


SECURE STEGANOGRAPHIC DATA TRANSMISSION METHOD FOR PERIODICALLY UPDATED DATA

Timucin Koroglu^{1*} , Refik Samet²

¹Pamukkale University, Computer Programming Department, Denizli, Turkey

²Ankara University, Department of Computer Engineering, Ankara, Turkey

Abstract. The main goal of the work in the field of steganography is to transmit data in a cover image with minimal distortion. The proposed method aims to transmit periodically updated data in a highly secure manner through a low-distortion steganographic cover image. Storing these data directly in the cover image poses a major security risk. Therefore, the data to be transmitted is not directly hidden in the cover image, but in the chaotic pattern image on the server side. In the cover images, the row and column address data of the logical fields of the chaotic pattern image are hidden. Thus, a small amount of data representing a large amount of data is sent to the receiver along with the cover object. Since the representative data is much smaller than the data to be transmitted, the distortion of the cover image is minimized. The method also includes reverse social engineering. In this application, attackers are misled to distinguish between the hidden data and the real data. PSNR metric is used to compare the method with other studies. Only the blue channel of the Baboon, Barbara and Lena cover images was loaded with data. In this case, the PSNR value in the red and green channels was infinite. The PSNR result in the blue channel is also higher than the methods used in other studies. The high results indicate that the distortion in the cover images is lower.

Keywords: Data hiding, secure data transmission, image-based steganography.

***Corresponding author:** Timucin, Koroglu, Computer Programming Department, Pamukkale University, Denizli, Turkey, e-mail: tkoroglu@pau.edu.tr

Received: 12 September 2022; Revised: 22 November 2022; Accepted: 2 December 2022;

Published: 29 December 2022.

1 Introduction

With the increasing power of the Internet in recent years, society is rapidly digitising and personal data is rapidly being transferred to cyber environments. However, insufficiently secure networks expose a potential danger to the security of our data. Therefore, secure data transmission has become one of the top priorities in the IT world. So much so that cyber-attacks and defence methods have started to be included in the national defence policies of countries.

Cryptology and steganography are the sciences that ensure that data transmitted over the internet reaches the recipient securely (Samet & Koroğlu, 2019). Cryptology is a whole of techniques and applications based on mathematical methods to transform data in data communication between two or more parties (Yalman et al., 2014). In cryptology, the non-secret text is called plaintext and the secret text is called ciphertext. The process of converting from plaintext to ciphertext is called encryption, and the process of converting ciphertext to plaintext is called decryption (Klima et al., 2018). Steganography is the science of data hiding. It aims to hide data on a cover and transmit it in this way to prevent third parties from examining and intercepting it (Li & Lu, 2018). The cover object can be text, audio, image or video. According to whether the secret data can be extracted from the cover object, it can be classified as invertible

or non-invertible (Jung, 2018).

The success of steganography depends on the imperceptibility of the data hidden on the cover object. Obscurity, robustness and imperceptibility are the three main characteristics of steganography. In steganography, data is hidden in spatial or frequency space domains. Data hiding in spatial domains is accomplished by changing the pixels of the cover image of the data to be hidden. In the frequency domain, data is hidden using mathematical methods (Subhedar & Mankar, 2018). Maximizing the hidden data without degrading the cover image quality is one of the most important issues in steganography. The human eye cannot detect small amounts of hidden data. However, the existence of large amounts of hidden data can be revealed by statistical tests (Vanmathi & Prabu, 2018).

The main advantage of steganography over cryptology is that it cannot be detected. Encrypted data made unintelligible by cryptology attracts the attention of attackers. However, when the data is to be transmitted without attracting the attention of attackers, it is hidden in a cover object using steganographic methods. The hidden data in the cover object is sent securely to the receiver and the security problem is overcome (Bai et al., 2017).

In this paper, we propose a steganographic method based on reverse social engineering to securely transmit large amounts of data, which can be expressed in small data sets, over the Internet. The proposed method can be used to transmit periodically updated data securely over the Internet. Examples of such data include usernames and passwords used to access and authorize remote systems, credit card information. Another type of data can be online questions. This is because the importance of distance learning has increased significantly during pandemics. Emirtekin et al. (2020). A natural consequence of distance learning is online exams. This type of data needs to be kept away from attackers. If the questions fall into the hands of attackers, it can have serious consequences. Therefore, secure transmission of online exam questions is very important. These data is hidden in several fields of a chaotic pattern image. The chaotic pattern image is located on the receiver/server side. These fields are addressed based on row and column data. The size of the address data is very small compared to the secret data to be transmitted. The addressing application aims to express large amounts of data with small amounts of data. In the proposed method, the address data is hidden in different cover images. The number of cover images is equal to the number of fields of the chaotic pattern image. The next process is the transfer of the cover images to the sender. This is the first stage of the process and is done periodically or as needed. The second stage of the process is the transmission of the data. At this stage of the process, fake data is generated that resembles real data. This process happens on the sender's side. The fake data is sent to the recipient side simultaneously with the real data. This is reverse social engineering to mislead attackers and distract them from the cover image. Real data is not sent directly. They are sent with the addresses of the chaotic pattern image fields hidden in the cover images. Due to the small amount of data hidden in the cover image, the cover image is very similar to the original image. This makes it very difficult for attackers to detect that there is hidden data in the cover image.

The implementation of the proposed method is based on the transmission of the username and password data. The application was developed with HTML, CSS, JQuery and PHP web technologies. With this application, low-distortion transmission of important data such as usernames and passwords is realized at a high data rate using steganographic techniques. Furthermore, the security level has been increased with reverse social engineering. The reason for choosing a username and password for the proposed method implementation can be explained as follows.

On the Internet, in structures based on a server/client architecture, ensuring secure communication is an important issue. Namely, there is a possibility that malicious people can eavesdrop on the communication between the server and the client. The security of communication between the server and the client is ensured by password authentication. The password sent by the client must be verified on the server. If the Internet network is not secure enough, the problem of password authentication arises. This problem often occurs in applications for logging in to web

servers or logging in for remote access to computer networks. For these reasons, implementing secure authentication over the insecure Internet is a very important issue (Hussain et al., 2015).

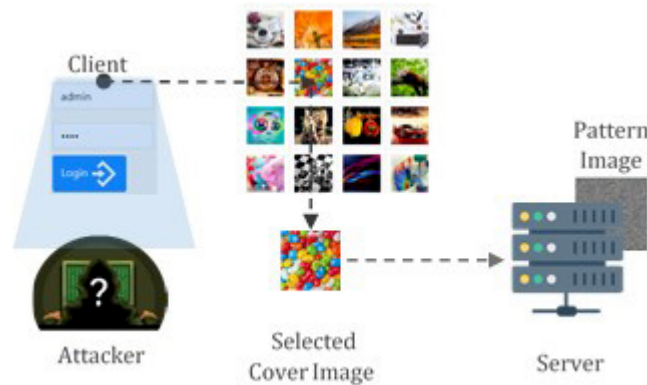


Figure 1: General working diagram of the proposed method.

The general operation representing the application is shown in Figure 1. In the application, 16 cover images and one chaotic pattern image were used. The chaotic pattern image is generated by the system. The user must register to the system to be authorized. During the registration phase, the username and password set by the user are considered "fake", while the username and password generated by the system are considered "real". Each character of the system-generated username and password is replaced by matching the row and column data of the characters matrix. The modified data is circularly hidden in each of the 16 different areas in the chaotic pattern image. Then, the address data of 16 different areas of the chaotic pattern image and the identity data of the cover images are hidden in the 16 cover images on the server side. The bit length of the address and cover image identification data is 12 bits. The bit length of the username and password that are changed using the character matrix is 96 bits. In this case, 96 bits are expressed with 12 bits. After that, all cover images are transferred to the client computer. The registration process takes place at times when the username and password data are updated regularly. For safety reasons, it is preferable that the periodic intervals of these processes be long.

In the second phase of the process, the user enters a fake username and password set by the user into the system interface. The data entered is actually a social engineering tool. The coordinate data generated based on the fake username and password is used to select a cover image from a 4x4 matrix of cover images. The selected cover image is sent to the server, which is the recipient. The server reads the data of the image ID and the coordinates of the chaotic pattern image from the cover image. This data addresses one of the 16 fields in the chaotic pattern image. From the addressed field, the equivalent of the username and password in a matrix of characters is read. The real username and password are then determined by remapping the read data with the character matrix.

In recent years, applications for hiding data using steganography have been deemed worthy of research in scientific circles and many publications have been made. Most of these publications aim to improve the basic principles of steganography. Some of them are as follows.

In Huan et al. (2019) did not randomly place the secret data in the cover image as in many traditional LSB methods. They have used a fast progression method to determine the location of the hidden data. Thus, they achieved high data capacity and low image distortion in the cover image.

Vanmathi & Prabu (2018) used a fuzzy logic approach to detect edges in an image, hiding data in regions distinct from edge regions to preserve the principle of imperceptibility. In data hiding, they preferred the method of hiding pixel data in the LSB bit.

Wu et al. (2018) hid the data in the contrast data rather than in the pixels of the image to minimize the distortion rate of the image while preserving the image quality.

Singh & Shaw (2018) proposed a hybrid model to achieve higher security. In this model, they presented a hybrid model by integrating QR code and cryptology into the LSB approach.

Sheshasaayee & Sumathy (2017) hybridized the plaintext sent by online banks to their users with cryptography and steganography techniques. They used text as a cover object. Then they encrypted the cover object to increase security.

Torvi et al. (2016) noted in their study that using an image as a cover object would take up a lot of space and result in high bandwidth. Therefore, they suggested using an enriched text file as a cover object for the secret message to be sent. The message has been transmitted via a text file after being encrypted.

Shirali-Shahreza & Shirali-Shahreza (2007) suggested that combining password transmission and CAPTCHA with steganographic techniques can make MMS sending more secure. In this work, the password is stored as an audio file on the server to which the message is to be sent. Then the MMS to be hidden is inserted into a cover image. Both files are sent to the recipient. The receiver learns the password by listening to the audio file, and then enters the password into special software to reveal the hidden data in the image.

In another study, Shirali-Shahreza tried to increase the security of data transmission using steganography in mobile banking. In this work, the secret data along with the password are stored in an image and placed on the website. The user is sent the address of the image on the website via SMS. When the user enters this address, the image is transmitted to the client side, and when the password is entered, the secret message inside the image is extracted Shirali-Shahreza (2007).

Wahab et al. (2021) have investigated how data transmitted over the Internet can be transferred securely and quickly while taking up little space on physical media. In their work, they compressed the plaintext losslessly with Huffman coding after encrypting it with the RSA algorithm. The image used as cover object is compressed with the discrete wavelet transform DWT, a lossy compression algorithm. The encrypted and compressed plaintext hidden in the compressed cover image using the DWT algorithm is sent to the receiving side. On the receiving side, the operations are performed in the reverse direction and the encrypted text is decrypted using the key.

Using a genetic algorithm, Shyla et al. (2021) investigated the selection of the appropriate cover image for the data to be hidden and the suitability of which pixels of the cover object can be used to hide this data. Thus, by selecting the appropriate cover image and pixels, the hidden data is aimed to cause very little change in the stego image. The hidden data is in another payload image. The proposed method consists of two stages. In the first stage, eight images are selected as cover images from a database of one hundred images. In the first stage, statistical data from the payload and cover images are analyzed for this process. In the second stage, it is optimized in which pixels of the selected cover image the hidden data should be placed. In this process, there are millions of options where data can be placed. The most optimal option among these possibilities is determined by the genetic algorithm.

Mohammed & Al Saffar (2021) used the McEliece cryptosystem in their study to make data transmission more secure with image steganography. The McEliece cryptosystem is a public-key cryptosystem based on error-correcting codes. In this paper, the plaintext is first encrypted using the McEliece algorithm. The ciphertext is then hidden in the cover image using the LSB method. The generated stego image is sent to the receiver. This makes the data transmission more secure.

2 Materials and Methods

The proposed method is based on a client-server architecture. In this method, data communication on both sides is performed using steganographic methods. In the usual steganographic methods, the distortion rate of the image increases with the amount of data hidden in the cover image. The proposed method performs indirect data transmission. To increase the amount of data to be transmitted indirectly, the number of fields in the chaotic pattern image must be reduced. Thus, areas with smaller address data can be defined. Since the address data is transmitted together with the cover images, the distortion rate in the cover image is reduced. It also ensures that the transmitted data is unintelligible if intercepted. The security level is increased by including reverse social engineering in the method. The concepts used in the proposed method are as follows.

2.1 Chaotic pattern image

It is a 512 x 512 pixel image, 24 bit deep and in png format. Each pixel of the chaotic pattern image is randomly generated by the system. Therefore, it is a meaningless picture and has no in-picture edges. Inside this image, the data to be transmitted indirectly will be hidden. The chaotic pattern image is server-side and generated separately for each user.

In the proposed method, the data that needs to be transferred securely can be username and password, credit card information, online exam system questions, etc. These data are hidden in the chaotic pattern picture on the server side using steganographic techniques. An example of a chaotic pattern picture is given in Figure 2.



Figure 2: Example of a chaotic pattern image.

2.2 Fake and real data

One of the most powerful methods attackers use to obtain important data is social engineering. In the proposed method, reverse social engineering is used. To achieve this, fake data is used alongside real data. Fake data is used for two purposes. The first is to mislead attackers by diverting their attention. The other is to generate row and column data to select the cover image to be sent to the recipient. The selected cover image is sent to the server. The server uses the cover image to resolve the actual data. Even if attackers get their hands on the fake data, they cannot access the real data because they do not have cover images on their devices. The real data is the data to be transmitted to the receiving end. On the server side, they are hidden in a chaotic pattern image located in a private folder designated for each user.

2.3 Cover Image

The cover image sent between client and server carries the address data of the chaotic pattern image containing the actual data. Even if the cover data is intercepted and the secret data is detected, it is then meaningless. Sending the address data instead of the data itself ensures obfuscation, which is one of the basic principles of steganography. The address data hidden in the cover image depends on the number of fields in the chaotic pattern image. If the number of fields is 256, the address data consists of 8 bits. Of the 8 bits of data, 4 bits indicate rows and the other 4 bits indicate columns. Reducing the number of fields increases the bit

capacity of the field. This allows the transmission of larger data with address data of smaller size. Along with the address data, the cover image identification data is also hidden in the cover image. The bit length of the cover image ID is calculated according to the formula in equation 1.

$$\log_2 n = k \tag{1}$$

In the equation, k is the number of cover images and n is the bit length of the image’s data ID. If 16 cover images are selected, the bit length of the image ID results as n=4 from the equation $\log_2 n = 16$.

2.4 How the method works (Phase I)

The operation of the method consists of two basic processes. In the first phase, before the communication starts, the server-side prepares the real data, the fake data, the chaotic pattern image and the cover images. The first of these preparations is the generation of the chaotic pattern image, the character matrix and the fake data. In this stage, the real data are loaded into the image fields of the chaotic pattern. Also, in this stage, the address data and the identity data of the cover image are loaded into the cover image and transferred to the client side. In the first phase of the process, all necessary procedures and details are shown in Figure 3.

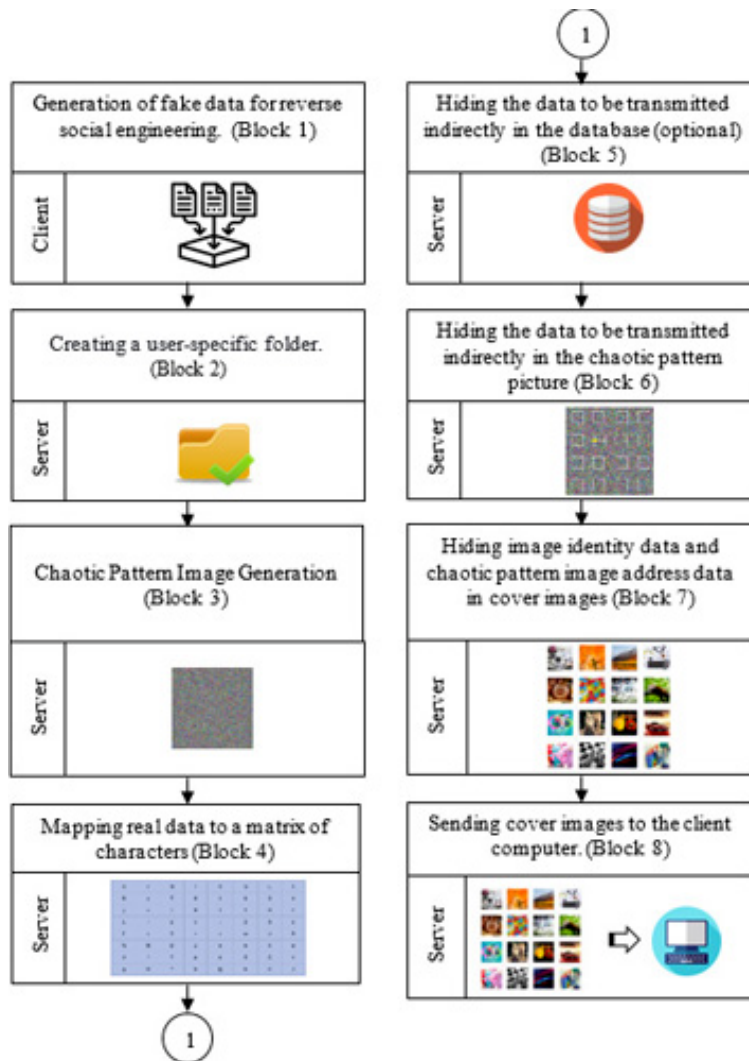


Figure 3: How the method works – phase I.

2.4.1 Generation of fake data for reverse social engineering (Block 1)

In this phase of the process, fake data is generated for reverse social engineering. The fake data has two tasks. One is to hide the cover image from the attacker’s eyes. The other is to generate address data to select the cover image to be sent to the server. The real data is sent to the server indirectly in the cover images. The operations performed in this phase contribute to secure communication by distracting the attackers from their target.

2.4.2 Creating a user-specific folder (Block 2)

In this process, a user-specific folder is created on the server side. This folder is used in the second phase of the proposed method. The purpose of the folder is to store the cover images sent from the client computer.

2.4.3 Chaotic pattern image generation (Block 3)

The chaotic pattern image is randomly generated by the system and stored on the server. Its function in the method is to store real data. This image of size nxn receives internal margins in all directions. The size of the image regions of the chaotic pattern in pixels is calculated according to the formula given in equation 2. The meaning of the abbreviations given in the equation and the formula is as follows.

n= coordinate data row/column length

IS = Image size

im1=First inner margin

im2=Second inner margin

$$FieldSize(Row/Column) = [(IS - 2im_1)/2^n] - 2im_2 \tag{2}$$

2.4.4 Mapping real data to a matrix of characters (Block 4)

Real data is not stored in ASCII character format. In the method, each character is converted into other data. For this purpose, a character matrix is randomly created for each user. The character matrix is 8 x 8 in size. In the character matrix, the rows and columns are addressed by 3-bit long data.

The row and column position data of each character of the real data in the character matrix are determined. The determined row and column data are concatenated at the bit level to obtain new data representing the character. The resulting data is in row-column data format. The data stored in this transformed form cannot be resolved by third parties without the character matrix. The characters matrix is shown in figure 4.

Matrix of Characters 8x8

B	1	H	u	J	N	4	E
C	p	0	*	Q	M	V	6
O	o	e	c	y	8	f	5
P	l	#	T	D	v	z	K
m	Z	l	3	g	2	t	S
A	-	n	R	F	L	l	h
U	+	G	j	k	i	9	a
Y	7	%	s	?	b	r	d

Figure 4: Matrix of characters 8 x 8.

2.4.5 Hiding the data to be transmitted indirectly in the database (Block 5)

The data resulting from matching real data with a matrix of characters can be stored in a database. This process depends on the application that uses the data to be transferred. If there is no need, this step is skipped.

2.4.6 Hiding the data to be transmitted indirectly in the chaotic pattern picture (Block 6)

The real data in the form of row and column data are hidden in their assigned areas of the chaotic pattern picture. Each field in the chaotic pattern picture is given inner margins to increase complexity. The data is written to the areas outside the inner margin.

2.4.7 Hiding image identification information and initial coordinate data of the chaotic pattern in cover images (Block 7)

The row and column data of n random fields in the chaotic pattern image are hidden separately in each cover image. Apart from these data, the identity data of the cover images are also hidden in the cover image. This data is used to find out where the field addresses of the chaotic pattern image are hidden in the cover image.

2.4.8 Sending cover images to the client's computer. (Block 8)

After uploading the data to be hidden in the cover images, the cover images are transferred from the server to the client side. The cover image is transferred at the time of data update.

2.5 Secure data transmission from the sender/client side to the receiver/server side (Phase II)

The general process of secure data transmission between sender and receiver is shown in Figure 5. Reverse social engineering and the transmission of data to the receiver via cover objects takes place in this phase.

2.5.1 Cover image selection with data produced through fake data (Block 1)

This phase is about reverse social engineering against attackers by using fake data. The DiGraph algorithm was chosen for this application Por et al. (2017). The generated data from the fake data generates row and column values to select the cover image. The generated values select one of the cover images in the matrix. A 4x4 matrix of cover images is shown in Figure 6.

2.5.2 Uploading the selected cover image to the server side (Block 2)

At this stage of the process, the cover image selected by the user is uploaded to the server. The folder to which the cover image is uploaded is the folder created for the user in the first phase of the process. This folder also contains the chaotic pattern image where the real data is hidden.

2.5.3 Reading the data in the loaded cover image (Block 3)

The 4-bit long image ID data hidden in the cover image uploaded to the server side is read. The image ID data is used to address another field in the cover image. The addressed field contains the row-column address data of one of the chaotic pattern image fields.

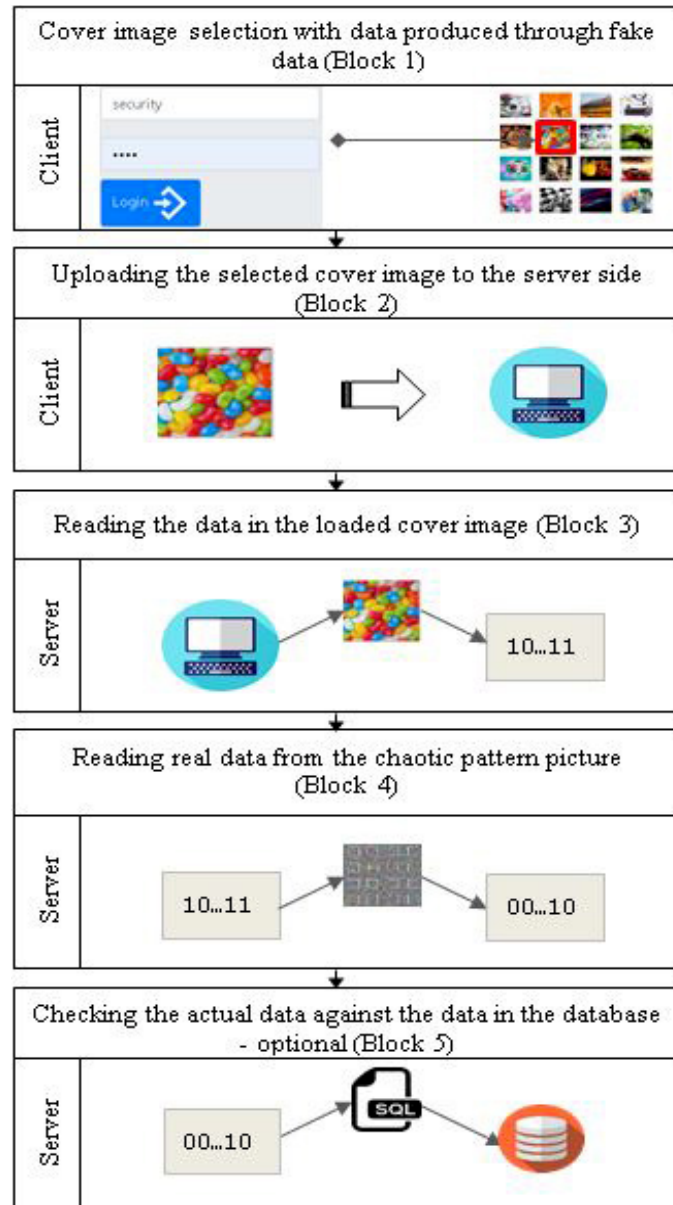


Figure 5: The general operation of secure data transmission between sender/client -receiver/server - phase II

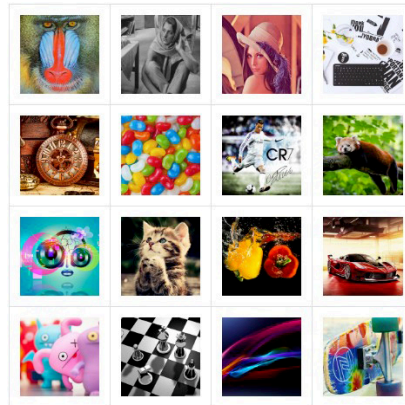


Figure 6: Cover images

2.5.4 Reading real data from the chaotic pattern picture (Block 4)

The address data of the chaotic pattern is read from the cover image and decoded by the system hiding the actual data.

2.5.5 Checking the actual data against the data in the database (Block 5)

In this phase, the read data can be compared with the database records. If as a result of the query a record of the read data is found, the correctness of the data can be confirmed.

2.6 Application of the method

To demonstrate the applicability of the method, a secure transmission of the username and password for system authorization from the sender to the receiver was implemented using steganographic techniques. The method should be analyzed in two basic phases. These are the processes of registration and login to the system. The interface developed for the application is shown in Figure 7.

In the first stage of this process, users are requested to register their username and password

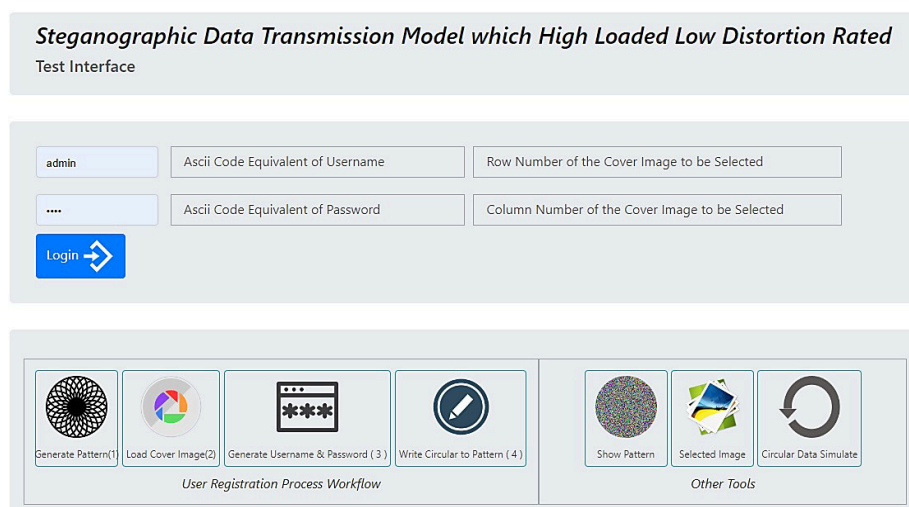


Figure 7: Method application interface

information in the system. Then a user-specific folder is created on the server side. This folder is used when users log on to the system.

In practice, the row and column values for addressing the chaotic pattern image are each

4 bits long. In total, the chaotic pattern image is divided into $16 \times 16 = 256$ equal areas. According to Equation 2, where the pattern image is 512×512 pixels, $im_1 = 56$ pixels, $im_2 = 5$ pixels, and the row and column data are each 4 bits long, each part of the pattern image is $[512 - (2 \times 56)] / (2^4) - 2 \times 5 = 15 \times 15$ pixels. The fact that the row and column data are each 4 bits long means that the full address is 8 bits long. In the method, the length of the cover image credential is 4 bits. According to these parameters, the data to be hidden in the cover image is $4 + 8 = 12$ bits long. The 12 bits of data in the cover image corresponds to $15 \times 15 = 225$ bits of data. This is because in the proposed method only the blue channel of the color pixels is loaded with data. In the method, the red and green channels are not loaded with data. Thus, one bit of data is loaded per pixel. This situation is shown in Figure 8. A representation of the area that needs to be loaded in 16 of the total 256 co-fields is shown in Figure 9.

Eight-character real usernames and passwords used for authorization are generated by the

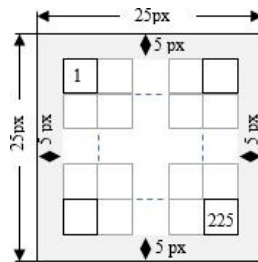


Figure 8: Illustration of one of the 15×15 areas of a chaotic pattern image.

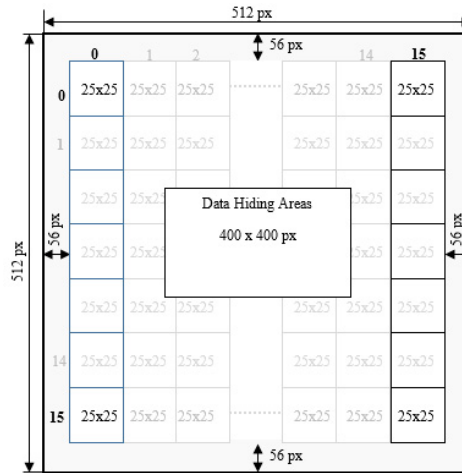


Figure 9: The division of a complex pattern picture into logical areas.

system on the server side. When users log in to the system, system-generated usernames and passwords are taken as the basis for authorization. When generating this data, care was taken to ensure that the username and password consist of lowercase letters, uppercase letters, characters and numbers.

In the application, the username is uoz9@Sad and the password is fzN?99p3, created by the system. This case and the character matrix generated specifically for the user are shown in Figure 10. The first letter of the user name "u" is generated according to the character matrix as follows: The position of the letter "u" in row and column format is 1 for row and 6 for column. From these values, the row value $(1)_{10}$ in decimal system becomes $(001)_2$ in binary system. Similarly, the value of $(6)_{10}$ columns in the decimal system becomes $(110)_2$ in the binary system. In the binary number system, the rows and columns are placed next to each other to get the new value of the letter "u". In this case, the ASCII equivalent of the letter "u" is expressed as $(001110)_2$ instead of $(01110101)_2$.

This data is used to authorize the user to access the system. The matching data is displayed below the character matrix in Figure 10.

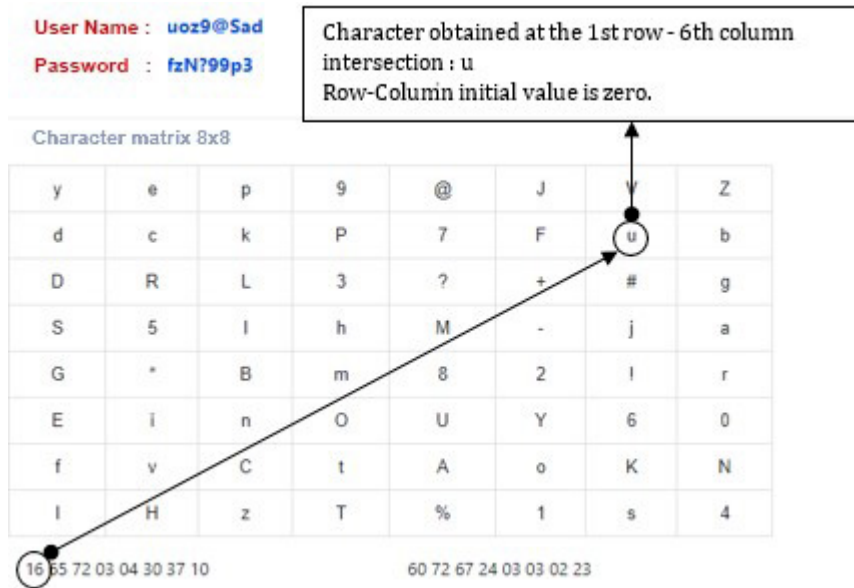


Figure 10: Coordinate data generation.

The data obtained from matching the real user name and password with the character matrix must be stored in the database. These data is used to allow the user to access the system for verification purposes.

Hiding the addresses of the 16 random fields in the chaotic pattern image, formatted as rows and columns, in the cover images is shown in Figure 11. The dark areas represent the logical regions where the data is hidden. 16 65 72 03 04 30 37 10 data is hidden as username and 60 72 67 24 03 03 02 23 data is hidden as password in the complex pattern image. In the proposed method, data is stored circularly to increase the complexity of storage. Figure 11 also shows how the data 16 65 72 03 04 30 37 10 corresponding to the username uoz9@Sad is hidden in the complex pattern image.

2.6.1 The process of logging into the system.

Each character of the entered fake username and password is converted into ASCII codes by the system. In the next step, these codes are normalized using the min-max function. The min-max function reduces the ASCII codes to the range [0-1]. Each reduced value in the range [0-1] is multiplied by 4, since the matrix of cover images is 4 x 4, and eight values in the range [0-3] are obtained. These give the row and column values of the cover images in the 4x4 matrix. Assuming that one of the characters of the username is r and the smallest ASCII code equivalent of the other characters of the username is 103, the function works as follows. The ASCII equivalent of the letter r is 134. In this case, according to Equation 3, $x = (134-103)/(137-103) = 0.91$. Then x' is multiplied by 4 to get $0.91 * 4 = 3.64$. When the result is rounded down, the row value is 3. The min-max function Jain et al. (2005) used to perform these operations is given in equation 3.

$$x' = (x_i - x_{min}) / (x_{max} - x_{min}) \tag{3}$$

In the next step, two numbers are selected from the two groups of eight numbers in the range [0-3], one of which represents the row and the other the column. This data determines which cover image is sent to the server side.

Figure 12 shows an example of the interface developed to implement the method, the ASCII codes generated for the username and password, and the data obtained after normalizing these

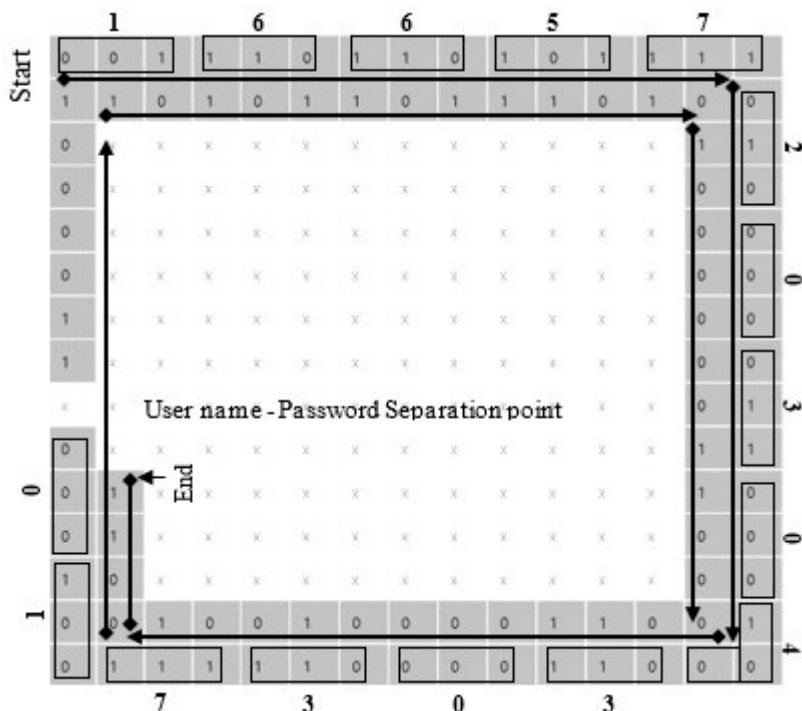


Figure 11: In Method Application Software, circular writing of the real user name into a complex pattern image.

codes using the min-max function.

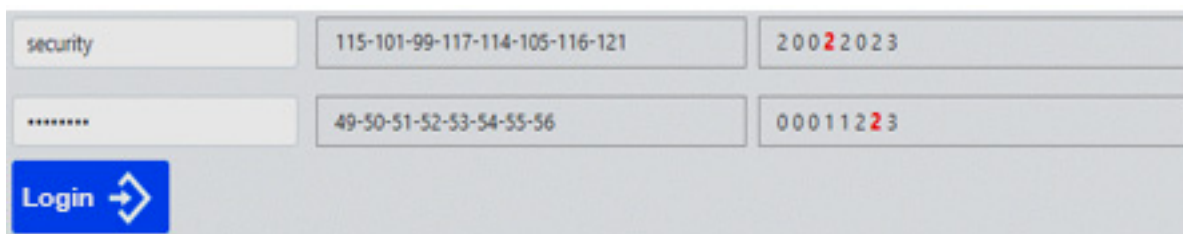


Figure 12: Selection of the cover image with a fake username and password.

Figure 12 shows that the data of 2 comes from both the username and the password. In this case, the cover image in row 2 and column 2 is sent to the server.

At this stage of the process, the cover image, which hides the real username and password, is uploaded to the server. By extracting the address data of the chaotic pattern image from the cover image, the 96-bit username and password data are circularly read in the chaotic pattern image stored on the server side.

The actual user and password data read at this stage is queried in the database. If the records belonging to the read data are found as a result of the query, the user is authorized for the system.

3 Experimental Results and Analysis

Any processing applied to an image can result in a significant loss of image quality. Methods for evaluating image quality fall into two categories: subjective and objective. Subjective methods are based on sensory evaluation by humans, without reference to specific criteria. Objective methods are expressed in terms of statistical parameters and tests. They are based on knowledge and facts. The measurement parameter MSE, PSNR (Peak Signal-to-Noise Ratio), used to

measure visual quality, is an objective method. The MSE formula is given in equation 4, and the PSNR formula is given in equation 5. As MSE (mean square error) approaches zero, PSNR approaches infinity. It can be concluded that higher PSNR values indicate higher image quality. In other words, a very low PSNR value indicates that there are large numerical differences between the two images Hore & Ziou (2010).The Peak Signal-to-Noise Ratio (PSNR) metric can also be used in other areas, such as measuring I-frame coding time for H.264/AVC and measuring the level of increase in output bit rate Li et al. (2013).

f : Original image

g : Test image

$$MSE(f, g) = \left(\frac{1}{M.N}\right) \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - g_{ij})^2 \quad (4)$$

$$PSNR(f, g) = 10.log_{10}(255^2/MSE(f, g)) \quad (5)$$

To demonstrate the ability of the proposed method to handle different amounts of data and to determine the degradation rate of the cover image according to these capacities, the chaotic pattern image is divided into logical regions of different sizes. These regions can be addressed with row/column data of 2, 3, 4, 5, 6, 7 and 8 bits. Therefore, in addition to the 4-bit data describing the cover images, address data of different lengths for different time periods were hidden using the LSB method. The capacity of the amount of data transmitted indirectly is calculated according to equation 2.



Figure 13: Baboon and Barbara images.

Table 1: PSNR values of Baboon and Barbara cover images for different amounts of data.

Amount of data hidden in Cover Image (bit)	Indirectly transmitted data (bit)	Baboon PSNR value (dB) - Blue channel	Barbara PSNR value (dB) - Blue channel
4 + (1+1) = 6	$(((512-2*56)/2^1)-(2x5))^2=36.100$	71,29	72,26
4 + (2+1) = 7	$(((512-2*56)/2^2)-(2x5)]x$ $(((512-2*56)/2^1)-(2x5)]=17.100$	70,5	71,29
4 + (2+2) = 8	$(((512-2*56)/2^2)-(2x5))^2=8100$	70,5	71,29
4 + (3+2) = 9	$(((512-2*56)/2^3)-(2x5)]x$ $(((512-2*56)/2^2)-(2x5)]=3600$	69,83	70,5
4 + (3+3) = 10	$(((512-2*56)/2^3)-(2x5))^2=1600$	69,83	70,5
4 + (4+3) = 11	$(((512-2*56)/2^4)-(2x5)]x$ $(((512-2*56)/2^3)-(2x5)]=600$	69,25	69,83
4 + (4+4) = 12	$(((512-2*56)/2^4)-(2x5))^2=225$	69,25	69,83

PSNR was used to determine the distortion rate between the embedded image and the original image. Different PSNR values were measured at different loads using the Baboon and Barbara cover images shown in Figure 13, various PSNR values were measured at different loads.

The results are shown in Table 1. The measured PSNR values show that the distortion of the cover image decreases as the amount of data increases. Thus, when large amounts of data need to be transmitted end-to-end, the data can be transmitted with very low distortion.

Table 2: Baboon histogram charts - payload : 12 bits.

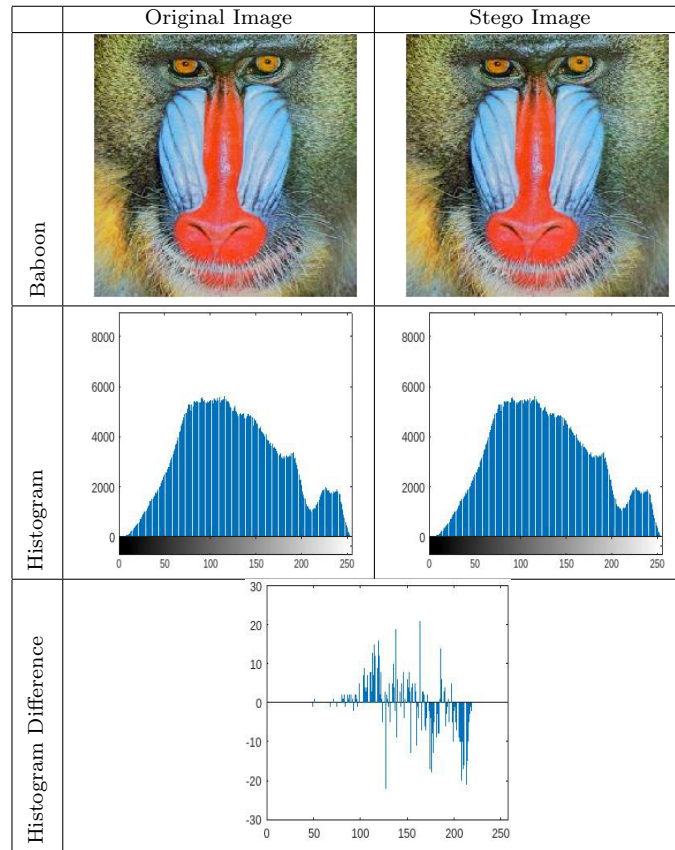


Table 3: In the model, measured PSNR values (cover image)

PSNR	Channels		
	R	G	B
	Inf	Inf	69,22

In the method implementation, 16 cover images and one chaotic pattern image with real data were used. The size of both images is 512 x 512 pixels. For applications where data transfer performance is important, the size and number of cover images can be reduced. For applications where a large amount of data needs to be stored, a larger chaotic pattern image or multiple chaotic pattern images can be used if necessary. In such cases, a different coordinate system other than the recommended coordinate system can also be used. Thus, large amounts of data can be moved indirectly by small data.

Table 2 shows the histograms of the original Baboon image and the cover image and the difference between the two images. The graphs were created using Matlab software. The PSNR value measured with the original Baboon image and the cover image applies only to the blue channel. The measured value is 69.2. Since no data was uploaded for the red and green channels, the measured PSNR value for these channels is infinite. The measured PSNR results are shown in Table 3.

Table 4 shows the histogram plots and the difference between the original and cover image of the chaotic pattern image where the real username and password are stored. The PSNR values measured for these images are given separately for the three channels in Table 5. The

Table 4: Histogram graphs of chaotic pattern picture.

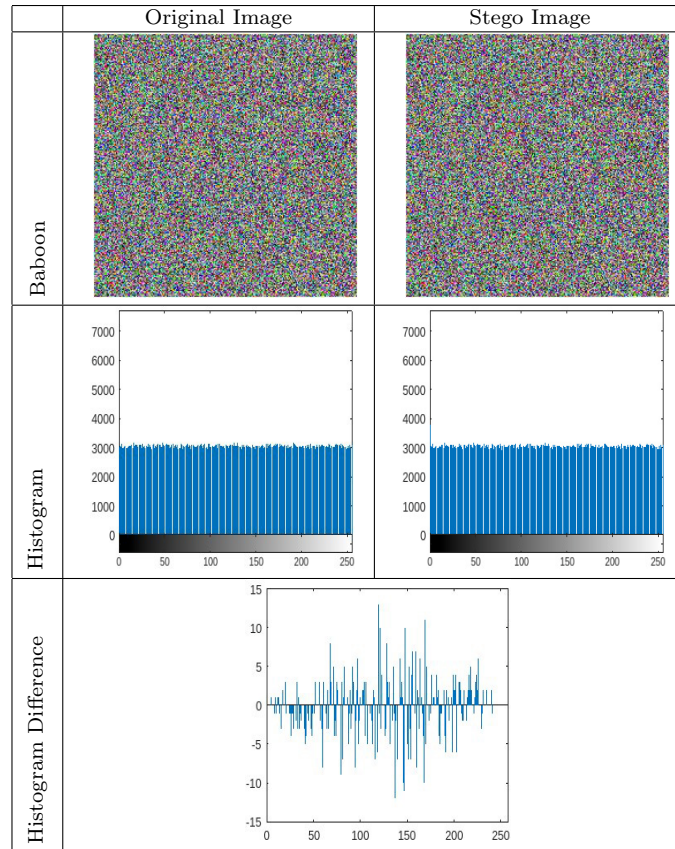


Table 5: In the model, measured PSNR values (chaotic pattern image)

PSNR	Channels		
	R	G	B
	Inf	Inf	69,22

results are lower than the Baboon image. The lower results are due to the higher amount of data hidden in the chaotic pattern image. By comparing the histogram plots of both figures, the differences between the histograms and the measured PSNR values, it is proved that indirect data transmission is more efficient with the proposed method.

A comparison of the proposed method with other studies is given in Table 6 and Table 7. The comparisons are based on studies using Baboon, Barbara and Lena images as cover objects. PSNR value is used as a criterion. Since only the blue channel of the color image is data hidden in the proposed method, the PSNR values of the blue channel are measured. The results for the red and green channels are infinite. In general, the infinite PSNR values in these channels contribute positively to the similarity of the cover image with the original image. The distortion is only due to the blue channel.

4 Limitations of the proposed model

The model ensures secure data transmission of periodically updated data. Examples of this data can be online exam system questions, system authorization data, credit card data, etc. This type of data only changes during update periods. The model provides highly secure data transmission for this type of data. However, it is not a suitable model for sending data that changes instantaneously from sender to receiver. The model has limitations for this type of data.

Table 6: Illustration of PSNR ratios of other working methods.




Works	Embedded bit	Images (512x512)		
		Baboon PSNR (db)	Barbara PSNR (db)	Lena (PSNR)
				
Elshare & El-Emam (2018)	Baboon = 582	49,9	59,8	50,4
	Barbara = 1300	49,8	62	51,2
	Lena = 1248	53,4	63,2	54,8
Wahab et al. (2021)	Lena = 22652	-	-	40,02
Peter et al. (2022)	Barbara = 71061	-	52,53	52,53
	Lena = 70965			
Kaur & Singh (2021)	Baboon = 47309	36,24	-	36,67
	Lena = 47309			
Chuang et al. (2021)	Lena = 775716	-	-	39,57

Table 7: Illustration of PSNR ratios of proposed method.

Works	Embedded bit		Images (512x512)								
			Baboon PSNR (db)			Barbara PSNR (db)			Lena (PSNR)		
Proposed method	Direct (Cover Image)	Indirect (Chaotic Pattern)	R	G	B	R	G	B	R	G	B
			4+(1+1) = 6	36100	Inf	Inf	71,29	Inf	Inf	72,26	Inf

5 Conclusion

A multilayer steganographic method is proposed for secure transmission of very important data that is periodically changed on the Internet. Steganographic data transmission methods are compared and evaluated based on high data transmission capacity and low image distortion rate. In this sense, the proposed method outperforms the existing methods. The advantage of the other works over the proposed method is that all the desired data can be sent to the receiver immediately. However, in these studies, data such as username, password, online exam system questions, credit card information, etc. may also be transmitted as in the proposed method. Therefore, the PSNR values of the proposed method and the methods proposed in other studies can be compared. In the proposed method, the pixel values of the red and green channels of the cover image are the same as the original image because no data on the pixel values are uploaded and the PSNR values of these channels are infinite. Therefore, only the PSNR value in the blue channel is compared with the PSNR values of other studies. According to the results in Table 2, although the PSNR values of the red and green channels of the proposed method are infinite, only the PSNR value in the blue channel is higher than the PSNR values in other studies. The PSNR values measured in other studies are in the range of 36.24-63.2. In the proposed method, it is infinite in the red and green channels and in the blue channel it is in the range of 71.26-71.29. These results prove that the distortion in the cover image is lower than the cover images used in other studies. The payload capacity of the method can be increased by adding data to the red and green channels.

The novelty of the proposed method, which is different from other works in the literature, can be expressed as follows. The model can be used for secure transmission of data that is updated periodically and needs to be sent continuously between the sender and the receiver at update intervals. The model guarantees a low distortion rate for the cover image throughout the data transmission. This is demonstrated by comparing the PSNR values obtained from the cover images used in the implementation developed for the model with the PSNR values of other works. Another novelty of the proposed method is the application of social engineering against attackers. With this application, the security level of data transmission is increased.

References

- Bai, J., Chang, C.C., Nguyen, T.S., Zhu, C., & Liu, Y. (2017). A high payload steganographic algorithm based on edge detection. *Displays*, 46, 42-51.
- Chuang, Y.H., Lin, B.S., Chen, Y.X., & Shiu, H.J. (2021). Steganography in RGB Images using adjacent mean. *IEEE Access*, 9, 164256-164274.
- Elshare, S., El-Emam, N.N. (2018). Modified Multi-level steganography to enhance data security. *International Journal of Communication Networks and Information Security*, 10(3), 509.
- Emirtekin, E., Karatay, M., & K1, T. (2020). Makale basligi. *Journal of Modern Technology and Engineering*, 5(3), 271-282.
- Hore, A., Ziou, D. (2010). Image quality metrics PSNR vs. SSIM. In *2010 20th International Conference on Pattern Recognition*, 2366-2369.
- Huan, X., Zhou, H., & Zhong, J. (2019). LSB based image steganography by using the fast marching method. *International Journal of Advanced Computer Science and Applications*, 10(3).
- Hussain, M., Wahab, A.W.A., Batool, I., & Arif, M. (2015). Secure password transmission for web applications over internet using cryptography and image steganography. *Int. J. Secur. its Appl.*, 9(2), 179-188.
- Jain, A., Nandakumar, K., & Ross, A. (2005). Score normalization in multimodal biometric systems. *Pattern recognition*, 38(12), 2270-2285.
- Jung, K.H. (2018). A survey of interpolation-based reversible data hiding methods. *Multimedia Tools and Applications*, 77(7), 7795-7810.
- Kaur, R., Singh, B. (2021). A hybrid algorithm for robust image steganography. *Multidimensional Systems and Signal Processing*, 32(1), 1-23.
- Klima, R., Klima, R.E., Sigmon, N., & Sigmon, N.P. (2018). *Cryptology: Classical and Modern*. CRC Press.
- Li, H., Li, J., Shokouh, G.S., & Samet, R. (2013). A low complexity algorithm for H. 264/AVC intra prediction. In *2013 International Conference on Cyberworlds* , 77-81.
- Li, P., Lu, A. (2018). LSB-based Steganography Using Reflected Gray Code for Color Quantum Images. *International Journal of Theoretical Physics*, 57(5), 1516-1548.
- Mohammed, H.A., Al Saffar, N.F.H. (yl). LSB based image steganography using McEliece cryptosystem. *Materials Today: Proceedings*.
- Peter, G., Sherine, A., Teekaraman, Y., Kuppasamy, R., & Radhakrishnan, A. (2022). Histogram Shifting-Based Quick Response Steganography Method for Secure Communication. *Wireless Communications and Mobile Computing*.
- Por, L.Y., Ku, C.S., Islam, A., & Ang, T.F. (2017). Graphical password: prevent shoulder-surfing attack using digraph substitution rules. *Frontiers of Computer Science*, 11(6), 1098-1108.
- Samet, R., K rođlu, T. (2019). Low Distortion Rate Steganographic Data Transmission Model. In *2019 4th International Conference on Computer Science and Engineering (UBMK)*, 444-448.

- Sheshasaayee, A., Sumathy, D. (2017). A framework to enhance security for OTP SMS in E-Banking environment using cryptography and text steganography. *In Proceedings of the International Conference on Data Engineering and Communication Technology* , 709-717.
- Shirali-Shahreza, M. (2007). Improving mobile banking security using steganography. *In Information Technology* , 885-887.
- Shirali-Shahreza, M.H., Shirali-Shahreza, M. (2007). New solution for password key transferring in steganography methods. *In Conference on Computational Intelligence and Multimedia Applications*, 123-125.
- Singh, R.K., Shaw, D.K. (2018). A hybrid concept of cryptography and dual watermarking (LSB DCT) for Data Security. . *International Journal of Information Security and Privacy (IJISP)*, 12(1), 1-12.
- Shyla, M.K., Kumar, K.S., & Das, R.K. (2021). Image steganography using genetic algorithm for cover image selection and embedding. *Soft Computing Letters*, 3, 100021.
- Subhedar, M. S., Mankar, V. H. (2018). Curvelet transform and cover selection for secure steganography. *Multimedia Tools and Applications*, 77(7), 8115-8138.
- Torvi, S.D., ShivaKumar, K.B., & Das, R. (2016). An unique data security using text steganography. *In Computing for Sustainable Global Development (INDIACom)*, 3834-3838.
- Vanmathi, C., Prabu, S. (2018). Image steganography using fuzzy logic and chaotic for large payload and high imperceptibility. *International Journal of Fuzzy Systems*, 20(2), 460-473.
- Wahab, O.F.A., Khalaf, A.A., Hussein, A.I., & Hamed, H.F. (2021). Hiding data using efficient combination of RSA cryptography, and compression steganography techniques. *IEEE Access*, 9, 31805-31815.
- Wu, H.T., Tang, S., Huang, J., & Shi, Y.Q. (2018). A novel reversible data hiding method with image contrast enhancement. *Image Communication*, 62, 64-73.
- Yalman, Y., Ertürk İ., & Çetin, Ö. (2014). Veri Gizleme. *Beta Publishing*, 13.