hyderabad, India , 2009.

[7] K. Ahsan and D. Kundur, Practical Data hiding in TCP/IP, Proceedings of the workshop on Multimedia security at ACM Multimedia, 2002

[8] J.Silman, Steganography and Steganalysis: An Overview, SANS Institute, 2001

[9] Y.K. Lee and L.H. Chen, High capacity image steganographic model, Visual Image Signal Processing,147: 03, June 2000

[10] R. Krenn, Steganography and Steganalysis,