

## Chaos based Zero-steganography algorithm

Muhammad Bilal · Sana Imtiaz · Wadood Abdul ·  
Sanaa Ghouzali · Shahzad Asif

Published online: 20 March 2013  
© Springer Science+Business Media New York 2013

**Abstract** Conventional steganography focuses on invisibility and undetectability, as the main concern is to make the algorithms immune to steganalysis. *Zero-steganography* is an imperceptible and undetectable data hiding technique as no change is made to the cover, hence not requiring any steganalysis. The proposed algorithm hides the payload based on certain relationship between the cover image, chaotic sequence and the payload, instead of directly embedding payload into the cover image which often leaves tell tale signs of steganography. Moreover, use of chaotic map in the process of data hiding provides improved security. Survivability of the proposed algorithm is analyzed against JPEG compression, noise and low-pass filtering attacks. Imperceptibility analysis reveals that the proposed algorithm is totally imperceptible regardless of the payload length. The proposed algorithm is also analyzed for security and is found to be secure, in highly compromised scenarios,

---

This work of Wadood Abdul and Sanaa Ghouzali was supported by the Research Center of College of Computer and Information Sciences and the Deanship of Scientific Research, King Saud University, under grant RC120911. The authors are grateful for this support.

---

M. Bilal (✉) · S. Imtiaz  
Department of Electrical Engineering, COMSATS Institute of Information Technology,  
Islamabad, Pakistan  
e-mail: mbilal\_ce@live.com

W. Abdul  
Department of Computer Engineering, College of Computer and Information Sciences,  
King Saud University, Riyadh, Kingdom of Saudi Arabia

S. Ghouzali  
Department of Information Technology, College of Computer and Information Sciences,  
King Saud University, Riyadh, Kingdom of Saudi Arabia

S. Asif  
Department of Electrical and Computer Engineering,  
Center for Advanced Studies in Engineering, Islamabad, Pakistan

where all except one of the key components required to retrieve payload are known to the adversary.

**Keywords** Steganography · Steganalysis · Zero-steganography · Imperceptibility · Chaos based steganography

## 1 Introduction

The practice of sharing information secretly has been going on for ages. One of the earliest methods was to etch the message on a messenger's head, wait for the hair to grow again, and then send him to the intended recipient. The recipient had to shave off the messenger's head to read the secret message. This was very tedious and time-taking process and could not be made use of much successfully during wars, except for few instances. A better approach was to use wooden tablets covered in wax. The technique was to remove the wax, write the message on the wood, and recover it with wax. Advancement led to development of better techniques such as the invisible ink and null ciphers, but the latter mostly raised suspicion as they sounded strange. Hence, almost all the methods used for secrecy were either detectable or at least, suspicious [16].

With the arrival of computers, cryptography took over all the conventional secret communication methods. It aimed at encrypting–replacing each letter with a combination of symbols–the data that is to be kept safe, thus making it incomprehensible for an adversary. However, encryption was not concerned with hiding the existence of secret communication. This approach leads to suspicion which in turn might make data transmission unsuccessful.

Information hiding techniques resolved all the above mentioned issues and are categorized in two branches: steganography and watermarking. Attributes of the information hiding techniques include (i) imperceptibility, (ii) survivability, (iii) capacity, and (iv) security. Steganography, in general, is the process of hiding information in the cover medium such that it is undetectable, which makes it a prime candidate for hidden communication. Image steganography is carried out by manipulating and altering image pixels to hide the data [6, 30]. In addition to spatial domain, image steganography is also carried out in several transform domains [19].

A generic description of the steganographic process is as follows:

$$\text{cover data} + \text{secret data} \xrightarrow{\text{embedding algorithm}} \text{stego data} \quad (1)$$

Nowadays, steganography is often used in conjunction with cryptography for dual protection of data. As the encrypted information is hidden using steganography, the adversary has to first find the hidden information–which itself is a difficult task–and then decrypt it to find the secret data.

With the rapid expansion in information technology, problems such as piracy, counterfeit and violation of copyrights increased dramatically. One of the proposed solutions was to have a digital signature in the data, so that the rightful owner may prove his ownership when required [17]. Watermarking - a technique to embed digital signature in data–served the purpose. However, in case of watermarking,

non-visibility was not a necessary criterion for every scenario. High capacity was also not needed as signatures are usually small. The only essential requirement was to embed the signature in such a way that it could be robust against removal or destruction attempts; hence robustness and security were the attributes of prime importance. In general, watermarking secures digital data against copyright violation and counterfeit issues, by embedding a digital signature with minimal distortion in cover data. But, in case of military and medical images, this minute distortion was not permissible.

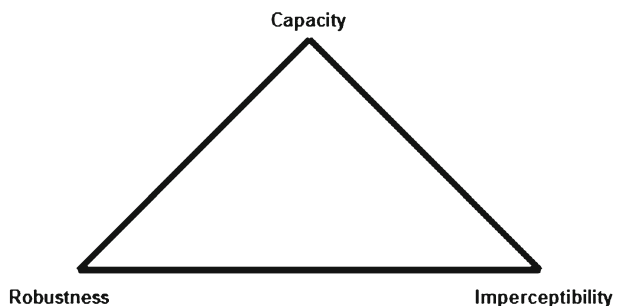
In case of steganography, survivability has not been an important attribute so far, rather more important is the imperceptibility and capacity with secure delivery. These requirements could be better understood by the visual requirement model given in Fig. 1 [15]. While watermarking algorithms are primarily concerned with the robustness of the embedded watermark, steganography algorithms are concerned with imperceptibility and capacity of the algorithm.

Cao *et al.* proposed a digital image zero-watermarking method based on Discrete Wavelet Transform (DWT) and chaotic modulation [5]. Zero-watermarking, as the name suggests, made no changes to the cover and solved the problems caused due to distortion. The employed technique was the development of a key, based on a relationship between watermark–digital signature–and cover image. Zero-watermarking is a relatively new field and is being explored to solve problems that were not addressable with former watermarking techniques. Inspired by zero-watermarking, the proposed work presents *Zero-steganography*—a steganography scheme that is totally imperceptible and hence, undetectable.

The proposed algorithm makes use of logistic map to generate a chaotic matrix. Stego-keys are developed based on certain relationships between cover image, payload and chaotic matrix. Payload is retrieved by means of a relationship between cover image, stego-keys and chaotic matrix. Cover image is not modified during the whole steganographic process, therefore the name *Zero-steganography*. Hence, instead of directly embedding payload into the cover image that leaves signs of steganography [7, 20, 23], the proposed algorithm performs the steganographic process without any modifications in the cover image.

Rest of the paper is organized as follows: Section 2 gives a brief introduction of chaos and chaotic map used in the proposed algorithm. The proposed algorithm is presented in Section 3. In Section 4, the algorithm is analyzed for survivability, imperceptibility and security. Finally, conclusions are drawn in Section 5.

**Fig. 1** Requirement model



## 2 Chaos and one-dimensional chaotic maps

Chaos represents a state of disorder. In scientific sense, there is much more to chaos than just disarray [37]. Instead of random disorder, the chaos that is useful for scientific applications is deterministic. Contrary to the intuition, it means that such kind of chaos can be generated using mathematical equations. However, the initial parameters required to reproduce the same chaotic sequence must be exact. Any slight variation in the initial parameters results in a totally different chaotic sequence.

It is rather difficult to define chaos completely. *Williams Garnett* defines it as, “Chaos is sustained and orderly-looking long term evolution that satisfies a certain special mathematical criteria and that occurs in deterministic nonlinear systems” [37].

Logistic map is arguably the simplest nonlinear difference equation and appears in many contexts [24]. It was introduced by *P.F. Verhulst* in order to simulate the growth of population in a closed area [32]. Logistic map belongs to the family of first order difference equations and can be represented mathematically as:

$$X_{k+1} = \mu X_k(1 - X_k) \quad (2)$$

where the system parameter  $\mu \in [0, 4]$  and initial condition  $X_0 \in (0, 1)$ .

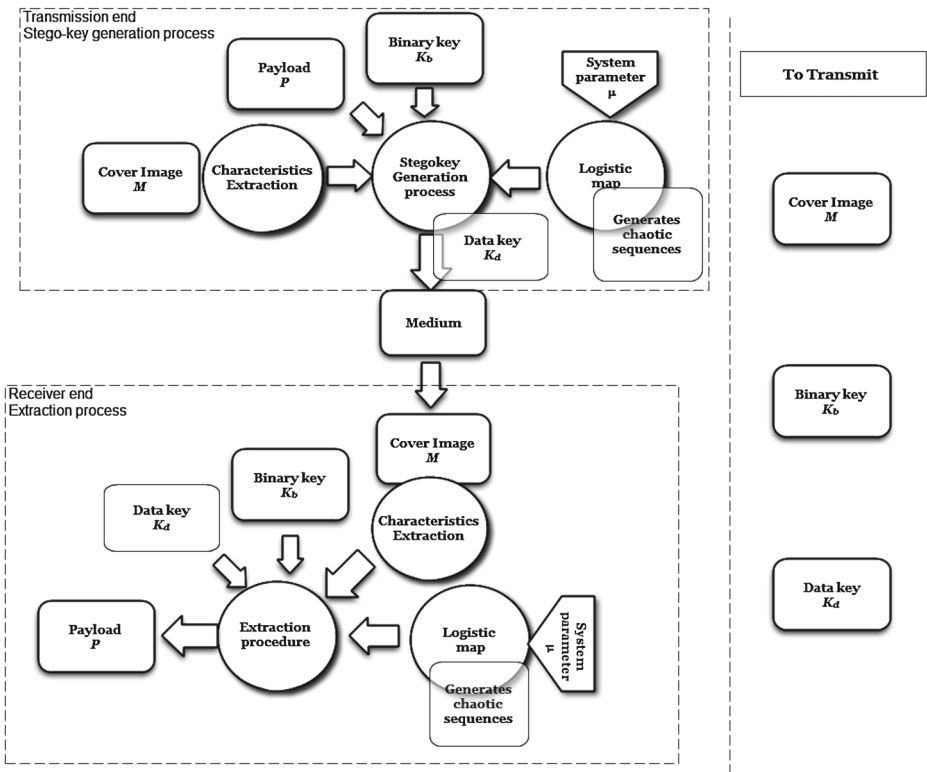
The logistic map behaves chaotic with  $\mu \in (3.5699456, 4]$  [27]. The chaotic sequences  $\{X_k; k = 1, 2, 3, \dots\}$  generated by this one-dimensional logistic map are unique for every system parameter within this range.

Logistic map is considered as one of the simplest chaotic maps. As it is highly sensitive to system parameter  $\mu$ , any two chaotic sequences produced using slightly different system parameters are uncorrelated statistically. The proposed algorithm makes use of logistic map to generate a chaotic matrix. Both the system parameter and initial condition are to be known to the intended recipient, in order to regenerate the same chaotic sequence on the receiver side. Early works in digital watermarking which used logistic map are [39, 40].

More complex chaotic maps like the Hénon map [12] and Lorenz attractor [22] can be used in order to directly generate two-dimensional and three-dimensional chaotic sequences respectively. We will now present the proposed algorithm in detail.

## 3 Proposed algorithm

Given the advantage of steganography over cryptography, and the lack of imperceptible yet robust steganography algorithms in the spatial domain, the proposed algorithm adopts a different approach to achieve steganography. Figure 2 shows an overview of the procedure involved in the proposed algorithm. The technique is to operate characteristics extraction procedure on the cover image. A pseudo random binary key to denote mathematical operations is generated. These mathematical operations are performed between payload and cover image characteristics, and the obtained data is used to construct an intermediate key. Then, the same mathematical operations are performed between intermediate key and chaotic sequence and resultant stego-key is stored as data key. Since the secret information resides in the relationship between cover image and keys, no change is made to the cover image during the whole procedure. Hence, making the proposed algorithm imperceptible



**Fig. 2** Block diagram of the steganographic process

and undetectable. We will now describe the procedure for stego-key generation and payload extraction in detail.

### 3.1 Stego-key generation

For a secret payload  $P$  of dimensions  $m \times n \times o$  and a cover image  $M$  with  $i$  rows,  $j$  columns and  $k$  depth, the stego-key generation procedure is as follows:

1. Characteristics of cover image  $M$  are extracted and stored as  $\overline{M}$ .
2. A binary key  $K_b$  of dimensions  $m \times n \times o$ —same as that of payload—is generated using a pseudo random process. The key  $K_b$  denotes mathematical operations that are to be performed between corresponding payload pixel and  $\overline{M}$ . The resultant data is stored as intermediate key  $K_i$ .

$$K_i(x, y, z) = \begin{cases} P(x, y, z) + \overline{M}(x, y, z), & \text{if } K_b(x, y, z) = 1 \\ P(x, y, z) - \overline{M}(x, y, z), & \text{if } K_b(x, y, z) = 0 \end{cases} \quad (3)$$

where  $x = 1, 2, 3, \dots, m$ ;  $y = 1, 2, 3, \dots, n$  and  $z = 1, 2, 3, \dots, o$ .

The procedure of generating  $K_i$ , (3) is repeated till all the locations of  $P$  have been catered for.

3. A chaotic sequence  $C$  is generated using (2), having the same dimensions as  $\overline{M}$ . The logistic map is provided with system parameter  $\mu$  and initial condition  $X_0$  to generate the sequence.  
System parameter  $\mu$  and initial condition  $X_0$  are to be known only to the sender and intended recipient, so only they may be able to generate the same chaotic sequence.
4. The chaotic sequence is then scaled and rounded off, such that a sequence of uniformly distributed integers in range  $[a,b]$  is obtained.
5. Now,  $K_i$  is operated with  $C$  in accordance with the operation defined by  $K_b$  as:

$$K_d(x, y, z) = \begin{cases} K_i(x, y, z) + C(x, y, z), & \text{if } K_b(x, y, z) = 1 \\ K_i(x, y, z) - C(x, y, z), & \text{if } K_b(x, y, z) = 0 \end{cases} \quad (4)$$

where  $K_d$  denotes the data key.

At the end of step 5, we will have two keys— $K_b$  and  $K_d$ —and cover image that are to be transmitted. The keys are to be transmitted through a secure channel.

### 3.2 Extraction

For successful retrieval of payload on the receiving side: cover image, secret keys, system parameter and initial condition for generating chaotic sequence are needed. Chaos is extremely sensitive to the system parameter  $\mu$ ; any minor difference in its value generates an entirely different chaotic sequence. Therefore, for accurate retrieval of the payload, exact system parameter  $\mu$  is required. Steps involved in extraction process are:

1. Characteristics of cover image  $\overline{M}$  are obtained by using the same characteristic extraction method as in stego-key generation procedure. Also,  $C$  is generated with system parameter  $\mu$  and initial condition using (2).
2. Now, information of payload and cover image relationship— $K_i$ —is retrieved from  $K_d$  using the binary key,  $K_b$ .

$$K_i(x, y, z) = \begin{cases} K_d(x, y, z) - C(x, y, z), & \text{if } K_b(x, y, z) = 1 \\ K_d(x, y, z) + C(x, y, z), & \text{if } K_b(x, y, z) = 0 \end{cases} \quad (5)$$

Step 2 is repeated till all the indices of  $K_b$  are catered for and stored in  $K_i$ .

3. Finally, characteristics of cover image are separated from secret data present in  $K_i$  using  $K_b$  to recover the payload  $P$ .

$$P(x, y, z) = \begin{cases} K_i(x, y, z) - \overline{M}(x, y, z), & \text{if } K_b(x, y, z) = 1 \\ K_i(x, y, z) + \overline{M}(x, y, z), & \text{if } K_b(x, y, z) = 0 \end{cases} \quad (6)$$

Step 3 is repeated till the payload matrix is completely retrieved.

We will now analyze the survivability, imperceptibility and security of the proposed algorithm in different scenarios.

## 4 Results and discussion

As imperceptibility and undetectability are the key issues that need to be addressed in steganography, steganography algorithms are evaluated for survivability against

steganalysis and statistical attacks [9, 10, 14, 21, 36, 38]. Steganalysis generally uses following techniques to identify steganography: i) changes between color values, ii) analysis of exaggerated noise, iii) stego-image size analysis for detecting abnormal changes in stego-image, and iv) existence of obvious and repetitive patterns [14]. However, in our case, since no changes have been made to the cover image, the proposed algorithm is undetectable by steganalysis. For our experiments, averaging has been used as a characteristic extraction method. Averaging is carried out on  $M$  using an averaging window of size  $p \times q$  and is stored as  $\overline{M}$ :

$$\overline{M}\left(\frac{x}{p}, \frac{y}{q}, z\right) = \left[ \frac{\sum_{a=x-(p-1)}^x \sum_{b=y-(q-1)}^y M(a, b, c)}{p \times q} \right] \tag{7}$$

where  $x = p, 2p, 3p, \dots, i$ ;  $y = q, 2q, 3q, \dots, j$  and  $z = c = 1, 2, \dots, k$ . In our analysis, we have selected  $p = q = 8$ .

As far as payload capacity is concerned for the tested scenario, the proposed algorithm can take a maximum payload up to the size of averaged cover image given by:

$$Cap_{max} = \frac{1}{(S_{AW})^2} * S_M \tag{8}$$

Where  $Cap_{max}$  is the maximum payload capacity,  $S_M$  is the size of cover image in bits and  $S_{AW}$  is the size of averaging window matrix,  $S_{AW} \in \{1, 2, 3, \dots\}$ .

Comparison between capacity of proposed scheme with several other steganography schemes is presented in Table 1. Though comparison between capacity of conventional steganography and proposed scheme is hard to do as in conventional steganography algorithms capacity is directly linked to the imperceptibility of the algorithm. We have presented a comparison table based on the capacity and imperceptibility trade-off. The results presented here can be found in [8]. All of the conventional steganography methods have been subjected to blind steganalysis [9] and detection of presence of steganography has been evaluated using the detection reliability  $\rho$ . Detection reliability is normalized so that  $\rho = 1$  means perfect detection and  $\rho = 0$  means no detection at all.

**Table 1** Comparison based on capacity and imperceptibility trade offs between: F5, F5 without matrix embedding (1,1,1) [36], OutGuess 0.2 (OG) [28], Model based Steganography without and with deblocking (MB1 and MB2 respectively) [31], Perturbed Quantization [8] and Chaos based Zero-steganography scheme (U = unachievable capacity)

Capacity (bpp/bpc)	F5	F5_111	OG	MB1	MB2	PQ	CBZS
0.05	0.241	0.645	0.879	0.220	0.163	~ 0	0
0.1	0.539	0.922	0.993	0.415	0.310	0.048	0
0.2	0.956	0.996	0.991	0.704	0.570	0.098	U
0.4	1.000	1.000	U	0.938	0.824	0.174	U
0.6	1.000	1.000	U	0.983	U	U	U
0.8	1.000	1.000	U	0.992	U	U	U

The algorithms under consideration are F5, F5 without matrix embedding (1,1,1) [36], OutGuess 0.2 (OG) [28], Model based Steganography without and with de-blocking (MB1 and MB2, respectively) [31], Perturbed Quantization scheme [8] and proposed *Zero-steganography* scheme. The capacity of all of the algorithms except the proposed scheme is measured in bits per non-zero DCT coefficient (bpc), while the metric of capacity for the proposed scheme is bit per pixel (bpp). The maximum capacity for the proposed algorithm would be 0.125 bpp for an  $8 \times 8$  window characteristic extraction method. All of the other algorithms under consideration have maximum capacity that is higher than the maximum capacity of the proposed algorithm. However, the increased capacity comes at the cost of imperceptibility as the blind steganalysis technique can detect these algorithms if high embedding rate is used. As shown in Table 1 even for capacity as low as 0.1 bpc or 0.05 bpc the detection reliability of the blind steganalysis technique stays well above 0. The only exception is the Perturbed Quantization method, for which  $\rho \approx 0$  for capacity of 0.05 bpc, meaning that for 0.05 bpc capacity, it is completely imperceptible against this particular steganalysis technique.

Even though the maximum capacity of proposed scheme is lower than many conventional steganography algorithms, the immunity of the proposed scheme against any steganalysis makes it sufficiently capacious while totally imperceptible. Note that using any other characteristics extraction procedure will result in different capacity and survivability, depending upon the methodology used for characteristics extraction and the type of attack [1]. The analyses of the proposed algorithm for survivability, imperceptibility, and security are given in the following Sections:

#### 4.1 Survivability

Survivability is the measure that indicates the robustness of steganography algorithm against an attack by the adversary to limit the usability of the steganography algorithm. There are several attacks in the literature that are used to render the steganography algorithms unusable to some extent [15, 26, 35]. Survivability or robustness is not a primary concern in steganography algorithms, however, a great deal of work has been done in order to improve robustness in digital watermarking algorithms [3, 13, 18]. In order to evaluate the survivability of the proposed algorithm, we tested it against common attacks such as low-pass filtering, JPEG compression and noise. Survivability is measured in terms of Bit Error Rate (BER), defined as:

$$\text{BER} = \frac{B_c}{B_t} \quad (9)$$

Where  $B_c$  is the number of corrupted payload bits upon extraction and  $B_t$  is the total number of payload bits.

For the purpose of our tests a set of  $512 \times 512$  and  $256 \times 256$  color images—shown in Figs. 3 and 4, respectively—were used as cover images. Whereas, a  $64 \times 64$  and  $32 \times 32$  color image was used as payload for the former and the latter set, respectively—shown in Fig. 5.

At first, respective keys were generated for each of the cover images, then the cover images were attacked and extraction was carried out from the attacked cover images. BER for extracted payload was computed by comparison between original payload bits and extracted payload bits. To further reduce the BER produced as





**Fig. 3** Test images of size  $512 \times 512$

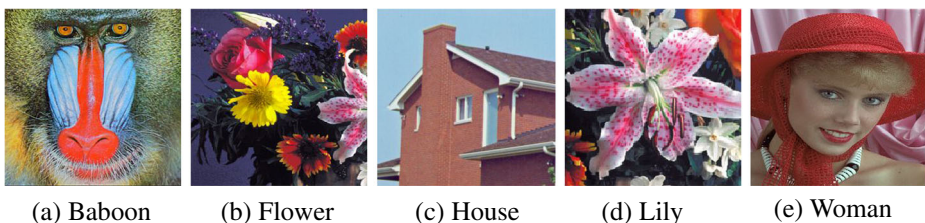
a result of attacks, Anderson and Petitcolas [2] suggested use of redundancy. We used the simple form of redundancy—data repetition—with a repetition factor of 24 for error correction and as a result, improved, lower BER was obtained. It is to be noted here that the payload capacity will decrease by a factor of 24 when we use repetition technique.

Combined results for survivability, with and without repetition, are plotted against different attack strengths for selected attacks. The following plots represent averaged results of all five cover images for selected payload image in terms of BER. The results for scenarios with data repetition use a random payload matrix instead of the payload image. Detailed analysis of the results is presented in the Sections 4.1.1, 4.1.2, and 4.1.3.

#### 4.1.1 Low-pass filtering attack

For the low-pass filtering attack, we used the averaging filter technique. Cover images were attacked with averaging filters of sizes  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  and  $9 \times 9$ —where  $9 \times 9$  represents the highest low-pass filtering attack strength. BER for retrieved payload was calculated for each attack strength. Figure 6 illustrates the average BER obtained with all the cover images for low-pass filtering attack for both cases—with and without repetition of payload. Subsequent results were computed accordingly for each attack strength.

Without repetition, in case of the highest attack strength, less than 14 % bits were corrupted and rest of the payload was extracted accurately. However, use of repetition reduced BER appreciably, as it dropped down to 2.2 %, for a filter size of  $3 \times 3$ . Similarly, for the worst attack case of filter size  $9 \times 9$ , it remained up to about 9 %, which was 6 % less than the previous case. As a whole, the proposed algorithm



**Fig. 4** Test images of size  $256 \times 256$

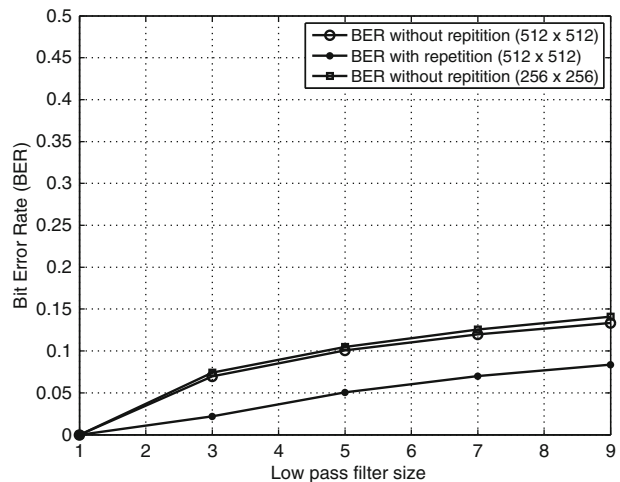
**Fig. 5** Payload image (Coin)

is fairly robust to low-pass filtering attack, promising an accurate payload retrieval rate of more than 90 %, even in the worst scenario using error-correction.

Table 2 provides the comparison between survivability of a Chaos based DCT steganography algorithm [34] and the proposed *Zero-steganography* scheme for Gaussian low-pass filter attack. The results show that for both values of variance the survivability of the proposed algorithm is better than the Chaos based DCT steganography scheme [34]. As the attack strength increases the difference between the corresponding BER values for each algorithm also increase. The proposed algorithm performs much better under high attack strength as compared to the DCT algorithm.

#### 4.1.2 Noise attack

The proposed algorithm is analyzed for two noise attacks, Additive White Gaussian Noise (AWGN) and salt and pepper noise. AWGN damages the data by linear addition of white noise with a constant spectral density and a Gaussian distribution of amplitude. The attack makes use of two parameters, mean  $m$  and variance  $\sigma^2$ . Mean  $m$  is 0 and  $\sigma^2$  varies from 0 to 1, with  $\sigma^2 = 1$  giving highest level of noise. For our tests, we used  $\sigma^2$  in the range 0.1 – 1 with a step size of 0.1. As illustrated

**Fig. 6** Low-pass filtering attack

**Table 2** Comparison between survivability of Chaos based DCT steganography [34] and Chaos based *Zero-steganography* (CBZS) scheme for Gaussian low-pass filter attack

Gaussian variance	Chaos based DCT algo	CBZS
0.5	0.031	0.023
1.0	0.088	0.050

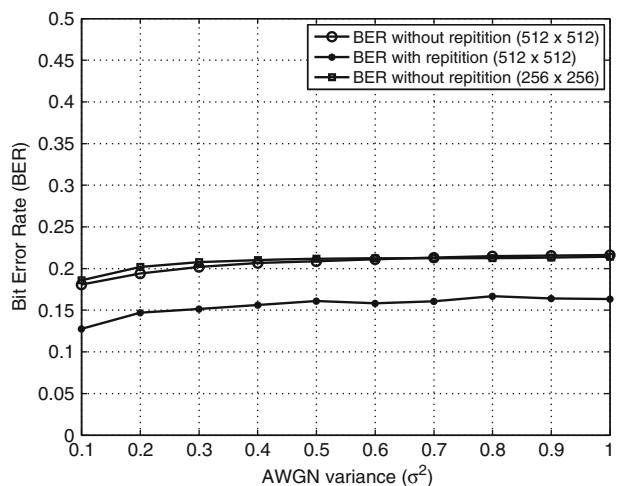
in Fig. 7, without repetition, the proposed algorithm gives a BER of 18.4 % and 22 % for  $\sigma^2 = 0.1$  and  $\sigma^2 = 1$  respectively. This represents a minimum of 78 % accurate payload retrieval even with highest attack strength. Repetition technique with a repetition factor of 24 for a pseudo random payload matrix, improved the BER by almost 5 % for each attacked case, resulting in 13 % BER for  $\sigma^2 = 0.1$  and 17 % BER for  $\sigma^2 = 1$ , collectively denoting a minimum of 83 % accurate payload retrieval for any AWGN attack strength.

Table 3 provides the comparison between survivability of Chaos based DCT steganography algorithm [34] and the proposed *Zero-steganography* scheme for salt and pepper noise attack. The proposed algorithm has a much higher survivability as compared to the other technique, for every value of noise density tested.

#### 4.1.3 JPEG compression attack

JPEG provides a compression method for compressing continuous-tone image data with a pixel depth of 6 to 24 bits. JPEG is primarily a lossy method of compression, having reasonable speed and efficiency. It may be adjusted to produce very small, compressed images that are of relatively poor quality in appearance but still suitable for many applications. At the same time, JPEG is also capable of producing very high-quality compressed images that are still far smaller than the original uncompressed data [25].

As JPEG is lossy, the information in a cover image is most likely to be damaged if this type of compression is applied. Therefore, we tested our algorithm for survivability against JPEG compression attack for various quality settings— $Q$ -ranging from

**Fig. 7** AWGN attack

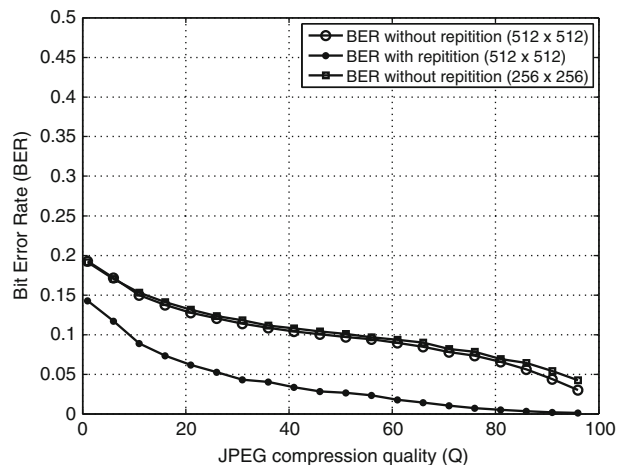
**Table 3** Comparison between survivability of Chaos based DCT steganography [34] and Chaos based *Zero-steganography* (CBZS) scheme for salt and pepper noise attack

Noise density	Chaos based DCT algo	CBZS
0.001	0.028	0.007
0.003	0.040	0.023
0.005	0.047	0.036
0.007	0.059	0.048
0.010	0.071	0.062
0.010	0.128	0.098

1 to 96 with a step size of 5. Here,  $Q = 1$  represents lowest image quality retained after compression and  $Q = 96$  represents highest quality retained after compression in the selected range. Figure 8 shows the results of BER for JPEG compression attack. For the proposed algorithm, worst quality JPEG compression gives a BER of about 20 % which keeps decreasing as we improve the compression quality and finally drops down to 3 % for  $Q = 96$ . Using repetition, a BER of 15 % was calculated for worst quality compression, while for the highest quality compression, BER was 0.1 % denoting a 99.9 % accurate payload retrieval rate.

Table 4 provides the comparison between survivability of a Chaos based DCT steganography algorithm [34], PN sequence based DCT algorithm [33] and the proposed *Zero-steganography* scheme for JPEG compression attack. The results show that for  $Q = 50$  the BER for the proposed algorithm is quite lower than that of other schemes. However, for higher values of  $Q$  the BER of the Chaos based DCT algorithm remains better than the proposed algorithm. Whereas, BER of PN sequence based DCT algorithm is better than the proposed *Zero-steganography* scheme in only intermediate cases. For both higher and lower attack strength the survivability of proposed algorithm is much better than that of PN sequence based DCT algorithm. While BER for the proposed algorithm remains quite high for high quality JPEG compression, the proposed algorithm performs much better under high attack strength as compared to other algorithms.

**Fig. 8** JPEG compression attack



**Table 4** Comparison between survivability of Chaos based DCT steganography [34], PN sequence based Data hiding scheme [33] and Chaos based *Zero-steganography* (CBZS) scheme for JPEG compression attack

Quality	Chaos based DCT algo	PN sequence based algo	CBZS
90	0.022	0.065	0.048
70	0.038	0.068	0.080
50	0.151	0.123	0.098

4.1.4 Summary

Table 5 summarizes the survivability analysis of the proposed algorithm for all three types of attacks discussed in Sections 4.1.1, 4.1.2 and 4.1.3. Attacked images of Lena in case of highest attack strengths, with corresponding payload retrieval rate and Peak Signal-to-Noise Ratio (PSNR) are displayed in Table 5. PSNR is given by the equation:

$$PSNR = 20 \times \log \left( \frac{1}{MSE} \right) db \tag{10}$$

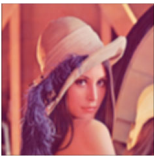
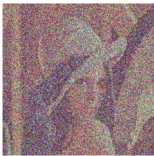
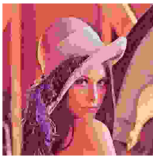
where, Mean Square Error (MSE) is given as:

$$MSE = \sqrt{\left( \frac{1}{N^2} \right) \sum_{i=1}^N \sum_{j=1}^N (x_{ij} - \hat{x}_{ij})^2} \tag{11}$$

where  $x_{ij}$  represent the pixels of the original image,  $\hat{x}_{ij}$  represent the pixels of distorted image and  $N$  is the total number of pixels.

Survivability of the proposed algorithm is highest against low-pass filtering as the least retrieval rate is above 86 %, obtained for filter size of  $9 \times 9$ . Statistically, the most distortion occurs in case of AWGN, where the retrieval rate obtained is 78.38 % for  $\sigma^2 = 1$ . However, in case of JPEG compression, the least retrieval rate is above 80 %, obtained in case of compression quality  $Q = 1$ . Despite the extremely distorted attacked cover images, for the proposed algorithm, the minimum retrieval rate stays above 78 %. We will now analyze the imperceptibility of the proposed *Zero-steganography* algorithm.

**Table 5** Attacked images, PSNR and retrieval rate

Attack	Low-pass filtering	AWGN	JPEG compression
Attack strength	Filter size = $9 \times 9$	$\sigma^2 = 1$	$Q = 1$
Attacked cover image			
Payload retrieval percentage	86.66	78.38	80.71
PSNR (dB)	52.8321	27.5512	29.6170

## 4.2 Imperceptibility

For our proposed algorithm, the stego-image quality is independent of the payload length and gives the maximum PSNR with no artifacts or change in cover image. This implies that, for a  $512 \times 512$  color image using averaging window of size  $8 \times 8$ , we can use a maximum payload length  $L = 98,304$  bits—as calculated by (8)—and still get the highest quality stego-image with a PSNR value of  $\infty$ . This is because the proposed algorithm makes no change to the cover image; and stego-image is the same as cover image in our case. Section 4.3 provides a detailed analysis of the security perspective of the proposed algorithm in several compromised scenarios.

## 4.3 Security

One of the key features of the proposed algorithm is that it provides high security by using chaos and stego keys. Security is compromised when a steganographic system is broken. This process is carried out in two stages [41]:

1. An attacker detects the use of steganography.
2. He is able to read the hidden message.

It is not possible to detect the use of steganography in the proposed *Zero-steganographic* system as no change is made to the cover image. An adversary can not detect the use of steganography due to absence of any distortion, artifacts and patterns in the stego image—it is the same as cover image. Hence, the proposed technique is secure and not vulnerable to visual and statistical attacks for detection.

Some of the types of attacks on steganographic systems include [14]: *stego-only*, *known cover*, and *chosen stego*. For the *stego-only* attack, the attacker is assumed to have stego medium only. In our case, it is the stego image—which is the same as cover image. From this information alone, an attacker can not determine if steganography is used and hence can not break the steganographic system. Similarly, for *known cover* attack, an attacker can not detect if steganography is used as the cover image and stego image are exactly the same. However, in case of *chosen stego* attack, the cover image is available along with the algorithm. Even then, it is computationally intensive for the attacker to extract the payload as he would require exact parameters for chaos and also the two keys.

Chaos has statistically infinite combinations of parameters—system parameter and initial condition—and along with two keys  $K_b$  and  $K_d$ , it is computationally intensive to figure out the right combination of the three elements to successfully retrieve the payload.

As stated in Section 3.2, an attacker would require i) cover image, ii) binary key  $K_b$ , iii) data key  $K_d$ , and iv) exact combination of parameters ( $\mu$  and  $X_0$ ) for generating chaotic sequence, for successful retrieval of payload. We have analyzed two special cases where three of the four necessary elements for extraction are compromised and only one is unknown to the attacker:

- *Case I—Unknown data key  $K_d$* : In this case, the attacker has cover image, binary key  $K_b$  and exact parameters for generating chaos. Now he may use a pseudo randomly generated data key  $K_d$  to extract the payload. To demonstrate the effect of using a randomly generated data key for extraction, we have considered two cases with Coin and some text as payload in the other. Tables 6 and 7 show



break the algorithm. Based on these results, we conclude that the proposed algorithm is secure in these highly-compromised scenarios. Following Section concludes the proposed work based on analysis and the presented results.

## 5 Conclusion

*Zero-steganography* implies no direct embedding which makes it totally undetectable. Conventional steganography methods are evaluated against steganalysis as the key concern is undetectability and imperceptibility. However, in case of *Zero-steganography*, no steganalysis is required as the cover image retains its original form after stego-key generation procedure. Along with imperceptibility and undetectability, the proposed algorithm also provides security and sufficient capacity while at the same time providing survivability of more than 75 % against highest attack strengths for selected attacks.

Imperceptibility analysis shows that the PSNR of stego image remains  $\infty$  regardless of the payload length. Survivability analysis of the proposed algorithm provides us with a maximum BER of about 14 % for low-pass filtering attack, occurring in case of  $9 \times 9$  filter size. For AWGN attack, maximum BER was 22 %, occurring in case of  $\sigma^2 = 1$ . For JPEG compression attack, maximum BER was 20 %, observed in case of compression quality  $Q = 1$ . Our test results show that by using error correcting techniques such as repetition, the BER is reduced by almost 6 % for all cases.

The major limitation of the proposed algorithm is the higher bandwidth required for the data key to be transmitted through secure means. However, even if the data key is discovered, the presence of a chaos based sequence limits any advantage an adversary can take out of the knowledge of data key. The proposed algorithm is not currently robust against geometric attacks, however, geometric transformation resistant characteristics extraction methods can be investigated to make the algorithm robust against such attacks. In addition, better error correcting techniques such as Reed Solomon codes [29], Bose, Chaudhuri, and Hocquenghem (BCH) codes [4] and Hamming codes [11] can be used to improve the survivability results [1].

The proposed algorithm satisfies the merit of imperceptibility due to absence of any artifacts and patterns as no change is made to the cover image. Moreover, using chaos provides extended security as not only the two secret keys, but also the accurate system parameter and initial condition for generating chaotic sequence are required for successful payload retrieval. As discussed in Section 2, even the slightest of change in the system parameter for a chaotic sequence leads to a chaotic map that is significantly different from the original one. Results in Tables 6–7 show that even in highly compromised scenarios, the payload can not be retrieved successfully. Hence, we conclude that the proposed *Zero-steganography* is totally imperceptible, undetectable, sufficiently survivable and highly secure.

## References

1. Abdul W, Carré P, Gaborit P (2009) List decoding of Reed Solomon codes for wavelet based colour image watermarking scheme. In: Proceedings of 16th IEEE international conference on image processing (ICIP), pp 3637–3640



2. Anderson R, Petitcolas F (1998) On the limits of steganography. *IEEE J Sel Areas Commun* 16(4):474–481
3. Barni M, Bartolini F, Cappellini V, Piva A (1998) A DCT-domain system for robust image watermarking. *Signal process* 66(3):357–372
4. Bose R, Ray-Chaudhuri D (1960) On a class of error correcting binary group codes. *Inf control* 3(1):68–79
5. Cao H, Xiang H, Li X, Liu M, Yi S, Wei F (2006) A zero-watermarking algorithm based on DWT and chaotic modulation. In: *Proceedings of SPIE*, vol 6247, pp 1–9
6. Chandramouli R, Memon N (2001) Analysis of LSB based image steganography techniques. In: *Proceedings of IEEE international conference on image processing (ICIP)*, vol 3, pp 1019–1022
7. Cheddad A, Condell J, Curran K, Mc Kevitt P (2010) Digital image steganography: survey and analysis of current methods. *Signal Process* 90(3):727–752
8. Fridrich J, Goljan M, Soukal D (2004) Perturbed quantization steganography with wet paper codes. In: *Proceedings of ACM multimedia security workshop*, vol 20, pp 4–15
9. Fridrich J (2005) Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In: *Information hiding, 7th international workshop, lecture notes in computer science*, pp. 67–81 (2005)
10. Fridrich J, Kodovský J, Goljan M, Holub V (2011) Steganalysis of content-adaptive steganography in spatial domain. In: *Information hiding, 13th international conference, lecture notes in computer science*, pp 102–117
11. Hamming R (1950) Error detecting and error correcting codes. *Bell Syst Tech J* 29(2):147–160
12. Hénon M (1976) A two-dimensional mapping with a strange attractor. *Commun Math Phys* 50(1):69–77
13. Huang HC, Wang FH, Pan JS (2001) Efficient and robust watermarking algorithm with vector quantisation. *Electron Lett* 37(13):826–828
14. Johnson N, Jajodia S (1998) Steganalysis of images created using current steganography software. In: *Information hiding, 2nd international workshop, lecture notes in computer science*, pp 273–289
15. Johnson NF, Duric Z, Jajodia S, Memon N (2001) Information hiding: steganography and watermarking—attacks and countermeasures. *J Electron Imaging* 10(3):825–826
16. Kahn D (1996) The history of steganography. In: *Information hiding, 1st international workshop, lecture notes in computer science*, pp 1–5
17. Kessler G (2012) Steganography: Hiding data within data. An edited version of this paper with the title “hiding data in data” appeared in April 2002 issue of windows & NET magazine. <http://www.garykessler.net/library/steganography.html>. Accessed 16 July
18. Kim JR, Moon YS (1999) A robust wavelet-based digital watermarking using level-adaptive thresholding. In: *Proceedings of 5th IEEE international conference on image processing (ICIP)*, vol 2, pp 226–230
19. Lenti J (2000) Steganographic methods. *Period Polytech, Electr Eng* 44(3/4), 249–258
20. Li B, He J, Huang J, Shi YQ (2011) A survey on image steganography and steganalysis. *JIH-MSP* 2(2):142–172
21. Li X, Zhang T, Zhang Y, Li W, Li K (2012) A novel blind detector for additive noise steganography in JPEG decompressed images. *Multimed Tools Appl* 1–18. doi:10.1007/s11042-012-1112-2
22. Lorenz E (1963) Deterministic nonperiodic flow. *J Atmos Sci* 20(2):130–141
23. Luo W, Huang F, Huang J (2010) Edge adaptive image steganography based on LSB matching revisited. *IEEE TIFS* 5(2):201–214
24. May R (1976) Simple mathematical models with very complicated dynamics. *Nature* 261(5560):459–467
25. Murray J, VanRyper W (1996) *Encyclopedia of graphics file formats*, 2nd edn. O’Reilly Media
26. Nutzinger M (2012) Real-time attacks on audio steganography. *JIH-MSP* 3(1):47–65
27. Peitgen H, Jürgens H, Saupe D (2004) *Chaos and Fractals: new frontiers of science*. Springer-Verlag
28. Provos N (2001) Defending against statistical steganalysis. In: *Proceedings of the 10th conference on USENIX security symposium*, vol 10, pp 323–336
29. Reed IS, Solomon G (1960) Polynomial codes over certain finite fields. *SIAM J Appl Math* 8:300–304
30. Sabeti V, Samavi S, Shirani S (2012) An adaptive LSB matching steganography based on ocotary complexity measure. *Multimed Tools Appl* 1–17. doi:10.1007/s11042-011-0975-y

31. Sallee P (2004) Model-based steganography. In: Proceedings of the 2nd international workshop on digital watermarking, lecture notes in computer science, vol 2939, pp 254–260
32. Schuster H, Just W (1988) Deterministic chaos. Wiley Online Library
33. Singh S, Siddiqui TJ, Singh HV (2011) DCT based digital data hiding in image cover. *IJSS* 5(1):45–49
34. Singh S, Siddiqui TJ (2012) A security enhanced robust steganography algorithm for data hiding. *IJCSI* 9(1):131–139
35. Westfeld A, Pfitzmann A (2000) Attacks on steganographic systems. In: Information hiding, 3rd international workshop, lecture notes in computer Science, pp 61–76
36. Westfeld A (2001) F5—A steganographic algorithm. In: Information hiding, 4th international workshop, lecture notes in computer science, pp 289–302 (2001)
37. Williams G (1997) Chaos theory tamed. CRC Press
38. Wu D, Tsai W (2003) A steganographic method for images by pixel-value differencing. *Pattern Recogn Lett* 24(9–10):1613–1626
39. Xiang H, Wang L, Lin H, Shi J (1999) Digital watermarking systems with chaotic sequences. In: Proceedings of the SPIE electronic imaging '99 conference, vol 3657, pp 449–457
40. Yen JC (2001) Watermarks embedded in the permuted image. In: IEEE international symposium on circuits and systems (ISCAS), vol 2, pp 53–56
41. Zöllner J, Federrath H, Klimant H, Pfitzmann A, Piotraschke R, Westfeld A, Wicke G, Wolf G (1998) Modeling the security of steganographic systems. In: Information hiding, 2nd international workshop, lecture notes in computer science, pp 344–354



**Muhammad Bilal** received his BS in Computer Engineering from COMSATS Institute of Information Technology, Islamabad, Pakistan, in 2012. He is currently pursuing admission in MS in Computer Engineering. His research interests include Steganography, Computer vision, Image processing, Microprocessor architectures, Multiprocessor and multi-core architectures, and Distributed computing systems.



**Sana Imtiaz** received her BS in Computer Engineering from COMSATS Institute of Information Technology, Islamabad, Pakistan, in 2012. Currently, she is pursuing admission in Masters in Computer Engineering. Her research interests include Steganography, Watermarking, Image Processing, Digital System design and Computer architecture.



**Wadood Abdul** received his BE Degree from COMSATS Institute of Information Technology, Islamabad, Pakistan, in 2004. He did Masters from University of Limoges, Limoges, France in 2007 and PhD in Signal and Image Processing from University of Poitiers, Poitiers, France in 2011. Currently he is working as an Assistant Professor at the Department of Computer Engineering, CCIS, King Saud University, Riyadh, Saudi Arabia. His research interests are focused on color image watermarking, steganography, fingerprinting and biometric template protection.



**Sanaa Ghouzali** received both the Master's and the Ph.D. degrees in computer science and telecommunications from University Mohamed V-Agdal, Rabat, Morocco, in 2004 and 2009, respectively. In 2005 she received a Fulbright grant to undertake dissertation research on a joint supervision program at the Visual and Communication Laboratory of Cornell University, Ithaca, NY, USA. Between 2009 and 2011, she was an Assistant Professor at ENSA (the National school of Applied Sciences) within Abdelmalek Essaadi University. Starting 2012, she joined King Saud University as an Assistant Professor in the College of Computer and Information Sciences. Her research interests include statistical pattern detection and recognition, Biometrics, Biometric Security and Protection.



**Shahzad Asif** received the Bachelors degree in computer engineering from COMSATS Institute of Information Technology (CIIT), Islamabad, Pakistan, in 2005, and the Masters degree in system-on-chip from Linköping University, Linköping, Sweden, in 2008. After his B.Sc. he worked as a Research Associate with CIIT, Islamabad, Pakistan, for two years. He then received a scholarship from HEC (Higher Education Commission), Pakistan, for his Masters Studies in Sweden. Since September 2008, he has been a Lecturer with CIIT, Islamabad, Pakistan. His research interests include low power circuit design, processor architectures, and FPGA.