DIGITAL DATA SECURITY USING LEAST SIGNIFICANT BIT SHIFT ALGORITHM

KORASIKHA SRI DURGA MURALI [#1], **B.SURYANARAYANA MURTHY** [#2]

[#1] MCA Student, Master of Computer Applications,

D.N.R. College, P.G.Courses & Research Center, Bhimavaram, AP, India.

[#2] Associate Professor, Master of Computer Applications,

D.N.R. College, P.G.Courses & Research Center, Bhimavaram, AP, India.

ABSTRACT

Stegnography is the craftsmanship and study of composing concealed messages so that nobody, aside from the sender and mean beneficiary, associates the presence with the message, a type of security through lack of clarity. Stegnography is the plan of concealing the presence of mystery data by hiding it into another medium, for example, sound, and video or picture records. In Stegnography the shrouded document doesn't change the structure of the mystery message, however conceals it inside a spread picture with the goal that it can't be seen. This application allows you to insert or shroud significant or private messages or documents into wmv without influencing the nature of genuine information or records. It accomplishes this by utilizing the least critical pieces of these documents for inserting information which are not utilized by the Image watchers or Image editors. It permits you to install the messages or records in scrambled structure utilizing 32 piece DES calculation which implies that once encoded, the message or document could be recovered (or unscrambled) from a Master record simply subsequent to indicating the right secret phrase which was utilized at the hour of . On the off chance that you indicate encryption to be utilized, you'll need to determine a secret phrase which is at least 8 characters long.

Key Words:

Stegnography, Video, Audio, Image, Encryption, Decryption

I. INTRODUCTION

The idea of making sure about messages through Steganography and Cryptography has long history. Cryptography is engrossed with the insurance of the substance of a message or data. Steganography originates from Greek and signifies "secured composing". Steganography is a strategy used to conceal data inside pictures. Utilizing transcription, watermarks and copyrights can be put on a picture to ensure the privileges of its proprietor without changing the presence of the picture. Practically like enchantment, pictures, executable projects, and instant messages can cover up in pictures. The spread picture doesn't seem, by all accounts, to be adjusted. Individuals take a gander at the spread picture and never presume something is covered up. Your data is covered up on display.

The way to conceal information is to gadget a covering up (encryption) system that is extremely hard to switch (i.e., to locate the first information) without utilizing the decoding key. Symmetric-key calculations are a class of calculations for cryptography that utilization the equivalent cryptographic keys for both encryption of plaintext and unscrambling of ciphertext. In uneven key one key is utilized for encryption and another key is utilized for decoding. All the more explicitly this Paper manages the Symmetric key cryptography. For the Steganography we are utilizing the idea of bit move calculation.

II. LITERATURE SURVEY

In this section we will mainly discuss about the background work that is carried out in order to prove the performance of our proposed Method. Now let us discuss about them in detail

MOTIVATION

The Unix input/yield (I/O) framework follows a worldview for the most part alluded to as Open-Read-Write-Close. Before a client procedure can perform I/O activities, it calls Open to indicate and acquire consents for the record or gadget to be utilized. When an item has been opened, the client procedure makes at least one calls to Read or Write information. Peruse peruses information from the article and moves it to the client procedure, while Write moves information from the client procedure to the item. After all exchange tasks are finished, the client procedure considers Close to illuminate the working framework that it has wrapped up that object.

At the point when offices for Inter-process Communication (IPC) and systems administration were added to UNIX, the thought was to make the interface to IPC like that of record I/O. In UNIX, a procedure has a lot of I/O descriptors that one peruses from and writes to. These descriptors may allude to documents, gadgets, or correspondence channels (attachments). The lifetime of a descriptor is comprised of three stages: creation (open attachment), perusing and composing (get and send to attachment), and annihilation (close attachment).

The IPC interface in BSD-like renditions of Unix is executed as a layer over the system TCP and UDP conventions. Message goals are indicated as attachment addresses; every attachment address is a correspondence identifier that comprises of a port number and an Internet address. The IPC activities depend on attachment sets, one having a place with a correspondence procedure. IPC is finished by trading a few information through communicating that information in a message between an attachment in one procedure and another attachment in another procedure. At the point when messages are sent, the messages are lined at the sending attachment until the hidden system convention has communicated them. At the point when they show up, the messages are lined at the getting attachment until the accepting procedure makes the important calls to get them.

TCP/IP And UDP/IP Communications

There are two communication protocols that one can use for socket programming: datagram communication and stream communication.

1) Datagram Communication:

The datagram communication protocol, known as UDP (user datagram protocol), is a connectionless protocol, meaning that each time you send datagrams, you also need to send the local socket descriptor and the receiving socket's address. As you can tell, additional data must be sent each time a communication is made.

2) Stream Communication

The stream communication protocol is known as TCP (transfer control protocol). Unlike UDP, TCP is a connection-oriented protocol. In order to do communication over the TCP protocol, a connection must first be established between the pair of sockets. While one of the sockets listens for a connection request (server), the other asks for a connection (client). Once two sockets have been connected, they can be used to transmit data in both (or either one of the) directions.

Now, you might ask what protocol you should use -- UDP or TCP? This depends on the client/server application you are writing. The following discussion shows the differences between the UDP and TCP protocols; this might help you decide which protocol you should use.

In UDP, as you have read above, every time you send a datagram, you have to send the local descriptor and the socket address of the receiving socket along with it. Since TCP is a connection-oriented protocol, on the other hand, a connection must be established before communications between the pair of sockets start. So there is a connection setup time in TCP.

In UDP, there is a size limit of 64 kilobytes on datagrams you can send to a specified location, while in TCP there is no limit. Once a connection is established, the pair of sockets behaves like streams: All available data are read immediately in the same order in which they are received.

UDP is an unreliable protocol -- there is no guarantee that the datagrams you have sent will be received in the same order by the receiving socket. On the other hand, TCP is a reliable protocol; it is guaranteed that the packets you send will be received in the order in which they were sent.

In short, TCP is useful for implementing network services -- such as remote login (rlogin, telnet) and file transfer (FTP) -- which require data of indefinite length to be transferred. UDP is less complex and incurs fewer overheads. It is often used in implementing client/server applications in distributed systems built over local area networks.

III. EXISTING METHODOLOGY

In the current framework we used to have security of advanced information by utilizing Image Stegnography, where a picture can be inserted inside a picture of same organization. It can't insert on other type of information. Similarly we use to have Audio Stegnography just as Video Stegnography procedures where there are likewise utilized for concealing one type of information into other type of information of same kind.

LIMITATIONS OF THE EXISTING METHODOLOGY

Coming up next are the impediment of existing framework. They are as per the following:

1) Existing framework flopped secluded from everything one type of information in side another type of information of various types. (I.e. Picture inside a sound or video) and the other way around.

2) Existing framework flopped in accomplishing the guideline of blended stegnography.

3) In the current stegnography it’s absolutely impossible of coordinating cryptography additionally so as to give security.

IV. PROPOSED METHODOLOGY

The framework proposes a calculation that expands the security of the shrouded information and improve it for JPEG assaults. In the initial segment of this paper, we propose a calculation is utilized for installing one type of information inside other type of information of same sort or various sorts. In the second step we additionally has an office of giving security to the transporter record by giving a secret phrase for the information. The legitimate client who substitute the secret phrase can open the information which is covered up inside the transporter record, the others can’t ready to see the document which is covered up inside a transporter record.

ADVANTAGES OF THE PROPOSED SYSTEM

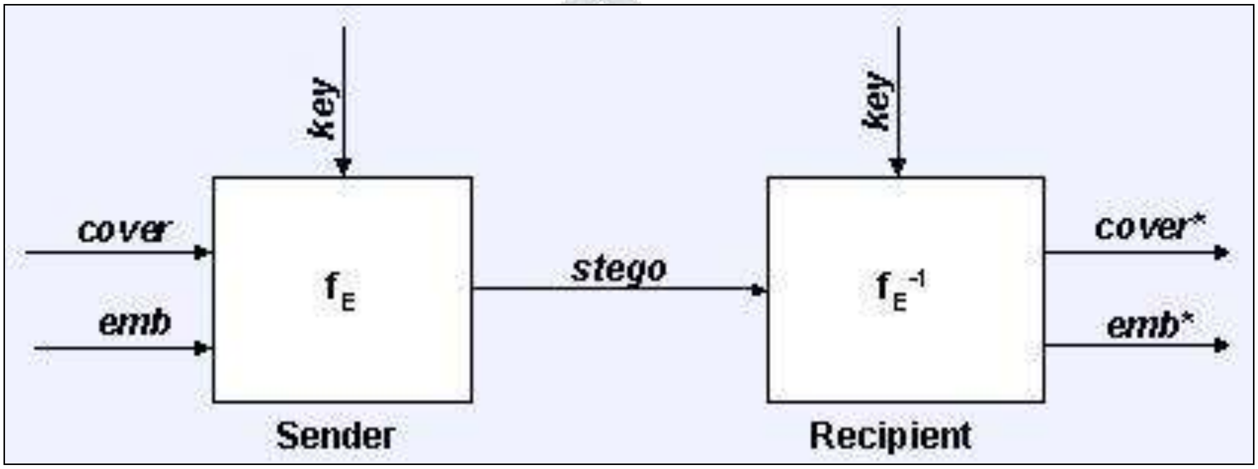
The following are the advantages of the proposed system. They are as follows:

Image....within….Image
Image...within…Video
Image…within…Audio
Text…Within…Audio

Audio....within….Audio
Audio....within….Image
Audio....within….Video
Text….Within….Video

Video....within….Video
Video....within….Image
Video....within….Audio
Text…..Within…Image

V. PROPOSED APPROACH



f_E	**:**	stegnographic function "embedding"
f_E^{-1}	**:**	steganographic function "extracting or De-Embedding"
cover	**:**	cover data in which ***emb*** will be hidden
emb	**:**	message to be hidden
key	:	parameter of f_E
stego	**:**	cover data with the hidden message

The following are the step by step procedure for the this proposed application. This is as follows:

STEP 1:

Initially the sender will try to choose the cover image and then he try to add the sensitive data inside the cover image by using a key parameter.

STEP 2:

Now he tries to use Key parameter for embedding the secret data into the cover image.

STEP 3:

The file will be send to the receiver node as a carrier file.

STEP 4:

The receiver will receive the carrier file and try to enter the password correctly to extract the hidden data.

STEP 5:

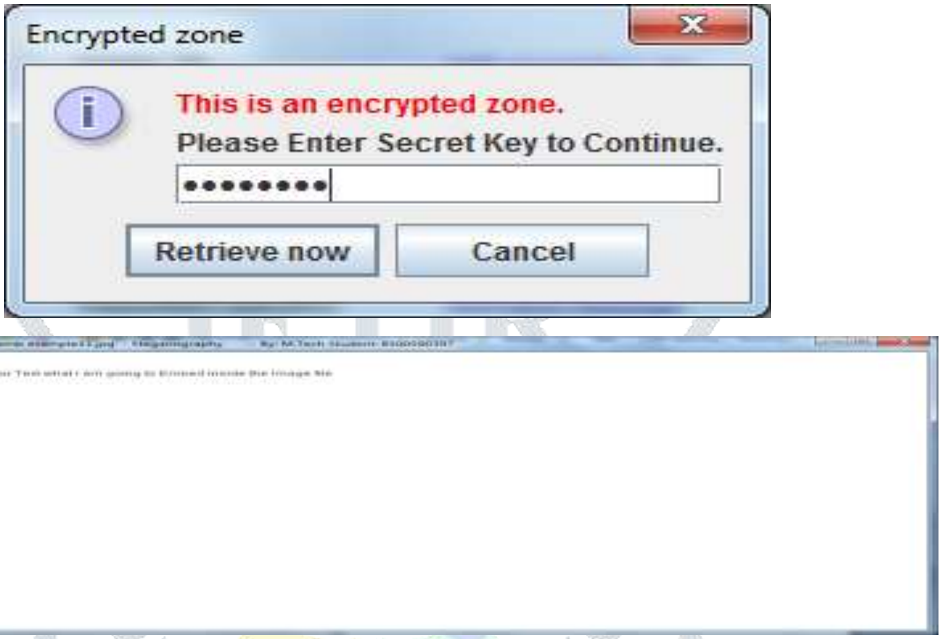
If the key is correctly substituted the data can be extracted from that carrier file.

STEP 6

The receiver can able to view the hidden data from the carrier file by substituting valid password.

VI. EXPERIMENTAL REPORTS

1) Extraction of Hidden Output by entering the correct password



Encrypted zone

This is an encrypted zone.
Please Enter Secret Key to Continue.

●●●●●●●●●

Retrieved message from the 'emb example11.jpg' - Steganography By: M.Tech Student: [illegible]

Hello M Tech Students

This is the Sample File or Text what I am going to Embed inside the image file

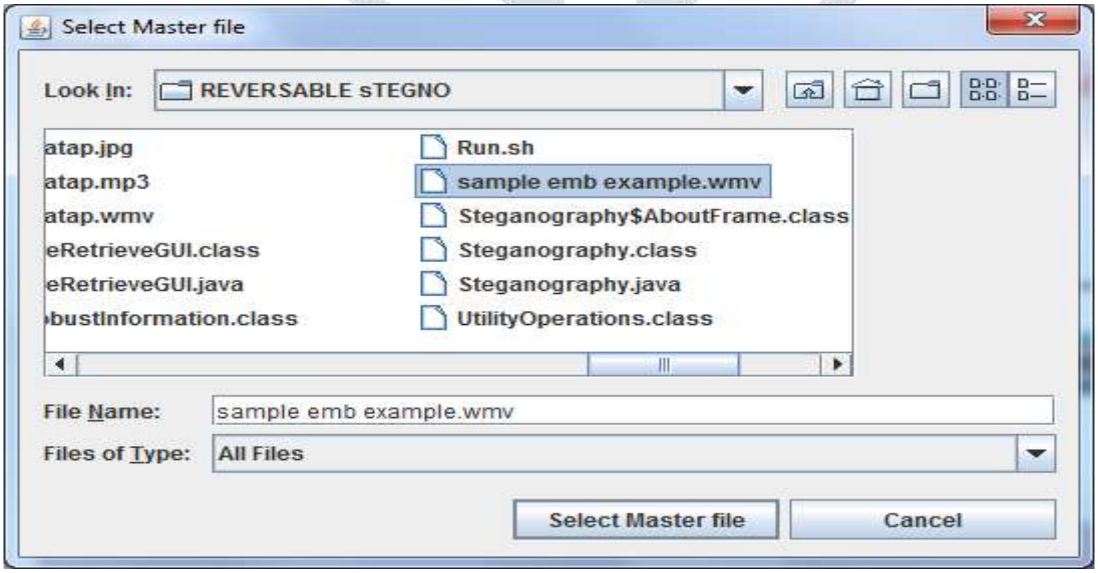
Regards

M.Tech Students
Visakhapatnam

Figure Represents the File retrieved from that carrier file

Explanation:

In the same way, if we enter correct password for the emb video file also we can get the hidden data. This is as follows:



Select Master file

Look In: REVERSABLE sTEGNO

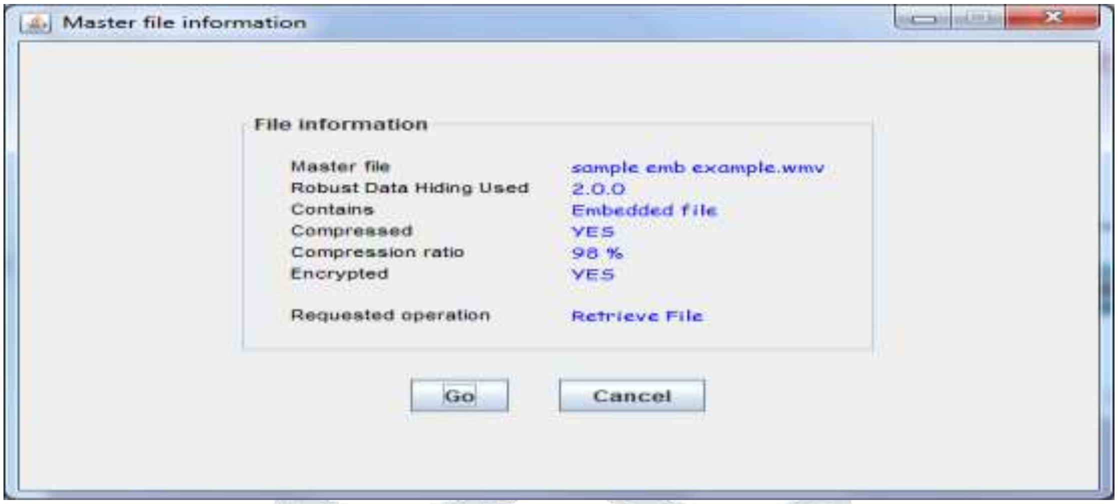
```
atap.jpg                 Run.sh
atap.mp3                 sample emb example.wmv
atap.wmv                 Steganography$AboutFrame.class
eRetrieveGUI.class       Steganography.class
eRetrieveGUI.java        Steganography.java
bustInformation.class    UtilityOperations.class
```

File Name: sample emb example.wmv

Files of Type: All Files

Figure Represents the Browse Carrier File

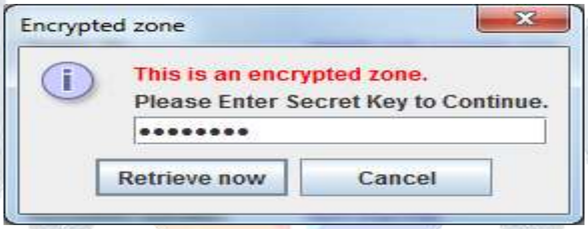
From the above window we can clearly find out that we have choosen the embedded file as input by the receiver ,in order to extract the hidden data I.e Image from the embedded file.



File information

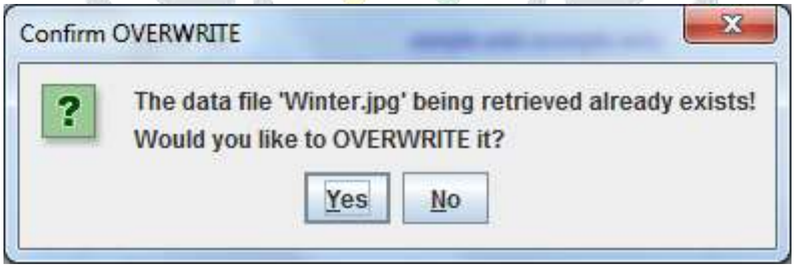
Master file	sample emb example.wmv
Robust Data Hiding Used	2.0.0
Contains	Embedded file
Compressed	YES
Compression ratio	98 %
Encrypted	YES
Requested operation	Retrieve File

The above window clearly tells that the file which is chosen as the master file contains some hidden information and also it is encrypted.So if we click on go button to view the data ,we will get the below Pop Up Window to enter the password.



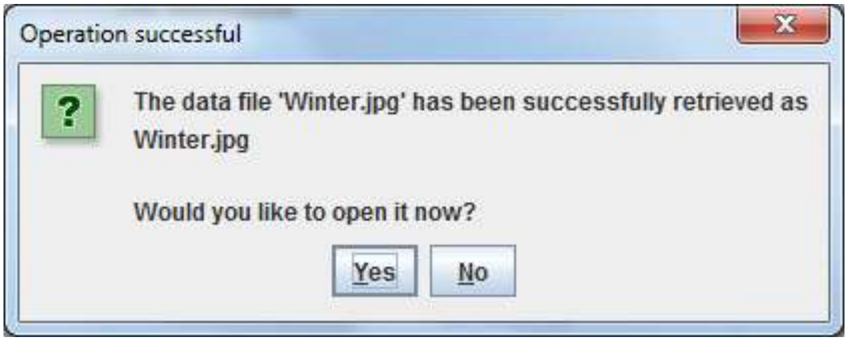
This is an encrypted zone.
Please Enter Secret Key to Continue.
●●●●●●●●●

In the above window if we enter the correct password, then it will be decrypted and the hidden data can be viewed. Or else it will show error like password not correct.



The data file 'Winter.jpg' being retrieved already exists!
Would you like to OVERWRITE it?

From the above confirm dialog box,we can get an idea that ,password is correct and it is prompting the receiver to overwrite the file to open as it is available in same folder.



The data file 'Winter.jpg' has been successfully retrieved as Winter.jpg

Would you like to open it now?

VII. Conclusion

In this paper, on the off chance that one had the option to shroud the message in the video record in such a way, that there would be no discernible changes in the sound document after the message addition. Simultaneously, if the message that will be covered up were encoded, the degree of security would be raised to a serious palatable level. This application allows you to implant or conceal significant or private messages or records into wmv and mpg without influencing the nature of real information or documents. It accomplishes this by utilizing the least huge pieces of these records for installing information which are not utilized by the Image watchers or Image editors.

VIII. References

[1] The Complete Reference, by Patrick Naughton, and Herbert Schildt

[2] JAVA Swings Matthew Robinson

[3] O'reillys JAVA Swings by Robert Eckstein, Mare Loy & Dave Wood

[4] Referred to http://www.tropsoft.com/strongenc/rijndael.htm

[5] Referred to http://www.bitpipe.com/tlist/Password-Authentication-Protocol.html

[6] Object-Oriented Software Engineering by Jacobson, Rambaugh, Booch

[7] Digital Watermarking and Stegnography, Second Edition; Cox, Miller, Bloom, Fridrich, and Kalker. Morgan Kaufmann. 2008

[8] A Handbook of Applied Cryptography by Alfred J. Menezes,Paul C.

[9] A Course in Number Theory and Cryptograph Neal Koblitz, Springer 1987

[10] Modern Cryptography, Probabilistic Proofs and Pseudorandomness OdedGoldreich, Springer-Verlag 1998

[11] Applied Cryptography: Protocols, Algorithms, and Source Code in C, by Bruce Schneier