electronics

MDPI

Article

Enhancing the Performance of the Data Embedment Process through Encoding Errors

A. H. M. Kamal [1,2,*] and Mohammad Mahfuzul Islam [2]

[1] Department of Computer Science and Engineering, Jatiya Kabi Kazi Nazrul Islam University, Trishal, Mymensingh, P.C. 2220, Bangladesh

[2] Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka 1000, Bangladesh; mahfuz@cse.buet.ac.bd

* Correspondence: kamal@jkkniu.edu.bd; Tel.: +880-173-222-6402

Academic Editor: Mostafa Bassiouni

Received: 21 August 2016; Accepted: 14 November 2016; Published: 22 November 2016

Abstract: Image steganography is a multipurpose-serving key emerging technology that is used for covertly transferring, storing, and governing various digital data, including intellectual properties and copyrights, social media data, multimedia data, and secrets of law-enforcing agencies. During the management in the stated information, nowadays, massive amounts of data are handled that require greater security. For that purpose, data are embedded into a cover image to hide them from any intruders. Nevertheless, the requirements of a larger embedding capacity, improved stego-image quality, and reduced time complexity is increasing. In this paper, the authors have presented a novel data-embedding scheme where the prediction error-based data-hiding scheme is modified in an intricate way so that all the image pixels can accept secret bits. A distance matrix between the pixel values of each image block and a reference value are measured first. Thereafter, the distances are encoded into two states: 1 and −1. That encoding process enables the scheme to implant one bit in every pixel of the cover image. During the bit implantation, the errors 1 and −1 are modified by shifting them to the right and left directions, respectively. This strategy enhances the embedding capacity by a factor of more than 2. The use of reference values reduces the computational complexity notably, and in the meanwhile increases the security and robustness of the scheme because the reference values are not open to any third party. The scheme also reduces the time complexity by 2–16 times with compared to its competing schemes. Experimental results prove the superiority of the proposed algorithm on embedding capacity, visual quality, and time complexity compared to the current well-accepted existing schemes.

Keywords: diamond encoding; pixel-pair; encoded errors; reference pixel; embedding capacity; visual quality; processing complexity

1. Introduction

Steganography is the art of hiding information within other media, known as "cover media". The objective of steganography is to conceal secret data into the contents of the "cover media" in a manner that renders the probability of unintended users finding this secret data extremely low [1]. This way of securing information in communication technology differentiates itself from the well-known encryption method because in the encrypted information one can presume the existence of the secret data and try guessing key(s) or a portion of message to decipher the secrets [2]. Another information concealment method in the field of secure data communication is watermarking. Although, it ensures the integrity of the data, it limits the secret data communication standard by exhibiting the concealed data to the third party [3]. Hence, steganography is hastily becoming a bewitching process in the field of secret data transmission for both wire and wireless communications. Consequently, its' uses are

spreading from hiding regular data to more specialized information, including medical [3] (a unique contribution by the authors on offline healthcare service) and forensic data [4]. As a working space, it is expanding its scope from worldwide communication to standalone applications like authenticating a smart card inserted into a terminal [2] or storing data by embedding them inside media [5]. For the reason of data security, nowadays, digital data like intellectual properties, copyrights, documents of forensics and law-enforcing agencies, social media data, and multimedia data in various applications are covertly transmitted, stored, and governed by this data embedment strategy.

Based on the embedding domain, image steganography can be classified into four subgroups—image domain [6–9], transform domain [10,11], histogram domain [12–15], and compressed domain [16–18]. In the image domain, also known as the spatial domain, embedding is done directly into pixel values. The common methods used in such domain are the replacement of the least significant bit (LSB), LSB matching, and addition or subtraction to/from the pixels' value. Another useful procedure is to transform the image data into coefficients such as the processes of discrete cosine transform (DCT) coefficients, first Fourier transforms (FFT) coefficients, wavelet transforms coefficients, and so on. These coefficients are used as an embedding space. After embedding, an inverse transformation is performed to return to the image domain. One more useful and famous technique is to embed information into the histogram of the image data or to the histogram of some processed information like pixel differences and prediction errors. A prediction-error histogram is generated from the difference of the cover values and their respective predicted values [13]. These errors are used as the embedding space. There, the histograms are either shifted unevenly or associated and reflected evenly by the pattern of the chunk of the concealed data. The other less likely used space is the compressed domain where the image data is first compressed by any standard method like vector quantization (VQ) [16,17], M-sequence coding [18] or truncation [19], and then the secret data is embedded contents.

Embedding processes are classified into two other major categories—reversible and reversible—based on their capability of extracting the implanted secret data as well as recovering the original image pixels. The data embedment rules modify the cover pixels by implanting the secret data. At the receiver end, the decoder retrieves the concealed information from the stego image. If the decoder extracts only the secrets and becomes unable to reconstruct the cover image, the method is termed as the irreversible embedding process [20–25]. On the other hand, if the decoder reconstructs the cover image during its message extraction, the process is referred to as a reversible data hiding (RDH) scheme [26–30]. The reversible data hiding schemes, by nature, offer less embedding capacity than irreversible embedding schemes, because to manage the reversibility, these schemes implant some addition information into the image and leave many of the pixels without concealing any bits. These nonconceived pixels are also shifted by an amount stipulated by the embedding rules. Consequently, the ratio of distortion to embedding capacity in the reversible processes is notably higher than the irreversible procedures. Nevertheless, the associated activities—such as measuring prediction errors or pixel differences, implanting additional bits, and defining the embeddable contents—have enriched the security of the reversible process because without the exact knowledge of this associated information, data extraction is impossible. Therefore, the involvement of such an associated activity of the reversible scheme (e.g., generation of prediction errors as an embedding space) in an irreversible process will enhance both the security of the implanted data and the embedding capacity.

In 2009, Chao et al. [30] proposed a diamond encoding (DE)-based irreversible data embedment process. A neighborhood function is used to make a diamond from each pair of pixels. The number of cells in the diamond is measured from a polynomial function. In 2012, Hong and Chen [28] observed that the diamond shape is not a compact region. The neighborhood function and the cells' set of the diamond are redefined by this scheme. During the embedment of a digit, in both the cases (i.e., in [28,30]) a pair of pixels are replaced by another pair in the cell of the diamond. In the scheme of Hong and Chen, the embedding capacity and the image quality are improved by the application of adaptive pixel pair matching. Again, Hong et al. [6] in 2012 modified Chao et al.'s scheme to embed data at multiple bases of the diamond. Bases are generated by partitioning the gray scale into multiple

parts. The shifting of pixels are bounded within the base. The difference of each adjacent pixel is used to map a gray partition. This strategy has improved the stego-image quality by reducing the amount of displacement of the stego pixels caused by the data embedment. In 2011, Liao et al. [9] implanted bits by replacing the least significant bits (LSBs) of the image pixels. They divided the cover image into blocks of four pixels. The average distance of the pixels from the minimum one in the block is measured. If the distance value is smaller than a threshold, fewer LSBs are replaced by the data bits; otherwise, a greater number of bits are replaced. This way, message bits are embedded into all blocks by the rules of LSB substitutions. Although, the scheme notably increases the embedding capacity, it noticeably decreases the image quality. In 2010, Hong et al. [13] proposed a reversible scheme to embed into two-peak presented errors in the prediction error histogram. The embedding rate of that scheme only depends on the frequencies of these two-peak presenting errors.

These diamond encoding-oriented irreversible schemes show some limitations. The calculation of diamond characteristic values (i.e., cells sets) during the diamond generation and again searching the characteristic values for a matching of pixel pair (i.e., matching of two pixel values with two diamond characteristic values) are time consuming processes. If the number of bits in each pair of pixels is a single bit, the scheme cannot present higher embedding capacity; rather, the distortions for each bit of data embedment is increased. In addition, the partitioning of image is fixed to two pixels only. On the contrary, the reversible scheme of Hong et al. [13] has minimized the amount of pixel shifting in the stego image by allowing the pixels to be shifted by one unit at most. Kamal and Islam [23] in 2015 improved the embedding capacity and the image quality regarding the results of [13] by introducing a new predictor that utilizes the Euclidean distance in their prediction rules. That scheme does not modify the embedding rules. The same authors, as well as the authors of this article, in 2016, [21] have applied multiple predictors to improve the prediction accuracy. Although this attempt has improved the prediction accuracy, the application of multiple predictors has increased the time complexity. These authors, in another work in 2016 [22], have improved the embedding capacity my embedding into several errors multiple times. However, this not a single-layer data embedment process. Besides, this increases the time complexity by embedding multiple times. The same authors, in [20,24], have improved the embedding capacity by employing prediction errors for shifting a block of pixels from one gray part to another. These works are devoted to improving the embedding capacity in the research area of the image distortion-based scheme, where a stego image is distorted to a level so that the cover contents are not sensible. This is another area of image steganography. Hence, in the area of quality preservation-based embedment processes, the application of the difference of an adjacent pixel (as is observed in [6]) and the embedding rules of [13,21–23] will improve both the embedding rate, the stego-image quality, as well as the security. In this paper, the scheme of [6,13,23] are modified by measuring the difference of each of the pixels from a reference pixel rather than from two adjacent pixels or block of pixels. This reference pixel will also improve the data security, as it is a private key. An encoder is used to encode these differences into two values only. The encoded differences are distributed only into two bins in the histogram of the encoded difference values. The embedding rules allow all the encoded difference values to accept a message bit. Thus, the quantity of embedded bits and the image quality are enhanced. Such a scheme will aid many applications running in devices like robots, personal digital assistance, and mobiles to embed massive data as well as manage the visual quality of a stego image. The security provided by that scheme will also make it effective for uses in such robotic applications/services.

The remaining paper contains four sections. The following first and foremost section is devoted to narrate the proposed work. Sections 3 and 4 delineate the embedding efficiency and resistance against threats, respectively. Finally, concluding remarks are provided in Section 5.

2. Proposed Scheme

Let the cover image and the secret message stream be I and S, respectively. Each secret message bit s (i.e., $s \varepsilon S$) is implanted into each different pixel of the image (e.g., s is implanted into the pixel $I_{i,j}$,

where (i, j) indicates the location of the pixel in the image). The size of the image is $h \times w$, where h and w are the height and width of the image, and these are divisible by m and n, respectively. Now divide the cover image into $bl = (h \times w)/(m \times n)$ blocks of size $sz = (m \times n)$ each as it is shown in the Figure 1a,b.

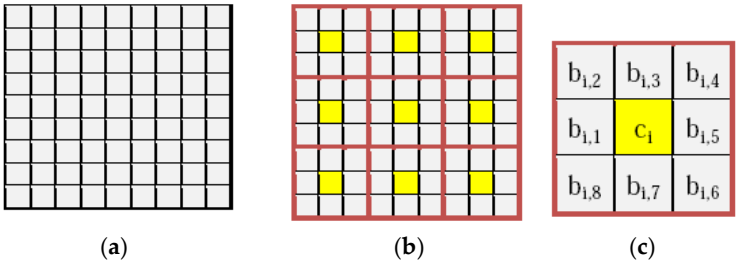


Figure 1. Partitioning the cover image into blocks. Specification of the above figures are: (**a**) cover image I; (**b**) I, partitioned by 3×3 blocks; and (**c**) ith Block.

The reference pixel can be of any value within the range from 0 to 255. This is either a negotiated value, a derived value, or a pixel value in the image (e.g., the center pixel of each block or the first pixel of the image). For convenience of better explaining the proposed scheme, let the center pixel c_i of each block i be used as a reference value for the other pixels $b_{i,j}$ in the block. These $b_{i,j}$ values of each block are read along a spiral path, as shown in the Figure 1c.

2.1. Measuring Distance from the Reference Pixel

Firstly, the distances $d_{i,j}$—indeed, the differences of all the jth pixels in the ith block from the reference pixel c_i—are calculated by Equation (1).

$$d_{i,j} = b_{i,j} - c_i \text{ for } 1 \leq j \leq (m \times n - 1) \tag{1}$$

The range of the values of $d_{i,j}$ is from -255 to 255; $d_{i,j}$ can then be categorized into two groups by based on whether it is less than 0 or not.

2.2. Calculating Encoded Errors and Encoded Values

The reference value distances $d_{i,j}$ are encoded into one of two values, say encoded errors $e_{i,j}$, by Equation (2).

$$e_{i,j} = \begin{cases} 1 \; if \; d_{i,j} \geq 0 \\ -1 \text{ otherwise} \end{cases} \tag{2}$$

Considering these encoded errors are similar to the prediction errors, we relate that process with the prediction error-based scheme. The concept of the predicted values is replaced here by the terminology of encoded values, where the encoded value $ep_{i,j}$ for each encoded error $e_{i,j}$ is found by Equation (3).

$$ep_{i,j} = b_{i,j} - e_{i,j} \tag{3}$$

These $e_{i,j}$ are then used as an embedding space.

2.3. Embedding Message Bits

First, the length of the message is measured, say this is L. This L is converted to 24-bit binary number. These 24 bits are embedded into the first three pixels of the cover image by replacing their values. Thereafter, the message bits are embedded into the encoded errors. During this data embedment process, the first three encoded errors of the first row of the error matrix are skipped so that this does not affect the values of the first three pixels, where the message length is embedded. Next,

each message bit s of the secret message stream S is embedded into each of the remaining encoded errors in the error matrix $e_{i,j}$. The embedding procedure modifies the error matrix by Equation (4).

$$\widetilde{e}_{i,j} = e_{i,j} + e_{i,j}\left(s\ XOR\ mod\left(\left|d_{i,j}\right|,\ 2\right)\right) \tag{4}$$

If the reference values are chosen from the image pixels, the list of these pixels are not changed during the data embedment. Consequently, their corresponding errors are skipped in Equation (4).

Due to the error modification, the cover pixels with values of 0 or 255 may exceed the gray scale in the stego image (e.g., -1 or 256). These events of exceeding the gray scale by the stego values are termed underflow and overflow, respectively. The overflow or underflow is handled by Equation (5).

$$\widetilde{e}_{i,j} = 2e_{i,j} - \widetilde{e}_{i,j} \text{ if } b_{i,j} = 255\ or\ b_{i,j} = 0 \tag{5}$$

Finally, the stego pixels $\widetilde{b}_{i,j}$ are calculated adding the modified errors $\widetilde{e}_{i,j}$ to the encoded values $ep_{i,j}$ in Equation (6).

$$\widetilde{b}_{i,j} = ep_{i,j} + \widetilde{e}_{i,j} \tag{6}$$

The whole embedding process is explained in Example 1.

2.4. Minimizing Distortion

Equation (4) states that the XOR operation $\left(s\ XOR\ mod\left(\left|d_{i,j}\right|,\ 2\right)\right)$ is responsible for shifting the encoded error values by 1 or -1. To minimize the amount of shifting, the message bits are processed first. The whole message is divided into t blocks BK_t of length l each. All the $d_{i,j}$ values are converted to a single dimensional data array by Equation (7).

$$dr_{(i-1)*(sz-1)+j} = d_{i,j} \tag{7}$$

The number of pixels to be shifted, say δ_t, among l pixels are computed by Equation (8), checking all the bits s_k of BK_t, where $1 \leq k \leq l$.

$$[\mathbb{C}_{t,k},\ \delta_t] = \Psi_t\left(s_k\ XOR\ mod\left(dr_{(t-1)*l+k},\ 2\right)\right) \tag{8}$$

Here, Ψ_t returns two values $\mathbb{C}_{t,k}$ and δ_t. The $\mathbb{C}_{t,k}$ is a matrix that contains the results of XOR operation $(s_k\ XOR\ mod\left(dr_{(t-1)*l+k},\ 2\right))$ for tth chunk and $\delta_t = \sum_k \mathbb{C}_{t,k}$.

If it holds that $\delta_t > {}^{l}/_{2}$, a complementation of message bits in BK_t is taken, which, indeed, results in $\overline{\mathbb{C}}_{t,k} = \left(\mathbb{C}_{t,k}\ XOR\ 1\right)$. Otherwise, set $\overline{\mathbb{C}}_{t,k} = \mathbb{C}_{t,k}$. Because $\overline{\mathbb{C}}_{t,k}$ is used in Equation (4) directly (e.g., $\widetilde{e}_{i,j} = e_{i,j} + e_{i,j} * \overline{\mathbb{C}}_{t,k}$), the computation time is minimized. A binary data array β_t is maintained to keep track of whether a complementation is done or not. This β_t is concatenated to the end of the message. This portion is executed just after Equation (3) and before the start of the message-embedding part. The process is explained here, rather than above, so that it does not introduce any ambiguity in realizing our proposed scheme.

Example 1. *Consider the following image block of size 3 × 3 in Figure 2a and the value of the center pixel, which is 220 in this example, as a reference value. Equations (1)–(3) result in Figure 2b–d, respectively.*

Let the to-be-embedded message stream be 01101110, (i.e., S = 01101110). The outcome of Equation (4) is depicted in Figure 2e. Again, to remove the problem of underflow and overflow, Equation (5) is applied on the modified errors in Figure 2e, and the result is tabulated in Figure 2f. Finally, the stego block is generated by applying Equation (6). The stego block is presented in Figure 2g.

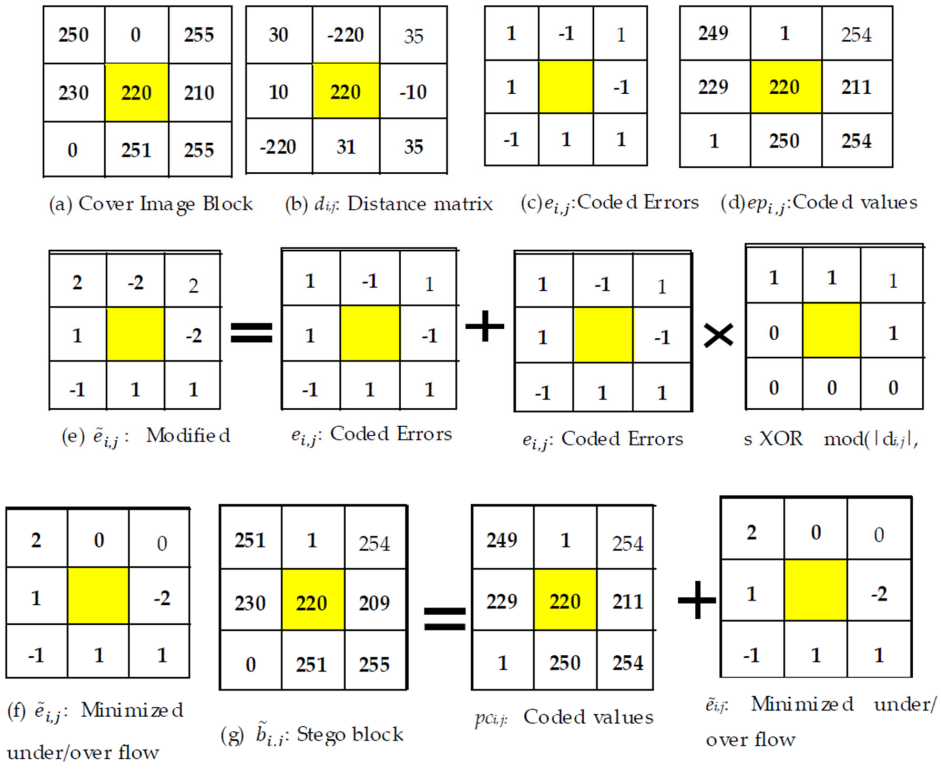


Figure 2. The complete message-embedding procedure.

2.5. Message Extraction at the Receiver

The message extractor (i.e., the decoder) divides the stego image into $bl = (h \times w)/(m \times n)$ blocks of size $sz = (m \times n)$ each. The reference values are unchanged at the embedding phase by the data hider. The absolute distance between each stego pixel $\widetilde{b}_{i,j}$ and its reference value c_i is calculated by Equation (9).

$$\widetilde{d}_{i,j} = \left|\widetilde{b}_{i,j} - c_i\right| \text{ for } 1 \leq j \leq (m \times n - 1) \tag{9}$$

Lastly, the secret message bit s_j is extracted from that distance matrix by Equation (10).

$$s_j = mod\left(\widetilde{d}_{i,j}, 2\right) \tag{10}$$

The scenario of message extraction is demonstrated in Example 2.

Example 2. *The distance matrix of Figure 3b and extracted message bits of Figure 3c are found by applying, respectively, Equations (9) and (10) in Figure 3a.*

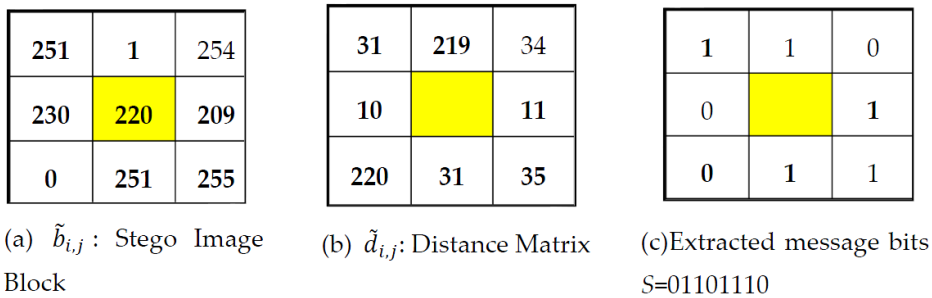


Figure 3. Message extraction process.

3. Performance Analysis in Terms of Embedding Efficiency

In the field of steganography, three performance parameters are used in measuring the embedding efficiency. These are payload or capacity, visual quality, and processing complexity. The visual quality is commonly assessed either by mean square error (MSE) or peak signal-to-noise ratio (PSNR). The embedding capacity, MSE, and PSNR of an image of size $m \times n$ are computed by Equations (11)–(13), respectively.

$$Capacity = \frac{Total\ Embedded\ bits}{m * n} bpp \tag{11}$$

$$MSE = \frac{\sum_{\substack{0 \le i < m \\ 0 \le j < n}} \left(I(i,j) - \widetilde{I}(i,j)\right)^2}{m * n} \tag{12}$$

$$PSNR = 10 \times log_{10} \frac{255^2}{MSE}\ (dB) \tag{13}$$

where I and $\widetilde{I}$ are the cover and the stego image, respectively. Again, when the length of the secret message is too short, only a portion of the image is accessed to conceal the message bit. In that case, Equations (11)–(13) will not provide the pure value, as all the $(m \times n)$ pixels do not participate in conceiving message bits. Hence, to find the pure capacity and the pure MSE, the value of m and n are replaced by x and y in Equations (11) and (12), where x and y are the total number of accessed rows and columns, respectively. In very usual case, y is n.

The processing complexity is another important parameter that bestows a comparison of processing speed in terms of time between the related algorithms. However, the measurement of execution times, t, of an algorithm by a processor may not depict the correct figure of time complexity, as t also depends on the uses and controlling mechanisms of instructions and loops in the programming code. Therefore, the same processor can spend more time executing a programming code of an algorithm over the other implementations if the algorithm is not written with optimum code and right placement of instructions. For this reason, the number of arithmetic and logical operators listed in the algorithms and their execution frequencies are compared as the performance of the schemes regarding the time complexity. The execution time for each arithmetic and logical operator (e.g., =, >>, <<, +, −, /, *, ↑, mod, ceil, floor) is assumed to be one clock cycle. The number of arithmetic and logical operators listed in the algorithms and their execution frequencies are tabulated in Table 4 and discussed later on.

To test the stated issues related to the embedding efficiency, a good number of experiments were done in MATLAB using BOSS image data set of 500 images and 200 collected standard images. These standard images were downloaded from several research sites. Among these experimented images, Nature, Mona Lisa, Lena, Boat, Mandrill, Camera Man, Peppers, Rice, Dorm Guest, Giant Hole, Cartoon Bee, and Ebola Virus are depicted in Figure 4. In all the experimented images, our proposal dominates Chao et al.'s DE scheme, Hong and Chen's adaptive pixel-pair matching (APPM) DE scheme, W. Hong et al.'s multiple base DE scheme, W. Hong et al.'s prediction error (PE), Kamal and Islam's Euclidean distance-based scheme (Euclidean), and Liao et al.'s LSB replacement-based scheme [9] in all the three performance parameters. In this article, only the results of DE, APPM, Euclidean, and LSB replacement schemes are compared with the proposed one.

(**a**) P1: Nature (**b**) P2: Mona Lisa (**c**) P3: Lena (**d**) P4: Boat

(**e**) P5: Mandrill (**f**) P6: Camera Man (**g**) P7: Peppers (**h**) P8: Rice

(**i**) P9: Dorm Guest (**j**) P10: Giant Hole (**k**) P11: Cartoon Bee (**l**) P12: Ebola Virus

Figure 4. Cover images used in the experiment.

3.1. Capacity Analysis

By encoding all the differences into two encoded errors, the scheme has obliged all the pixels (other than the reference values and first three pixels) to accept message bit. That strategy has notably boosted the embedding capacity. The comparisons of histograms of the proposed method with Euclidean are depicted in Figure 5. The figure clearly depicts that the values of all the encoded errors are either -1 or 1. The distributions of encoded errors into -1 and 1 allow the proposed embedding rules to implant 1 bit into each of the pixels.

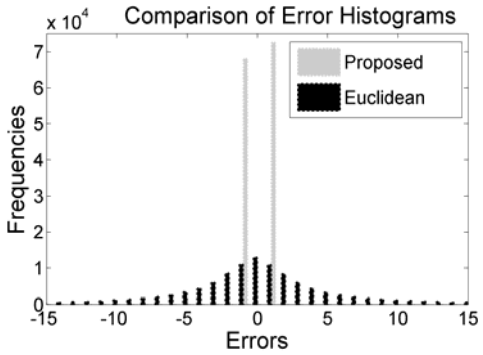


Figure 5. Comparison of histograms.

Summation of these two peaks of the proposed scheme is greater than the summation of the frequencies of all the bins in the errors histogram of Euclidean because the Euclidean scheme skips many blocks which are less likely for embedding. The skipping of blocks in both Hong et al.'s and Kamal and Islam's scheme are governed by a threshold parameter to protect image distortions. The effect of threshold is also depicted in Figure 6. The skipping of blocks decreases as the value of the threshold becomes higher; however, the larger threshold increases the probability of the image

distortions which, indeed, demolishes the prime objective of the schemes and, hence, the capacity cannot be improved.

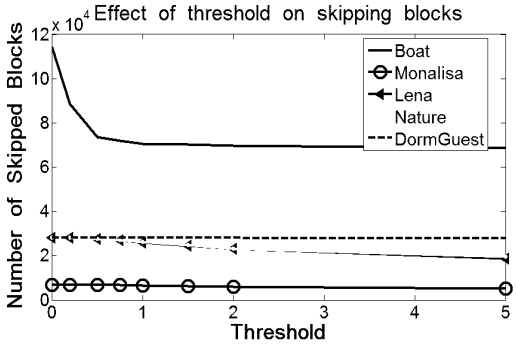


Figure 6. Effect of threshold in skipping blocks.

Again, DE schemes reduce the scope of embedding by making pixel groups and allowing the embedding procedure to embed only one bit per group. Although the LSB replacement-based scheme supports modification of multiple LSBs, in our experiment, we restricted that scheme to modify only a single LSB for the sake of comparison with our scheme. Besides, replacement of multiple bits will obviously destroy the image quality, noticeably. On the contrary, the proposed scheme allows all the pixels to conceive message bits. In the proposed scheme, the errors are shifted to the right by 1 when "1"s are embedded into the error 1, and to the left by 1 when "1"s are implanted into -1. That strategy has led our scheme to present dominating results in terms of capacity over the others by obliging all pixels equally to accept message bits. Figure 7 demonstrates some of the results. The embedding capacity of DE, APPM, Euclidean, LSB replacement, and proposed scheme are depicted as groups for each image. Our proposed scheme and the LSB replacement scheme achieve the capacity of 1 bit per pixel (bpp), while the others provide no more than 0.5 bpp. Therefore, to meet the higher embedding capacity, the proposed algorithm and the LSB replacement scheme would serve the objective. Nevertheless, the LSB replacement schemes are more vulnerable to statistical attacks, as demonstrated in Section 4.

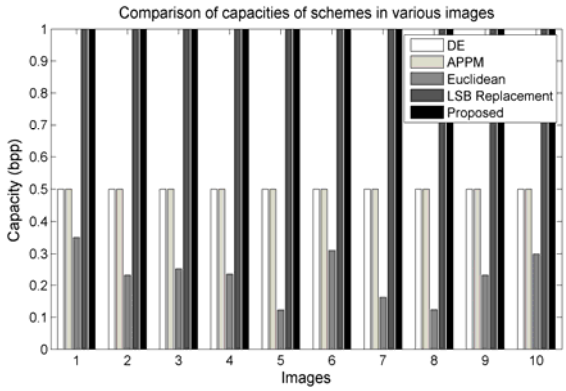


Figure 7. Comparison of embedding capacities among the schemes.

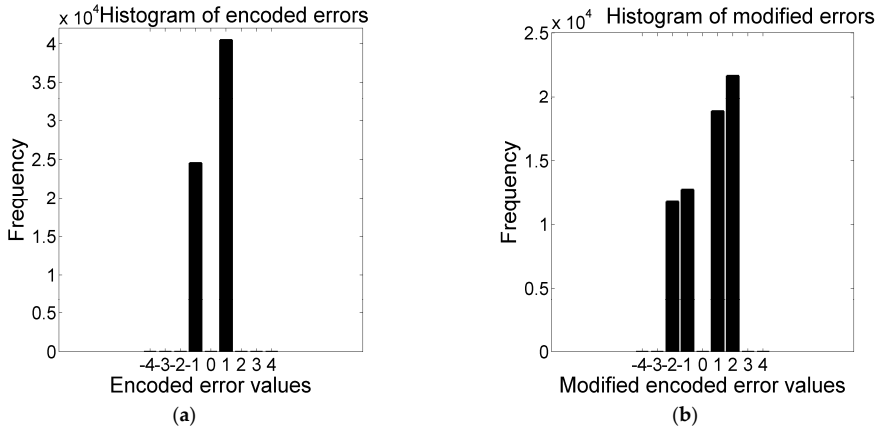
3.2. Visual Quality Analysis

According to the embedding rules, only the modified pixels can be changed by 1 because the encoded errors are shifted by 1 at most, as shown in the Figure 8. Now, say the probability of changing the value of a pixel by 1 is p and thus the probability of not changing is $(1 - p)$. The total number of pixels that changed by 1 after the data embedment is $(p \times m \times n)$. As all the pixels accept message bits

and these are changed by 1 at most, it can be said that $\sum_{\substack{0 \le i < m \\ 0 \le j < n}} \left(I(i,j) - \widetilde{I}(i,j)\right)^2 = m \times n \times p$.

Therefore, Equation (12) is rewritten by:

$$MSE = \frac{m \times n \times p}{m \times n} = p \tag{14}$$



(**a**) (**b**)

Figure 8. Histogram of (**a**) encoded error and (**b**) stego errors, which are generated by experimenting on the Lena image of size 255 × 255 pixels.

The cover pixels are equally probable for changing their values in the stego pixels. Hence, the theoretical value of MSE is 0.5. The adopted minimizing distortion strategy presents much smaller value of MSE than 0.5.

The experimental values of visual quality are measured to check the actual distortion rate. The experimental results in Table 1 demonstrate that although the embedding capacity of the proposed algorithm is more than twice of all the others regarding the distortion parameters, the tabulation of competing results, because the application of Equation (8), assists the encoder in generating better stego-image quality. However, this proposed scheme provides better image quality than the LSB replacement scheme, as shown in the Tables 1–3, only for the use of Equation (8). Our scheme provides better image quality than the diamond-encoding scheme for all images, and even better than the Euclidean-based scheme and APPM scheme for some images. Frequencies of dominations for PSNR values are mostly observed by the APPM scheme because it reduces shifting quantity by using multiple bases with its diamond-encoding procedures. However, the amount of degradation in the proposed scheme, if it happens, is very small and even negligible compared to the others, though the proposed scheme conceives message bits more than twice the quantity of the other methods.

Table 1. Gain of PSNR (peak signal-to-noise ratio) over the others.

Image	PSNR in					PSNR Gain Over			
	DE	APPM	PE	LSB	Proposed	DE	APPM	PE	LSB
P1	52.11	53.00	53.79	52.24	53.05	1.77	0.1	−1.39	1.53
P2	52.11	54.14	54.83	52.19	52.45	0.65	−3.21	−4.54	0.50
P3	52.11	54.17	52.68	52.25	52.99	1.66	−2.24	0.57	1.40
P4	52.11	54.16	53.94	52.35	53.00	1.68	−2.20	−1.78	1.23
P5	52.12	53.9	57.2	53.21	53.99	3.47	0.17	−5.94	0.14
P6	52.12	54.18	52.28	51.48	53.00	1.66	−2.23	1.36	2.87
P7	52.13	54.17	56.37	52.31	52.99	1.63	−2.22	−6.38	1.28
P8	52.13	54.15	55.72	52.14	52.46	0.63	−3.22	−6.21	0.61
P9	52.12	54.16	56.94	52.27	52.99	1.64	−2.21	−7.46	1.36
P10	52.12	54.17	52.59	52.23	52.99	1.64	−2.23	0.74	1.43

The number of bits that can be embedded into an image (i.e., payload) varies from scheme to scheme. Due to the variation in the payload, the MSE as well as the PSNR also differ from scheme to scheme. The matter of boosting up the payload to some multiples can be tolerated if the amount of the image degradation is not changed multiplicatively and can be appreciated if it is not noticeable and realizable by any steganalyzer. However, payload directly affects the MSE [26]. With the increment of payload, MSE will show additive results. Again, for a much smaller payload, distortion will be limited to a smaller region of the image (where embedding is done), whereas the MSE by Equation (12) will be computed for the entire image.

Therefore, to have an exact idea of image degradation per tempered pixel, the MSE is generalized in Equation (15).

$$GenMSE = \frac{MSE}{Payload} ImageSize \tag{15}$$

Here, the proposed algorithm shows incredible improvement. The gain figures yielded by our scheme in Table 2 demonstrate outstanding performance. Each row represents the results of individual image. The gains in Tables 1–3 are computed by the following relation (16).

$$Gain = \frac{X - Y}{X} 100\% \tag{16}$$

where X = MSE or PSNR of the proposed algorithm and Y = MSE or PSNR of the other schemes.

Table 2. Gain of generalized mean square error (GenMSE) over the others.

Generalized MSE in					Generalized MSE Gain Over			
DE	**APPM**	**PE**	**LSB**	**Proposed**	**DE**	**APPM**	**PE**	**LSB**
0.80	0.53	0.88	0.35	0.32	−148.37	−65.65	−172.07	−9.38
0.80	0.50	1.99	0.38	0.37	−116.36	−35.72	−437.74	−2.70
0.80	0.50	1.86	0.35	0.33	−144.97	−52.16	−470.42	−6.01
0.80	0.50	2.40	0.36	0.33	−145.47	−52.83	−635.26	−9.10
0.79	0.50	4.19	0.35	0.33	−144.46	−52.08	−1183.7	−6.06
0.79	0.50	1.78	0.35	0.33	−144.83	−52.33	−446.56	−6.06
0.79	0.50	3.36	0.35	0.33	−144.12	−52.58	−928.24	−6.06
0.79	0.50	5.61	0.39	0.37	−115.81	−35.64	−1420.2	−5.41
0.79	0.50	1.44	0.35	0.33	−144.42	−52.77	−339.39	−6.06
0.79	0.50	2.26	0.34	0.33	−144.31	−52.40	−589.91	−3.03

The improvements in PSNR gains (GenPSNR) are summarized in the Table 3. These are also very much alike. There, each row represents the result of an individual image.

Table 3. Gain of GenPSNR over the others.

Generalized PSNR Gain Over			
DE	**APPM**	**Euclidean**	**LSB**
7.447	4.132	8.194	0.73
6.39	2.529	13.928	0.22
7.343	3.44	14.271	0.48
7.359	3.476	16.349	0.71
7.326	3.436	20.919	0.48
7.338	3.449	13.919	0.48
7.314	3.463	19.099	0.48
6.368	2.524	22.53	0.44
7.325	3.473	12.131	0.48
7.322	3.454	15.831	0.24

Another important parameter is the size of message chunk l. This l is used in Equation (11) to reduce the amount of the image distortion. That length, l is also analyzed in this article. Experimental result are demonstrated in Figure 9. This shows that to minimize the distortions, smaller sizes of message blocks are preferred. However, too small a value for l will increase the size of β_t as well as the size of the assistant information because β_t is a part of the assistant information. Figure 9 delineates that the MSE is affected sharply for $l \in \{20–512\}$. For $l = 512$, MSE is about the maximum. Again, though, MSE is at the minimum for $l = 16$, and it increases the size of β_t tremendously, which is, in fact, 1/17th of the portion of the transmitted data and thus it will reduce the effective payload. Therefore, l should be a value so that it maintains the constraint $32 \leq l \leq 256$.

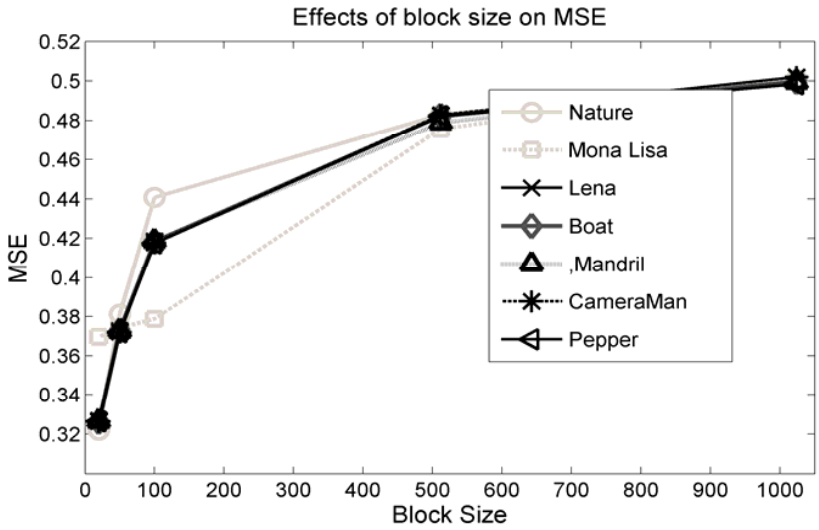


Figure 9. Effect of message block size on MSE (mean square error).

Finally, to observe more closely, only the face portion of the Lena image is clipped and zoomed in by 3. This is performed both for the cover and stego image. The resultant images are delineated in Figure 10. These two images are so close that it seems as though one is a clone of the other.

(**a**) (**b**)

Figure 10. A portion of (**a**) cover and (**b**) stego of the Lena image is zoomed in by 3. No difference is observed between these images.

3.3. Analysis on the Processing Complexity

The processing complexity is compared by the number of arithmetic and logical operators that are needed to be executed by that algorithm. The arithmetic and logical operators are symbolically defined

here by ∇_A and ∇_L, respectively. The comparison is summarized in Table 4, where scheme S_1, S_2, S_3, S_4, and S_5 stand for the scheme of Chao et al.'s DE, Hong and Chen's DE, Hong, Wien et al.'s DE, Kamal and Islam's Euclidean, and proposed encoding errors (EE), respectively. At the DE, the block's size is set to 2 pixels, while 9 pixels are used in the Euclidean. Hence, the comparison is presented for each processing of 18 pixels only.

Table 4. Comparisons regarding executed arithmetic and logical operators.

Scheme	Per Block Executed Operators in				Total per 18 Pixels	Slower Than Proposed, S_5 (in Times)
	Predictions *	Preprocessing *	Embedding	Extraction		
S_1	Nil	$25\nabla_A + 17\nabla_L$	$2\nabla_A + 18\nabla_L$	$2\nabla_A + 16\nabla_L$	$48\nabla_A + 612\nabla_L$	5.4
S_2	Nil	$113\nabla_A + 32\nabla_L$	$34\nabla_A + 32\nabla_L$	$16\nabla_L$	$2340\nabla_A + 1008\nabla_L$	16.3
S_3	Nil	$18\nabla_A + 16\nabla_L$	$13\nabla_A + 22\nabla_L$	$7\nabla_A + 17\nabla_L$	$504\nabla_A + 639\nabla_L$	5.6
S_4	$47\nabla_A + 16\nabla_L$	$8\nabla_A + \nabla_L$	$16\nabla_A + 20\nabla_L$	$24\nabla_A + 16\nabla_L$	$300\nabla_A + 140\nabla_L$	2.2
S_5	Nil	$17\nabla_A + 17\nabla_L$	$119\nabla_A + 18\nabla_L$	$34\nabla_A$	$170\nabla_A + 35\nabla_L$	0

* Required at either embedding or extraction phase. So, it is counted twice when computing the second last column.

From Table 4, it is clear that the proposed scheme executes less number of instructions to complete the task than the others. The results prove the superiority of the proposed scheme regarding the processing complexity. It is at least two times faster than the Euclidean scheme and some multiples (up to 16) of others.

4. Resistance against Attacks

Generally, image steganography ensures the secrecy of data transmission by hiding them in a cover media and it makes the embedded data visually insensible. Nevertheless, in a random fashion, one (person, device, or thread) can deploy statistical analysis [31–36] to realize the existence of the secret message within the cover image. To check the reliability of our proposed scheme against statistical attacks, two very commonly used analyses—histogram differences and chi-square test—are presented here.

4.1. Histogram Differences

In 2009, Zhao et al. [31] showed that, at the image of archetype, there is a similarity between the histograms of vertical differences and horizontal differences of the neighbor pixels. Hence, any major dissimilarity between these two histograms will signal the possibility of existing embedded information. Therefore, the histogram difference scheme was employed in this research to study its steadfastness against such attacks. First, pairwise differences of neighbor pixels were calculated along the vertical and horizontal directions. Histograms of vertical differences and horizontal differences are measured, say these are $\check{H}_v$ and $\check{H}_h$, respectively. A statistical difference D between $\check{H}_v$ and $\check{H}_h$ was computed by following Equation (17). In computing D, the scheme reckons only $2 \times T$ bins of the histograms, where T is a threshold.

$$D = \left(\sum_{i=-T}^{T} \left(\check{H}_h(i) - \check{H}_v(i) \right)^2 \right)^{1/2} \tag{17}$$

In our experiment, T was set to 20. Usually, a small value of D typifies a lower probability of the image being modified. The experimental results of the vertical and the horizontal difference histograms, obtained in the proposed scheme and in the LSB replacement-based scheme, are plotted in Figure 11. Both the figures show very similar behaviors. However, the LSB replacement-based scheme demonstrates a bit of unlikeness between the vertical and horizontal difference histogram in and around the "0" valued difference. In comparison with this point, the proposed scheme presents more similar histogram.

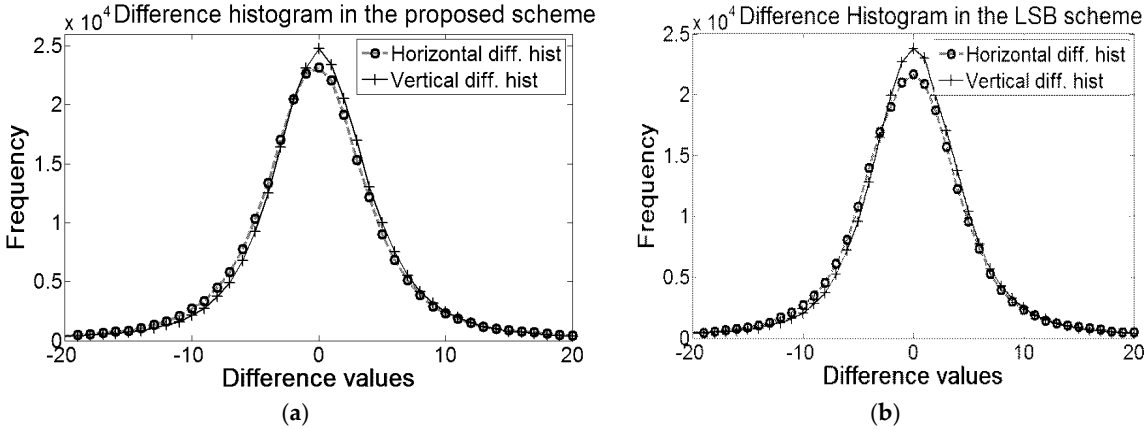


Figure 11. Vertical and horizontal difference histograms which are computed in the (**a**) proposed scheme and (**b**) LSB replacement scheme.

4.2. Chi-Square Test

Chi-square test is a very illustrious process to check the modification of pixel values [32] in the stego image with regard to the cover image. Hence, a chi-square test was conducted to verify the reliability of our proposed scheme against such statistical attacks. During the computation of the chi-square values, every pixel is considered an independent sample. Then, degree of freedom (DF) is measured by DF = (row_size $-$ 1)*(column_size $-$ 1). In the relation of measuring chi-square statistics, cover pixels are considered as expected values, and stego pixels are applied as observed values. The chi-square statistics of these pixels, whose frequencies in the histogram of the stego image are greater than 4, are measured by Equation (18).

$$X^2 = \sum_{i=1}^{row\_size} \sum_{j=1}^{col\_size} \frac{Observed_{i,j} - Expecteded_{i,j}}{Expecteded_{i,j}} \tag{18}$$

As the degree of freedom is always greater than 30, chi-square statistics will not be found from a chi-squared table. Therefore, the chi-squared distributions are approximated from the normal distribution by Equation (19).

$$X^2_{ND} = \sqrt{2X^2} - \sqrt{2DF - 1} \tag{19}$$

Again, a critical value, for the test of null hypothesis, is measured by MATLAB tools chi2inv, setting probability to 0.05. Let the measured critical value be $X^2_{\propto}$. Then, in all the experiments, it is found that $X^2_{ND} \ll X^2_{\propto}$. So, the null hypothesis, H_0, is accepted.

To realize the performance of the proposed scheme, the experimental results for seven images are demonstrated in Figure 12. The figure shows that our proposed scheme presents smaller chi statistics than the critical chi-square values. The proposed scheme also delineates that the chi-square statistics of the LSB replacement-based scheme are higher than the ones in the proposed scheme because chi-square statistics can detect the movement of pixel values between a pair of values. The LSB replacement-based scheme always modifies the values between a pair of pixels. Consequently, our proposed scheme shows more resistance against statistical attacks.

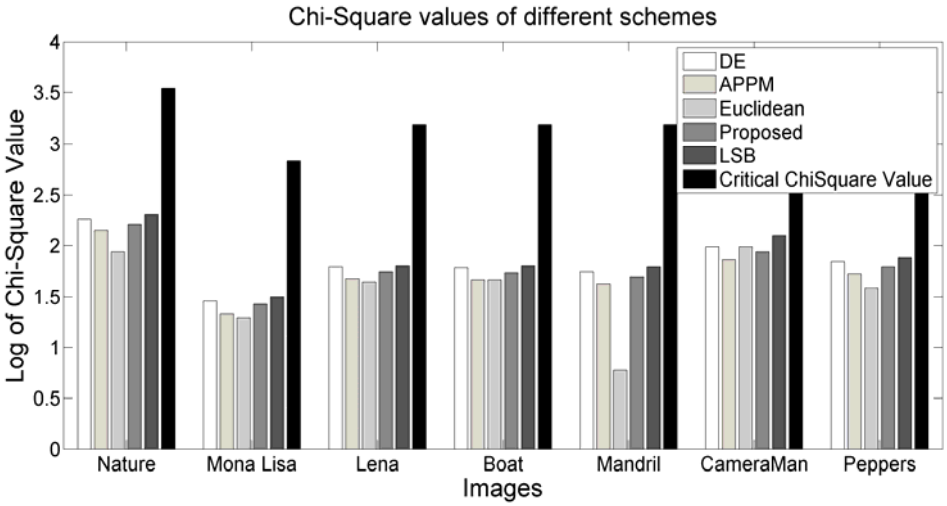


Figure 12. Comparison of chi-square values among the schemes in the logarithmic scale.

5. Conclusions

The proposed scheme embeds massive information related to intellectual properties, copyrights, forensics, and law-enforcing agencies data. It clearly reigns over all the competing methods regarding the embedding capacity, the MSE, and the time complexity. The capacity is more than twice of the other competing schemes; the generalized PSNR is improved by a gain of 2%–20%; and the computational time is reduced by a factor of 2–16. The uses of reference values have made the scheme more secure and robust. Therefore, we hope that the proposed scheme will appear in the field of hiding digital data as a notable contribution for its capabilities of massive data embedment and fast extraction of secret data.

The proposed scheme currently works in a spatial domain and will not operate in the transformed domain. In our future works, we will aim to acclimatize this concept for embedding data in the transform coefficients. We also hope to present a reversible methodology to embed 1 bpp, when 1 bpp is tried.

Acknowledgments: The author A.H.M.K. is funded by the ICT division of the Ministry of Post, Telecommunication and Information Technology of the Government of Bangladesh through a fellowship program. Therefore, the authors like to acknowledge the stated ministry of Bangladesh.

Author Contributions: The first author, A.H.M.K., is a PhD student of the department of Computer Science and Engineering of the Bangladesh University of Engineering and Technology. He is working under the supervision of second author, M.M.I. Hence, the whole work is supervised and guided by M.M.I. M.M.I. has been consulted on all the way to the progress of the research work by the author A.H.M.K. A.H.M.K. has completed the experiments and made the draft of the manuscript. M.M.I. has revised the manuscript and given the final approval to submit it to that journal.

Conflicts of Interest: The authors do not have any economical interest from that article. The first author is a PhD student and working under the supervision of the second author. To meet the requirement for achieving the PhD degree, the first author has to publish his research works on ranked journals which are published by well recognized publishers. Therefore, the authors have chosen this journal to publish that work. Both the authors are aware of that submission. The first author is a fellow of ICT division of the Ministry of Post, Telecommunication and Information Technology of the Government of Bangladesh. However, the fellowship neither covers any publication charges, nor claims any financial interest from that research.

References

1. Kamal, A.H.M. Steganography: Securing Message in wireless network. *Int. J. Comput. Technol.* **2013**, *4*, 797–801.
2. Chen, C.-C.; Tsai, Y.-H. Adaptive reversible image watermarking scheme. *J. Syst. Softw.* **2011**, *84*, 428–434. [CrossRef]

3. Kamal, A.H.M.; Mahfuzul Islam, M. Facilitating and securing offline e-medicine service through image steganography. *Healthc. Technol. Lett.* **2014**, *1*, 74–79. [CrossRef] [PubMed]
4. Böhme, R.; Kirchner, M. Counter-Forensics: Attacking Image Forensics. In *Digital Image Forensics*; Springer: New York, NY, USA, 2013; pp. 327–366.
5. Brindha, S.; Vennila, I. Hiding Fingerprint in Face using Scattered LSB Embedding Steganographic Technique for Smart card based Authentication System. *Int. J. Comput. Appl.* **2011**, *26*, 51–55. [CrossRef]
6. Hong, W.; Chen, T.-S.; Luo, C.-W. Data embedding using pixel value differencing and diamond encoding with multiple-base notational system. *J. Syst. Softw.* **2012**, *85*, 1166–1175. [CrossRef]
7. Lee, C.-F.; Chen, H.-L. A novel data hiding scheme based on modulus function. *J. Syst. Softw.* **2010**, *83*, 832–843. [CrossRef]
8. Yang, C.-H.; Weng, C.Y.; Wang, S.J.; Sun, H.M. Varied PVD + LSB evading detection programs to spatial domain in data embedding systems. *J. Syst. Softw.* **2010**, *83*, 1635–1643. [CrossRef]
9. Liao, X.; Wen, Q.; Zhang, J. A steganographic method for digital images with four-pixel differencing and modified LSB substitution. *J. Vis. Commun. Image Represent.* **2011**, *22*, 1–8. [CrossRef]
10. Gujjunoori, S.; Amberker, B.B. DCT based reversible data embedding for MPEG-4 video using HVS characteristics. *J. Inf. Secur. Appl.* **2013**, *18*, 157–166. [CrossRef]
11. Kamstra, L.; Heijmans, H.J.A.M. Reversible data embedding into images using wavelet techniques and sorting. *IEEE Trans. Image Process.* **2005**, *14*, 2082–2090. [CrossRef] [PubMed]
12. Chung, K.-L.; Huang, Y.-H.; Yan, W.-M.; Teng, W.-C. Distortion reduction for histogram modification-based reversible data hiding. *Appl. Math. Comput.* **2012**, *218*, 5819–5826. [CrossRef]
13. Hong, W.; Chen, T.-S. A local variance-controlled reversible data hiding method using prediction and histogram-shifting. *J. Syst. Softw.* **2010**, *83*, 2653–2663. [CrossRef]
14. Liu, C.-L.; Liao, S.-R. High-performance JPEG steganography using complementary embedding strategy. *Pattern Recognit.* **2008**, *41*, 2945–2955. [CrossRef]
15. Huang, F.; Luo, W.; Huang, J. Steganalysis of JPEG steganography with complementary embedding strategy. *IET Inform. Secur.* **2011**, *5*, 10–18. [CrossRef]
16. Lu, Z.-M.; Wang, J.-X.; Liu, B.-B. An improved lossless data hiding scheme based on image VQ-index residual value coding. *J. Syst. Softw.* **2009**, *82*, 1016–1024. [CrossRef]
17. Wang, W.-J.; Huang, C.-T.; Wang, S.-J. VQ applications in steganographic data hiding upon multimedia images. *Syst. J. IEEE* **2011**, *5*, 528–537. [CrossRef]
18. Tian, H.; Zhou, K.; Jiang, H.; Liu, J.; Huang, Y.; Feng, D. An M-sequence based steganography model for voice over IP. In Proceedings of the ICC'09 2009 IEEE International Conference on Communications, Dresden, Germany, 14–18 June 2009.
19. Chang, I.-C.; Hu, Y.-C.; Chen, W.-L.; Lo, C.-C. High Capacity reversible data hiding scheme based on residual histogram shifting for block truncation coding. *Signal Process.* **2015**, *108*, 376–388. [CrossRef]
20. Kamal, A.H.M.; Islam, M.M. Enhancing the Embedding Payload by Handling the Affair of Association and Mapping of Block Pixels through Prediction Errors Histogram. In Proceedings of the International Conference on Networks Systems and Security, Dhaka, Bangladesh, 5–7 January 2016.
21. Kamal, A.H.M.; Islam, M.M. Enhancing embedding capacity and stego image quality by employing multi predictors. *J. Inform. Secur. Appl.* **2016**, in press. [CrossRef]
22. Kamal, A.H.M.; Islam, M.M. Boosting up the data hiding rate multi cycle embedment process. *J. Vis. Commun. Image Represent.* **2016**, *40*, 574–588. [CrossRef]
23. Kamal, A.H.M.; Islam, M.M. Capacity Improvement of Reversible Data Hiding Scheme through Better Prediction and Double Cycle Embedding Process. In Proceedings of the IEEE International Conference on Advance Networks and Telecommunication Systems, Kolkata, India, 16–18 December 2015.
24. Habiba Sultana, A.H.M. Kamal and Mohammad Mahfuzul Islam, Enhancing the Robustness of Visual Degradation Based HAM Reversible Data Hiding. *J. Comput. Sci.* **2016**, *12*, 88–97. [CrossRef]
25. Fridrich, J.; Goljan, M.; Du, R. Lossless data embedding for all image formats. In *Electronic Imaging 2002*; International Society for Optics and Photonics: San Francisco, CA, USA, 2002; pp. 572–583.
26. Lin, C.-C. An information hiding scheme with minimal image distortion. *Comput. Stand. Interfaces* **2011**, *33*, 477–484. [CrossRef]
27. Islam, S.; Gupta, P. Effect of morphing on embedding capacity and embedding efficiency. *Neurocomputing* **2014**, *137*, 136–141. [CrossRef]

28. Hong, W.; Chen, T.-S. A novel data embedding method using adaptive pixel pair matching. *IEEE Trans. Inform. Forensics Secur.* **2012**, *7*, 176–184. [CrossRef]
29. Provos, N.; Honeyman, P. Hide and seek: An introduction to steganography. *Secur. Priv. IEEE* **2003**, *1*, 32–44. [CrossRef]
30. Chao, R.M.; Wu, H.C.; Lee, C.C.; Chu, Y.P. A novel image data hiding scheme with diamond encoding. *EURASIP J. Inform. Secur.* **2009**, *2009*, 658047. [CrossRef]
31. Zhao, H.; Wang, H.; Khan, M.K. Statistical analysis of several reversible data hiding algorithms. *Multimed. Tools Appl.* **2009**. [CrossRef]
32. Rocha, A.; Goldenstein, S. Progressive randomization: Seeing the unseen. *Comput. Vis. Image Underst.* **2010**, *114*, 349–362. [CrossRef]
33. Andriotis, P.; Oikonomou, G.; Tryfonas, T. JPEG steganography detection with Benford's Law. *Digit. Investig.* **2013**, *9*, 246–257. [CrossRef]
34. Arshadi, L.; Jahangir, A.H. Benford's law behavior of Internet traffic. *J. Netw. Comput. Appl.* **2014**, *40*, 194–205. [CrossRef]
35. Zaharis, A.; Martini, A.; Tryfonas, T.; Ilioudis, C.; Pangalos, G. Lightweight steganalysis based on image reconstruction and lead digit distribution analysis. *Int. J. Digit. Crime Forensics (IJDCF)* **2011**, *3*, 29–41. [CrossRef]
36. Fu, D.; Shi, Y.Q.; Su, W. A Generalized Benford's Law for Jpeg Coefficients and Its Applications in Image Forensics. In *Electronic Imaging 2007*; International Society for Optics and Photonics: San Francisco, CA, USA, 2007.