AN APPLICATION USING ATTACK PREVENTION METHODS FOR SECURE DATA TRANSMISSION

Elshad KARIMOV[1*], Ali GUNES[1] and Adem OZYAVAS[1]

[1]Computer Engineering Departmant, Istanbul Aydin University, Turkey

***Corresponding Author:**
Email: karimovelshad@gmail.com

Abstract

The rapid development of information systems and means of communication, has led to increase in the importance of developing new techniques to ensure the secrecy of information. To minimize important information being compromised by unauthorized persons, steganographic and cryptographic techniques have been developed.

The purpose of this study, is to take measures against passive attacks that can be done by possible third parties, during data transmission period. The first encrypted message during communication, has been encrypted by using the algorithm called LUCIFER. To protect the encrypted message to be found out by unauthorized people the existence of the encrypted data has been hidden by using the methods of digital steganography. The encrypted message and the decryption key was buried into the video frame and sent to the receiver and with this way unauthorized people did not suspect anything.

Keywords: Cryptography, symmetric cryptography, steganography, image steganography

1. Introduction

One of the most important trends of modern society, is that information technology has rapidly penetrated in all fields of human activity. Thanks to the technology, there are many conveniences in every aspect of human life. Aside from the convenience that is provided, confidentiality of information has been always one of the values about which people care most. Therefore, throughout human history, to maintain the confidentiality of the information people have used different methods.

In the introduction section of the article, the importance of confidentiality of information is highlighted, the parts of the study are summarized, and the examples about previous research are presented.

In the second section, cryptographic and steganographic methods are discussed that are used in the project.

In the third part, the detailed information is provided about the application which has been developed by using steganographic and cryptographic methods and user interface design, and its working process is explained.

In the fourth part, conclusions and recommendations are presented.

Literature Survey

This section shows examples of the previous works about using steganographic techniques to hide data into digital files. Some of these examples are listed below.

In their study published in 2012, Rahul Jain and Naresh Kumar (Jain and Kumar, 2012) proposed a data-hiding draft by using the method of image steganography and compression technique.

A study was published in 2012, by S. Hemalatha and colleagues (Hemalatha and dig. 2004), in which they suggested a method of image steganography based on Integer Wavelet Transform.

In a study by Nosrati and colleagues (Nosrati and others. 2015), they are talking about hiding confidential information in a cover image by using techniques of steganography based on heuristic genetic algorithm.

A Study published in 2015, by Bing and colleagues Feng (B. Feng, W. Lu, W. Sun 2015*)* proposed rapprochement of binary image steganography with the latest technology. With this technique, they intended to minimize the deterioration of the structure.

2. Methods

Cryptography is one of the branches that ensures the security of technology. Not just in technology, cryptography, is also used to secure the privacy of information that is peculiar to the specific people. Therefore, judging by the origin of the word cryptography, in Greek kryptos means privacy, graphyen means writing (D. Elizabeth, R. Denning 1982).

In the project, the message is encrypted by using secret-key cryptography, namely the Standard DES - LUCIFER algorithm. Because this method is realized with the secret key encryption system, each user must know the secret key. This secret key, could be shared by meeting in person, or over any other secure channel, or over an insecure channel by using the Diffie Hellman key exchange.

The algorithm LUCIFER developed by IBM in 1974 and had been adopted by the company NBS. This algorithm was further developed and was officially adopted with the name of DES in 1977.

DES algorithm separates the data into 64 bits of blocks, enciphers it with 64-bit key (56 of these are used by the algorithm) (E. Biham, A. Shamir 1993). The general structure of the algorithm is shown in Figure 1.

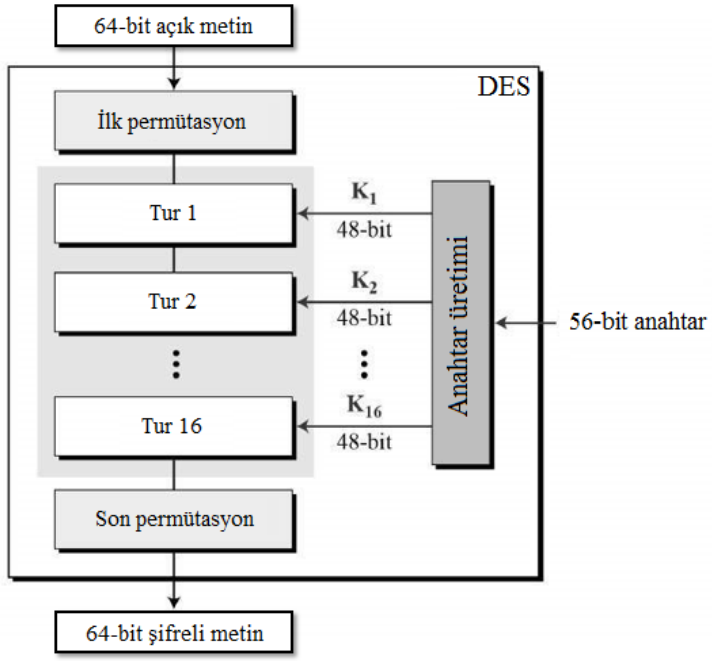

Figure 1: The general structure of the DES algorithm

A message that is encrypted with cryptography technique, will reach to an extra level if it is hidden with the method of steganography. The word steganography is of Greek origin and the "*steganos*" means secret, "*graphien*" means writing (Andach Shahin, 2007).

To hide the encrypted text in a picture digital steganographic technique; the LSB method is used. *LSB Insertion method* is one of the most widely used techniques in digital steganography. In the literature it is known as "the least significant bits". Considering the limited ability of the human senses, small changes would go unnoticed to the eye. On a 24-bit RGB bitmap image each pixel consists of 3 bytes containing red, green and blue color. The intensity of color in each 3-color channel defines the color of the pixel. Modification of one or two of the least important bits, is almost unnoticeable to the human eye (Min Wu, Bede Liu 2003).

Figure 2 shows the pixel byte in the form of bits.

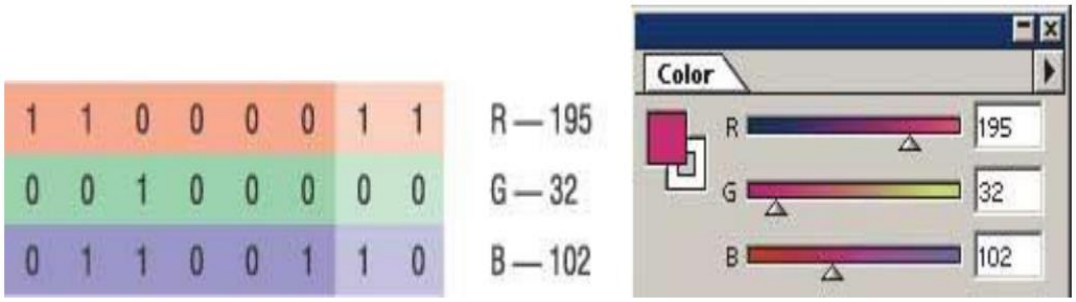

Figure 2: The pixel byte in the form of bits

To hide the encrypted text, and image in the video the steganographic technique is used. Video consists of combination of multiple image frames in a second consecutively. The higher the amount of scanning in one second of video, the better video is seamless. Today, in the TV industry and digital camera 24p, 25p and 30p frame rate per second is used. However, among all of these, the most commonly used one is 24p. The most common Video Coding standard is known as MPEG-2 and MPEG-4. MPEG stands for "Moving Picture

Experts Union". Steganographic techniques to bury information into a video of MPEG-2 standard, designed to function on a real-time basis (P. Shinde, T. B. Rehman 2015). A simple example of a video file transmission is as follows.

1) Collect information on the video;
2) Divide the video into frames;
3) Select the random frame;
4) Place the hidden message into the frame;
5) Combine the original video with stego picture;
6) This way, one can obtain the steganographic video.

Figure 3 shows a block diagram for encoding video.

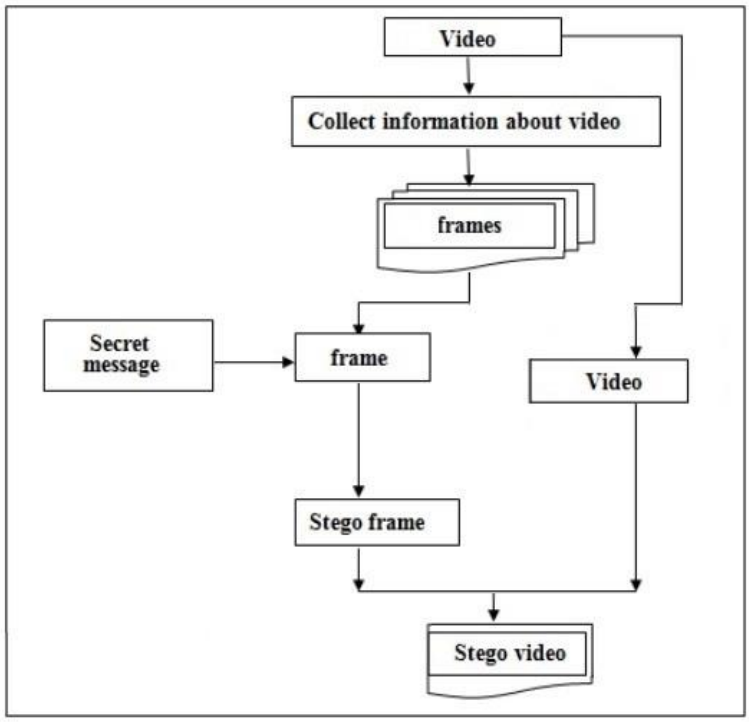

Figure 3: Block diagram for video encoding

3. The Application Developed By Using Cryptographic And Steganographic Methods

As it is known, since, in a secret key cryptography, ciphering and deciphering keys are the same this key needs to be sent over the secure channel to a receiver. Since there is not a secure channel, the key sharing process was carried out by applying the steganographic techniques.

Messaging in the project is realized in three ways; normal messaging, encrypted messaging, and video messaging.

By using normal messaging, a user, sends the message to a receiver after writing the message by clicking on the "Send" button. Figure 4 shows Aylin's user interface of normal messaging, (Hasan and Aylin's messaging interface is the same).

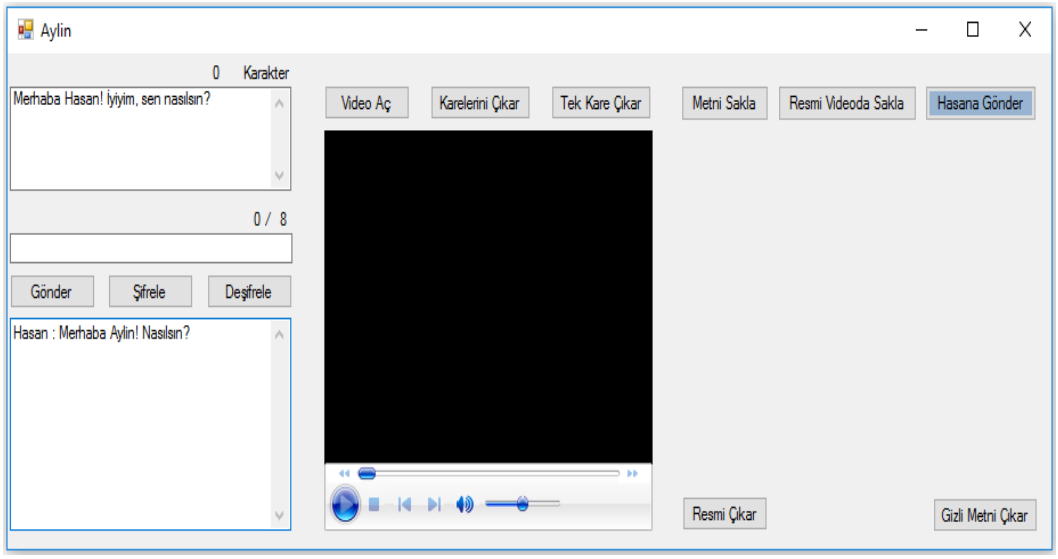

Figure 4: Aylin's normal messaging interface

While Hasan and Aylin are texting to each other, the attacker Ali can see those messages. Figure 5 shows Ali the attacker's normal messaging interface.

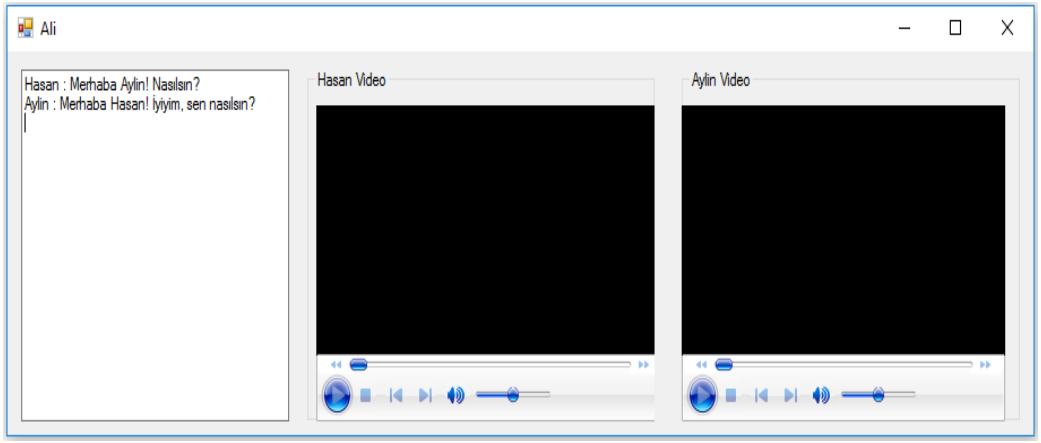

Figure 5: Interface of how the attacker Ali sees normal messaging

Second, *encrypted messaging* method, is to send the message to a receiver after encrypting it with the LUCIFER algorithm. According to the Standard DES, the secret key must be 8 characters, could include, by preference, numbers, letters, and other characters. If text and password fields are left blank the user will be warned. After text and password fields are filled the encryption process starts. Firstly, the equivalent of the text and of the secret key are obtained in the form of bytes. Before the text being encrypted, a new key is generated for each round of algorithm. The text that is separated into blocks of 64 bits, is encrypted with a secret key. Obtained encrypted text blocks, is converted into the character and entered in the text box and by clicking the "Send" button the message is sent to the receiver.

The disadvantage of this method of messaging, is that the middleman realizes the messaging contains confidential information. İn case of active attackers, they may try to intervene in any of the communications. Figure 6 shows the user interface of Aylin's encrypted messaging.

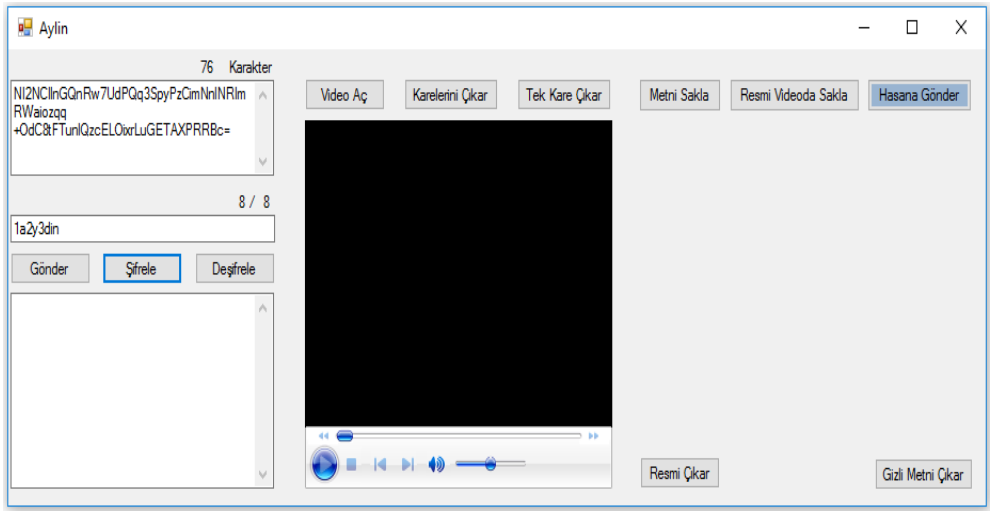

Figure 6: User interface of Aylin's encrypted Messaging

Figure 7 shows the user interface of encrypted messaging that belongs to Ali attacker. Ali receives the same encrypted message but is not able to read it because he doesn't know the secret key. By sending an encrypted message using this method, users attract the wrong kind of attention. If the attacker can interfere with this communication, the method of this messaging is inadequate.

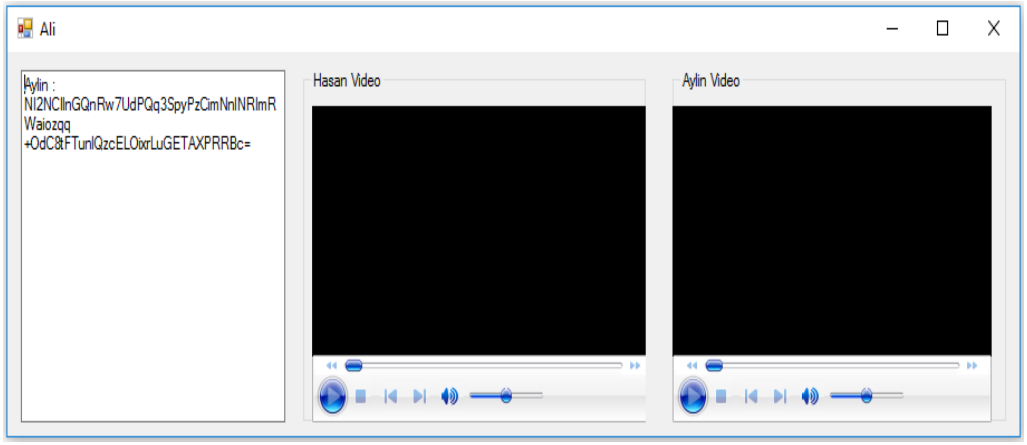

Figure 7: The user interface of how attacker Ali sees the encrypted message.

Thirdly, the video messaging method, compared to the previous ones is more complex and secure than others. Its functional process is shown in Figure 8, through the Hasan example.

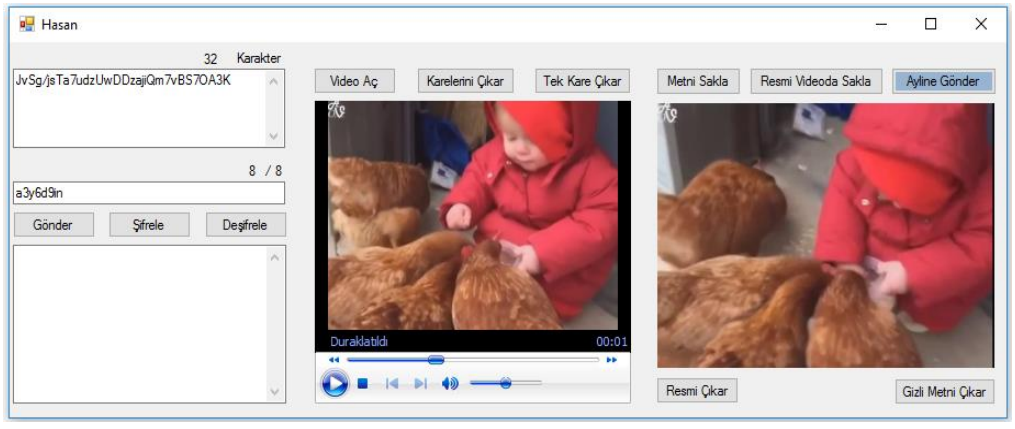

Figure 8: The user interface of Hasan's stego messaging

1) The original message has been entered to the text box;
2) The secret key has been entered to the key box;
3) By clicking on the “Encrypt” button, the message is encrypted;
4) “Open Video” button, opens the selected video;
5) By clicking on the button of “Frame” the selected video frames have been extracted;
6) By clicking “Single-Frame” button, one frame selected among from the extracted frames and it is shown in the right part of the interface.
7) “Save text” button, hides encrypted text and the secret key in the selected picture.
8) “Hide the Picture in the Video” button hides the text hidden image in the video;
9) Lastly, "Send to Aylin" button sends the video to Aylin;

After receiving the video, Aylin follows the following steps. Figure 9 shows Aylin's user interface of stego messaging.

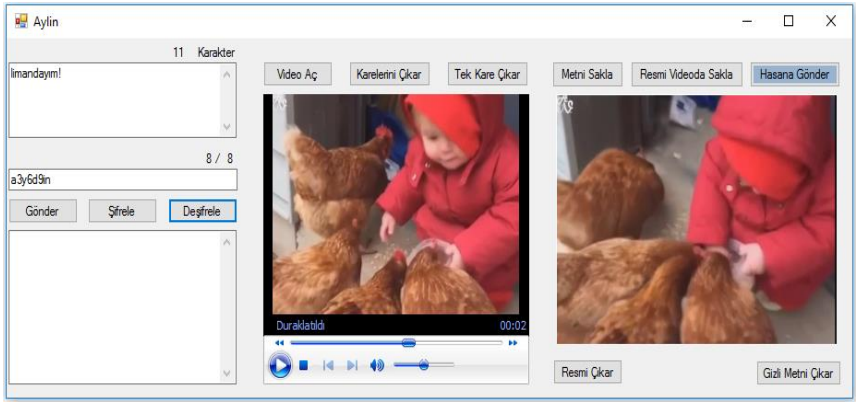

Figure 9: Aylin’s user interface of stego messaging

1) “Extract the Picture” button extracts the image buried into the video to the right part of the interface;
2) “Extract the Hidden Text” button, extracts the encrypted hidden text embedded in the picture and the secret key into the text box and the secret key box;
3) Finally, the “Decipher” button, deciphers the hidden text and presents the original text.

Ali receives the message too. However, he sees the message as an ordinary video, he does not doubt that there is a hidden text buried into the video. Figure 10 shows the user interface of how the attacker Ali sees the video message

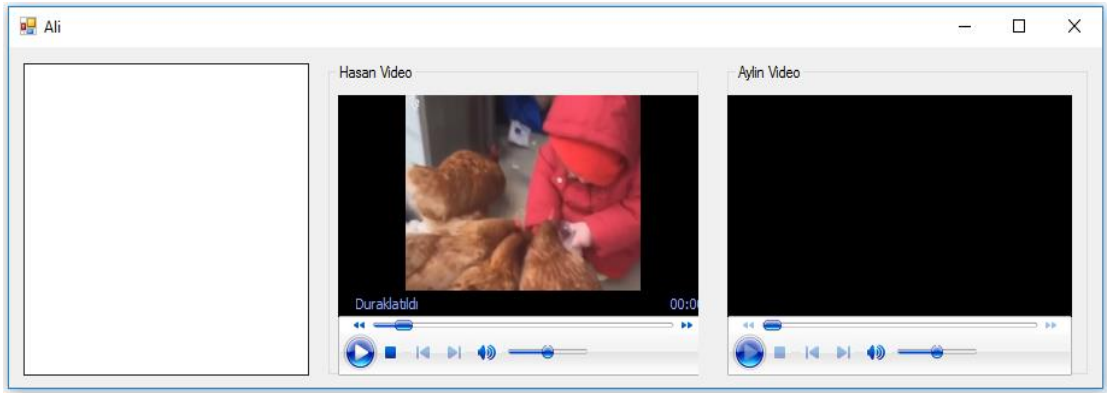

Figure 10: The user interface of how the attacker Ali sees the video message

Methods of messaging with video using Steganographic techniques are described below in more detail.

During *Open Video* process, "Open Video" button selects video from any of desired folders. The selected video is randomly generated with a new name in numbers, is saved in MP4 format to the "Video" folder within the project. The selected video is played in the specified video player part of the interface of the application.

Extract the frames operation, divides the video into its frames. If you try to carry out this transaction without selecting a video, "Please, choose the video first!" message warns you. "Extract the Frames" button firstly deletes the old pictures in the "Frames" folder of the project. Every time you select a new video to prevent the mix up with the old frames the folder is being formatted. To find the number of frames, the duration of the video is multiplied by the frame rate. In the next operation, for loop is created, and it is executed as many times as the number of the frames. Each time it is rotated, starting from 0, the loop takes all the frames with the number of loop, saves them in the "Frames" folder in PNG format. After this process is completed, the "Video is divided into its frames!" message pops up. 25 fps is the frame rate of the video sample. Since the length of the video is 5 seconds, and 5x25=125 frames.

"Extract Single-Frame" is carried out by clicking on the *Single-frame extraction* button. For example, if the number of frames is a three-digit number, two digits of that number is the frame sequence. 12th frame out of 125 frames is selected from this video that is extracted to the "Frames" folder. For example, if the video is 667 frames, 66th frame is selected.

For *Save the Text* process, firstly a new white (R-255, g-255, B-255) picture is created in 1 pixel larger than the width and height of the original image. In the next process, the original picture is placed on the created image, leaving a gap in the height of the image to the right by 1 pixel, and in the width to the down by 1 pixel. By renaming it, the enlarged image is saved to the "Picture" folder. In the next process, the equivalent of the secret key and the encrypted text in bit is obtained. After obtaining the amount of the pixel of the height of the image the height pixel is checked. By changing the least important RGB values of each pixel by 3 bits, the data is saved. If the number of bits of the encrypted text, exceeds the size of height pixels by three times "The text you have entered cannot be fit in the image" warning message pops up. The same terms are valid for the secret key; the bits number of encrypted text should not be more than three times the number of pixels width. Figure 11, is an example for an encrypted text, the size of the encrypted text and an image in which a secret key is hidden.

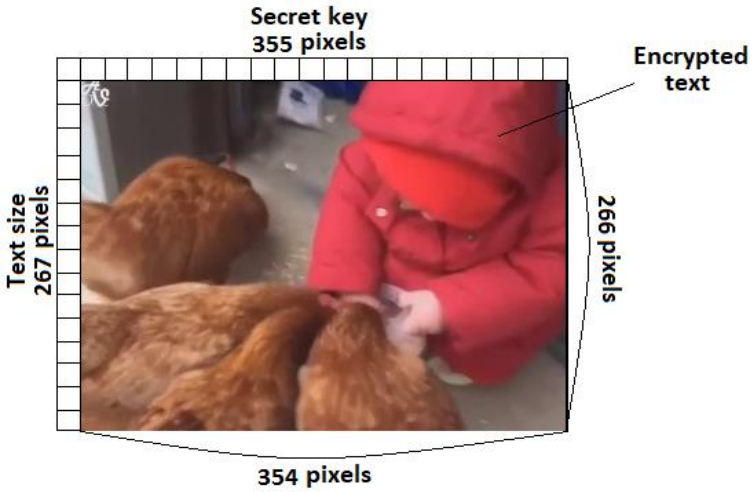

Figure 11: Encrypted text, Text Size, Picture with a secret key

To the top pixels of the width, from the 1st index to the right, the bit equivalent of the secret key, to the left most of the height pixels, from the 0th index to the bit equivalent of the size of the encrypted text and to the original image part, the bit equivalent of encrypted text is hidden. In the illustration, for the data storage process the LSB method is used. If the RGB value of a pixel is odd number, that means its bit equivalent ends with 1, if its value is even number that means its bit equivalent ends with zero. The size of encrypted text and the pixel values in which the secret key is going to be hidden, are reduced to "nR-252, nG-252, nB-252".

If white colored width and height pixel's RGB values are different than 255, that means there is data hidden there, if they are even to 255, that means it is empty. The process of hiding the encrypted text is a little different. Since the RGB values of each pixel in the original image are different, by converting the odd numbers to even ones, data is hidden to the last bit. After the text hiding process is complete "Text Hiding process has been successfully completed... A text is hidden in the picture that is shown!" message pops up. An obtained text hidden image is saved to "SteganoPictures" folder.

For the process of *Hide the picture in the video,* firstly the byte equivalent of the picture then the byte equivalent of video is obtained. A new array is created, as the sum of the size of the video and picture. First, the byte equivalent of the video is inserted, then to the end of that video, byte equivalent of the image is placed in this array and the new array is saved as mp4 file. The name of the saved video is equal to the sum of byte length of the image and frame number of the video. After hiding the picture in the video "Picture has been hidden in the video!" message is displayed. Figure 12 shows an example for a technique of hiding an image in the video.

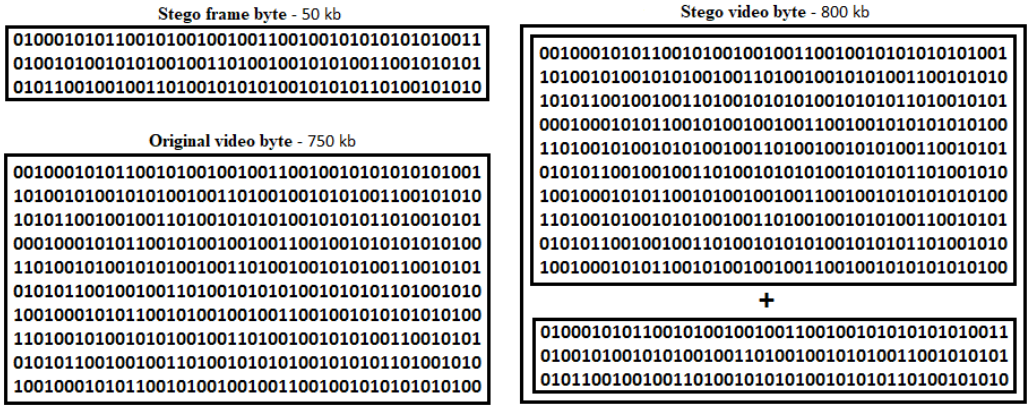

Figure 12: An Example for a technique of hiding an image in the video

During the process of *Extracting the image from the video* to calculate the frame amount of the original video, duration of the video is multiplied by the speed of the transition between frames. Subtracting frame number of the video from the video name which is a digital number, gives you the size of the image. To obtain the byte equivalent of the new video, subtract byte of the image from byte of the new video. Since the first hidden one in new video is the original video, during the process of extracting the image, by commanding the next thread to be handled from the end of the original video byte, the bits of the text hidden image are extracted. The extracted bits are saved in the new image thread. This new image bytes are combined into the picture form again and saved to the "Video" folder with the name of HiddenPicture.png.

For the process of *Extract the Hidden Text*, text hidden image should be handled firstly, in the next step, the size of the image should be handled. Since the size of the encrypted text's RGB values are different than 255, the algorithm extracts the size of the hidden text by creating a loop. To extract the

hidden text, execute a loop as many time as the bit equivalent of the size of the encrypted text in the text hidden part of the original image and find the text. After the loop, collate the bits of encrypted text in the correct order which are collated backwards, and turn the string and extract the text to the text box.

When removing the secret key, the same methods are used. Since the key hidden pixel's RGB values are different than 255, the algorithm extracts the hidden key by creating a loop. It is already known that the size is 64 bits, because a secret key is 8 characters. For this reason, in the width pixel, loop is executed 64 times and the hidden key is found out. After the loop, collate the bits of hidden key that are collated backwards, in the correct order and turn the string and extract the key to the key box. Since the encrypted text and the secret key is extracted, the "Decipher" button executes the original text. Thus, the secret communication between the two users is realized.

4. Conclusion

The aim of the thesis is to prevent the attack of third parties during the process of communication between the forms, by providing necessary safety precautions.

Considering the research within the scope of the thesis, a sample of the thesis study application has been developed in C# programming language of the Visual Studio Platform. In this project which offers three different messaging methods, the method for sending a message within the video file which is more secure is discussed in detail. This information application has been developed by using cryptographic techniques along with steganographic methods. The original text which is encrypted with 8-character secret key by using cryptographic techniques, is hidden in a picture first by using the steganographic techniques, then the picture is hidden in a video and is sent to the receivers. Although the passive-attacker gets this message, because he sees it as a video file, he cannot realize the hidden text placed in it. The contributions of this project are to send the secret key in a video without being suspected to the attacker, in the absence of secure channels or without using Diffie Helman's key exchange or any other method.

It is expected that the features of the project will have important contributions to the future studies.

REFERENCES

Andach Shahin (2007). "New methods and reliability of these methods used in image steganography", PhD thesis, Cornell University, Institute of Sciences

B. Feng, W. Lu, W. Sun (2015) "Secure Binary Image Steganography Based on Minimizing the Distortion on the Texture". IEEE transactions on Information Forensics and Security, Feb. 2015.

D. Elizabeth, R. Denning (1982). "Cryptography and Data Security", Addison-Wesley Publishing Company Inc.

E. Biham, A. Shamir (1993) "Differential Cryptanalysis of the Data Encryption Standard", Springer Verlog

Hemalatha S., Acharya U.S., Renuka A., Kamath P.R. (2012). "A secure image steganography technique using Integer Wavelet Transform", Information and Communication Technologies (WICT), 2012 World Congress

M. Nosrati, A. Hanani, R. Karimi (2015). "Steganography in Image Segments Using Genetic Algorithm", 5th IEEE International Conference on Advanced computing & Communication Technologies

Min Wu, Bede Liu (2003). "Multimedia Data Hiding", New York, Springer Science + Business Media

P. Shinde, T. B. Rehman (2015) "A Novel Video Steganography Technique", International Journal of Advanced Research in Computer Science and Software Engineering.

R. Jain, N. Kumar (2012). "Efficient data hiding scheme using lossless data compression and image steganography", International Journal of Engineering Science and Technology (IJEST),