symmetry

MDPI

Article

A Large Payload Data Hiding Scheme Using Scalable Secret Reference Matrix

Jason Lin [1], Chia-Wei Tsai [2], Chun-Wei Yang [3,*] and Kuan-Hung Liu [1]

[1] Department of Computer Science and Engineering, National Chung Hsing University, No. 145, Xingda Road, South District, Taichung 40227, Taiwan; jasonlin@nchu.edu.tw (J.L.); g107056191@mail.nchu.edu.tw (K.-H.L.)
[2] Department of Computer Science and Information Engineering, National Taitung University, No. 369, Section 2, University Road, Taitung 95092, Taiwan; cwtsai@nttu.edu.tw
[3] Master Program for Digital Health Innovation, College of Humanities and Sciences, China Medical University, No. 100, Section 1, Jingmao Road, Beitun District, Taichung 406040, Taiwan
* Correspondence: cwyang@mail.cmu.edu.tw

Abstract: When imperceptibility is an issue, data-hiding techniques typically become limited to small payloads. In this paper, a novel data-hiding scheme is proposed that allows embedding large payloads while maintaining high stego-image quality. The scheme utilizes an $N \times N$ sub-block for constructing a reference matrix as a secret key that allows the symmetric embedding and extraction of secret data from a grayscale cover image, where N is a positive integer greater than or equal to two. With proper modification, the proposed method can be directly converted to a symmetric cryptosystem. For cases with acceptable stego-image quality (i.e., PSNR > 34), the experimental results showed that the proposed method can embed, on average, four bits per pixel (bpp), a higher hiding capacity than in previous works, and also offers the ability to adjust the capacity by varying N. The number of solutions for a sub-block reaches the factorial of N^2, guaranteeing the security of data embedding and extraction. In addition, the proposed method has low computational complexity and can be implemented in a straightforward manner.

Keywords: flexible embedding capacity; reference matrix; steganography

check for updates

Citation: Lin, J.; Tsai, C.-W.; Yang, C.-W.; Liu, K.-H. A Large Payload Data Hiding Scheme Using Scalable Secret Reference Matrix. *Symmetry* **2022**, *14*, 828. https://doi.org/10.3390/sym14040828

Academic Editors: Kuo-Hui Yeh and Alexander Shelupanov

Received: 15 March 2022
Accepted: 12 April 2022
Published: 15 April 2022

1. Introduction

Technological advancements and near-ubiquitous connectivity have fueled Internet communications. Although cryptography can be used for encrypting and transmitting secret messages on the Internet, encryption has attracted the attention of hackers. Upon identifying encrypted data, attackers may intercept and attempt to decode secret messages. Currently, considerable research has been conducted on data hiding as an alternative to avoiding easy detection, with the primary goal of ensuring secret communication via the Internet. A secret message can be embedded into a cover medium using a concealing algorithm to create a stego-medium. The secret message carried by the stego-medium can then be transmitted through the Internet without being detected, even if it is intercepted. Digital images are commonly used as cover media for data hiding; the resulting images are known as stego-images.

There are two domains of data hiding in image steganography: the spatial [1–20] and frequency domains [21–26]. In the spatial domain, secret data are embedded by adjusting the least significant bit (LSB) of each pixel. A high embedding capacity can be achieved, and embedding and extraction procedures can be easily implemented. However, stego-images in the spatial domain are easily attacked by steganalysis. In the frequency domain, secret data are embedded in the frequency coefficients of the image, following the discrete cosine transform (DCT). The advantage of the frequency domain is its robustness. In general, secret data are difficult to detect, although only small quantities can be embedded owing to image distortion. This study focuses on the data-hiding mechanism in the spatial domain.

The data-hiding scheme in this work is designed considering two main factors: (1) the embedding capacity (the number of covered bits per pixel) and (2) the image's visual quality (image distortion). Some existing works have focused on designing reversible data-hiding schemes [12–20] to allow full recovery of cover images from stego-images. The embedding capacity of those methods is limited, owing to the need to store information related to the recovery of the original images. For irreversible data-hiding schemes [1–11], achieving high embedding capacity and visual quality is desirable. However, it is difficult to ensure both of these metrics at the same time. Increasing the embedding capacity increases the stego-image distortion, and vice versa. Therefore, the trade-off between embedding capacity and visual quality depends on the user and on the application context. For example, if embedded information is used for authenticating the copyright of cover images, as is the case in digital watermarking [21–23,25,26] or image authentication [27], then perhaps a larger acceptable distortion can be tolerated. However, if the goal is to protect the secrecy of the embedded information, then the visual quality should be such that the transmitted information is undetectable, at least to the human visual system.

In the spatial domain, the simplest LSB-based method amounts to replacing the least significant bits of the covering pixels with bits of the binary secret message. This simple method can be easily implemented for concealing confidential data but is prone to common statistical attacks [28–30]. In 2006, Mielikainen [2] proposed LSB matching, a revised LSB-like data-hiding method. In the LSB matching approach, pairs of pixels are grouped into basic embedded units. The least significant bits of a single pixel per pair are then modified to embed two bits. However, Mielikainen's method did not fully explore all special modification cases. Zhang et al. [3] proposed an exploiting-modification direction (EMD) method, expanding Mielikainen's method and exploiting special cases. The EMD method dynamically groups N pixels of the cover image to conceal $(2N + 1)$-base numeral system digits, where N is a positive integer. Although this minimizes distortion, the embedding capacity of this method is limited. More recently, an LSB-based approach was proposed by Singh [10], using a novel adaptive pixel value differencing (PVD) scheme along with LSB embedding to achieve a high embedding capacity while retaining the visual quality of the cover image.

In 2008, Chang et al. [4] introduced a novel family of data-hiding schemes that uses Sudoku-based reference matrices for concealing and extracting secrets. Their method utilizes the properties of Sudoku [31] and applies an approach that uses pixel-pairs from the cover image as coordinates for matching their secrets in the digital format within a table. Lin et al. [5] subsequently proposed an approach for improving visual quality, inspired by the Sudoku method of Chang et al. [4]. Recently, Wu et al. [6] used three-dimensional reference tables for improving security and for customizing embedding capacity. Chang et al. [7] also freely adjusted the embedding in a one-dimensional vector. Hsiao et al. [11] extended the 9×9 Sudoku matrix to a 16×16 matrix for increasing the possibility of its solution and used a two-layer reference matrix for embedding additional information. The present study focuses on two-dimensional reference matrices. The main goal is to address the idea inspired by combining the EMD [3] and Sudoku [4] methods to improve the visual quality of the Sudoku method and to improve its high embedding capacity and security properties. We present a scalable secret reference matrix (SSRM), a secret matrix composed of many $N \times N$ submatrices. The procedure of the proposed method first converts binary secret data into secret digits in the N^2-base numeral system, and then modifies the values of the cover pixel-pairs for concealing the secret digits. Moreover, the SSRM can be also used as a secret key to symmetrically encrypt and decrypt the secret message via a cover image.

2. Related Work

In this section, since the proposed method is mainly based on the EMD method [3] and the Sudoku method [4], we briefly discuss the most relevant works based on these two approaches.

The pioneering work of EMD can date back to an optimal strategy (LSB-re) proposed by Jarno Mielikainen [2] to improve the visual quality of stego-images in the classic LSB-based replacement approach when embedding one bit per pixel (bpp) in a cover image. The LSB-re alters the second least significant bit if the pixel value after the LSB replacement has exceeded a certain predefined range to reduce its distortion. Zhang et al. [3] subsequently proposed an EMD method that described the generalization of all cases in [2] as well as produced the first data-hiding scheme that utilized a reference matrix. The main idea of EMD is to hide a $(2N+1)$-base digit d into n cover pixels, where d is an integer in the 0–$2N$ range, and N is a non-negative integer. The worst-case implication of embedding a $(2N+1)$-base digit in N cover pixels is that only one out of N pixels should be increased or decreased by one. In 2008, Chang et al. [4] proposed a data-hiding scheme using a new reference matrix based on a 9 × 9 Sudoku grid, for improving the embedding capacity of the EMD method. In the Sudoku method, each pixel-pair of a cover image can hide a 9-base digit, in contrast to the EMD method, which can only embed a 5-base digit. Chang et al. used the Sudoku solution for creating a reference matrix to guide the modification of the cover image. The number of possible solutions to a 9 × 9 Sudoku problem is approximately 6.671×10^{21}, showing that the security of the Sudoku method is higher than that of the EMD method.

In early 2016, Wu et al. [6] proposed a magic cube–based (MCB) scheme to improve the efficiency of the Sudoku method of Chang et al. [4] The MCB method utilizes a $2^R \times 2^R \times 2^R$ magic cube for $1 \leq R \leq 8$ to embed and extract the secret data in base 2^{3R} numeral secret digits where the embedding capacity is R bpp. A three-dimensional reference table can be constructed by the magic cube with distinct values of 0 to $2^{3R}-1$ randomly assigned to all magic cubes. That is, every sub-cube in the magic cube stands for a different number. All pixels in the cover image are modified by substitution according to the spatial coordinates mapped from the numeral secret digits of the secret data. Later in 2017, Chang et al. [7] proposed another data-hiding scheme based upon a permutation vector (PV). The PV is concatenated repeatedly to construct a one-dimensional reference matrix in order to improve the visual quality of an embedding capacity less than 1.5 bpp. Recently, Hsiao et al. [11] proposed two methods using the 16 × 16 Sudoku matrix and the double-layer magic matrix, respectively. For the first method, the 16 × 16 Sudoku matrix is an extended version of the 9 × 9 Sudoku matrix in [4]. Note that the numbers in each column, row, and diagonal of the 4 × 4 magic matrix must sum to 30. Hsiao et al. designed a double-layer magic matrix (DLMM) to improve embedding capacity compared to the 16 × 16 Sudoku matrix. As a result, the bpp of the embedding capacity increases from 2 to 3. The data-hiding scheme proposed by Hsiao et al. also modified the pixels through a reference matrix RM to complete the embedding of a secret message. A double-layer magic matrix is constructed to help embed the secret message in a base 64 numeral system.

3. The Proposed SSRM Method

The proposed method requires a 256 × 256 special scalable secret reference matrix (SSRM) as a key map. This key map guides the cover image to modify its pixels to imply the secret message indirectly. An overall flow diagram of the developed system is shown in Figure 1. The same key map is used to extract the hidden secret message as well. The proposed method is introduced in four subsections. Section 3.1 describes how to create an SSRM and its characteristics. Section 3.2 proves the characteristics of SSRM. Finally, Sections 3.3 and 3.4 give details of the embedding process and extraction process, respectively.

3.1. The Generating Process of SSRM

In this section, a 256 × 256 SSRM is proposed to be generated as a key map to guide pixel modification while in the embedding phase. It will also be required when extracting the secret data from the stego-image. SSRM is a special matrix that consists of two conditions as follows. First, the SSRM is filled with all digits 0 ~ $\left(N^2-1\right)$, where N is a positive integer in the range of $1 < N \leq 256$ (i.e., N is greater than 1 because the signal in image processing has at least two digits, '$0'$ and '$1'$, to represent the multimedia data. On the other hand, the maximum of N can only be 256 due to the full size of the SSRM being 256 × 256). Secondly, any $N \times N$ square window in SSRM has all digits $0 \sim \left(N^2 - 1\right)$ in it. Suppose an empty set of a 256 × 256 matrix M has been generated, and an $N \times N$ table T has been created with all digits $0 \sim \left(N^2 - 1\right)$ randomly filled in using a random number generator along with an arbitrary seed as a key. Then a matrix such as the SSRM can be generated by the following steps.

Step 1: Let $T_{i,j}$ represents the location of T on SSRM where i and j are the row and column of the upper-left most corner of T on SSRM, respectively.

Step 2: From left to right, slide $T_{i,j}$ horizontally one column. Then, $T_{i,j+1}$ will be $T_{i,j}$'s new position, where the leftmost column elements of T are slid to the rightmost column of $T_{i,j+1}$, as an example of $N = 5$ is shown in Figure 2a.

Step 3: Repeat Step 2 until the $N \times N$ window scans to the right edge of the SSRM.

Step 4: Locate the $N \times N$ sliding window back to position (0,0) of the SSRM (i.e., the left end of the SSRM).

Step 5: From top to bottom, vertically slide the $N \times N$ window T one row. Make the new position of the $N \times N$ window $T_{i+1,j}$ and complement $T_{i+1,j}$'s bottommost elements with $T_{i,j}$'s topmost elements. An example of $N = 5$ is shown in Figure 2b.

Step 6: Repeat Step 3 to Step 6 until M is filled with digits, as shown in Figure 3.

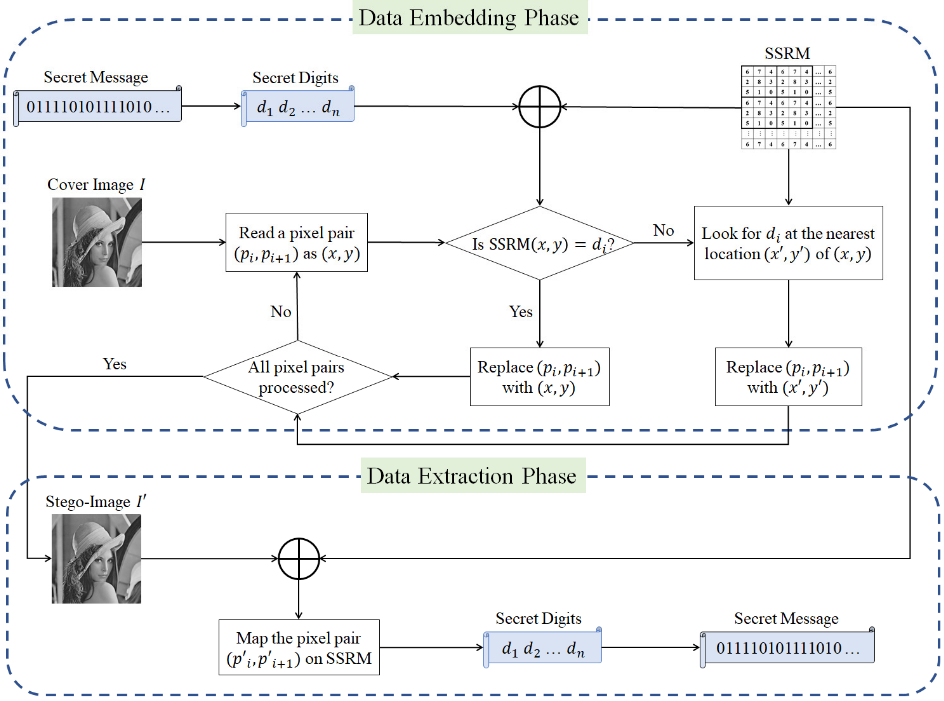
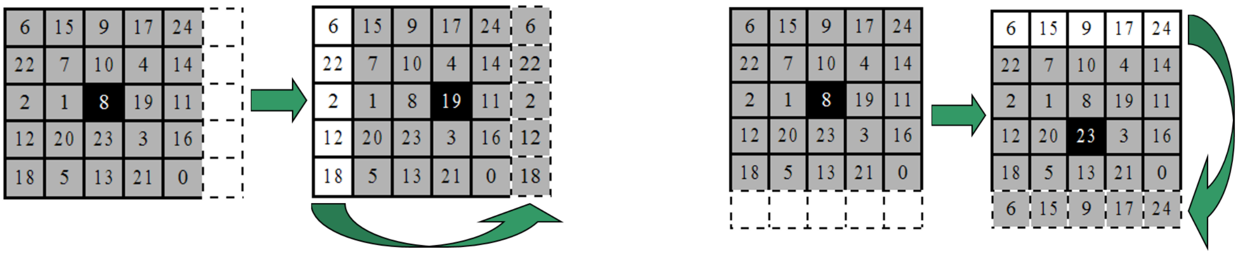

Data Embedding Phase
Secret Message
011110101111010…
Secret Digits
$d_1\ d_2 \ldots d_n$
SSRM
Cover Image I
Read a pixel pair (p_i, p_{i+1}) as (x, y)
Is SSRM$(x, y) = d_i$?
No
Look for d_i at the nearest location (x', y') of (x, y)
Yes
No
Yes
All pixel pairs processed?
Replace (p_i, p_{i+1}) with (x, y)
Replace (p_i, p_{i+1}) with (x', y')
Data Extraction Phase
Stego-Image I'
Map the pixel pair (p'_i, p'_{i+1}) on SSRM
Secret Digits
$d_1\ d_2 \ldots d_n$
Secret Message
011110101111010…

Figure 1. The overall flow diagram of the developed SSRM system.



6	15	9	17	24
22	7	10	4	14
2	1	8	19	11
12	20	23	3	16
18	5	13	21	0

6	15	9	17	24	6
22	7	10	4	14	22
2	1	8	19	11	2
12	20	23	3	16	12
18	5	13	21	0	18

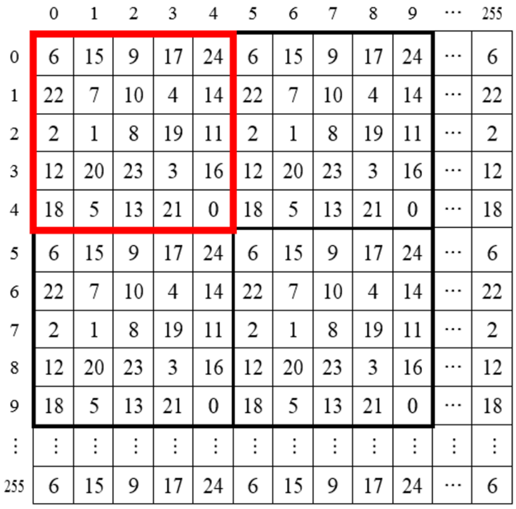
(**a**)

6	15	9	17	24
22	7	10	4	14
2	1	8	19	11
12	20	23	3	16
18	5	13	21	0

6	15	9	17	24
22	7	10	4	14
2	1	8	19	11
12	20	23	3	16
18	5	13	21	0
6	15	9	17	24

(**b**)

Figure 2. Steps for generating a 5×5 table–based SSRM. (**a**) Slides horizontally, (**b**) Slides vertically.



	0	1	2	3	4	5	6	7	8	9	⋯	255
0	6	15	9	17	24	6	15	9	17	24	⋯	6
1	22	7	10	4	14	22	7	10	4	14	⋯	22
2	2	1	8	19	11	2	1	8	19	11	⋯	2
3	12	20	23	3	16	12	20	23	3	16	⋯	12
4	18	5	13	21	0	18	5	13	21	0	⋯	18
5	6	15	9	17	24	6	15	9	17	24	⋯	6
6	22	7	10	4	14	22	7	10	4	14	⋯	22
7	2	1	8	19	11	2	1	8	19	11	⋯	2
8	12	20	23	3	16	12	20	23	3	16	⋯	12
9	18	5	13	21	0	18	5	13	21	0	⋯	18
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
255	6	15	9	17	24	6	15	9	17	24	⋯	6

Figure 3. An example of 5 × 5 table–based SSRM.

The SSRM process can be summarized by tiling an $N \times N$ table to a 256×256 matrix. By using the SSRM, it costs two pixels each time to map on the element of the SSRM (i.e., the SSRM is a two-dimensional matrix), which means that every two pixels can embed an N-base digit. Therefore, the embedding capacity increases by adjusting the value of N. On the other hand, the distortion also increases by growing the size of the $N \times N$ table (i.e., the correct element could be found far from the located element, whereby the difference between the new pixel-pair and the original pixel-pair could be large).

3.2. The Correctness of the Property in SSRM

The 256×256 SSRM has a property that every coordinate's element can find other base N^2 digits through its surrounding neighbor elements. To prove this phenomenon, a 256×256 SSRM base on the 3×3 table ($N = 3$) is given as an example for illustration. First, create a 3×3 table T filled up randomly with unrepeated 9 base digits 0–8. Define three sets SE_1, SE_2, and SE_3 that respectively include the elements of T in columns 0, 1, and 2, where $SE_1 = \{6,\ 4,\ 2\}$, $SE_2 = \{5,\ 7,\ 1\}$, $SE_3 = \{3,\ 0,\ 8\}$. Secondly, suppose a 3×3 sliding window, SW, that attaches to table T. Then, the three columns of SW are WC_1, WC_2, WC_3, respectively. While SW is sliding, the new three columns of SW can be represented in general form as $WC_1 = 0 + \Delta$, $WC_2 = 1 + \Delta$, $WC_3 = 2 + \Delta$ (i.e., variable Δ is defined as the sliding distance of SW, where Δ is a positive integer at the range of $0 \leq \Delta < 256$. The fact of the SSRM can be proven by the following mathematical induction.

Step 1: When $\Delta = 0$, then $WC_1 = 0 + 0 = 0$, $WC_2 = 1 + 0 = 1$, $WC_3 = 2 + 0 = 2$, where the element sets $\{WC_1\} \cup \{WC_2\} \cup \{WC_3\} = \{0, 1, \ldots, 8\} = \{T\}$ as shown in Figure 4a. SW has not moved.

Step 2: Let $\Delta = 1$, then $WC_1 = 0 + 1 = 1$, $WC_2 = 1 + 1 = 2$, $WC_3 = 2 + 1 = 3$. Because of $\{WC_3\}$ is an empty set, the vacant elements is $\{6,\ 4,\ 2\} = SE_1$. Therefore, the column $0'$s elements have to complement column $3'$s empty set, where SE_1 is assigned to $\{WC_3\}$ to contain the three sets of $\{WC_1\} \cup \{WC_2\} \cup \{WC_3\} = \{0, 1, \ldots, 8\} = \{T\}$ as shown in Figure 4b. SW has moved by 1.

Step 3: When $\Delta = k$, then $WC_1 = k$, $WC_2 = k + 1$, $WC_3 = k + 2$. Suppose this consists of the expression $\{WC_1\} \cup \{WC_2\} \cup \{WC_3\} = \{T\}$ as shown in Figure 4c. SW has moved by k.

Step 4: Let $\Delta = k + 1$, then $WC_1 = 0 + (k + 1) = k + 1$, $WC_2 = 1 + (k + 1) = k + 2$, $WC_3 = 2 + (k + 1) = k + 3$. Since $\{WC_3\}$ is an empty set, the lacking elements should be at column k. Therefore, assign column k's elements to $\{WC_3\}$ as shown in Figure 4d to consist of $\{WC_1\} \cup \{WC_2\} \cup \{WC_3\} = \{T\}$. SW has moved by $k + 1$.

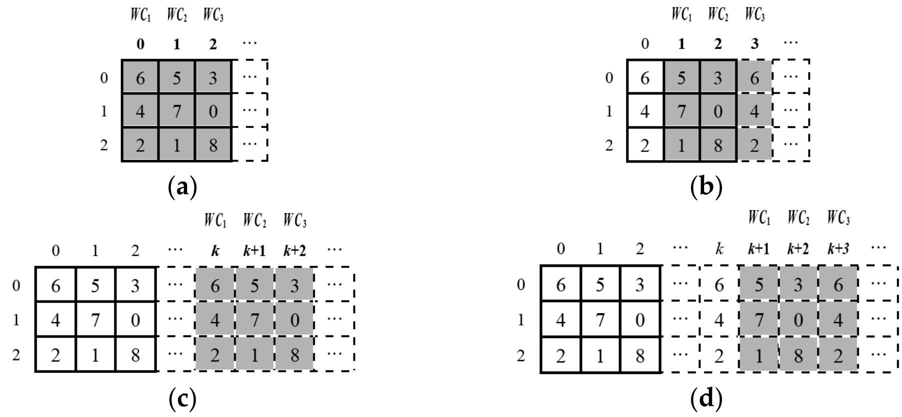


Figure 4. The inductive proof of the property in SSRM. (**a**) SW sliding with distance Δ = 0, (**b**) SW sliding with distance Δ = 1, (**c**) SW sliding with distance Δ = *k*, (**d**) SW sliding with distance Δ = *k* + 1.

From the deduction above, the horizontal sliding is proven to have the characteristic that every location can find other digits around its neighbors. On the other hand, the same proof method can be used while sliding vertically. Thus, the characteristic of SSRM is proven to be accurate.

3.3. The Embedding Section

The main procedure of the embedding process can be summarized as the upper part of Figure 1. First, an $H \times W$ grayscale cover image $I = \{p_i \mid 0 \leq p_i < 256, 0 \leq i < (H \times W)\}$, where H and W represent the height and width of the cover image, respectively. Any possible pairing technique can be used to group every two pixels on the cover image. Secondly, prepare a size 256 × 256 SSRM that contains digits from 0 to $\left(N^2 - 1\right)$, where N is a positive integer. Suppose the embedded secret message is a bit stream $S = \left\{b_j \mid 0 \leq j \leq m-1,\ b_j \in [0,1]\right\}$, where m represents the bits length of S. Then S will be divided into segments of α bits. Each α bit can be converted into an N^2-base digit. The value of α can be calculated by $\left|\log_2 N^2\right|$. If the digit converted by α bits is greater than N^2, catch an $\alpha - 1$ bits segment to range the conversion in 0–N^2 digits. After the bit stream is converted into N^2-base secret digits, each pair of pixels will be guided by the SSRM for modification to imply a secret digit.

Suppose an N^2-base digit, d, is about to embed into a pixel-pair (p_i, p_{i+1}). Locate the pixel-pair (p_i, p_{i+1}) as a coordinate on the SSRM. The element of the coordinate (p_i, p_{i+1}) on SSRM is $M(p_i, p_{i+1})$. If $M(p_i, p_{i+1}) = d$, then embed d without any modification on (p_i, p_{i+1}). If $M(p_i, p_{i+1}) \neq d$, then search for the neighbor elements to find digit d. To find the modification that maintains a higher quality of PSNR, the Euclidean distance should be used as a measurement standard. The Euclidean distances $ED(\cdot)$ for two locations $v = (p_i, p_{i+1})$ and $u = \left(p'_i, p'_{i+1}\right)$ in the SSRM can be calculated by (1).

$$ED(v,u) = \sqrt{\left(p'_i - p_i\right)^2 + \left(p'_{i+1} - p_{i+1}\right)^2} \tag{1}$$

The reason for selecting Euclidean distance over other distance metrics, such as city-block distance or chessboard distance, is because Euclidean distance tends to have a more precise estimation when distance is actually the value differences between two pixel-pairs. That is, a diagonal step (alter each of the two pixels by one) is better than two vertical or horizontal steps (alter one of the two pixels by two) in terms of PSNR. For example, given four coordinates in a two-dimensional space: $u = (p_i, p_{i+1})$, $v = (p_i + 1, p_{i+1} + 1)$, $s = (p_i + 1, p_{i+1})$, and $t = (p_i + 2, p_{i+1})$. The city-block distance tends to overestimate the cost of a diagonal step since it estimates that the distance between u and v is the same as that between u and t. On the contrary, the chessboard distance tends to underestimate the cost of a diagonal step since it estimates that the distance between u and v is the same as that between u and s. However, the distortions caused by the substitution of u is $t > v > s$. Therefore, only the Euclidean distance that applies the straight-line distance is more consistent with the cost of value differences between two pixel-pairs.

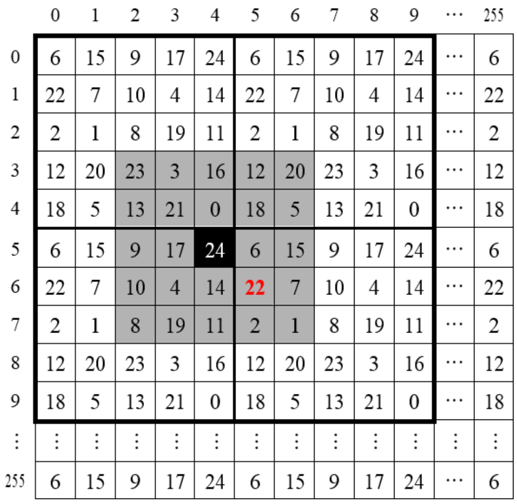
Before finding the candidate element of (p_i, p_{i+1}), an $N \times N$ size block will be marked as a searching area (SA). Try to locate (p_i, p_{i+1}) on the center of the SA as possible. That way, the searched neighborhood elements can nearly stay around the coordinate (p_i, p_{i+1}), where a suitable element with the closest Euclidean distance can be found. Due to the fact that the candidate element in the SA can only exist once, which is also the optimal one to have the closest Euclidean distance, the search need only be in the SA. Suppose (x, y) is the beginning location for searching the SA. The pseudo code of the search procedure is described in Algorithm 1 as follows.

Algorithm 1 The step by step searching process of the SSRM algorithm.

```
• Initialize $S$ to be an **empty** set
• Initialize the radius $r = \frac{N}{2}$, and the complement $\bar{r} = 255 - r$
  if $N\ (mod\ 2) \equiv 0$, then $r = r - 1$
  end if
• Initialize $x$ and $y$ by the following cases:
  if $r \le p_i \le \bar{r}$ and $r \le p_{i+1} \le \bar{r}$ then $x = p_i - r, y = p_{i+1} - r$
  else if $p_i < r$ and $p_{i+1} < r$ then $x = y = 0$
  else if $p_i > \bar{r}$ and $p_{i+1} > \bar{r}$ then $x = y = 255 - n + 1$
  else if $p_i < r$ and $r \le p_{i+1} \le \bar{r}$ then $x = 0, y = p_{i+1} - r$
  else if $r \le p_i \le \bar{r}$ and $p_{i+1} < r$ then $x = p_i - r, y = 0$
  else if $p_i > \bar{r}$ and $0 \le p_{i+1} \le 255$ then $x = 255 - n + 1, y = p_{i+1} - r$
  else if $r \le p_i \le \bar{r}$ and $p_{i+1} > \bar{r}$ then $x = p_i - r, y = 255 - n + 1$
  else if $p_i > \bar{r}$ and $p_{i+1} < r$ then $x = 255 - n + 1, y = 0$
  else $p_i < r$ and $p_{i+1} > \bar{r}$ then $x = 0, y = 255 - n + 1$
  end if
• Collect all the elements of $SA$ to $E$ by the following loop:
  for $k = 0 \ldots (N-1)$ do
      for $l = 0 \ldots (N-1)$ do
          $S = S \cup \{M(x+k, y+l)\}$;
  end for
• return $S$;
```

The search starts from the element on the leftmost column and topmost row, and sequentially continues from left to right, top to bottom. Suppose the optimal pixel-pair (p'_i, p'_{i+1}) is found to have $M(p'_i, p'_{i+1}) = d$. The value of pixel-pair (p_i, p_{i+1}) is switched to (p'_i, p'_{i+1}) as the stego pixel-pair, implying an N^2-base digit, d.

Consider the following example to illustrate the embedding process. Suppose embedding a segment of binary streams $(10110)_2$ into the pixel-pair $(5, 4)$. First, turn $(10110)_2$ to a 25-base digit 22_{25}, and create a 256 × 256 SSRM that contains 0–24 digits as shown in Figure 5. Secondly, map the pixel-pair on the SSRM in Figure 5 to match the element with 22_{25}. The location $(5, 4)$ of SSRM is discovered as $M(5, 4) = 24_{25} \neq 22_{25}$. Therefore, a big Oh $O(5 \times 5)$ searching area (SA) is marked in gray, as shown in Figure 5. The process of searching is defined as initializing the first searching element as the location of the leftmost column and topmost row in the searching area. Follow the sequential index from left to right, top to bottom in SA to search for the digit 22_{25}, as shown in Figure 5. The digit 22_{25} is, however, found at $(6, 5)$ on the SSRM, as shown in Figure 5. Substitute $(5, 4)$ with $(6, 5)$ as a means of hiding 22_{25} in pixel-pair $(5, 4)$.



	0	1	2	3	4	5	6	7	8	9	⋯	255
0	6	15	9	17	24	6	15	9	17	24	⋯	6
1	22	7	10	4	14	22	7	10	4	14	⋯	22
2	2	1	8	19	11	2	1	8	19	11	⋯	2
3	12	20	23	3	16	12	20	23	3	16	⋯	12
4	18	5	13	21	0	18	5	13	21	0	⋯	18
5	6	15	9	17	24	6	15	9	17	24	⋯	6
6	22	7	10	4	14	22	7	10	4	14	⋯	22
7	2	1	8	19	11	2	1	8	19	11	⋯	2
8	12	20	23	3	16	12	20	23	3	16	⋯	12
9	18	5	13	21	0	18	5	13	21	0	⋯	18
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
255	6	15	9	17	24	6	15	9	17	24	⋯	6

Figure 5. The searching area for hiding digit 22_{25} in $(5, 4)$.

3.4. The Extracting Section

The main procedure for data extraction is as shown in the lower part of Figure 1. Before extracting the secret message, the same stego-image and SSRM in the embedding section is required. The SSRM should contain digits $0 \sim N^2$, where N is a positive integer greater than 2. Before extracting the secret message, the same grouping technique that is used in the embedding section is first used to identify the same location of pixel-pairs on the stego-image. Every pair of stego pixels $\left(p'_i, p'_{i+1}\right)$ is seen as a coordinate on the SSRM. The location on the SSRM contains an element $M\left(p'_i, p'_{i+1}\right)$ which points out the N^2-base secret digit d' that is implied by the modified pixel-pair $\left(p'_i, p'_{i+1}\right)$. Then extract all d' by using the same mapping technique. After every d' has been extracted, the original secret message can be recovered by converting every d' into binary form. Here is an example for the extraction section. A pixel-pair $(6, 5)$ is cached to locate onto the SSRM in Figure 5 (i.e., $N = 5$). The element $M(6, 5)$ on the SSRM is a 25-base digit 22_{25}. Convert the 22_{25} back into its binary form, and the binary stream can be obtained as 10_2, which is part of the original secret message.

4. Experimental Results

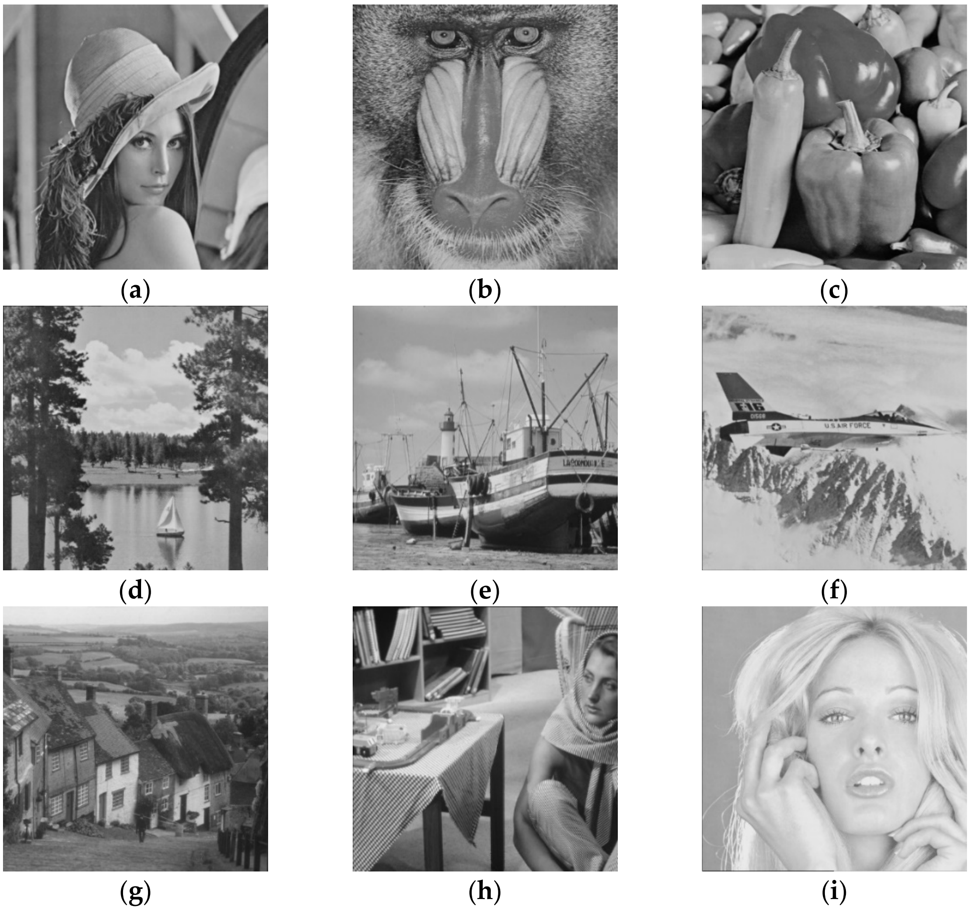
In this section, the experimental results and comparisons between the related work and proposed method are presented. The proposed data-hiding method was implemented using the Java IDE Eclipse 4.18 on a machine with CPU i5-8250U and 8.00 GB RAM. The source code is available at GitHub [32]. Comparisons are made with OPAP [1], LSB-re [2], EMD [3], Sudoku [4], MCB [6], PV [7], and DLMM [11]. The test images in this experiment are nine grayscale natural images, 'Lena', 'Baboon', 'Pepper', 'Scene', 'Boats', 'F-16', 'Goldhill', 'Barbara', and 'Tiffany', with resolutions of 512 × 512, as shown in Figure 6. These test images are available from the USC–SIPI image database [33]. The secret messages used in the experiments are randomly generated by a random number generator.

To evaluate the performance of the data-hiding technique, visual qualities and embedding capacity, two important factors, must be taken into consideration. To compare the visual quality, the peak signal-to-noise ratio (PSNR) is used as an international measure standard to present the visual quality of the stego-images. PSNR can be calculated by (2).

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

$$MSE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left(I_{ij} - I'_{ij}\right)^2 \tag{3}$$

where *MSE* represents the mean square error, which indicates the pixels' difference between the cover image I and the stego-image I'. Its formula is defined in (3), where H and W respectively denote the height and width of the image. Higher values of *PSNR* mean lower distortion in the stego-image.



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

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

(**g**) (**h**) (**i**)

Figure 6. Nine grayscale test images. (**a**) Lena, (**b**) Baboon, (**c**) Pepper, (**d**) Scene, (**e**) Boats, (**f**) F-16, (**g**) Goldhill, (**h**) Barbara, (**i**) Tiffany.

For embedding capacity, bits-per-pixel (bpp) is a basic unit being used in the measurement. The formula for calculating the embedding capacity is defined in (4).

$$C = \frac{S}{H \times W}\ (\text{bpp}) \tag{4}$$

where C represents the embedded capacity with the unit bits-per-pixel (bpp). S is the total number of bits of random secret message that are to be embedded into the cover image. H and W denote the height and width of the cover image, respectively. If the value of C is high, it means the data-hiding system exhibits a high embedding capacity.

For algorithmic efficiency, since all methods take the same input size $H \times W$ pixels as a cover image, the main difference between these methods lies in the search of the secret digit d_i on the reference matrix for the data embedding. However, this operation only requires constant time c, where $c \ll H \times W$. Furthermore, the extraction of the secret data is merely an act of direct access to the LSBs or lookup table. Therefore, the time complexity for all methods in this section is estimated as $O(H \times W + c) = O(H \times W)$.

The proposed method uses a scalable secret reference matrix (SSRM) to embed and extract data. The SSRM is composed of several $N \times N$ sub-blocks. The value of N affects the visual quality, embedding capacity, and security. First of all, Figure 7 is a trend diagram of PSNR from $N = 2$ to $N = 23$. It is discovered that the reduction of PSNR is very slow. When

$N = 23$, PSNR is 30 dB. On the other hand, Figure 8 shows that the embedding capacity from $N = 2$ to $N = 23$ can be adjusted very flexible. It can achieve a high embedding capacity of 4.5 bpp, which is much higher than that of most previous data-hiding techniques. Table 1 demonstrates that a decent PSNR has been achieved while the embedding capacity increases proportional to N. These results are reflected in comparison with the Sudoku method. In Table 2, let the proposed method set in the case of $N = 3$. For the same embedding capacity, the proposed method has a visual quality of, on average, 5.07 dB higher than that of the Sudoku method. Moreover, when the proposed method is under the case of $N = 5$, not only the embedding capacity is 0.74 (bpp) higher than the Sudoku method, but also the PSNR is on average 0.3 dB higher than their PSNR. To summarize these two factors, the visual quality and the embedding capacity of the proposed method, the visual quality of SSRM is not inferior to most of the recent methods, as shown in Table 3. On the other hand, the proposed SSRM method shared the same flexibility as the PV method to adjust the embedding capacity from 1 to 8 bpp, as shown in Table 4. Although the MCB method can also do similar adjustments, it only considers the cases of integer bpp. As for the security aspect, the DLMM method used a 16 × 16 Sudoku matrix to increase the number of possible solutions to the reference matrix. However, it sacrificed the visual quality of the stego-image, which can be obviously seen in Table 3 when the embedding capacity C is set to 2. Under the comprehensive evaluation, the proposed SSRM method has relatively more advantages than the other related works.

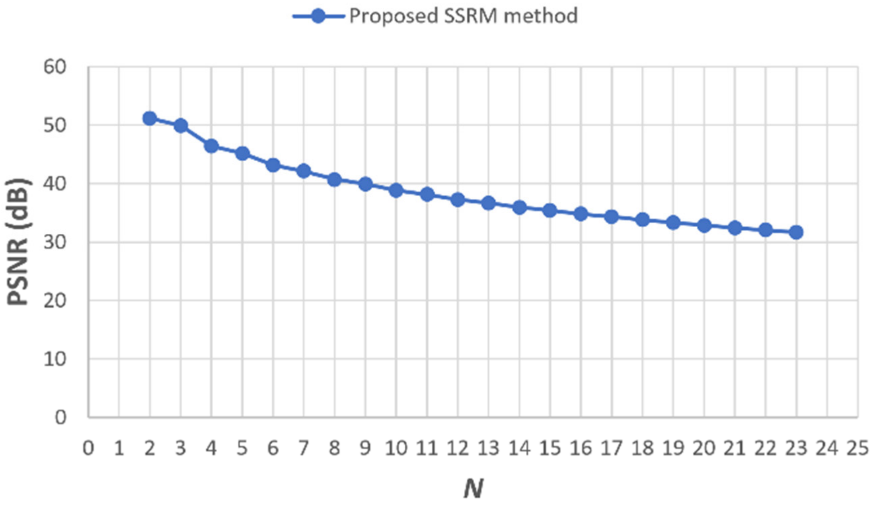


Figure 7. The PSNR trend diagram.

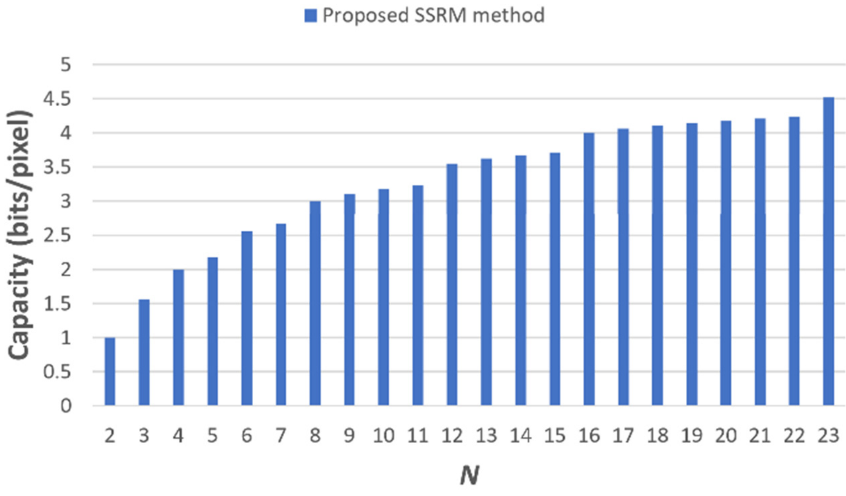


Figure 8. The embedding capacity trend diagram.

Table 1. The visual quality and embedding capacity in different values of N.

Cover Image	N	PSNR (dB)	C (bpp)	C/PSNR (%)
Random	2	51.15	1	1.96
	3	49.91	1.58	3.17
	4	46.39	2	4.31
	5	45.14	2.32	5.14
	6	43.14	2.58	5.98
	7	42.13	2.81	6.67
	8	40.75	3	7.36

Table 2. The visual quality and embedding capacity of the proposed SSRM method in N = 3 to N = 5.

Cover Image	Sudoku [4]		Proposed SSRM Method					
			N = 3		N = 4		N = 5	
	PSNR	C (bpp)	PSNR	C (bpp)	PSNR	C (bpp)	PSNR	C (bpp)
Lena	44.97	1.58	49.90	1.58	46.39	2.00	45.14	2.32
Baboon	44.68	1.58	49.92	1.58	46.39	2.00	45.15	2.32
Pepper	44.67	1.58	49.91	1.58	46.39	2.00	45.14	2.32
Scene	44.67	1.58	49.91	1.58	46.39	2.00	45.13	2.32
Boats	44.94	1.58	49.90	1.58	46.38	2.00	45.14	2.32
F-16	45.02	1.58	49.90	1.58	46.40	2.00	45.14	2.32
Goldhill	44.84	1.58	49.91	1.58	46.40	2.00	45.13	2.32
Barbara	44.77	1.58	49.90	1.58	46.37	2.00	45.14	2.32
Tiffany	45.02	1.58	49.91	1.58	46.39	2.00	45.14	2.32
Average	44.88	1.58	49.91	1.58	46.39	2.00	45.14	2.32

Table 3. Comparisons of related works in visual quality with embedding capacity C = 2 and C = 3.

Cover Image	C = 2 Bpp				C = 3 Bpp			
	MCB [6]	PV [7]	DLMM [11]	SSRM	MCB [6]	PV [7]	DLMM [11]	SSRM
	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR	PSNR
Lena	46.37	46.37	42.58	46.38	40.73	40.73	40.73	40.73
Baboon	46.37	46.38	42.62	46.38	40.73	40.73	40.73	40.73
Pepper	46.37	46.38	42.58	46.38	40.73	40.73	40.73	40.73
Scene	46.37	46.38	42.55	46.37	40.73	40.73	40.73	40.73
Boats	46.37	46.39	42.52	46.38	40.73	40.73	40.72	40.72
F-16	46.37	46.39	42.56	46.39	40.73	40.73	40.73	40.73
Goldhill	46.37	46.38	42.62	46.38	40.73	40.72	40.72	40.72
Barbara	46.37	46.37	42.60	46.37	40.73	40.73	40.73	40.73
Tiffany	46.37	46.38	42.63	46.39	40.72	40.73	40.72	40.73
Average	46.37	46.38	42.59	46.38	40.73	40.73	40.73	40.73

Table 4. Comparisons of related works in the flexibility of embedding capacity and the size of solutions.

	LSB-re [2]	EMD [3]	Sudoku [4]	MCB [6]	PV [7]	DLMM [11]	SSRM
bpp	1	1~1.36	1.58	1, 2, . . . , 8	1~8	2, 3	1~8
Possible Solutions	1	1	6.671×10^{21}	$\left(2^{3N}\right)!$	$N!$	5.958×10^{98} *	$N^2!$

In addition, an experiment was conducted on all nine test images to compare the performance between the proposed SSRM and the LSB-based approaches. From Table 5, the experimental results show that at the same condition for embedding capacity ($C = 1$ bpp $\sim$ 4 bpp), the PSNR of the proposed method is higher than the traditional LSB technique with an average of 2 dB higher PSNR. On the other hand, when the PSNR is compared with the OPAP [1], it is nearly the same in any condition for the embedding capacity. The slightly difference in the resulting stego-image is not easily detected by the naked eye. As a real-life example, for the two Lena images in Figure 9 it is difficult to distinguish which has more distortion than the other. Normally, the human eye cannot perceive the loss of stego-image quality when PSNR is greater than 40 dB. In the spatial domain, OPAP is known as the state-of-the-art baseline algorithm for large-payload data hiding. However, the main goal of the proposed method is to pursue an additional advantage in terms of security with the same performance of visual quality and embedding capacity. More specifically, the secret data that uses the OPAP can be easily attacked by trying to extract the least significant bits of every pixel. That is, at most eight attempts are required to obtain the secret message (i.e., every pixel value is represented by 8 bits). However, for the proposed method, the SSRM acts as a key for embedding and extracting the secret data. Therefore, it takes, at most, $N^2!$ attempts to find the correct SSRM for the corresponding stego-image.

Table 5. The comparison of visual quality and embedding capacity.

Cover Image	C	SSRM	OPAP	LSB
Lena	1	51.16	51.14	51.14
	2	46.39	46.37	44.15
	3	40.74	40.73	37.92
	4	34.84	34.81	31.78
Baboon	1	51.15	51.14	51.14
	2	46.39	46.37	44.15
	3	40.73	40.73	37.92
	4	34.81	34.80	31.86
Pepper	1	51.14	51.14	51.14
	2	46.39	46.37	44.16
	3	40.76	40.72	37.93
	4	34.81	34.80	31.84
Scene	1	51.16	51.14	51.14
	2	46.39	46.37	44.15
	3	40.75	40.73	37.89
	4	34.82	34.81	31.85
Boats	1	51.16	51.13	51.13
	2	46.38	46.37	44.13
	3	40.75	40.74	37.92
	4	34.81	34.81	31.77
F-16	1	51.15	51.14	51.14
	2	46.40	46.37	44.11
	3	40.75	40.73	37.96
	4	34.82	34.81	31.85
Goldhill	1	51.14	51.14	51.14
	2	46.40	46.37	44.16
	3	40.73	40.72	37.91
	4	34.82	34.80	31.86
Barbara	1	51.15	51.14	51.14
	2	46.37	46.38	44.15
	3	40.74	40.72	37.93
	4	34.82	34.80	31.82

Table 5. *Cont.*

Cover Image	C	SSRM	OPAP	LSB
Tiffany	1	51.17	51.15	51.15
	2	46.39	46.38	44.16
	3	40.76	40.73	37.90
	4	34.82	34.82	31.94

Figure 9. Two grayscale Lena images, of which (**a**) is the original image and (**b**) is the stego-image (PSNR = 34.84 dB, C = 4 bpp).

The final goal of the steganography is to securely cover the secret message to prevent hackers from extracting the secret message. The proposed method offers a further advantage for security. There are two reasons behind these advantages. The first is that the same SSRM is required to embed and extract the secret message. Second, due to the multiple solutions of the SSRM, it is difficult to decide on the correct one. Both the method of Chang et al. and the proposed method offer high security for a data-hiding scheme, but the proposed method is safer due to the SSRM's possible solutions, which number about $N^2!$, that is, $\prod_{i=0}^{N^2-1}\left(N^2-i\right)$ possibilities, as shown in Table 6. It is higher than the Sudoku solutions of Chang et al. of 6.671×10^{21} when $N \geq 5$.

Table 6. The size of SSRM solutions in different *N*.

N	Contain Digits *d*	Possible Solutions
2	$0 \leq d \leq 3$	$4! \doteqdot 2.4 \times 10$
3	$0 \leq d \leq 8$	$9! \doteqdot 3.63 \times 10^{5}$
4	$0 \leq d \leq 15$	$16! \doteqdot 2.09 \times 10^{13}$
5	$0 \leq d \leq 24$	$25! \doteqdot 1.55 \times 10^{25}$
6	$0 \leq d \leq 35$	$36! \doteqdot 3.72 \times 10^{41}$
7	$0 \leq d \leq 48$	$49! \doteqdot 6.08 \times 10^{62}$
8	$0 \leq d \leq 63$	$64! \doteqdot 1.27 \times 10^{89}$

5. Conclusions

This study proposes a data-hiding scheme that utilizes a scalable secret reference matrix (SSRM). The characteristics of the EMD method and Sudoku method are combined to achieve a high embedding capacity, low distortion, and high security. Empirical evidence suggests that in the acceptable range of the stego-image quality values ($PSNR > 34$), the embedding capacity of the proposed method is, on average, 4 bpp higher than that associated with the EMD method (on average 1.16 bpp). The PSNR of the proposed method is, on average, higher than that of the EMD method, and is slightly higher than the values yielded by other methods. On the other hand, compared with the Sudoku method, both visual quality and embedding capacity are higher than those produced by the Sudoku method for the cases of $N = 3$ to $N = 5$. The embedding capacity can be dynamically adjusted according to user requirements (i.e., by changing the value of N). In addition, compared with the best large-payload data-hiding method, OPAP, in the spatial domain, the proposed method offers the advantage of high security while embedding and extracting data owing to the various solutions of the SSRM. The SSRM solutions can be multiplied by changing the value of N, where the solution can be computed in $\left(N^2!\right)$. The proposed method is not only a simple and sufficient algorithm, but also represents a novel and secure idea for data-hiding in the spatial domain. Future work will further explore data-hiding for the SSRM edge cases. The possibility of combining other data-hiding schemes with edge detection will be considered for reducing the distortion of stego-images.

Author Contributions: Conceptualization, J.L. and C.-W.Y.; methodology, J.L., C.-W.T., and C.-W.Y.; software, J.L. and K.-H.L.; investigation, J.L. and K.-H.L.; formal analysis, J.L.; writing—original draft, J.L. and C.-W.T.; writing—review and editing, J.L. and C.-W.Y.; project administration, C.-W.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially supported by the Ministry of Science and Technology, Taiwan, R.O.C. (Grant Nos. MOST 110-2221-E-039-004, MOST 110-2221-E-143-003, MOST 110-2221-E-259-001, MOST 110-2221-E-143-004, MOST 110-2222-E-005-006, MOST 110-2634-F-005-006, MOST 110-2218-E-005-008-MBK, and MOST 111-2218-E-005-007-MBK) and by China Medical University, Taiwan (Grant No. CMU110-S-21).

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. Chan, C.K.; Cheng, M. Hiding data in images by simple LSB substitution. *Pattern Recognit.* **2004**, *37*, 469–474. [CrossRef]
2. Mielikainen, J. LSB matching revisited. *IEEE Signal Process. Lett.* **2006**, *13*, 285–287. [CrossRef]
3. Zhang, X.; Wang, S. Efficient steganographic embedding by exploiting modification direction. *IEEE Commun. Lett.* **2006**, *10*, 781–783. [CrossRef]
4. Chang, C.C.; Chou, Y.C.; Kieu, T.D. An information hiding scheme using Sudoku. In Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control, Dalian, China, 18–20 June 2008; p. 17.
5. Lin, C.; Chang, C.; Lee, W.; Lin, J. A novel secure data hiding scheme using a secret reference matrix. In Proceedings of the 2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kyoto, Japan, 12–14 September 2009; pp. 369–373.
6. Wu, Q.; Zhu, C.; Li, J.-J.; Chang, C.-C.; Wang, Z.-H. A magic cube based information hiding scheme of large payload. *J. Inf. Secur. Appl.* **2016**, *26*, 1–7. [CrossRef]
7. Chang, C.-C.; Chang, J.-C.; Chou, Y.-H.; Wu, H.-L. A high embedding capacity data hiding scheme based upon permutation vectors. In *International Workshop on Digital Watermarking: Digital Forensics and Watermarking*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 578–587.
8. Malik, A.; Kumar, R.; Singh, S. A new image steganography technique based on pixel intensity and similarity in secret message. In Proceedings of the 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 12–13 October 2018; pp. 828–831.
9. Chen, Y.; Sun, W.; Li, L.; Chang, C.C.; Wang, X. An efficient general data hiding scheme based on image interpolation. *J. Inf. Secur. Appl.* **2020**, *54*, 102584. [CrossRef]
10. Singh, S. Adaptive PVD and LSB based high capacity data hiding scheme. *Multimed. Tools Appl.* **2020**, *79*, 18815–18837. [CrossRef]
11. Hsiao, T.-C.; Liu, D.-X.; Chen, T.-L.; Chen, C.-C. Research on image steganography based on Sudoku matrix. *Symmetry* **2021**, *13*, 387. [CrossRef]
12. Tian, J. Reversible data embedding using a difference expansion. *IEEE Trans. Circuits Syst. Video Technol.* **2003**, *13*, 890–896. [CrossRef]
13. Ni, Z.; Shi, Y.-Q.; Ansari, N.; Su, W. Reversible data hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2006**, *16*, 354–362.
14. Malik, A.; Singh, S.; Kumar, R. Recovery based high capacity reversible data hiding scheme using even-odd embedding. *Multimed. Tools Appl.* **2018**, *77*, 15803–15827. [CrossRef]

15. Chen, Y.-C.; Hung, T.-H.; Hsieh, S.-H.; Shiu, C.-W. A new reversible data hiding in encrypted image based on multi-secret sharing and lightweight cryptographic algorithms. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 3332–3343. [CrossRef]
16. Chen, Y.-C.; Shiu, C.-W. Distributed encrypted image-based reversible data hiding. *J. Internet Technol.* **2021**, *22*, 101–107.
17. Xu, S.; Chang, C.-C.; Liu, Y. A high-capacity reversible data hiding scheme for encrypted images employing vector quantization prediction. *Multimed. Tools Appl.* **2021**, *80*, 20307–20325. [CrossRef]
18. Wang, X.; Chang, C.-C.; Lin, C.-C. High capacity reversible data hiding in encrypted images based on prediction error and block classification. *Multimed. Tools Appl.* **2021**, *80*, 29915–29937. [CrossRef]
19. Gao, K.; Horng, J.-H.; Chang, C.-C. High-capacity reversible data hiding in encrypted images based on adaptive block encoding. *J. Vis. Commun. Image Represent.* **2022**, *84*, 103481. [CrossRef]
20. Nguyen, T.-S.; Chang, C.-C.; Lin, C.-C. High capacity reversible data hiding scheme based on AMBTC for encrypted images. *J. Internet Technol.* **2022**, *23*, 55–66.
21. Barni, M.; Bartolini, F.; Cappellini, V.; Piva, A. A DCT-domain system for robust image watermarking. *Signal Process.* **1998**, *66*, 357–372. [CrossRef]
22. Lin, S.D.; Chen, C.F. A robust DCT-based watermarking for copyright protection. *IEEE Trans. Consum. Electron.* **2000**, *46*, 415–421. [CrossRef]
23. Chu, W.C. DCT-based image watermarking using subsampling. *IEEE Trans. Multimed.* **2003**, *5*, 34–38. [CrossRef]
24. Wong, K.; Qi, X.; Tanaka, K. A DCT-based Mod4 steganographic method. *Signal Process.* **2007**, *87*, 1251–1263. [CrossRef]
25. Zhang, D.; Pan, Z.; Li, H. A contour-based semi-fragile image watermarking algorithm in DWT domain. In Proceedings of the 2nd International Workshop Education Technology and Computer Science, Wuhan, China, 6–7 March 2010; Volume 3, pp. 228–231.
26. Wu, X.; Sun, W. Robust copyright protection scheme for digital images using overlapping DCT and SVD. *Appl. Soft Comput.* **2013**, *13*, 1170–1182. [CrossRef]
27. Hong, W.; Chen, J.; Chang, P.S.; Wu, J.; Chen, T.S.; Lin, J. A color image authentication scheme with grayscale invariance. *IEEE Access* **2021**, *9*, 6522–6535. [CrossRef]
28. Ker, A.D. A general framework for structural steganalysis of LSB replacement. In Proceedings of the 7th International Conference on Information Hiding, Barcelona, Spain, 6–8 June 2005; Volume 3727, pp. 296–311.
29. Ker, A.D. A fusion of maximum likelihood and structural steganalysis. In Proceedings of the 9th International Conference on Information Hiding, Saint Malo, France, 11–13 June 2007; pp. 204–219.
30. Ker, A.D.; Böhme, R. Revisiting weighted stego-image steganalysis. In Proceedings of the Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, San Jose, CA, USA, 2 April 2008.
31. Sudoku. Available online: https://en.wikipedia.org/wiki/Sudoku (accessed on 14 April 2022).
32. GitHub Repository. Available online: https://github.com/senyalin/SSRM (accessed on 14 April 2022).
33. SIPI Image Database. Available online: http://sipi.usc.edu/database/ (accessed on 14 April 2022).