symmetry

MDPI

Article

Intra-Block Correlation Based Reversible Data Hiding in Encrypted Images Using Parametric Binary Tree Labeling

Arun Kumar Rai [1], Neeraj Kumar [2], Rajeev Kumar [3], Hari Om [1], Satish Chand [4] and Ki-Hyun Jung [5,*]

[1] Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad 826004, India; arun.18DP000380@cse.iitism.ac.in (A.K.R.); hari@iitism.ac.in (H.O.)
[2] Department of Electronics and Communication Engineering, Nagarjuna College of Engineering and Technology, Bengaluru 562164, India; neeraj.mohiwal@gmail.com
[3] Department of Computer Science and Engineering, Delhi Technological University, Delhi 110042, India; rajeevkumar@dtu.ac.in
[4] School of Computer and System Sciences, Jawaharlal Nehru University, Delhi 110067, India; schand@mail.jnu.ac.in
[5] Department of Cyber Security, Kyungil University, Kyeongbuk 38424, Korea
* Correspondence: kingjung@kiu.kr or khanny.jung@gmail.com; Tel.: +82-53-600-5626

Abstract: In this paper, a high capacity reversible data hiding technique using a parametric binary tree labeling scheme is proposed. The proposed parametric binary tree labeling scheme is used to label a plaintext image's pixels as two different categories, regular pixels and irregular pixels, through a symmetric or asymmetric process. Regular pixels are only utilized for secret payload embedding whereas irregular pixels are not utilized. The proposed technique efficiently exploits intra-block correlation, based on the prediction mean of the block by symmetry or asymmetry. Further, the proposed method utilizes blocks that are selected for their pixel correlation rather than exploiting all the blocks for secret payload embedding. In addition, the proposed scheme enhances the encryption performance by employing standard encryption techniques, unlike other block based reversible data hiding in encrypted images. Experimental results show that the proposed technique maximizes the embedding rate in comparison to state-of-the-art reversible data hiding in encrypted images, while preserving privacy of the original contents.

Keywords: image encryption; reversible data hiding; parametric binary tree labeling; privacy; intra-block correlation

Citation: Rai, A.K.; Kumar, N.; Kumar, R.; Om, H.; Chand, S.; Jung, K.-H. Intra-Block Correlation Based Reversible Data Hiding in Encrypted Images Using Parametric Binary Tree Labeling. *Symmetry* **2021**, *13*, 1072. https://doi.org/10.3390/sym13061072

Academic Editors: Kuo-Hui Yeh and Peng-Yeng Yin

Received: 20 April 2021
Accepted: 7 June 2021
Published: 16 June 2021

1. Introduction

Cloud computing is a great technology for remote sharing and accessing of information across the world. It renders a simplistic model for real-time information sharing, accessing, and/or storing on the cloud server [1] on a pay-per-use basis. Due to its simplicity, today, cloud computing services have millions of subscribers who store their public/private data on cloud servers. The cloud servers possess several data storage devices for storing subscribers' data which can be any multimedia data (such as image, video, audio), text data, program data, etc., in any format. To protect against information leakage and maintain confidentiality, subscriber data is first encrypted and then stored on the cloud servers [2]. To manage the cloud services and data storages, and also to control loading–unloading of data between storages for load balancing, the cloud servers makes use of a cloud administrator which attaches some auxiliary information, such as subscriber name, file name, file type, source information, etc., to subscriber's data for its efficient handling without decoding the subscriber data. The auxiliary information is attached secretly by embedding the same in the encrypted subscriber data so that unauthorized access can be prevented without incurring additional storage cost. Since the auxiliary information is cloud management information, it is embedded in a lossless and reversible manner in the subscriber data. For this, there exist several reversible data hiding (RDH) techniques

which have been developed in the past for lossless embedding and recovery of both secret data and subscriber data. Some of the popular RDH techniques are lossless compression based [3–6], difference expansion-based [7], histogram expansion, and prediction error expansion based [8–12] RDH techniques. However, these techniques are only suitable for plaintext data, but not for encrypted data.

In 2008, Puech et al. [13] unveiled an RDH technique for encrypted images (RDHEI). They first encrypted the original image using Advanced Encryption Standard (AES) and then embedded the secret data bits at random locations in each 4×4 by simple substitution. At the receiver end, local standard deviation analysis is performed to reinstate the original image and get back the hidden secret data. After Puech et al.'s technique [9], the RDHEI field has been explored by various researchers [14–36].

Based on the analysis of available RDHEI literature, the RDHEI techniques can be classified into three major categories, namely, vacating room after encryption (VRAE) [12,15,21,22], vacating room by encryption (VRBE) [32,34,35], and reserving room before encryption (RRBE) [23,29–31,33]. The VRAE techniques create vacancies for secret data hiding in encrypted data whereas techniques that fall into the VRBE category create vacancies for secret data hiding during encryption. As far as the RRBE category is concerned, first, a room is reserved in the unencrypted media which is then exploited for embedding the secret data after the media's encryption. Among these three approaches, the VRAE approach offers very limited embedding capacity (EC) because encrypted data provides limited room for secret data hiding as encryption destroys the correlations of subscriber data. Therefore, the cloud administrator is unable to produce many vacancies for secret data embedding in encrypted data. In the VRBE approach, specific encryption tactics are applied to subscriber data, which maintain some correlations in encrypted data. Later, the correlations present in the encrypted data are exploited by the cloud administrator for embedding auxiliary information. Still, this approach does not yield a high EC as spatial redundancy existing in the subscriber data is not fully utilized. However, the RRBE approach is an altogether different approach, in which subscriber data in unencrypted form is exploited to reserve vacancies. This means that, later on, the subscriber data can be encrypted and auxiliary information can be embedded in the encrypted data, by exploiting the redundancy of non-encrypted data due to the reserved vacancies. Thus, the RRBE approach renders higher EC than the VRAE and VRBE approaches.

In RDHEI, the encryption of plain-text data is performed by the subscriber using an encryption key K_e, who then uploads the encrypted data on the cloud server. Next, the cloud administrator performs embedding of auxiliary information using a secret key K_s to transform the encrypted data into marked encrypted data. The marked encrypted data accommodates both subscriber data and auxiliary information in encrypted form, and is stored on the cloud server. At the decoder side, three cases may exist. In the first case, if a user owns both the encryption key K_e and the secret key K_s, he/she can retrieve both subscriber data and auxiliary information. In case only the encryption key K_e or the secret key K_s is possessed by the user then only subscriber data or the auxiliary information can be recovered, respectively. A schematic diagram of the RDHEI approach is shown in Figure 1.

In the literature, some RDHEI techniques have been introduced [14–18] which allow retrieval of auxiliary information only after decryption of marked encrypted data. In other words, auxiliary information can be extracted using a secret key K_s only after performing the decryption using the encryption key K_e. TheseRDHEI techniques are known as non-separable techniques. In [19–35], another variant of RDHEI techniques, i.e., separable RDHEI techniques, was introduced to overcome the limitation of non-separability. These techniques allow separate restoration of subscriber data and extraction of auxiliary information from the marked encrypted information. In other words, an encryption key K_e keeper can perform the decryption process separately without being restricted by the auxiliary information extracting process, and similarly a secret key K_s keeper can perform the auxiliary information extracting process without being restricted by the decryption process.

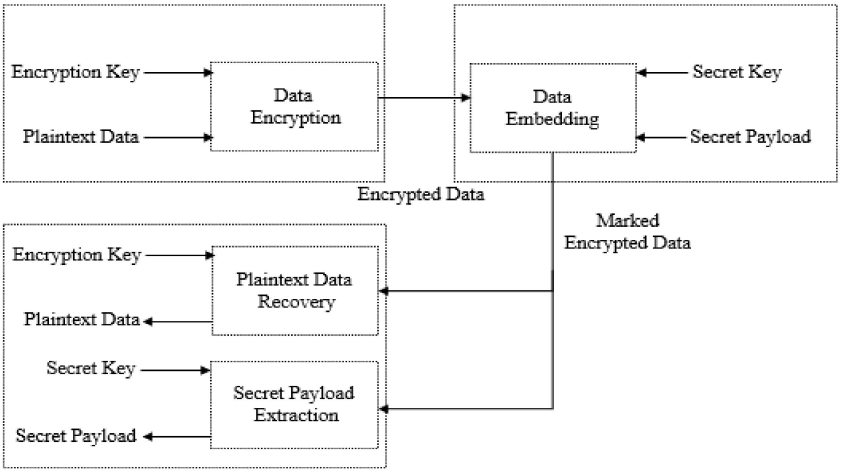

Figure 1. A schematic diagram of RDHEI technique.

In this paper, an intra-block correlation based high capacity RDHEI technique using PBTL is proposed. In the proposed technique, the original image is first partitioned into non-overlapping blocks of a predetermined size (e.g., 4×4) which are then categorized into smooth, moderately complex, and highly complex blocks. Now, prediction errors and residual errors are calculated for pixels of the smooth blocks and moderately complex blocks. Next, the proposed technique applies a stream encryption method (different from Yi et al.'s technique) to the original host image to improve the security of the encrypted subscriber data (or the encrypted host image). The pixels of the encrypted image are then labeled using the PBTL scheme and the calculated prediction and residual errors. Finally, the labelled pixels are exploited to reversibly embed the secret data. Contributions of the proposed RDHEI technique can be summarized as follows:

(1) The proposed work discloses a high capacity RDHEI technique using PBTL. The proposed RDHEI technique first partitions the original image into uniformly sized blocks and then categorizes the image blocks into three categories, i.e., smooth, moderately complex, and highly complex blocks based on the prevalent correlation of each block. This granular classification of the image blocks helps in reserving pixels for efficiently embedding the secret information.
(2) The smooth and moderately complex blocks generally possess high and decent pixel correlation, respectively. Thus, both types of blocks are used for embedding the secret information whereas the highly complex blocks are snubbed in the data embedding process as there is no or minimal correlation.
(3) Further, the proposed technique uses a stream encryption method for encrypting the original image. The stream cipher provides higher security in comparison to block cipher as it completely destroys the correlation of image pixels.
(4) In addition, the proposed technique is a separable RDHEI technique which allows separate recovery and extraction of subscriber data and the secret message in a lossless manner.
(5) Experimental results show that the proposed RDHEI technique has superior embedding performance in comparison to related previous RDHEI techniques while ensuring security of the image contents.

The rest of this paper is structured as follows. Section 2 discusses related separable RDHEI techniques. Section 3 introduces the review of parametric binary tree labeling (PBTL) scheme. The proposed RDHEI technique is discussed in Section 4. Section 5 shows the experimental results and their comparative analysis. Lastly, a conclusion as well as the scope of future works has been drawn for the proposed work.

2. Related Works

In this section, some of the popular RDHEI techniques are briefly reviewed. In 2012, Zhang discussed a separable RDHEI technique [19]. In the separable RDHEI technique [19], the subscriber data (which is an image) is first encrypted using a standard encryption method and then a room/space is reserved for embedding the secret/auxiliary information by compressing three least significant bits (LSBs) of the encrypted data. However, the generated space is sparse in nature which limits the embedding rate. Xiaotian et al. [17] discussed a separable RDHEI technique which outperforms the earlier technique [19]. Dragoi et al. [21] proposed a new separable RDHEI scheme by creating space after the encryption. The main feature of this scheme is the use of a two-stage data hiding process in which the reference pixel is predicted based on the median context value. As far as performance improvement is concerned, the proposed scheme marginally improves embedding capacity in comparison to erstwhile related schemes. In 2018, Dragoi et al. [22] again came up with a new separable RDHEI scheme with color images, where correlation between RGB color planes of a target pixels and its correlation with neighboring pixels are exploited to embed data in a color image. In 2013, Ma et al. unfolded the first RRBE approach based RDHEI technique [23]. The technique partitions the image to get smooth and complex regions. Next, one or more LSBs of complex regions are embedded into the smooth regions using any of the conventional RDH algorithms such as [5–7] to create room in the rough regions. Finally, the reserved room in the encrypted images are filled by auxiliary information. This reservation of space in the complex regions yields a high embedding rate which goes up to 0.5 bit-per-pixel (bpp). Similarly, Zhang et al. reserved room for secret data hiding using the prediction error (PE) based histogram shifting method [18]. In [24], Xu et al. disclosed the calculation of prediction error based on interpolation technique then applied histogram shifting and the difference expansion technique to exploit the prediction error for data hiding. However, only a small improvement in performance was achieved. In [25], Mathew et al. refine the work of Ma et al. [23] by introducing a new pixel intensity variation criterion for classifying image blocks into smooth blocks and rough blocks. However, the work only marginally improves the embedding rate. To further upgrade the embedding rate, Cao et al. [26] discuss a patch-level sparse representation method for RDHEI. The method performs encryption in phases so that the maximum correlation between the image pixels can be maintained in the encrypted image. Additionally, a room is created inside the encrypted image for embedding the secret data. Therefore, the technique archives higher embedding capacity than the erstwhile state-of the art RDHEI techniques. In 2018, Li et al. [27] disclosed a novel RDHEI technique which makes use of combined block permutation and stream cipher for image encryption. Next, prediction errors in nearby pixels are exploited for data hiding. Thus, Li et al.'s technique improves the embedding rate by 0.5 bpp in comparison to the existing related RDHEI techniques, which signifies high embedding capacity.

In literature, it has been observed that a number of techniques [28–31] have been introduced for MSB prediction and then exploiting the MSB for data hiding. In 2018, one such work is proposed by Puteaux et al. [29] in which a simple but powerful high capacity RDHEI technique was discussed. The technique utilizes the correlation with neighboring pixel to predict the reference pixel value and uses a location binary map for marking the prediction errors. Next, the image is encrypted using stream cipher, and embedding of the secret message is done in the MSBs of the encrypted pixels with the help of location binary map. The location binary map helps the receiver in extracting the secret message and in the complete recovery of the original image. However, the MSB prediction technique can only embed up to one bit of the secret message into an encrypted pixel. To overcome this limitation, Puyang et al. [30] discuss a new RDHEI technique which can replace up to two MSBs of an encrypted pixel to significantly improve the embedding capacity without affecting the reversibility. To further improve the EC, Chen et al. [31] makes use of run length encoding (RLE) to compress the binary sequence of MSB data so that a room can be reserved in the image. For optimal compression, the image is first divided into blocks and

then the binary sequence of MSB data is created. Thus, the technique further boosts the embedding rate in comparison to [29,30] without compromising the reversibility.

Yi et al. [32] propounded a headway technique in the separable RDHEI domain using parametric binary tree labeling (PBTL). The host image is first partitioned into blocks of a certain size (either 2×2 or 3×3 pixels) and then calculates prediction errors in neighboring pixels within the block. Next, the host image is encrypted using a block-based encryption method. Based on the prediction errors and PBTL scheme, pixels of the encrypted are labeled which are exploited for embedding the secret information later on. The PBTL-RDHEI is extended by Su et al. [33] by combining the PBTL and absolute moment block truncation coding (AMBTC) for efficiently exploiting the correlation of the host image. Yin et al. [26] used the AMBTC technique for data embedding, although not in the RDHEI domain. Further, it has been observed that the AMBTC technique is a popular lossy compression technique which has been widely used in data hiding [14,36–38]. As far as working method of [33] is concerned, Su et al.'s technique first scrambles and then compresses the host image into triplets in a block-wise manner, where each triplet includes two quantization level (high & low) and a bitmap. Next, the triplets are encrypted in such a way that the correlation between the two quantization levels is retained. The retained correlation is then exploited to create a room for embedding the secret information. Therefore, a compressed marked-encrypted image in the form of AMBTC codes is obtained by the receiver. Since the bits required to represent the AMBTC coded image are lesser, the embedding capacity is also lower than that of Yi et al. [32]. It can be easily stated that Su et al.'s case is one of the earliest methods to utilize AMBTC in the RDHEI domain. In the next section, the parametric binary tree labeling scheme is reviewed in the context of the proposed work.

3. Parametric Binary Tree Labeling Scheme

This Parametric binary tree labelling (PBTL) scheme is first postulated by Yi et al. in 2018 for labelling the pixels of the plaintext image so that the auxiliary information can be efficiently embedded. The plaintext image pixels are represented by 8-bit depth; hence, the seven layers are generated in the parametric binary tree as shown in Figure 2. In the PBTL, each layer has 2^n nodes and each node is represented by n binary bits where $n \in \{1, 2, \ldots, 7\}$ is the layer number. The details regarding the total number of nodes in each layer of the seven-layered PTBL is provided in Table 1.

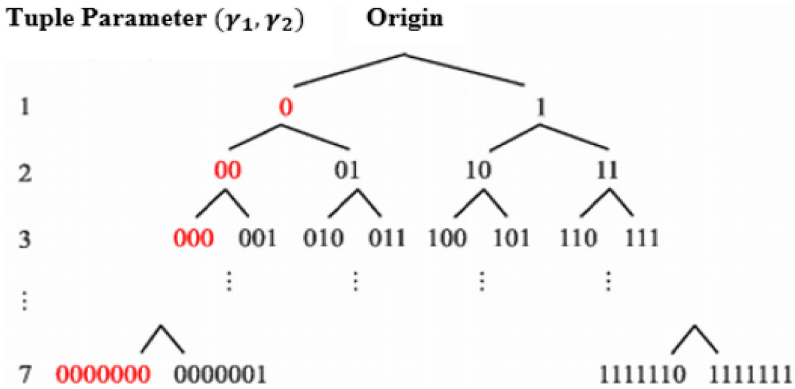

Figure 2. A 7-layered PBTL structure for 8-bit depth plaintext image.

Table 1. Number of nodes in a 7-layered PBTL.

Layer	1	2	3	4	5	6	7
Number of Nodes	2	4	8	16	32	64	128

The nodes at each layer are distributed into two different sets, namely S_1 and S_2, based on a tuple parameter (γ_1, γ_2). The first node of eachlayer is assigned into set S_1 and is labeled by n number of zero bits based on the value of γ_1, where n is the layer number. The number of nodes assigned to set S_2 are determined in accordance to the relation between the tuple parameter $\gamma_1 \wedge \gamma_2$ using Equation (1), defined as follows:

$$N_{\gamma_2} = \begin{cases} 2^{\gamma_2} - 1, \ \gamma_2 \leq \gamma_1 \\ (2^{\gamma_1} - 1) * (2^{\gamma_2 - \gamma_1}), \ \gamma_2 \leq \gamma_1 \end{cases} \quad (1)$$

where N_{γ_2} represents the total number of nodes in the set S_2 and the range of values of the tuple parameter is defined as $1 < \gamma_1, \gamma_2 < 7$. The specific details regarding the number of nodes in set S_2 for each layer of the PBTL based on the different values of $\gamma_1 \wedge \gamma_2$ are provided in Table 2. As per Table 2, if the value of tuple parameter $(\gamma_1, \gamma_2) is (2, 3)$, then number of nodes in set S_2 are 6 as per Equation (1), which are labeled as (111_2), (110_2), (101_2), (100_2), (011_2), and (010_2). Basically, the node labelling in the set S_2 is started from the right-most side to the left side of the PBTL (in accordance with Figure 2) until the N_{γ_2} is not labelled. Further, the first node in set S_1 will be labeled as (000_2).

Table 2. N_{γ_2}-based on $\gamma_1 \wedge \gamma_2$ for 7-layered PBTL.

γ_1	**Number of Nodes (N_{γ_2}) in S_2**						
	γ_2						
	1	**2**	**3**	**4**	**5**	**6**	**7**
1	1	2	4	8	16	32	64
2	1	3	6	12	24	48	96
3	1	3	7	14	28	56	112
4	1	3	7	15	30	60	120
5	1	3	7	15	31	62	122
6	1	3	7	15	31	63	126
7	1	3	7	15	31	63	127

4. The Proposed EPBTL-RDHEI Technique

The proposed High Capacity RDHEI technique is described in two phases, where the subscriber's data encryption and secret payload embedding is done in the first phase, and subscriber's data decryption and extraction of the secret payload is done in the second phase. Usually, subscribers upload their data which can be image, text, video, audio, etc., in encrypted form on the cloud server and a cloud administrator associates secret payload (which also includes auxiliary information) with the encrypted data. Although the proposed technique is equally applicable to all types of data, for simplicity, the plaintext images are considered as subscriber data. The proposed technique is a separable RDHEI technique which allows separate extraction of the hidden message and recovery of the original image at the receiving end. The RDHEI technique first uses an AMBTC based method for classifying the image regions and then uses a PBTL scheme for labelling the pixels. Figure 3 illustrates a framework of the proposed RDHEI technique.

The first phase of the proposed technique performed in five steps. In the first step, the subscriber's image is uniformly partitioned into non-overlapping blocks of a predetermined size and then each block is categorized into as smooth, moderately complex, and highly complex block based on the difference between high and low quantization levels which are calculated using the AMBTC method. In the second step, prediction errors and residual errors are generated based on the difference between pixel and corresponding block's mean. In the third step, the subscriber's image is encrypted using a stream encryption technique. In the fourth step, pixels of the encrypted image are categorized into regular pixels, irregular pixels, base pixels, and special pixels. Then, regular pixels and irregular

pixels are labeled according to the PBTL scheme and thereby labeled encrypted image is obtained. In the final (i.e., fifth) step, the secret payload embedding is done to get the marked encrypted image. The detailed working of each step is discussed in following subsections.

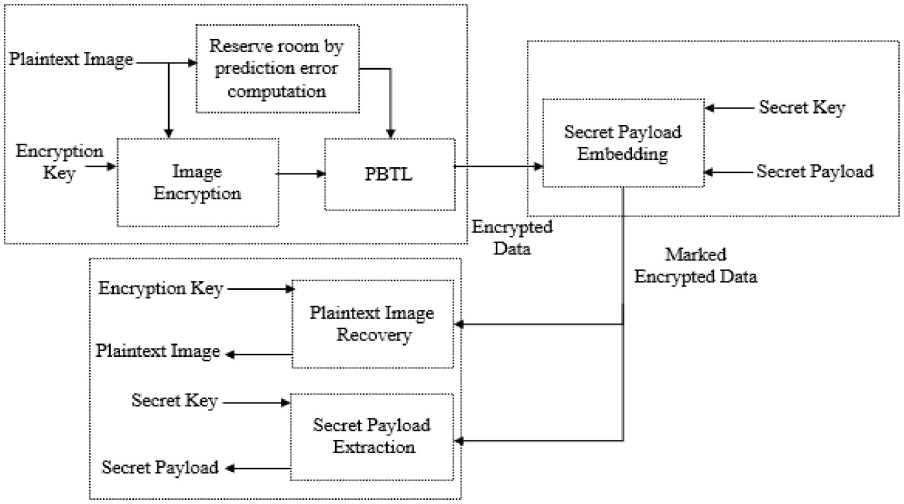

Figure 3. A framework of the proposed EPBTL-RDHEI technique.

4.1. Block Categorization (Step-1)

Initially, the subscriber-provided plaintext image $\mathbb{I}$ of size $N_1 \times N_2$ is partitioned into a number of non-overlapping blocks (P) of size $n_1 \times n_2$ pixels where $P = [N_1/n_1].[N_2/n_2]$. Therefore, each block (B_k) has $n_1 \times n_2$ pixels, such that, $\{x_1, x_2 \ldots . x_{n_1 \times n_2}\}$ if scanned in raster scan manner, where $k \in (1, 2 \ldots P)$. Now, mean (μ) of each block is computed using Equation (2):

$$\mu = \frac{1}{n_1 * n_2} \sum_{i=1}^{n1*n2} (x_i) \tag{2}$$

After computing mean (μ) of each block B_k, a bit-plane (b) of size $n_1 \times n_2$ is formed, where every pixel is represented by either bit '0' or '1' as follows.

$$b_i = \begin{cases} 0 \ \ if \ x_i < \mu, \\ 1 \qquad else. \end{cases} \tag{3}$$

As per the Equation (3), the pixel x_i of block B_k is represented by '0' if its value is less than mean (μ) of the block, otherwise represented by '1'. Thus, the bit-plane (b) for B_k has $n_1 \times n_2$ bits, meaning every pixel of the block is represented by 1 bit in the bit-plane. Next, two quantization levels, i.e., high quantization level q_0 and low quantization level q_1, are computed using Equations (4) and (5), respectively.

$$q_0 = \frac{1}{b_0} \sum_{x_i \geq \mu} x_i \tag{4}$$

$$q_1 = \frac{1}{b_1} \sum_{x_i < \mu} x_i \tag{5}$$

where b_0 and b_1 represent number of zeros and number of ones, respectively, in b. Thus, the high quantization level (q_0) is calculated by taking the mean of the pixels which have a value greater than the mean of the block. Similarly, the low quantization level (q_1) is computed by taking the mean of the pixels which have a value lower than the mean of the block. Thus, for each image block, a triplet $\{q_0, q_1, b\}$ is obtained.

Next, the image blocks are categorized into smooth, moderately complex, and highly complex block categories based on the difference between q_0 and q_1. More specifically, if the difference between q_0 and q_1 of block B is less than the first user-defined threshold T_s, then the block B is characterized as a smooth block type; if the difference between q_0 and q_1 is greater than or equal to the threshold T_s and less than a second user-defined threshold T_c, then the block B is characterized as moderately complex; and, otherwise, the block B is characterized as a highly complex block. In this step, the number of smooth blocks are counted as P_s, the number of moderately complex blocks are counted as P_m, and the number of high complex blocks are counted as P_h. A smooth block indicates that pixels in the block are high correlated and less distributed. A moderately complex block indicates that pixels in the block are less correlated and moderately distributed. A highly complex block indicates that pixels in the block are uncorrelated (or very little correlated) and highly distributed.

4.2. Computation of Prediction and Residual Errors (Step-2)

Once all the blocks of the plaintext image $\mathbb{I}$ are categorized, then prediction errors and residual errors are computed to reserve room for secret payload embedding as follows:

- If the block (B_k), where $k \in (1, 2 \ldots P)$ is a smooth block, then it indicates that pixels in the block B_k are high correlated and less distributed. Thus, each pixel in smooth block B_k can be best predicted by its mean value μ_k. Prediction error e_i is determined using the difference between the original pixel x_i and the mean value μ_k of the block (B_k) as per Equation (6). It is to be noted that the original pixel value (x_i) can be recovered by adding the μ_k to the corresponding prediction error (e_i) as per Equation (7), at the decoding side.

$$e_i = x_i - \mu_k \tag{6}$$

$$x_i = e_i + \mu_k \tag{7}$$

- If the block (B_k) is a moderately complex block, then it indicates that pixels in B_k are less correlated. In case of a moderately complex block, prediction errors (e_i) and residual errors (r_i) need to be calculated using Equations (8) and (9), respectively. It is to be noted that the original pixel value (x_i) can be recovered using Equation (10), at the decoder side.

$$e_i = \lfloor x_i/2 \rfloor - \lfloor \mu_k/2 \rfloor \tag{8}$$

$$r_i = mod(x_i/2) \tag{9}$$

$$x_i = e_i + r_i + 2 * \mu_k \tag{10}$$

- If the block (B_k) is a highly complex block, then it indicates that that pixels in B_k are uncorrelated (or very less correlated) and highly distributed. Thus, it requires a greater number of bits to represent the errors, and it is suggested that highly complex blocks are not used for secret payload embedding.

4.3. Image Encryption (Step-3)

After calculating the prediction errors and mean values for smooth blocks, and predication errors, residual errors, and mean values for moderately complex blocks, a stream encryption process is performed on the plaintext image $\mathbb{I}$. For this, a pseudo-random matrix $\mathbb{Z}$ of size $N_1 \times N_2$ is generated using an encryption key K_e. Each pixel $x_{i,j}$ of the image $\mathbb{I}$ is converted into an 8-bit binary sequence as $(b_8 b_7 b_6 b_5 b_4 b_3 b_2 b_1)$, where $b \in (0, 1)$ using Equation (11). Similarly, each element $z_{i,j}$ of pseudo-random matrix $\mathbb{R}$ is converted into 8-bit binary sequence using Equation (11).

$$x_{i,j}^k = \left\lfloor x_{i,j}/2^{k-1} \right\rfloor mod(2), k = (1, 2 \ldots 8) \tag{11}$$

Then, bitwise exclusive-or-operation is performed between plaintext image $\mathbb{I}$ and pseudo-random matrix $\mathbb{Z}$ using Equation (12) to encrypt the image.

$$xe_{i,j}^{k} = x_{i,j}^{k} \oplus z_{i,j}^{k} \tag{12}$$

Therefore, an encrypted 8-bit binary sequence for each pixel of the encrypted image is obtained. The binary sequence is converted into decimal form using Equation (13), to obtain an encrypted pixel which in turn helps in getting the encrypted image I^e.

$$xe_{i,j} = \sum_{k=1}^{8} xe_{i,j}^{k} \cdot 2^{k-1} \tag{13}$$

4.4. Pixel Grouping and Labelling Using PBTL (Step-4)

In this step, pixels of the encrypted image $\mathbb{I}^e$ are grouped and then labeled using PBTL structure based on tuple parameter (γ_1, γ_2) as described in Section 3. Firstly, the encrypted pixels of $\mathbb{I}^e$ are grouped into four sets which are special set (x^s), a base set (x^b), a regular set (x^r) and an irregular set (x^i). The special set (x^s) contains special pixels, the base set (x^b) contains base pixels, a regular set (x^r) contains regular pixels, and an irregular set (x^i) contains irregular pixels. Special set (x^s) includes the last pixel xe_{N_1,N_2} and the second last pixel $xe_{N_1-1,\ N_2-1}$ of the encrypted image $\mathbb{I}^e$. However, other locations can also be used in special set (x^s). One of the special pixels is used to indicate block size and other one indicates a tuple parameter (γ_1, γ_2). A base set (x^b) includes all the first encrypted pixel $xe_{1,1}$ of each block, where MSB bit-planes of the base pixels (the first encrypted pixel $xe_{1,1}$) are used to indicate block type that can be smooth, moderately complex, and highly complex. Table 3 illustrates that how the 7th & 8th MSB bit-planes of the encrypted pixel $xe_{1,1}$ is being used to identify block type.

Table 3. Block type identification based on 7th–8th bit-plane of $xe_{1,1}$ of each block B_k.

Bit-Plane of $xe_{1,1}$	**8th Bit-Plane**	**7th & 8th Bit-Planes**	**7th & 8th Bit-Planes**
Bit-Value	0	01	11
Block Type	Smooth block	Moderate complex block	High complex block

Pixels in the regular set (x^r) and irregular set (x^i) are determined in accordance to prediction errors $(e_{i,j})$ calculated using Equation (5) for smooth block and using Equation (7) for moderately complex block. If prediction errors $(e_{i,j})$ of block (B_k) of $\mathbb{I}$ meets the condition mentioned in Equation (13), then the pixel $(xe_{i,j})$ of the encrypted image $\mathbb{I}^e$ is grouped into the regular set (x^r), otherwise, it is grouped into the irregular set (x^i). In this step, the number of regular pixels and irregular pixels in smooth blocks are counted as x_s^r and x_s^i, respectively, and the number of regular pixels and irregular pixels in moderately complex blocks are counted as x_m^r and x_m^i, respectively.

$$\lceil -N_{\gamma_2}/2 \rceil \leq P_e \leq \lceil (N_{\gamma_2} - 1)/2 \rceil \tag{14}$$

After grouping the pixels into four sets, pixels of sets x^r and x^i are labeled using the PBTL scheme. Note that the pixels of highly complex blocks are not labeled as they do not participate in the data embedding process. In x^i, the γ_1-LSB bits of all the pixels are labeled as $(0...0_2)$ as per the PBTL scheme. In x^r, γ_2-LSB bits of all the pixels are labeled using different binary sequences represented by N_{γ_2} sub-categories as per the PBTL scheme. Thus processed, the labeled encrypted image I^l is obtained.

4.5. Secret Payload Embedding (Step-5)

After pixel grouping and pixel labelling process, secret payload embedding process is carried-out in the labeled encrypted image $\mathbb{I}^l$. This step outputs a marked encrypted image $\mathbb{I}^m$ by replacing the (8- γ_2) bits of pixels of x^r by secret payload. The secret payload (S_b) contains two type of data; one is encrypted auxiliary information (A_b) which is provided by cloud administrator and other is overhead (O_b) which is obtained during transforming encrypted image $\mathbb{I}^e$ into labeled encrypted image $\mathbb{I}^l$. To protect the auxiliary information (A_b) from unauthorized access, it is encrypted using a secret key k_s. The overhead (O_b) is required for lossless recovery of original image. The length of the overhead (O_b) is calculated as below:

(a) 16 bits to represent original pixel values of $xe_{N_1-1,\ N_2-1}$ and $xe_{N_1,\ N_2}$.
(b) Total bits (*equal to* $P_s + 2 * P_m + 2 * P_h$) of base set to store original bit values which are used in block types indication.
(c) Total bits (equal to $\gamma_2 * x_s^i + \gamma_2 * x_m^i$) of irregular set to store original replaced bit-planes.
(d) Total bits (equal to $8 * P_s$) to store mean value μ_k of each smooth block.
(e) Total bits (equal to $7 * P_m$) to store mean value $\mu_k/2$ of each complex block.

Now, auxiliary information (A_b) is obtained by reducing overhead (O_b) from the secret payload (S_b) using Equations (15)–(17), and effective embedding rate $ER_{\gamma_1,\ \gamma_2}$ is calculated using Equation (18).

$$A_b = S_b - O_b \tag{15}$$

$$S_b = (8 - \gamma_2) * x^{rs} + (7 - \gamma_2) * x^{rm} \tag{16}$$

$$O_b = (16 + \gamma_2 * x^i + 8 * P_s + 7 * P_m + P + P_m + P_h) \tag{17}$$

$$ER_{\gamma_1,\gamma_2} = \frac{A_b}{N_1 * N_2} = \frac{S_b - O_b}{N_1 * N_2} \tag{18}$$

The second phase of decryption of the original image and data recovery from the marked encrypted image has two steps. The first step is related to extraction of auxiliary information from the marked encrypted and second step is to restore original image from marked encrypted image. Both these steps are mutually exclusive.

4.6. Extraction of Auxiliary Information

At the receiver end, once the marked encrypted image $\mathbb{I}^m$ is received, the extraction process for auxiliary information is started. Initially, the pixels of x^s at locations xe_{N_1-1,N_2-1} and xe_{N_1,N_2} in $\mathbb{I}^m$ are utilized to determine tuple parameter $(\gamma_1,\ \gamma_2)$ and image block size. Then, $\mathbb{I}^m$ is divided into number of non-overlapping blocks as per the determined block size. After this, pixels of x^b in each image block are utilized to determine the type of the image block i.e. the smooth block, moderately complex block and highly complex block as per Table 3. Now, the tuple parameter $(\gamma_1,\ \gamma_2)$ is utilized to determine pixels x_s^r of regular set and pixels x_s^i of irregular set corresponding to smooth blocks, and pixels x_m^r of regular set and pixels x_m^i of irregular set corresponding to moderately complex blocks. Then, secret payload is extracted from (8- γ_2) bit-planes of regular pixels x_s^r of smooth blocks, and also from (8- γ_2) bit-planes of regular pixels x_m^r of moderately complex blocks. Further, residual errors are also extracted from regular pixels x_m^r of moderately complex blocks. Now, from the secret payload, encrypted auxiliary information is obtained by removing overhead as per Equations (15)–(17). Further, encrypted auxiliary information is decrypted by only authorized user which has secret key k_s.

4.7. Recovery of Plaintext Image

Now, recovery process for encrypted auxiliary information is discussed. The overhead information is utilized to recover plaintext image $\mathbb{I}$. Initially, 16 bits of overhead corresponding to $xe_{N_1-1,\ N_2-1}$ and $xe_{N_1,\ N_2}$ are restored on their predetermined position using x^s. Next, original bits (*equal to* $P_s + 2 * P_m + 2 * P_h$) are restored at locations of pixels of base set. Then, subsequent overhead bits equal to length $\gamma_2 * (x_s^i + x_m^i)$ are utilized to restore γ_2-bits of irregular pixels. After this, next overhead bits up to $8 * P_m$ are read to determine mean values of smooth blocks and subsequent overhead bits up to $7 * P_s$ are read to determine mean values of moderately complex blocks. Now, the authorized user which has encryption key K_e can generate the pseudo-random matrix $\mathbb{Z}$ and can obtain the decrypted image using Equations (12) and (13). Now, the user has only original pixel values of first two rows and two columns, which are utilized to recover other pixels of the image. The original pixel values of regular pixels of smooth blocks are obtained by their corresponding mean values stored as $8 * P_s$ overhead bits and prediction errors P_e calculated using Equation (7) from PBTL labeled pixels corresponding to $\gamma_2 - bits$. Similarly, to get original values of regular pixels of moderately complex blocks, mean values stored in next $7 * P_m$ overhead bits and residual errors P_r and prediction errors P_e calculated using Equation (10) from PBTL labeled pixels corresponding to $\gamma_2 - bits$. Now, remaining overhead bits are related to irregular pixels which are restored at their original location based on PBTL labeled pixels corresponding to $\gamma_1 - bits$. Thus, original image is retrieved in lossless manner.

Figure 4 shows an illustrative example of proposed RDHEI technique. The tuple parameter is taken as $(\gamma_1,\ \gamma_2) = (2, 3)$ and block size is taken as 4×4. Figure 4a shows a part of the baboon image of size 12×4. The input image is partitioned into non-overlapping blocks of size 4×4. Now, the blocks are categorized into smooth blocks, moderately complex blocks, and highly complex blocks using their corresponding quantization levels and mean values calculated by Equations (3)–(5), considering first threshold parameter $T_s = 16$ and second threshold parameter $T_c = 32$. The absolute difference between quantization levels $q_0 - q_1$ of the first block is 3, so it is a smooth block. The absolute difference between quantization levels $q_0 - q_1$ of the second block is 29, so it is a moderately complex block. The absolute difference between quantization levels $q_0 - q_1$ of the third block is 32, so it is a highly complex block. Highly complex blocks do not participate in secret payload embedding because pixels in the block are uncorrelated or correlated very little. Thus, only smooth blocks and moderately complex blocks are utilized for data embedding. The mean value of the first block is computed as 190 and the mean value of second block is computed as 103 using Equation (3). In Figure 4b,c, prediction errors are shown for each pixel of the block except for highly complex block. Based on the prediction errors, pixels of special set (x^s), base set (x^b), regular set (x^r) and irregular set (x^i) are determined as shown in Figure 4d. Now, input image is encrypted using stream encryption with encryption key K_e as shown in Figure 4e. Figure 4f shows an 8-bit binary representation of encrypted image. Using the tuple parameter (2,3), labelling bits for each pixel are determined PBTL structure. Pixels of the encrypted image corresponding to regular set (x^r) and irregular set (x^i) are labeled based on predictor errors. Figure 4h shows number of vacancies created in labeled regular pixels for embedding secret payload.

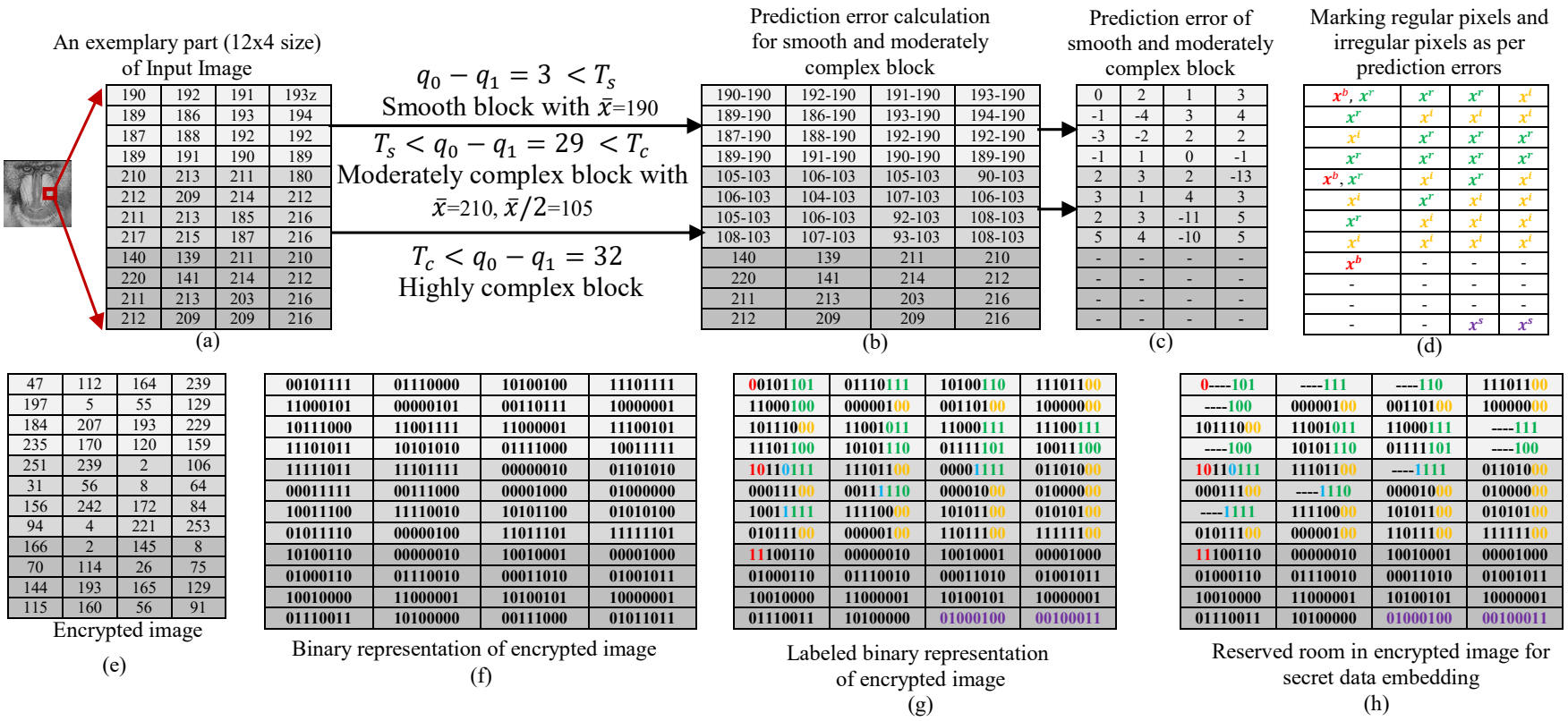

Figure 4. An illustrative example of the proposed scheme. (please resize to the format).

5. Experiment Results and Analysis

In this section, experiment results of the proposed RDHEI technique are discussed and compared with state-of-the-art techniques. To evaluate the performance of the proposed technique, we have taken four test images as shown in Figure 5. The test images are Lena, Airplane, Man, and Baboon, each one of size 512 × 512 with 8-bit depth gray values. The proposed technique has been evaluated using encryption performance and embedding rate. Encryption performance is estimated on two quality parameters, PSNR (peak signal-to-noise ratio) and SSIM (structural similarity), and one security parameter. Embedding rate (ER) basically represents embedding performance, which is measured in terms of bits per pixel (bpp).

(**a**) Lena (**b**) Airplane (**c**) Man (**d**) Baboon

Figure 5. Test images: (**a**) Lena, (**b**) Airplane, (**c**) Man, (**d**) Baboon.

5.1. Encryption Performance of Proposed Technique

In this subsection, encryption performance of the proposed technique is examined by computing pixel distributions of the marked encrypted images. Ideally, it should be uniform for the encrypted data as the encryption process completely destroys correlations present in original data to provide robustness. The proposed technique also provides uniform pixel distribution which is evident from Figure 6a–f, showing histograms of all test images, and Figure 6g–l, showing histograms of marked encrypted images for all the test images. The *X*-axis and *Y*-axis of the histograms represents number of pixels and intensity range (0–255), respectively.

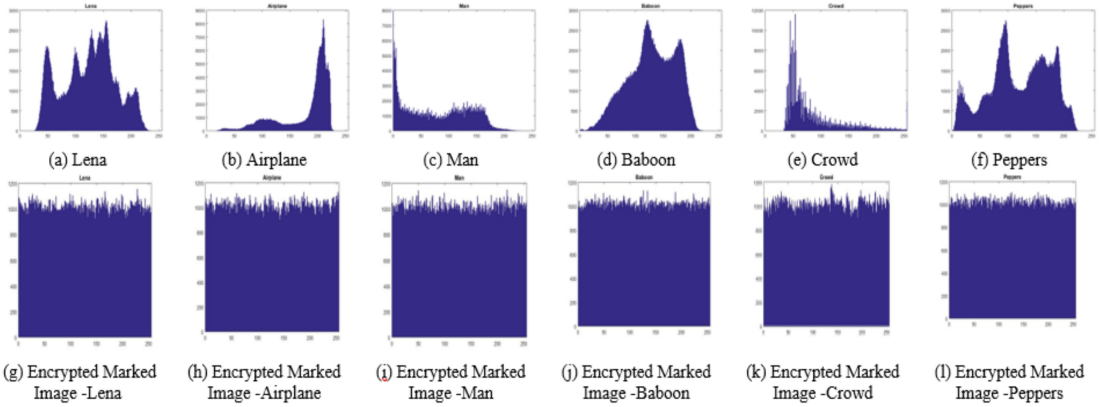

Figure 6. (**a**–**f**) histogram of test images, and (**g**–**l**) histograms of the encrypted marked images.

To further show the encryption performance of proposed technique, PSNR and SSIM values for marked encrypted images corresponding to the original test images are calculated and the results are provided in Table 4. The results are taken by considering tuple parameter (γ_1, γ_2) = (4,4) and block size of 4 × 4 pixels. It is to be noted that the experimental results on other tuple parameter values and block size are also similar. From Table 4, it can be clearly seen that PSNR of each encrypted marked image is very small and the SSIM value is also nearly 0, which indicates that encrypted marked image does not provide any information about the original image and secret payload.

Table 4. PSNR and SSIM values of encrypted marked image.

Test Image	PSNR (dB)	SSIM
Lena	9.2629	0.0203
Airplane	9.0388	0.0355
Man	8.3069	0.0260
Baboon	9.0285	0.0630
Crowd	9.1914	0.0116
Peppers	9.0116	0.0145

Thus, it is validated that the proposed technique provides good encryption performance. Figure 7a–f show the outcomes of different stages of image encryption and secret payload embedding for one of the test images. Figure 7a shows an original Lena image and Figure 7b shows an encrypted Lena image which is encrypted using the encrypted key K_e by employing a standard stream encryption algorithm.

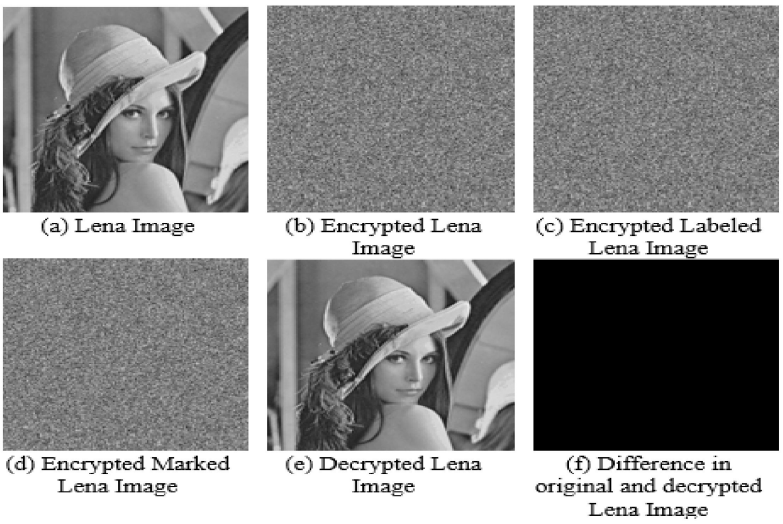

Figure 7. Encryption and decryption of Lena image.

In Figure 7c, an encrypted labeled Lena image is shown and Figure 7d shows a marked encrypted Lena image which includes a secret payload containing the encrypted auxiliary information, encrypted using secret key K_s. In Figure 7e, a decrypted Lena image is shown, which is similar to the original Lena image. In Figure 7f, the difference between Figure 7a,e is seen, showing a black image as all the pixel values are zero. Thus, Figure 7e depicts that the original image is fully recovered as it has PSNR $\rightarrow +\infty$ and SSIM = 1 with respect to the original image.

5.2. Embedding Performance of the Proposed Technique

In this subsection, embedding rate (ER) of the proposed technique is discussed in terms of bpp. The embedding rate has been shown for various values of tuple parameters (γ_1, γ_2 of the PBTL scheme and on different block sizes 4×4, 8×8, 16×16. Further, the thresholds are taken as $T_s = 5$, $T_c = 25$ to categorize image blocks as smooth, moderately complex, and highly complex blocks. Experimental results are provided in Tables 5–8 for test images. From the experimental results, it can be seen that embedding rate is increased, when tuple parameters are also increased up to a limit. Thereafter, embedding rate starts to decrease as tuple parameters are increased. This is because increasing in tuple parameters reduces vacancies for embedding secret data. Further, embedding rate is decreased for parameter γ_2 = 3, 4, 5 when block size is increased and the embedding rate is increased for

parameter γ_2 = 1, 2, 6, 7. The negative values of bpp depict that overhead bits are higher than or approximately equal to reserved room for secret data embedding.

Table 5. Embedding rates of Lena image at different values γ_1 and γ_2 for 4 × 4, 8 × 8, 16 × 16 blocks.

Lena Image, Block Size 4 × 4							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	−0.3802	0.2735	1.2016	1.8779	1.5697	0.7723	−0.085
2	−1.1117	0.5411	1.8115	2.2203	1.6197	0.7736	−0.085
3	−1.8432	0.0459	2.018	2.2900	1.6251	0.7737	−0.085
4	−2.5747	−0.4493	1.8354	2.3141	1.6265	0.7738	−0.085
5	−3.3062	−0.9445	1.6528	2.289	1.6266	0.7737	−0.085
6	−4.0378	−1.4397	1.4701	2.2639	1.6258	0.7737	−0.085
7	−4.7693	−1.9349	1.2875	2.2388	1.6249	0.7737	−0.085
Block Size 8 × 8							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	−0.0894	0.4042	1.1410	1.7434	1.5565	0.8958	0.1655
2	−0.7225	0.5738	1.6104	2.0712	1.6096	0.8983	0.1656
3	−1.3556	0.1291	1.7698	2.1362	1.6177	0.8987	0.1656
4	−1.9887	−0.3156	1.5864	2.1544	1.6197	0.8986	0.1656
5	−2.6218	−0.7603	1.403	2.1244	1.6199	0.8985	0.1656
6	−3.2548	−1.2051	1.2196	2.0944	1.618	0.8986	0.1656
7	−3.8879	−1.6498	1.0362	2.0644	1.6162	0.8985	0.1656
Block Size 16 × 16							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	−0.085	0.2539	0.7632	1.2429	1.189	0.6916	0.1261
2	−0.5862	0.3104	1.0661	1.5345	1.2394	0.6947	0.1263
3	−1.0874	−0.0584	1.1793	1.6044	1.2473	0.6951	0.1264
4	−1.5886	−0.4273	1.0082	1.6205	1.2484	0.695	0.1264
5	−2.0898	−0.7961	0.8371	1.5898	1.2472	0.695	0.1264
6	−2.591	−1.1649	0.666	1.5591	1.2444	0.6949	0.1264
7	−3.0921	−1.5337	0.495	1.5284	1.2415	0.6946	0.1264

Table 6. Embedding rates of Airplane image at different values γ_1 and γ_2 for 4 × 4, 8 × 8, 16 × 16 blocks.

Airplane Image, Block Size 4 × 4							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	0.0450	0.716	1.6555	2.0797	1.6411	0.8698	0.0542
2	−0.6023	1.2755	2.2124	2.3159	1.6763	0.8711	0.0542

Table 6. *Cont.*

Airplane Image, Block Size 4 × 4							
3	−1.2495	0.8933	2.3982	2.3613	1.6808	0.8712	0.0542
4	−1.8968	0.5111	2.2717	2.3693	1.6819	0.8712	0.0542
5	−2.544	0.129	2.1452	2.35	1.6822	0.8712	0.0542
6	−3.1913	−0.2532	2.0187	2.3307	1.6813	0.8712	0.0542
7	−3.8385	−0.6353	1.8922	2.3114	1.6805	0.8712	0.0542
Block Size 8 × 8							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	0.2153	0.7296	1.4998	1.9414	1.6306	0.9878	0.2948
2	−0.3497	1.1507	1.9766	2.1751	1.6671	0.9899	0.2949
3	−0.9147	0.799	2.1473	2.2203	1.6736	0.9902	0.2949
4	−1.4797	0.4472	2.0201	2.2242	1.6749	0.9902	0.2949
5	−2.0447	0.0955	1.8929	2.2018	1.6739	0.9902	0.2949
6	−2.6098	−0.2563	1.7657	2.1794	1.672	0.9902	0.2949
7	−3.1748	−0.6081	1.6385	2.157	1.6701	0.9901	0.2949
Block Size 16 × 16							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	0.1614	0.5260	1.1428	1.5400	1.3086	0.7920	0.2357
2	−0.3015	0.8476	1.5346	1.7483	1.3354	0.7946	0.2359
3	−0.7643	0.5505	1.6901	1.7828	1.3396	0.7947	0.236
4	−1.2271	0.2534	1.5821	1.7851	1.3406	0.7947	0.2361
5	−1.6899	−0.0437	1.4741	1.7664	1.3384	0.7943	0.2361
6	−2.1528	−0.3408	1.3662	1.7478	1.3357	0.794	0.2361
7	−2.6156	−0.6379	1.2582	1.7292	1.333	0.7936	0.2361

Table 7. Embedding rates of Man image at different values γ_1 and γ_2 for 4 × 4, 8 × 8, 16 × 16 blocks.

Man Image, Block Size 4 × 4							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	−0.4069	0.1733	0.9315	1.4777	1.2980	0.5978	−0.2026
2	−1.0968	0.3123	1.3616	1.8139	1.3764	0.6004	−0.2026
3	−1.7867	−0.1722	1.4769	1.8907	1.3885	0.6007	−0.2026
4	−2.4766	−0.6567	1.2620	1.9133	1.3912	0.6007	−0.2026
5	−3.1665	−1.1412	1.0471	1.8714	1.3919	0.6007	−0.2026
6	−3.8565	−1.6256	0.8321	1.8295	1.3902	0.6007	−0.2026
7	−4.5464	−2.1101	0.6172	1.7877	1.3884	0.6007	−0.2026

Table 7. *Cont.*

Block Size 8 × 8							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	−0.1553	0.2655	0.8052	1.2536	1.1992	0.6775	0.0286
2	−0.7307	0.2760	1.0819	1.5326	1.2917	0.6838	0.0287
3	−1.3061	−0.1497	1.1359	1.6029	1.3090	0.6842	0.0287
4	−1.8815	−0.5755	0.9212	1.6142	1.3120	0.6843	0.0287
5	−2.4569	−1.0012	0.7066	1.5595	1.3125	0.6843	0.0287
6	−3.0323	−1.4270	0.4919	1.5048	1.3084	0.6843	0.0287
7	−3.6077	−1.8527	0.2773	1.4502	1.3042	0.6841	0.0287
Block Size 16 × 16							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	−0.1489	0.1368	0.4943	0.8332	0.8754	0.5204	0.0399
2	−0.5902	0.0772	0.6419	1.0590	0.9603	0.5275	0.0402
3	−1.0315	−0.2643	0.6461	1.1145	0.9773	0.5288	0.0402
4	−1.4728	−0.6059	0.4532	1.118	0.9811	0.5290	0.0403
5	−1.9141	−0.9474	0.2602	1.0616	0.9795	0.5289	0.0403
6	−2.3554	−1.2890	0.0673	1.0052	0.9734	0.5287	0.0403
7	−2.7967	−1.6305	−0.1256	0.9488	0.9672	0.5283	0.0403

Table 8. Embedding rates of Baboon at different values γ_1 and γ_2 for 4 × 4, 8 × 8, 16 × 16 blocks.

Baboon Image, Block Size 4 × 4							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	−0.5311	−0.3264	−0.0006	0.4225	0.6605	0.3702	−0.1360
2	−1.0109	−0.4916	0.0785	0.7055	0.8125	0.3801	−0.1360
3	−1.4906	−0.8955	0.0214	0.786	0.8445	0.3810	−0.1360
4	−1.9704	−1.2994	−0.2418	0.7943	0.8536	0.3812	−0.1360
5	−2.4502	−1.7033	−0.505	0.7081	0.8552	0.3813	−0.1360
6	−2.9299	−2.1072	−0.7682	0.6218	0.8490	0.3814	−0.1360
7	−3.4097	−2.5111	−1.0314	0.5356	0.8428	0.3813	−0.1360
Block Size 8 × 8							
				γ_2			
γ_1	1	2	3	4	5	6	7
1	−0.2529	−0.1006	0.1333	0.4396	0.6186	0.4100	0.0215
2	−0.6273	−0.2436	0.1757	0.6395	0.7409	0.4217	0.0216
3	−1.0017	−0.5607	0.1120	0.6929	0.7663	0.4232	0.0216
4	−1.3760	−0.8779	−0.0999	0.6905	0.7723	0.4236	0.0216
5	−1.7504	−1.1951	−0.3117	0.6151	0.7712	0.4237	0.0216

Table 8. *Cont.*

6	−2.1248	−1.5122	−0.5236	0.5397	0.7633	0.4238	0.0216
7	−2.4992	−1.8294	−0.7355	0.4642	0.7553	0.4236	0.0216
Block Size 16 × 16							
	γ_2						
γ_1	1	2	3	4	5	6	7
1	−0.1643	−0.052	0.1171	0.3358	0.4629	0.3089	0.0168
2	−0.4502	−0.1665	0.1343	0.4706	0.5529	0.3198	0.0174
3	−0.7361	−0.4093	0.0777	0.5044	0.5715	0.3212	0.0174
4	−1.0220	−0.6522	−0.0874	0.5004	0.5745	0.3214	0.0174
5	−1.3079	−0.8951	−0.2525	0.4383	0.5720	0.3213	0.0174
6	−1.5937	−1.1379	−0.4176	0.3761	0.5634	0.3210	0.0174
7	−1.8796	−1.3808	−0.5827	0.3140	0.5548	0.3204	0.0174

5.3. Comparison of Results of the Proposed Technique with State of Art Techniques

In this subsection, experimental results of the proposed techniques are compared with related techniques including those of Yi et al. [32], Su et al. [33], Li et al. [27], Puteaux et al. [29], Puyang et al. [30], Chen et al. [31]. Since PBTL was first introduced in the RDHEI domain by Yi et al., the same [24] has been primarily considered for comparison. Su et al.'s [33] technique demonstrates application of AMBTC concept along with PBTL. However, it is a lossy technique where a decrypted image is obtained in compressed form. Since the proposed technique also considers some aspects of AMBTC in block categorization, Su et al.'s [33] technique has also been used to comparatively evaluate the performance. However, the proposed technique is a lossless and fully reversible technique in which both secret payload and original image are retrieved in undistorted form. Additionally, the performance of the proposed technique is evaluated against some of the early high capacity RDHEI techniques, e.g., Li et al. [27], Puteaux et al. [29], Puyang et al. [30], and Chen et al. [31]. Table 9 shows the encryption method of the proposed technique and state-of-art techniques. In comparison to this, Yi et al. [32] retains some pixel correlations in blocks which makes encryption week.

Table 9. Comparison of encryption performance with state-of-art techniques.

Technique	Encryption Method	Preserving Pixel Redundancy
Yi et al. [32]	Block Permutation and block modulation	Yes
Su et al. [33]	Block Scrambling and Stream Encryption	Yes
Li et al. [27]	Block Permutation and Stream Encryption	No
Puteaux et al. [29]	Stream Encryption	No
Puyang et al. [30]	Stream Encryption	No
Chen et al. [31]	Stream Encryption	No
Proposed RDHEI	Stream Encryption	No

For optimal embedding performance, $\gamma_1 = 4$ & $\gamma_2 = 4$ and a block size of 4×4 pixels is considered for the proposed RDHEI technique. The embedding performance is compared with some of the high-capacity state-of-the-art techniques that provide highest embedding

rate. Since Yi et al.'s [32] and Chen et al. [33] have optimal performance at $\gamma_1 = 2$ & $\gamma_2 = 5$ with block size of 3×3 pixels and block size of 4×4 pixels, respectively.

Figure 8 shows the maximal embedding rates of the proposed technique as well as existing RDHEI techniques for all the test images. It clearly shows that proposed technique has highest embedding rate than other techniques. It is also evident from the figure that the proposed scheme provides higher embedding capacity with smooth images like Lena, Airplane, whereas it has lower embedding capacity with complex images such as Baboon.

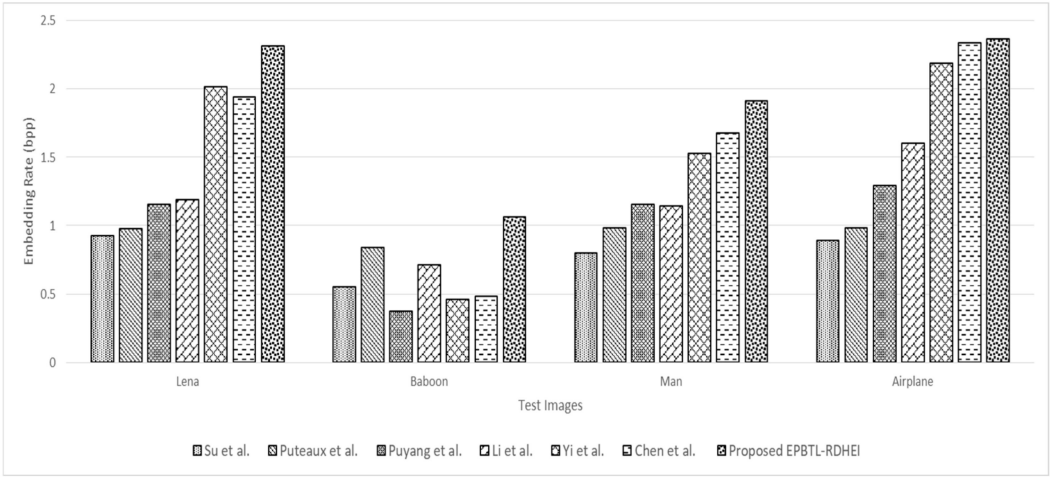

Figure 8. Comparison of embedding rate of the proposed and state-of-art techniques.

6. Conclusions

In this paper, an intra-block correlation based high capacity RDHEI technique using a parametric binary tree labelling scheme has been proposed. In the proposed technique, the cover image was divided into blocks and then the blocks were categorized according to the prevalent correlation through a symmetric or asymmetric process. Next, an adaptive method for reserving the room inside the image was applied, based on the block categories so that a large amount of secret data could be embedded. Further, the proposed RDHEI technique used a stream cipher for encrypting the image contents. Experimental results showed that the proposed RDHEI technique provided the highest embedding rate in comparison to all the aforementioned state-of-the art techniques. Additionally, the proposed method provided a good level of security in encryption process to protect the privacy of the original plaintext image. In future work, compression methods can be explored to further condense the size of prediction errors, and also an improvised AMBTC method may be designed to predict the pixel values more accurately.

Author Contributions: Conceptualization, A.K.R. and N.K.; Methodology, A.K.R. and N.K.; Visualization, R.K.; Writing—A.K.R., N.K.; Writing—review and editing, R.K., H.O., S.C. and K.-H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Brain Pool program funded by the Ministry of Science and ICT through the National Research Foundation of Korea (2019H1D3A1A01101687) and Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1A09081842 and 2021R1I1A3049788).

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Symbols	**Meaning**
A_b	auxiliary information
μ	Mean
B_k	Block
b_i	Logic bit value
K_e	Encryption Key
K_s	Secret Key
n	Number of Layers
γ_1, γ_2	Tuple Parameter
S_1, S_2	Two different sets
N_{γ_2}	Total number of nodes in the set S_2
$\mathbb{I}$	plaintext image
I^e	Encrypted image
$N_1 \times N_2$	Image Size
$n_1 \times n_2$	Block Size
O_b	Overhead
p	Number of non-overlapping blocks
q_0, q_1	Quantization values
T_s, T_c	Threshold values
x_i	i-th pixel
x^s	Special set
x^b	Base set
x^r	Regular set
x^i	Irregular set
Abbreviation	**Meaning**
AES	Advanced Encryption Standard
AMBTC	Absolute Moment Block Truncation Coding
BPP	Bit-Per-Pixel
EC	Embedding Capacity
ER	Embedding Rate
LSB	Least Significant Bits
MSB	Maximum Significant Bits
PBTL	Parametric Binary Tree Labeling Scheme
PE	Prediction Error
PSNR	Peak Signal-To-Noise Ratio
RDH	Reversible Data Hiding
RDHEI	Reversible Data Hiding in Encrypted Images
RLE	Run Length Coding
RRBE	Reserving Room Before Encryption
SSIM	Structural Similarity Index
VRAE	Vacating Room After Encryption
VRBE	Vacating Room by Encryption

References

1. Pahl, C.; Brogi, A.; Soldani, J.; Jamshidi, P. A State-of-the-Art Review. *IEEE Trans. Cloud Comput.* **2019**, *7*, 677–692. [CrossRef]
2. Chandramouli, R.; Iorga, M.; Chokhani, S. Cryptographic key management issues and challenges in cloud services. *Secure Cloud Comput.* **2014**, 1–30. [CrossRef]
3. Kumar, R.; Chand, S. A reversible high capacity data hiding scheme using pixel value adjusting feature. *Multimed. Tools Appl.* **2016**, *75*, 241–259. [CrossRef]
4. Lee, C.; Shen, J.; Wu, Y.; Agrawal, S. PVO-based reversible data hiding exploiting two-layer embedding for enhancing image fidelity. *Symmetry* **2020**, *12*, 1164. [CrossRef]
5. Khan, S.; Khan, K.; Arif, A.; Hassaballah, M.; Ali, J.; Ta, Q.; Yu, L. A modulo function-based robust asymmetric variable data hiding using DCT. *Symmetry* **2020**, *12*, 1659. [CrossRef]
6. Kim, S.; Cho, N.I. Hierarchical Prediction and Context Adaptive Coding for Lossless Color Image Compression. *IEEE Trans. Image Process.* **2014**, *23*, 445–449. [CrossRef] [PubMed]

7. Alattar, A.M. Reversible Watermark Using the Difference Expansion of a Generalized Integer Transform. *IEEE Trans. Image Process.* **2004**, *13*, 1147–1156. [CrossRef] [PubMed]
8. Kumar, R.; Jung, K.-H. Enhanced pairwise IPVO-based reversible data hiding scheme using rhombus context. *Inf. Sci.* **2020**, *536*, 101–119. [CrossRef]
9. Wu, H.; Li, X.; Zhao, Y.; Ni, R. Improved reversible data hiding based on PVO and adaptive pairwise embedding. *J. Real-Time Image Process.* **2019**, *16*, 685–695. [CrossRef]
10. Ou, B.; Li, X.; Wang, J. High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction-error expansion. *J. Vis. Commun. Image Represent.* **2016**, *39*, 12–23. [CrossRef]
11. Gao, G.; Tong, S.; Xia, Z.; Wu, B.; Xu, L.; Zhao, Z. Reversible data hiding with automatic contrast enhancement for medical images. *Signal Process.* **2021**, *178*, 107817. [CrossRef]
12. Kumar, R.; Jung, K.-H. Robust reversible data hiding scheme based on two-layer embedding strategy. *Inf. Sci.* **2020**, *512*, 96–107. [CrossRef]
13. Puech, W.; Chaumont, M.; Strauss, O. A reversible data hiding method for encrypted images. In Proceedings of the SPIE, San Jose, CA, USA, 18 March 2008; pp. 68191E-1–68191E-9.
14. Bhardwa, J.R.; Aggarwal, A. An improved block based joint reversible data hiding in encrypted images by symmetric cryptosystem. *Pattern Recognit. Lett.* **2020**, *139*, 60–68. [CrossRef]
15. Hong, W.; Chen, T.-S.; Wu, H.-Y. An Improved Reversible Data Hiding in Encrypted Images Using Side Match. *IEEE Signal Process. Lett.* **2012**, *19*, 199–202. [CrossRef]
16. Zhou, J.; Sun, W.; Dong, L.; Liu, X.; Au, O.C.; Tang, Y.Y. Secure Reversible Image Data Hiding Over Encrypted Domain via Key Modulation. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *26*, 441–452. [CrossRef]
17. Wu, X.; Sun, W. High-capacity reversible data hiding in encrypted images by prediction error. *Signal Process.* **2014**, *104*, 387–400. [CrossRef]
18. Zhang, W.; Ma, K.; Yu, N. Reversibility improved data hiding in encrypted images. *Signal Process.* **2014**, *94*, 118–127. [CrossRef]
19. Zhang, X. Separable Reversible Data Hiding in Encrypted Image. *IEEE Trans. Inf. Forensics Secur.* **2012**, *7*, 826–832. [CrossRef]
20. Kaur, G.; Singh, S.; Rani, R.; Kumar, R.; Malik, A. High-quality reversible data hiding scheme using sorting and enhanced pairwise PEE. *IET Image Process.* **2021**, 1–15. [CrossRef]
21. Dragoi, I.C.; Coanda, H.-G.; Coltuc, D. Improved reversible data hiding in encrypted images based on reserving room after encryption and pixel prediction. In Proceedings of the 25th European Signal Processing Conference (EUSIPCO), Kos Island, Greece, 28 August–2 September 2017; pp. 2186–2190.
22. Dragoi, I.C.; Coltuc, D. Reversible Data Hiding in Encrypted Color Images Based on Vacating Room After Encryption and Pixel Prediction. In Proceedings of the 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1673–1677.
23. Ma, K.; Zhang, W.; Zhao, X.; Yu, N.; Li, F. Reversible Data Hiding in Encrypted Images by Reserving Room Before Encryption. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 553–562. [CrossRef]
24. Xu, D.; Wang, R. Separable and error-free reversible data hiding in encrypted images. *Signal Process.* **2016**, *123*, 9–21. [CrossRef]
25. Mathew, T.; Wilscy, M. Reversible data hiding in encrypted images by active block exchange and room reservation. In Proceedings of the 2014 International Conference on Contemporary Computing and Informatics (IC3I), Mysore, India, 27–29 November 2014; pp. 839–844.
26. Cao, X.; Du, L.; Wei, X.; Meng, D.; Guo, X. High Capacity Reversible Data Hiding in Encrypted Images by Patch-Level Sparse Representation. *IEEE Trans. Cybern.* **2016**, *46*, 1132–1143. [CrossRef] [PubMed]
27. Li, Q.; Yan, B.; Hui, L.; Chen, N. Separable reversible data hiding in encrypted images with improved security and capacity. *Multi. Tools Apps.* **2018**, *77*, 30749–30768. [CrossRef]
28. Hang, Y.; Wei, T.; Menghan, G.; Shoushun, C. A two-step prediction ADC architecture for integrated low power image sensors. *IEEE Trans. Circuits Syst. I* **2017**, *64*, 50–60.
29. Puteaux, P.; Puech, W. An Efficient MSB Prediction-Based Method for High-Capacity Reversible Data Hiding in Encrypted Images. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1670–1681. [CrossRef]
30. Puyang, Y.; Yin, Z.; Qian, Z. Reversible Data Hiding in Encrypted Images with Two-MSB Prediction. In Proceedings of the 2018 IEEE International Workshop on Information Forensics and Security (WIFS), Hong Kong, China, 11–13 December 2018; pp. 1–7.
31. Kaimeng, C.; Chang, C.C. High-capacity Reversible Data Hiding in Encrypted Images Based on Extended Run-Length Coding and Block-based MSB Plane Rearrangement. *J. Vis. Commun. Image Represent.* **2019**, *58*, 334–344.
32. Yi, Y.; Zhou, Y. Separable and Reversible Data Hiding in Encrypted Images Using Parametric Binary Tree Labeling. *IEEE Trans. Multimed.* **2019**, *21*, 51–64. [CrossRef]
33. Su, G.-D.; Chang, C.-C.; Lin, C.-C. A High Capacity Reversible Data Hiding in Encrypted AMBTC-Compressed Images. *IEEE Access* **2020**, *8*, 26984–27000. [CrossRef]
34. Yin, Z.; Niu, X.; Zhang, X.; Tang, J.; Luo, B. Reversible data hiding in encrypted AMBTC images. *Multimed. Tools Appl.* **2018**, *77*, 18067–18083. [CrossRef]
35. Yin, Z.; Wang, H.; Zhao, H.; Luo, B.; Zhang, X. Complete separable reversible data hiding in encrypted image. In Proceedings of the 1st International Conference on Cloud Computing and Security, Nanjing, China, 13–15 August 2015; pp. 101–110.

36. Wang, R.; Wu, G.; Wang, Q.; Yuan, L.; Zhang, Z.; Miao, G. Reversible Data Hiding in Encrypted Images Using Median Edge Detector and Two's Complement. *Symmetry* **2021**, *13*, 921. [CrossRef]
37. Kumar, R.; Kim, D.-S.; Jung, K.-H. Enhanced AMBTC based data hiding method using hamming distance and pixel value differencing. *J. Inf. Secur. Appl.* **2019**, *47*, 94–103. [CrossRef]
38. Kim, S. Reversible Data-Hiding Systems with Modified Fluctuation Functions and Reed-Solomon Codes for Encrypted Image Recovery. *Symmetry* **2017**, *9*, 61. [CrossRef]