High Capacity Information Hiding Method Based on Pixel-value Adjustment with Modulus Operation

Teng Li, Yu Zhang*, Sha Wang, and Jun-jie Sun
School of Computer and Information Science, Southwest University
Beibei District of Chongqing, 400715 China
[e-mail: liteng0264@email.swu.edu.cn, zhangyu@swu.edu.cn, wangsha1028@email.swu.edu.cn, sjj6191055@email.swu.edu.cn]
*Corresponding author: Yu Zhang

Received December 20, 2020; revised February 23, 2021; March 21, 2021; published April 30, 2021

Abstract

Through information hiding technique, secret message can be hidden in pictures. Stego-image quality and hiding capacity are two important metrics for information hiding. To enhance these metrics, many schemes were proposed by scholars in recent years. Some of them are effective and successful, but there is still a room for further improvement. A high capacity information hiding scheme (PAMO, Pixel-value Adjustment with Modulus Operation Algorithm) is introduced in this paper. PAMO scheme uses pixel value adjustment with modulus operation to hide confidential data in cover-image. PAMO scheme and some referenced schemes are implemented in Python and experiments are carried out to evaluate their performance. In the experiments, PAMO scheme shows better performance than other methods do. When secret message length is less than 72000 bits, the highest hiding capacity of PAMO can reach 7 bits per pixel, at the same time the $PSNR$ of stego-images is greater than 30 dB.

Keywords: Cover-image, Hiding Capacity, Information Hiding, Modulus Operation, Stego-image

This work is supported in part by National Key Research and Development Program of China under Grant 2018YFB2004100, Centre for Research and Innovation in Software Engineering, the School of Computer and Information Science(Southwest University, Chongqing 400715, China), Chongqing Graduate Education Teaching Reform Research Project (yjg203032), Chongqing Industrial Control System Information Security Technology Support Center, and Chongqing Intelligent Instrument and Control equipment Engineering Technology Research Center.

http://doi.org/10.3837/tiis.2021.04.018 **ISSN : 1976-7277**

1. Introduction

In information security field, information hiding technique is an important and useful tool. It helps to make confidential message safe [1, 2]. Usually, researchers concern mainly on hiding capacity and stego-image quality. It is difficult to improve both aspects at the same time. Improving hiding capacity while keeping the stego-image quality acceptable ($PSNR > 30\ dB$) is a common strategy.

In the past decades, many hiding methods have been proposed, such as the least significant bit (LSB) replacement method [3-6], pixel value differencing (PVD) steganography [7-9], exploiting modification direction (EMD) method [10-12], extraction function with modulus method [13-10] and so on. In LSB method, the least significant bit of the cover pixel is used to embed secret message; PVD steganography divides the gray scale image into many non-overlapping blocks, each block consists of two pixels, the secret message is embedded in the difference between the values of two pixels; EMD method can embed secret message by modifying the value of one pixel in a group of gray scale pixels, it has the characteristics of small change in cover pixel value and high information hiding ability; extraction function with modulus method uses extraction function to convert the secret message, then embeds it into cover pixel by modulus operation. In 2007, Yi-Ren Wang et al. proposed an improved EMD method [7], it reached maximum bpp at 1.5 and kept high stego-image quality. In 2009, another improved EMD method was proposed by Ki-yun Jung et al. [17], it uses $5 - ary$ notational system to update the pixel value, this method achieved maximum rate of 2 bpp and $PSNR$ of 47.95 dB.

By learning these excellent methods, we found that many hiding methods still have room for improvement. In order to embed more information in cover-image while ensuring the quality of stego-image, in this paper, a high capacity information hiding scheme is proposed. It can greatly improve the hiding capacity while maintain acceptable quality of stego-images. The contributions and innovations of PAMO scheme are as follows:

1) PAMO scheme has the ability of adjusting hiding capacity. Each pixel can carry up to 7 confidential bits when secret message with a proper length;

2) The correctness of PAMO scheme is proved mathematically, and the effectiveness is proved by comprehensive experiments;

3) All codes, images and other related materials are public available at https://github.com/liteng0264/Information-Hiding.git. This lets other researchers verify our work easier.

The rest of this paper is organized as follows. Section 2 introduces Ki-Hyun Jung's method [17] and T.D.Sairam's method [19]. In Section 3, PAMO is introduced and the math proof is given. Comprehensive experiments are carried out in Section 4. Finally, Section 5 concludes the paper.

2. Related Works

In this section, let g be the pixel value in cover-image, g' be the pixel value in stego-image, n be the number of binary bits in each secret group, d be the secret digit converted from the secret group, d' be the secret digit recovered from the stego-pixel, b' be the binary bits converted from d'.

2.1 JK09 Algorithm

In 2009, Ki-Hyun Jung et al. proposed an improved EMD method [17]. It converts n binary numbers to a secret digit firstly, then uses one cover pixel to carry the secret digit. More details of JK09 Algorithm are shown as follows:

Step 1 Compute the extraction function f given in (1).

$$f = (g_i + x) \ mod \ (2n + 1) \tag{1}$$

$$\begin{aligned} &If \ 0 \leq g \leq 1, \ then \ 0 \leq x < 2n + 1; \\ &If \ 254 \leq g \leq 255, \ then \ -(2n + 1) < x < 0; \\ &If \ 1 < g < 254, \ then \ |x| < 2n + 1. \end{aligned}$$

Step 2 When f is equal to d, g' can be calculated by (2).

$$g' = g + x \tag{2}$$

Step 3 Use the extraction function f to recover the secret digit d.

Through JK09 Algorithm, secret message can be embedded in cover-image efficiently. Besides, stego-image quality is kept acceptable. It can achieve maximum rate of 2 bpp and $PSNR$ of 47.95 dB.

2.2 SB19 Algorithm

In 2019, T.D.Sairam et al. proposed a high capacity information hiding scheme [19]. The first step of this scheme is taking n bits as a secret group, then converting the secret group to a $n^2 - ary$ notational system secret digit d, finally the secret digit can be hidden in the cover pixel. Details of SB19 Algorithm are shown as follows:

Step1 Before using SB19 Algorithm, secret message should be divided into multiple groups, each group with n binary numbers. Then convert each group to a $n^2 - ary$ notational system number.

Step2 Embed all secret digits by loop 1 giving in the follows:

```
$for(-\lfloor n^2/2 \rfloor \leq x \leq \lfloor n^2/2 \rfloor)\{$
    $f = (g + x) \ mod \ n^2$
    $if(f == d)\{$
        $g' = g + x$
        $break$
    $\}$
$\}$
```

* $\lfloor \cdot \rfloor$ represents the flooring operation

Step 3 Recover secret message from stego-image by loop 2 giving in the follows:

```
$for\ all\ g'\ in\ stego - image\ \{$
        $d' = g'\ mod\ n^2$
        $b' = d'\ convert\ to\ binary\ digits$
$\}$
$for\ all\ b'\ \{$
        $secret\ message = combine\ b'$
$\}$
```

3. Proposed Method

In this section, let g be the pixel value in cover-image, g' be the pixel value in stego-image. Secret messages would be divided into multiple groups, each group with n binary numbers. d be the secret digit converted from the secret group and d be a $(2^n + n) - ary$ notational system number, d' be the secret digit recovered from the stego-pixel, b' be the binary bits converted from d'.

3.1 PAMO Algorithm

This paper proposes a new information hiding scheme (PAMO Algorithm). It can not only improve the hiding capacity but also keep the stego-image quality acceptable. The maximum embedding rate is 7 while the $PSNR$ of stego-image still more than 30 dB.

Step 1 Before using PAMO Algorithm, secret message should be divided into multiple secret groups by (3).

$$num\_group = \lceil secret\_message\_length/n \rceil \tag{3}$$

* $\lceil \cdot \rceil$ represents the ceiling operation

Step 2 After step1, every secret group has n binary numbers. These binary numbers should be converted to a $(2^n + n) - ary$ notational system number d.

Step 3 Embed secret digit d in cover pixel by loop 3 giving in the follows:

```
$for\ i\ in\ [1\ , num\_gropu]\ \{$
    $for(-\lfloor (2^n + n)/2 \rfloor \leq x \leq \lfloor (2^n + n)/2 \rfloor)\ \{$
        $f = (g + x)\ mod\ (2^n + n)$
        $if(f == d)\ \{$
            $temp = g + x$
            $if(temp < 0)$
                $g' = temp + (2^n + n)$
            $if(temp > 255)$
                $g' = temp - (2^n + n)$
            $if(0 < temp < 255)$
                $g' = temp$
            $break$
            }
        }
    }
```

* $\lfloor\cdot\rfloor$ represents the flooring operation

After step1, step2 and step3, secret message can be hidden into cover-image successfully. At the same time, a stego-image is generated.

Step 4 d' can be obtained by (4).

$$d' = g' \, mod \, (2^n + n) \tag{4}$$

Step 5 Recover secret message from stego-image by loop 4 giving in the follows:

```
$for\ all\ g'\ in\ stego - image$ {
    $d' = g'\ mod\ (2^n + n)$
    $b' = d'\ convert\ to\ binary\ digits$
}
$for\ all\ b'$ {
    $secret\ message = combine\ b'$
}
```

In order to illustrate the steps of PAMO more clearly, here is a simple example:

Assume that there is a secret message 1011100100010110; And there is a gray scale image, the pixel values of the first four pixels are 255, 150, 50, 5……; We take n as 4, it means that 4 bits will be embedded in each pixel.

Step 1 Divide the secret message into four groups, each group is 1011, 1001, 0001 and 0110.

Step 2 Convert each group to a $20 - ary$ notational system number, respectively are b, 9, 1, 6.

Step 3 Embed the first secret digit into the first pixel:

$x \in [-10,10]$, $f = (255 + x) \, mod \, 20$, so when x is -10, f is 5, f is not equal to the first secret digit; when x is -9, f is 6; when x is -8, f is 7; when x is -7, f is 8; when x is -6, f is 9; when x is -5, f is 10; when x is -4, f is 11, f is equal to the first secret digit b. According to embedding rules, the pixel value of the first stego-pixel is 251. So far, the first secret group has been successfully embedded in the cover pixel.

Step 4 Recover secret digit from stego-pixel, $d' = g' \, mod \, 20 = 11$, then convert it to binary bits, that is 1011. So far, the first secret group has been successfully recovered from the stego-pixel.

According to the above steps, other secret digit can be embedded in the cover pixels, the secret digit can also be recovered from the stego-pixels.

3.2 Math Proof

3.2.1 PAMO can embed secret digit into cover-image correctly

When n equals 7, the value of secret digit ranges from 0000000 to 1111111 (In Binary system), that is 0 to 127 (In Decimal system). The necessary condition for embedding is $'f == d'$, $f =$

$(g + x)\ mod\ 135$, then some conditions can be derived:

$$g \in [0,255], d \in [0,127]$$

This part needs to prove that, no matter what the value of g or d is, there must exist f equals to d.

$\because -\lfloor (2^n + n)/2 \rfloor \le x \le \lfloor (2^n + n)/2 \rfloor, n = 7$

$\therefore x \in [-68,67]$

$\because f = (g + x)\ mod\ 135$

$\therefore f \in [0,134]$, this is determined by the nature of modulo operation.

$\because$ From the rules of modulo operation: $(a + b)\ mod\ p = \big((a\ mod\ p) + (b\ mod\ p)\big)\ mod\ p$

$\therefore f = (g + x)\ mod\ 135$
$\quad = \big((g\ mod\ 135) + (x\ mod\ 135)\big)\ mod\ 135$

$\therefore$ Set $y = g\ mod\ 135$ and $z = x\ mod\ 135$, then $f = (y + z)\ mod\ 135$

$\because$ Each pixel value is ranging from 0 to 255, but for a certain pixel, its value is certain.

$\therefore g$ is a certain value.

$\therefore y$ is a certain value and a constant, ranging from 0 to 134.

$\because x \in [-68,67]$

$\therefore z \in [0,134]$, z is an uncertain value and a variable.

$\because f = (y + z)\ mod\ 135$, y is a constant and ranging from 0 to 134, z is a variable and belonging to 0 to 134.

$\therefore f \in [0,134]$, f is an uncertain value and a variable, f can take all values in its range.

The range of f is larger than the range of d, so secret digit definitely can be embedded in cover pixel.

3.2.2 PAMO can recover secret digit from stego-image correctly

From proof 3.2.1, there must exist f equals to d. Then could follow the below rules to update the cover pixel value:

```
$temp = g + x$
$if(temp < 0)$
    $g' = temp + (2^n + n)$
$if(temp > 255)$
    $g' = temp - (2^n + n)$
$if(0 < temp < 255)$
    $g' = temp$
```

This part needs to prove that, variable d' can be obtained by (4). Besides, d' is absolutely equal to d.

① $if\ temp < 0,\ then\ g' = (g + x) + 135$

$\because d' = g'\ mod\ (2^n + n)$

$$\begin{aligned}\therefore d' &= (g + x + 135)\ mod\ (2^n + n)\\ &= \big(((g + x)\ mod\ 135) + (135\ mod\ 135)\big)\ mod\ 135\\ &= \big((g + x)\ mod\ 135\big)\ mod\ 135\\ &= f\ mod\ 135\end{aligned}$$

$\because$ Known in proof 3.2.1, $f \in [0,134]$.

$\therefore d' = f\ mod\ 135 = f = d$

$\therefore$ If $temp$ is less than 0, secret digit can be recovered.

② $if\ temp > 255,\ then\ g' = (g + x) - 135$

$\because d' = g'\ mod\ (2^n + n)$

$$\begin{aligned}\therefore d' &= (g + x - 135)\ mod\ (2^n + n)\\ &= (g + x - 135)\ mod\ 135\\ &= \big(((g + x)\ mod\ 135) - (135\ mod\ 135)\big)\ mod\ 135\\ &= \big((g + x)\ mod\ 135\big)\ mod\ 135\\ &= f\ mod\ 135\end{aligned}$$

$\because$ Known in proof 3.2.1, $f \in [0,134]$.

$\therefore d' = f\ mod\ 135 = f = d$

$\therefore$ If $temp$ is larger than 255, secret digit can be recovered.

③ $if\ 0 < temp < 255,\ then\ g' = g + x$

$\because d' = g'\ mod\ (2^n + n)$

$$\begin{aligned}\therefore d' &= (g + x)\ mod\ (2^n + n)\\ &= (g + x)\ mod\ 135 = f = d\end{aligned}$$

$\therefore$ If $temp$ is larger than 0 and less than 255, secret digit can be recovered.

Therefore, (4) can help recover the secret digit d from stego-image. It has been proved that the secret digit d can be embedded in cover-image and recovered from stego-image when n is 7. When n less than 7, reader can prove it by the above steps if he is interested.

Comparing with JK09 Algorithm and SB19 Algorithm, PAMO is more flexible. Through PAMO Algorithm, users have more room to choose better stego-image quality or higher hiding capability, this is an obvious advantage. For example, if users make n be 2 or 3, thus the stego-image quality could be pretty good. If hiding capacity is the first consideration, n can be 5 or 6, thus the maximum embedding rate of PAMO could be higher than other methods.

4. Experimental Results and Performance Discussion

In this section, the experimental results are given to evaluate the performance of three methods. JK09 Algorithm, SB19 Algorithm and PAMO Algorithm are implemented by Python and run in a PC with an Intel(R) Core(TM) i5-4200M CPU @ 2.50 GHz and a 4 GB RAM, the operating system is Windows 7 Professional 64-bit and the experiment software is Pycharm. The experimental images are standard gray scale image of size 512 × 512, this paper get all images from USC-SIPI image data base [21].

Six experiments are designed in this section, secret message length in different experiments is different. The gray scale image has 262144 pixels totally (512 × 512), so these experiments use 49000, 72000, 262144, 524288, 786432, 1048576 as secret message length in turn.

The Peak Signal to Noise Ratio ($PSNR$) is a main metrics in information security field, many researchers use it to evaluate the performance of information hiding method. $PSNR$ higher than 40 dB indicates that the stego-image is very close to the cover-image; $PSNR$ between 30 dB and 40 dB indicates that the distortion of stego-image is perceptible but acceptable; $PSNR$ between 20 dB and 30 dB indicates poor stego-image quality; Finally, $PSNR$ less than 20 dB indicates that the stego-image quality is unacceptable.

In information security field, if $PSNR$ is higher than 30 dB, the secret message hidden in stego-image is considered imperceptible to the human visual system. $PSNR$ is defined in (5). The formula of Mean Square Error (MSE) calculation is defined in (6).

$$PSNR = 10log_{10}\left(\frac{255^2}{MSE}\right)dB \tag{5}$$

$$MSE = \frac{1}{M\times N}\sum_{i=1}^{N}\sum_{j=1}^{M}(g_{ij} - g'_{ij}) \tag{6}$$

* M and N represent the number of rows and columns in cover-image and stego-image
* g_{ij} and g'_{ij} represent the pixel value of cover-image and stgeo-image, their location is row i and column j

This paper uses bpp to represent the number of binary bits embedded in each pixel.

When the length of secret message is 49000 bits, the performance of three algorithms is shown in **Table 1**.

Table 1. Experimental results when the length of secret message is 49000 bits

Method bpp PSNR	JK09 Algorithm	SB19 Algorithm	PAMO Algorithm
1	41.44		56.93
2	37.72	46.38	53.14
3		39.85	50.11
4		34.82	46.11
5			41.99
6			37.27
7			31.56

As **Table 1** shows, JK09 Algorithm can keep the maximum *bpp* at 2, SB19 Algorithm can keep the maximum *bpp* at 4, while PAMO Algorithm can keep the maximum *bpp* at 7.

Therefore, PAMO Algorithm holds the best performance in embedding capacity aspect. Besides, when keeping the same *bpp*, PAMO Algorithm get the best stego-image quality, which is much greater than JK09 Algorithm and SB19 Algorithm. The stego-images of Peppers generated by different methods are shown in **Fig. 1**.

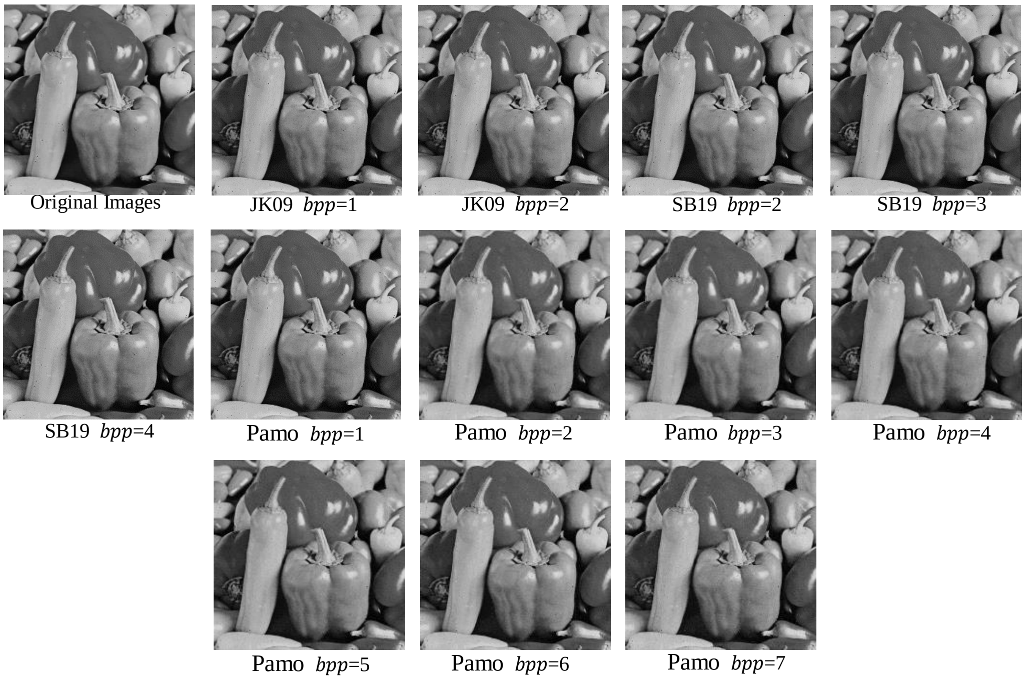
Original Images
JK09 *bpp*=1
JK09 *bpp*=2
SB19 *bpp*=2
SB19 *bpp*=3
SB19 *bpp*=4
Pamo *bpp*=1
Pamo *bpp*=2
Pamo *bpp*=3
Pamo *bpp*=4
Pamo *bpp*=5
Pamo *bpp*=6
Pamo *bpp*=7

Fig. 1. The stego-images when the length of secret message is 49000 bits

When the length of secret message is 72000 bits, the performance of three algorithms as **Fig. 2** shows.

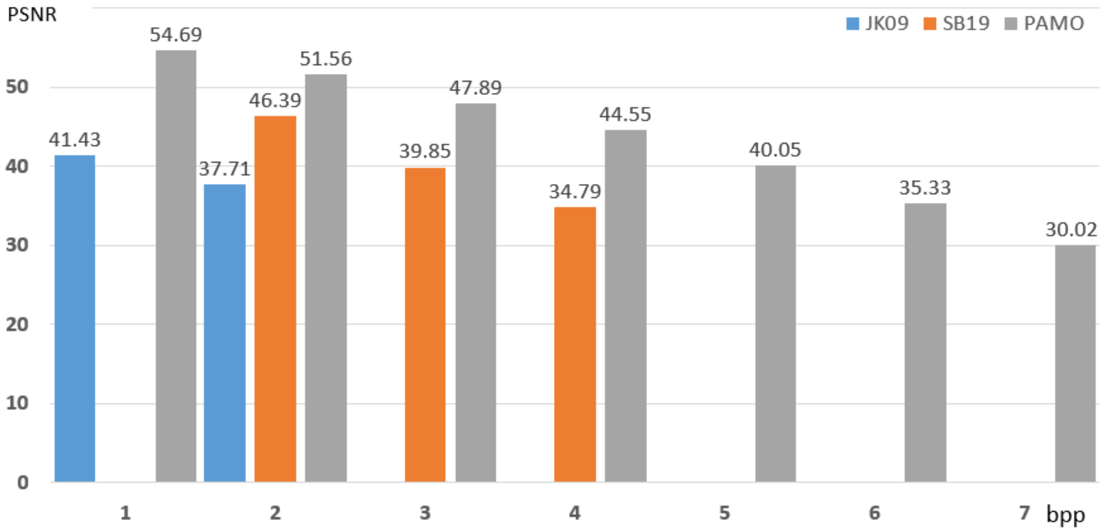

Fig. 2. Experimental results when the length of secret message is 72000 bits

From **Fig. 2**, JK09 Algorithm can keep the maximum bpp at 2, SB19 Algorithm can keep the maximum bpp at 4, while PAMO Algorithm can keep the maximum bpp at 7. Therefore, from the aspect of embedding capacity, PAMO Algorithm can achieve higher maximum bpp. From the aspect of stego-image quality, PAMO Algorithm can achieve higher $PSNR$.

When secret message length is 262144 bits, the performance of three algorithms is shown in **Table 2**. As **Table 2** shows, JK09 Algorithm can keep the maximum bpp at 2, SB19 Algorithm can keep the maximum bpp at 4, PAMO Algorithm can keep the maximum bpp at 5. Therefore, PAMO Algorithm has the best hiding capacity. Besides, when bpp is kept same, PAMO Algorithm can get higher stego-image quality than other methods.

Table 2. Experimental results when the length of secret message is 262144 bits

Method bpp PSNR	JK09 Algorithm	SB19 Algorithm	PAMO Algorithm
1	41.45		49.89
2	37.71	46.37	46.14
3		39.89	41.12
4		34.77	38.93
5			33.67
6			29.01
7			23.92

When the length of secret message is 524288 bits, the performance of three algorithms is shown in **Fig. 3**. In **Fig. 3**, JK09 Algorithm can keep the maximum bpp at 2, SB19 Algorithm can keep the maximum bpp at 4, PAMO Algorithm can keep the maximum bpp at 5. So PAMO Algorithm has better hiding capacity than others.

What is different from before is that, when bpp is 2 or 3, PAMO Algorithm has lower $PSNR$ than SB19 Algorithm, but it is still acceptable.

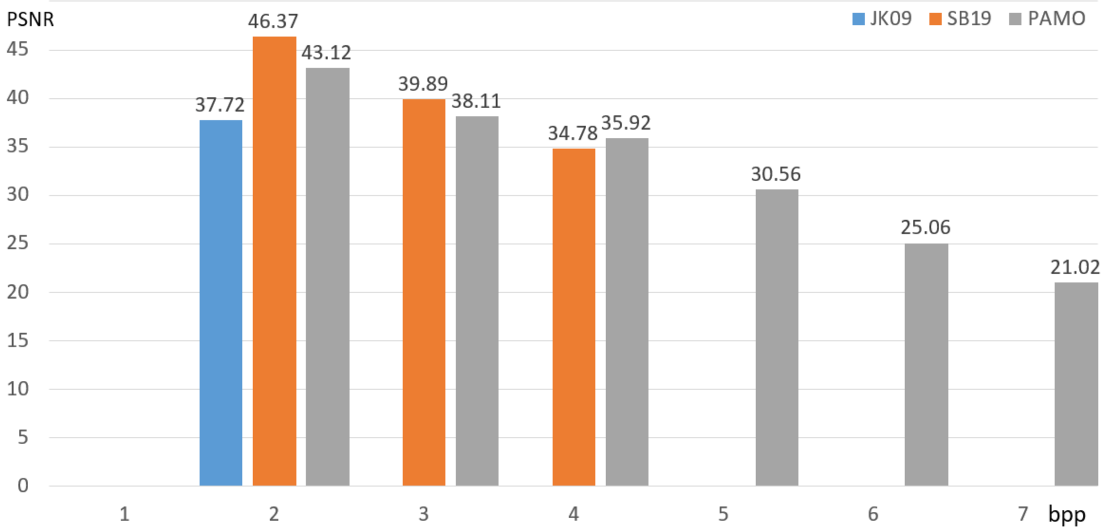

Fig. 3. Experimental results when the length of secret message is 524288 bits

When the length of secret message is 786432 bits, the performance of three algorithms is shown in **Table 3**. As **Table 3** shows, JK09 Algorithm can't embed it in a gray scale image of size 512 × 512. The maximum *bpp* JK09 can achieve is limited to 2, the cover-image has 262144 pixels totally, therefore, only 524288 bits can be embedded at most. Both SB19 Algorithm and PAMO Algorithm can keep the maximum *bpp* at 4, they can also make the stego-image quality acceptable.

Table 3. Experimental results when the length of secret message is 786432 bits

Method bpp PSNR	JK09 Algorithm	SB19 Algorithm	PAMO Algorithm
1			
2			
3		39.88	38.12
4		34.77	32.87
5			27.55
6			25.04
7			19.24

When the length of secret message is 1048576 bits, the performance of different algorithms is shown in **Fig. 4**.

As is shown in **Fig. 4**, both SB19 Algorithm and PAMO Algorithm can keep the maximum *bpp* at 4. Even in this special situation, PAMO Algorithm still can get same hiding capacity as SB19 Algorithm. Besides, both of them can maintain stego-image quality acceptable. In other words, PAMO Algorithm can achieve same effect as SB19 Algorithm.

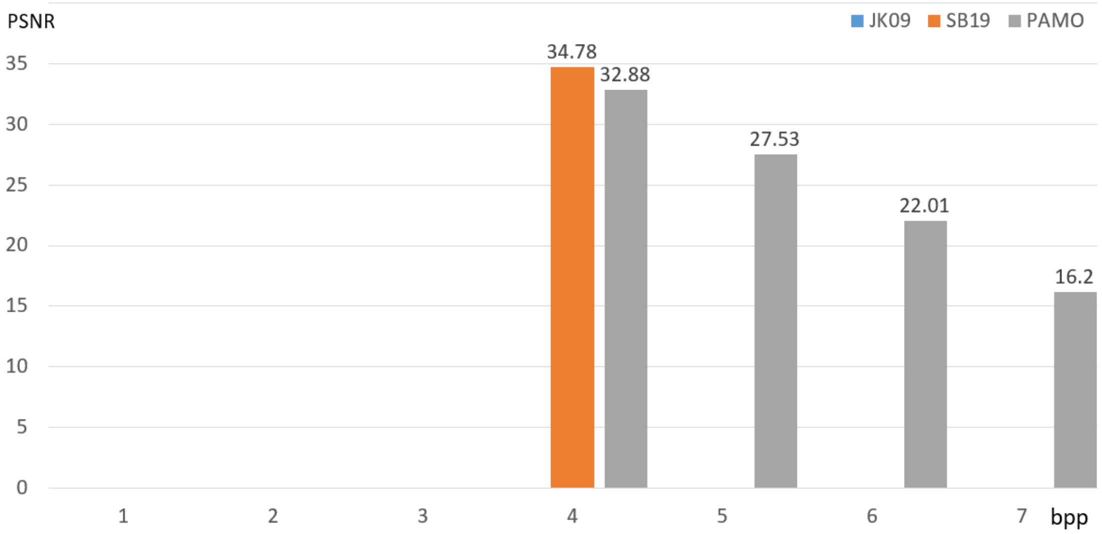

Fig. 4. Experimental results when the length of secret message is 1048576 bits

According to the above experiments, when the length of the secret message changes, the maximum *bpp* each algorithm can obtain will change accordingly. The detail of changes and the maximum *bpp* from each algorithm are shown in **Fig. 5**.

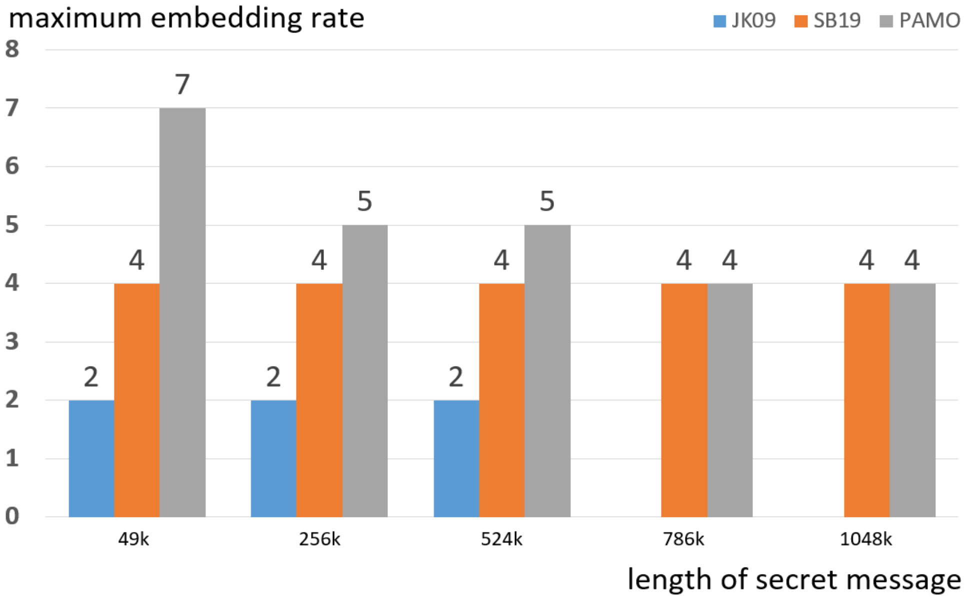

Fig. 5. Hiding capacity of three Algorithms

Next, we first prove the security of PAMO theoretically, then use the visual attack method to evaluate the security of PAMO.

If someone want to recover the secret message from setgo-image, variable n and the extraction function must be known. Because recovering secret digit from the stego-pixel requires the correct modulus. In addition, n is also a necessary condition when the secret digit is converted to a binary secret group. The length of each secret group is determined by n, for example, when the value of secret digit is 1, if n is 4, then secret group is 0001, if n is 7, then secret group is 0000001. Therefore, when n is unknown, the secret message cannot be obtained from the stego-image, in other words, the safety of PAMO can be guaranteed.

In order to prove the security of PAMO more convincingly, bit plane attack is applied to process stego-images and cover-image. As a visual attack method, bit plane attack constructs plane images by extracting the corresponding bit of each pixel, it can extract the meaningful information directly.

Four pictures were selected for bit plane attack experiments, they are the gray scale cover-image "Tiffany" of size 512 × 512, its stego-image when the length of secret message is 49000 bits and *bpp* is 2 ($PSNR$ is 53.14 dB), its stego-image when the length of secret message is 262144 bits and *bpp* is 5 ($PSNR$ is 33.67 dB), its stego-image when the length of secret message is 1048576 bits and *bpp* is 7 ($PSNR$ is 16.2 dB). Each picture can generate eight plane images by bit plane attack, the plane images from each picture are shown in **Fig. 6**, **Fig. 7**, **Fig. 8** and **Fig. 9**.

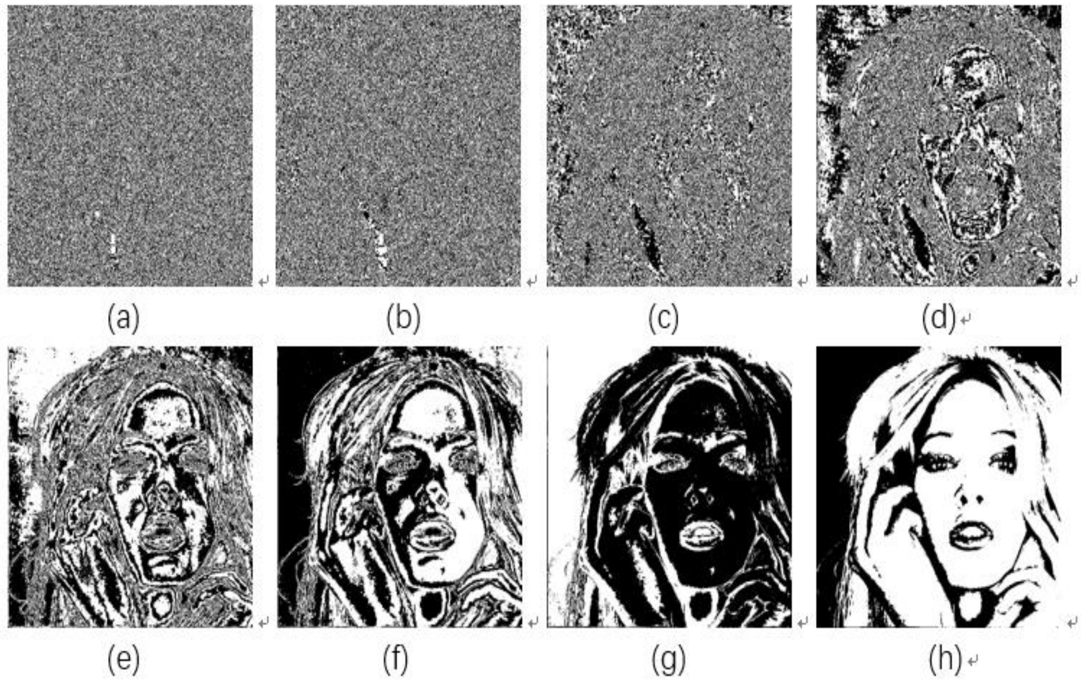

Fig. 6. Plane images of cover-image
(a)1th bit; (b)2th bit; (c)3th bit; (d)4th bit; (e)5th bit; (f)6th bit; (g)7th bit; (h)8th bit

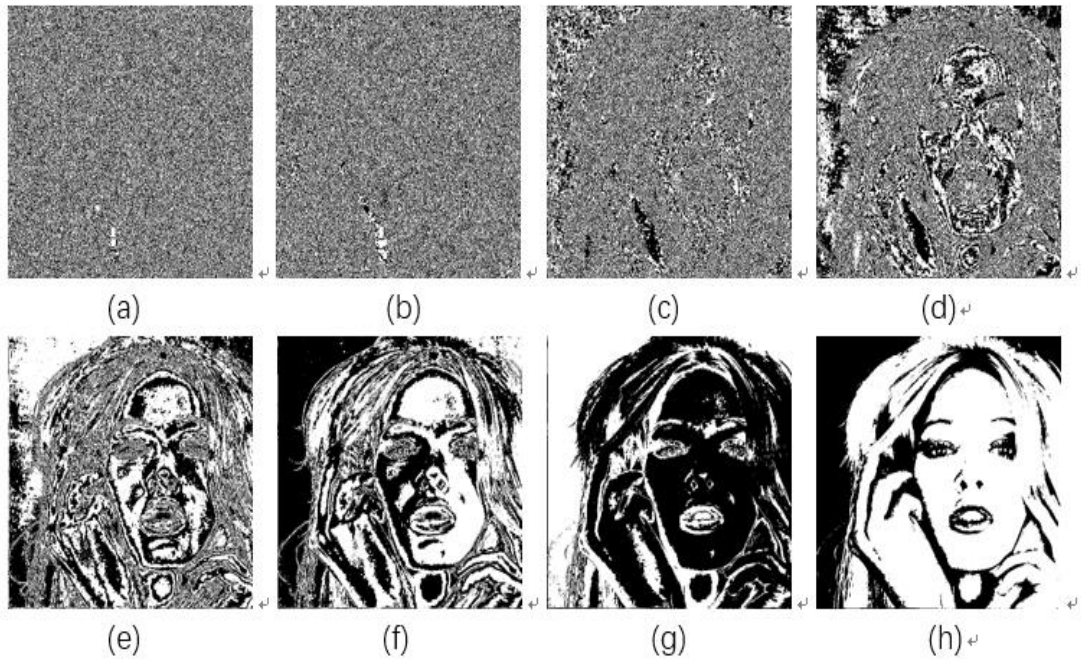

Fig. 7. Plane images of stego-image when the length of secret message is 49000 bits and *bpp* is 2
(a)1th bit; (b)2th bit; (c)3th bit; (d)4th bit; (e)5th bit; (f)6th bit; (g)7th bit; (h)8th bit

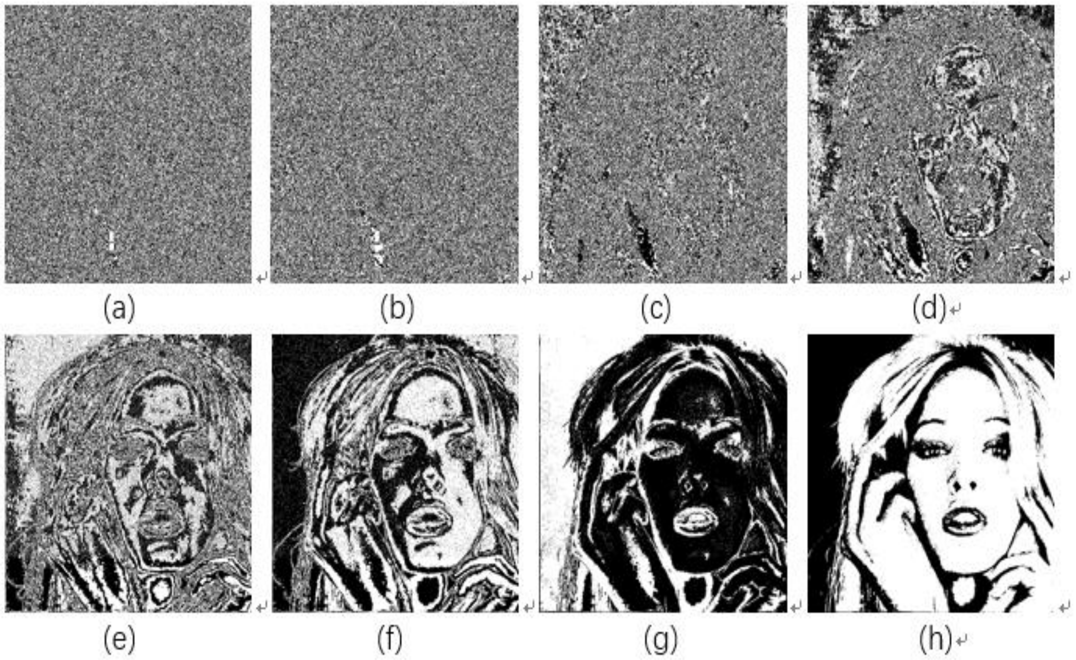

Fig. 8. Plane images of stego-image when the length of secret message is 262144 bits and *bpp* is 5 (a)1th bit; (b)2th bit; (c)3th bit; (d)4th bit; (e)5th bit; (f)6th bit; (g)7th bit; (h)8th bit

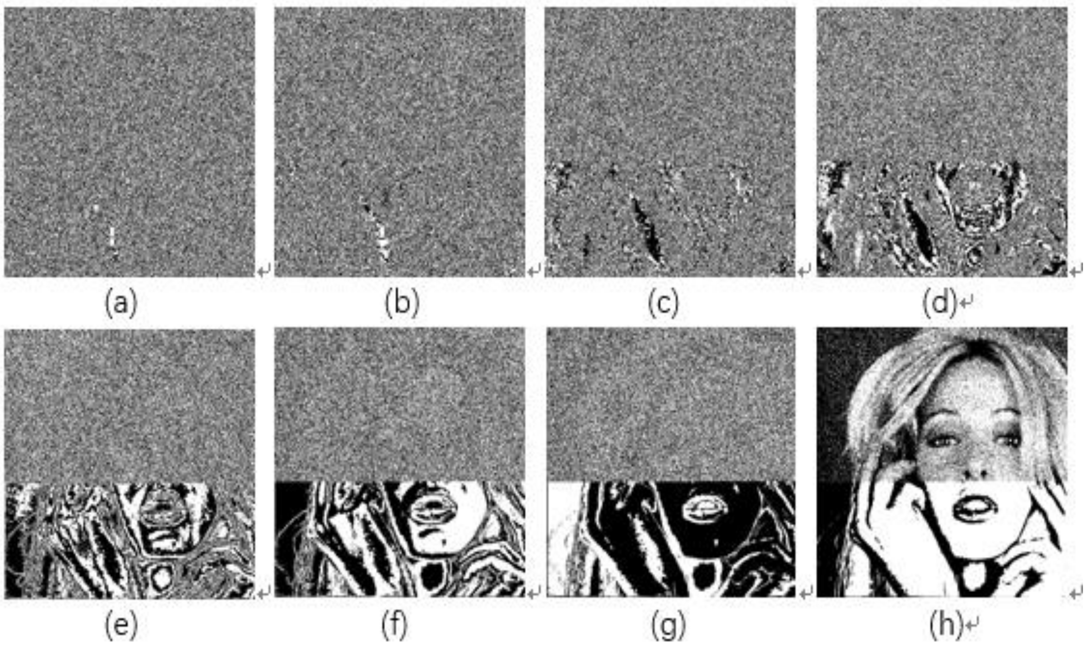

Fig. 9. Plane images of stego-image when the length of secret message is 1048576 bits and *bpp* is 7 (a)1th bit; (b)2th bit; (c)3th bit; (d)4th bit; (e)5th bit; (f)6th bit; (g)7th bit; (h)8th bit

By comparing **Fig. 6** with **Fig. 7** and **Fig. 8**, there is no significant difference in each corresponding plane image, therefore bit plane attack is unable to reveal the secret message hidden in the stego-image. As **Fig. 9** shows, the plane images are different from the previous, this is because excessive secret messages are embedded in the cover-image, the $PSNR$ of stego-image is less than 30 dB. When choosing a stego-image with acceptable quality for an attack experiment, as shown in **Fig. 8**, the security of PAMO can be guaranteed.

5. Conclusion

Based on many information hiding methods proposed in recent years, this paper designs an improved high capacity information hiding method. It relies on pixel value adjustment with modulus operation.

The main idea of our method is using $(2^n + n) - ary$ notational system to represent the secret digit, then updating the cover pixel value by PAMO Algorithm, finally the secret message can be embedded in the cover-image. If users want recover the secret message from stego-image, just need make the stego pixel value divide by $(2^n + n)$, then convert the output to binary system, and combine all the binary bits, finally can get the secret message back.

For gray scale image of size 512 * 512, T.D.Sairam et al. achieved maximum embedding capacity of 10,48,576 bits, at the cost of an average $PSNR$ of 34.805 dB. When facing the same size image, PAMO Algorithm achieved maximum embedding capacity of 10,48,576 bits too, at the cost of an average $PSNR$ of 32.873 dB.

However, if secret message length is not that long, PAMO Algorithm can get higher bpp than other methods. For example, when secret message has only 49000 binary bits, PAMO Algorithm can keep the maximum bpp at 7. That means 7 binary bits could be embedded in one cover pixel, only 7000 cover pixels will be used. But the maximum bpp of SB19 Algorithm still be 4, 12250 pixels in cover-image will be used.

So a conclusion can be drawn: when secret message length is appropriate, PAMO Algorithm has better hiding capacity than other methods. When secret message length is excessive, PAMO Algorithm has the same hiding capacity as SB19 Algorithm, SB19 Algorithm holds the best hiding effect in recent years. Therefore, PAMO Algorithm has greatly improved the hiding capacity.

We upload all codes and images on https://github.com/liteng0264/Information-Hiding.git. If there is anyone interested in PAMO Algorithm, please download it and do some experiments.

Though we have proposed an efficient method which achieves good hiding capacity, there is still a room for us to improve it, and we will try our best to reach this goal.

References

[1] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Information hiding - a survey,” *Proceedings of IEEE*, vol. 87, no. 7, pp. 1062-1078, 1999. Article (CrossRef Link)

[2] S. Y. Shen and L. H. Huang, “A data hiding scheme using pixel value differencing and improving exploiting modification directions,” *Computers and Securrity*, vol. 48, pp. 131-141, 2015. Article (CrossRef Link)

[3] C. K. Chan and L. M. Cheng, “Hiding data in images by simple LSB substitution,” *Pattern Recognittion*, vol. 37, no. 3, pp. 469-474, 2004. Article (CrossRef Link)

[4] J. Mielikainen, “LSB matching revisited,” *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285-287, 2006. Article (CrossRef Link)

[5] C. C. Chang, M. H. Lin, and Y. C. Hu, "A fast and secure image hiding scheme based on LSB substitution," *International Journal Pattern Recognition and Artificial Intelligence*, vol. 16, no. 4, pp. 399-416, 2002. Article (CrossRef Link)

[6] C. H. Yang, "Inverted pattern approach to improve image quality of information hiding by LSB substitution," *Pattern Recognition*, vol. 41, no. 8, pp. 2674-2683, 2008. Article (CrossRef Link)

[7] C. F. Lee, Y. R. Wang, and C. C. Chang, "A steganographic method with high embedding capacity by improving exploiting modification direction," in *Proc. of the 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing (IIHMSP)*, vol. 1, pp. 497-500, 2007. Article (CrossRef Link)

[8] Y. Y. Tsai, C. S. Chan, C. L. Liu, and B. R. Su, "A reversible steganographic algorithm for BTC-compressed images based on difference expansion and median edge detector," *Imaging Science Journal*, vol. 62, no. 1, pp. 48-55, 2014. Article (CrossRef Link)

[9] K. H. Jung, J. G. Yu, S. M. Kim, K. J. Kim, J. Y. Byun, and K. Y. Yoo, "The hiding of secret data using the run length matching method," in *Proc. of KES International Symposium on Agent and Multi-Agent Systmes: Technologies and Applications*, vol. 4496, pp. 1027-1034, 2007. Article (CrossRef Link)

[10] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Communications Letters*, vol. 10, no. 11, pp. 781-783, 2006. Article (CrossRef Link)

[11] H. J. Kim, C. Kim, Y. Choi, S. Wang, and X. Zhang, "Improved modification direction methods," *Computers and Mathematics with Applications*, vol. 60, no. 2, pp. 319-325, 2010. Article (CrossRef Link)

[12] J. C. Cheng, W. C. Kuo, and B. R. Su, "Data-hiding based on sudoku and generalized exploiting modification direction," *Journal of Electronic Science Technology*, vol. 16, no. 2, pp. 123-128, 2018. Article (CrossRef Link)

[13] W. C. Kuo, "Secure modulus data hiding scheme," *KSII Transactions on Internet Information Systems*, vol. 7, no. 3, pp. 610-622, 2013. Article (CrossRef Link)

[14] W. C. Kuo, C. C. Wang, and Y. C. Huang, "Binary power data hiding scheme," *AEU - International Journal of Electronics and Communications*, vol. 69, no. 11, pp. 1574-1581, 2015. Article (CrossRef Link)

[15] W. C. Kuo, C. C. Wang, and H. C. Hou, "Signed digit data hiding scheme," *Information Processing Letters*, vol. 116, no. 2, pp. 183-191, 2016. Article (CrossRef Link)

[16] W. Hong and T. S. Chen, "A novel data embedding method using adaptive pixel pair matching," *IEEE Transactions on Information Forensics Security*, vol. 7, no. 1, pp. 176-184, 2012. Article (CrossRef Link)

[17] K. Jung and K. Y. Yoo, "Improved Exploiting Modification Direction Method by Modulus Operation," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 2, no. 1, pp. 79-88, 2009. Article (CrossRef Link)

[18] W. C. Kuo, S. H. Kuo, C. C. Wang, and L. C. Wuu, "High capacity data hiding scheme based on multi-bit encoding function," *Optik (Stuttg)*, vol. 127, no. 4, pp. 1762-1769, 2016. Article (CrossRef Link)

[19] T. D. Sairam and K. Boopathybagan, "An improved high capacity data hiding scheme using pixel value adjustment and modulus operation," *Multimedia Tools Applications*, vol. 79, pp. 17003-17013, 2019. Article (CrossRef Link)

[20] Y. Zhang, S. Wang, T. Li, B. Liu, and D. B. Pan, "Modulus Calculations on Prime Number Algorithm for Information Hiding with High Comprehensive Performance," *IEEE Access*, vol. 8, pp. 85309-85320, 2020. Article (CrossRef Link)

[21] USC-SIPI image database. Article (CrossRef Link)

Teng Li was born in Xuancheng, Anhui, China in 1997. He received the B.E. degree in computer science and technology from the Southwest University of China in 2019. Since 2019, he is studying for a master's degree in Southwest University of China. His research interests include machine learning and information hiding technology. Mr. Li is a member of CCF. In 2017, he won the third prize in Mathematical Modeling Competition of Southwest University of China.

Yu Zhang was born in Banan, Chongqing, China in 1979. He received the B.S. and M.S. degrees in computer science and technology from the Southwest Normal University of China, in 2002 and 2005, and the Ph.D. degree in communication engineering from Chongqing University, China, in 2016. From 2005 to 2012, he was a Lecture with the Department of Automation of China Southwest University. Since 2012, he has been an Associate Professor with the College of Computer and Information Science, Chongqing, China. He is the author of five books, more than 30 articles, and more than 10 inventions. His research interests include optimization in automation systems, machine learning, intelligent control, integration of control systems, and industrial communications. He drafted an international standard about devices integration and more than 10 national standards. Dr. Zhang was a member of IEC and SC2/TC124/SAC, and a recipient of the Standard Innovation Contribution Award in 2007, Chongqing Science and Technology Progress Award (second class) in 2014 and Chongqing Science and Technology Achievements Award in 2015.

Sha Wang was born in Changshou, Chongqing, China in 1995. She received the B.S. degree in computer science and technology from Shenyang University of Technology of China, in 2018. Since 2019, she is studying for a master's degree in computer science and technology in Southwest University of China. Her research interests include machine learning and network security. Ms. Wang is a member of CCF, she won the third prize in the Chinese Computer Competition in 2017.

Jun-jie Sun was born in Huzhou, Zhejiang, China in 1997. He received the B.E. degree in Computer science and technology from the Zhejiang Gongshang University, China, in 2019, now he is studying for a master's degree at Southwest University. His research interests are computer vision and image processing. Mr.Sun is a member of CCF. In 2018, he won the second prize in e-commerce competition of Zhejiang Gongshang University and the third prize in 2017 undergraduate physics innovation competition of Zhejiang Province.