applied sciences

MDPI

Article

Data Hiding of Multicompressed Images Based on Shamir Threshold Sharing

Haoyang Kang [1], Lu Leng [1,*] and Byung-Gyu Kim [2,*]

[1] School of Software, Nanchang Hangkong University, Nanchang 330063, China
[2] Department of IT Engineering, Sookmyung Women's University, Seoul 04310, Korea
* Correspondence: leng@nchu.edu.cn (L.L.); bg.kim@sookmyung.ac.kr (B.-G.K.); Tel.: +86-791-8645-3251 (L.L.)

Abstract: Image-based data hiding methods have been used in the development of various applications in computer vision. At present, there are two main types of data hiding based on secret sharing, namely dual-image data hiding and multi-image data hiding. Dual-image data hiding is a kind of secret sharing-based data hiding in the extreme case. During the image transmission and storage process, the two shadow images are visually highly similar. Multi-image data hiding disassembles the cover image into multiple meaningless secret images through secret sharing. Both of the above two methods can easily attract attackers' attention and cannot effectively guarantee the security of the secret message. In this paper, through the Shamir threshold scheme for secret sharing, the secret message is disassembled into multiple subsecrets that are embedded in the smooth blocks of multiple different images, by substituting the bitmap of block truncation coding. Thus, the shortcomings of the above two data hiding methods are effectively avoided. The proposed method embeds the secret messages in the compressed images, so it satisfactorily balances the visual quality and the embedding capacity. In our method, the shadow images make sense while they are not visually similar. The compression ratio is four, so the embedding capacity of our method has an obvious advantage under the same storage space.

Keywords: data hiding; secret sharing; high embedding capacity; low loss

check for updates

Citation: Kang, H.; Leng, L.; Kim, B.-G. Data Hiding of Multicompressed Images Based on Shamir Threshold Sharing. *Appl. Sci.* **2022**, *12*, 9629. https://doi.org/10.3390/app12199629

Academic Editor: Andrea Prati

Received: 31 August 2022
Accepted: 21 September 2022
Published: 25 September 2022

1. Introduction

Traditional encryption protects the content of secret messages through the incomprehensibility of ciphertext. However, the incomprehensibility of ciphertext exposes the importance of the secret message, which easily arouses attackers' curiosity and attention. Thus, attackers would attempt to decipher the ciphertext or destroy the communication [1,2].

Image-based data hiding (DH) methods have been used in the development of various applications in computer vision. In contrast to traditional cryptography, DH hides secret messages in multimedia files by taking advantage of the redundancy of multimedia files, while it does not cause significant perceptive distortion [3]. Even if an interceptor knows of the existence of the secret message, it is difficult to extract the secrets without authorization; accordingly, DH ensures the confidentiality and security of the secret message. DH can be applied in many fields, such as digital signature, fingerprint identification, authentication and secret communication [4].

Digital images are often used as the cover media because they are accessible and available in the redundancy [5]. DH in digital images has two principal standards: embedding capacity and the visual quality of the stego image. Embedding more secret data typically leads to more severe image distortion, so an excellent DH algorithm should embed secret data as much as possible, while ensuring that there is no obvious visual difference between the stego image and the original image.

In accordance with the restoration of the cover image after secret message extraction, DH can be briefly divided into two classes, namely nonreversible DH (NRDH) and reversible DH (RDH). RDH should satisfy the visual quality requirement, and ensure that the receiver can extract the secret message correctly and completely. In addition, RDH must restore the cover image completely with no distortion.

Representative RDH algorithms typically include Difference Expansion (DE), image compression, Histogram Shifting (HS), Prediction Error Expansion (PEE), Pixel Value Order (PVO) [6–9] and encrypted image reversible data hiding [10,11]. Representative NRDH algorithms typically include Block Truncation Coding (BTC) [12], Absolute Moment Block Truncation Coding (AMBTC) [13–15] and Least Significant Bit (LSB) [16–18].

Although the traditional data hiding scheme can prevent malicious attackers from stealing the secret message embedded in the stego image to a certain extent, once the data hiding method is cracked by the attacker, the security of the secret message will be questioned [19]. Secret sharing is one of the important technologies to ensure the storage and transmission of secret messages. Instead of hiding secret messages on a single carrier, it splits the secret messages into n secrets and stores and transmits them separately. A subsecret is called a share, and an image embedded with the share as the secret message is called a shadow image. Only when m or more than m shares cooperate, the merging and extraction of secret messages can be completed. Therefore, the data hiding method based on secret sharing can improve the security of secret message in storage and transmission. The number m of shares used in secret message extraction is often smaller than the total number of splits n; even if some shares are destroyed during storage and transmission, it does not affect the final secret message extraction, so this method can also improve the fault tolerance of the data hiding method.

(a) Dual-image secret sharing data hiding

Chang et al. first proposed dual-image data hiding based on Exploiting Modification Direction (EMD) in 2007 [20]. In this method, the binary secret message is converted into 5-based secret digitals, and then two binary secret digitals are embedded into a pixel pair. The embedding capacity of this method is 1 bit per pixel (bpp), and although the visual quality of the stego images is slightly lower, this method can fully restore the original image after secret message extraction. This method is a kind of secret sharing data hiding in the extreme case. Although the security of the secret message is improved, the authorized receiver cannot recover the secret message without the complete two shadow images.

In 2013, Chang et al. proposed a magic matrix-based two-image RDH method [21]. Each group of two original pixels can be used to embed a three-bit secret message. During the embedding process, the maximum modification level of the covered pixels is four. Although the embedding capacity of this method reaches 1.55 bpp, the visual quality of the two stego images is not good.

In 2015, Qin et al. proposed a reversible data hiding method based on EMD [22]. In the process of secret embedding, the traditional EMD method is used to modify the pixels in the first secret digital by no more than one gray level to realize the embedding of secret data. Although the pixels in the second stego image are adaptively modified by referring to the first stego image, there is no confusion in the image restoration process. At the receiving end, the secret data can be easily extracted, and the original cover image can be correctly recovered from the two stego images. The method maintains acceptable visual quality for two cryptic images, while keeping the embedding rate of 1.16 bpp.

In 2018, Liu et al. proposed secret sharing data hiding based on the Turtle shell (TS) matrix [23], which splits the secret message into two parts. The secret message is hidden in the cover image through the TS matrix. Compared with the EMD method, the visual quality of the covert image of this method is greatly improved, but the visual quality of the two shadow images is different, which easily attracts the attackers' attention.

In 2022, Chen et al. proposed a reversible data hiding method based on the combination of pixel pair orientations in double stego images [24]. The pixel pairs of the two hidden images are divided into the main embedded pixel pair and the auxiliary embedded pixel pair. There are 25 combinations of dual stego-pixel pairs in a 3×3 block and the original pixel value can be uniquely determined in extracting phase. The embedding capacity of this scheme is about 1.14 bpp, and the two generated shadow images also have advanced visual quality of 49.92 dB.

In 2021, Chen et al. made full use of the characteristics of the EMD matrix. One secret bit and a base-5 digit can be concealed in a cover pixel to generate a pixel pair that is assigned to two stego images. Unlike previous methods, the authors hid a secret bit and a base-5 secret number in a pixel instead of a pixel pair [25]. As a result, the embedding rate reached 1.56 bpp, but the Peak Signal to Noise Ratio (PSNR) of the two covert images was lower than most related algorithms.

(b) Multi-image secret sharing data hiding

In 2018, Wu et al. proposed an image encryption algorithm based on Shamir secret sharing [26], which uses secret sharing technology to encrypt the original image into multiple meaningless ciphertext shadow images and hides the original reversible data in shadow images; the proposed method is extended using two algorithms, Difference Expansion and Histogram Shift. The experimental results show that the method has low computational complexity, large embedding capacity, and can restore the original image losslessly.

In 2020, Zhou et al. proposed a homomorphic encryption reversible data hiding algorithm based on secret sharing [27]. The algorithm first uses the Shamir secret sharing scheme to encrypt the image. The secret message is embedded by using the homomorphic properties of the Shamir secret sharing scheme. The encryption and embedding stages use the same operation. Compared with similar algorithms, this algorithm has lower time complexity.

In 2020, Xiong et al. proposed a novel reversible data hiding over Distributed Encrypted-Image Servers (RDH-DES) based on secret sharing [28]. The Chinese Remainder Theorem and block-level scrambling were developed as a lightweight cryptography to generate the encrypted image shares. However, the image shares are meaningless and easily attract attackers' attention.

The data hiding technology based on secret sharing is mainly divided into double-image data hiding and multi-image data hiding. Double-image data hiding is a secret sharing data hiding technique in extreme cases, which has considerable hiding performance but no fault tolerance. Although multi-image data hiding has certain fault tolerance, the shadow images embedded in the secret message are transmitted and stored as a ciphertext image, which cannot avoid the defects of traditional encryption technology and reduces the security of the secret message. Considering the advantages and disadvantages of the above two hiding techniques, the algorithm proposed in this paper uses the secret sharing technique to divide the secret message into multiple shares and embed them into multiple images, to ensure that it is fault-tolerant. The embedded carrier selects meaningful digital images to ensure the concealment and security of secret messages. At the same time, in practical application, the proposed method embeds the secret message into multiple AMBTC compressed images to reduce the cost for transmission and storage.

The organization of this paper is as follows: Section 2 revisits related works, and briefly describes the Shamir threshold secret sharing scheme and AMBTC. The proposed method is elaborated in Section 3. The experimental results are demonstrated in Section 4. Finally, the conclusions and future works are given in Section 5.

2. Related Works

2.1. Shamir Threshold Secret Sharing Scheme

Secret sharing is a cryptographic technology that separates and stores secrets. The purpose is to prevent secrets from being too concentrated, so as to achieve the purpose of dispersing risks and tolerating intrusions. It is an important means in information security and data secrecy. The idea of secret sharing is to split the secret in an appropriate way. Each share is managed by a different participant. A single participant cannot recover the secret information. Only several participants can cooperate to recover the secret message. More importantly, in the event of a problem with any of the corresponding inscope participants, the secret can still be fully recovered.

The (*m*, *n*) threshold secret sharing scheme, is a famous secret sharing solution proposed by the Turing Award winner, Shamir, in 1979. The scheme divides the secret message into multiple parts by constructing a polynomial of degree *m*-1, and distributes it to multiple participants, thereby ensuring the security of the secret message; m denotes that at least m participants can jointly recover the secret message, any less than or equal to *m*-1 participants cannot recover the secret message, and n denotes the number of participants participating in splitting the secret message.

The (*m*, *n*) threshold scheme works as follows:

(1) Split the secret message *s* into n shares; each participant gets one share.
(2) When splitting, it is preset that at least *m* ($m \leq n$) participants gather together to restore s.
(3) When *s* needs to be restored, and there are at least m participants gathered together, they take out their respective shares and restore *s*, and when less than m participants gather together, it is impossible to restore *s*.

Shamir gives a clever way to construct the threshold scheme. The (*m*, *n*) threshold scheme includes two algorithms: a secret segmentation algorithm and a secret reconstruction algorithm.

2.1.1. Secret Segmentation Algorithm

Pick a random prime *p* and generate a random polynomial of degree $m-1$:

$$f(x) = a_{m-1}x^{m-1} + a_{m-2}x^{m-2} + \cdots + a_2x^2 + a_1x + a_0 \ mod \ p$$

where $s = a_0,\ f(0) = a_0 = s$.

Take any *n* positive integers $x_1,\ x_2,\ \cdots,\ x_n$ that are not equal to each other, have $f(x_1), f(x_2), \cdots, f(x_n)$.

$(x_1, f(x_1)), (x_2, f(x_2)), \cdots, (x_n, f(x_n))$ will be distributed to n participants.

2.1.2. Secret Reconstruction Algorithm

When the number of participants reaches m or more, $f(x)$ can be reconstructed by any *m* shares, assuming that the shares of *m* participants are $(x_1, f(x_1)), (x_2, f(x_2)), \cdots, (x_m, f(x_m))$. *m* shares can be regarded as points on the $f(x)$. According to the properties of the Lagrangian interpolation polynomial, if the number of point is not less than *m*, $f(x)$ can be exactly reconstructed.

$$f(x) = \sum_{i=1}^{m} f(x_i) \prod_{\substack{j = 1 \\ j \neq i}}^{m} \frac{x - x_j}{x_i - x_j} \tag{1}$$

After reconstruction $f(x)$, the secret message *s* can be extracted.

2.2. Absolute Matrix Block Truncation Coding

The Block Truncation Coding (BTC) algorithm is a widely used image compression technology. Two appropriate gray levels can be selected to approximately represent the original gray level of the image block, and then it is necessary to indicate which gray level each pixel in the image block belongs to. The BTC compression algorithm has low complexity and a fast encoding and decoding speed, and can be applied to devices with relatively low computing power, such as mobile phones and drones, that require real-time transmission to obtain reconstructed images with better visual performance.

The Absolute Moment Block Truncation Coding (AMBTC) algorithm is an improved version of BTC. It uses absolute moments to calculate and reconstruct high and low gray values, while BTC uses variance to calculate high and low brightness values, so this method is complicated to calculate. The degree of compression is lower, and the compression error is smaller under the same compression ratio; that is, the obtained image quality is better.

In AMBTC, the image is first divided into non-overlapping image blocks, where the value of k can be set to 4, 8, 16, etc. The average value of each image block is calculated by:

$$\overline{x} = \frac{1}{k \times k} \sum_{i=1}^{k^2} x_i \tag{2}$$

where x_i represents the gray value of the *i*-th pixel. For each image block, the bitmap BM represented by {0, 1} is obtained by comparison with the average value:

$$b_i = \begin{cases} 1, \ if \ x_i \geq \overline{x} \\ 0, \ if \ x_i < \overline{x} \end{cases} \tag{3}$$

The high and low volumes are calculated by:

$$\begin{cases} H = \frac{1}{t} \sum\limits_{x_i \geq \overline{x}} x_i \\ L = \frac{1}{k \times k - t} \sum\limits_{x_i < \overline{x}} x_i \end{cases} \tag{4}$$

where *t* represents the number of pixels represented by "1" in the bitmap. After the high amount H, the low amount L and the bitmap BM are calculated, they can be combined into the basic unit triplet {H, L, BM}, and then the image is compressed by AMBTC. For *k* = 4, 8 + 8 + 16 = 32 bits can be used to represent a 4 × 4 block of an original image, so the compression ratio is (16 × 8)/32 = 4. That is, for an image, 2 M bits can be compressed into 0.5 M bits.

The Shamir threshold scheme can be used to segment the original secret message and store the subsecret messages in different servers, thereby it can improve the confidentiality and fault tolerance of the original secret message. However, the shadow images generated by these methods are often meaningless and easily attract attackers' attention. These shadow images cannot be effectively compressed. In this paper, the subsecret messages were embedded into multiple meaningful images that are compressed by AMBTC, which can effectively avoid attackers' attention.

3. Methodology

As shown in Figure 1, the user first generates n subsecret messages through the secret sharing method, and the subsecret messages are embedded into n original images through the Direct Bitmap Substitution (DBS) algorithm after AMBTC compression. The encoded bitmap is then uploaded to multiple servers for separate storage, rather than having all the covert images in one server. This method can prevent malicious attackers from obtaining all the shadow images at the same time, effectively ensuring the security of secret messages. Since the AMBTC used is a compression method with a high compression rate, it can greatly reduce the space required for transmission and storage. Figure 2 shows the extraction process of the secret message. The user downloads at least k shadow images from different servers, then extracts the subsecret messages of the secret images, respectively, and finally restores the secret message through the secret reconstruction algorithm. Even if a small number of secret images are damaged during storage or transmission, as long as the number of secret images available to the user is at least *k*, the secret message can be recovered, thus improving the fault tolerance of the method.

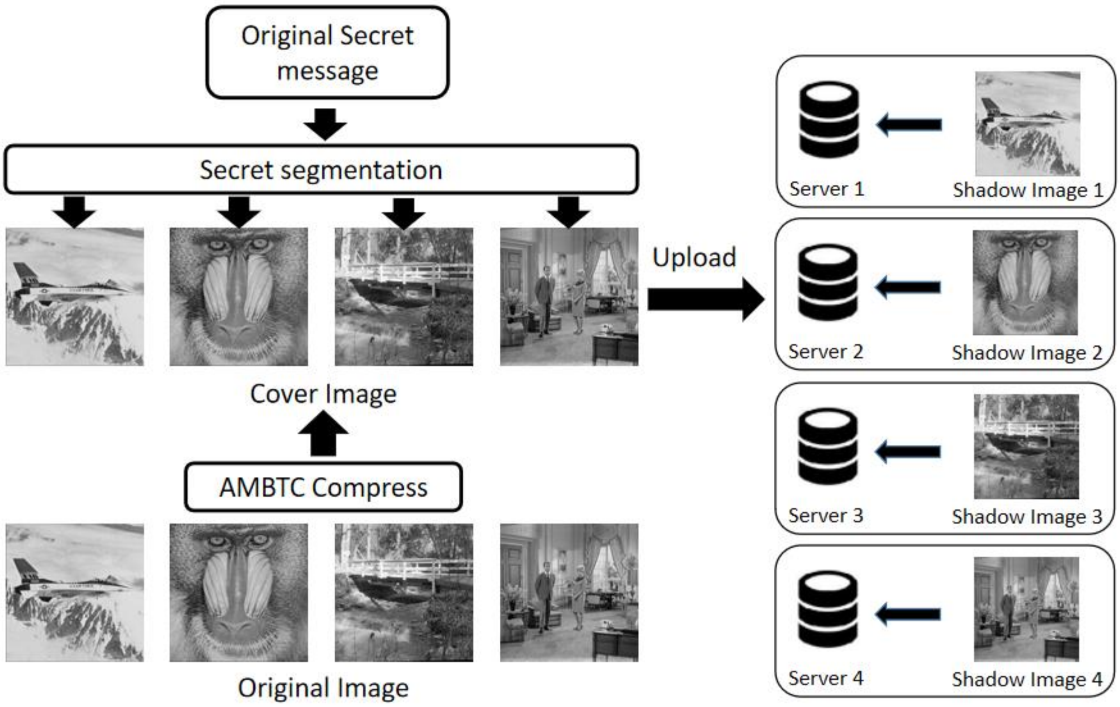


Figure 1. Embedding framework of Shamir threshold sharing data hiding.

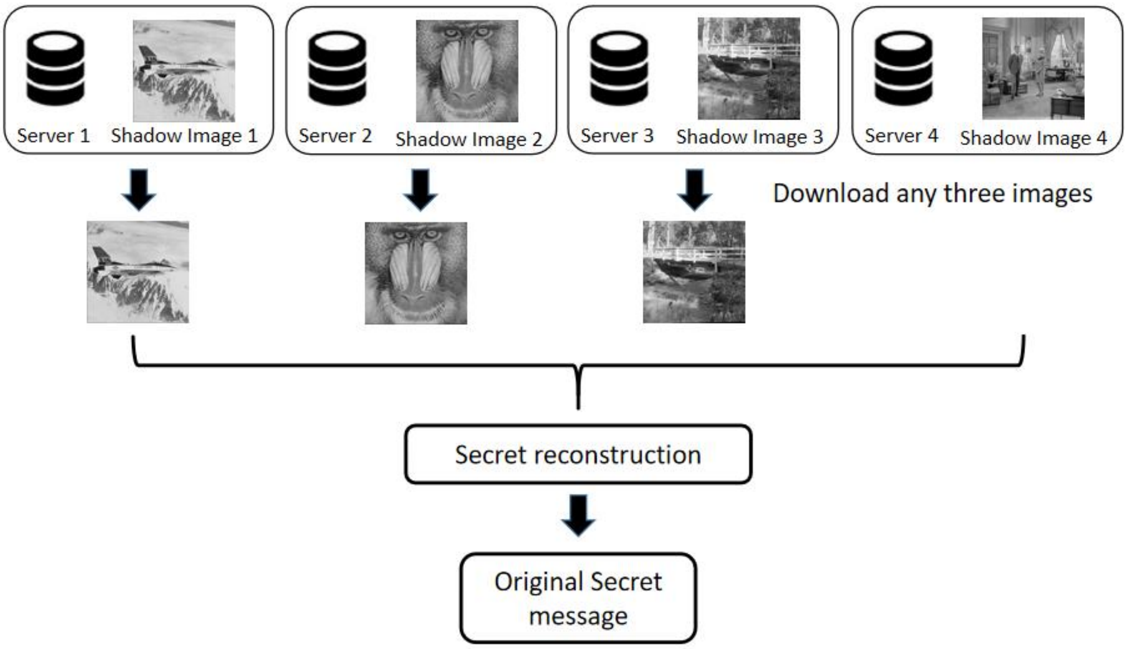


Figure 2. Extracting framework of Shamir threshold sharing data hiding.

3.1. Embedding Procedure

Assuming that the secret message is divided into n = 4 parts by the Shamir threshold scheme, it is preset that m = 3 parts of the secret message can restore the original secret message. When using the Direct Bitmap Substitution (DBS) method to embed secret messages, because the number of smooth blocks in cover images is different, it is necessary to first count the number of smooth blocks in each image, select the minimum value of the number of smooth blocks num, and the bitmap of each smooth block. A total of 16 bits can be embedded, that is, 6 secret numbers 0~15 represented by 4 bits, so the total amount of embedding is *l* = *num* × 6.

3.1.1. Secret Sharing Stage

Unlike the EMD method, which converts the secret message into 5-based secret digital, the method proposed in this paper firstly groups the random binary secret message with every 4 bits to obtain a secret number sequence of $S = \{S_1, S_2, S_3, S_4, \cdots, S_n | 0 \leq S \leq 15\}$, and then let *S* = *S* + 1, and *S* becomes a secret number sequence from 1 to 16.

Let the prime number *p* = 17, take three secret numbers *s* = {*s*1, *s*2, *s*3} from the secret number sequence *S* in turn, and use *s* as the coefficient of the polynomial to generate a second-degree polynomial:

$$f(x) = s_1 x^2 + s_2 x + s_1 \; mod \; p$$

where $s \geq 1$ and $0 \leq f(x) \leq 16$.

Take any 4 positive integers $x = [x_1, x_2, x_3, x_4]$, $(1 \leq x \leq 16)$ that are not equal to each other and substitute them into the polynomial to get $y = [f(x_1), f(x_2), f(x_3), f(x_4)]$. If *y* contains an element of 0, recalculate 4 different integers until *y* does not contain a 0 element.

Let $x = x - 1$, $y = y - 1$, $0 \leq x \leq 15$, $0 \leq y \leq 15$. Let [*x*, *y*] be converted to a 4-bit binary secret. $(x_1, f(x_1)), (x_2, f(x_2)), (x_3, f(x_3)), (x_4, f(x_4))$ will be distributed to 4 participants as a share.

Repeat the above steps until all the secret numbers are processed.

As shown in Figure 3, in the secret sharing stage, *S* = {1, 2, 3, 4, 5, 6, 7,$\cdots$,S_l}, and the two groups of three secret numbers are S_1 ={1, 2, 3}, S_2 ={5, 6, 7}, respectively. Take them as the binomial coefficient, the two generated polynomials are $f_1(x) = x^2 + 2x + 3$ and $f_2(x) = 5x^2 + 6x + 7$. X_1 = [1, 3, 5, 7] and X_2 = [2, 4, 6, 8] are any positive integers obtained randomly, obtain Y_1 = [6, 1, 4, 15] and Y_2 = [5, 9, 2, 1] by calculating the polynomials $f_1(x)$ and $f_2(x)$. After subtracting 1 from X_1, X_2, Y_1, Y_2, obtain X_1 = [X_1^1, X_1^2, X_1^3, X_1^4] = [0, 2, 4, 6], X_2 = [X_2^1, X_2^2, X_2^3, X_2^4] = [1, 3, 5, 7], Y_1 = [Y_1^1, Y_1^2, Y_1^3, Y_1^4] = [5, 0, 3, 14] and Y_2 = [Y_2^1, Y_2^2, Y_2^3, Y_2^4] = [4, 8, 1, 0]. X_1, X_2, Y_1, Y_2 are converted to binary and embedded into the bitmap of the smooth blocks of the compressed image.

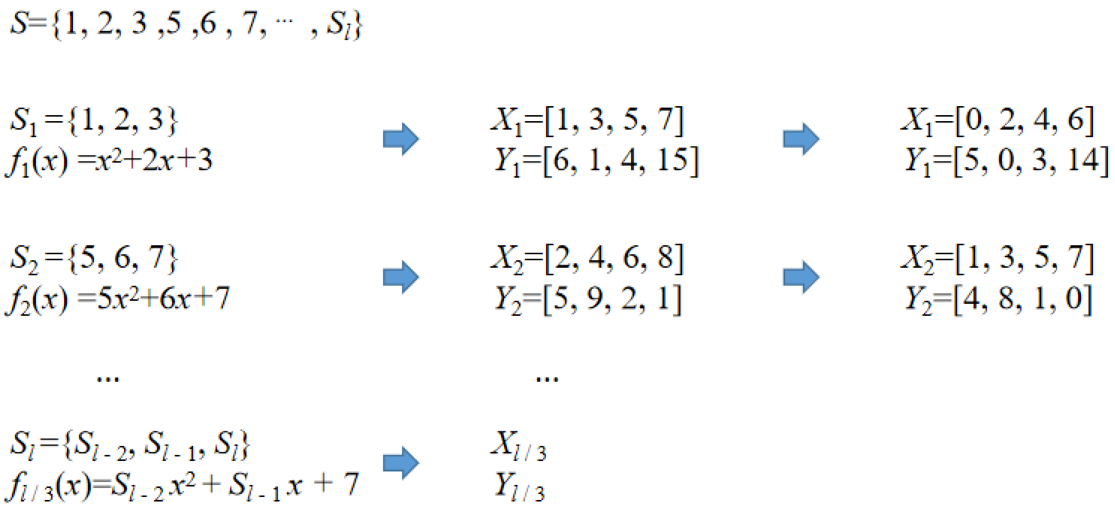
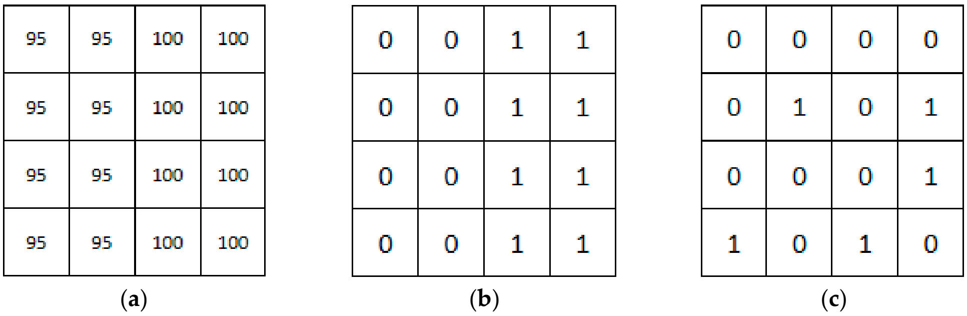


Figure 3. Example for secret sharing stage.

3.1.2. Subsecret Embedding Stage

The main idea of the DBS data hiding method is to directly replace the bitmap compressed by the smooth block with the secret message. If the difference between the high amount and the low amount is less than the preset threshold T, this block is considered as smooth block. During decoding, the change of the bitmap does not have a significant impact on the visual quality of the entire image due to the closeness of the high and low gray levels.

The binary of the first elements in X_1, Y_1, X_2, Y_2 are {$X_1{}^1$, $Y_1{}^1$, $X_2{}^1$, $Y_2{}^1$} = {[0000], [0101], [0001], [1010]}, respectively. Set the threshold value to T = 10. Figure 4a is an original image block; the triplet of this image block after AMBTC compression is {100, 95, BM}, BM is shown in Figure 4b and H − L = 100 − 95 = 5 < T. Replace the bitmap BM with {$X_1{}^1$, $Y_1{}^1$, $X_2{}^1$, $Y_2{}^1$,}, and the modified triplet is {100, 95, [0000 0101 0001 1010]}, as shown in Figure 4c. The 2nd, 3rd and 4th elements of X_1, Y_1, X_2, Y_2 are embedded in the compressed bitmaps of the four images in the same way, and so on, embedding all X and Y dimensions into n images, respectively.



95	95	100	100
95	95	100	100
95	95	100	100
95	95	100	100

(**a**)

0	0	1	1
0	0	1	1
0	0	1	1
0	0	1	1

(**b**)

0	0	0	0
0	1	0	1
0	0	0	1
1	0	1	0

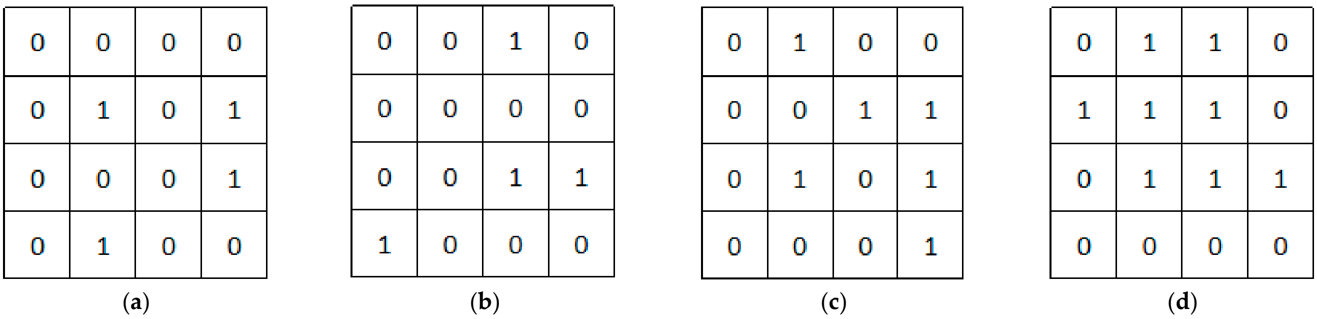
(**c**)

Figure 4. Example for subsecret embedding stage. (**a**) Cover image. (**b**) Bitmap before embedding. (**c**) Embedded bitmap.

3.2. Secret Message Extraction Stage

In the subsecret message extraction phase, the user first downloads at least m = 3 triple representations of covert images containing subsecrets from different servers. First, judge whether the difference between the H and L of the triplet is less than the preset threshold T = 10. If H − L < T, take out the binary subsecret message in the bitmap in this triplet.

Figure 5 shows 4 bitmaps of triples containing subsecrets, obtained from 4 shadow images stored on different servers. All bits are extracted from these 4 bitmaps, and then every 4 bits are converted into decimal, for example. As shown in Figure 5a, the extracted 16 bits is [0000 0101 0001 0100], which is [$X_1{}^1$, $Y_1{}^1$, $X_2{}^1$, $Y_2{}^1$] = {0, 5, 1, 4] after conversion to decimal; the decimal numbers extracted from the remaining 3 bitmaps are [$X_1{}^2$, $Y_1{}^2$, $X_2{}^2$, $Y_2{}^2$] = [2, 0, 3, 8], [$X_1{}^3$, $Y_1{}^3$, $X_2{}^3$, $Y_2{}^3$] = [4, 3, 5, 1], [$X_1{}^4$, $Y_1{}^4$, $X_2{}^4$, $Y_2{}^4$] = [6, 14, 7, 0]. After 1 is added to all elements, the four points {(1, 6), (3, 1), (5, 4), (7, 15)} on the $f_1(x)$ are obtained, while the four points {(2, 5), (4, 9), (6, 2), (8, 1)} on the $f_2(x)$ are obtained. Take any three points on the two functions and bring them into the Equation (1) to restore the coefficients; that is, the embedded secret numbers are {1, 2, 3} and {5, 6, 7}, respectively.



0	0	0	0
0	1	0	1
0	0	0	1
0	1	0	0

(**a**)

0	0	1	0
0	0	0	0
0	0	1	1
1	0	0	0

(**b**)

0	1	0	0
0	0	1	1
0	1	0	1
0	0	0	1

(**c**)

0	1	1	0
1	1	1	0
0	1	1	1
0	0	0	0

(**d**)

Figure 5. The bitmaps from four shadow images with embedded secret messages. (**a**) Bitmap 1. (**b**) Bitmap 2. (**c**) Bitmap 3. (**d**) Bitmap 4.

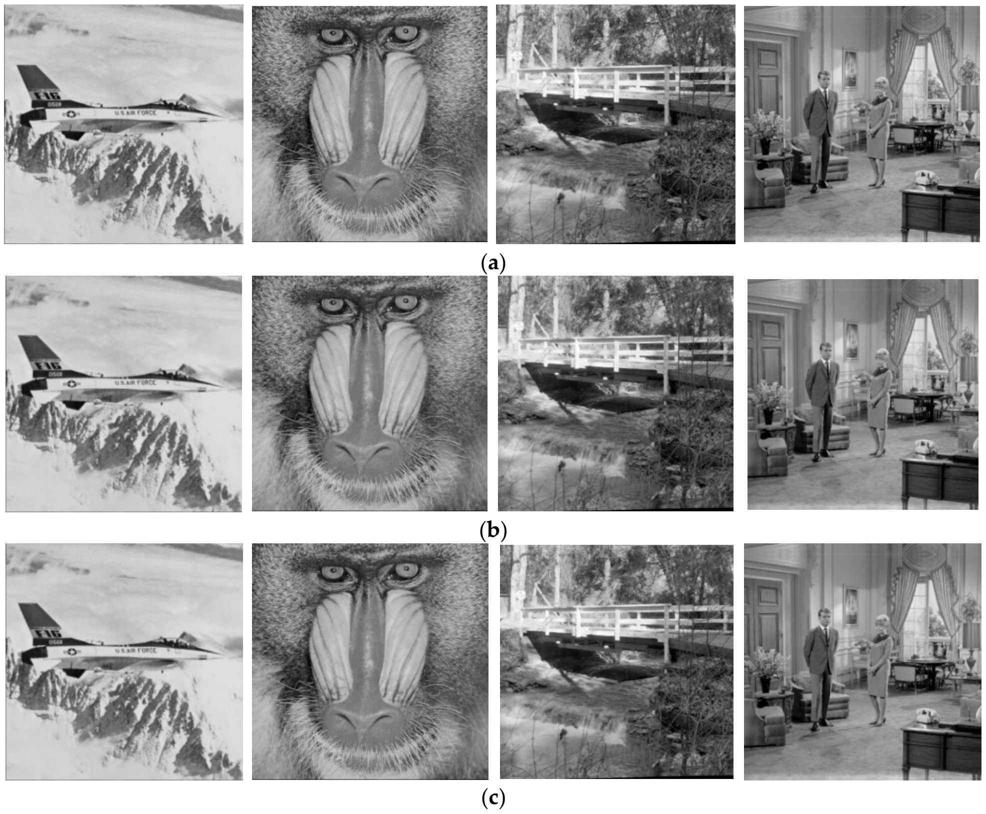
Finally, the n shadow images embedded with the subsecret messages are stored on different servers, respectively. In our method, only m shadow images are needful to extract the original secret message. The remaining $(n - m)$ shadow images do not affect the extraction of the secret message.

4. Experimental Results

At present, the performance metrics of image-based data hiding are mainly the Peak Signal to Noise Ratio (PSNR), Embedding Capacity (EC) and Embedding Rate (ER).

The PSNR measures the distortion between the stego image (SI) and the original image (OI). The EC is the total number of bits of the secret message embedded in the cover image (CI). The ER is the ratio between the EC and the total number of pixels of the CI. There is conflict between the two indicators, namely, the PSNR and EC; that is, embedding more secret data often leads to aggravating image distortion. An excellent data hiding algorithm should ensure that there is no obvious visual difference between the SI and the CI, and try to embed a large amount of the secret message.

The size of the images used in the experiments are standard grayscale images of 512 × 512, and the secret message is a randomly generated binary sequence. Figure 6 shows the original image (OI), the cover image (CI) after AMBTC compression, and the shadow image (SI) after embedding the secret message. There is almost no visual difference between the three sets of images. Table 1 shows the comparison of the PSNR between the OI and the CI, the OI and the SI, and the CI and the SI, under the threshold T = 10. Figure 7 and Table 2 show the PSNR of the covert image and the original image, the PSNR of the shadow image and the cover image, and the comparison of the embedding capacity under different thresholds, T = 5, T = 10, T = 15 and T = 20. It can be seen from the table that with the increase in T, there is only a slight decrease in the PSNROI-SI. As can be seen from Table 1, the PSNROI-CI is very close to the PSNROI-SI, indicating that our embedding method has little effect on AMBTC decompressed images.



(**a**)

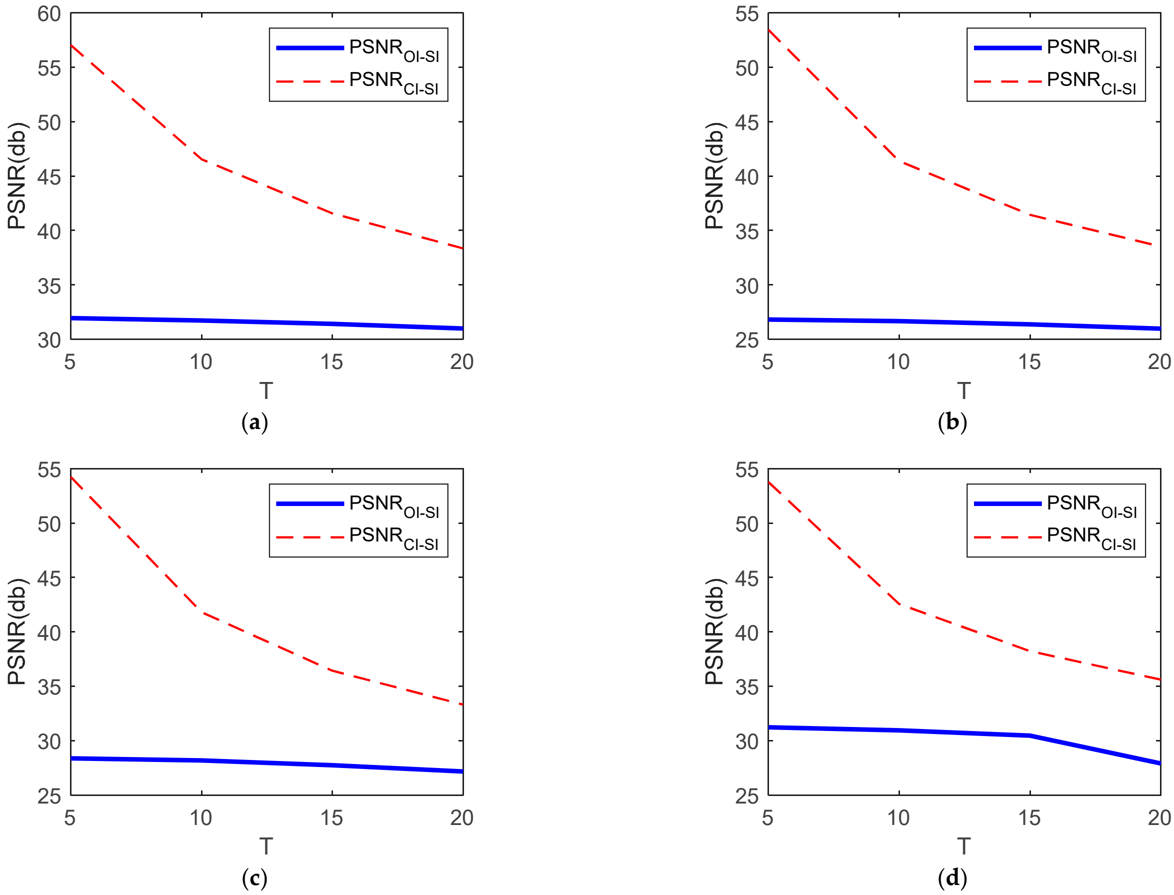
(**b**)

(**c**)

Figure 6. Visual comparison of original image, cover image and shadow image. (**a**) Original Image, OI. (**b**) Cover Image, CI. (**c**) Shadow Image, SI.

Table 1. Comparison of EC and PSNR of original image, cover image and shadow image.

Image	PSNROI-CI	PSNROI-SI	PSNRCI-SI	EC
Airplane	31.865	31.7193	46.5513	63,768
Baboon	26.8191	26.6704	41.3761	
Bridge	28.3827	28.1893	41.8135	
Man	31.2594	30.9492	42.5611	



(**a**)

(**b**)

(**c**)

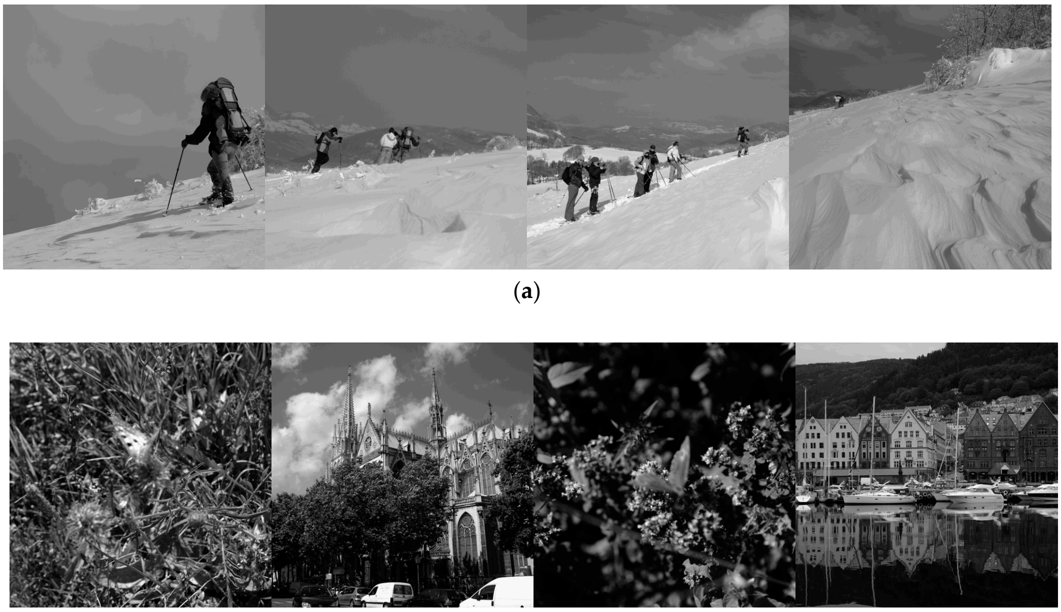
(**d**)

Figure 7. PSNR comparison of four test images under different thresholds. (**a**) Airplane. (**b**) Baboon. (**c**) Bridge. (**d**) Man.

Figure 8 shows two sets of images with different smoothness; the first set of images has more embeddable blocks. The method proposed in this paper chooses to embed the subsecret messages segmented by the Shamir threshold scheme into the encoding of AMBTC of multiple different images. Since complex images have fewer embeddable blocks, it can be seen from Table 3 that there is a large difference in the amount of information embedded between smoother images and complex images, under the same threshold. The method proposed in this paper is more suitable for using smooth images as the cover image.

Table 2. Comparison of EC and PSNR under different thresholds.

T	Image	PSNROI-SI	PSNRCI-SI	EC
T = 5	Airplane	31.852	57.0449	12,000
	Baboon	26.8096	53.4486	
	Bridge	28.3714	54.2450	
	Man	31.2343	53.7736	
T = 10	Airplane	31.7193	46.5513	63,768
	Baboon	26.6704	41.3761	
	Bridge	28.1893	41.8135	
	Man	30.9492	42.5611	
T = 15	Airplane	31.424	41.5739	111,216
	Baboon	26.3699	36.4424	
	Bridge	27.7495	36.4353	
	Man	30.4662	38.2309	
T = 20	Airplane	30.9865	38.3526	145,632
	Baboon	25.9763	33.5268	
	Bridge	27.1726	33.3167	
	Man	27.9083	35.6121	



(**a**)

(**b**)

Figure 8. Two sets of images with different smoothness. (**a**) smooth image. (**b**) complex image.

Table 3. Comparison of EC and PSNR under different smoothness.

T	Smooth Image					Complex Image				
	PSNRa	PSNRb	PSNRc	PSNRd	EC	PSNRe	PSNRf	PSNRg	PSNRh	EC
10	33.5	32.6	35.6	34.8	309,600	29.6	30.7	28.1	27.4	39,528
15	32.6	32.1	35.1	33.9	342,888	29.3	30.4	27.8	27.1	89,712
20	31.8	31.6	34.6	33.2	356,832	28.8	29.7	27.5	26.4	135,912
25	31.1	31.2	34.1	32.6	362,688	28.2	29.1	26.9	25.6	180,576

Table 4 shows the comparison results. Columns 6, 7 and 8 in the table give the embedding rate, storage volume and the PSNR of the shadow images, respectively. The methods in [20–25] are dual-image data hiding methods, which generate two visually similar shadow images. As shown in Column 8, the PSNRs of the two shadow images generated by [22,23] are highly different. Although the shadow images are meaningful, they are not suitable for practical application scenarios. Many shadow images in [26–28] are meaningless ciphertext images. Thus, similar to traditional encryption methods, they also easily attract attackers' attention. The method proposed in [27] generates multiple meaningless shadow images of the same size by secret sharing of a cover image, and embeds the secret message in the shadow image, while the method in this paper uses multiple carrier images. Therefore, the embedding rate of [27] is higher than that of the method proposed in this paper, and the method proposed in this paper takes the smooth image as the carrier and its storage capacity is much higher than that of [27]. The embedding payload of [28] is bounded by the embedding method. When the PSNR ≈ 35 dB, the maximum payload of [28] is around 0.65 bpp. In addition, the compression ratio of the method proposed in this paper is four, which means that the embedding capacity of our method has obvious advantages under the same storage space.

Table 4. Comparison results.

Method	Does Shadow Image Make Sense?	Number of Shadow Images	Shadow Images Are Visually Similar?	Bases of Secret Message	ER	EC	PSNR	Compression Ratio
[20]	Y	2	Y	5	1	262,144	45.1/45.1	1
[21]	Y	2	Y	5	1.55	406,323	39.9/39.9	1
[22]	Y	2	Y	5	1.16	304,087	41.3/52.1	1
[23]	Y	2	Y	8	/	/	51.2/45.7	1
[24]	Y	2	Y	25	1.14	298,844	49.9/49.9	1
[25]	Y	2	Y	5	1.56	408,944	43.0/43.0	1
[26]	N	N	N	2	/	/	/	1
[27]	N	N	N	2	0.5	130,989	/	1
[28]	N	N	N	2	0.65	/	/	1
Ours	Y	N	N	16	0.46	362,688	/	4

5. Conclusions and Future Works

This paper proposes a data hiding method of multicompressed image based on secret sharing, which divides the original secret message into multiple subsecret messages through the method of secret sharing, and then embeds the subsecret messages into the AMBTC compression code. The proposed method can avoid the problem faced by current mainstream data hiding, based on secret sharing attracting attackers' attention during the transmission process. In our method, the shadow images make sense while they are not visually similar. The compression ratio is four, so the embedding capacity of our method has an obvious advantage under the same storage space. However, the embedding performance of our method is limited by the smoothness of the carrier image. In future work, we will try to improve the method in terms of embedding performance, visual loss and storage cost.

Author Contributions: Conceptualization, H.K. and L.L.; methodology, H.K. and L.L.; software, H.K.; validation, L.L. and B.-G.K.; formal analysis, L.L. and B.-G.K.; investigation, L.L.; resources, B.-G.K.; data curation, B.-G.K.; writing—original draft preparation, H.K. and L.L.; writing—review and editing, B.-G.K.; visualization, H.K.; supervision, L.L.; project administration, L.L.; funding acquisition, B.-G.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 61866028, and the Technology Innovation Guidance Program Project (Special Project of Technology Cooperation, Science and Technology Department of Jiangxi Province), grant number 20212BDH81003.

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

References

1. Kim, C. Separable Reversible data hiding in encrypted AMBTC images using Hamming code. *Appl. Sci.* **2022**, *12*, 8225. [CrossRef]
2. Nguyen, T.S. Reversible Data Hiding Scheme Based on Coefficient Pair Mapping for Videos H. 264/AVC without Distortion Drift. *Symmetry* **2022**, *14*, 1768. [CrossRef]

3. Hung, C.C.; Lin, C.C.; Wu, H.C.; Lin, H.C. A study on reversible data hiding technique based on three-dimensional prediction-error histogram modification and a multilayer perceptron. *Appl. Sci.* **2022**, *12*, 2502. [CrossRef]
4. Megías, D.; Mazurczyk, W.; Kuribayashi, M. Data Hiding and Its Applications: Digital Watermarking and Steganography. *Appl. Sci.* **2021**, *11*, 10928. [CrossRef]
5. Qu, X.; Kim, H.J. Pixel-based pixel value ordering predictor for high-fifidelity reversible data hiding. *Signal Process.* **2015**, *111*, 249–260. [CrossRef]
6. Su, W.; Wang, X.; Li, F.; Shen, Y.; Pei, Q. Reversible data hiding using the dynamic block-partition strategy and pixel-value-ordering. *Multimed. Tools Appl.* **2019**, *78*, 7927–7945. [CrossRef]
7. Weng, S.; Shi, Y.; Hong, W.; Yao, Y. Dynamic improved pixel value ordering reversible data hiding. *Inf. Sci.* **2019**, *489*, 136–154. [CrossRef]
8. Di, F.; Zhang, M.; Liao, X.; Liu, J. High-fidelity reversible data hiding by quadtree-based pixel value ordering. *Multimed. Tools Appl.* **2019**, *78*, 7125–7141. [CrossRef]
9. Wu, H.; Li, X.; Zhao, Y.; Ni, R. Improved ppvo-based high-fidelity reversible data hiding. *Signal Process.* **2020**, *167*, 1072640. [CrossRef]
10. Zhang, X. Reversible data hiding in encrypted image. *IEEE Signal Process. Lett.* **2011**, *18*, 255–258. [CrossRef]
11. Puteaux, P.; Puech, W. An efficient msb predictionbased method for high-capacity reversible data hiding in encrypted images. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1670–1681. [CrossRef]
12. Delp, E.; Mitchell, O. Image compression using block truncation coding. *IEEE Trans. Commun.* **1979**, *27*, 1335–1342. [CrossRef]
13. Lema, M.; Mitchell, O. Absolute moment block truncation coding and its application to color images. *IEEE Trans. Commun.* **1984**, *32*, 1148–1157. [CrossRef]
14. Kumar, R.; Singh, S.; Jung, K. Human visual system based enhanced ambtc for color image compression using interpolation. In Proceedings of the 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 7–8 March 2019; pp. 903–907.
15. Hong, W.; Chen, T.; Shiu, C. Lossless steganography for AMBTC-compressed images. In Proceedings of the 2008 Congress on Image and Signal Processing, Sanya, China, 27–30 May 2008; Volume 2, pp. 13–17.
16. Chan, C.; Cheng, L. Hiding data in images by simple lsb substitution. *Pattern Recognit.* **2004**, *37*, 469–474. [CrossRef]
17. Solak, S. High embedding capacity data hiding technique based on EMSD and LSB substitution algorithms. *IEEE Access* **2020**, *8*, 166513–166524. [CrossRef]
18. Mielikainen, J. LSB matching revisited. *IEEE Signal Process. Lett.* **2006**, *13*, 285–287. [CrossRef]
19. Gutub, A. Watermarking images via counting-based secret sharing for lightweight semi-complete authentication. *Int. J. Inf. Secur. Priv.* **2022**, *16*, 1–18. [CrossRef]
20. Chang, C.C.; Kieu, T.D.; Chou, Y.C. Reversible data hiding scheme using two steganographic images. In Proceedings of the TENCON 2007 IEEE Region 10 Conference, Taipei, Taiwan, 30 October–2 November 2007; pp. 1–4.
21. Chang, C.C.; Lu, T.C.; Horng, G.; Huang, Y.H.; Hsu, Y.M. A high payload data embedding scheme using dual stego-images with reversibility. In Proceedings of the 2013 9th International Conference on Information, Communications & Signal Processing, Tainan, Taiwan, 10–13 December 2013; pp. 1–5.
22. Qin, C.; Chang, C.C.; Hsu, T.J. Reversible data hiding scheme based on exploiting modification direction with two steganographic images. *Multimed. Tools Appl.* **2015**, *74*, 5861–5872. [CrossRef]
23. Liu, Y.; Chang, C.C. A turtle shell-based visual secret sharing scheme with reversibility and authentication. *Multimed. Tools Appl.* **2018**, *77*, 25295–25310. [CrossRef]
24. Chen, X.; Guo, W. Reversible data hiding scheme based on fully exploiting the orientation combinations of dual stego-images. *Int. J. Netw. Secur.* **2020**, *22*, 126–135.
25. Chen, X.; Hong, C. An efficient dual-image reversible data hiding scheme based on exploiting modification direction. *J. Inf. Secur. Appl.* **2021**, *58*, 102702. [CrossRef]
26. Wu, X.; Weng, J.; Yan, W.Q. Adopting secret sharing for reversible data hiding in encrypted images. *Signal Process.* **2018**, *143*, 269–281. [CrossRef]
27. Zhou, N.; Zhang, M.; Liu, M. Reversible data hiding algorithm in homomorphic encrypted image based on secret sharing. *Sci. Technol. Eng.* **2020**, *20*, 7780–7786.
28. Xiong, L.; Han, X.; Yang, C.N.; Xia, Z. RDH-DES: Reversible Data Hiding over Distributed Encrypted-Image Servers based on Secret Sharing. *ACM Trans. Multimid. Comput. Commun. Appl.* **2022**, *in press*.