iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 2017 **Issue:** onferende **Month of publication:** September 15, 2017

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor:6.887
Volume 5 Issue IX, September 2017- Available at www.ijraset.com

Advanced Java Based Information Security Algorithm by Embedding Data INAN Image Using Digital Water Marking

Amol Baviskar[1], Jaypal Baviskar[2], Devrshi Pathak[3], Sanket Kasar[4]
[1,2]*Department Of EXTC Engineering Vishwatmak Om Gurudev College Of Engineering Aghai-Thane, India*
[3,4]*Department Of Electronics Engineering, Universal College Of Engineering Thane- Vasai, India*

***Abstract:** Since the technology is fast growing in a rapid a huge amount of information is being processed, modified, transmitted and need to be secured. This information or data can be in the form of text, image or audiovisual. Data size and the data security is major issue need to be handled at each and every point of time while processing it. Hence compression techniques are used to reduce the size of data in order enhance storage capacity and transmission rate along with this various encryption techniques are used to secured the information in order to avoid data violation. This paper proposed JAVA based novel approach to compress and encrypt the data by using concept of digital watermarking. In this a required text file gets selected that should be in the "*.txt" format then compressed by specially design compression algorithm. This obtained compression file then followed by data encryption in which the compressed text data gets embedded with an image and transmitted to the receiving end or to a client by establishing a path. At the receiving end decryption algorithms and decompression algorithm is applied in order to retrieve the original data. The main motto of this paper is explaining the implemented systems which successfully retrieve the required original data without any loss or error.*

I. INTRODUCTION

The proposed system is basically consists of specially designed dedicated algorithms i.e. compression algorithm, encryption algorithms, decryption algorithm and decompression algorithm also it includes GUI of main server and client. In this each and every user is having a unique login id and password. After login at server main GUI will be opened on the screen with various options explained in details in system implementation. First "file compressed" option is selected and desired text file is chosen. After compression the text file gets converted into a garbage value which is stored at appropriate location provided by the user. Next step is to select "file encrypt" on the window after selecting this option we need to load an image and garbage date file to embed in each other. The selected image and garbage data gets encrypted by using a technique of digital watermarking. Encrypted file followed by transmission but before transmitting a care should be taken is that "establish a path for transmission". At receiving end exactly reverse procedure is done and original text file is retrieved. Many such techniques of Digital Watermarking have been discussed in [1,2,3]

II. SYSTEM METHODOLOGY

A. Obtaining The Compress File: Main window consists of all the modules of the Project. The flow of process is as Compression, Encryption, File transfer, Receive file, Decryption and Decompression. First user id and password need to be entered. On main window click on the compress file option then desired text file is browsed,

III. SYSTEM BLOCK DIAGRAM

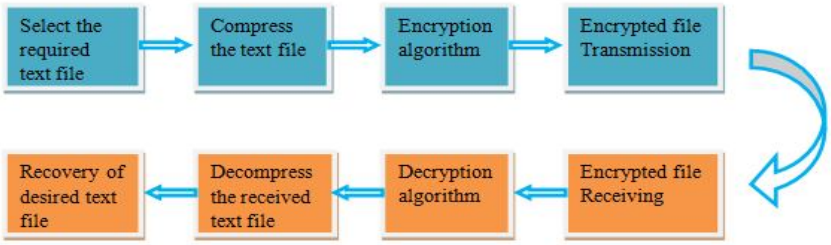


Figure 1 System block diagram

Selected and then message gets appeared that file is compress successfully. Verification of compressed file is done that is successfully stored in desired location or not.

A. File Encryption

After compression of the text file the date gets converted into garbage value. Then encryption is done by selecting an image and embedding above garbage file in selected image. Now encryption is done successfully.

B. Transmission of Encrypted image

Information at left hand corner of an encrypted image has been observed and then path connection is established using concept of socket programming. This concept established the connection between sender and receiver. Now send file to receiver.

C. Receiving of the Encrypted file

In another window file gets received. Then path and location is selected to stored encrypted received file. Then click on "start server button". It will show the status "running and waiting to receive file" after completion of process the file needs to be saved in .bmp format.

D. Decryption of the file

Select "File Decrypt" and load the received image. Also create a path to store the image and select the location then message will appeared "Decrypted file have been save successfully.

E. Decompression and retrieval of an original image

An image and garbage text file gets separated. For decompression garbage text file selected, path is created and decompression is done. The decompressed image is stored at selected location. Now the retrieved text file compared with original one and it has been observed that all the text information gets retrieved successfully.

IV. SYSTEM IMPLEMENTATION ALGORITHM

A. Compress the text file before embedding it to the Image.

B. The compression process:

1) Initialize the components.
2) Initialize public variables.
3) Open dialog box to browse and select a text file "*.txt" to be compressed.
4) Now specify path and extension where it is to be saved after compression.
5) For compression, first object of file stream is created. This creates a stream of text which after compilation is accessed to Read-only.
6) GzipStream is Stream Reader in library called to compress the file and save output.
7) After completion of the above steps, the process is flushed and closed i.e. afterrunning and saving, we close connection.
8) The file is compressed successfully.

C. After compression Encryption is to be done.

1) Initialize the components.
2) Initialize public variables.
3) Create function to browse and load the image. Image size is set to small decimal "kb", height and width to pixel.
4) Select the path where the image is to be saved i.e. "C:\\".
5) Click encrypt button on window for encryption.
6) If (EnImage_tbx.Txt==String.Empty || EnFile_tbx.Txt==String.Empty) then Error message is generated as "Encryption info. is incomplete! \n Please complete them first." OR
7) If (8*((height)*width/3*3)/3-1) <filesize+ FileNamesize) then Error message is generated as "File size is too large! \n Please use a larger image to hide the file."
8) If no error occurs Message: "Encrypting... Please wait"
9) The image is changed to Bitmap while encrypting. Bitmap image is converted to 8*8 matrix to compare each pixel value.
10) r/g/b values of each pixel are compared to the function to get Boolean values so as to spread the message overall with consistency without tampering the image.

D. The compressed and encrypted file is forwarded to the receiver using socket programming concept to create connection between sender and receiver.

1) First we check whether the Client is idle and specify its IP Address for connection within common port "5656."
2) Now select the file to be sent (which is an image file embedded with text).
3) If the file size is greater than 850 kb then error generates "File size is more than 850 kb. Please try with small file." Else buffering continues.
4) Then server is connected and file is sent. The server gets Disconnected only after sending the whole file.
5) If the connection is not proper or is refused by other m/c, then error occurs and file sending fails.
6) To transfer the file from the server to client, first the client has to specify the receiving path.
7) Object is created to access the server.
8) Now for file transfer IPEndpoint is to be specified (IPEndpoint,5656).
9) The socket gets connected and client accepts to receive the sent file.
10) The received file is now compressed and encrypted for secure delivery of the message.
11) Now to retrieve the original text message from the received image file, Decryption and Decompression are done which are of encryption and compression respectively.
12) Decryption:
13) 7.1.- 7.4.Steps are similar to encryption.
14) 7.5. Click the decrypt button on the window.
15) 7.6. If (DeSavefile_tbx.Txt==String.Empty || Deload Image_tbx.Txt==String.Empty) then error as" Text boxes must not be empty!" is generated. OR If (System.To.File.Exists (DeloadImage_tbx.Txt==False)) then error is generated as "select image file."
16) 7.7. If the process is proper then successful decryption happens.
17) 7.8. From each pixel bitmap is decrypted from each row of the image by comparing r/g/b values as r=pixel.R; g=pixel.G; b=pixel.B;
18) 7.9. The values are compared as 0,1 (boolean values) and assigned to each pixel.
19) E.g.: t[0]=rb[layer]; t[1]=gb[layer];t[2]=bb[layer];…so on.
20) If true then boolean values are converted to byte values to load true image and separate text message.
21) Decompression process is exactly opposite to that of Compression.
22) - 8.4.: similar steps as of compression to select and load.
23) Gzipstream is StreamReader in library called to decompress the file and save the output message separated from image file.
24) After completion the process is flushed and closed to disconnect the connection.

V. RESULT ANALYSIS

A. The Main Window displays all the processes together consisting of encryption, decryption, compression, decompression, file transfer (send and receive) which looks as follows:

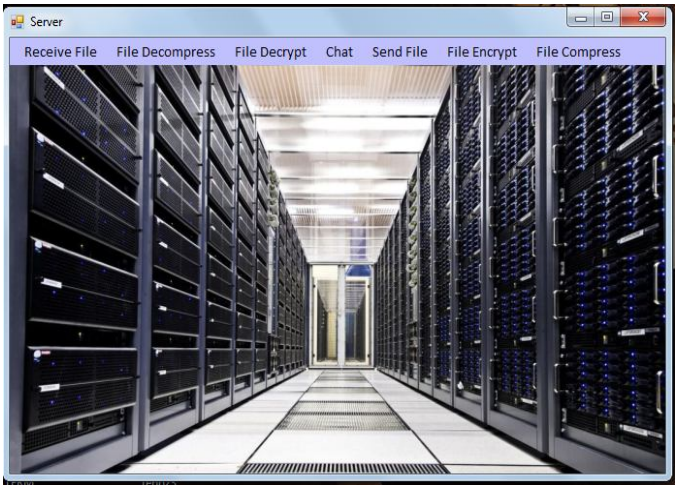


Figure 2 Main Window

B. The File Compression window helps to browse and select the file to be compressed and saves it to the specified path described by the user after compression

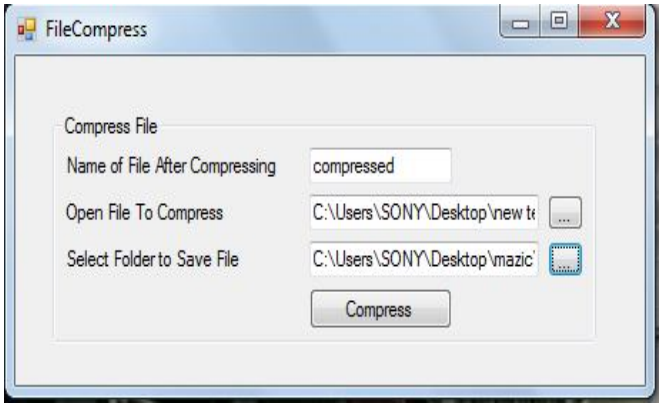


Figure 3 Text File Compressions

C. After the above step of selecting the file and undergoing compression the following window is displayed showing the successful completion of compression method.

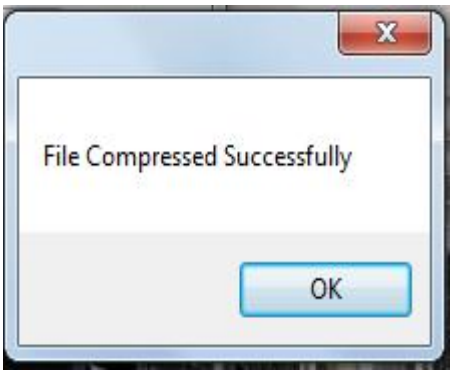


Figure 4 File Compressed Successfully

D. The following window displays garbage value of the text file being compressed. This garbage value helps the sender to send his message secretly without any intruder to damage the content.

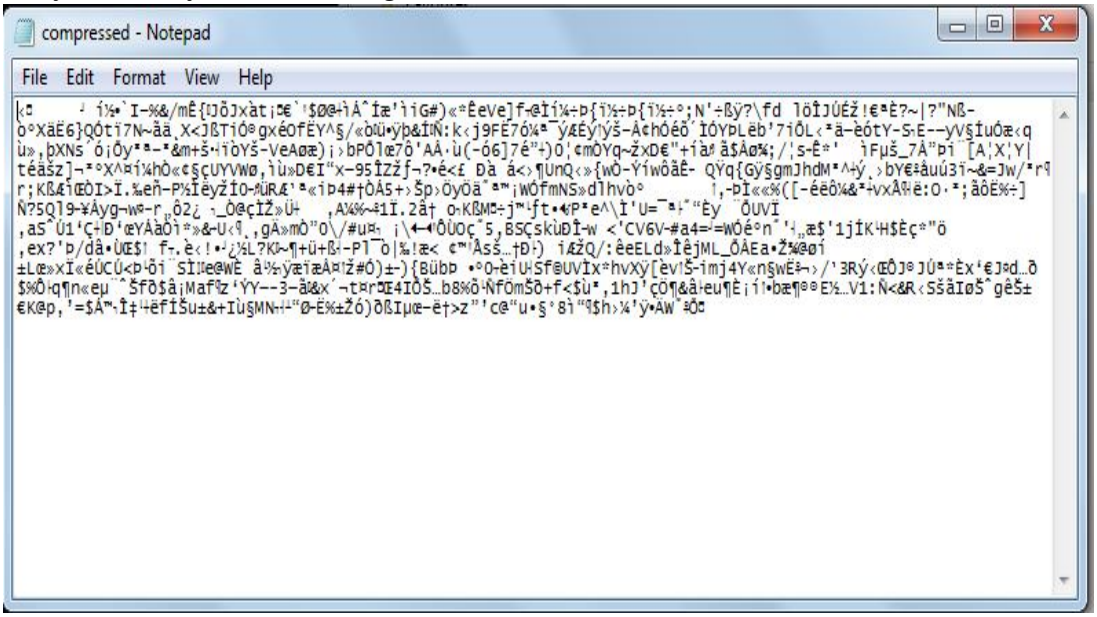


Figure 5 Compressed file

E. The next step involves Encryption of the compressed file within image as follows. The image is browsed and selected. Now the compressed text file i.e. to be encrypted are selected and image information is to be remembered.

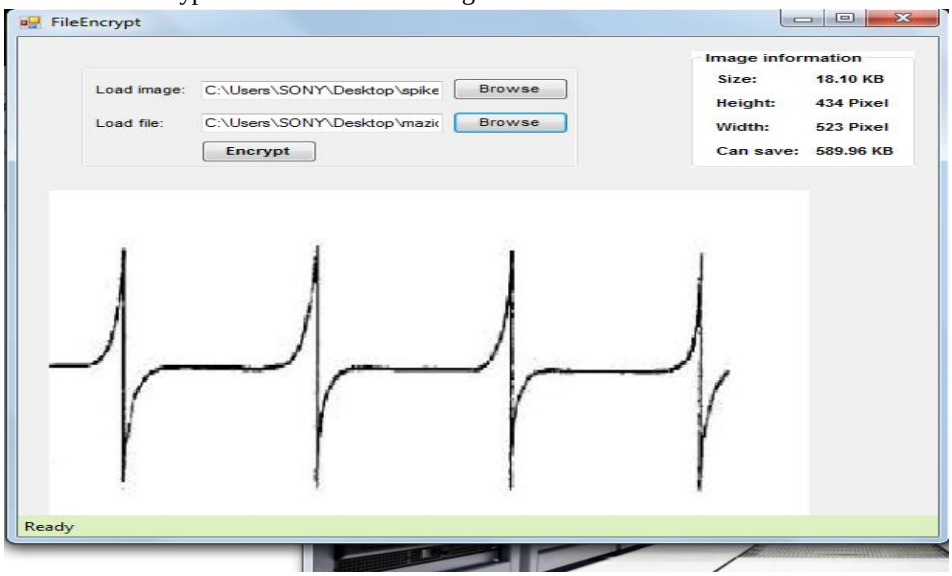


Figure 6 Image Encryption

F. Now if no error occurs i.e. proper text file and image file of required size are selected so that encryption is successful the following window is displayed.

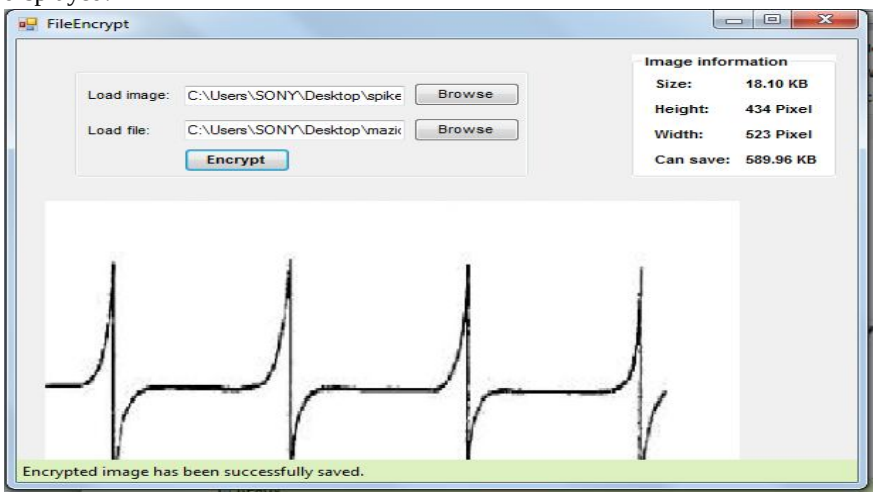


Figure 7 Successful Encryption

G. The encrypted file is now ready to be forwarded at the other end (receiver). Hence the following window shows the initialization of the sending process.

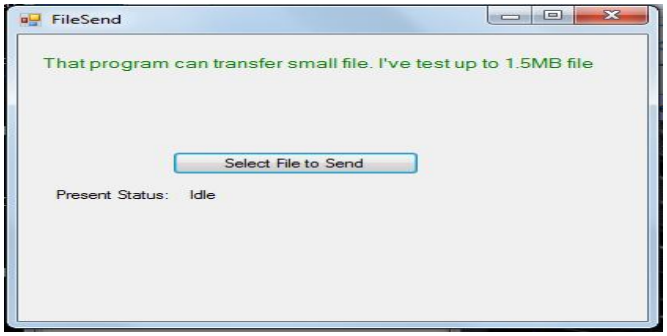


Figure 8 File Sending

H. Next step describes the connection to be initiated between the sender and the receiver which shows "start server" and "select the receiving path" for where the file is to be reached after transfer.

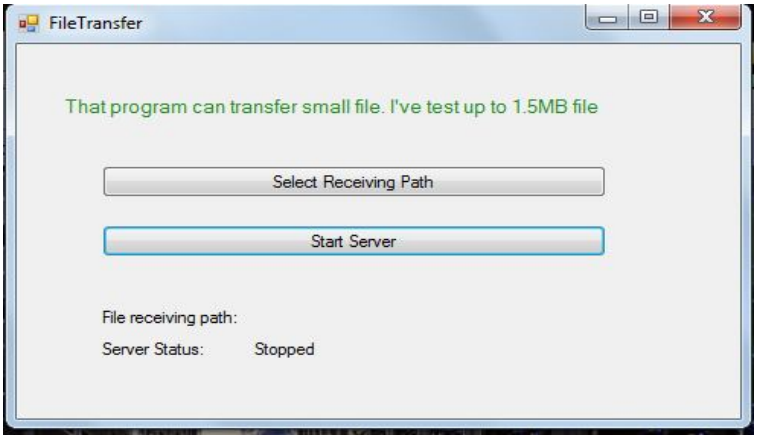


Figure 9 Server Connection

I. This step shows after connection the path selected "C:\Users\Sony\Desktop" will save the file sent to the receiver at this specified path successfully and stop the server only after the file is fully transferred.

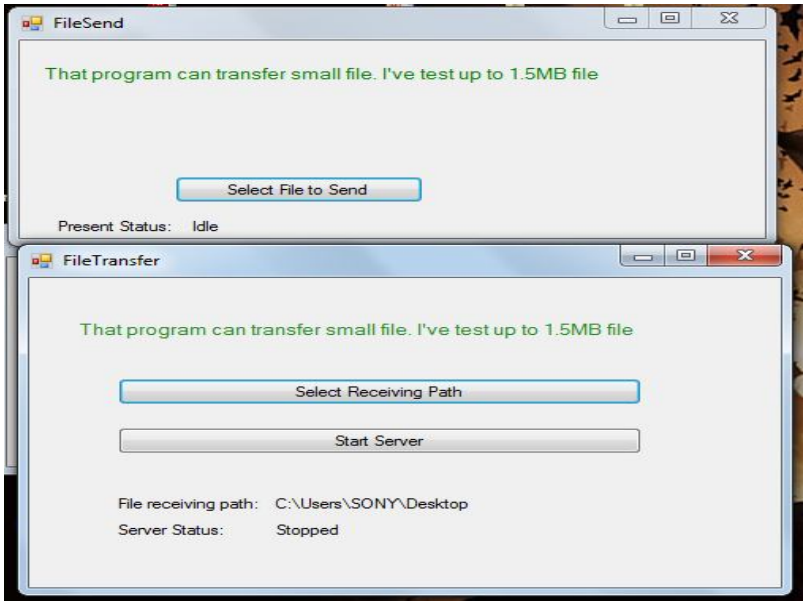
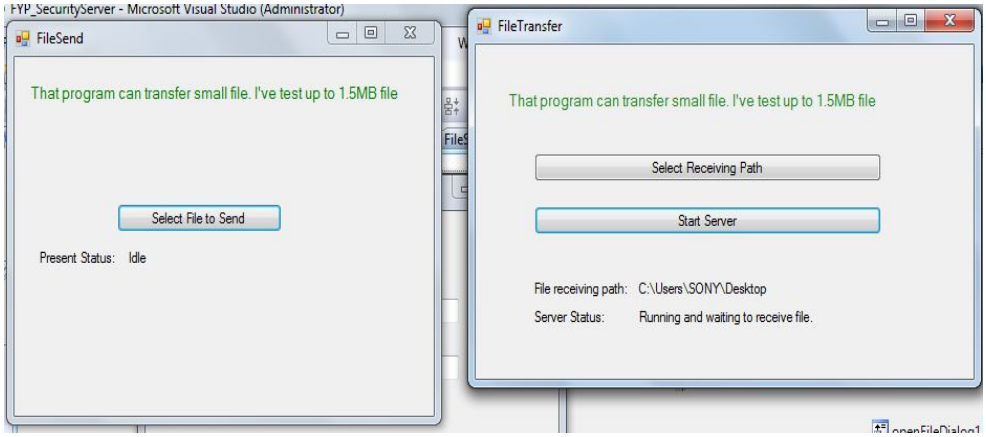


Figure 10 Path Selection

J. After the file path is selected the server is to be started to receive the sent file as follows. And the file (encrypted) is received.



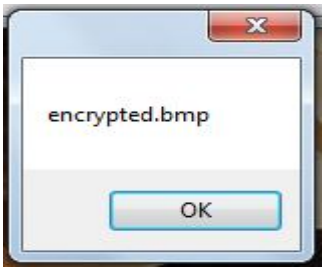


Fig11 Receiving file.

K. After receiving the encrypted file the server status changes to received and saved file and then only the server gets disconnected (stopped).

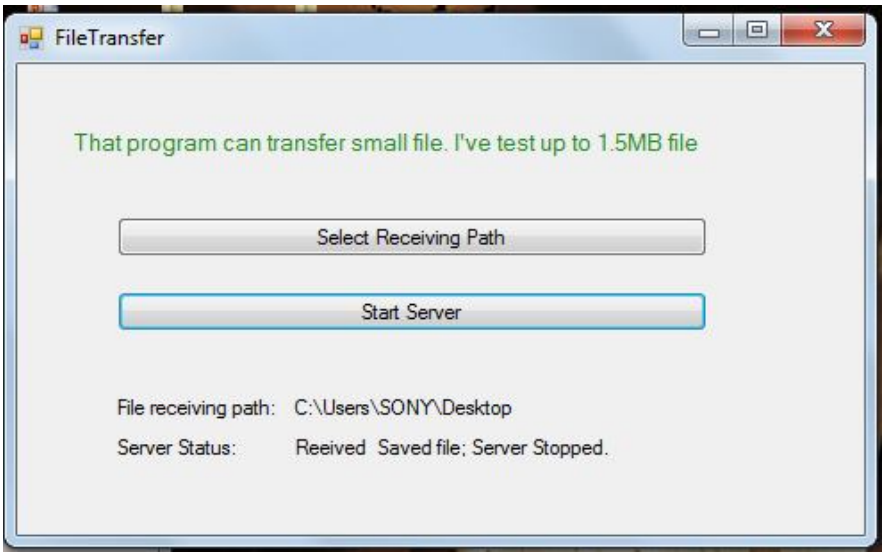


Figure12. End of the session.

L. Now the file received by the receiver at the other end is to be extracted with the first process File Decryption. The window below describes the loaded image sent by the sender that is to be decrypted and then the location where it is to be saved after it is decrypted. Here the image information is again to be noted and compared with the encrypted one to see the difference.

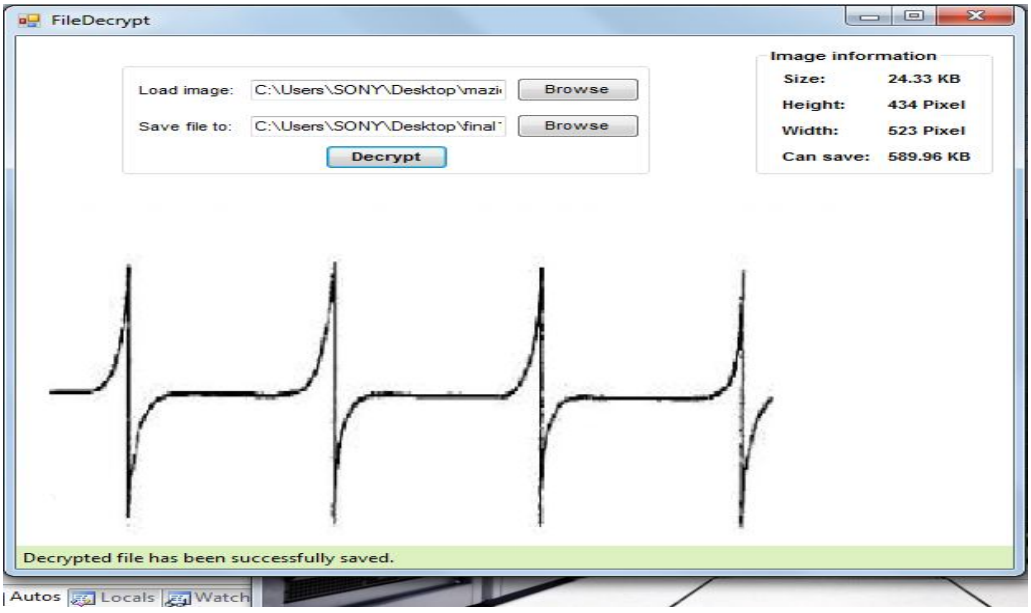


Figure 13 Successful Decryption

M. After the decryption happens successfully now the last step to retrieve the original text file or message is Decompression. For decompression also the decrypted file is to be selected and specify the path where the receiver wants it to be saved. Once decompression happens successfully a message box showing successful process displayed.

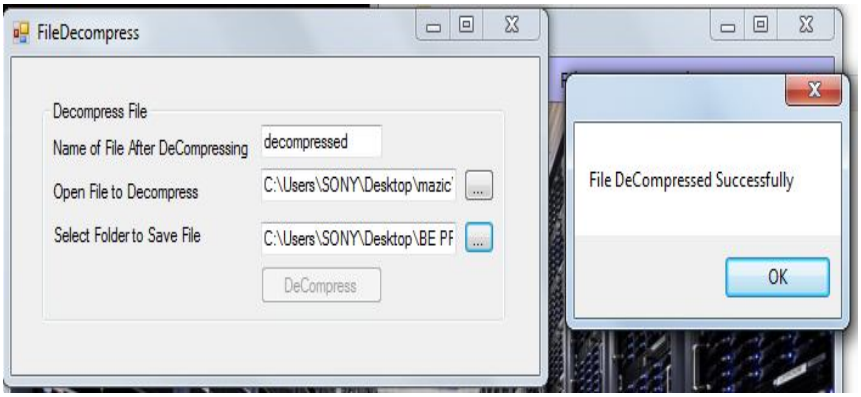


Figure 14 Successful Decompression

N. Finally after performing all the above processes original text file is retrieved securely and successfully without letting any intruder to damage the contents or message. The following window displays the original text.

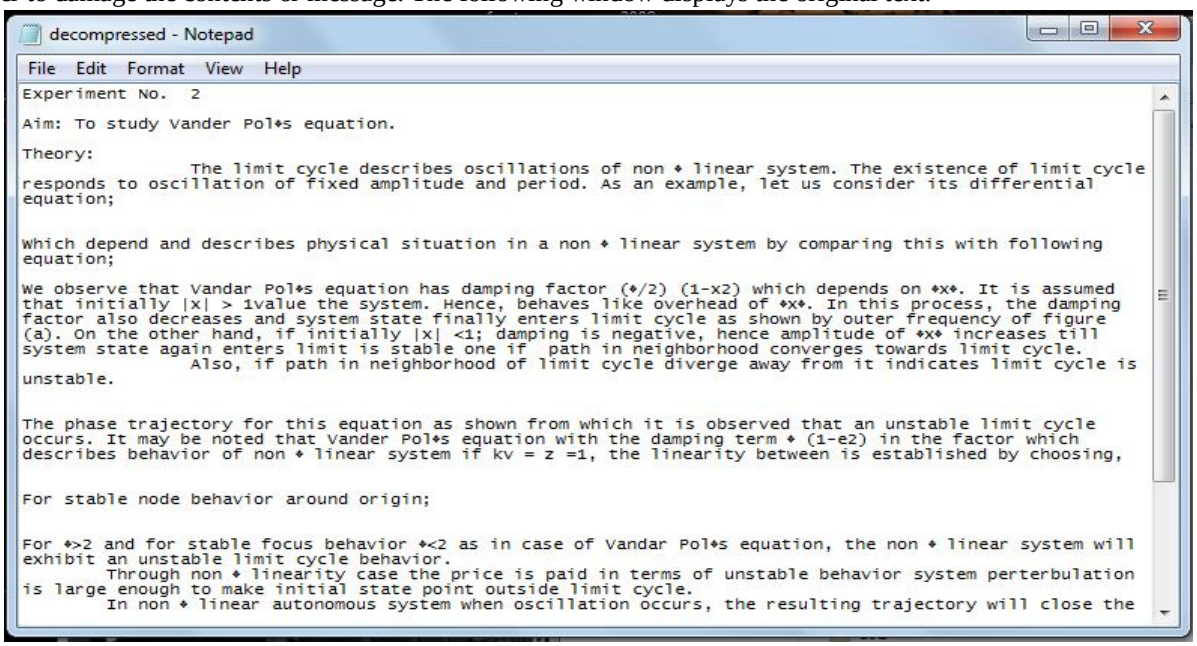


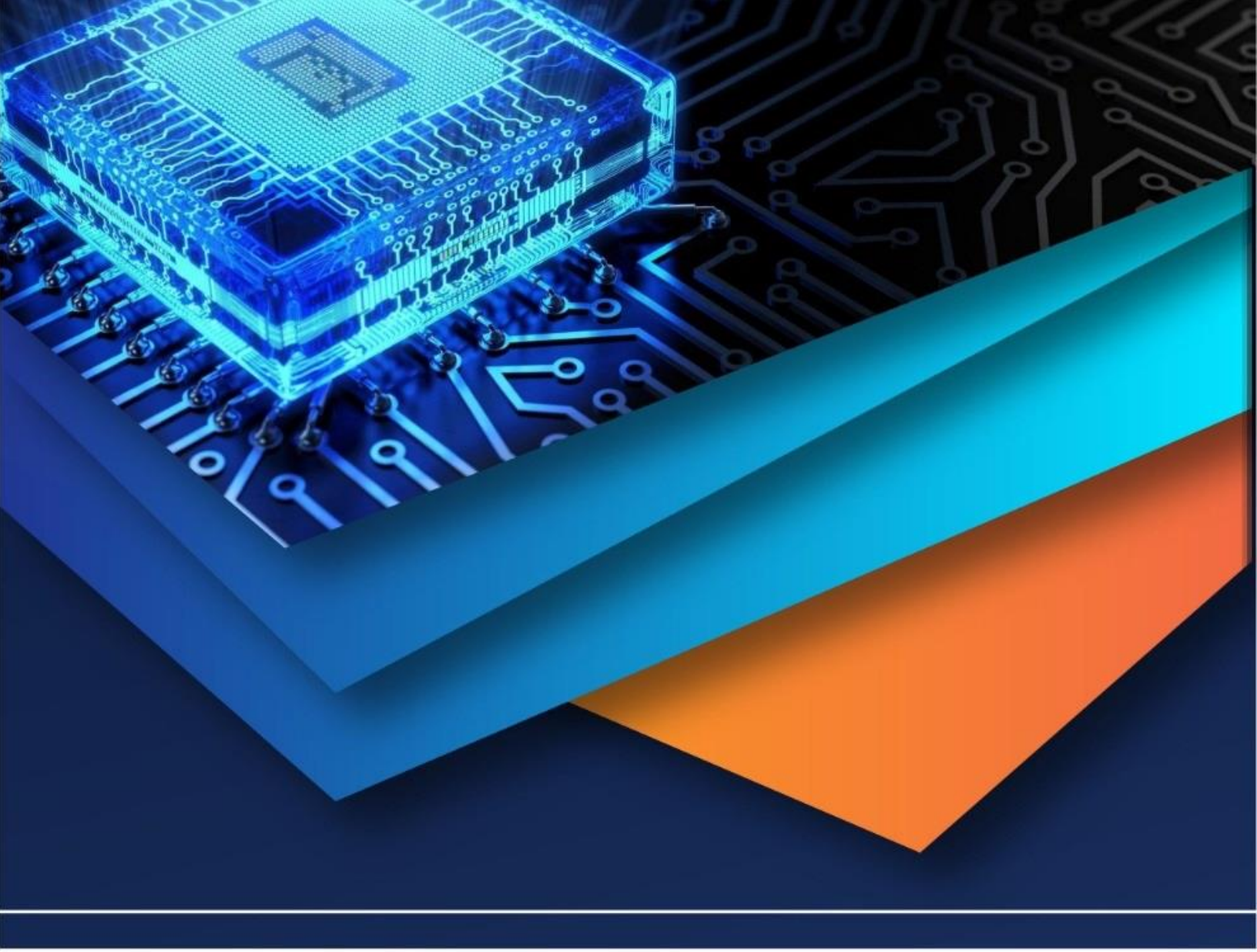
Figure 15 Final Extracted File

VI. CONCLUSION

The JAVA based novel approach to compress and encrypt the data by using concept of digital watermarking has been described in this paper. The algorithm is explained in detail and results are illustrated with help of screenshots captured at different stages of algorithm.

REFERENCES

[1] Munesh Chandra, ShikhaPandel, Rama Chaudhary, ―Digital watermarking technique for protecting digital images‖ Third IEEE International Conference on Computer and Information Science and Technology (ICCSIT 2010), pp.226-233.
[2] M. A. Dorairangaswamy and B. Padhmavathi, "An effective blind watermarking scheme for protecting rightful ownership of digital images," TENCON 2009 - 2009 IEEE Region 10 Conference, Singapore, 2009, pp. 1-6
[3] S Rawat, Keshav& S Tomar, Dheerendra. (2010). DIGITAL WATERMARKING SCHEMES FOR AUTHORIZATION AGAINST COPYING OR PIRACY OF COLOR IMAGES. Indian Journal of Computer Science and Engineering. 1.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY