Hindawi
Security and Communication Networks
Volume 2017, Article ID 5397082, 14 pages
https://doi.org/10.1155/2017/5397082

Research Article

A Secure Steganographic Algorithm Based on Frequency Domain for the Transmission of Hidden Information

A. Soria-Lorente[1,2] and S. Berres[2]

[1]*Department of Basic Sciences, Granma University, Bayamo, Cuba*
[2]*Departamento de Ciencias Matemáticas y Físicas, Casilla 15 D, Universidad Católica de Temuco, Temuco, Chile*

Correspondence should be addressed to A. Soria-Lorente; asorial@udg.co.cu

Received 13 September 2016; Revised 1 December 2016; Accepted 20 December 2016; Published 1 March 2017

Academic Editor: Barbara Masucci

This contribution proposes a novel steganographic method based on the compression standard according to the Joint Photographic Expert Group and an Entropy Thresholding technique. The steganographic algorithm uses one public key and one private key to generate a binary sequence of pseudorandom numbers that indicate where the elements of the binary sequence of a secret message will be inserted. The insertion takes eventually place at the first seven AC coefficients in the transformed DCT domain. Before the insertion of the message the image undergoes several transformations. After the insertion the inverse transformations are applied in reverse order to the original transformations. The insertion itself takes only place if an entropy threshold of the corresponding block is satisfied and if the pseudorandom number indicates to do so. The experimental work on the validation of the algorithm consists of the calculation of the peak signal-to-noise ratio (PSNR), the difference and correlation distortion metrics, the histogram analysis, and the relative entropy, comparing the same characteristics for the cover and stego image. The proposed algorithm improves the level of imperceptibility analyzed through the PSNR values. A steganalysis experiment shows that the proposed algorithm is highly resistant against the Chi-square attack.

1. Introduction

1.1. Motivation. Nowadays, there is a risk that confidential information may be acceded by nonauthorized people, being consulted, divulged, modified, destroyed, or sabotaged, affecting its availability and legal access. Evidently, the information that transits through insecure channels can be easily intercepted. Therefore, the use of Cryptography and Steganography plays a significant role in information security management [1].

Cryptography is the science of protecting information by encryption techniques. As cryptography on its own does not hide the fact that a message is secret, to provide this steganography is used.

The ability to protect sensible information from adversaries, especially during their transmission through channels that are opposed to have leaks, is crucial in a world of emerging cyberwar. Nowadays, all electronic communications are being continuously and automatically monitored by both private and state-owned intelligent systems that have an enormous computer power. In particular, every transmission of cipher-text calls the attention of any of these systems and certainly is chosen to be analyzed, among others, by competitors and any sort of opposing forces. The use of electronic transmission media requires a method that calls less attention of the supervisory automatic systems. Modern Steganography offers a level of service that includes privacy, authenticity, integrity, and confidentiality of the transmitted data.

1.2. State of the Art. Steganography is the art of hiding information by impeding the detection of hidden messages [2]. Steganography involves communicating secret data in an appropriate multimedia carrier, like image [3], audio [4], or video files [5]. The media with or without hidden information are called Stego Media and Cover Media, respectively. Thus, the cover image with the secret message embedded is called the stego image.

1.2.1. Embedding Capacity. The performance of steganographic algorithms can be measured by two main criteria, embedding capacity and detectability. Thus, novel steganographic algorithms are expected to increase the image capacity and the encryption strength of the message. The image capacity can be increased by adaptive strategies which decide where to insert best the message. For example, the Pixel-Value Differencing method [6] proposes to embed more data in edged areas than in smooth areas.

In steganography the *embedding capacity* is defined as the maximum number of bits that can be embedded in a given cover image. However, the *steganographic capacity* is the maximum number of bits that can be embedded in a given cover image with a negligible probability of detection by an adversary. Therefore, the embedding capacity is larger than the steganographic capacity [7].

1.2.2. Methods of Embedding Data. There are two common methods of embedding data, which can be classified into two categories, namely, spatial-domain and frequency-domain methods. In the spatial domain method, the secret message is inserted directly into the least significant bit (LSB) of image pixels [8, 9]. In the frequency-domain method [10] the cover image is first transformed from the spatial to a frequency-domain using transformation methods such as discrete cosine transform (DCT) [11, 12], discrete Fourier transform (DFT) [13], or discrete wavelet transform (DWT) [13–15]. Then the secret message is embedded in the transformed coefficients [16], and finally the data are transformed back from the frequency-domain to the spatial domain.

There are different strategies to additionally encrypt hidden steganographic data, like permutation [17] or by a statistical threshold [18]. One should not confuse cryptography with steganography: Cryptography modifies the data to make them incomprehensible, while steganography on its own simply hides them between other data. In many instances the combination of both techniques provide a high level security to the protected information; see, for instance, [3, 19–23].

In this contribution we adopt the two-dimensional discrete cosine transform [11] as the most common frequency domain method used in image processing. For a recent survey on steganography see [24], where our approach can be classified within DCT based steganography. A frequent method that uses the DCT is the Jpeg-Jsteg embedding method [25], where the secret message is embedded in those LSB of the quantized DCT coefficients whose values are not 0, 1, or −1.

1.2.3. Design Principles That Counter Steganalysis. Westfeld [26] introduced the F5 steganographic algorithm, where, instead of replacing the LSB of the quantized DCT coefficients by the secret bits, the absolute value of the coefficients is reduced by 1. Since the F5 algorithm randomly chooses DCT coefficients to embed the secret bits it is strong against the Chi-square attack.

To counter the Chi-square attack, Provos [25, 27] proposed the OutGuess steganographic algorithm. This method embeds data by a similar way as Jpeg-Jsteg, though its embedding capacity is much lower than that of Jpeg-Jsteg.

There are general design principles for less detectable stegosystems [28]. However, some steganalyis concepts are specialized in the detection of data hidden in the least significant bits of a natural image [29]. For recent improvements in LSB steganography approaches see [28, 30].

In [31], Chang et al. presented a new steganography method based upon Joint Photographic Expert-Group (JPEG) and a quantization table modification. In this case, the secret message is first encrypted and then embedded in the 26 coefficients located in the middle-frequency area of the cover image. In [32], Noda et al. proposed two JPEG steganography schemes using a quantization index modulation in DCT domain.

In [33], Wong et al. presented a novel DCT based on a blind Mod4 steganography method for still images. In [34], Chang et al. proposed a lossless and reversible embedding of secret data in each block of DCT coefficients based on the compressed image technique JPEG. In this scheme, two successive zero coefficients of the medium-frequency components in each block are used to hide the secret data. Li and Wang [35] proposed a novel steganographic method, based on JPEG and the Particle Swarm Optimization algorithm. Here, the transformed messages are embedded in the 36 coefficients located in the middle-frequency components of the quantized DCT coefficients of the cover-image.

1.2.4. Embedding Strategies. In [36], Velasco-Bautista et al. present the Entropy Thresholding method, where the secret message is inserted in the DCT domain. In [37], Lin and Shiu suggest a high capacity data hiding scheme, where the secret data are embedded in the middle frequency of the DCT coefficients. Narayana and Prasad [21] introduce two new methods wherein cryptography and steganography are combined to encrypt the secret message that is hidden. In [38], Lin proposes a reversible data hiding method which is based on the DCT of the cover image. The cover image is decomposed into different frequencies, where the secret messages are embedded into the high-frequency parts. Mali et al. propose a novel image steganographic method using a block level Entropy Thresholding technique where the secret message is embedded in the 26 coefficients located in the middle-frequency area of the cover image [16]. Amin et al. present an efficient data hiding technique based on the DCT, where secret bits are embedded in all frequency components of the quantized DCT coefficient [11]. Jaheel and Beiji combine two steganography algorithms, namely, Jpeg-Jsteg and OutGuess algorithms, with the purpose of enhancing a major security level [39]. In [1], Soria et al. propose a new steganographic algorithm in the frequency domain. This method uses a private key of 64 bits that suggest the positions where the secret bits are inserted after applying the Entropy Thresholding method; the experimental analysis accomplishes a comparison of the results with respect to the Entropy Thresholding method proposed by Velasco-Bautista et al. [36].

In the present work, we show an experimental comparison of the proposed method with respect to the two previous methods. All three approaches use the Entropy Thresholding

method. The Velasco-Bautista method does not use keys, the Soria method uses a private key of 64 bits, and the proposed method uses two keys, one public key and one private key of 64 bits.

1.3. Transformation of the Blocks. A digital image is represented in computer systems as an array of pixels. In this paper, we work with a steganography algorithm for embedding a secret message into a RGB 24-bit color image, where each pixel has three color components: Red, Green, and Blue (RGB). Each RGB component is represented by a byte. The values range from 0 to 255, where zero represents the darkest and 255 the brightest shade of a color.

Since we consider a RGB 24-bit color image as three blocks of bytes corresponding to each color component, in the proposed method the cover image C of size $m \times n$ is divided into $3mn/64$ different blocks of 8×8 bytes such that the DCT transformation can be performed on each one of them. Here we assume that m and n are divisible by 8.

Let $(B_{i,j}^k)$ be the kth block of 8×8 bytes of the image, with $i, j = 0, \ldots, 7$, $k \in \mathcal{K} = \{1, 2, \ldots, 3mn/64\}$ and let $(\mathcal{B}_{u,v}^k)$ be its two dimensional discrete cosine transform, with $u, v = 0, \ldots, 7$. The relationship between $\mathcal{B}_{u,v}^k \equiv \mathrm{DCT}(u, v)$ and its inverse $B_{i,j}^k \equiv \mathrm{IDCT}(i, j)$ is given by

$$\begin{aligned} \mathcal{B}_{u,v}^k &= 4^{-1}\sigma(u)\,\sigma(v) \sum_{0 \le i,j \le 7} B_{i,j}^k \\ &\quad \times \cos\left(\frac{\pi u(2i+1)}{16}\right) \cos\left(\frac{\pi v(2j+1)}{16}\right), \\ B_{i,j}^k &= 4^{-1} \sum_{0 \le u,v \le 7} \sigma(u)\,\sigma(v)\,\mathcal{B}_{u,v}^k \\ &\quad \times \cos\left(\frac{\pi u(2i+1)}{16}\right) \cos\left(\frac{\pi v(2j+1)}{16}\right), \end{aligned} \tag{1}$$

where $\sigma(x) = \sqrt{2}/2$ for $x = 0$ and $\sigma(x) = 1$ otherwise. A DCT of a 8×8 integer matrix becomes a 8×8 matrix of real numbers. By definition, the coefficient $\mathcal{B}_{0,0}^k$ is the DC coefficient (zero frequency) and all others are called the AC coefficients.

1.4. This Contribution. In this paper, two steganography methods are combined, namely, the JPEG steganography method and the Entropy Thresholding technique [36, 39]. This combination allows to hide secret messages in images while maintaining a high visual quality and a low detectability. The JPEG method divides the cover image into nonoverlapping blocks of 8×8 bytes and then applies the discrete cosine transform with the purpose of hiding the secret message in the DCT domain by modifying certain coefficients located in the low-frequency area of the cover image.

The Entropy Thresholding method decides whether or not to embed the secret message in a certain matrix of order 8 of transformed coefficients, depending on the entropy within that matrix.

2. Determination of Location of Message

In this section we describe how to determine the encrypted random locations where the secret message is inserted. It is assumed that N is the length of the binary sequence of the secret message

$$S = \{s_i \in \{0, 1\} : 1 \le i \le N\}, \tag{2}$$

where s_i is a bit containing 0 or 1. The locations where the secret message is hidden is determined by a pseudorandom sequence defined by

$$\mathcal{A} = \{a_1, a_2, \ldots, a_\ell, \ldots, a_M\}. \tag{3}$$

We denote by $|X|$ the number of bits of an array of bits X which are equal to 1. We continue to concatenate the pseudorandom sequence as long as we reach a cardinality $|\mathcal{A}| = N$; that is, the length of the message is covered.

In the next subsections, first, we describe the generation of 15 subkeys, and then how the subkeys are used to determine a pseudorandom sequence of locations where the secret bits are inserted.

2.1. Creation of 15 Subkeys. There are two types of cryptography techniques, namely, private key and public key cryptography. Public key cryptography is an asymmetric cryptography technique which encrypts the message with a private key and decrypts the message with a public key. Private key cryptography is a symmetric cryptography technique which encrypts and decrypts a message with the same key. The DES (Data Encryption Standard) algorithm [40] is the most widely used symmetric encryption algorithm in the world and its design idea is still used in numerous block ciphers.

In this paper we use steps similar to those developed for the Data Encryption Standard (DES) algorithm to generate subkeys, but we only generate 15 new subkeys C_i of 96 bits, with $i = 1, \ldots, 15$, using compressions (permutations) of 112 and 96 bits, respectively.

To generate the 15 subkeys we use one key of 128 bits (as defined in the first step of Section 2.2), by the following steps:

(1) The 128-bit key is permuted according to the following permutation:

42	37	53	51	33	123	128	114
27	107	38	29	74	17	92	64
104	58	45	60	103	23	63	7
85	8	39	117	65	13	2	101
35	124	31	97	44	41	110	25
120	3	105	61	50	70	102	95
113	77	115	59	75	21	48	125
20	47	99	100	90	32	19	9
4	94	68	122	109	89	28	83
111	82	12	54	84	73	14	78
62	88	57	55	79	91	10	121
93	108	98	30	127	87	34	24
49	1	11	40	71	43	80	69
15	119	112	22	52	5	81	67

The first entry in the table being 42 means that the 42th bit of the original key becomes the first bit of

the permuted key. The 37th bit of the original key becomes the second bit of the permuted key. The 67th bit of the original key is the last bit of the permuted key. Note that only 112 bits of the original key appear in the permuted key.

(2) Next, split this key into a left and a right half, c_0^{L} and c_0^{R}, where each half contains 56 bits.

(3) Then, with c_0^{L} and c_0^{R} defined, we now create fifteen blocks c_i^{L} and c_i^{R}, $1 \leq i \leq 15$. Each pair of blocks c_i^{L} and c_i^{R} is formed from the previous pair c_{i-1}^{L} and c_{i-1}^{R}, respectively, for $i = 1, \ldots, 15$, using the following schedule of "left shifts" of the previous block:

$$\begin{aligned} 1 \quad & \text{for } i \in \{1, 2, 9\}, \\ 2 \quad & \text{for } i \in \{1, \ldots, 15\} \setminus \{1, 2, 9\}. \end{aligned} \tag{4}$$

To do a left shift, move each bit one place to the left, except for the first bit, which is cycled to the end of the block. A double shift is twice a single shift. This means, for example, c_9^{L} and c_9^{R} are obtained from c_8^{L} and c_8^{R}, respectively, by one left shift, and c_{15}^{L} and c_{15}^{R} are obtained from c_{14}^{L} and c_{14}^{R}, respectively, by two left shifts.

(4) In order to finish, we generate keys C_i of 96 bits, with $1 \leq i \leq 15$, by applying the following permutation to each of the concatenated pairs $[c_i^{\mathrm{L}} \mid c_i^{\mathrm{R}}]$:

44	14	77	49	45	111	68	37
23	47	81	31	32	67	9	12
102	73	69	29	88	3	27	62
75	15	89	34	11	10	97	58
105	104	84	70	5	103	100	78
87	21	83	63	1	71	17	30
65	13	50	4	112	53	41	20
8	109	54	61	38	94	40	80
74	42	55	101	110	79	64	92
22	85	91	107	59	24	35	98
2	33	52	93	99	19	28	51
90	48	43	82	7	95	25	57

Notice that the first bit of C_i is the 44th bit of $[c_i^{\mathrm{L}} \mid c_i^{\mathrm{R}}]$, the second bit is the 14th, and so on, ending with the 96th bit of C_i being the 57th bit of $[c_i^{\mathrm{L}} \mid c_i^{\mathrm{R}}]$.

In summary, this procedure uses one key of 128 bits, to determine 15 subkeys of 96 bits, which, through the procedure that is described in continuation, generates a pseudorandom sequence that determines the location to insert the secret bits.

2.2. Generation of Pseudorandom Sequence of Locations. In this subsection we describe how the location of the secret message is determined by using the subkeys as defined in Section 2.1. The generation of the pseudorandom sequence is realized according to the following steps:

(1) Generate a public key of 64 bits. Concatenate the private key and the public key. Alternate its bytes to create a new key of 128 bits. The first element of the composed key of 128 bits is the first element of the private key and the second is the first element of the public key and so on. Extract the least significant bits of each byte of the composed key, getting this way a binary sequence C^0 of 16 bits.

(2) From the newly concatenated key generate 15 new subkeys C_i of 96 bits, with $i = 1, \ldots, 15$, according to the steps that were described in Section 2.1.

(3) Apply the permutation.

41	60	75	61	7	68	77	71
1	87	23	31	87	27	57	87
65	62	5	48	92	24	22	36
93	5	80	10	13	87	7	6
72	85	96	76	63	7	19	9
26	4	66	4	55	42	53	79
86	2	7	64	33	4	39	35
58	54	32	70	77	12	27	34
89	5	15	52	5	46	44	27
56	82	81	73	57	78	43	20
77	59	83	8	45	67	77	77
27	88	77	77	29	40	11	51
94	3	49	14	4	21	14	37
90	47	14	84	38	30	50	91
57	5	87	17	69	5	74	7
77	16	57	28	18	95	25	77

to the binary sequence $C_j \oplus C_{j+1}$, with $j = 1, \ldots, 14$, expanding it to a binary sequence $\overline{C}_j$ of 128 bits. Here, the symbol $\oplus$ defines the binary operation $0 \oplus 0 = 1 \oplus 1 = 0$ and $0 \oplus 1 = 1 \oplus 0 = 1$. Then, apply the operation $\oplus$ between the 16 bits of sequence C^0 and each of the 8 blocks of 16 bits of the binary sequence $\overline{C}_j$ of 128 bits, creating a new binary sequence $\widehat{C}_j$ of 128 bits.

(4) Concatenate $\overline{C} = \widehat{C}_1 \cup \widehat{C}_2 \cup \cdots \cup \widehat{C}_{14}$. If the cardinality of concatenated keys is $|\overline{C}| \geq N$, then the subsequence $\mathscr{A} = \{a_1, a_2, \ldots, a_\ell, \ldots, a_N\}$ of $\overline{C}$ is extracted such that the cardinality $|\mathscr{A}| = N$ covers the whole secret message being a_1 the first bit of $\overline{C}$ and so on.

(5) If the length of the secret message exceeds the cardinality $|\overline{C}|$ then the set $\overline{C}$ is extended by adding another pseudorandom sequence obtained using the first 64 bits of $\widehat{C}_{13}$ as private key and the last 64 bits of $\widehat{C}_{14}$ as public key. This procedure continues until the desired length is obtained.

3. Proposed Algorithm

In this section we propose a new algorithm for steganography. Given a procedure for the determination of the possible locations of the sub-keys (as described in Section 2.2), the algorithm is mainly about the effective insertion of the message according to a threshold. The embedding algorithm has a sandwich structure, where first several transformations are applied subsequently. In the core of the algorithm the secret message is inserted respectively extracted whenever

▷ Subtract 128 from each byte of image to produce a data range that is centered around zero, so that the modified range is $[-128, 127]$.
(i) Divide the cover image into $3mn/64$ non-overlapping blocks of 8×8 bytes as indicated in Section 1.3.
for $k \in \mathcal{K}$ **do**
▷ Calculate the Discrete Cosine Transform (DCT) coefficients $\mathcal{B}^k_{u,v}$ for each 8×8 byte matrix $(B^k_{i,j})$.
(i) Compute the entropy

$$E_k = \sum_{0 \le u,v \le 7} \left| \mathcal{B}^k_{u,v} \right|^2.$$

end for
(i) Calculate mean entropy $\overline{E}$ of all transformed blocks.
(ii) $\ell = 1$.
(iii) $i = 1$.
for $k \in \mathcal{K}$ **do**
if $E_k > \overline{E}$ **then**
▷ $\Theta^k \leftarrow \mathcal{B}^k$: Quantify $(\mathcal{B}^k_{u,v})$ according to (9).
▷ $\nu^k \leftarrow \Theta^k$: Apply the zigzag scan, see Figure 3.
for $j = 2, \ldots, 8$ **do**
if $a_\ell = 1$ **then**
if mode = EMBEDDING **then**
if $\nu^k_j < 0$ **then**
$\overline{\nu}^k_j \leftarrow -R(|\text{round}(\nu^k_j)|, s_i)$: Apply the replacement rule according to (5) and (6).
else
$\overline{\nu}^k_j \leftarrow R(\text{round}(\nu^k_j), s_i)$
end if
else
$s_i = R^{-1}(|\text{round}(\nu^k_j)|)$: Apply extraction rule (8).
end if
$i = i + 1$: Goto next bit of secret message.
else
$\overline{\nu}^k_j \leftarrow \nu^k_j$
end if
$\ell = \ell + 1$: Goto next index of location indicator.
end for
◁ $\overline{\Theta}^k \leftarrow \overline{\nu}^k$: Reorganize the vector ν^k with the secret message to a matrix of order 8, taking into account the zigzag scan order.
◁ $\overline{\mathcal{B}}^k \leftarrow \Theta^k$: Multiply the previous matrix by the quantification matrix (10).
◁ $\overline{B}^k \leftarrow \overline{\mathcal{B}}^k$: Apply the Inverse Discrete Cosine Transform (IDCT).
else
◁ $\overline{B}^k \leftarrow B^k$: Copy the matrix.
end if
end for
◁ Add to each byte 128 to reconstruct the image, obtaining the expected stego image.

Algorithm 1: Embedding algorithm.

an entropy threshold criterion is satisfied. Finally, inverse transformations are applied in reverse order. See Algorithm 1 for a pseudocode, where the operating mode is switched to embedding or extraction by setting the parameter `mode` to `EMBEDDING` or `EXTRACTION`, respectively.

Roughly speaking, bits of the secret message are inserted at given locations whenever the entropy of a block is above a certain threshold value. Therefore, the proposed algorithm divides the cover image into nonoverlapping blocks of 8×8 bytes and then applies the discrete cosine transform to each matrix of them. The Entropy Thresholding method determines the matrices that will be quantized. The combined private-public key determines a binary sequence of pseudo-random numbers that indicate the first seven AC coefficients where the elements of the binary sequence of the secret image will be inserted.

3.1. Replacement Rule. We denote by $R(x, s_i)$ the function that replaces a value $x \in \mathbb{N}$ by the corresponding secret message bit s_i. For $s_i = 0$ the rule $R(x, s_i)$ is defined by

$$R(x, s_i) = \begin{cases} x - 1, & \text{if } x \text{ is odd,} \\ x, & \text{if } x \text{ is even.} \end{cases} \quad (5)$$

Analogously, for $s_i = 1$ we define $R(x, s_i)$ as

$$R(x, s_i) = \begin{cases} x, & \text{if } x \text{ is odd,} \\ x + 1, & \text{if } x \text{ is even.} \end{cases} \tag{6}$$

The effect of different values of x and s_i on $R(x, s_i)$ is shown in (7). The replacement rule is designed

$$\begin{array}{c|cc} & x \text{ even} & x \text{ odd} \\ \hline s_i = 0 & R(x, s_i) \text{ even} & R(x, s_i) \text{ even} \\ s_i = 1 & R(x, s_i) \text{ odd} & R(x, s_i) \text{ odd} \end{array} \tag{7}$$

such that $s_i = 0$ corresponds to even and $s_i = 1$ to odd values of $R(x, s_i)$ and thus provides the rule for the inverse operation R^{-1} for the following extraction function defined by

$$x = R^{-1}(y) = \begin{cases} 0, & \text{if } y \text{ is even,} \\ 1, & \text{if } y \text{ is odd.} \end{cases} \tag{8}$$

The process of extraction follows the same procedure as executed in the process of insertion, except that instead of inserting the least significant bit, the corresponding message bit is extracted. For both processes, embedding and extraction, the input consists of the stego image, private key, public key, and the quality factor.

3.2. Embedding Algorithm. The secret message is inserted into the cover image by the embedding procedure described in Algorithm 1.

Input. Secret message, cover image, private key, public key, and quality factor.

Procedure. In the proposed algorithm, it is assumed that the emitter as well as the receiver holds the same system of private keys. Indeed, the receiver sends the public key to the emitter by an insecure channel. Then, the emitter generates the stego image with both keys and sends it through another insecure channel to the receiver, which can extract the secret message from the public and private key; see Figure 2.

The emitter generates the stego image according to Algorithm 1. Firstly, the proposed algorithm subtracts 128 from each byte of the image. Next, it splits the cover image up into nonoverlapping blocks of 8×8 bytes and then applies the discrete cosine transform to each one of them. From each transformed block it calculates the entropy.

By the Entropy Thresholding method it decides whether or not to embed the secret message in the transformed block. If the entropy of a block is greater than the overall average entropy then each selected block is quantified using the quantification matrix (9); the zigzag scan is applied, with the purpose to align frequency coefficients in ascending order; see Figure 3. Afterwards, the elements of the binary sequence of the secret message are inserted in the first seven AC coefficients whenever the elements $a_\ell \in \mathscr{A}$ are equal to 1. Since the replacement rule operates on natural numbers, the real valued coefficients have to be rounded.

After insertion of the secret message the back transformation is realized in reverse order: By the zigzag scan the matrix of order 8 is reconstructed, which afterwards is unquantified multiplying by the quantification matrix (10). Finally, the Inverse Discrete Cosine Transform (IDCT) is applied, and to each byte of the resultant image the value 128 is added in order to reconstruct the image, obtaining the expected stego image.

3.3. Details on Quantification Procedure and Zig-Zag Scan. In the quantification procedure each block of 8×8 bytes selected by the Entropy Thresholding method is quantified using the given quantification matrix that scales the quantized values by a compression quality factor qF; see [36] for more details. The quantized DCT coefficients $\Theta^k_{u,v}$ are computed as

$$\Theta^k_{u,v} = \text{round}\left(\frac{\mathscr{B}^k_{u,v}}{Q^{\text{qF}}_{u,v}}\right), \quad 0 \le u, v \le 7, \tag{9}$$

where Q^{qF} is the matrix

$$Q^{\text{qF}} = \begin{cases} \dfrac{100 - \text{qF}}{50} Q_{50}, & \text{if qF} > 50, \\ \dfrac{50}{\text{qF}} Q_{50}, & \text{otherwise.} \end{cases} \tag{10}$$

Here, the quantization matrix Q_{50} is chosen to be the Lohscheller matrix:

$$Q_{50} = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}. \tag{11}$$

The transformation of the matrix $(\Theta^k_{u,v})$, with $0 \le u, v \le 7$, to a vector $\nu^k = \{\nu^k_i : 1 \le i \le 64\}$ of length 64 is done by the zigzag order scan; see Figure 3, which aligns frequency coefficients in ascending order starting from frequency zero (DC coefficient) to nonzero frequency components (AC coefficients). Indeed, the DC coefficient contains a significant fraction of the image energy. In addition, the AC coefficients can be classified in three groups, those that occur at low (yellow color), at middle (green color), and at high (blue color) frequency, respectively. Out of the 9 low frequency AC coefficients (yellow color) only the first 7 AC coefficients are considered, whenever the pseudorandom sequence and the entropy threshold permits.

Usually, zero AC coefficients occur at middle and high frequency, so modifications to them break the structure of continuous zeros. Abrupt nonzero values give a hint of the existence of secret bits. The nonzero AC coefficients occur at low and middle frequency such that perturbations to them do not affect the visual quality [41].

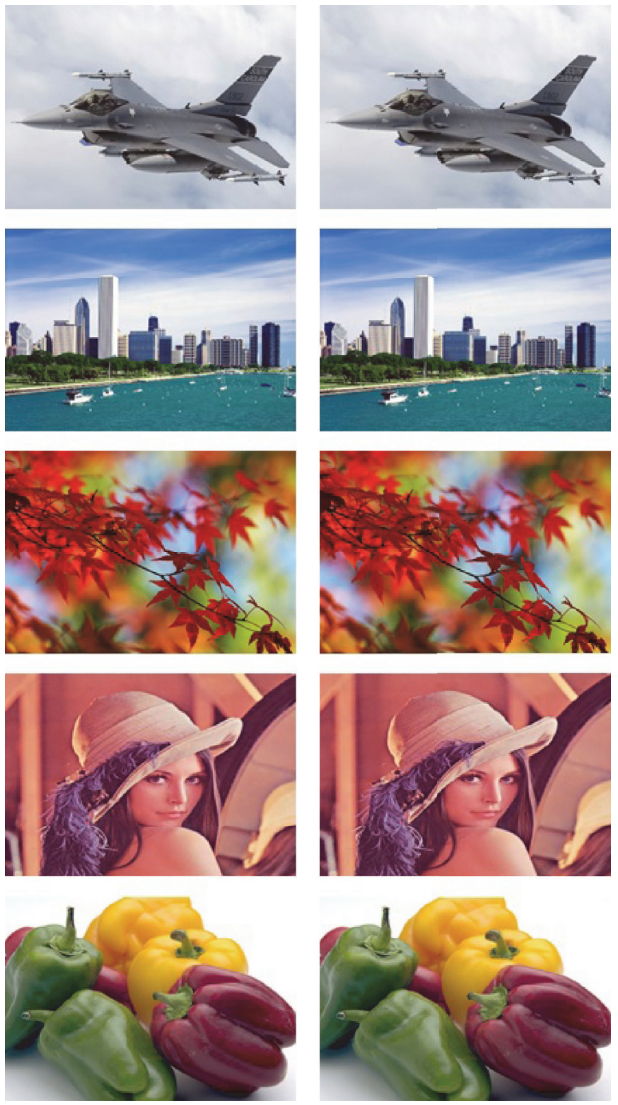

Figure 1: The images before and after embedding the secret message. The first row contains the cover images while the second contains the stego images.

4. Experimental Results

In this Section the experimental results of the proposed algorithm are presented. The proposed algorithm is implemented in Matlab. Five different images consisting of 784×512 pixels are used, with a quality factor equal to qF = 57. The cover images and the stego images are shown in Figure 1.

The used keys were taken randomly. We have tested our proposed algorithm by inserting the following message to each of the five images:

> *Nec vero habere virtutem satis est quasi artem aliquam nisi utare; etsi ars quidem cum ea non utare scientia tamen ipsa teneri potest, virtus in usu sui tota posita est.* (Cicero, "De re publica")

In addition, a comparison of the proposed method with respect to the Entropy Thresholding method proposed by Velasco-Bautista et al. [36] and to the method proposed by Soria et al. [1] is included in this section. The performance of the proposed approach has been studied using different kinds of statistical measures.

4.1. Imperceptibility Test. The information security through steganography depends in great part on the level of imperceptibility, since a steganographic system has to generate a sufficiently innocent stego image. Therefore, the degree of distortion or imperceptibility of a stego image in relation to the original image plays a fundamental role. Usually, the image distortion is measured by the peak signal-to-noise ratio (PSNR), which is given by

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{256^2}{\mathrm{MSE}} \right), \tag{12}$$

where

$$\mathrm{MSE} = (3mn)^{-1} \sum_{\gamma \in \Gamma} \left(C(\gamma) - S(\gamma) \right)^2, \tag{13}$$

and C and S are the cover and stego image, respectively. The index set $\gamma = (\gamma_1, \gamma_2, \gamma_3)$ sums over the set of bytes as

$$\Gamma = \{1, \ldots, m\} \times \{1, \ldots, n\} \times \{1, 2, 3\}, \tag{14}$$

where m, n account for the image size, β_3 addresses the three colours, and $C, S \in \{0, 1, \ldots, 255\}$.

In the first experiment, the PSNR values indicate the level of imperceptibility and distortion of those images. In this experiment, 128 randomly chosen pairs of different keys were used. The experimental results show that the proposed algorithm produces high quality stego images with appropriate PSNR values; see Figures 4–8, which is in correspondence with the heuristic values of PSNR (30 to 50 db) found in literature [42]. Moreover, this experiment shows that the proposed method gives usually better results than the methods proposed by Soria et al. [1] and Velasco-Bautista et al. [36].

4.2. Image Quality Measures. The relationship between the display and the human visual system can be quantitatively expressed by mathematical relationships of Image Quality Measures (IQMs). Steganographic schemes eventually leave statistical evidence that can be used to quantify the hidden content in the stego image relative to the cover image; see [43]. The metrics measure the similarity between the cover image C and the stego image S after insertion of the message, summing over the set of all bytes Γ as defined by (14):

$$\begin{aligned}
\mathrm{CQ} &= \frac{\sum_{\gamma \in \Gamma} C(\gamma) S(\gamma)}{\sum_{\gamma \in \Gamma} C(\gamma)}, \\
\mathrm{SC} &= \frac{\sum_{\gamma \in \Gamma} C(\gamma)^2}{\sum_{\gamma \in \Gamma} S(\gamma)^2}, \\
\mathrm{IF} &= 1 - \frac{\sum_{\gamma \in \Gamma} \left(C(\gamma) - S(\gamma) \right)^2}{\sum_{\gamma \in \Gamma} I(\gamma)^2}, \\
\mathrm{AD} &= (3mn)^{-1} \sum_{\gamma \in \Gamma} \left| C(\gamma) - S(\gamma) \right|.
\end{aligned} \tag{15}$$

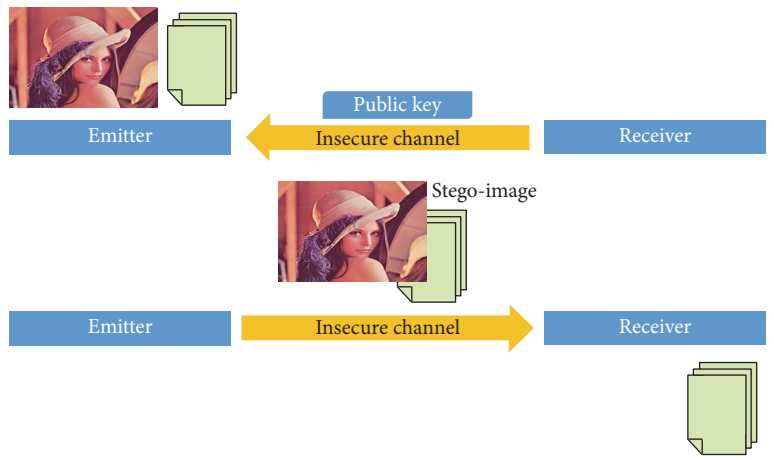

Figure 2: Exchange of information.

Table 1: AD values for the Lenna image.

Keys	Method		
	Proposed	Soria	Velasco
1	0.006847	0.006607	0.005021
3	0.006806	0.007415	0.005021
3	0.006800	0.006327	0.005021
4	0.006673	0.006941	0.005021
5	0.006753	0.006786	0.005021
6	0.006728	0.006656	0.005021
7	0.006812	0.006360	0.005021

The IQMs based on correlation of the content of the images include Correlation Quality (CQ) and Structure Content (SC); see [43, 44]. In addition, the IQMs based on difference distortion and pixel distance include Image Fidelity (IF) and Average Absolute Difference (AD), respectively; see [43, 44].

In the case of the metrics CQ, SC, IF, and AD, the closer the value is to one, the higher the level of similarity is. However, in the case of the metric AD, the closer the value is to zero, the lower the global distortion of the stego image is with respect to the cover image. A large value of AD means that the stego image is very poor in quality.

In the second experiment, the values of CQ, SC, IF, and AD were found for seven pairs of different keys. It can be observed that CQ, SC, and IF tend to one (see Figures 9 and 10) while the AD values are small (see Table 1), which shows that between the cover image and the stego image there are no significant differences.

4.3. Histogram Analysis. In the third experiment we use the Peppers image. The calculation of the previous quantities produced similar results (not shown). In Figure 11, the histograms for the red block of the analyzed image are shown. A distortion metric is the Histogram Similarity (HS), which is calculated as

$$\mathrm{HS} = \sum_{0 \le l \le 255} \left| f_C(l) - f_S(l) \right|, \tag{16}$$

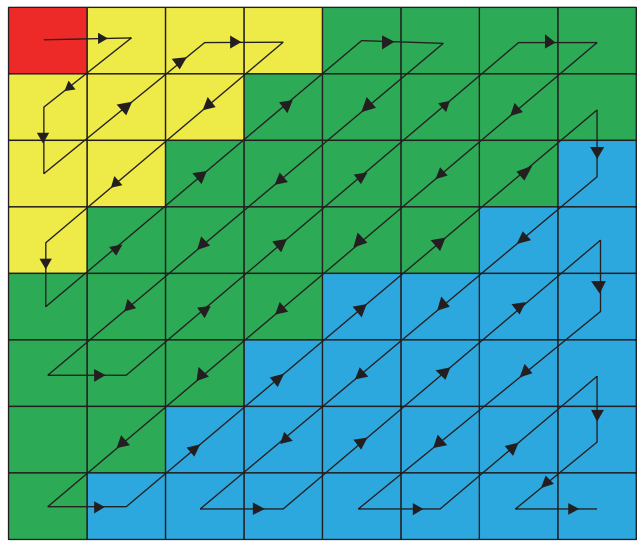

Figure 3: Zigzag order scan.

where $f_C(l)$ is the relative frequency of level l in a 256-level image; see [43, 45]. This measure is connected to the differences between each histogram pair. Table 2 shows that the values of HS are close to zero, which corresponds to the values calculated in the previous example.

4.4. Security Test. The security of a steganographic system can be evaluated beyond examining the distribution of the cover and stego image. Cachin [46] proposed a statistical measure for steganographic systems, which is called ε-secure and given by

$$\mathrm{RE}\left(P_C \parallel P_S\right) = \sum P_C \left| \log \frac{P_C}{P_S} \right| \le \varepsilon, \tag{17}$$

where P_C and P_S represent the distribution of cover image and stego image, and $\mathrm{RE}(P_C \parallel P_S)$ is the relative entropy between the two probability distributions. Moreover, a steganographic system is called perfectly secure if $\mathrm{RE}(P_C \parallel P_S) = 0$.

Table 2: Values of histogram similarity.

Keys	Proposed method	Soria method	Velasco method
		Airplane	
1	0.002534	0.002496	0.002033
2	0.002377	0.002860	0.002033
3	0.002256	0.002476	0.002033
4	0.002457	0.002557	0.002033
5	0.002444	0.002573	0.002033
6	0.002271	0.002406	0.002033
7	0.002387	0.002439	0.002033
		City	
1	0.000965	0.001002	0.000954
2	0.000901	0.001031	0.000954
3	0.000968	0.000987	0.000954
4	0.001051	0.000951	0.000954
5	0.000989	0.000947	0.000954
6	0.001067	0.001072	0.000954
7	0.001104	0.000893	0.000954
		Leaves	
1	0.001501	0.001622	0.001224
2	0.001755	0.001649	0.001224
3	0.001634	0.001568	0.001224
4	0.001648	0.001475	0.001224
5	0.001680	0.001652	0.001224
6	0.001788	0.001631	0.001224
7	0.001643	0.001469	0.001224
		Lenna	
1	0.002055	0.002050	0.001408
2	0.002204	0.002098	0.001408
3	0.002075	0.001983	0.001408
4	0.002003	0.002027	0.001408
5	0.001925	0.001944	0.001408
6	0.002152	0.001948	0.001408
7	0.002068	0.001855	0.001408
		Peppers	
1	0.004432	0.004207	0.002718
2	0.004313	0.004803	0.002718
3	0.004288	0.004031	0.002718
4	0.004490	0.004672	0.002718
5	0.004433	0.004226	0.002718
6	0.004416	0.004138	0.002718
7	0.004485	0.004093	0.002718

Table 3: Values of relative entropy pairs.

Keys	Proposed method	Soria method	Velasco method
		Airplane	
1	0.002525	0.002487	0.002025
2	0.002370	0.002849	0.002025
3	0.002249	0.002468	0.002025
4	0.002449	0.002550	0.002025
5	0.002436	0.002564	0.002025
6	0.002263	0.002398	0.002025
7	0.002378	0.002434	0.002025
		Leaves	
1	0.001501	0.001622	0.001225
2	0.001755	0.001650	0.001225
3	0.001635	0.001569	0.001225
4	0.001648	0.001475	0.001225
5	0.001680	0.001652	0.001225
6	0.001789	0.001631	0.001225
7	0.001644	0.001469	0.001225
		Lenna	
1	0.002055	0.002051	0.001409
2	0.002204	0.002098	0.001409
3	0.002076	0.001983	0.001409
4	0.002004	0.002028	0.001409
5	0.001925	0.001944	0.001409
6	0.002152	0.001949	0.001409
7	0.002068	0.001856	0.001409
		Peppers	
1	0.004259	0.004068	0.002649
2	0.004155	0.004607	0.002649
3	0.004113	0.003909	0.002649
4	0.004323	0.004485	0.002649
5	0.004253	0.004045	0.002649
6	0.004230	0.003969	0.002649
7	0.004308	0.003974	0.002649

In the fourth experiment we observe that the values of the relative entropy are close to zero, which affirms that the steganographic system obtained from the proposed algorithm is sufficiently secure; see Table 3 and Figure 12 corresponding to the City image.

4.5. Steganalysis Experiment: Chi-Square Attack Resistance Test. Steganalysis is the science of detecting hidden information. In other words, steganalysis intends to find the secret information that carries some stego-information by attacking the security of the used steganography algorithm. During the design of secure steganographic algorithms, the simulation of attacks is essential to evaluate the security. In this work some of the stego images produced by the Soria-Lorente method, the Velasco-Bautista method, and the proposed method have been tested against the well-known Chi-square attack [47]. The Chi-square attack is one of the algorithms to test the detectability of any secret data embedding algorithm. This test is based on the frequency with which pixel values appear. It is unusual for the frequency of pixel value $2j$ to be (nearly) equal to the frequency of pixel value $2j + 1$ in a typical image with no embedded information. The Chi-squared attack was designed to detect these near-equal biases in images and bases the probability of embedding on how close to equal the even pixel values and their corresponding odd pixel values are in the test image. Indeed, in [48] it was shown that the Chi-square attack detects successfully the existence of the embedded secret bits in the first half of the DCT coefficients

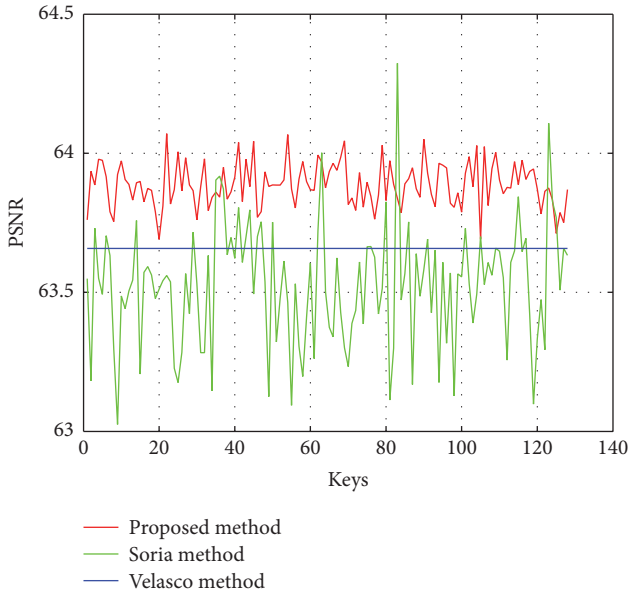

Figure 4: PSNR values for the Airplane image.

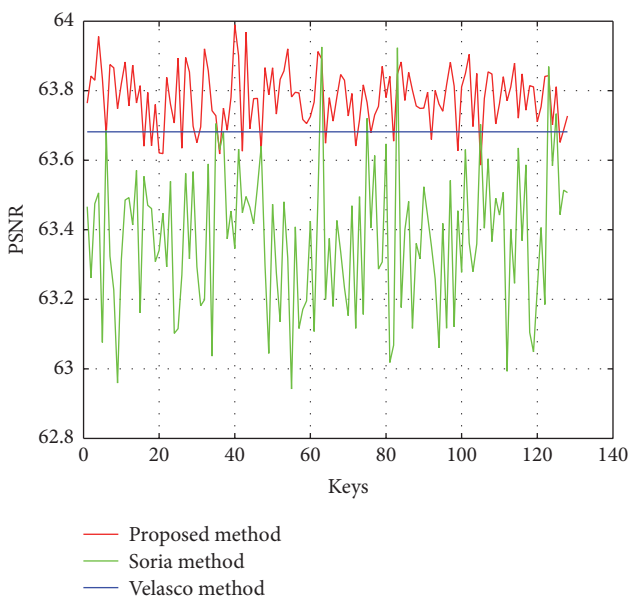

Figure 5: PSNR values for the City image.

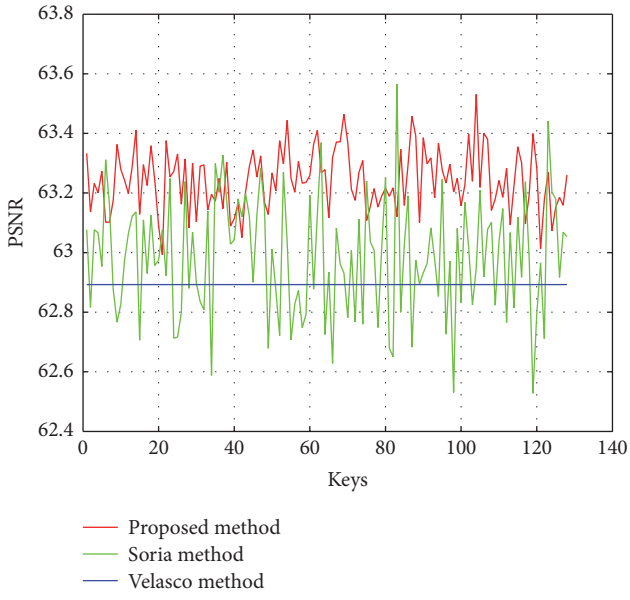

Figure 6: PSNR values for the Leaves image.

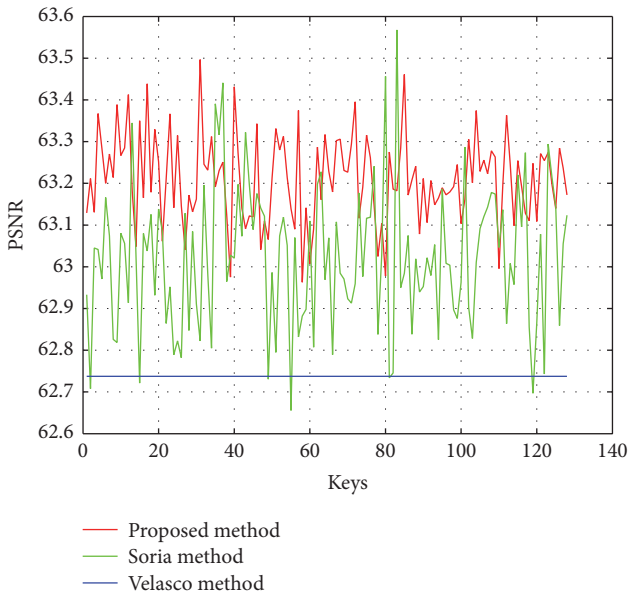

Figure 7: PSNR values for the Lenna image.

of the stego image created by the Jpeg-Jsteg method; see also [32].

The Chi-square χ^2_{k-1} statistic with $k-1$ degrees of freedom is given as

$$\chi^2_{k-1} = \sum_{1\le j\le k} \frac{\left(\eta_j - E\left(\eta_j\right)\right)^2}{E\left(\eta_j\right)}, \quad E\left(\eta_j\right) = \frac{\eta_j + \lambda_j}{2}, \tag{18}$$

where η_j and λ_j are the total pixels in image sample of gray value $2j$ and $2j+1$, respectively. The values of the Chi-square statistic are always positive. The expectation is that, for a stego image, χ^2_{k-1} is relatively small because η_j should be near λ_j, by the hypothesis, and for a non-stego image, χ^2_{k-1} is relatively large because η_j should be far from λ_j. The final step of the process is calculating p, the probability of embedding, by integrating the density function with χ^2_{k-1} as its upper limit:

$$p = 1 - \frac{\sqrt{2^{1-k}}}{\Gamma\left((k-1)/2\right)} \int_0^{\chi^2_{k-1}} e^{-x/2} x^{(k-1)/2-1} dx. \tag{19}$$

This probability of embedding is the probability of χ^2_{k-1} under the condition that $\eta_j = \lambda_j$ for $1 \le j \le k$. The density function, $1-p$, converges to 1 as χ^2_{k-1} approaches infinity, so p approaches 0 as χ^2_{k-1} approaches infinity. Therefore, for large χ^2_{k-1}, the probability of embedding is near 0. In other

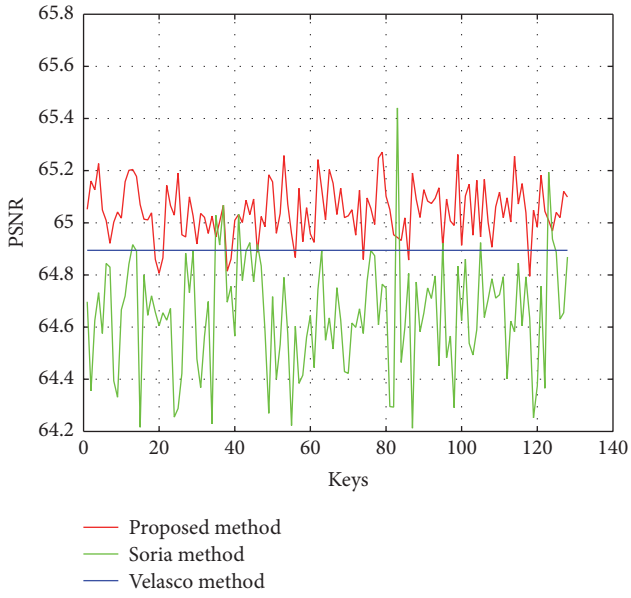

Figure 8: PSNR values for the Peppers image.

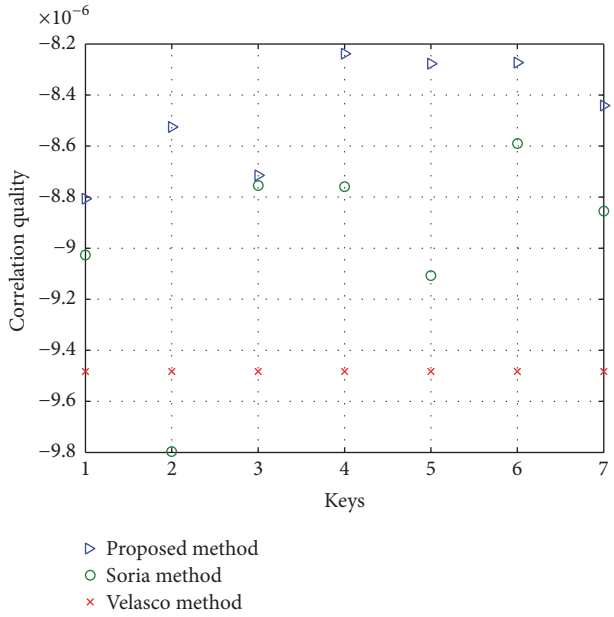

Figure 9: CQ values for the Lenna image (displayed: CQ − 1).

words, larger values of the χ^2_{k-1} mean a lower probability of embedding. However, when χ^2_{k-1} is small relative to $k-1$, then $1-p$ is near to zero and hence p is near to 1. Thus for relatively small χ^2_{k-1}, the probability of embedding is near 1.

In the fifth experiment we verify that the stego images created by the Soria-Lorente method and the proposed method cannot be detected by the Chi-square attack; see Figures 13 and 14. This means that the proposed method is robust against the Chi-square attack. However, in Figure 14, one can observe that the Chi-square attack detects the existence of the embedded secret bits approximately in 1.02% of the

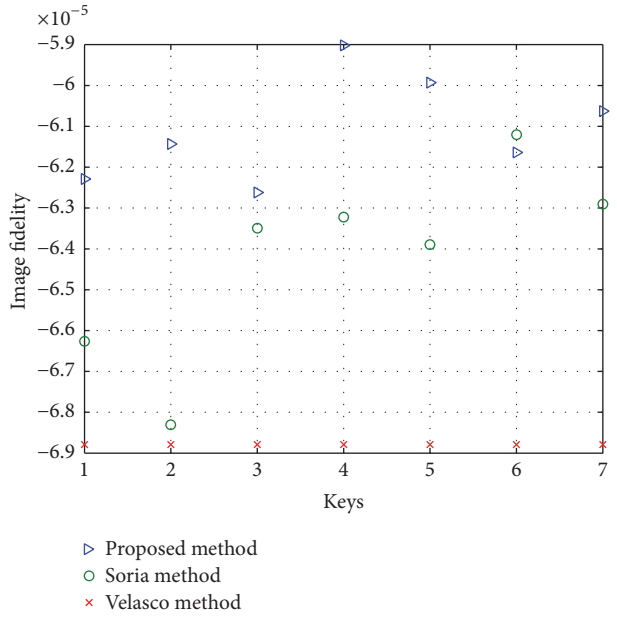

Figure 10: IF values for the Lenna image (displayed: IF − 1).

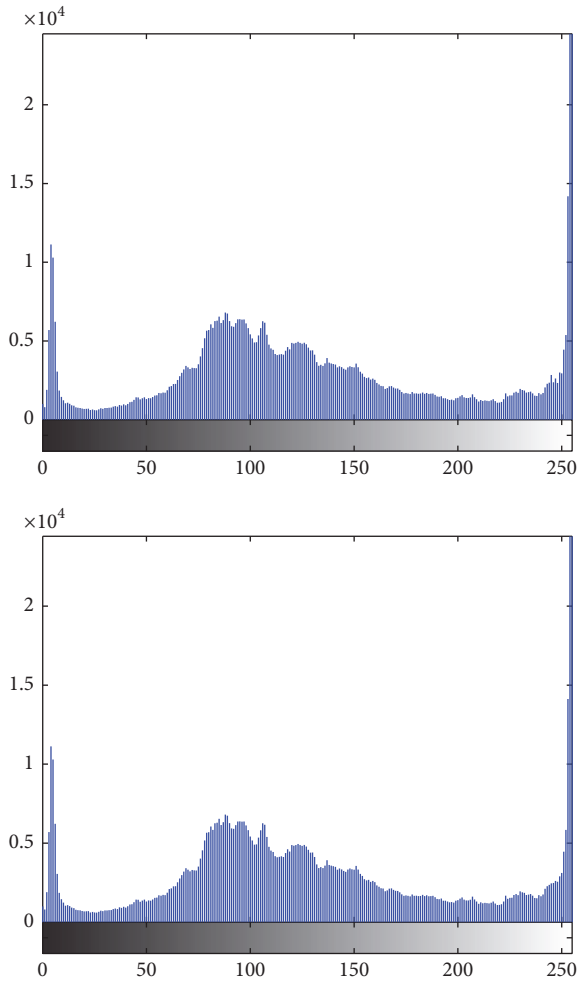

Figure 11: The histogram of Peppers image before and after embedding the secret message.

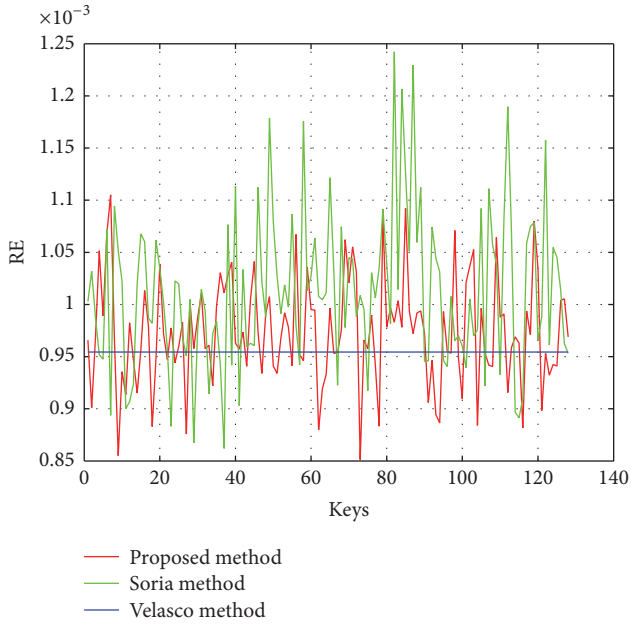

Figure 12: RE values for the City image.

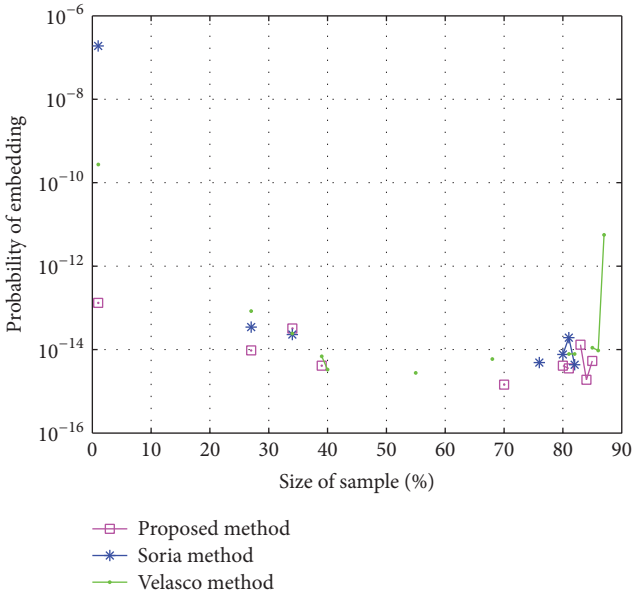

Figure 13: The detection results of the Chi-square attack to the stego image (Airplane) created by the Soria-Lorente method, the Velasco-Bautista method, and the proposed method.

stego images created by the Velasco-Bautista method, with a probability of embedding of 0.94.

5. Conclusions and Discussion

In this contribution, we propose a new steganographic algorithm which uses two keys, one public and another private, in order to reduce the detectability. According to the analyses of PSNR, of histograms and IQMs values, it is demonstrated that in the stego image there are no detectable anomalies with

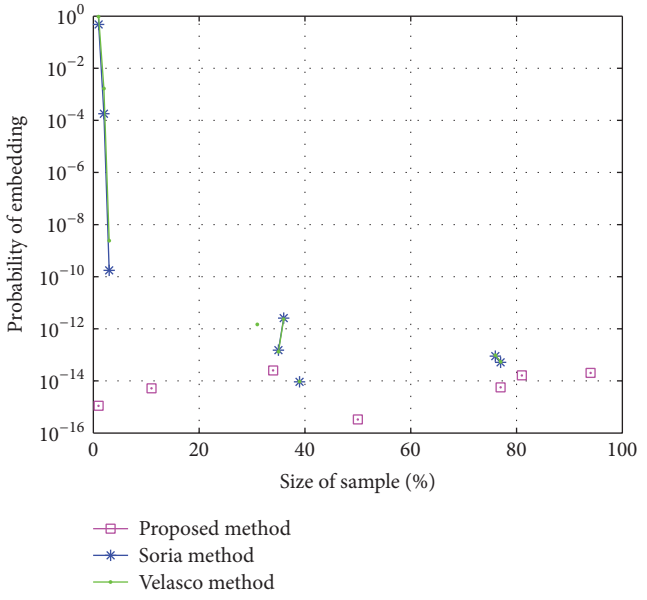

Figure 14: The detection results of the Chi-square attack to the stego image (City) created by the Soria-Lorente method, the Velasco-Bautista method, and the proposed method.

respect to the cover image. Moreover, the obtained values for the relative entropy show that the steganographic system obtained by the proposed algorithm is sufficiently secure. In addition, by the values of embedding probability it was demonstrated that the proposed algorithm is highly resistant against the Chi-square attack.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The authors wish to thank the Clavemat project, financed by the European Union, the University of Granma, and Fundación EPAS. The Universidad Católica de Temuco supported the authors through the project DGIPUCT no. 2015CA-SF-04 (supported by the "Dirección General de Investigación y Postgrado") and through the Master Program of Applied Mathematics.

References

[1] A. Soria, R. Cumbrera, and Y. Fonseca, "Steganographic algorithm of private key on the domain of the cosine discrete transform," *Revista Cubana de Ciencias Informáticas*, vol. 10, no. 2, pp. 116–131, 2016.

[2] G. Kumar and A. Rana, "Data hiding techniques in digital multimedia," *International Journal of Engineering Science Invention Research and Development*, vol. 1, pp. 333–337, 2015.

[3] V. M. Ladwani and S. Murthy K, "A new approach to securing images," *IJARCCE: International Journal of Advanced Research*

in Computer and Communication Engineering, vol. 4, no. 1, pp. 224–227, 2015.

[4] P. Lino Babu, J. John, B. D. Parameshachari, C. Muruganantham, and H. S. DivakaraMurty, "Steganographic method for data hiding in audio signals with LSB and DCT," *International Journal of Computer Science and Mobile Computing*, vol. 2, pp. 54–62, 2013.

[5] K. S. Jenifer, G. Yogaraj, and K. Rajalakshmi, "LSB approach for video steganography to embed images," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 1, pp. 319–322, 2014.

[6] D. Wu and W. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognition Letters*, vol. 24, no. 9-10, pp. 1613–1626, 2003.

[7] S. Sachdeva, A. Sharma, and V. Gill, "Colour image steganography using modified JPEG quantization technique," *International Journal of Latest Research in Science and Technology*, vol. 1, pp. 1–5, 2012.

[8] X. Liao, Q. Wen, and J. Zhang, "A steganographic method for digital images with four-pixel differencing and modified LSB substitution," *Journal of Visual Communication and Image Representation*, vol. 22, no. 1, pp. 1–8, 2011.

[9] T. Lu, C. Tseng, and J. Wu, "Dual imaging-based reversible hiding technique using LSB matching," *Signal Processing*, vol. 108, pp. 77–89, 2015.

[10] M. Kumar and M. Yadav, "Image steganography using frequency domain," *International Journal of Scientific and Technology Research*, vol. 3, pp. 226–230, 2014.

[11] M. Amin, H. M. Abdullkader, H. M. Ibrahem, and A. S. Sakr, "A steganographic method based on DCT and new quantization technique," *International Journal of Network Security*, vol. 16, no. 4, pp. 265–270, 2014.

[12] S. Mitra, M. Dhar, A. Mondal, N. Saha, and R. Islam, "DCT based Stegano graphic Evaluation parameter analysis in Frequency domain by using modified JPEG luminance Quantization Table," *Journal of Computer Engineering*, vol. 17, no. 1, pp. 68–74, 2015.

[13] S. Kalaivanan, V. Ananth, and T. Manikandan, "A survey on digital image steganography," *International Journal of Emerging Trends & Technology in Computer Science*, vol. 4, no. 1, pp. 30–33, 2015.

[14] J. Mazumder and K. Hemachandran, "A high capacity and secured color image steganographic technique using discrete wavelet transformation," *International Journal of Computer Science and Information Technologies*, vol. 4, no. 4, pp. 583–589, 2013.

[15] P. Patil and D. S. Bormane, "DWT based invisible watermarking technique for digital images," *International Journal of Engineering and Advanced Technology*, vol. 2, no. 4, pp. 603–605, 2013.

[16] J. Mali, V. Sonawane, and R. Awale, "Image steganography using block level entropy thresholding technique," *Journal of Biological Systems*, vol. 3, pp. 141–152, 2013.

[17] P. Praveenkumar, G. Ashwin, S. P. Kartavya Agarwal et al., "Rubik's Cube Blend with logistic map on RGB: a way for image encryption," *Research Journal of Information Technology*, vol. 6, no. 3, pp. 207–215, 2014.

[18] R. Amirtharajan, V. Rajesh, P. Archana, and J. B. B. Rayappan, "Pixel indicates, standard deviates: a way for random image steganography," *Research Journal of Information Technology*, vol. 5, no. 3, pp. 383–392, 2013.

[19] V. Jain, L. Kumar, M. Mohan, M. Sadiq, and K. Rastogi, "Public-key steganography based on modified LSB method," *Journal of Global Research in Computer Science*, vol. 3, no. 4, pp. 26–29, 2012.

[20] S. Ahmed Laskar, "Secure data transmission using steganography and encryption technique," *International Journal on Cryptography and Information Security*, vol. 2, no. 3, pp. 161–172, 2012.

[21] S. Narayana and G. Prasad, "Two new approaches for secured image steganography using cryptographic techniques and type conversions," *Signal & Image Processing*, vol. 1, no. 2, pp. 60–73, 2010.

[22] M. J. Saeed, "A new technique based on chaotic steganography and encryption text in DCT domain for color image," *Journal of Engineering Science and Technology*, vol. 8, no. 5, pp. 508–520, 2013.

[23] T. Shahanar, "A secure DCT image steganography based on public-key cryptography," *International Journal of Computer Trends and Technology*, vol. 4, no. 7, pp. 2039–2043, 2013.

[24] M. S. Subhedar and V. H. Mankar, "Current status and key issues in image steganography: a survey," *Computer Science Review*, vol. 13-14, pp. 95–113, 2014.

[25] N. Provos and P. Honeyman, "Hide and seek: an introduction to steganography," *IEEE Security and Privacy*, vol. 1, no. 3, pp. 32–44, 2003.

[26] A. Westfeld, "F5-a steganographic algorithm: high capacity despite better steganalysis," in *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings*, vol. 2137 of *Lecture Notes in Computer Science*, pp. 289–302, Springer, Berlin, Germany, 2001.

[27] N. Provos, "Defending against statistical steganalysis," in *Proceedings of the 10th Conference on USENIX Security Symposium (SSYM '01)*, vol. 10, pp. 323–336, Washington, DC, USA, 2001.

[28] T. Pevný, T. Filler, and P. Bas, "Using high-dimensional image models to perform highly undetectable steganography," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6387, pp. 161–177, 2010.

[29] R. Cogranne, C. Zitzmann, F. Retraint, I. V. Nikiforov, P. Cornu, and L. Fillatre, "A local adaptive model of natural images for almost optimal detection of hidden data," *Signal Processing*, vol. 100, pp. 169–185, 2014.

[30] M. Afrakhteh and J.-A. Lee, "Adaptive least significant bit matching revisited with the help of error images," *Security and Communication Networks*, vol. 8, no. 3, pp. 510–515, 2015.

[31] C. Chang, T. Chen, and L. Chung, "A steganographic method based upon JPEG and quantization table modification," *Information Sciences*, vol. 141, no. 1-2, pp. 123–138, 2002.

[32] H. Noda, M. Niimi, and E. Kawaguchi, "High-performance JPEG steganography using quantization index modulation in DCT domain," *Pattern Recognition Letters*, vol. 27, no. 5, pp. 455–461, 2006.

[33] K. Wong, X. Qi, and K. Tanaka, "A DCT-based Mod4 steganographic method," *Signal Processing*, vol. 87, no. 6, pp. 1251–1263, 2007.

[34] C.-C. Chang, C.-C. Lin, C.-S. Tseng, and W.-L. Tai, "Reversible hiding in DCT-based compressed images," *Information Sciences*, vol. 177, no. 13, pp. 2768–2786, 2007.

[35] X. Li and J. Wang, "A steganographic method based upon JPEG and particle swarm optimization algorithm," *Information Sciences*, vol. 177, no. 15, pp. 3099–3109, 2007.

[36] C. L. Velasco-Bautista, J. C. López-Hernández, M. Nakano-Miyatake, and H. M. Pérez-Meana, "Esteganografía en una imagen digital en el dominio DCT," *Científica*, vol. 11, no. 4, pp. 169–176, 2007.

[37] C.-C. Lin and P.-F. Shiu, "High capacity data hiding scheme for DCT-based images," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, no. 3, pp. 220–240, 2010.

[38] Y.-K. Lin, "High capacity reversible data hiding scheme based upon discrete cosine transformation," *The Journal of Systems and Software*, vol. 85, no. 10, pp. 2395–2404, 2012.

[39] H. L. Jaheel and Z. Beiji, "A novel approach of combining steganography algorithms," *International Journal on Smart Sensing and Intelligent Systems*, vol. 8, no. 1, pp. 90–106, 2015.

[40] B. Schneier, *Applied Cryptography. Protocols, Algorithms and Source Codein C*, John Wiley & Sons, New York, NY, USA, 2nd edition, 1996.

[41] L. Yu, Y. Zhao, R. Ni, and Z. Zhu, "PM1 steganography in JPEG images using genetic algorithm," *Soft Computing*, vol. 13, no. 4, pp. 393–400, 2009.

[42] İ. Coşkun, F. Akar, and Ö. Çetin, "A new digital image steganography algorithm based on visible wavelength," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 21, pp. 548–564, 2013.

[43] M. Kutter and A. P. Petitcolas, "A fair benchmark for image watermarking systems," in *Electronic Imaging '99. Security and Watermarking of Multimedia Contents*, vol. 3657, pp. 1–14, 1999.

[44] C. Varnan, A. Jagan, J. Kaur, D. Jyoti, and D. Rao, "Image quality assessment techniques pn spatial domain," *International Journal of Computer Science and Technology*, vol. 2, no. 3, pp. 177–184, 2011.

[45] A. K. Gulve and M. S. Joshi, "A high capacity secured image steganography method with five pixel pair differencing and lsb substitution," *International Journal of Image, Graphics and Signal Processing*, vol. 7, no. 5, pp. 66–74, 2015.

[46] C. Cachin, "An information-theoretic model for steganography," *Information and Computation*, vol. 192, no. 1, pp. 41–56, 2004.

[47] O. Zanganeh and S. Ibrahim, "Adaptive image steganography based on optimal embedding and robust against chi-square attack," *Information Technology Journal*, vol. 10, no. 7, pp. 1285–1294, 2011.

[48] C.-L. Liu and S.-R. Liao, "High-performance JPEG steganography using complementary embedding strategy," *Pattern Recognition*, vol. 41, no. 9, pp. 2945–2955, 2008.