World of Computer Science and Information Technology Journal (WCSIT)
ISSN: 2221-0741
Vol. 4, No. 4, 42-49, 2014

Data Hiding in Digital Images using Cryptography and Steganography Techniques (CryptSteg)

Rashiq R. Marie
Computer Science Department
Faculty of Science and Information Technology, Zarka University
Zarka, Jordan

Abstract— The two common different techniques for securing data transmission are cryptography and steganography. Steganography is not intended to replace cryptography but rather to supplement it. If a message is encrypted and hidden with a steganographic method it provides an additional layer of protection and reduces the chance of the hidden message being detected. In this paper, two layers of security are used to secure the hidden information and to add more complexity for steganalysis. We combined schemes of cryptography with steganography in one system called CryptSteg for hiding secret messages. By CryptSteg we first encoded the secret message (plain-text) using chaotic stream cipher based on a secret key (crypto-key) then the encoded data was hidden behind a cover-image by changing Kth least significant bits (k-LSB) of cover-image pixels in random way which makes it superior to the conventional approach. A random-like sequence generated by a chaotic map is used as the reference for embedded positions. The randomness of the position of pixels on which the encrypted message to be embedded is decided by the stego- key. The two keys are shared between the sender and the receiver and they are encrypted and transmitted to the other party in a secured form. Experimental results show that the proposed CryptSteg system achieve a much higher visual quality as indicated by the high PSNR values of hiding secrete message bits in the image thus reduces the chance of the confidential message being detected and enables secret communication.. Moreover, using unauthorized keys (Crypto-key, Stego-Key) gets messages totally different from the original ones even the error keys are very close to the authorized one.

Keywords- Chaotic Stream; MSB; Stego-Key; Crypto-Key; LSB; Cover-image; Stego-Image.

I. INTRODUCTION

With rapid developments in the technologies of telecommunications especially the Internet and mobile networks have expanded the domain of information transmission, which in turn present new challenges for protecting the information from unauthorized access and use, the data integrity and confidentiality are required. Cryptography and steganography are well known and widely used techniques that manipulate information (messages) in order to cipher or hide their existence. These techniques have many applications in computer science and other related fields; they are used to protect e-mail messages, credit card information, corporate data, etc. Cryptography is the study of methods of sending messages in disguised form (not understood) so that only the intended recipients can remove the disguise and read the message. It protects information by transforming it into an unreadable format [1].

Plain-text refers to the message we want to send, while cipher-text is the disguised message. The process of converting a plain-text to a cipher-text is called encryption and the reverse process is called decryption. Encryption protects contents during the transmission of data from sender to receiver. Only those who possess a secret key can decrypt the cipher text into plain-text. Cryptanalysis is the science of 'breaking' or 'cracking' encryption schemes, i.e. discovering the decryption key.

Cryptography systems can be broadly classified into symmetric-key systems that use a single key that both the sender and the receiver have, and public-key systems that use two different keys(a public-key and a private-key), where the public key known to everyone and the private key that only the recipient of messages uses, (see Figure 1).

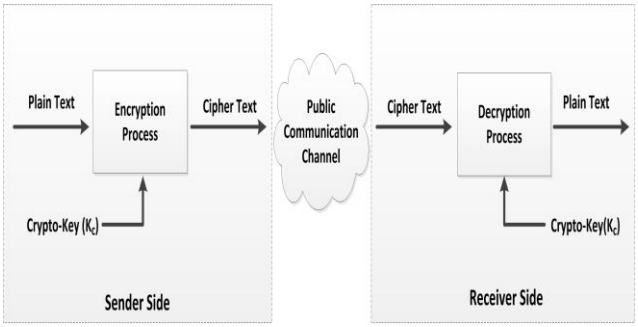

Figure1: Symmetric-key based cryptography

The word Steganography is basically coming from the Greek words stegos, meaning covered and graphia which means writing, is the art and science of hiding the fact that communication is taking place, by hiding information in other information. [2]. Both Steganography and cryptography are two integral parts of information security. As the encrypted data is itself evidence of the existence of valuable information that has been encrypted and it can be of worth attacking by an intruder. Steganography provides a significant advantage over cryptography alone in that messages do not attract attention to themselves, to messengers, or to recipients. For that both sciences can be combined to produce better protection of the message. In this case, when the steganography fails and the message can be detected, it is still of no use as it is encrypted using cryptography techniques.

In this paper, two layers of security are used to secure the hidden information and to add more complexity for steganalysis. We combined schemes of cryptography with steganography in one system called CryptSteg to secure the transmitted data over an insecure channel. By CryptSteg we first encoded the secret message (plain-text) using chaotic stream cipher based on a secret key (crypto-key) then the encoded data is hidden behind a cover-image using key-based steganography that changing a kth least significant bits (k-LSB) of cover-image pixels in random way which makes it superior to the conventional approach. A random-like sequence generated by a chaotic map is used as the reference for embedded positions. The randomness of the position of pixels on which the encrypted message to be embedded will be decided by a secret key (stego-key). More over the location of the bit into the pixel to embed the message is randomly determined. The two keys of the system are shared between the two party of communication, where they itself are encrypted and transmitted to the other party in a secured form.

II. OVERVIEW OF AKEY BASED STEGANOGRAPHY MODEL

Basically, the key-based steganography model as shown in Figure 2 consists of cover-object, secret message, embedding/extracting algorithm and key (stego-key). Cover-object is also known as carrier, which embeds the secret message and serves to hide its existence. Secret message or payload is the data that the sender wishes to remain it confidential. It can be plain text, cipher-text or anything that can be embedded in a bit stream. The cover-object with the secretly embedded message is then called the stego-object. This stego-object, the goal of steganographer, is then transferred to other end; there we have an extracting algorithm that used to extract the message from cover object. Many different objects have been employed to embed messages into for example image file, audio file, video file. Figure 3 shows the four main categories of the file formats that can be used for data hiding.

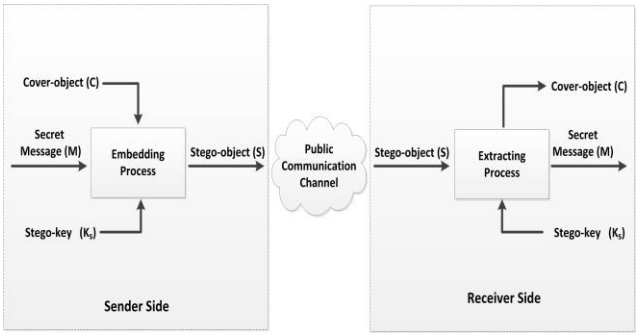

Figure 2: Symmetric-key based Steganography

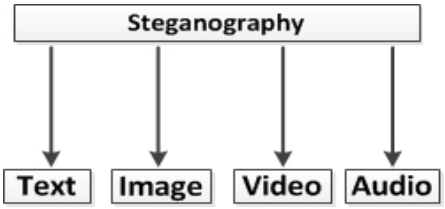

Figure 3: Cover-object types

Stego-key (Ks), which ensures that only the recipient who knows can extract the message from a cover-object. Stego-key is used to control the hiding process such as the selection of pixels or coefficients carrying the message, and to restrict detection and /or recovery of the embedded data to parties who know it. The usage of a Stego-key is important, because the security of a protection system should not be based on the secrecy of the algorithm itself, instead of the choice of a secret key. This makes the system more secure because the reader of the message must know the key in order to determine in which bytes the message bits are hidden.

The key must remain unknown to the attacker. However, if the cover image was known to the attacker, embedding the message in a random way, where the pixels to be substituted with information are selected randomly, would improve its security.

In the literature, several approaches for data hiding have been proposed [3, 4]. The most well known approach is the Least Significant Bit (LSB) that modifies the bits of a cover image based on the assumption that the LSB data are insignificant. LSB methods typically achieve high capacity. The most important property of a cover source is the amount of data that can be stored inside it, without changing the noticeable properties of the cover. Digital images are one of the preferred media for information hiding due to their high capacity and low impact on visibility and because people often transmit digital image over email and other communication media. Figure 4 shows the embedding of the secret message '0110' in the k-LSB of a grid of adjacent pixels of an 8-bit image.

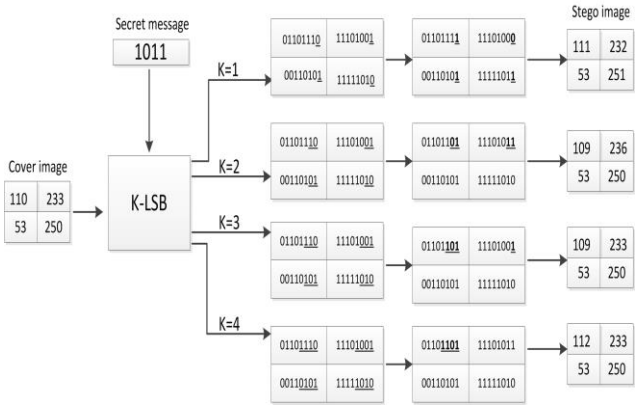

Figure 4: K-LSB Steganography Insertion

In the proposed system, we adopted two modifications on the LSB, where instead of laid out the secret message across the cover-image data sequentially, we include a chaotic-based pseudo-random number generator in the hiding algorithm so that the locations of the secret message bits to be embedded in the cover are randomized. We used a stego-key, that is shared between the communication parties, to extract initial values of chaotic maps. Moreover, in somehow, the number of message bits to be embedded in the LSB part of a nominated cover's pixel are depend on the most significant bits (MSB) part of the pixel. The stego-key chosen by the user gives randomization property which incorporates the diffusion property in cryptography that can resist steganalysis process.

III. PSEUDO RANDOM NUMBERS GENERATOR (PRNG) BASED ON CHAOTIC MAPS

A pseudo random number generator (PRNG) is a deterministic algorithm for generating a sequence of numbers that approximates the properties of random numbers. The sequence is not truly random in that it is completely determined by a relatively small set of initial values, called the PRNG's state, which includes a truly random seed.

Chaos word has been derived from the Greek, which refers to unpredictability and it is defined as a study of nonlinear dynamic system. Chaotic maps exhibit many interesting features such as sensitivity to the initial conditions: any slight change in the initial conditions yields widely diverged outcomes which makes impossible for long-term prediction and system parameter, which match with the requirements for a good crypto-system. The general form of one-dimensional chaotic map is

$$X_{n+1} = f(X_n), \quad n = 0, 1, 2, \ldots \tag{1}$$

With these maps we can produce a sequence $\{X_n : n = 0,1,2,3,....\}$ that are full chaotic and $f : I \to I,\ where \quad I = [0, 1]$.

This sequence is like noise but they are absolutely certain [5]. Because of these properties, they have been used to generate random numbers; with a tiny change in their initial values the generated sequences are completely different.

In this proposed system we used the generated chaotic floating values for two purposes, the first is to encrypt the plain message and the second is to embed the encrypted message bits in the cover image pixels. Two well known one-dimensional chaotic maps, Logistic map and Tent map, are combined together, to generate a chaotic sequence. Mathematically, the Logistic map is defined by:

$$X_{n+1} = 4 * r * X_n * (1 - X_n) \tag{2}$$

where $X_n \in (0,1)$, and $r \to 1$ is system parameter .When initial sate $X_0 \in (0,1)$, and $r = 1$ the system is in chaotic state,.

The iterative relation of the Tent map is given by:

$$Y_{n+1} = \begin{cases} \dfrac{Y_n}{a}, & Y_n \in [0, a] \\ \dfrac{1 - Y_n}{1 - a}, & Y_n \in (a, 1] \end{cases} \tag{3}$$

Where $Y_0 \in (0,1)$ an initial condition and a is a control parameter. The tent map is chaotic if $a \in (0,1)$ and $a \neq 0.5$.

The chaotic sequence used in the proposed system is defined by:

$$\{Z_n = (X_n^* + Y_n^*)\%1\}, \tag{4}$$

where

$$X_n^* = X_n / \max(X_n), \quad \text{and}$$

$$Y_n^* = Y_n / \max(Y_n) \tag{5}$$

The initial values X_0 and Y_0 are extracted from the system keys, namely, crypto-key (K_c) and stego-key (K_s) in each of the encryption phase and embedding phase, respectively.

Assume $KEY = (k_1 k_2 k_3 \ldots k_8)_{Ascii}$ represents either K_c or K_s, here k_i's are the alphanumeric characters (0-9 and A-F).

Let $X_0 = Y_0 = (\alpha + \beta)\%1$, where

$$\alpha = \sum_{i=1}^{8} \left(\frac{k_i}{256} \right), \quad \beta = \frac{k_1 \oplus k_2 \oplus k_3 \oplus \cdots \oplus k_8}{256} \tag{6}$$

IV. THE PROPOSED CRYPTSTEG SYSTEM

In this section, we explain how the proposed system hides information into a cover-image and retrieve the information from the stego-image without damaging it. Figure 10 shows the conceptual diagram of the overall processes of the system, in the first layer the user enter a crypto-key and a plain message, and then the message is encrypted using chaotic stream cipher.

The encrypted message will pass to the second layer, where in this layer, the user enters a stego- key and a cover image, and then the encrypted message is embedded in the cover image based on a stego-key. The two key are required to retrieving back the data that have been embedded inside the image.

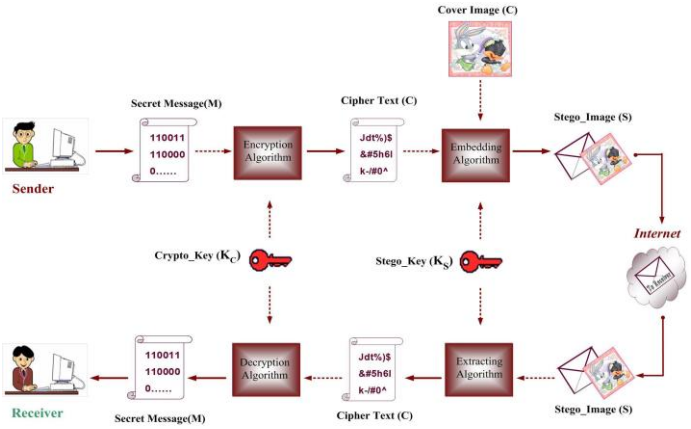

Figure 5: The framework for the CryptSteg System

There are two assumptions in the proposed system, the first one is sender and receiver are share the two secret keys which are firstly encrypted and send to the other party of communication in secure manner where without the secret keys; the data cannot be retrieved from the image. The second assumption is both sender and receiver either agrees on the length of message "OR" the length of the message is hidden with the message itself at some pre-specified locations which are known to both sender and receiver. In the following we explain the main processes that are implemented at sender side and receiver side, respectively:

-Encrypting and embedding the plain message in a cover-image.

- Extracting and decrypting the cipher message from the stego-image.

A. Sender Side: Encrypting and Embedding Processes

At the sender side there are two phases to secure the plain message using cryptography and steganography techniques: By these techniques the sender who wishing to send a secret message to the receiver will encrypt the secret message using chaotic encrypting algorithm and then conceal the existence of this message through embedding the cipher message in the cover-image. The embedding algorithm takes the encrypted message and scatters it randomly throughout a selected cover-image and produce the stego-image. The process of embedding depend on variable of bits K-LSB insertion which replaces variable least significant bits of the random selected pixel values with the encrypted information bits.

Algorithm of Encrypting the Secret Message

Input*: Plaintext (Secret Message), Crypto-key*

Output: *Cipher message*

Step1 *: Read a plain-text (secret message) and convert each character of the secret message to decimal number and then to a stream of binary bits (0's and 1's) , say,{B_n: n=1, 2, 3,...,N}, where N =8*size(text).*

Step2 *: Enter a Crypto-Key , say K_c , to generate a chaotic sequence a pseudo chaotic random numbers of size N as described previously,{Z_n: n=1,2,3,....N}.*

Step3*: Generate a stream of binary bits, say {W_n: n=1, 2, 3,...,N}, by applying the thresholding technique on {Z_n}*

$$W_n = \begin{cases} 1, \ if \ \ Z_n \geq 0.5 \\ 0, \ if \ \ Z_n > 0.5 \end{cases} \qquad (7)$$

Step4: *Apply the XOR operation between both the generated binary bits in step1 and Step 4, to get a stream of cipher binary bits:*

$$C_n = W_n \oplus B_n, \qquad n = 1,2,3,..., N \qquad (8)$$

Step5: *Forward the encrypted message, C_n, to the Embedding Algorithm.*

Algorithm of embedding the secret message

Input: *encrypted secret message, Cover Image, Stego-key.*

Output: *Stego-image.*

Step1: *Read the cover-image, say $C_{M\times N}$ and convert it to image vector of size 1×M.N*

Step2: *Enter a stego-key to generate a chaotic sequence,{ Z_n }, of size M.N that randomly locate the cover pixels positions nominated for holding the secret binary bits.*

Step 3: *Convert the sequence Z_n to a sequence of unsigned integer {g_n } and generate a row vector of indices{d_n},*

g_n *= floor (Z_n *10^{15}) , d_n=g_n%(M*N),* (9)

where n=0,1,2,....M.N-1

Step 4*: Permutate the indices vector {d_n} and discard the repetition if occurred then choose the image vector pixels corresponding to the generated indices vector where these selected pixels are nominated to hold the ciphered message bits.*

Step 5: *Convert the selected pixel intensity into 8 binary bits, say P=$b_1b_2b_3$ ...$b_5b_6b_7$, where b_k=0* or 1, *and extract the 4-bits MSB and 4-bits LSB*

Step 6: *Use the variable K-bits insertion into LSB part of the selected pixel that depends on the sum of 1's, say* $q = \sum_{i=4}^{7} b_i$, *in the MSB part of the selected pixel (i.e. the left four bits of the pixel), see Figure 5, as follows:*

$$K-LSB=\begin{cases}4\ if & q=0\ or\ \ q=4\\ 1\ if & q=1\\ 2\ if & q=2\\ 3\ if & q=3\end{cases} \qquad (10)$$

Step7: *Reshape the resultant image vector with embedded bits as stego-image, say* $S_{M\times N}$, *that will be sent via insecure channel to the receiver who will retrieve the hidden information using inverse steganography.*

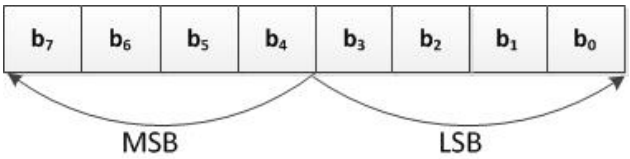

Figure 5: MSB and LSB of a pixel.

B. Receiver Side: Extracting and Decrypting Processes.

Upon reception of stego-image, the receiver firstly, extracts the hidden cipher message from the stego-image. Extracting the message from the stego-image includes inverse comparison to that used in embedding. The hidden data extraction is achieved by using the same random selection algorithm, based on the stego-key, to select the positions of the pixels where the secret bits had been embedded in the stego-image.

Finally, the original message is retrieved by decrypting the extracted cipher-message from the stego-image. The decryption algorithm demands the same crypto-key that was used by the sender in the encryption algorithm. The cipher-message has the same size as the plain-message due to the cryptosystem maps bring about a one-to-one correspondence between them.

V. EXPERIMENTAL RESULTS AND DISCUSSION

In this section, some experiments are carried out to prove the efficiency of the proposed scheme where it is simulated on Matlab 8 program. A set of 8- bit greyscale images of size 256 × 256 and of size 512×512 are used as the cover-images to form the stego-images. We tested these images with various sizes of data to be hidden.

With the proposed algorithm we found that the visual differences between the original cover-images (i.e. Figure 6 (a)-(f)) and the corresponding stego images with the encrypted hidden data (i.e., Figure7 (a) – (f)) can be hardly detected and the stego-images don't have a noticeable distortion on it as seen by Human Visual System (HVS). This is because there is a little changes of the pixel values and thus no significant difference. This is the most important of any steganography application, well achieved by using the HSV.

A. PSNR Analysis

To ensure that the distortion caused by embedding process is acceptable to (HVS), quality metrics are used to measure the different between the cover-image and the stego-image, MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ratio) are two common measurements for evaluating the quality of reconstructed signal.

MSE is the averaged pixel-by-pixel squared difference between the cover-image and the stego-image. Mathematically MSE can be derived as:

$$MSE=\frac{1}{N*M}\sum_{i=1}^{N}\sum_{j=1}[C(i,j)-S(i,j)]^2 \qquad (11)$$

where, M and N are the rows and columns respectively of the image, and C(i, j) and S(i, j) means the pixel value at the at position (i, j) in the cover-image and the corresponding stego-image, respectively.

The PSNR is expressed in dB's and can be calculated using MSE as

$$PSNR=10*\log(\frac{P^2}{MSE}) \qquad (12)$$

where, P is the peak signal value of the cover- image, $P=255.$

The larger PSNR indicates the higher the image quality i.e. there is only little difference between the cover-image and the stego-image. On the other hand, a smaller PSNR means there is huge distortion between the cover-image and the stego-image.

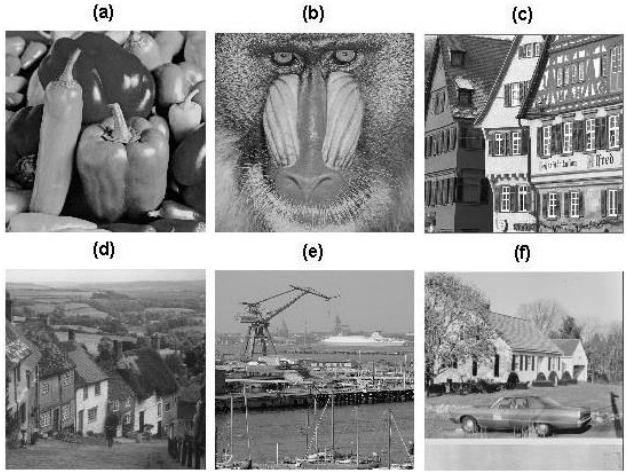

Figure 6: Original cover-images of size 512×512

TABLE 1: MSE AND PSNR VALUES VERSUS OF SECRET MESSAGE SIZE,(COVER IMAGE SIZE 256×256)

Cover Image (256×256)	Msg1 487 Bytes		Msg2 1012 Bytes		Msg3 2813 Bytes		Msg4 3564 Bytes		Msg5 5207 Bytes	
	MSE	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**
Peppers.bmp	0.146	56.478	0.294	53.445	0.775	49.240	0.995	48.154	1.476	46.440
Baboon.bmp	0.127	57.092	0.258	54.013	0.761	49.316	0.919	48.494	1.403	46.659
Boat.bmp	0.119	57.378	0.246	54.216	0.662	49.920	0.828	48.953	1.228	47.238
Houses.bmp	0.146	56.498	0.323	53.043	0.918	48.503	1.119	47.642	1.654	45.945
Sailboat.bmp	0.137	56.761	0.270	53.814	0.783	49.195	0.993	48.159	1.439	46.549
Car.bmp	0.129	57.013	0.126	53.963	0.7115	49.608	0.940	48.399	1.333	46.879
peppers.png	0.147	56.451	0.307	53.259	0.827	48.951	1.042	47.950	1.598	46.095
Airplane.png	0.156	56.189	0.309	53.226	0.842	48.878	1.083	47.783	1.590	46.114
Goldhill.png	0.126	57.109	0.244	54.259	0.687	49.756	0.863	48.769	1.292	47.016
Boy.png	0.110	57.684	0.227	54.562	0.608	50.290	0.790	49.154	1.197	47.348
Fruits.png	0.126	57.123	0.250	54.142	0.713	49.598	0.899	48.592	1.342	46.855
Cameraman.tiff	0.275	53.741	0.556	50.678	1.574	46.159	1.965	45.197	2.906	43.498
Bridge.tiff	0.130	56.985	0.269	53.818	0.824	48.972	0.997	48.142	1.468	46.462
Watch.tiff	0.175	55.713	0.382	52.316	1.082	47.790	1.308	46.965	1.975	45.174

TABLE 2: MSE AND PSNR VALUES VERSUS OF SECRET MESSAGE SIZE,(COVER IMAGE SIZE 512×512)

Cover Image (512×512)	Msg1 487 Bytes		Msg2 1012 Bytes		Msg3 2813 Bytes		Msg4 3564 Bytes		Msg5 5207 Bytes	
	MSE	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**	**MSE**	**PSNR**
Peppers.bmp	0.037	62.456	0.082	58.969	0.234	54.438	0.296	53.409	0.439	51.702
Baboon.bmp	0.032	63.075	0.065	59.943	0.186	55.447	0.233	54.450	0.337	52.861
Houses.bmp	0.043	61.812	0.095	58.330	0.267	53.859	0.332	52.909	0.491	51.218
Goldhill.bmp	0.030	63.301	0.058	60.441	0.166	55.926	0.211	54.869	0.314	53.157
Kiel.bmp	0036	62.502	0.078	59.179	0.214	54.826	0.258	54.003	0.393	52.185
Car.bmp	0.031	63.155	0.065	59.944	0.185	55.446	0.225	54.591	0.346	52.733

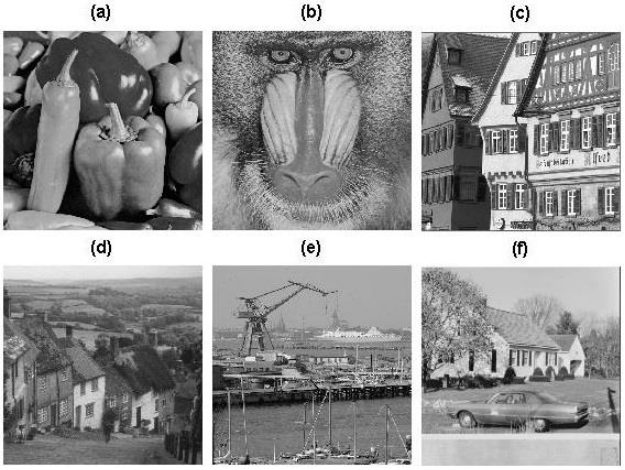

Figure 7: Stego-images of size 512 × 512 embedded with 5KB of the secret message

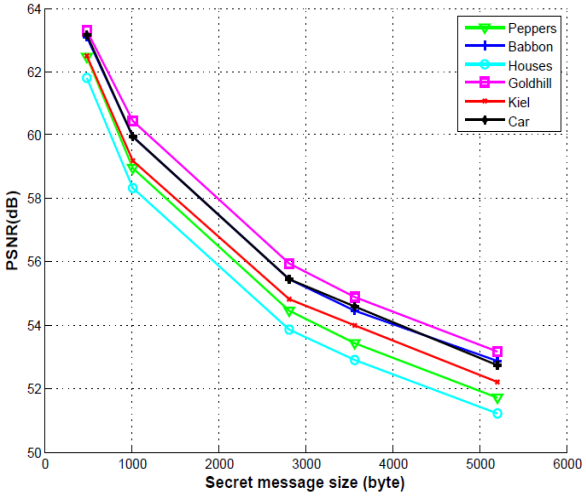

Figure 8: Message size vs. PSNR values, (cover-image 512×512)

Table 1 and Table 2 give the measured values of MSE and PSNR of different types of cover –images of size 256×256 and 512×512, respectively. It is observed that when the payload increases the MSE increases and this affects the PSNR inversely and for all cover- images PSNR is greater than 50, this indicates good performance of the proposed system. As can be seen in Figure 8 and Figure 9, the reduction in PSNR is very slight as compared with the increases in the size of embedded message and this suggests that the quality of the image remains almost constant when the message size increases. It means that the stego-images created with proposed system can survive the common -cover-carrier attack.

B. Histogram Analysis

An image's histogram explains how the image's pixels are distributed by graphing the number of pixels at each colour intensity. We have calculated and analysed the histograms of the given cover-images and their corresponding stego-images. Figure 10 and Figure 11 show the histogram of the cover-images given in Figure 6 and Figure 7, respectively. As we see by comparing the histograms of cover-image and stego-image, we found that they are the approximately the same with very less change in the stego-image. As the size of embedded message increases, this change in the stego-image also increases accordingly, see Figure 12.

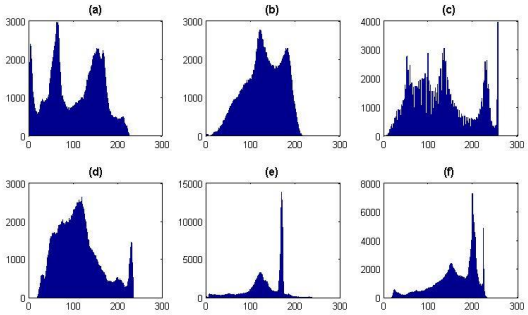

Figure 10: Histograms of cover-images of size 512×512

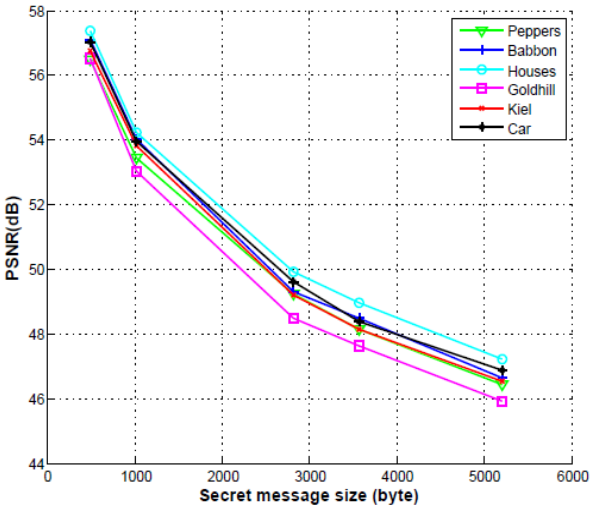

Figure9: Message size vs. PSNR values,(cover-image 256×256)

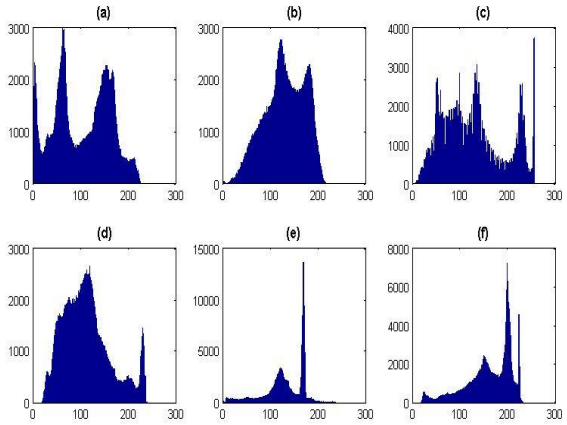

Figure 11: Histograms of stego-images of size 512×512

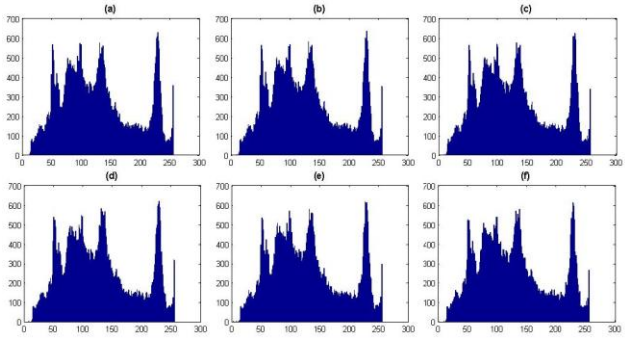

Figure 12: (a) Histogram of the cover-image (Houses.bmp); (b)-(c) Histograms of the corresponding stego-images with message sizes (byte): 487, 1012, 2813, 3564 and 5207.

IV. CONCLUSION AND FUTURE WORK

Steganography has its place in security. It is not intended to replace cryptography but supplement it. In this paper we introduced an idea to enhance the security of data transmission by combining the two techniques in one system called CryptSteg. Here message was first encrypted using chaotic maps and then embedded in an image file with help of steganographic methods. In this case, when the steganography fails and the message can be detected, it is still of no use as it is encrypted using cryptography techniques. The system security is further enhanced by using crypto-key and stego-key as two external keys shared between two parties of communication. The two keys are, respectively, applied to the system during encryption and embedment of the message into the cover-image.

Experimental results show that the proposed CryptSteg system achieve a much higher visual quality as indicated by the high PSNR values of hiding secrete message bits in the image thus reduces the chance of the confidential message being detected and enables secret communication.. Moreover, using unauthorized keys (Crypto-key, Stego-Key) gets messages

totally different from the original ones even the error keys are very close to the authorized one. Certainly, the time complexity of the proposed system increases but at the same time the security achieved at this cost is well worth it. Thus we expect that proposed system as a good alternative to other technique because of the high level of security. For future work we may compress the data before encryption as further securing and use other type of cover-object to hiding the data.

ACKNOWLEDGMENT

This research is funded by the Deanship of Research and Graduate Studies in Zarqa University /Jordan.

REFERENCES

[1] A. Menezes, P. Oorschot, S. Vanstone, and A. J. Menezes, "Handbook of Applied Cryptography", CRC Press, Boca Raton, FL, 1997.

[2] F. Petitcolas, R. Anderson, M. Kuhn, "Information Hiding-A survey", Proc. of the IEEE, vol. 87, no. 7, pp. 1062-1078 , July, 1999.

[3] S. Katzenbeisser S. and F. Petitcolas , "Information Hiding Techniques for Steganography and Digital Watermarking", Artech House Press, 2000.

[4] B Li, J He, J Huang and YQ Shi., "A Survey on Image Steganography and Steganalysis", Journal of Information Hiding and Multimedia Signal Processing, vol. 2, no. 2, pp. 142-172, 2011.

[5] K. Alligood, T. Sauer, J. Yorke., "Chaos: An Introduction to Dynamical Systems", Textbooks in Mathematical Sciences, Springer, New York, NY, USA, 1997.

[6] N. Johnson, and S. Jajodia, "Exploring steganography: Seeing the unseen", IEEE Computer Magazine, pp. 26-34, 1998.

[7] N. Provos N. and P. Honeyman, "Hide and seek: An introduction to steganography", IEEE Security and Privacy, vol. 1, no.3, pp. 32-44, 2003

[8] R. Chandramouli R., N. Memon, "Analysis of LSB Based Image Steganography Techniques", IEEE pp. 1019-1022, 2001.

[9] K Babu, S. Kumar and A. Babu, "A Survey on Cryptography and Steganography Methods for Information Security", International Journal of Computer Applications", pp. 13-17, vol. 12, no. 2, 2010.

[10] [10] L. Kumar, "Novel Security Scheme for Image Steganography using Cryptography Technique", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, No. 4, 2012.

[11] [11] V. Patidar V. and K Sud, "A pseudo Random Bit Generator base on Chaotic Logistic Map and its Statistical Testing ,"Informatica, vol. 33, pp. 441-452, 2009.

AUTHOR PROFILE

Rashiq R. Marie received the B.Sc. degree in Statistics and Computer Science from Yarmouk University, Jordan in 1989, the M.Sc. degree. in Mathematics from Jordan University, Jordan in 1992 and the Ph.D. in Computer Science from Loughborough University , UK in 2007. He is an assistant professor in Computer Science department at Zarka University/Jordan. His research includes Cryptography, Steganography, Digital watermark and Fractal Analysis of digital signals.