Journal of
Imaging

MDPI

Article

Progressive Secret Sharing with Adaptive Priority and Perfect Reconstruction

Heri Prasetyo [1,*], Chih-Hsien Hsia [2,*] and Alim Wicaksono Hari Prayuda [1]

[1] Department of Informatics, Universitas Sebelas Maret (UNS), Surakarta 57126, Indonesia; wicayudha.wy@gmail.com

[2] Department of Computer Science and Information Engineering, National Ilan University, Yilan 26047, Taiwan

* Correspondence: heri.prasetyo@staff.uns.ac.id (H.P.); chhsia625@gmail.com (C.-H.H.)

Abstract: A new technique for progressive visual secret sharing (PVSS) with adaptive priority weight is proposed in this paper. This approach employs the bitwise and eXclusive-OR (XOR) based approaches for generating a set of shared images from a single secret image. It effectively overcomes the former scheme limitation on dealing with an odd number of stacked or collected shared images in the recovery process. The presented technique works well when the number of stacked shared images is odd or even. As documented in experimental results, the proposed method offers good results over binary, grayscale, and color images with a perfectly reconstructed secret image. In addition, the performance of the proposed method is also supported with theoretical analysis showing its lossless ability to recover the secret image. However, it can be considered as a strong substitutive candidate for implementing a PVSS system.

Keywords: adaptive priority weight; eXclusive-OR; lossless; odd number; progressive; secret sharing

check for updates

Citation: Prasetyo, H.; Hsia, C.-H.; Wicaksono Hari Prayuda, A. Progressive Secret Sharing with Adaptive Priority and Perfect Reconstruction. *J. Imaging* **2021**, *7*, 70. https://doi.org/10.3390/jimaging7040070

Academic Editor: Alessandro Piva

Received: 31 December 2020
Accepted: 30 March 2021
Published: 3 April 2021

1. Introduction

Recently, several approaches have been devoted to dealing with secure image communication. Transferring a secret image via a communication channel has become an open issue nowadays. Two parties often communicate one to the one another via Internet, cloud computing, communication technology, etc. In this way, a digital image is often transmitted or sent via the communication channels with the security and integrity requirements. A simple means for transferring or exchanging secret information between two or multiple parties is by inserting the secret image into the digital cover image. One can select an appropriate technique for transmitting a secret image. Among of them are the secret sharing technique [1–3], the image watermarking technique [4], the multiple secret sharing technique [5–7], the progressive secret sharing technique [8–15], the lossless progressive secret image technique [16], etc. These aforementioned methods offer promising performances on in regards to rendering the information of the secret image into the other form (such as a digital image host or shared images).

The secret sharing method aims to convert a meaningful secret image into a non-friendly appearance before transmission to the other parties. The noise-like form can be selected to hide the content of secret image such that an unauthorized malicious attacker cannot recognize the important information contained in the secret image. In recent years, a lot of methods have been developed in the field of secret sharing. The most well-known secret image methods are multiple secret sharing [5–7], progressive secret sharing [8–15], lossless progressive secret sharing [16], and more sophisticated secret sharing techniques. The multiple secret sharing method [5–7] changes a set of secret images into multiple images or a set of shared images, whereas the progressive secret sharing method simply converts a single secret image into a set of shared images. In the multiple secret sharing method, all shared images are required to reconstruct the secret image. If only a partial subset of shared images is involved in the recovery process, one obtains nothing. The progressive secret sharing method [8–15] offers different ways to reconstruct the secret image. Either a partial or a full set of shared images may be used to obtain the recovered secret image. A higher number of involved shared images gives a better quality of the recovered secret image, and vice versa. However, the progressive secret sharing method cannot give warranty in the lossless recovered secret image. However, a new technique (namely, lossless progressive secret sharing [16]) is able to recover the secret image with lossless quality. Some modifications have been made to the progressive visual secret sharing (PVSS) method with the adaptive priority weight [15]. We give illustrations of PVSS and PVSS with adaptive priority weight in the following example: Figure 1 displays a Lena image in color format. Figure 2 shows a set of shared images generated from the PVSS method [16] and the PVSS scheme with adaptive priority weight [15]. Figure 3 gives the reconstructed secret image by stacking several images obtained from the PVSS method [16] and the PVSS scheme with adaptive priority weight [15]. The adaptive priority weight offers better results regarding the quality of the reconstructed secret images.

Figure 1. Lena image in color formatting.

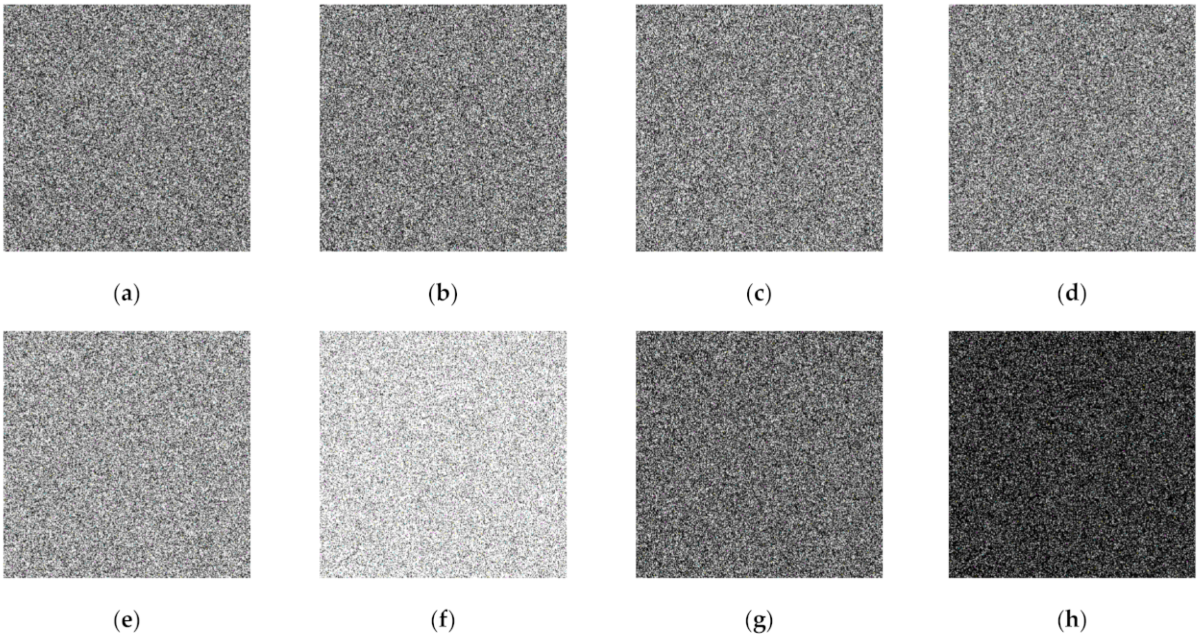
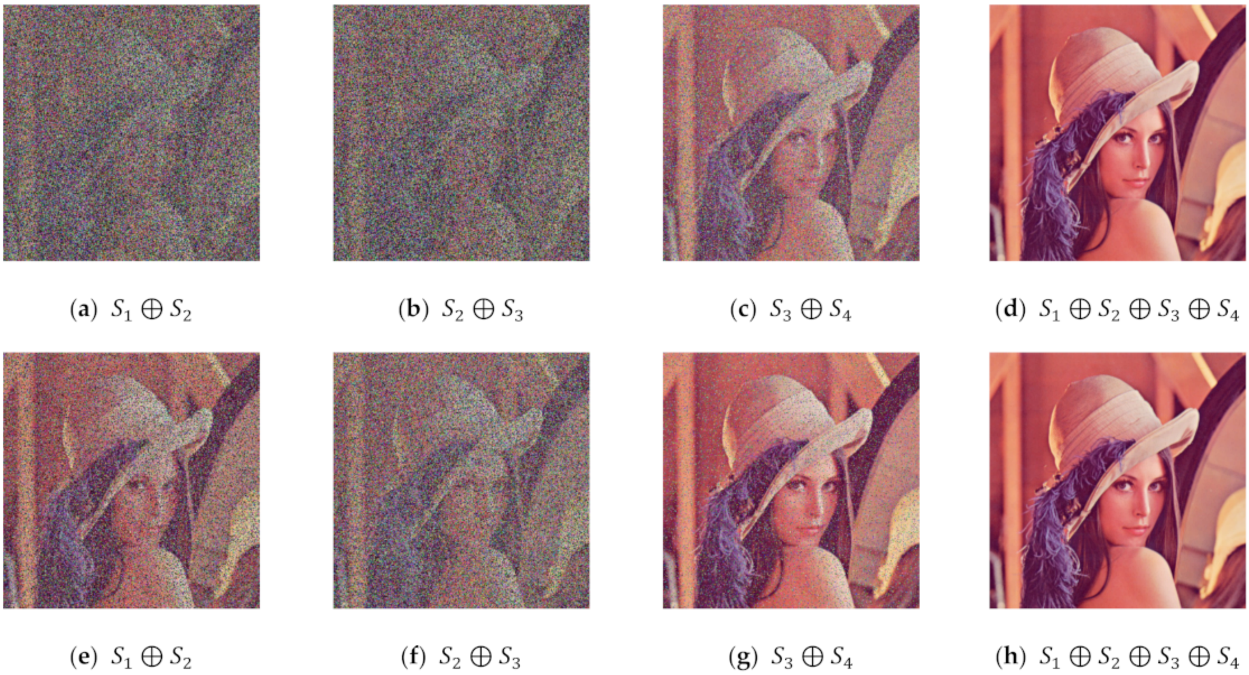


Figure 2. A set of generated shared images: (**a**–**d**)$\{S_1, S_2, S_3 S_4\}$ from the progressive visual secret sharing (PVSS) scheme [16], and (**e**–**h**) $\{S_1, S_2, S_3 S_4\}$ from a PVSS scheme with adaptive priority weight [15].

This paper first reviews the former scheme [15] on generating a set of shared images from a secret image and recovering a secret image. The former scheme employs the adaptive priority weight to progressively reconstruct a secret image. The former scheme shows its usability in the PVSS task as reported in [15]. Based on our observation, however, the former scheme suffers from a slight limitation in the secret image recovery process if the number of stacked or collected images is odd. This paper delineates this limitation, along with the theoretical analysis required to prove this issue. Some experiments concerning this limitation are also reported. Thereafter, we develop a new technique for overcoming this problem. This new technique inherits the PVSS with a random grids approach from the former scheme with a simple modification implemented to improve the recovery process when the number of stacked images is odd. This simple modification effectively yields a perfect reconstructed secret image whether the number of stacked images is odd or even. The correctness of the proposed method is also supported by mathematical analysis as well as experimental findings. The proposed PVSS method can be touted as a strong alternative candidate for substitution in place of the former scheme [15].



(**a**) $S_1 \oplus S_2$ (**b**) $S_2 \oplus S_3$ (**c**) $S_3 \oplus S_4$ (**d**) $S_1 \oplus S_2 \oplus S_3 \oplus S_4$

(**e**) $S_1 \oplus S_2$ (**f**) $S_2 \oplus S_3$ (**g**) $S_3 \oplus S_4$ (**h**) $S_1 \oplus S_2 \oplus S_3 \oplus S_4$

Figure 3. The quality of the reconstructed secret images obtained from: (**a**–**d**) the PVSS scheme [16] and (**e**–**h**) the PVSS method with adaptive priority weight [15].

The former scheme [15] and proposed method exploit the eXclusive-OR (XOR) operation for producing a set of shared images as well as obtaining a recovered secret image. The XOR operation is very simple, with various symmetric properties. These XOR properties are very important for designing the PVSS algorithm. One can produce a set of generated shared images using an XOR operation. Conversely, the XOR operation can be used to recover a secret image without the use of any additional computational techniques. The XOR performs differently with common arithmetic operations. The XOR operator has no negation/inverse operation, while the arithmetic has a negation/inverse operator. For example, the arithmetics addition operator owns the negation operator (i.e., arithmetics subtraction). What follows are some useful XOR properties [7] for the PVSS methods. Herein, we provide examples of each property with both the binary and the decimal number representation.

- **Identity:**

This property indicates that performing XOR over any arbitrary number with zero yields an identical arbitrary number itself. This property is defined as:

$$A \oplus 0 = A \tag{1}$$

For example, we have $A = (6)_{10}$ (i.e., a number 6 in decimal representation) with the corresponding binary string set as $(110)_2$. This property tells us that $(110)_2 \oplus (000)_2 = (110)_2$ in binary representation or $(6)_{10} \oplus (0)_{10} = (6)_{10}$ in decimal format.

- **Performing XOR over "odd number" times:**

If we perform XOR operation on the same arbitrary number over "odd number" times, we will receive this arbitrary number itself. Specifically, this process is denoted as:

$$\underbrace{A \oplus A \oplus \ldots \oplus A}_{n \text{ is odd number}} = A \oplus (A \oplus A) = A \oplus 0 = A \tag{2}$$

For example, XOR-ing the decimal number 6 over "odd" times gives $\underbrace{6 \oplus 6 \oplus \ldots \oplus 6}_{n \text{ is odd number}} = 6 \oplus (6 \oplus 6) = 6 \oplus 0 = 6$ or, in binary representation, $\underbrace{(110)_2 \oplus (110)_2 \oplus \ldots \oplus (110)_2}_{n \text{ is even number}} = (110)_2 \oplus \{(110)_2 \oplus (110)_2\} = (110)_2 \oplus (000)_2 = (110)_2$.

- **Performing XOR over "even number" times:**

In contrast to the previous XOR property, performing an XOR operation on the same arbitrary number over "even number" times yields zero results. This property is shown as follows:

$$\underbrace{A \oplus A \oplus \ldots \oplus A}_{n \text{ is even number}} = A \oplus A = 0 \tag{3}$$

A simple example for this property is in the case $\underbrace{6 \oplus 6 \oplus \ldots \oplus 6}_{n \text{ is even number}} = 6 \oplus 6 = 0$ in decimal number representation, or, in binary representation, $\underbrace{(110)_2 \oplus (110)_2 \oplus \ldots \oplus (110)_2}_{n \text{ is even number}} = (110)_2 \oplus (110)_2 = (000)_2$.

- **Symmetric Inverse:**

This property is almost similar to the common arithmetic inverse, e.g., an addition operator against the subtraction operator. The XOR has no inverse operator. Yet, the XOR can solely perform symmetric inverse by itself. This property is defined as follows:

$$\text{If } A \oplus B = C, \text{ Then } B \oplus C = A. \tag{4}$$

Suppose that there are two decimal numbers $A = 6$ and $B = 3$, with the corresponding binary strings $A = (110)_2$ and $B = (011)_2$, respectively. Thus, we have $A \oplus B = 6 \oplus 3 = 5$ or $A \oplus B = (110)_2 \oplus (011)_2 = (101)_2$. It is implied that $C = 5$ or, in binary representation, $C = (110)_2$. Conversely, we obtain $B \oplus C = 3 \oplus 5 = 6$ or $B \oplus C = (011)_2 \oplus (101)_2 = (110)_2$. This value is identical to $A = 6$ or $(110)_2$, showing that the XOR has a unique symmetric inverse property.

- **Commutative:**

The XOR has a similar property as the common arithmetics (i.e., a commutative property). This property is specified as:

$$A \oplus B = B \oplus A \tag{5}$$

Let $A = 6$ and $B = 3$, or, in binary representation, $A = (110)_2$ and $B = (011)_2$. The following computations yield identical results (i.e., $A \oplus B = (110)_2 \oplus (011)_2 = (101)_2$ and $B \oplus A = (011)_2 \oplus (110)_2 = (101)_2$) or, in decimal representation, $6 \oplus 3 = 3 \oplus 6$.

- **Associative:**

This property herein is similar to that of the common arithmetic operation. The XOR operation also offers associative computation as formally defined as follows:

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C \quad (6)$$

Suppose that we have $A = 6$, $B = 3$, and $C = 5$. This property gives $A \oplus (B \oplus C) = (000)_2$ and $(A \oplus B) \oplus C = (000)_2$ in binary form. This result resembles as in the decimal form $6 \oplus (3 \oplus 5) = (6 \oplus 3) \oplus 5$. However, the XOR operation has been shown to have an associative property.

The main contribution of this paper is to develop a new technique for PVSS with a lossless ability in the secret image reconstruction process whether the number of stacked shared images is odd or even. The other contribution of this paper is a formal mathematical analysis for showing the limitation and correction of the PVSS method. This work has been motivated with the increased demand for secure image transmissions. This work has also motivated the success rate of lossless PVSS in [16]. The organization of this paper is arranged as follows: Section 2 briefly discusses the former PVSS scheme with adaptive priority [15]. This section also shows the limitation of the former scheme [15] as supported with experimental finding and theoretical analysis. Section 3 describes the proposed PVSS method, with adaptive priority analyzed in detail. The proposed method's usability is supported with the formal mathematical analysis. Section 4 extensively reports some experimental results. The end of this paper delivers the conclusions and direction for future works.

2. Former PVSS Scheme

This section briefly reviews the former PVSS scheme [15] wherein the secret and shared images are in binary format. It presents the step-by-step process of shared image generation and secret image reconstruction. A slight shortcoming of this aforementioned method is also provided in this section along with formal theoretical analysis.

2.1. PVSS Scheme for Binary Image

The PVSS scheme in [15] for binary image generation is first presented in this section. This scheme exploits the random grids to perform the secret sharing task. The former method converts a secret image in binary format into a set of shared binary images. The procedure of shared image generation is described as follows: Suppose that I is a binary secret image of size $M \times N$. Each pixel on I is denoted as $I(x, y)$, for $x = 1, 2, \ldots, M$ and $y = 1, 2, \ldots, N$. Since I is a binary image, then $I(x, y)$ simply consists of two values (i.e., a black (or 0) and white pixel (or 1)). The former scheme [15] changes I into n shared images $\{S_1, S_2, \ldots, S_n\}$ in pixel-based processing and in a bitwise fashion. The symbol S_i is the i-th shared image, for $i = 1, 2, \ldots, n$. These sets (i.e., shared images) are further transferred to the receiver or decoder side via communication or transmission channel. Each pixel in the shared image S_i is denoted as $S_i(x, y)$, for $x = 1, 2, \ldots, M$ and $y = 1, 2, \ldots, N$. Herein, the size of each shared image is identical to that of the size of the secret image (i.e., $M \times N$).

In order to convert a secret image into a set of shared images, the former scheme [15] requires a set of spatial pixel locations ($\updownarrow_j$) for $j = 1, 2, \ldots, n$. These locations are specified by their adaptive priority weight (w_j) for $j = 1, 2, \ldots, n$. The adaptive priority weight can be predetermined based on user preference. Figure 4 exhibits some examples of spatial pixel location ($\updownarrow_j$) for $j = 1, 2, \ldots, 4$, over various adaptive priority weights $\{w_1 = 0.5, w_2 = 0.3, w_3 = 0.1, w_4 = 0.1\}$. In this figure, the pixel location $\updownarrow_1$ owns around 50% of the occupied pixels in order to generate a shared image. The $\updownarrow_1$ has a higher adaptive weight compared to the other spatial pixel locations. In this example, the additional pixels in $\updownarrow_1$ are caused by the rounding operator in the adaptive priority weight determining the spatial pixel location. Setting a higher priority weight w_j indicates more pixel locations $\updownarrow_j$ arranged in the j-th shared image, and vice versa. Thus, the recovered secret image becomes quickly or easily recognized by human vision in the reconstruction process. In addition, the correct pixel of the recovered secret image will be rapidly obtained by utilizing a higher priority weight.

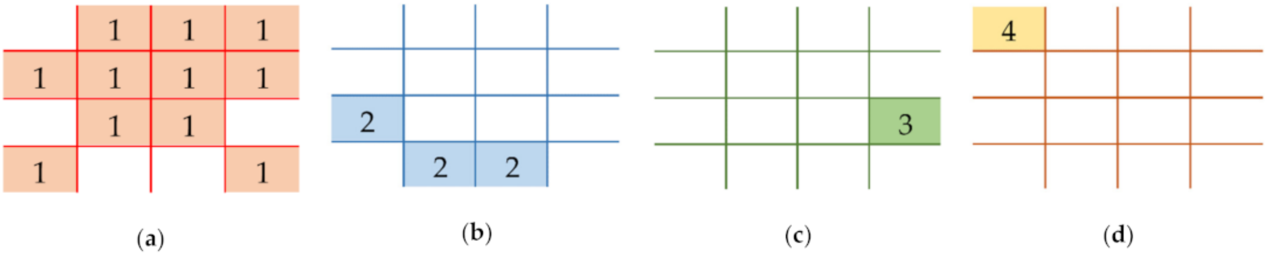


Figure 4. Examples of spatial pixel locations with various adaptive priority weights $\{w_1 = 0.5, w_2 = 0.3, w_3 = 0.1, w_4 = 0.1\}$: **(a)** $\updownarrow_1$, **(b)** $\updownarrow_2$, **(c)** $\updownarrow_3$, and **(d)** $\updownarrow_4$.

The generation of shared images is formally defined as follows: For each pixel in the secret image (i.e., $I(x,y)$) we first must determine two selected indices of shared images for the purpose of encoding. The first and second indices of the selected shared images are denoted as r_1 and r_2. The value of r_1 is determined based on the information of the pixel location $\updownarrow_j$, whereas r_2 is simply computed as $r_2 \leftarrow \text{mod}\{r_1, n\} + 1$. The symbols $\text{mod}\{\cdot,\cdot\}$ and $\leftarrow$ indicate the arithmetic modulus operator and the assignment operator, respectively. A pixel in a spatial position (x,y) in the first selected shared image is assigned the following value:

$$S_{r_1}(x,y) \leftarrow U_I(0,1), \tag{7}$$

where $S_{r_1}(x,y)$ represents the pixel in the spatial position (x,y) over the r_1-th shared image. The computation in Equation (7) requires a random number generator. Herein, $U_I(0,1)$ denotes the uniformly random number generator producing the integer in range $[0,1]$. A different strategy is then applied to determine the pixel value of the second selected shared image $S_{r_2}(x,y)$. The pixel in the secret image specifies the value of $S_{r_2}(x,y)$. If the processed secret image is a black pixel (i.e., $I(x,y) = 0$), then the pixel value in $S_{r_2}(x,y)$ is assigned as:

$$S_{r_2}(x,y) \leftarrow S_{r_1}(x,y). \tag{8}$$

If the secret image is a white pixel (i.e., $I(x,y) = 1$), the former scheme provides the pixel $S_{r_2}(x,y)$ as follows:

$$S_{r_2}(x,y) \leftarrow \sim S_{r_1}(x,y), \tag{9}$$

where $\sim$ is the bitwise NOT operator. Subsequently, the pixel values of all shared images excluding S_{r_1} and S_{r_2} (i.e., all S_i for $1 \le i \le n$ and $i \ne r_1, r_2$) are set as $S_{r_2}(x,y)$:

$$S_i(x,y) \leftarrow S_{r_2}(x,y). \tag{10}$$

This process is conducted for all pixels in the secret image I. A set of shared images $\{S_1, S_2, \ldots, S_n\}$ is created at the end of this process. The shared image is in binary format if the binary secret image is fed into the former scheme [15]. Algorithm 1 explains the shared image generation process in detail.

Algorithm 1: Former Scheme [15].

```
Input: Secret image in binary format, $I$, of size $M \times N$
Number of shared images, $n$
Output: A set of generated shared images, $\{S_1, S_2, \ldots, S_n\}$, each of size $M \times N$

Step 1: Based on priority weight $w_j$, determine the location set $\updownarrow_j$, for $j = 1, 2, \ldots, n$.
Step 2: For Each Pixel $(x,y)$. Based on information $\updownarrow_j$, select two shared images $r_1$ and $r_2$. Do
Step 3: $S_{r_1}(x,y) \leftarrow U_I(0,1)$
Step 4: If $I(x,y) = 0$, then $S_{r_2}(x,y) \leftarrow S_{r_1}(x,y)$
Step 5: Else $S_{r_2}(x,y) \leftarrow\sim S_{r_1}(x,y)$
Step 6: For Each other shared images, $S_i$, with condition $1 \le i \le n$ and $i \ne r_1, r_2$ Do
Step 7: $S_i(x,y) \leftarrow S_{r_2}(x,y)$
Step 8: Obtain $n$ generated shared images, $\{S_1, S_2, \ldots, S_n\}$
```

In normal situations, all shared images are transmitted to the decoder or receiver side. However, the receiver often collects a partial set of shared images to reconstruct the secret image. Let $\{S_{t_1}, S_{t_2}, \ldots, S_{t_T}\}$ be a partial set of collected shared images in the receiver side, where $t_1, t_2, \ldots, t_T$ denotes the index of the received shared image and $T \leq n$. The reconstruction process of the secret image can be easily performed by stacking a partial set of collected shared images with the bitwise XOR-based operation. This process is descibed as follows:

$$\widetilde{I} = S_{t_1} \oplus S_{t_2} \oplus \ldots \oplus S_{t_T}, \tag{11}$$

where $\widetilde{I}$ is a recovered secret image. The quality can be improved if the PVSS scheme [15] involves more stacked shared images in the secret image reconstruction process. Hopefully, the quality of the recovered secret image would be as similar as possible to that of the original secret image by stacking all shared images with the XOR operator.

2.2. Limitation of PVSS Scheme

As reported in literature [15], the former PVSS scheme offers a promising result in the shared image generation and secret image reconstruction processes. The former scheme yields a correct reconstructed secret image if and only if the number of stacked shared images is even. However, it is little regrettable that the former PVSS scheme cannot restore the secret image if the number of stacked shared images is odd. The following theorem explains this limitation.

Theorem 1. *The former PVSS scheme [15] yields perfect or partial reconstruction if and only if the number of stacked shared images is even.*

Proof. *Let $\{S_{t_1}, S_{t_2}, \ldots, S_{t_T}\}$ be a set of collected or received shared images involved to reconstruct a secret image. In this proof, we investigate the quality of $\widetilde{I}$. The reconstruction process is performed with the XOR operation in a bitwise-based manner over all collected shared images. The reconstruction process is defined as:*

$$\widetilde{I} = S_{t_1} \oplus S_{t_2} \oplus \ldots \oplus S_{t_T},$$

If the receiver module collects all shared images, it is implied that $T = n$. The recovered secret image $\widetilde{I}$ can be obtained as:

$$\widetilde{I} = S_1 \oplus S_2 \oplus \ldots \oplus S_n. \tag{12}$$

There are two selected shared images (S_{r_1} and S_{r_2}) in the shared image generation. The other shared images have the value of S_{r_2} (i.e., $S_i \leftarrow S_{r_2}$ for $1 \leq i \leq n$ and $i \neq r_1, r_2$). For sake of simplicity, we remove the pixel position (x, y) for all proofs in this paper. Thus, the form in Equation (12) can be alternatively computed as:

$$\widetilde{I} = S_{r_2} \oplus S_{r_2} \oplus \ldots \oplus S_{r_1} \oplus \ldots \oplus S_{r_2} \oplus \ldots \oplus S_{r_2}$$

Arranging the S_{r_1} and S_{r_2} in an orderly fashion, we gain the following form:

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \oplus \underbrace{S_{r_2} \oplus S_{r_2} \oplus \ldots \oplus S_{r_2}}_{n-2}$$

The value of $n - 2$ is an even number if n is an even number. This implies the computation of $\widetilde{I}$ as follows:

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \oplus \underbrace{S_{r_2} \oplus S_{r_2} \oplus \ldots \oplus S_{r_2}}_{n-2 \text{ is even}} \tag{13}$$

The basic property of XOR operation over "even number times" indicates the result $\underbrace{S_{r_2} \oplus S_{r_2} \oplus \ldots \oplus S_{r_2}}_{n-2 \text{ is even}} = 0$*. Thus, the form in Equation (13) can be further simplified as:*

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \oplus 0$$

The XOR property with "zero number" produces the following result:

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \tag{14}$$

If the observed pixel of the secret image is black (i.e., $I = 0$*), then* $S_{r_1} = S_{r_2}$*. The XOR property for two identical numbers implies the following result:*

$$\widetilde{I} = S_{r_1} \oplus S_{r_1} = 0 \tag{15}$$

The result in Equation (15) tells that the original and the recovered secret image are identical if the secret image is 0 *and n is even number. Based on this fact, we can conclude that* $\widetilde{I} = I$*. However, one obtains a correct recovered secret image.*

For the situation where n is an even number and the secret image is a white pixel (i.e., $I = 1$*), the second selected shared image is set as* $S_{r_2} \leftarrow\sim S_{r_1}$*. The substitutive computation of* $\widetilde{I}$ *in Equation (14) is indicated as follows:*

$$\widetilde{I} = S_{r_1} \oplus \sim S_{r_1}.$$

The XOR property on two complementary numbers yields the following result:

$$\widetilde{I} = 1 \tag{16}$$

The result in Equation (16) reveals that the qualities of the recovered and the original secret image are identical if n is an even number (i.e., $\widetilde{I} = I$*) while I is white. The former scheme [15] yields a correct result if n is an even number.*

If n is an odd number, the computation of $\widetilde{I}$ *is given as:*

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \oplus \underbrace{S_{r_2} \oplus S_{r_2} \oplus \ldots \oplus S_{r_2}}_{n-2 \text{ is odd}}$$

The XOR property "odd number times" yields $\underbrace{S_{r_2} \oplus S_{r_2} \oplus \ldots \oplus S_{r_2}}_{n-2 \text{ is odd}} = S_{r_2}$*. The recovered secret image is then:*

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \oplus S_{r_2}$$

The XOR operation concerning two identical numbers results in 0. It gives the following result:

$$\widetilde{I} = S_{r_1} \oplus 0 = S_{r_1} \tag{17}$$

The recovered image obtained from Equation (17) is actually a random image (i.e., $\widetilde{I} = S_{r_1}$ *with* $S_{r_1} \leftarrow U_I(0,1)$*). The recovered secret image cannot be correctly produced if n is an odd number (i.e.,* $\widetilde{I} \neq I$*).*

Under the similar deduction for $T < n$ *(i.e., only a partial set of shared images is involved in the reconstructed process), the image* $\widetilde{I}$ *is computed as:*

$$\widetilde{I} = S_{t_1} \oplus S_{t_2} \oplus \ldots \oplus S_{t_T}$$

Suppose that $r_1 < r_2 \leq T$*. The computation of* $\widetilde{I}$ *is then given as:*

$$\widetilde{I} = S_{t_1} \oplus S_{t_2} \oplus \ldots \oplus S_{r_1} \oplus \ldots \oplus S_{r_2} \oplus \ldots \oplus S_{t_T}$$

In the former scheme [15], all shared images are simply determined as $S_{t_i} \leftarrow S_{r_2}$ for all t_i with $i = 1, 2, \ldots, T$ and $t_i \neq r_1, r_2$. This condition implies:

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \oplus S_{r_2} \oplus \ldots \oplus S_{r_2}$$

The XOR properties "even number times" and "odd number times" indicate the following result:

$$\widetilde{I} = \begin{cases} S_{r_1} \oplus S_{r_2} \oplus \underbrace{S_{r_2} \oplus S_{r_2} \oplus \ldots \oplus S_{r_2}}_{T-2 \text{ is odd}} = S_{r_1} \oplus S_{r_2} \oplus S_{r_2} \\ S_{r_1} \oplus S_{r_2} \oplus \underbrace{S_{r_2} \oplus S_{r_2} \oplus \ldots \oplus S_{r_2}}_{T-2 \text{ is even}} = S_{r_1} \oplus S_{r_2} \oplus 0 \end{cases}$$

Then, the image $\widetilde{I}$ can be finally obtained as follows:

$$\widetilde{I} = \begin{cases} S_{r_1}, & If\ T \text{ is odd} \\ S_{r_1} \oplus S_{r_2}, & If\ T \text{ is even} \end{cases} \tag{18}$$

It is clearly revealed from Equation (18) that the recovered secret image can be perfectly reconstructed if T is an even number. Thus, the perfect recovered secret image can be obtained if and only if the number of stacked shared images is even. This concludes the theorem. ■

3. Proposed PVSS Method

This section gives a detailed explanation for the proposed PVSS method. This new method modifies the former scheme [15] for computing a set of generated shared images. This modification is made to achieve the perfect reconstruction result in the recovery process. Similarly to the former scheme [15], the proposed method also incorporates the adaptive priority weight for a progressive recovery process. However, the proposed method and the former scheme employ an identical approach for performing the secret image reconstruction (i.e., stacking several or all shared images using a bitwise XOR approach). This section presents two techniques for the proposed method using the bitwise-based and XOR-based PVSS approaches.

3.1. Proposed Bitwise-Based PVSS Method

The proposed bitwise-based PVSS method is discussed in detail in this subsection. It inherits the usability of the former scheme [15] with a slight modification. This simple modification simply solves a minor limitation in [15] present when the number of stacked or collected shared images is odd. The proposed method also utilizes the random grid technique for generating a set of shared images. The proposed bitwise-based PVSS method for computing a set of shared images is formally explained with the following procedure: Suppose I is a binary image of size $M \times N$. The proposed method transforms the secret image I into n shared images. Let $\{S_1, S_2, \ldots, S_n\}$ be a set of generated shared images, and (x, y) be the spatial position of an image pixel. Similarly to the former scheme [15], the proposed method first determines two selected shared images (denoted as r_1 and r_2). The determinations of r_1 and r_2 are based on the priority weight w_j and location set $\updownarrow_j$. Each pixel (x, y) in the r_1-th shared image is set with a uniformly random number as follows:

$$S_{r_1}(x, y) \leftarrow U_I(0, 1) \tag{19}$$

for $x = 1, 2, \ldots, M$ and $y = 1, 2, \ldots, N$. Subsequently, each pixel (x, y) in the r_2-th shared image is determined by considering the pixel value $I(x, y)$. If an investigated pixel value is black (i.e., $I(x, y) = 0$), afterward, the value of $S_{r_2}(x, y)$ is assigned as follows:

$$S_{r_2}(x, y) \leftarrow S_{r_1}(x, y) \tag{20}$$

Meanwhile, the value of $S_{r_2}(x,y)$ is simply set with the bit negation of $S_{r_1}(x,y)$ as formulated below:

$$S_{r_2}(x,y) \leftarrow\sim S_{r_1}(x,y) \tag{21}$$

In contrast to the former scheme [15], the proposed method solely utilizes the zero value for all pixels in the shared images S_i under the constraints $1 \leq i \leq n$ and $i \neq r_1, r_2$. This strategy is formally defined as:

$$S_i(x,y) \leftarrow 0 \tag{22}$$

This shared image generation is applied over all pixel values (x,y), for $x = 1,2,\ldots,M$ and $y = 1,2,\ldots,N$. Algorithm 2 summarizes the procedure of the proposed method for computing a set of shared images $\{S_1, S_2, \ldots, S_n\}$.

```
Algorithm 2: Proposed Bitwise-Based PVSS Method.
Input: Secret image in binary format, $I$, of size $M \times N$
       Number of shared images, $n$
Output: A set of generated shared images, $\{S_1, S_2, \ldots, S_n\}$, each of size $M \times N$
Step 1: Based on priority weight $w_j$, determine the location set $\updownarrow_j$, for $j = 1,2,\ldots,n$.
Step 2: For Each Pixel $(x,y)$. Based on information of $\updownarrow_j$, select two shared images $r_1$ and $r_2$. Do
Step 3: $S_{r_1}(x,y) \leftarrow U_I(0,1)$
Step 4: If $I(x,y) = 0$, Then $S_{r_2}(x,y) \leftarrow S_{r_1}(x,y)$
Step 5: Else $S_{r_2}(x,y) \leftarrow\sim S_{r_1}(x,y)$
Step 6: For Each Generated shared images, $S_i$, with the condition $1 \leq i \leq n$ and $i \neq r_1, r_2$ Do
Step 7: $S_i(x,y) \leftarrow 0$
Step 8: Obtain $n$ generated shared images, $\{S_1, S_2, \ldots, S_n\}$
```

Similar to [15], the proposed method collects a partial or full set of generated shared images in order to recover the secret image. Herein, the proposed method also performs an XOR operation over these collected shared images. This process is defined as follows:

$$\widetilde{I} = S_{t_1} \oplus S_{t_2} \oplus \ldots \oplus S_{t_T} \tag{23}$$

where $\widetilde{I}$ denotes the recovered secret image and t_T is the total number of collected shared images on the receiver side. The proposed bitwise-based PVSS method is quite simple, yet it effectively solves the lossless problem in [15]. The following analysis supports the proposed method performance theoretically.

Theorem 2. *The proposed bitwise-based PVSS method yields a perfectly reconstructed secret image by stacking a partial or full set of generated shared images.*

Proof. *We begin this proof with the quality of the recovered secret image produced by the proposed bitwise-based PVSS method. The XOR-ed process over a partial or full set of generated shared images is denoted as:*

$$\widetilde{I} = S_{t_1} \oplus S_{t_2} \oplus \ldots \oplus S_{t_T}$$

We first investigate the proposed method performance when one involves all shared images in the recovery process. In this occasion, it similarly performs a recovery process under a condition $T = n$. However, the computation of $\widetilde{I}$ can be performed as follows:

$$\widetilde{I} = S_1 \oplus S_2 \oplus \ldots \oplus S_n \tag{24}$$

We know that $1 \leq r_1, r_2 \leq n$ *and* $S_i \leftarrow 0$ *for all* $1 \leq i \leq n$ *and* $i \neq r_1, r_2$. *This implies that Equation (24) can be recalculated when considering n as an even or odd number as follows:*

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \oplus 0 \oplus \ldots \oplus 0, \widetilde{I} = S_{r_1} \oplus S_{r_2} \oplus \underbrace{0 \oplus 0 \oplus \ldots \oplus 0 \oplus 0}_{n-2 \ is \ odd/even \ number} \tag{25}$$

Performing XOR on any arbitrary number with zero is equivalent to the arbitrary number itself. Thus, the Equation (25) can be simplified as:

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \tag{26}$$

Stacking all shared images actually resembles the process of performing an XOR operation between S_{r_1} and S_{r_2}. *In addition, the proposed method gives various values for* S_{r_2}, *depending on the value of* I. If $I = 1$, *then the value of* S_{r_2} *is identically set with the value of* S_{r_1}. *While* $I = 0$, *the value of* S_{r_2} *is set in bitwise negation of* S_{r_1} *(i.e.,* $S_{r_2} \leftarrow S_{r_1}$*). This condition gives* $\widetilde{I}$ *as follows:*

$$\widetilde{I} = \begin{cases} S_{r_1} \oplus S_{r_1}, & If\ I = 0 \\ S_{r_1} \oplus \sim S_{r_1}, & If\ I = 1 \end{cases}$$

The following result is obtained based on the XOR property:

$$\widetilde{I} = \begin{cases} 0, & If\ I = 0 \\ 1, & If\ I = 1 \end{cases} \tag{27}$$

The last form indicates an important result (i.e., $\widetilde{I} = I$ *). Herein, the recovered and the original secret image are identical. Thus, the proposed method yields a perfectly reconstructed secret image when all shared images are involved in the recovery process.*

If only a partial set of shared images is involved under condition $T < n$, *the recovered secret image* $\widetilde{I}$ *can be computed as follows:*

$$\widetilde{I} = S_{t_1} \oplus S_{t_2} \oplus \ldots \oplus S_{t_T}$$

Suppose that the two selected shared images (S_{r_1} *and* S_{r_2}*) are in this partial set under the condition* $r_1 < r_2 \leq T$. *One cannot obtain a perfectly reconstructed secret image if this condition is not satisfied. The computation of* $\widetilde{I}$ *is then given as:*

$$\widetilde{I} = S_{t_1} \oplus S_{t_2} \oplus \ldots \oplus S_{r_1} \oplus \ldots \oplus S_{r_2} \oplus \ldots \oplus S_{t_T}$$

Based on the fact that $S_i \leftarrow 0$ *for all* $1 \leq i \leq n$ *and* $i \neq r_1, r_2$, *one can trivially obtain the following form:*

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \oplus S_{t_1} \oplus \ldots \oplus S_{t_T} = S_{r_1} \oplus S_{r_2} \oplus \underbrace{0 \oplus 0 \oplus \ldots \oplus 0 \oplus 0}_{T-2 \ is \ odd/even \ number}$$

The XOR property implies the following result:

$$\widetilde{I} = S_{r_1} \oplus S_{r_2}$$

The last form indicates that the value of $\widetilde{I}$ *is identical to that of the XOR-ed result between* S_{r_1} *and* S_{r_2}. *By investigating the value of* I, *we acquire the following conclusion:*

$$\widetilde{I} = \begin{cases} S_{r_1} \oplus S_{r_1} = 0, & If\ I = 0 \\ S_{r_1} \oplus \sim S_{r_1} = 1, & If\ I = 1 \end{cases}$$

In the case of $T < n$, *we achieve an important deduction (i.e.,* $\widetilde{I} = I$*) To simplify, the quality of the recovered secret image is identical to that of the original secret image. In addition, a perfect* *recovered secret image can be yielded if either a partial set or all generated shared images are involved in the reconstruction process. This completes the proof.* ■

3.2. Proposed XOR-ed Based PVSS Method

In this approach, the proposed method performs a simple computation involving an XOR operation in order to generate a set of shared images. The proposed method takes an image I of $M \times N$ as the secret image to produce the targeted shared images $\{S_1, S_2, \ldots, S_n\}$. Herein, the secret image can be present as binary, grayscale, or color space. For each pixel (x, y) in the secret image, we perform the following procedure to generate n shared images: We first decide two selected shared images (r_1 and r_2). In contrast to the former scheme [15] and the proposed bitwise-based approach, the proposed XOR-based method needs to first generate the following constant:

$$C \leftarrow U_I(a, b) \tag{28}$$

where C is a constant. The symbol $U_I(a, b)$ denotes a uniform random number generator producing an integer in range $[a, b]$. We utilize $U_I(0, 1)$ and $U_I(0, 255)$ for the binary image and the 8-bit grayscale image, respectively. A three dimensional image of $U_I(0, 255)$ can be used for the 24-bit color image (i.e., generating a random number for three dimensional color spaces). Subsequently, all pixels in two selected shared images (S_{r_1} and S_{r_2}) are determined as follows:

$$\mathrm{S_{r_1}(x, y) \leftarrow I(x, y)} \oplus C \tag{29}$$

$$S_{r_2}(x, y) \leftarrow C \tag{30}$$

All pixels in S_i are simply set with zero value for $1 \leq i \leq n$ and $i \neq r_1, r_2$. Alternatively, the pixels are set according to the following process:

$$S_i(x, y) \leftarrow 0 \tag{31}$$

The proposed XOR-ed PVSS method requires simple steps to compute a set of shared images. This simple approach is also applicable for grayscale and color images. The contents of all shared images are totally different compared to that of the original secret image. In addition, the proposed XOR-based PVSS method is designed to solve a slight problem in the former scheme [15]. Algorithm 3 illustrates the shared image generation using the proposed XOR-based PVSS approach.

Algorithm 3: Proposed XOR-ed Based PVSS Method.

```
Input: A grayscale or color image as secret, $I$, of size $M \times N$
Number of shared images, $n$
Output: Full set of generated shared images, $\{S_1, S_2, \ldots, S_n\}$, each of size $M \times N$

Step 1: Based on priority weight $w_j$, determine the location set $\updownarrow_j$, for $j = 1, 2, \ldots, n$.
Step 2: For Each Pixel Position $(x, y)$. Based on the information in $\updownarrow_j$, decide the selected shared images $r_1$ and $r_2$. Do
Step 3: $C \leftarrow U_I(a, b)$
Step 4: $S_{r_1}(x, y) \leftarrow I(x, y) \oplus C$
Step 5: $S_{r_2}(x, y) \leftarrow C$
Step 6: For Each other generated shared images, $S_i$, under the condition $1 \leq i \leq n$ and $i \neq r_1, r_2$ Do
Step 7: $S_i(x, y) \leftarrow 0$
Step 8: Obtain the $n$ generated shared images, $\{S_1, S_2, \ldots, S_n\}$
```

The proposed method reconstructs the secret image in a similar fashion as compared to the former scheme [15]. Herein, the proposed method simply needs to perform an XOR operation over either a partial set or all generated shared images. The following theorem supports the correctness of the proposed method.

Theorem 3. *The proposed XOR-ed based PVSS method yields a perfectly reconstructed secret image by stacking partial or all generated shared images.*

Proof. *In order to prove this theorem, we first examine the quality of recovered secret image produced by our proposed method. Suppose that all generated shared images are involved in the recovery process of a secret image. This indicates that $T = n$. Thus, the $\widetilde{I}$ can be produced as follows:*

$$\widetilde{I} = S_1 \oplus S_2 \oplus \ldots \oplus S_n = S_{r_1} \oplus S_{r_2} \tag{32}$$

The simplified form in Equation (32) is actually identical to that of Equation (26). The proposed method applies a similar strategy on S_i (i.e., $S_i \leftarrow 0$ for $1 \leq i \leq n$ and $i \neq r_1, r_2$). The selected shared images S_{r_1} and S_{r_2} are set with the value of $I \oplus C$ and C, respectively. The form in Equation (32) is similar to following computation:

$$\widetilde{I} = (I \oplus C) \oplus C\widetilde{I} = I \oplus (C \oplus C) \tag{33}$$

The XOR property indicates that an XOR operation between two identical scalars yields zero value. Thus, the form in Equation (33) has the following result:

$$\widetilde{I} = I \oplus 0$$

The XOR operation between scalar and zero produces the scalar itself. However, one obtains the following result:

$$\widetilde{I} = I \tag{34}$$

The last form in Equation (34) clearly reveals that the proposed XOR-based PVSS method achieves a lossless result. The qualities of the recovered and the original secret image are identical if all generated shared images are utilized in the recovery process.

If only several shared images are involved in the recovery of a secret image (i.e., in the case of $T < n$), the recovered secret image $\widetilde{I}$ is computed as follows:

$$\widetilde{I} = S_{t_1} \oplus S_{t_2} \oplus \ldots \oplus S_{t_T}$$

The condition $r_1 < r_2 \leq n$ implies the following result:

$$\widetilde{I} = S_{r_1} \oplus S_{r_2} \tag{35}$$

A similar deduction on Equation (33) can be applied for Equation (35). Thus, we conclude an important result (i.e., $\widetilde{I} = I$). The proposed XOR-based PVSS method is able to reconstruct a secret image with the lossless condition even if only a partial set of generated shared images is involved in the reconstruction process. The proposed XOR-based PVSS approach yields a perfect recovered secret image whether all or several generated shared images are utilized in the recovery process. This concludes the proof. ■

4. Experimental Results

The performances of the proposed method and the former scheme [15] are extensively reported in this section in terms of dealing with the PVSS tasks. We first explain several image quality assessment metrics to objectively measure the degree of similarity between the original and the recovered secret image. Subsequently, the performances of the proposed method are compared under visual investigation and objective measurement over binary, grayscale, and color images. These assessments are conducted to further investigate the proposed method's usability and superiority. The comparisons in terms of algorithm aspects between the proposed method and competing schemes are summarized at the end of this section.

4.1. Performance Evaluation

We first evaluated the performance under the subjective and objective assessments. For the subjective image quality assessment, the quality of the recovered secret image was simply inspected and judged based on a visual observation. Herein, we visually compared the similarities between the original and the recovered secret image under the perception of human vision, whereas the objective image quality assessment utilizes several metrics to calculate the degree of similarity between the original and the recovered secret image. These metrics are referred to as image contrast α, bit error rate β, peak signal-to-noise ratio (PSNR), structural similarity index metric (SSIM), and mean absolute error (MAE). All of these objective metrics are formally defined as follows:

$$\alpha = \frac{T\left(\widetilde{I}[I(1)]\right) - T\left(\widetilde{I}[I(0)]\right)}{1 + T\left(\widetilde{I}[I(0)]\right)} \tag{36}$$

$$\beta = \frac{\sum_{x=1}^{M}\sum_{y=1}^{N} I(x,y) \oplus \widetilde{I}(x,y)}{MN} \tag{37}$$

$$PSNR(I,\widetilde{I}) \ = 20\log_{10}\frac{255}{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[I(i,j)\ -\ \widetilde{I}(i,j)\right]^2} \tag{38}$$

$$SSIM\left(I,\widetilde{I}\right) = \frac{(2\mu_I\mu_{\widetilde{I}} + c_1)\,(2\sigma_{I\widetilde{I}} + c_2)}{\left(\mu_I^2 + \mu_{\widetilde{I}}^2 + c_1\right)\left(\sigma_I^2 + \sigma_{\widetilde{I}}^2 + c_2\right)} \tag{39}$$

$$MAE\left(I,\widetilde{I}\right) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|I(i,j) - \widetilde{I}(i,j)\right| \tag{40}$$

where I is the secret image, and $\widetilde{I}$ is the recovered secret image. These two images are of the same size (i.e., $M \times N$).

In the SSIM computation, the symbols μ_I and $\mu_{\widetilde{I}}$ are the mean values of I and $\widetilde{I}$, respectively. However, the standard deviations of I and $\widetilde{I}$ are denoted as σ_I and $\sigma_{\widetilde{I}}$, respectively. Meanwhile, the covariance between I and $\widetilde{I}$ is denoted as $\sigma_{I\widetilde{I}}$. The c_1 and c_2 are two predetermined constants. In the case of a binary image, the symbols $T\left(\widetilde{I}[I(1)]\right)$ and $T\left(\widetilde{I}[I(0)]\right)$ denote the average light transmission [8] of the recovered secret images over a white pixel (1) and a black pixel (0), respectively. In our subsequent experiment, a better performance is indicated by higher scores of α, PSNR, and SSIM, and vice versa. On the other hand, a better performance is also implied by lower values of β and MAE, and vice versa.

4.2. Visual Evaluation on Binary Image

The visual investigation between the proposed method and the former scheme [15] in terms of a binary image is reported in this subsection. We examined the performances of the proposed method and the former scheme [15] under a set of binary images as displayed in Figure 5. In this experiment, we simply set the adaptive priority weights w_j as $\{0.4,\ 0.3,\ 0.15,\ 0.1,\ 0.025,\ 0.025\}$ and $\{0.4,\ 0.3,\ 0.2,\ 0.05,\ 0.05\}$ for $n = 6$ and $n = 5$, respectively. Figure 6 exhibits a set of generated shared images for $n = 6$ with the proposed bitwise-based PVSS method. Setting a higher value for adaptive priority weight implies a brighter shared image compared to that obtained by setting a lower value of w_j. In addition, the contents of all generated shared images are in a noise-like appearance, meaning that each image cannot easily be distinguished. This clearly reveals that the proposed method satisfies the PVSS constraint (i.e., that the content of the generated shared images cannot be recognized by an unauthorized party).

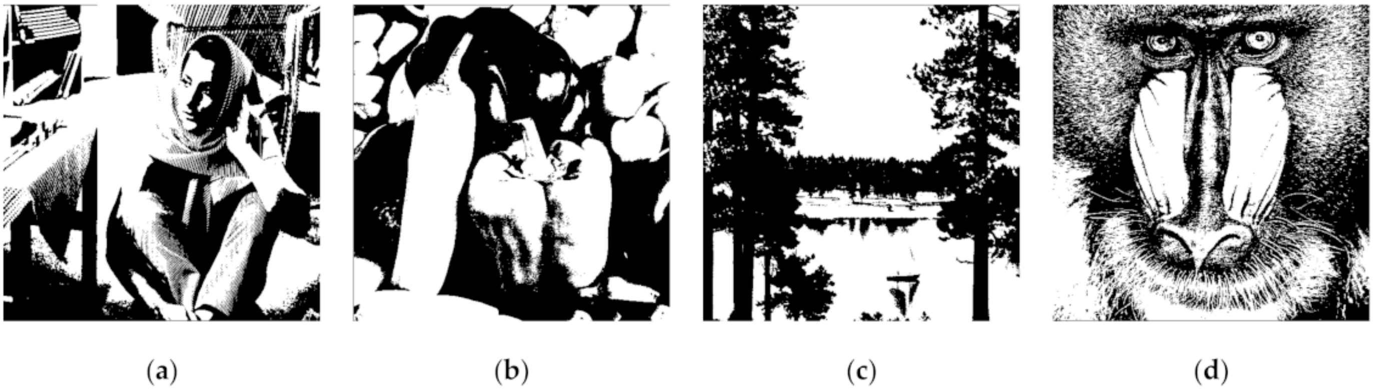


Figure 5. Four secret images in a binary format for experiment (**a**) I_1, (**b**) I_2, (**c**) I_3, and (**d**) I_4.

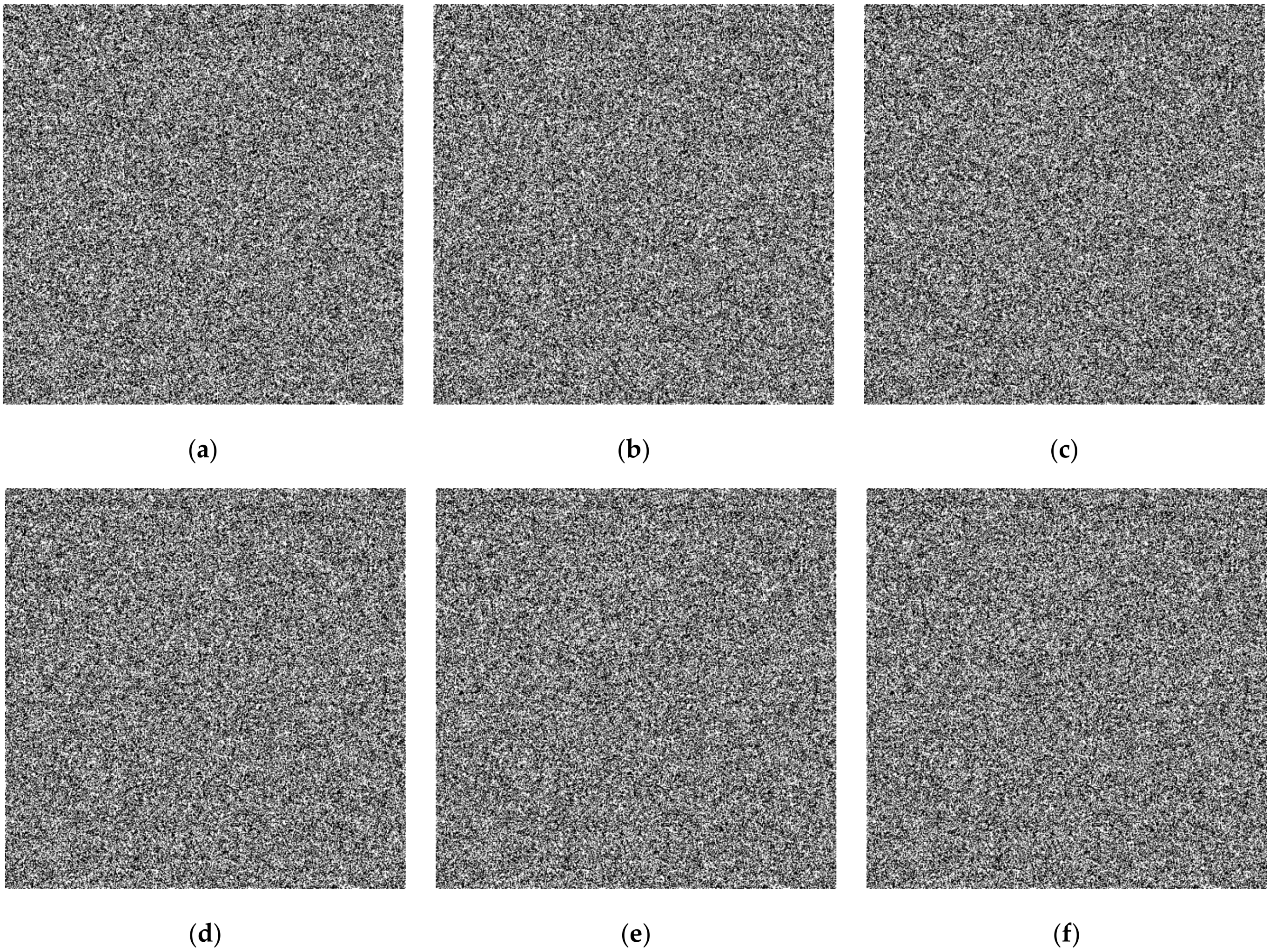


Figure 6. Generated binary shared images using the proposed bitwise-based PVSS method for $n = 6$: (**a**–**f**) $\{S_1, S_2, \ldots, S_6\}$.

Subsequently, we verified the quality of the recovered secret image. We select Barbara from Figure 5a as a binary secret image. We investigated and compared the performances of the proposed method and former scheme using visual inspection of the recovered secret image. Figure 7 displays the recovery process of the Barbara secret image for $n = 5$, while Figure 8 displays the recovery process for $n = 6$. These two figures demonstrated the superiority of the proposed method compared to that of [15]. The proposed method was able to produce the recovered secret image when the number of stacked shared image T

is odd. However, one cannot reconstruct the secret image using the former scheme [15] if T is an odd number. In addition, the recovered secret image produced by the proposed method is lossless if all shared images are stacked using an XOR operation. This experiment indicates that the proposed method offers a promising result in the PVSS task.

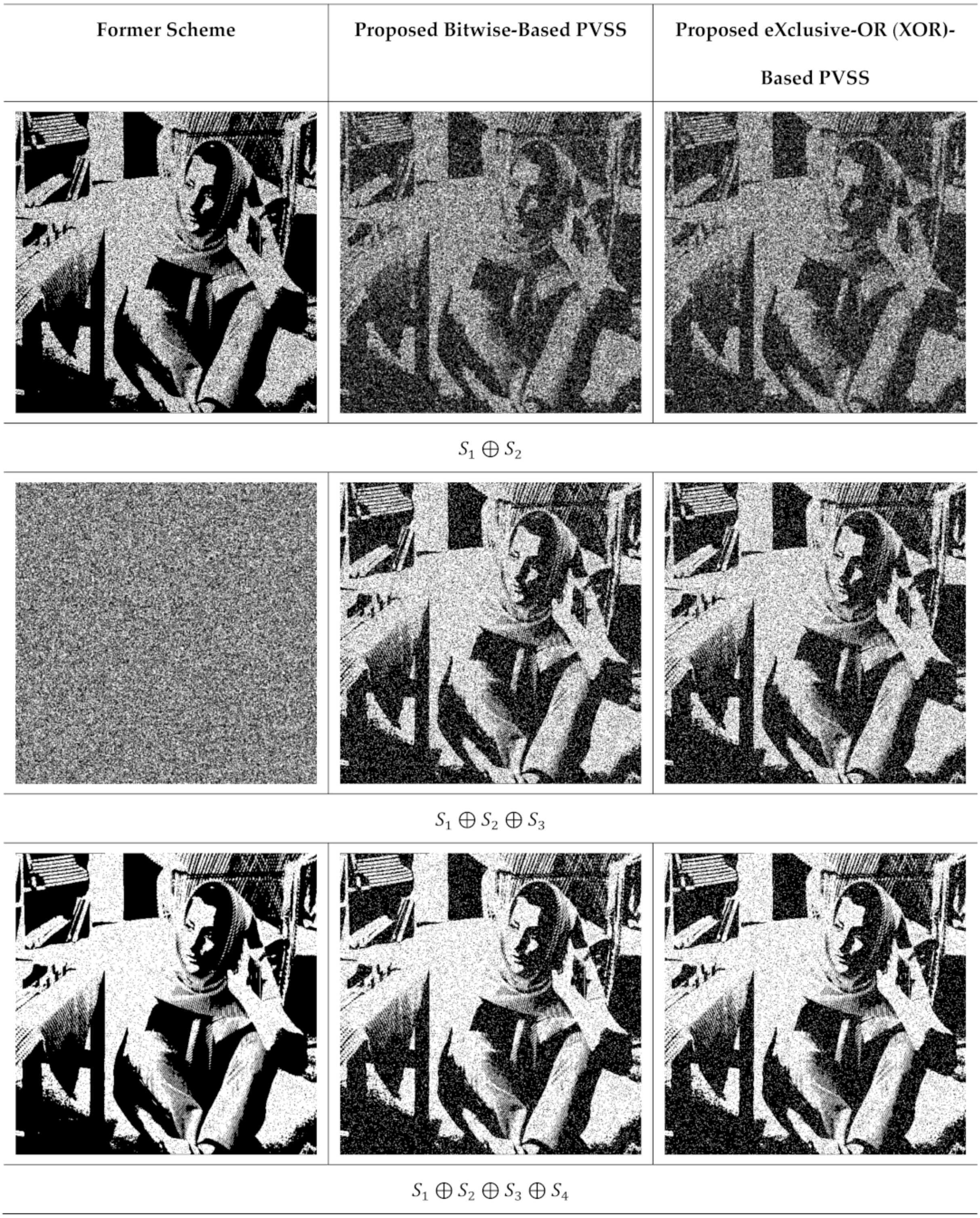

Former Scheme
Proposed Bitwise-Based PVSS
Proposed eXclusive-OR (XOR)-Based PVSS
$S_1 \oplus S_2$
$S_1 \oplus S_2 \oplus S_3$
$S_1 \oplus S_2 \oplus S_3 \oplus S_4$

Figure 7. *Cont.*

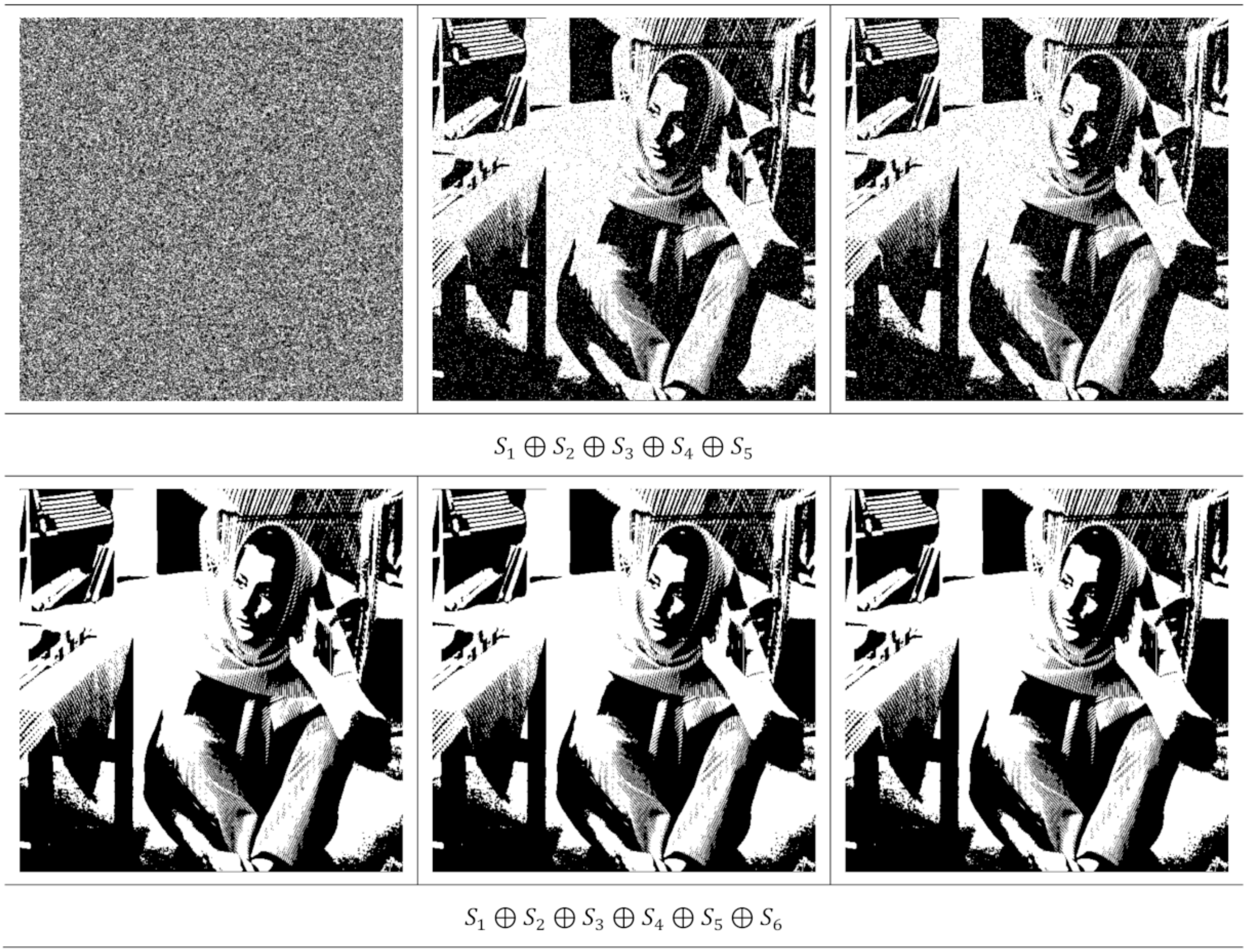

$S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5$
$S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5 \oplus S_6$

Figure 7. The results of stacking several shared images with $t = 2, 3, \ldots, 6$, by setting $n = 6$. The first colum is from the former scheme [15], while the second and third columns are from the proposed method.

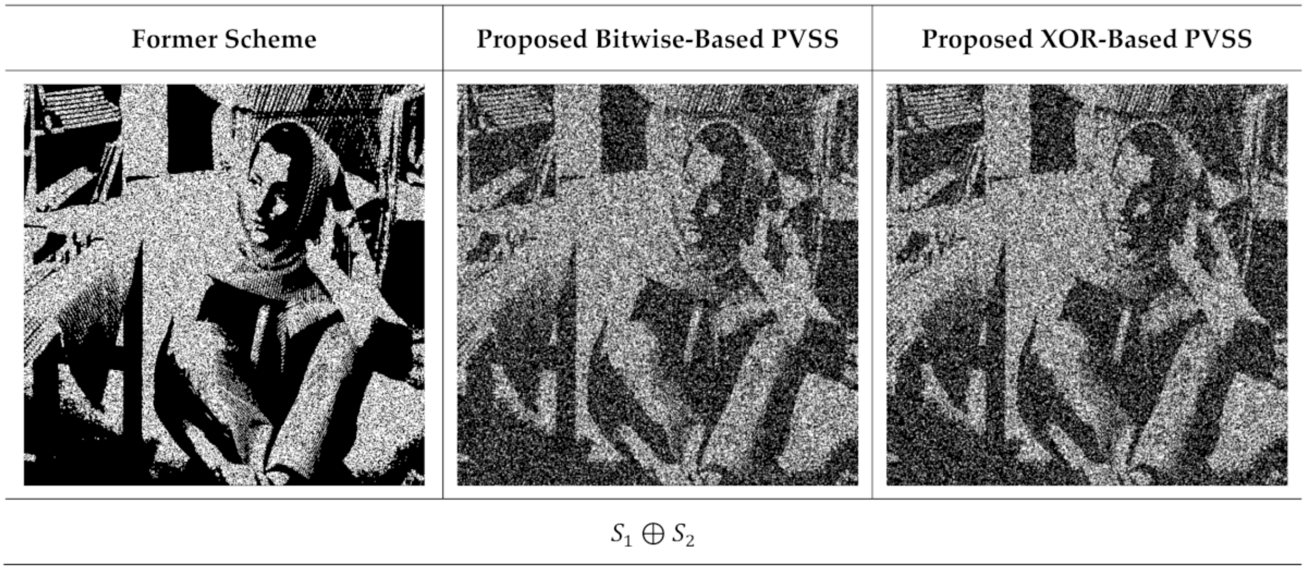
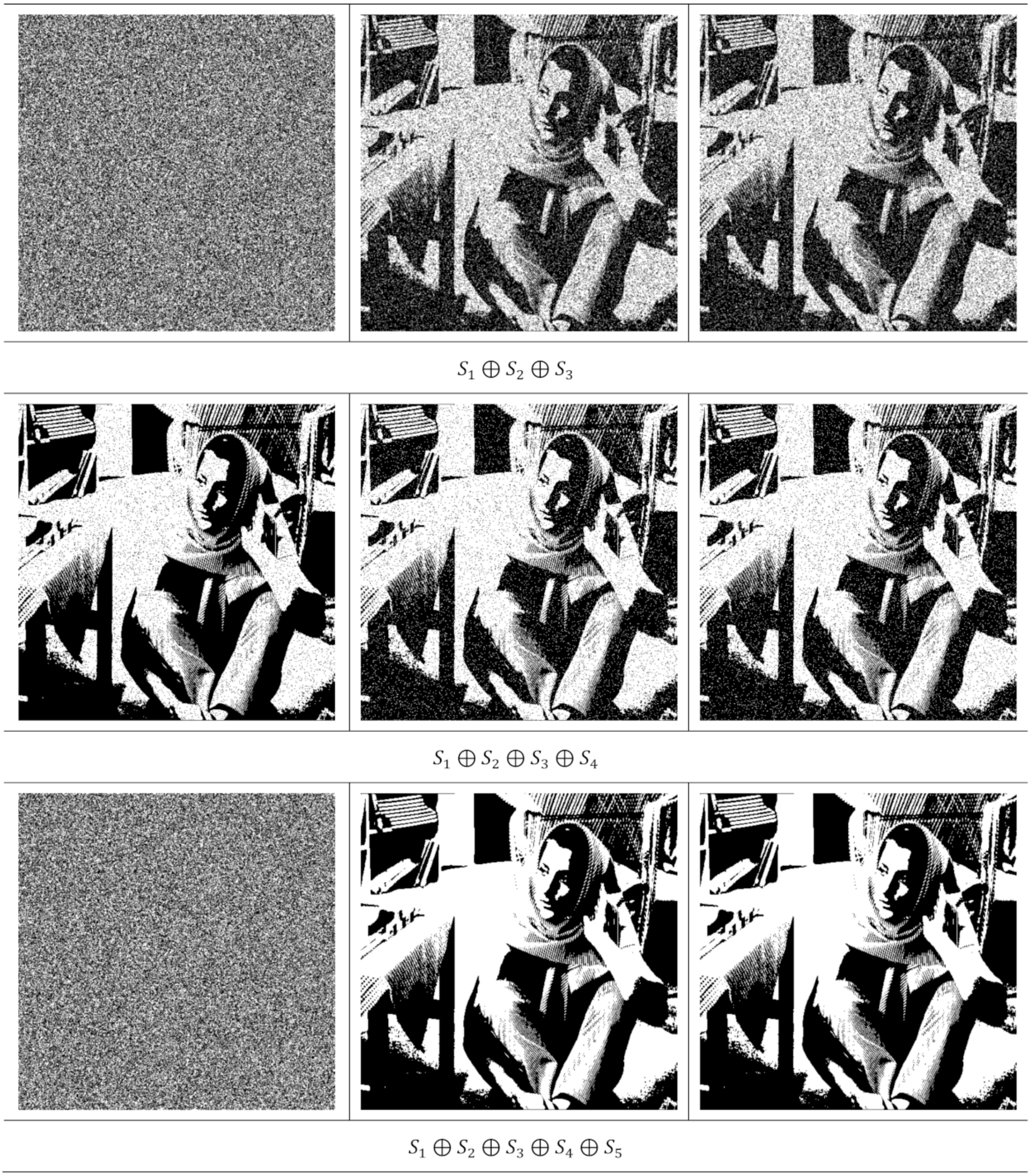


Figure 8. *Cont.*



$S_1 \oplus S_2 \oplus S_3$

$S_1 \oplus S_2 \oplus S_3 \oplus S_4$

$S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5$

Figure 8. Stacking several shared images $t = 2, 3, \ldots, 5$, by setting $n = 5$. The first column is from the former scheme [15], while the second and third columns are from the proposed method.

4.3. Visual Investigation on Grayscale Image

We subsequently considered the performance of the proposed method and the former scheme [15] under visual investigation. In this experiment, we examined the performances of four secret images in grayscale, as shown in Figure 9. The adaptive priority weights were identically set to those used in the binary image case. In the shared image generation, we simply employed $n = 5$ and $n = 6$. Figure 10 displays a set of shared images for $n = 6$ when the Barbara grayscale image is selected as a secret image. As depicted in this figure, the content of the generated shared images cannot be perceived and understood by human vision. This means that the shared images are effectively produced by the proposed method.

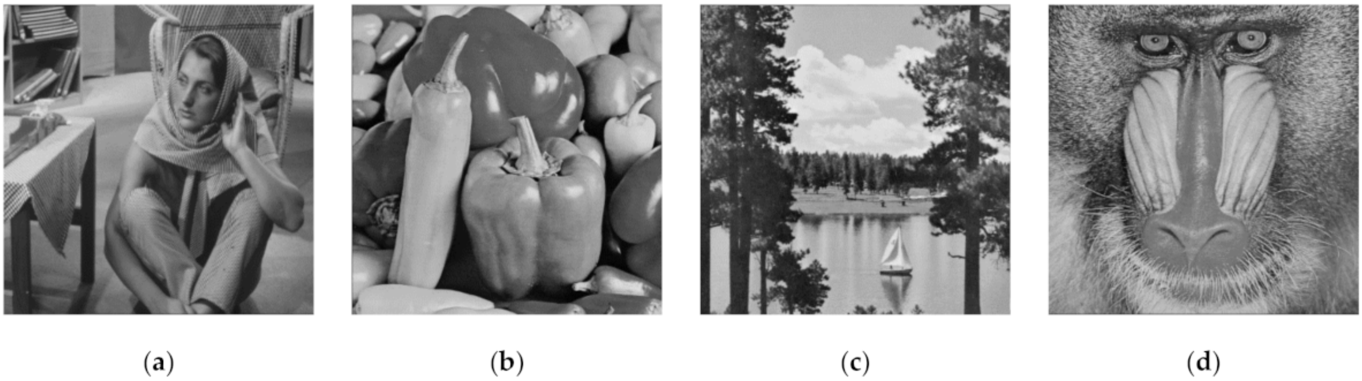


Figure 9. Four grayscale images as secret for experiment: (**a**–**d**) $\{I_1, I_2, I_3, I_4\}$.

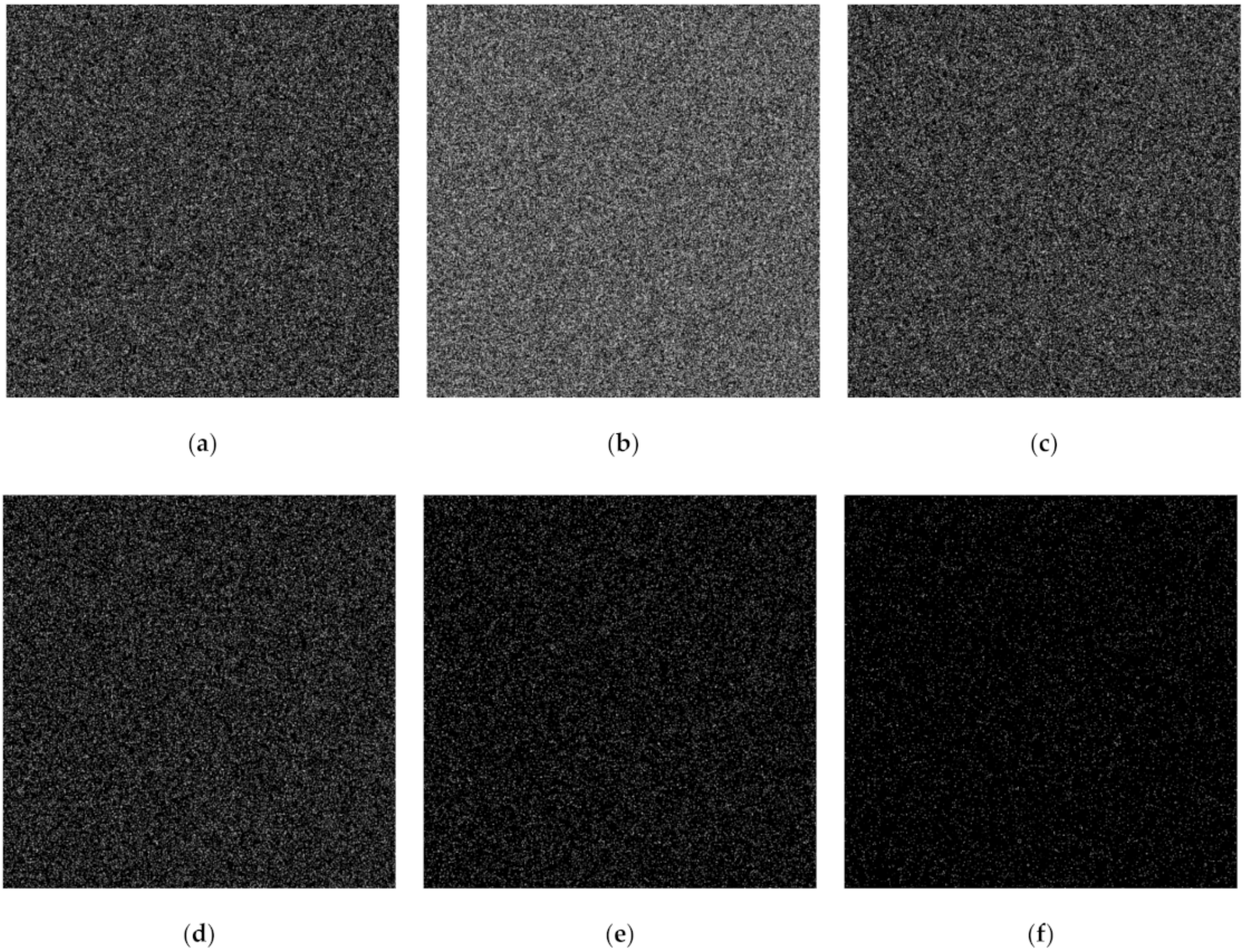


Figure 10. Generated shared images using the proposed eXclusive-OR (XOR)-based PVSS method: (**a**–**f**) $\{S_1, S_2, \ldots, S_6\}$, by setting $n = 6$.

The qualities of the recovered secret image were further inspected under visual investigation. Herein, the recovered secret image produced by stacking several shared images is shown in Figures 11 and 12 for $n = 6$ and $n = 5$, respectively. As shown in these two figures, the quality of the recovered secret image is increased if more shared images are involved in the reconstruction process. However, the former scheme [15] produces an incorrectly recovered secret image if the number of stacked shared images is odd (i.e., the content of the recovered secret image cannot be correctly reconstructed after the stacking process). Conversely, the proposed XOR-based method works well, indicating its superiority compared to that of the former scheme [15].

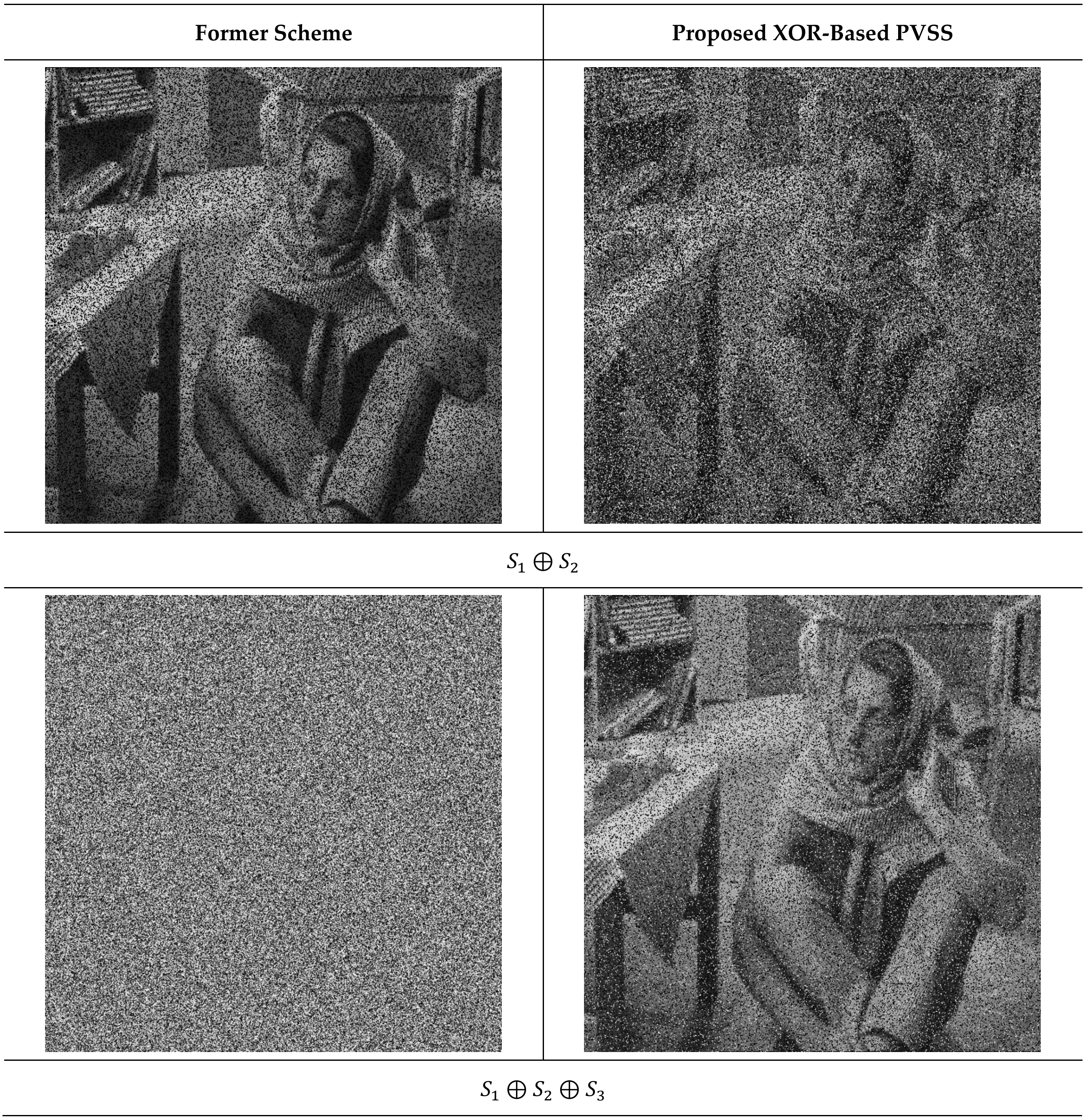
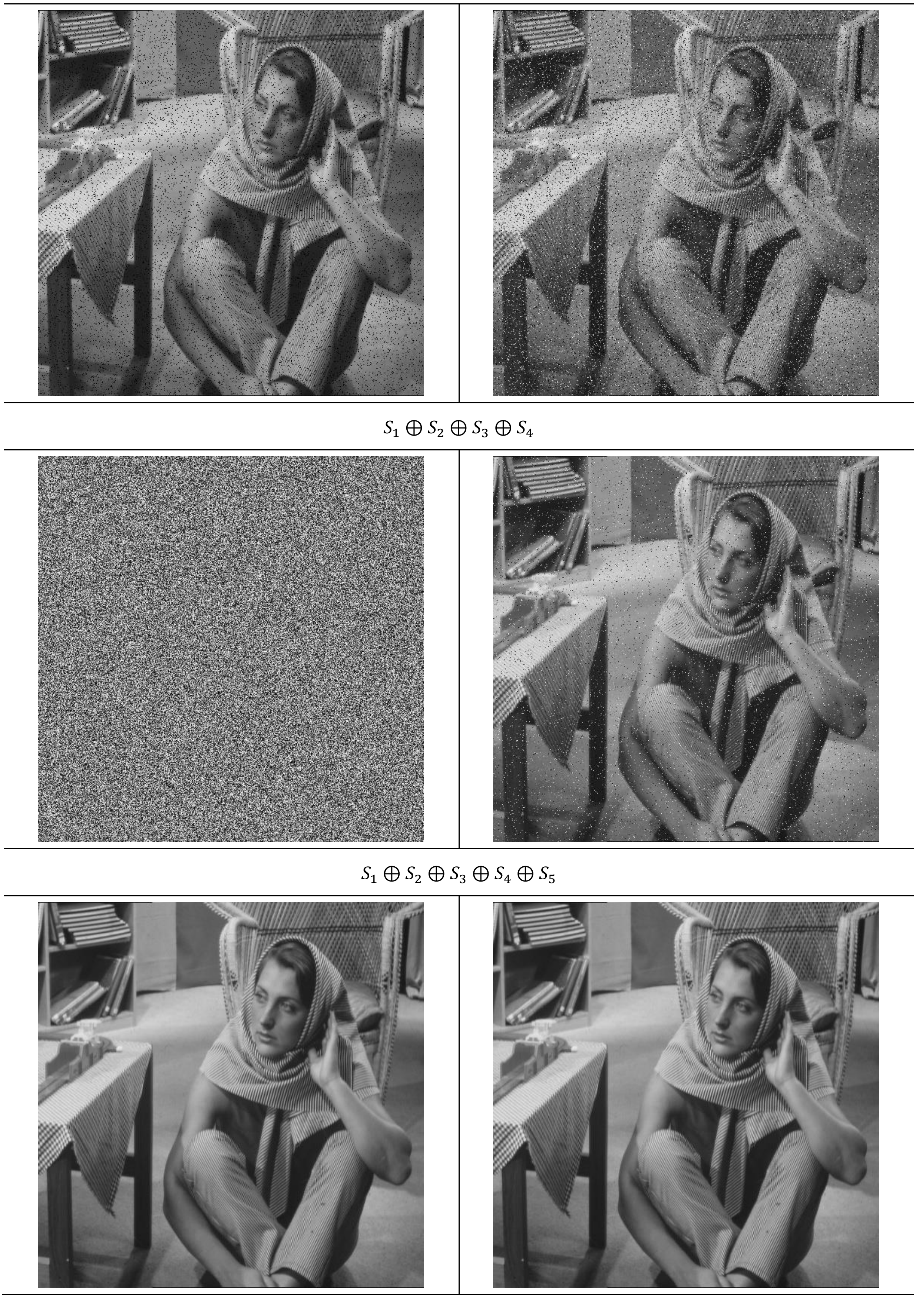


Figure 11. *Cont.*



$S_1 \oplus S_2 \oplus S_3 \oplus S_4$

$S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5$

Figure 11. *Cont.*

$S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5 \oplus S_6$

Figure 11. Stacking several shared images with $t = 2, 3, \ldots, 6$ and $n = 6$. The left and right columns are from the former scheme [15] and the proposed method, respectively.

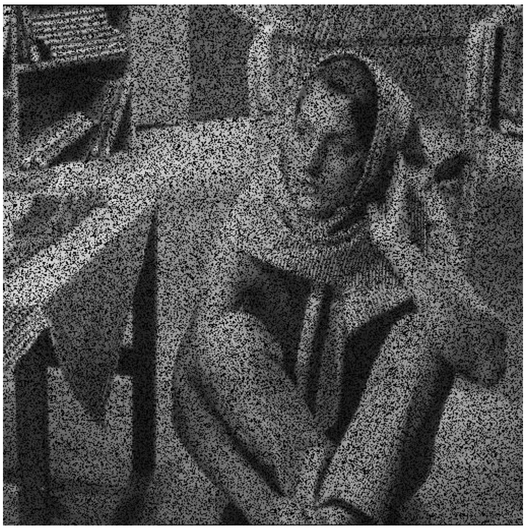
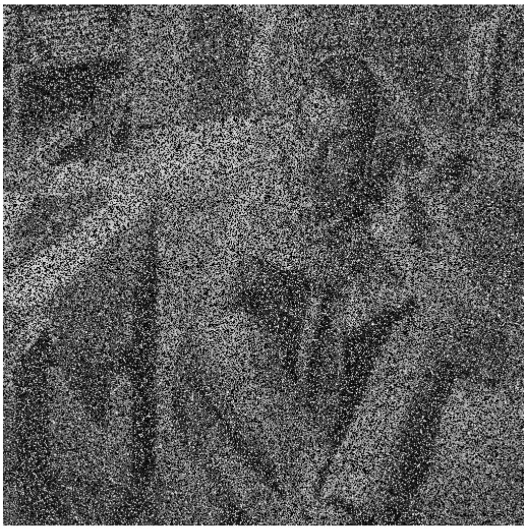
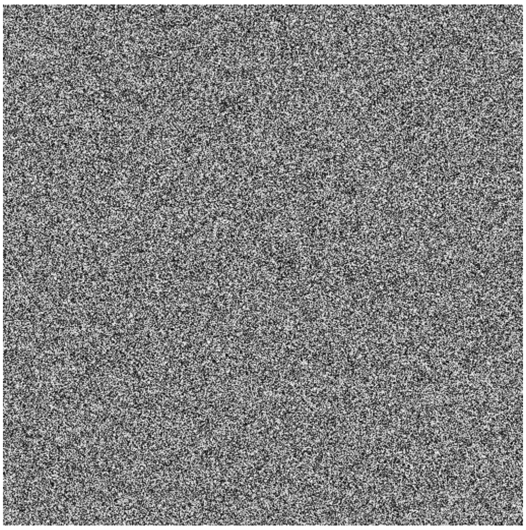
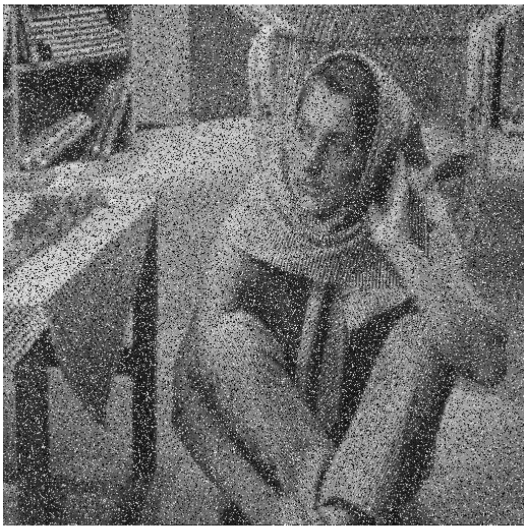
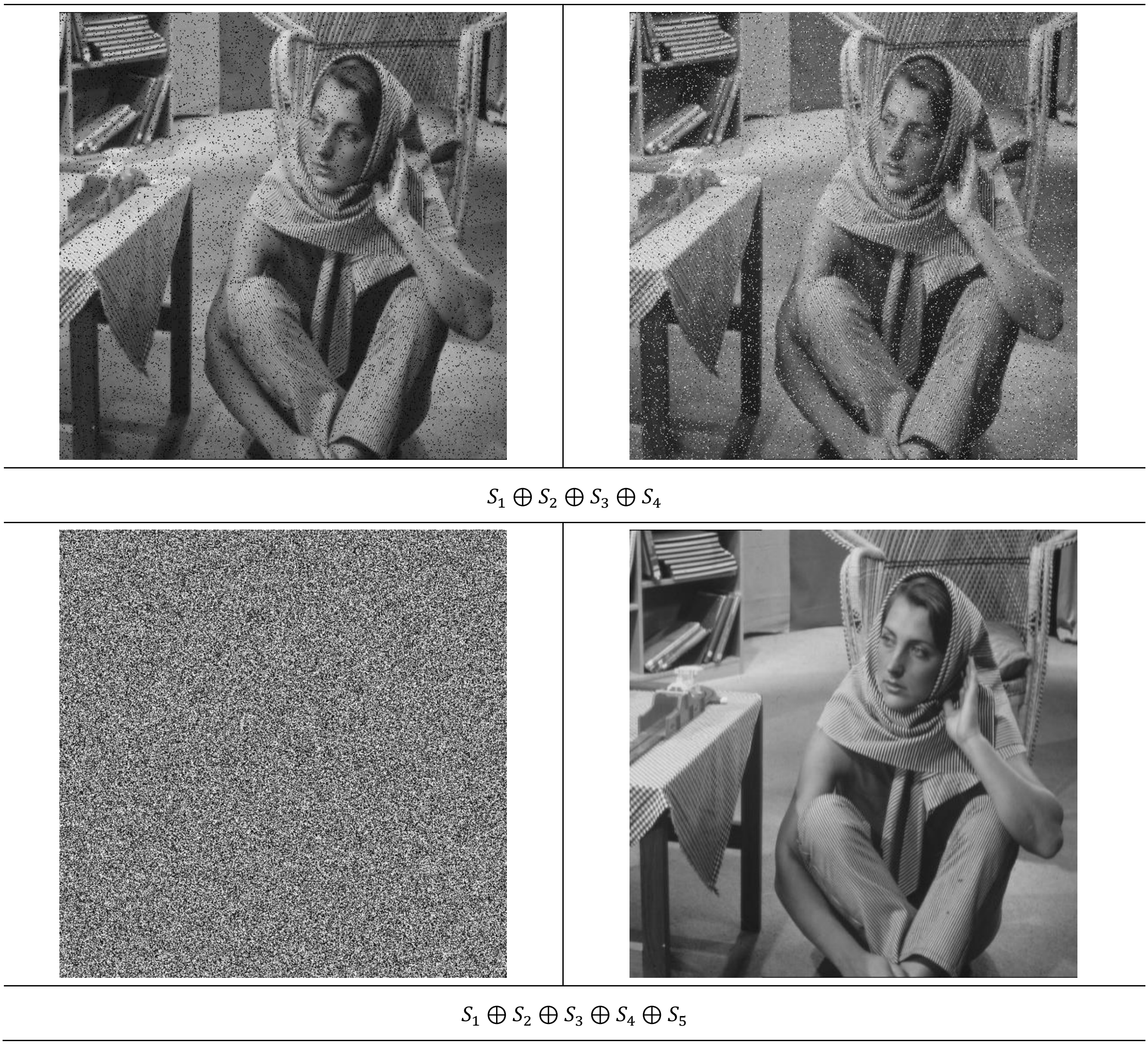
Former Scheme	Proposed XOR-Based PVSS
	
$S_1 \oplus S_2$	
	
$S_1 \oplus S_2 \oplus S_3$	

Figure 12. *Cont.*



$S_1 \oplus S_2 \oplus S_3 \oplus S_4$

$S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5$

Figure 12. Reconstructed secret images when several shared images are stacked, with $t = 2, 3, \ldots, 5$ and $n = 5$. The left and right columns are from the former scheme [15] and the proposed method, respectively.

4.4. Visual Assessment of Color Image

This subsection compares the performances of the former scheme [15] and the proposed method under the visual inspection on color image. Four color images (as shown in Figure 13) were used for experimentation. Herein, the number of shared images is set as $n = 5$ and $n = 6$. We applied an identical adaptive priority weight, as used in the binary image case. Figure 14 displays a set of shared images in color format, while the color image in Figure 13a was chosen as a secret image. Human vision cannot recognize the object or image content from all shared images as delivered in Figure 14. Thus, it can be concluded that the proposed method effectively produces a set of shared images in color format.

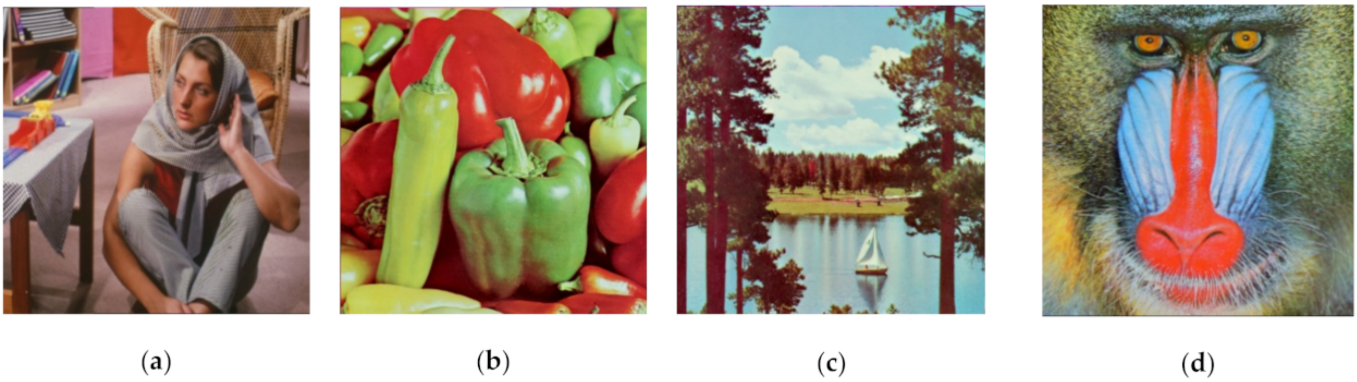


Figure 13. A set of color images used as secret images in the experiment, denoted as: (**a**) I_1, (**b**) I_2, (**c**) I_3, and (**d**) I_4

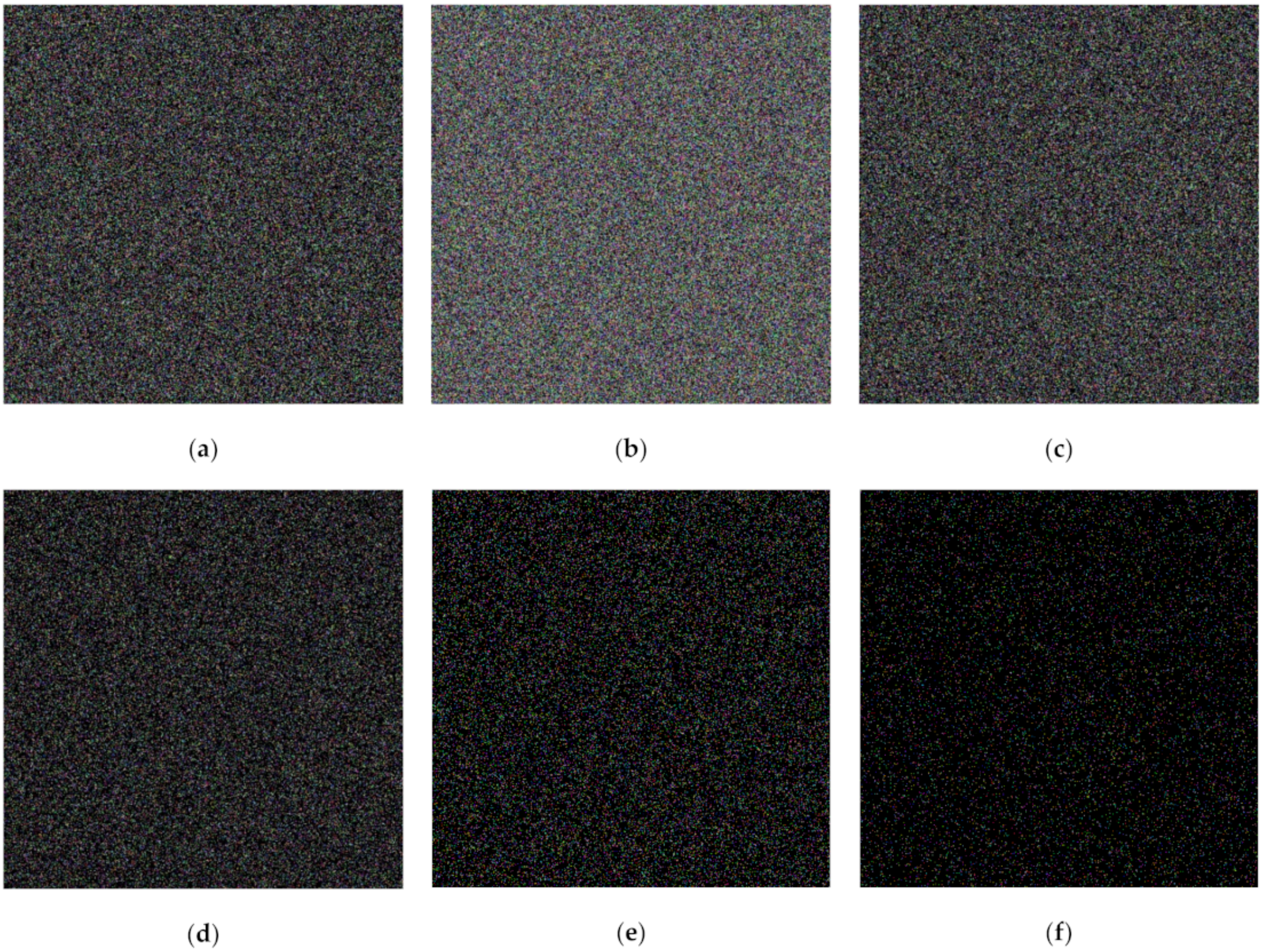


Figure 14. Shared images obtained from the secret image in the color format using the proposed XOR-based PVSS method: (**a–f**) $\{S_1, S_2, \ldots, S_6\}$.

Subsequently, we observed the quality of the recovered secret image after stacking several shared images using an XOR operation. In this experiment, we reconstructed the secret image by stacking two shared images until reaching n shared images. Figure 15 displays the recovered secret image obtained from the former scheme [15] and the proposed method for $n = 6$, while Figure 16 shows the results for $n = 5$. It can be observed from Figures 15 and 16 that the former scheme [15] and the proposed method satisfy the progressive constraint (i.e., the quality of the recovered secret image is increased if more shared images are involved and stacked with an XOR operation). Similarly to the binary and grayscale image cases, the proposed XOR-based PVSS method produces a good result whether the number of stacked shared images is odd or even, whereas the former scheme [15] cannot correctly yield the recovered secret image if the number of stacked shared images is odd. The proposed XOR-based PVSS overcomes the limitation of [15] with a simple approach. Thus, the proposed method delivers a promising result for binary, grayscale, and color images.

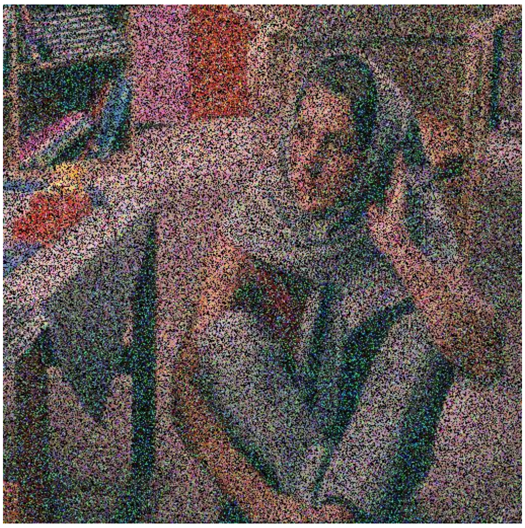
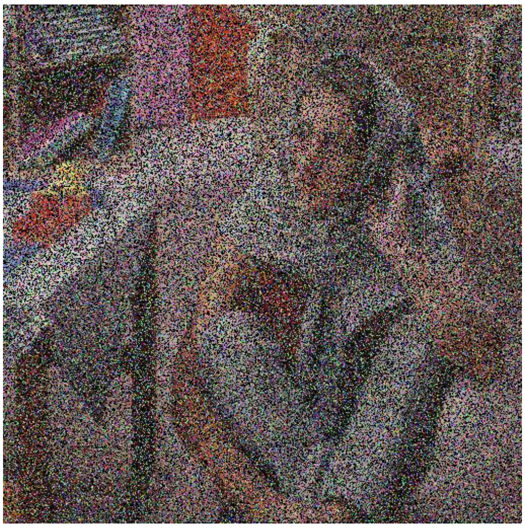
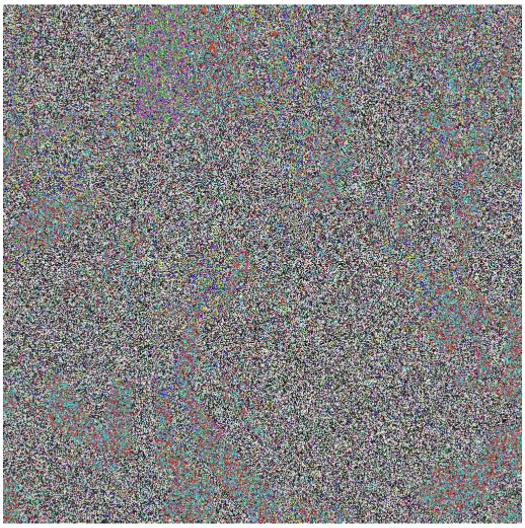
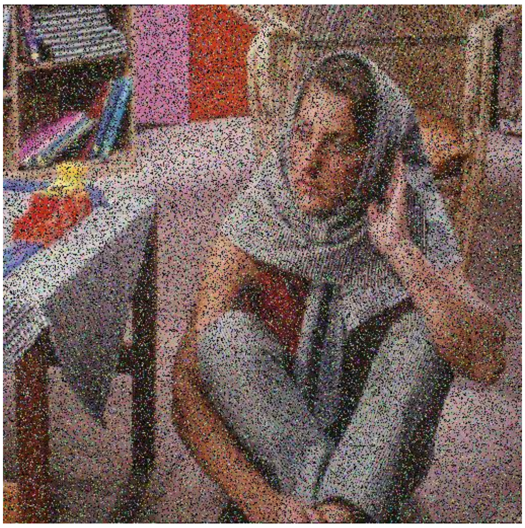
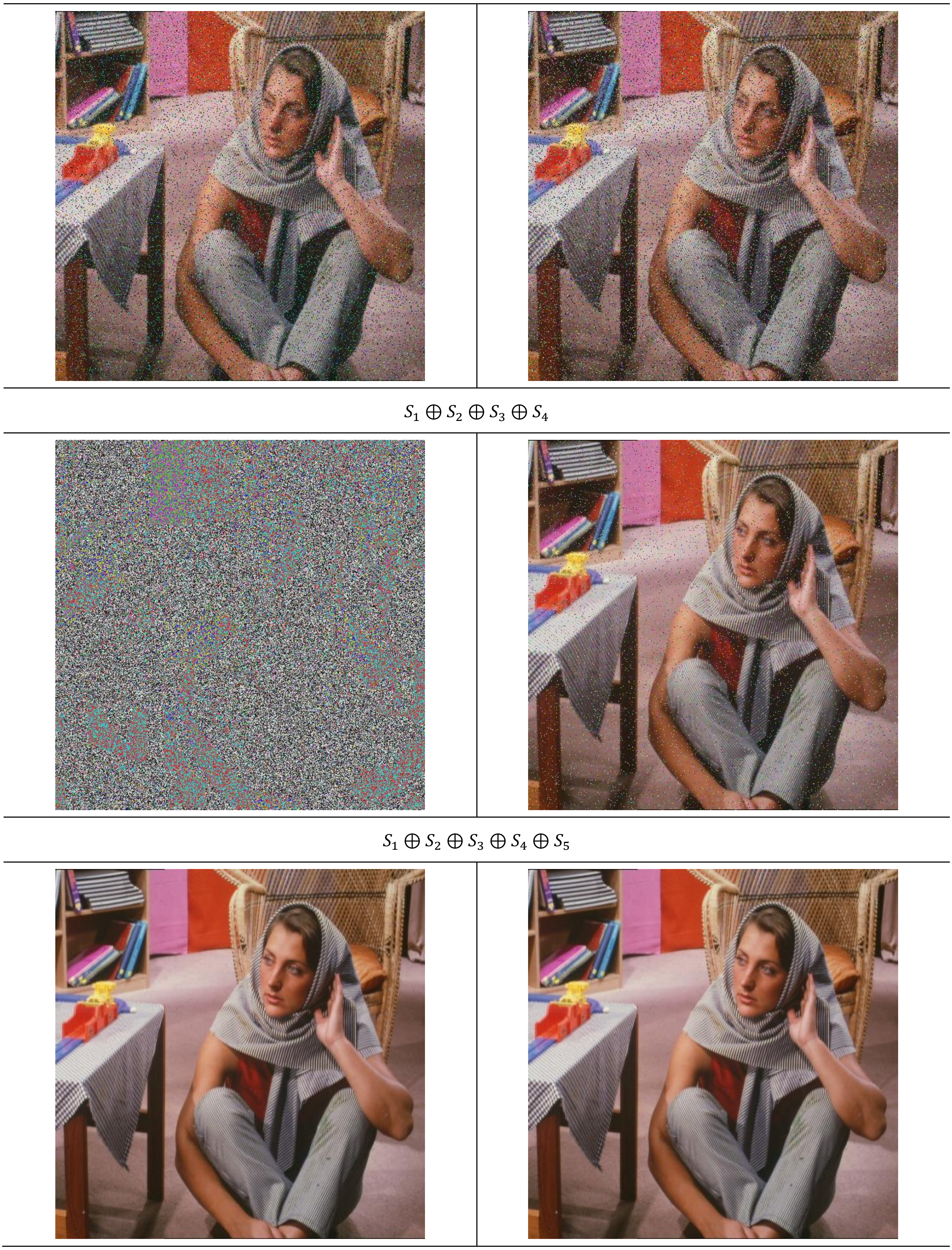
Former Scheme	**Proposed XOR-Based PVSS**
	
$S_1 \oplus S_2$	
	
$S_1 \oplus S_2 \oplus S_3$	

Figure 15. *Cont.*



$S_1 \oplus S_2 \oplus S_3 \oplus S_4$

$S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5$

Figure 15. *Cont.*

$S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5 \oplus S_6$

Figure 15. The results of stacking several shared images with $t = 2, 3, \ldots, 6$ and $n = 6$. The left and right columns are from the former scheme [15] and the proposed method, respectively.

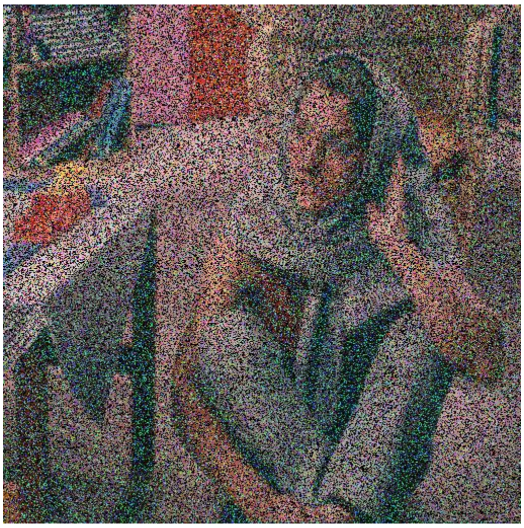
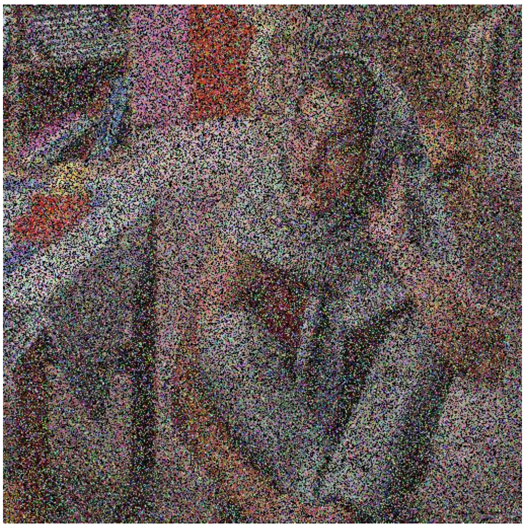
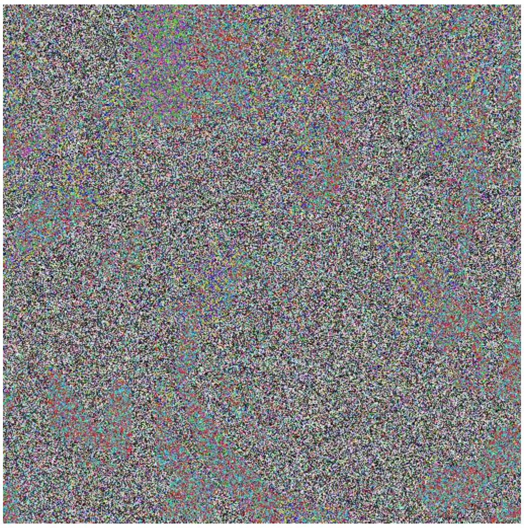
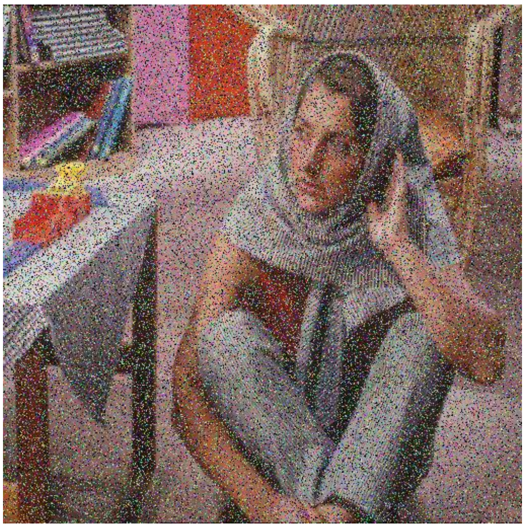
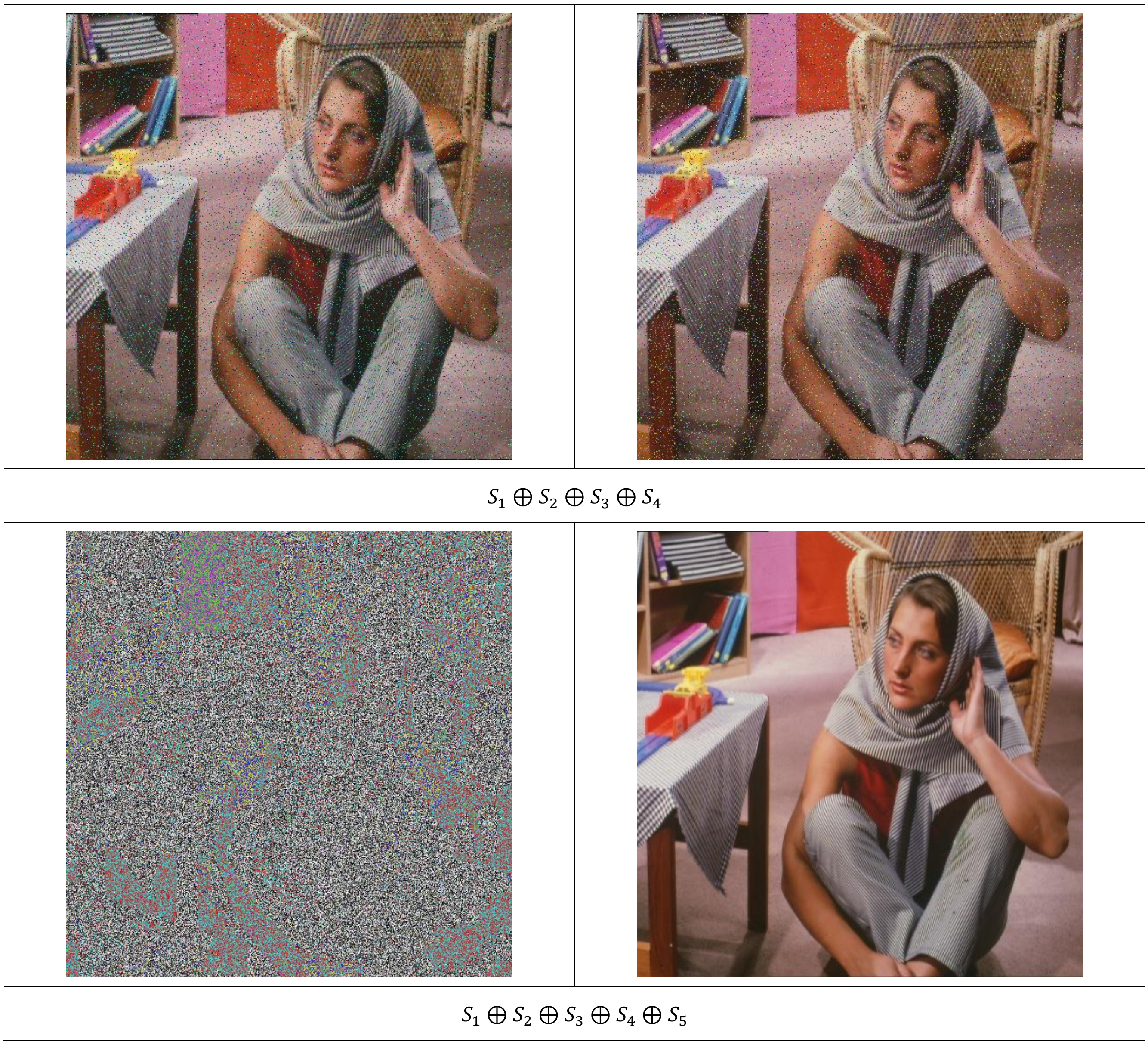
<table>
<tr><th>Former Scheme</th><th>Proposed XOR-Based PVSS</th></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">$S_1 \oplus S_2$</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">$S_1 \oplus S_2 \oplus S_3$</td></tr>
</table>

Figure 16. *Cont.*



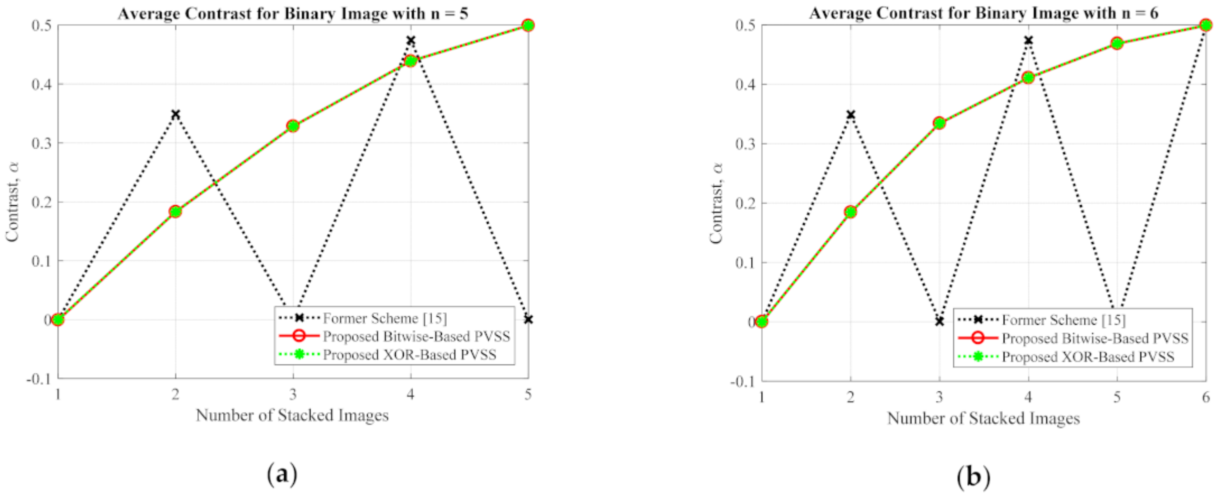
$S_1 \oplus S_2 \oplus S_3 \oplus S_4$

$S_1 \oplus S_2 \oplus S_3 \oplus S_4 \oplus S_5$

Figure 16. The results of stacking several shared images $t = 2, 3, \ldots, 5$ and $n = 5$. The left and right columns are the recovered secret image from the former scheme [15] and the proposed method, respectively.

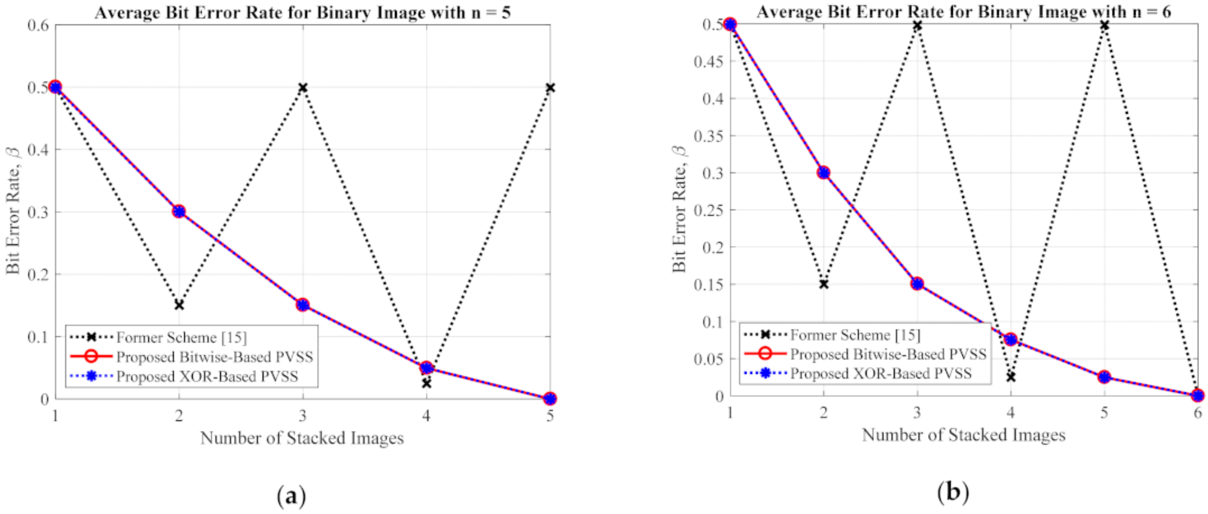
4.5. Performance Comparisons in Terms of Objective Image Quality Assessment

This subsection compares the performances of the proposed method and the former scheme [15] in detail based on an objective image quality assessment. For a binary image, the performance is simply measured and compared under two objective measurements (i.e., an average image contrast and average bit error rate). Herein, four secret images (as shown in Figure 5) were first converted into a set of shared images. The recovery process was subsequently conducted on these generated shared images to produce the recovered secret image. The averages of α and β were then computed for all recovered secret images. Figures 17 and 18 display the performance comparisons in terms of average α and average β, respectively, between the proposed method (with a bitwise and XOR-based approach) and the former scheme [15]. In this experiment, we set the number of shared images as $n = 5$ and $n = 6$. As shown in Figures 17 and 18, the former scheme [15] yields an unacceptable average α and β, respectively, if n or T is an odd number. However, the proposed method performs well for n or T whether they are odd or even numbers. In addition, the proposed method yields progressive results, indicating the increasing average value of α and the decreasing average value of β over different values of n or T.



(a) (b)

Figure 17. The average image contrast between the proposed method and the former scheme [15] of the binary secret image with (**a**) $n = 5$, and (**b**) $n = 6$.



(a) (b)

Figure 18. The average bit error rate between the proposed method and the former scheme [15] for a binary secret image with: (**a**) $n = 5$, and (**b**) $n = 6$.

For the grayscale and color images, the comparisons between the proposed method and the former scheme [15] are examined based on the average values of PSNR, SSIM, and MAE. We selected all secret images in grayscale and color spaces shown in Figures 9 and 13 as secret images. All secret images were then converted into a set of shared images. The recovered secret images were further computed by stacking several shared images using an XOR operation. The qualities of all of the recovered secret images were then measured in terms of average PSNR, SSIM, and MAE. Figures 19 and 20 display the performance comparisons for grayscale and color image, respectively. As depicted in these two figures, the former scheme [15] delivers unacceptable results if n or T is an odd number. The proposed method gives correct results whether n or T is an odd or an even number. The proposed method satisfies the progressive constraint for the PVSS task, as indicated by the improving PSNR and SSIM scores that result if more stacked images are utilized in the secret image reconstruction stage. It also gives a good result, decreasing the average MAE value if more stacked images are used to recover a secret image. However, the proposed method is a good candidate for implementing PVSS with adaptive priority and a perfect reconstruction process.

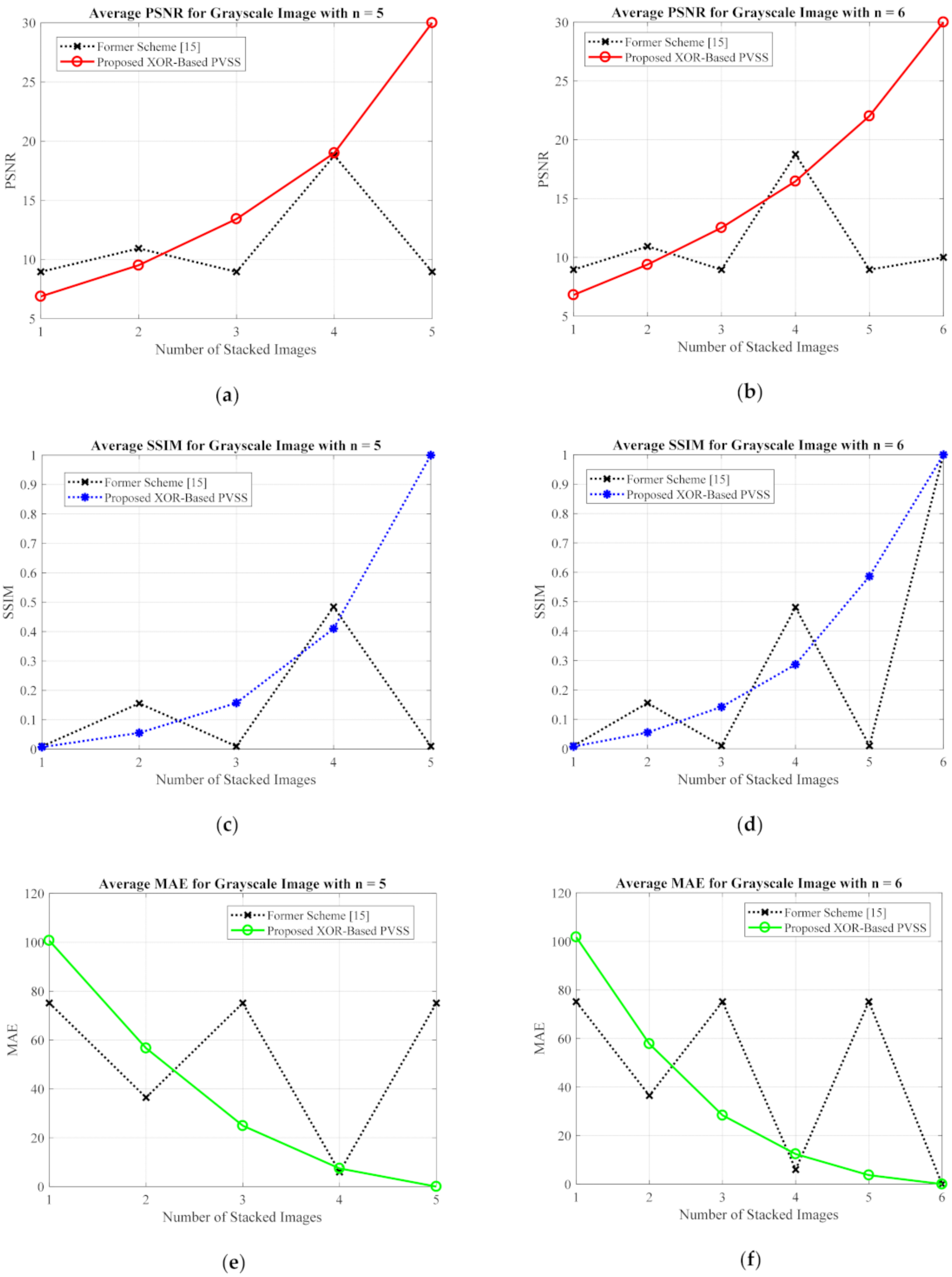
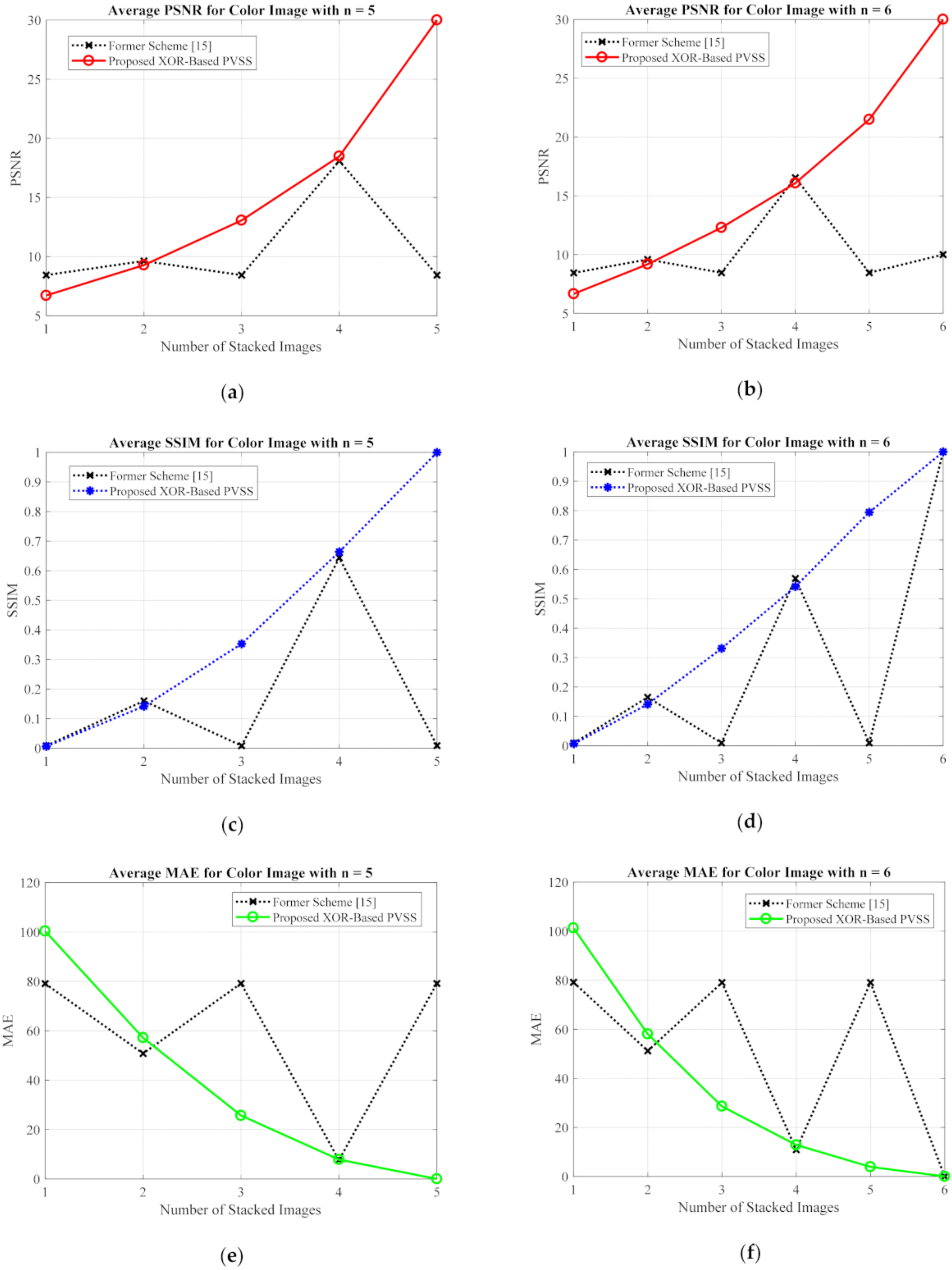


Figure 19. Comparisons between the proposed method and the former scheme [15] in terms of (**a**,**b**) PSNR, (**c**,**d**) SSIM, and (**e**,**f**) MAE values. The comparisons are conducted for a secret image in grayscale format.



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

(**c**) (**d**)

(**e**) (**f**)

Figure 20. Comparisons between the proposed method and former scheme [15] in terms of (**a**,**b**) PSNR, (**c**,**d**) SSIM, and (**e**,**f**) MAE values. Herein, the secret image is in a color format.

4.6. Comparison of Algorithm Aspects for the Proposed Method and Other Schemes

The proposed method and former scheme [15] works on a pixel-by-pixel basis in the shared image generation and secret image reconstruction processes. The computational times of these two methods completely depends on the image size. Let M and N be the width and height of an original secret image. The computational complexity for generating one shared image is $\mathcal{O}(MN)$ for both the proposed method and the former scheme [15]. In reality, the former scheme requires a slightly higher computational burden, since it involves more steps to be conducted in order to compute the shared image, compared to those required in the proposed method. However, the difference is not quite significant. The proposed method and the former scheme [15] need identical computational complexity in the secret image reconstruction process (i.e., $\mathcal{O}(MN)$). These two methods simply perform a stacking process with an XOR operation in order to reconstruct a secret image. However, the proposed method and the former scheme have am almost identical computational complexity, except in terms of the quality of the recovered secret image. Thus, the proposed method is a better choice for implementing a PVSS algorithm.

This subsection also reports the algorithm aspects between the proposed method and other competing schemes. Herein, we simply compared the proposed method with others PVSS schemes [9–15] based on the share style, encoding matrix, pixel expansion, and adaptive priority weight. Table 1 summarizes this comparison. This table shows that the proposed method is able to perform the PVSS task with an priority adaptive weight similar to that of [12,14,15]. The other schemes cannot utilize the priority adaptive weight in the recovery stage of the secret image. The former approach [15] is the most competitive candidate when compared to the proposed method in these terms. However, the proposed method reveals its superiority since it works regardless of whether the number of stacked shared images is odd or even. The former scheme [15] has a limitation when the number of stacked images is odd. In addition, the proposed method does not require the encoding matrix and pixel expansion in the secret image recovery step, meaning that it requires a lower amount of storage space. In addition, the proposed method generates a set of shared images in the form of a noise-like appearance. Thus, the content of the shared images cannot be easily distinguished from one to the other. The contents of each shared image cannot be easily recognized and perceived by human vision. At the end, the proposed method offers its benefit for the PVSS task with adaptive priority and a perfect reconstruction process for recovering a secret image.

Table 1. Comparisons between the proposed method and the former scheme in terms of algorithm aspects.

Method	Share Style	Encoding Matrix	Pixel Expansion	Adaptive Priority	Quality
Fang's Scheme [9]	Noise-Like Form	Require	Need	No	Lossless for n is even
Wang's Scheme [10]	Noise-Like Form	Require	Need	No	-
Hou's Scheme [11]	Noise-Like Form	Require	No	No	-
Hou's Scheme [12]	Noise-Like Form	Require	No	Adaptive Priority	Lossy
Lin's Scheme [13]	Friendly Appearance	No	No	No	Lossy
Yang's Scheme [14]	Noise-Like Form	Require	No	Adaptive Priority	Lossy
Former Scheme [15]	Noise-Like Form	No	No	Adaptive Priority	Lossy, if n is oddLossless, if n is even
Prasetyo's Scheme [16]	Noise-Like Form	No	No	No	Lossless for n is odd or even
Proposed Method	Noise-Like Form	No	No	Adaptive Priority	Lossless for n is odd or even

5. Conclusions

A simple approach for overcoming the limitation of the former PVSS with adaptive priority weight is presented in this paper. The proposed method is designed to satisfy the lossless constraint and adaptive priority weight required for the PVSS system. The proposed method exploits the bitwise-based and XOR-based techniques for generating a set of shared images. It achieves perfect reconstruction on a recovered secret image whether the number of stacked or collected images is odd or even. While this works for a binary image, the proposed method also works well for grayscale and color images. This superiority can be further applied and extended for video processing or other image processing applications. Thus, the proposed method can be considered and viewed as a strong PVSS alternative with a perfect reconstruction ability.

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

Funding: This work was supported by the Hibah Penelitian Fundamental (PF-UNS), Universitas Sebelas Maret (UNS), Indonesia, year 2020, under contract numbers 452/UN27.21/PN/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zarepour-Ahmadabadi, J.; Shiri-Ahmadabadi, M.; Latif, A. A cellular automata-based multi-stage secret image sharing scheme. *Multimed. Tools Appl.* **2018**, *77*, 24073–24096. [CrossRef]
2. Bharti, S.S.; Gupta, M.; Agarwal, S. A novel approach for verifiable (n, n) audio secret sharing scheme. *Multimed. Tools Appl.* **2018**, *77*, 25629–25657. [CrossRef]
3. Liu, Y.-N.; Zhong, Q.; Xie, M.; Chen, Z.-B. A novel multiple-level secret image sharing scheme. *Multimed. Tools Appl.* **2017**, *77*, 6017–6031. [CrossRef]
4. Guo, J.-M.; Riyono, D.; Prasetyo, H. Hyperchaos permutation on false-positive-free SVD-based image watermarking. *Multimed. Tools Appl.* **2019**, *78*, 29229–29270. [CrossRef]
5. Prasetyo, H.; Hsia, C.-H. Improved multiple secret sharing using generalized chaotic image scrambling. *Multimed. Tools Appl.* **2019**, *78*, 29089–29120. [CrossRef]
6. Guo, J.M.; Riyono, D.; Prasetyo, H. Improved Beta Chaotic Image Encryption for Multiple Secret Sharing. *IEEE Access* **2018**, *6*, 46297–46321. [CrossRef]
7. Prasetyo, H.; Guo, J.-M. A Note on Multiple Secret Sharing Using Chinese Remainder Theorem and Exclusive-OR. *IEEE Access* **2019**, *7*, 37473–37497. [CrossRef]
8. Yan, X.; Lu, Y. Contrast-improved visual secret sharing based on random grid for general access structure. *Digit. Signal Process.* **2017**, *71*, 36–45. [CrossRef]
9. Fang, W.-P.; Lin, J.-C. Progressive viewing and sharing of sensitive images. *Pattern Recognit. Image Anal.* **2006**, *16*, 632–636. [CrossRef]
10. Wang, R.-Z. Region Incrementing Visual Cryptography. *IEEE Signal Process. Lett.* **2009**, *16*, 659–662. [CrossRef]
11. Hou, Y.-C.; Quan, Z.-Y. Progressive Visual Cryptography with Unexpanded Shares. *IEEE Trans. Circuits Syst. Video Technol.* **2011**, *21*, 1760–1764. [CrossRef]
12. Hou, Y.-C.; Quan, Z.-Y.; Tsai, C.-F. A privilege-based visual secret sharing model. *J. Vis. Commun. Image Represent.* **2015**, *33*, 358–367. [CrossRef]
13. Lin, C.-H.; Lee, Y.-S.; Chen, T.-H. Friendly progressive random-grid-based visual secret sharing with adaptive contrast. *J. Vis. Commun. Image Represent.* **2015**, *33*, 31–41. [CrossRef]
14. Yang, C.-N.; Liao, J.-K.; Wang, D.-S. New privilege-based visual cryptography with arbitrary privilege levels. *J. Vis. Commun. Image Represent.* **2017**, *42*, 121–131. [CrossRef]
15. Chao, H.-C.; Fan, T.-Y. Random-grid based progressive visual secret sharing scheme with adaptive priority. *Digit. Signal Process.* **2017**, *68*, 69–80. [CrossRef]
16. Prasetyo, H.; Hsia, C.-H. Lossless progressive secret sharing for grayscale and color images. *Multimed. Tools Appl.* **2019**, *78*, 24837–24862. [CrossRef]