

Received September 2, 2020, accepted September 7, 2020, date of publication September 10, 2020, date of current version September 23, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3023197

High Embedding Capacity Data Hiding Technique Based on EMSD and LSB Substitution Algorithms

SERDAR SOLAK 

Information System Engineering, Kocaeli University, 41001 İzmit, Turkey

e-mail: serdars@kocaeli.edu.tr

ABSTRACT Data hiding called steganography is a security technique to protect secret data throughout the transmission from malicious attackers. The purposes of steganography are to obtain good stego-image quality, high embedding-capacity, low computational complexity, visual imperceptibility, undetectability, and more security. In this paper, we offer a new hybrid image steganography technique based on least significant bit (LSB) substitution and enhanced modified signed digit (EMSD) algorithms. The proposed algorithm utilizes n adjacent cover image pixels to hide the secret data with EMSD algorithm, and least significant k -bit for LSB substitution algorithm. Hence, it has more embedding capacity than the EMSD algorithm and exploiting modification direction (EMD) based algorithms. We obtain that the stego-image quality is better than 43 dB when the payload is 2.404 bpp. The results of experiment represent that this algorithm ensures high embedding-capacity while preserving acceptable visual stego image quality that can be undetectable by human eyes. Also, the hybrid of the EMSD and LSB substitution algorithms is to difficult for malicious people to consolidate data by scrambling secret data bits.


INDEX TERMS Data hiding, data security, exploiting modification direction (EMD), enhanced modified signed digit (EMSD), generalized exploiting modification direction (GEMD), least significant bit (LSB), sparse modified signed digit (SMSD).

I. INTRODUCTION

As a result of the rapid development of information and communication technologies in recent years, people have started to store large numbers of digital data via obtained using the camera, self-phone, computer. These data are also distributed and transferred more efficiently using the network, internet, and cloud technologies. Malicious people may obtain and alter these data during communication, so it is a critical issue to protect data security. In order to ensure the security of data, there are generally two techniques as cryptography and steganography. The cryptography [1] encrypts the secret data in such a way that it has incomprehensible data to malicious people. The secret data is scrambled using a secret key, and only people with the secret key can decrypt the original message. In cryptology, there are two categories: symmetric methods using public keys and asymmetric methods using the public for encryption and private key to decrypt. Triple Data Encryption algorithm (3DES), Advance Encryption Standard (AES), Data Encryption Standard (DES), and an

alternative encryption method Blowfish use the symmetric method to encrypt secret data. Also, The RSA (Rivest, Shamir & Adelman) algorithm, which uses an asymmetric encryption method, is one of the widely used encryption algorithms [2]. If the secret keys are obtained by malicious people, the secret data can be intercepted. Steganography is another popular technique that has been used to secure data in recent years. Steganography embeds secret data into text, audio, image, and video files so malicious people cannot notice the existence of secret data.

The data hiding technique as namely Steganography is an important method that dates back to many years. People used various materials such as words, images, trees, and furs over different time periods in order to protect their secret information from being understood by others. Nowadays, the steganography technique is utilized in many applications for instance medical, military, commercial, authentication, internet of things (IoT) based applications [3]–[7]. In these applications, the image and video files are generally preferred due to their high embedding-capacity as carrier media in order to hide data [8]–[10]. When the carrier media used to hide secret data is the image, it is called image

The associate editor coordinating the review of this manuscript and approving it for publication was Noor Zaman .

steganography. The image steganography comprises generally a cover-image, data hiding algorithm, secret data, and stego-image. While the cover-image expresses the carrier media which is utilized to hide information, the stego-image represents the result media that includes the secret data. The difference between cover and stego images is so small that the human eye cannot perceive it. Also, the secret data is generally applied to the encryption process with secret key before the embedding procedure to happen more securely. After the encryption process, the stego image is obtained by using data hiding algorithm [8], [10]. A good data hiding algorithm has to have high embedding capacity, good visual quality, imperceptibility and low complexity. However, in most cases, it is hard to have a data hiding method that includes these features at the same time. Therefore, researchers have improved on data hiding algorithms that require low computation time with acceptable visual quality at high embedding capacity in recent years. The least significant bits (LSB) [10]–[14], pixel value difference (PVD) [15]–[19], and exploiting modification direction (EMD) [20]–[24] algorithms are the most commonly used methods of image steganography in the spatial domain. The LSB substitution, PVD, and EMD algorithms are used together in the literature and it is observed good visual quality, high embedding capacity, and more secure [24]–[33].

A well-known data hiding technique is LSB substitution, in which uses the k -least significant bits of the cover image pixel values that are used in data embedding. While a grayscale cover image pixel hides one bit for the $k = 1$, colored cover image conceals three bits. In the LSB method, the embedding capacity increases as a result of increasing the k value, but the imperceptibility and stego image quality decrease. The PVD algorithm, which is another widely used technique, uses adjacent pixel pairs of the cover image when the secret data is concealed. The difference between these two adjacent pixel pairs determines the number of secret data bits to embed. Since the difference is usually tiny in adjacent pixel pairs, it is concealed average three bits of secret data, but adjacent pixel pairs where the difference is higher, more bits of data can be hidden. The EMD algorithm [34] is utilized to embed secret data for n adjacent pixels values in $(2n+1)$ -ary notational system. The only one in n adjacent pixels is modified to conceal the secret data. Therefore, the EMD based algorithms have usually low embedding capacity but very good image quality. There are various studies in the literature in order to increase the data capacity of the EMD algorithm [35]–[38].

While evaluating the data hiding algorithms used in image steganography, it is observed that embedding capacity or payload, peak signal-to-noise ratio (PSNR), and structural similarity index (SSIM) are used as measurement parameters in literature. The performance of data hiding algorithms is to obtain high and acceptable PSNR and SSIM values at high embedding capacity. It is also desired that the secret data in the stego image is imperceptible and resistant to attacks.

In this article, we present a high embedding capacity data hiding technique based on enhanced modified signed digit (EMSD) and LSB substitution algorithms. The purpose of this algorithm is to increase EMSD algorithms embedding capacity while the stego image is to keep the visual quality acceptable. As reported by the studies in the literature [23], [26] and [37], while the PSNR value above 40 dB is good stego image quality, the PSNR value above 30 dB is also acceptable. It means that the cover and stego images are considered visually indistinguishable.

We have also provided more security by using these two algorithms together. In the design of the algorithm, the EMSD algorithm is implemented first and new pixel values are obtained, then, LSB substitution is applied using the new pixel values. While the number of secret data bits in each pixel for LSB substitution is represented by k , the number of adjacent pixels for the used EMSD algorithm is symbolized by n . The contributions of this paper include the following: (1) A new data hiding algorithm is proposed based on EMSD and LSB substitution algorithm, (2) The proposed algorithm improves significantly the embedding capacity of the EMSD based data hiding algorithm, (3) In spite of the increased embedding capacity, there is an acceptable stego image quality, (4) A solution for the fall of boundary problem (FOBP) has been developed.

The remainder of this article is organized as follows. In Section II, related works are presented in detail, which contains the LSB substitution algorithm, EMD algorithm, GEMD algorithm, SMSD algorithm and EMSD algorithm. The proposed method, a high embedding capacity data hiding technique based on EMSD and LSB substitution algorithms, is given in section III. The experimental results and comparisons to evaluate the proposed algorithm is offered in section IV. The conclusion is finally presented in section V.

II. RELATED WORK

In this section, the LSB substitution, EMD, GEMD, SMSD and EMSD algorithms are introduced in detail. The LSB substitution algorithm is easy to use and rapidly, and owns high embedding capacity and good PSNR values. The EMD and GEMD data hiding algorithms have good stego image quality. The SMSD and EMSD algorithms, which have good stego image quality, utilize three symbols as 1, 0, $\underline{1}$, and weight minimization algorithm (WMA) to obtain the minimum weighted MSD representing an integer. But, The EMD, GEMD, SMSD and EMSD algorithm have the low embedding capacity and required some calculation for embedding and extracting procedure.

A. LSB SUBSTITUTION

The LSB substitution algorithm is the most common algorithm that hides secret data to the least significant bits of the cover image. It is easy to implement, fast and has high embedding capacity and good PSNR and SSIM values. As seen in literature studies, secret data is usually hidden in the first least significant bits of the cover image pixels, because the

presence of hidden information cannot be detected by the human eye.

Since the LSB Substitution algorithm uses the least significant bits in the data hiding stage, the secret data is first converted to the binary number system. After, the least significant bits of the cover image pixels are changed with the secret data bits stream. The LSB Substitution algorithm has the embedding and extracting process shown in the following;

Embedding Process

Input: W x H size of cover image (C_1), Secret Data (SD), the least significant bits number (k)

Output: W x H size of stego image (S_1)

Step1: All of the SD data to be hidden is converted to binary system (b). $(b)_2 = (b_{n-1}, b_{n-2}, \dots, b_0)_2$ where $b_t \in \{0, 1\}$

Step2: Calculate the stego pixel value(P'_i) using (1).

$$P_i = P_i - (P_i \bmod 2^k) \quad P'_i = P_i + \sum_{t=0}^{k-1} b_t x 2^t \quad (1)$$

Step 3: Repeat step2 until all secret data is embedded into cover image. Finally, the stego image is obtained (S_1).

Extracting Process

Input: W x H size of stego image (S_1), the least significant bits number (k)

Output: Secret Data (SD)

Step1: The least significant k bits value of each stego pixel are computed by (2).

$$d_i = (P'_i \bmod 2^k) \quad (2)$$

Step2: Each d_i value is converted into a k-bit binary value.

Step3: Repeat step2 and step3 until all secret data is extracted from stego image.

Step4: The whole extracted bit stream is combined and meaningful secret data is obtained.

Example

Embedding process:

Four pixels from the grayscale Baboon cover image (145, 116, 77, 71), $k=2$, and $SD='A'$ (decimal value =65), $(65)_2 = (01000001)_2$ converted decimal to binary and obtained $(b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)$.

Stego image pixels are computed according to (1).

$$P'_{i=1,2,3,4} = (145, 116, 76, 69).$$

Extracting process:

The secret data is obtained by (2). $d_{(1,2,3,4)} = (1, 0, 0, 1)$.

After, converted to k-bit binary values (01, 00, 00, 01).

Finally, $(SD)_{10} = 65$ ('A').

B. EMD ALGORITHM

The EMD algorithm is offered by Zhang and Wang in 2006 [34]. Their algorithm hides secret data which is in $(2n + 1)$ -ary notational system into n adjacent pixels in images. The algorithm replaces at most one pixel in the n adjacent pixels groups for rapid and effective data embedding. In these groups, at most, only one-pixel value is decreased or increased by 1. For instance, the 7-ary notational system data stream is hidden in three adjacent pixels. The EMD has good

stego image quality but low embedding capacity. The payload of EMD algorithm is computed using (3).

$$P_{EMD} = \frac{\log_2(2n + 1)}{n} \quad (3)$$

The EMD algorithm has the embedding and extracting process shown in the following;

Embedding Process

Input: W x H size of cover image (C_1), adjacent pixels (n), Secret Data (SD),

Output: W x H size of stego image (S_1)

Step1: The n adjacent pixels (p_1, p_2, \dots, p_n) are taken from the cover image (C_1).

Step2: Compute g_{EMD} value using (4)

$$g_{EMD}(p_1, p_2, \dots, p_n) = \left[\sum_{k=1}^n (p_k \times k) \right] \bmod (2 \times n + 1) \quad (4)$$

Step3: The secret data is converted to $(2n + 1)$ -ary notational system (SD).

Step4: Calculate the difference value (diff) between g_{EMD} and SD_i using (5).

$$diff = (SD_i - g_{EMD}) \bmod (2 \times n + 1) \quad (5)$$

Step5: According to diff value, compute stego pixels values $(p'_1, p'_2, \dots, p'_n)$ using (6).

$$\left\{ \begin{array}{l} p_1, p_2, \dots, p_n \longrightarrow diff = 0 \text{ or } diff = g_{EMD} \\ p'_{diff} = p_{diff} + 1, \longrightarrow diff \leq n \\ p'_{(2n+1-diff)} = p_{(2n+1-diff)} - 1, \longrightarrow diff > n \end{array} \right\} \quad (6)$$

Step6: Repeat step1 to 5 until all secret data is embedded into cover image. Finally, the stego image is obtained (S_1).

Extracting Process

Input: W x H size of stego image (S_1), adjacent pixels (n),

Output: Secret Data (SD)

Step1: The n adjacent pixels $(p'_1, p'_2, \dots, p'_n)$ are taken from the stego image (S_1).

Step2: Compute $(2n + 1)$ -ary Secret Data (SD_i) using (7),

$$(SD_i)_{(2n+1)} = \left[\sum_{k=1}^n (p'_k \times k) \right] \bmod (2 \times n + 1) \quad (7)$$

Step3: Repeat step1 and step2 until all secret data is extracted from stego image.

Example

Embedding process:

Four pixels from the grayscale Baboon cover image (145, 116, 77, 71), $n=4$, $SD_1 = (5)_{10}$ Firstly, calculate the $g_{emd}(p_1, p_2, p_3, p_4)$ value, $g_{emd}(145, 116, 77, 71) = (145 \times 1 + 116 \times 2 + 77 \times 3 + 71 \times 4) \bmod (2 \times 4 + 1)$ $g_{emd} = 892 \bmod 9 = 1$, then convert secret data (SD_1) to $(2 \times 4 + 1)$ -ary notational system $(5)_9$, calculate diff value,

$diff = (5 - 1) \bmod 9 = 4$, $diff \leq n$, only changed p'_{diff} , using (5). $p'_4 = p_4 + 1 = 71 + 1 = 72$.

Finally, stego pixel group is obtained as (145, 116, 77, 72).

Extracting process:

The secret data is obtained by (6),

$$(SD_1)_{(9)} = (145x1 + 116x2 + 77 x3 + 72x4) \text{ mod } 9$$

$$(SD_1)_{(9)} = 896 \text{ mod } 9 = 5$$

C. GEMD ALGORITHM

The generalized exploiting modification direction (GEMD) data hiding algorithm is proposed by Kuo and Wang in 2013 [35]. The payload of the EMD algorithm is enhanced with the GEMD algorithm, which also ensures better stego-image quality. The payload is higher than 1 bpb when n adjacent pixels groups increases in GEMD algorithm. We calculate the payload of GEMD using (8).

$$P_{GEMD} = (n + 1)/n \tag{8}$$

The GEMD algorithm has the embedding and extracting process shown in the following;

Embedding Process

Input: W x H size of cover image (C₁), adjacent pixels (n), Secret Data (SD),

Output: W x H size of stego image (S₁)

Step1: The n adjacent pixels (p₁, p₂, , p_n) are taken from the cover image (C₁).

Step2: Step2: Compute g_{GEMD} value using (9).

$$g_{GEMD}(p_1, p_n) = \left[\sum_{k=1}^n (p_k \times (2^k - 1)) \right] \text{ mod } (2^{n+1}) \tag{9}$$

Step3: The (n+1) bits secret data is converted to decimal (SD).

Step4: Calculate the difference value (diff) between g_{GEMD} and SD_i using (10).

$$diff = (SD_i - g_{GEMD}) \text{ mod } (2^{n+1}) \tag{10}$$

Step5: According to diff value, compute stego pixels values (p'₁, p'₂, p'_n) using the following situations;

- a) if diff = 2ⁿ → p'₁ = p₁ + 1, p'_n = p_n + 1
- b) if 0 < diff < 2ⁿ → convert diff value (n+1) bits binary data (s_n, s_{n-1}, s₀)₂ for (x=n; x≥1;x=x-1)
 - if (s_x = 0 and s_{x-1} = 1) p'_x = p_x + 1
 - else if (s_x = 1 and s_{x-1} = 0) p'_x = p_x - 1
 - else p'_x = p_x
- end
- c) if diff > 2ⁿ → temp = 2ⁿ⁺¹-diff, convert temp value (n+1) bits binary data (s_n, s_{n-1}, s₀)₂ for (x=n; x≥1;x=x-1)
 - if (s_x = 0 and s_{x-1} = 1) p'_x = p_x - 1
 - else if (s_x = 1 and s_{x-1} = 0) p'_x = p_x + 1
 - else p'_x = p_x
- end

Step6: Repeat step1 to 5 until all secret data is embedded into cover image. Finally, the stego image is obtained (S₁).

Extracting Process

Input: W x H size of stego image (S₁), adjacent pixels (n),

Output: Secret Data (SD)

Step1: The n adjacent pixels (p'₁, p'₂, , p'_n) are taken from the stego image (S₁).

Step2: Compute Secret Data (SD_i) using (11),

$$(SD_i)_{10} = \left[\sum_{k=1}^n (p'_k \times (2^k - 1)) \right] \text{ mod } (2^{n+1}) \tag{11}$$

Step3: Convert decimal secret data to (n+1) bits binary data

Step4: Repeat step 1 to 3 until all secret data is extracted from stego image.

Example

Embedding process:

Four pixels from the grayscale Baboon cover image (145, 116, 77, 71), n=4, (SD₁)₂ = (11001)₂ →5-bit secret data, →(SD₁)₁₀ = (25)₁₀ Firstly, Compute the g_{GEMD}(p₁,p₂,p₃,p₄) value, g_{GEMD}(145,116,77,71) = 145+116 × 3+77 × 7+71 × 15 mod (2⁵) g_{GEMD} = 2097 mod 32 = 17, Then, calculate diff value, diff=(25 - 17) mod 32 = 8, 0< 8 < 16 → follow step 5b, convert diff value 5-bit binary, diff=(01000) Finally, stego pixel group is obtained as (145,116, 76, 72).

Extracting process:

The secret data is obtained by (11),

$$(SD_1)_{(10)} = (145x1 + 116x3 + 76 x7 + 72x15) \text{ mod } 32$$

$$(SD_1)_{(10)} = 2105 \text{ mod } 32 = 25, \text{ and convert 5-bit binary data } (SD_1)_2 = (11001)_2.$$

D. SMSD ALGORITHM

The sparse modified signed digit (SMSD) algorithm has been proposed to increase payload of EMD algorithm and to improve the stego image quality [36]. The SMSD algorithm uses three symbols as 1, 0, 1, and a weight minimization algorithm. An n-bit binary secret data for SMSD has two MSD specifications, which are the maximum n/2 number of non-zero bits, and no adjacent non-zero bits. The number of integers (T_n) that is represented by n-bit SMSD is calculated using (12). When n is even number, the binary secret data is between (-V_n) (1010... 10) and (V_n) (1010... 10), otherwise (1010... 101) through (1010... 101). The payload of SMSD algorithm is computed using (13).

$$\text{if } n \text{ is even, } V_n = \frac{2x4^t - 2}{3}, \quad T_n = 2xV_n + 1, \{t = n/2\}$$

$$\text{if } n \text{ is odd, } V_n = \frac{4^{t+1} - 1}{3},$$

$$T_n = 2xV_n + 1, \{t = (n - 1)/2\} \tag{12}$$

$$P_{SMSD} = \frac{\log_2(T_n)}{n} \tag{13}$$

The SMSD algorithm has the embedding and extracting process shown in the following;

Embedding Process

Input: W x H size of cover image (C₁), adjacent pixels (n), Secret Data (SD),

Output: W x H size of stego image (S₁)

Step1: Calculate T_n value using (12).

Step2: The n adjacent pixels (p_1, p_2, \dots, p_n) are taken from the cover image (C_1).

Step3: Calculate g_{SMSD} value using (14).

$$g_{SMSD}(p_1, \dots, p_n) = \left[\sum_{k=1}^n (p_k \times (2^{k-1})) \right] \text{mod} (T_n) \quad (14)$$

Step4: Calculate the difference value (diff) between g_{SMSD} and SD_i using (15).

$$\text{diff} = (SD_i - g_{SMSD}) \text{mod} (T_n) \quad (15)$$

Step5: Convert diff value to binary system, and use weight minimization algorithm for SMSD.

diff value (n) bits binary data $(s_n, s_{n-1}, \dots, s_1)_2$

Step6: Calculate stego pixels values $(p'_1, p'_2, \dots, p'_n)$ using (16).

$$p'_1 = p_1 + s_1, \dots, p'_n = p_n + s_n \quad (16)$$

Step7: Repeat step 2 to 6 until all secret data is embedded into cover image. Finally, the stego image is obtained (S_1).

Extracting Process

Input: $W \times H$ size of stego image (S_1), adjacent pixels (n),
Output: Secret Data (SD)

Step1: Compute T_n value using (12).

Step2: The n adjacent pixels $(p'_1, p'_2, \dots, p'_n)$ are taken from the stego image (S_1).

Step3: Compute Secret Data (SD_i) using (17),

$$SD_i = \left[\sum_{k=1}^n (p'_k \times (2^{(k-1)})) \right] \text{mod} (T_n) \quad (17)$$

Step4: Repeat step 2 and 3 until all secret data is extracted from stego image.

Example

Embedding process:

Four pixels from the grayscale Baboon cover image (145, 116, 77, 71), $n=4$, $SD_1 = 20$

Firstly, Compute the T_n value using (12), n is even, so $V_n = (2 \times 4^2 - 2)/3$, $V_n = 10$, $T_n = 2 \times 10 + 1$, $T_n = 21$. Next, Compute the g_{SMSD} value, $g_{SMSD}(145, 116, 77, 71) = 145 + 116 \times 2 + 77 \times 4 + 71 \times 8 \text{mod} (21)$, $g_{SMSD} = 1253 \text{mod} 21 = 14$. Then, compute diff value, $\text{diff} = (20 - 14) \text{mod} 21 = 6$, convert the diff value 4-bit SMSD using weight minimization algorithm. (s4, s3, s2, s1), (0110) is not suitable for SMSD, so we use (1010). Finally, stego pixel group is obtained as (145, 115, 77, 72).

Extracting process:

First, we compute T_n value using (12), $T_n = 21$. Then, the secret data is obtained by (17),

$$\begin{aligned} SD_1 &= (145 \times 1 + 115 \times 2 + 77 \times 4 + 72 \times 8) \text{mod} 21 \\ SD_1 &= 1259 \text{mod} 21 = 25. \end{aligned}$$

E. EMSD ALGORITHM

The enhanced MSD (EMSD) data hiding algorithm is proposed by Liu *et al.* in 2019 [38] to increase payload of SMSD algorithm. The EMSD algorithm uses a modified weight minimization algorithm (MWMA) for secret data EMSD binary representation. The nonzero bits located adjacent do not cause problems in MWMA. The number of integers (M_n) that can be represented by n -bit EMSD is calculated using (18). When n is even number, the binary secret data is between $(-U_n)$ ($1010\dots 1100$) and (U_n) ($1010\dots 1100$), otherwise ($11010\dots 1100$) through ($11010\dots 1100$). The payload of EMSD algorithm is computed using (19).

$$\text{if } n \text{ is even, } U_n = \frac{2x4^t - 2}{3}, \quad M_n = 2xU_n + 3, \quad \left\{ t = \frac{n}{2} \right\}$$

$$\text{if } n \text{ is odd, } U_n = \frac{5x4^t - 2}{3},$$

$$M_n = 2xU_n + 3, \quad \left\{ t = \frac{(n-1)}{2} \right\} \quad (18)$$

$$P_{EMSD} = \frac{\log_2(M_n)}{n} \quad (19)$$

The EMSD algorithm has the embedding and extracting process shown in the following;

Embedding Process

Input: $W \times H$ size of cover image (C_1), adjacent pixels (n),
Secret Data (SD),

Output: $W \times H$ size of stego image (S_1)

Step1: Calculate M_n value using (18).

Step2: The n adjacent pixels (p_1, p_2, \dots, p_n) are taken from the cover image (C_1).

Step3: Calculate g_{EMSD} value using (20).

$$g_{EMSD}(p_1, \dots, p_n) = \left[\sum_{k=1}^n (p_k \times (2^{k-1})) \right] \text{mod} (M_n) \quad (20)$$

Step4: Calculate the difference value (diff) between g_{EMSD} and SD_i using (21).

$$\text{diff} = (SD_i - g_{EMSD}) \text{mod} (M_n) \quad (21)$$

Step5: Convert diff value to binary system, and use modified weight minimization algorithm for EMSD.

diff value (n) bits binary data $(s_n, s_{n-1}, \dots, s_1)_2$

Step6: Calculate stego pixels values $(p'_1, p'_2, \dots, p'_n)$ using (22).

$$p'_1 = p_1 + s_1, \dots, p'_n = p_n + s_n \quad (22)$$

Step7: Repeat step2 to 6 until all secret data is embedded into cover image. Finally, the stego image is obtained (S_1).

Extracting Process

Input: $W \times H$ size of stego image (S_1), adjacent pixels (n),
Output: Secret Data (SD)

Step1: Calculate M_n value using (18).

Step2: The n adjacent pixels $(p'_1, p'_2, \dots, p'_n)$ are taken from the stego image (S_1).

Step3: Calculate Secret Data (SD_i) using (23).

$$SD_i = \left[\sum_{k=1}^n \left(p'_k \times 2^{(k-1)} \right) \right] \bmod (M_n) \quad (23)$$

Step4: Repeat step 2 and 3 until all secret data is extracted from stego image.

Example

Embedding process:

Four pixels from the grayscale Baboon cover image (145, 116, 77, 71), n=4, SD₁ = 14

Firstly, Compute the M_n value using (18), n is even, so U_n = (2 × 4²-2)/3, U_n = 10, M_n = 2 × 10+3, M_n = 23.

Next, Compute the g_{EMSD} value, g_{EMSD}(145, 116, 77, 71) = 145+ 116 × 2 +77 × 4 +71 × 8 mod (23),

$$g_{EMSD} = 1253 \bmod 23 = 11.$$

Then, compute diff value, diff = (14-11) mod 23 = 3, convert the diff value 4-bit EMSD using MWMA. (s4,s3,s2,s1), The EMSD can be used (0011) binary value.

Finally, stego pixel group is obtained as (146,117, 77, 71).

Extracting process:

First, we compute M_n value using (18), M_n = 23.

Then, the secret data is obtained by (22),

$$SD_1 = (146x1 + 117x2 + 77 x4 + 71x 8) \bmod 23$$

$$SD_1 = 1256 \bmod 23 = 14.$$

III. PROPOSED ALGORITHM

In this section, a new hybrid LSB substitution and EMSD based high capacity data hiding algorithm is proposed. The principal aim of the proposed LSB substitution and EMSD based algorithm is that makes a serious enhancement in the embedding capacity while providing acceptable visual quality. In the proposed algorithm, the bits of secret data are hidden using a group of n-adjacent pixels obtained from the cover image and k-least significant bits. In the proposed algorithm, firstly, the temporary pixel values for n adjacent pixels are calculated using the k value. If the stego pixel values are likely to generate FOBP (p'_k < 0 or, p'_k > 255), the temporary pixel values are updated in a way that does not generate FOBP. Then, data hiding is performed on this pixel value using the EMSD algorithm. Finally, the k-bit of secret data is hidden each pixel in n adjacent pixel groups by the LSB substitution algorithm. The payload of the proposed algorithm is calculated using (24) since we are using both EMSD and LSB algorithms.

$$P_{proposed} = \frac{nxk + \log_2(M_n)}{n} \quad (24)$$

The proposed algorithm has the embedding and extracting process shown in the following:

Embedding Process

Input: W x H size of cover image (C₁), adjacent pixels (n), (k) value for least significant bits, Secret Data (SD),

Output: W x H size of stego image (S₁)

Step1: Calculate M_n value using (18).

Step2: The n adjacent pixels (p₁, p₂, , p_n) are taken from the cover image (C₁).

Step3: Calculate the temporary pixel values using (25).

$$(p_1^{temp}, \dots, p_n^{temp}) = \left(\left\lfloor \frac{p_1}{2^k} \right\rfloor, \dots, \left\lfloor \frac{p_n}{2^k} \right\rfloor \right) \quad (25)$$

Step4: Check to the temporary pixel values for FOBP, and recalculate temporal pixel values using (26), if necessary.

$$(p_1^{temp}, \dots) = \begin{pmatrix} \text{if } (p_1^{temp} = 0, \dots) \rightarrow (p_1^{temp} + 1, \dots) \\ \text{if } ((p_1^{temp} + 1) x 2^k \geq 255, \dots) \rightarrow (p_1^{temp} - 1, \dots) \end{pmatrix} \quad (26)$$

Step5: Calculate g_{EMSD} value according to (p₁^{temp}, . . . , p_n^{temp}) using (27).

$$g_{EMSD}^{temp} = \left(\sum_{t=1}^n \left(p_t^{temp} x 2^{(t-1)} \right) \right) \bmod (M_n) \quad (27)$$

Step6: We calculate the difference value (diff) between g_{EMSD}^{temp} and SD_i using (28).

$$diff = (SD_i - g_{EMSD}^{temp}) \bmod (M_n) \quad (28)$$

Step7: Convert diff value to binary system, and use modified weight minimization algorithm [38] for EMSD algorithm.

diff value (n) bits binary data (s_n, s_{n-1}, . . . s₁)₂

Step8: Calculate the temporary stego pixels values (p₁^{temp'}, p₂^{temp'}, , p_n^{temp'}) using (29).

$$p_1^{temp'} = p_1^{temp} + s_1, \dots, p_n^{temp'} = p_n^{temp} + s_n \quad (29)$$

Step9: nxk bits of data (b) are taken from secret data (SD).

Step10: It is used (29) to hide k-bit secret data for each pixel using the LSB substitution algorithm.

$$(p_1^l, \dots) = \left((p_1^{temp'} x 2^k) + \sum_{s=1}^k b_{s-1} x 2^{s-1}, \dots \right) \quad (30)$$

Step11: Repeat step2 to 10 until all secret data is embedded into cover image. Finally, the stego image is obtained (S₁).

Figure 1 shows flowchart of the proposed algorithm for embedding process.

Extracting Process

Figure 1 shows flowchart of the proposed algorithm for embedding Input: W x H size of stego image (S₁), adjacent pixels (n), (k) value for least significant bits

Output: Secret Data (SD)

Step1: Calculate M_n value using (18).

Step2: The n adjacent pixels (p'₁, p'₂, , p'_n) are taken from stego-image (S₁).

Step3: Firstly, we extract k-bits secret data from n adjacent stego pixels using (31).

$$(b_1, \dots, b_n) = \left(p_1^l - \left\lfloor \frac{p_1^l}{2^k} \right\rfloor x 2^k, \dots, p_n^l - \left\lfloor \frac{p_n^l}{2^k} \right\rfloor x 2^k \right) \quad (31)$$

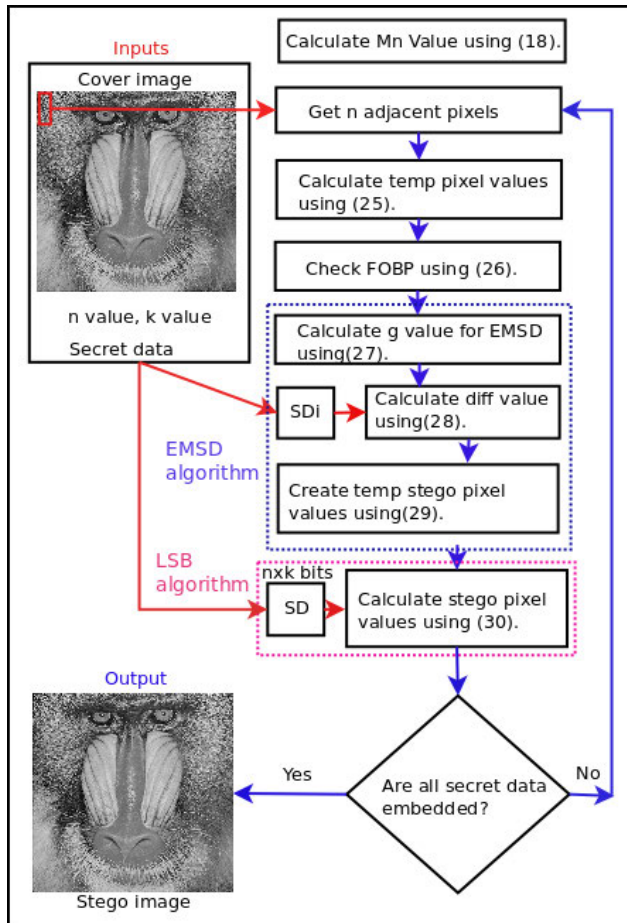


FIGURE 1. Flowchart of the proposed algorithm data hiding process.

Step4: Calculate the temporary stego pixels values for EMSD algorithm using (32).

$$(p_1^{temp'}, \dots, p_n^{temp'}) = \left(\left\lfloor \frac{p_1^l}{2^k} \right\rfloor, \dots, \left\lfloor \frac{p_n^l}{2^k} \right\rfloor \right) \quad (32)$$

Step5: Calculate Secret Data (SD_i) using (33),

$$SD_i = \left(\sum_{t=1}^n p_i^{temp'} x^{2^{t-1}} \right) \text{ mod } (M_n) \quad (33)$$

Step6: Repeat step 2 to 5 until all secret data is extracted from stego image.

Figure 2 shows flowchart of the proposed algorithm for data extracting process. We provide below two examples of data hiding and extracting process to demonstrate the operation of the proposed algorithm.

Example-1

Embedding process:

$$(p_1, p_2, p_3, p_4) = (145, 116, 77, 71), n=4, k=2, SD_1 = 17, SD_2 = 147$$

Firstly, Compute the M_n value using (18), n is even, so $U_n = (2 \times 4^2 - 2)/3, U_n = 10, M_n = 2 \times 10 + 3, M_n = 23$.

Then, we calculate $(p_1^{temp}, p_2^{temp}, p_3^{temp}, p_4^{temp}) = (36, 29, 19, 17)$ using (25). There is no FOBP according to (26).

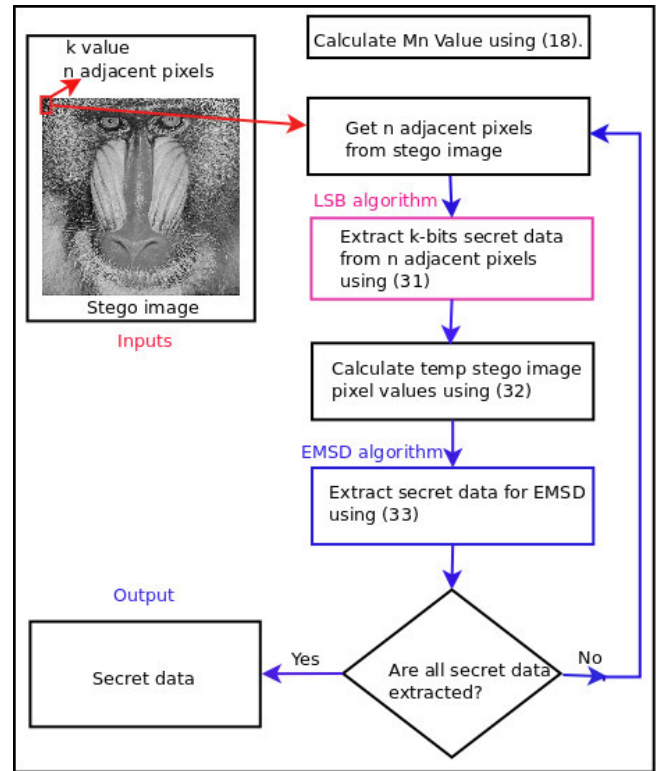


FIGURE 2. Flowchart of the proposed algorithm for data extracting process.

Calculate $g_{EMSD}^{temp} = 7$ using (26). $diff = 17 - 7 = 10$ in $\{(-M_n/2) \leq diff \leq (+M_n/2)\}$ $diff$ (1010) is represented in EMSD algorithm using modified weight minimization algorithm. We calculate the temporary stego pixels values as (36, 30, 19, 18). After that, SD_2 is convert to binary secret data (10010011), and we use (30) for stego pixel values. Finally, stego pixel group is obtained as (146, 121, 76, 75).

Extracting process:

$$(p_1^l, p_2^l, p_3^l, p_4^l) = (146, 121, 76, 75) n=4, k=2,$$

Firstly, we compute M_n value using (18), $M_n = 23$. Then we use (31) to extracting k -bit secret data from each pixel value. We obtain SD_2 secret data values (2, 1, 0, 3), (10, 01, 00, 11)₂, $SD_2 = 147$. Now, let's extract the other data (SD_1). Using (32), we obtain the temporary stego pixels values (36, 30, 19, 18). After, extracting (SD_1) using (33), $SD_1 = 316 \text{ mod } 23 = 17$.

Example-2

Embedding process:

$$(p_1, p_2, p_3, p_4) = (3, 167, 146, 255) n=4, k=2 SD_1 = 17, SD_2 = 147$$

Firstly, Compute the M_n value using (18), n is even, so $U_n = (2 \times 4^2 - 2)/3, U_n = 10, M_n = 2 \times 10 + 3, M_n = 23$. Then, we calculate $(p_1^{temp}, p_2^{temp}, p_3^{temp}, p_4^{temp}) = (0, 41, 36, 63)$ using (25). We have FOBP according to (26), so updated pixel values as (1, 41, 36, 62). Calculate $g_{EMSD}^{temp} = 10$ using (27). $diff = 17 - 10 = 7$ in $\{(-M_n/2) \leq diff \leq (+M_n/2)\}$ $diff$ (1001) is represented in EMSD

algorithm using modified weight minimization algorithm. We calculate the temporary stego pixels values as (0, 41, 36, 63). After that, SD_2 is convert to binary secret data (10010011), and we use (30) for stego pixel values. Finally, stego pixel group is obtained as (2,165, 144, 255).

Extracting process:

$$(p'_1, p'_2, p'_3, p'_4) = (2, 165, 144, 255), n=4, k=2$$

Firstly, we compute M_n value using (18), $M_n = 23$. Then we use (31) to extracting k-bit secret data from each pixel value. We obtain SD_2 secret data values (2, 1, 0, 3), $(10,01,00,11)_2$, $SD_2 = 147$. Now, let's extract the other data (SD_1). Using (32), we obtain the temporary stego pixels values (0, 41, 36, 63). After, extracting (SD_1) using (33), $SD_1 = 730 \text{ mod } 23 = 17$.

IV. EXPERIMENTAL RESULTS AND COMPARISONS

We present comparisons and experimental results to evaluate the proposed algorithm in this section. All experiments results are performed by Matlab R2015b in a desktop computer with an Intel(R) Core(TM) i5-7400 CPU @ 3.0 GHz, 4 GB RAM and Windows 10 Professional 64-bit operating system. The proposed algorithm is tested on a series of standard grayscale cover images to evaluate by the data hiding measurement metrics such as embedding capacity (EC), payload (P), peak signal to noise ratio (PSNR), and structural similarity index measure (SSIM). Figure 3 shows eight 512×512 grayscale cover images, which are used to hide secret data with various embedding capacity in the experiment. Also, the secret data, generated randomly by the computer, are hidden inside these cover images in experimental studies.

The PSNR value, which is one of the commonly used evaluation metrics in image steganography studies, is used to evaluate stego-image quality. The high PSNR value depicts better stego image quality. Generally, when the PSNR value of the stego-image is higher than 30 dB, the cover and stego images are considered visually indistinguishable. Formulas showing the calculation of the mean square error (MSE) and PSNR values are given in (34) and (35). M and N values represent the size of the images in (34) [10], [38].

$$MSE = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N (\text{Cover}(i, j) - \text{Stego}(i, j))^2 \tag{34}$$

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE} \tag{35}$$

The SSIM value is another metric used to show the similarity ratio between cover and stego image. SSIM values are between 0 and 1, which is close to 1 indicates that the original and result images are akin. The SSIM value is computed as presented in (36) [10].

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{36}$$

The EC is described as the total secret bits that is hidden into the cover-image, while, the P value in (37) is symbolized

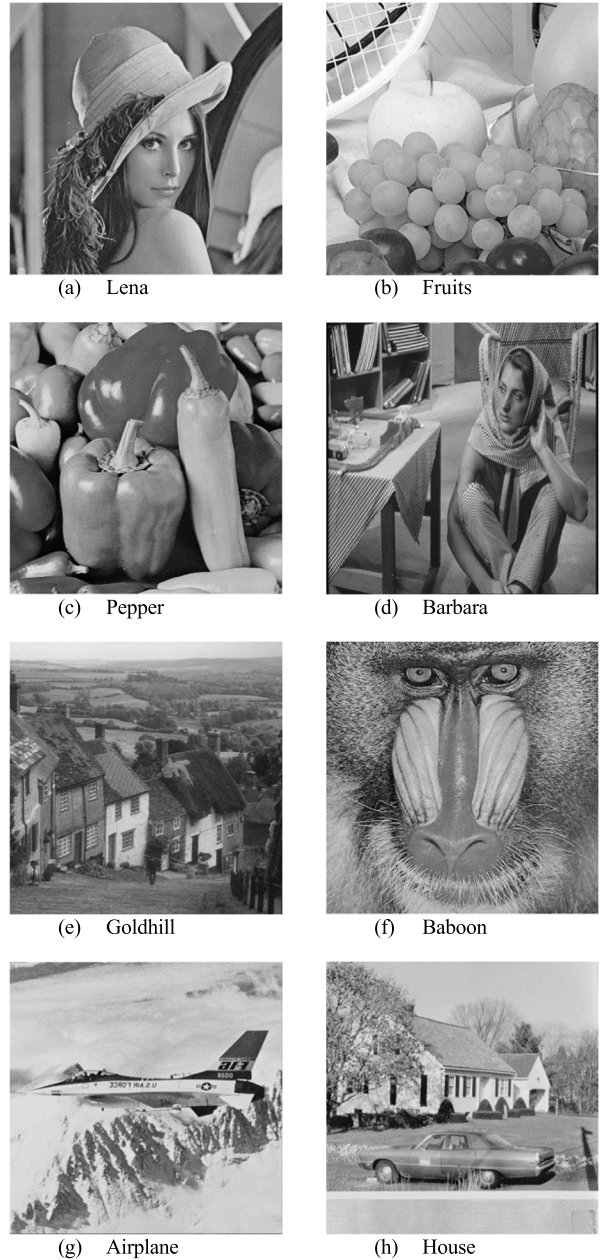


FIGURE 3. Eight grayscale cover images (512 x 512).

bit per byte (bpb), the secret bits number that is embedded in a gray image pixel.

$$P = \frac{EC}{M \times N} \tag{37}$$

The PSNR, SSIM, EC and P results are analyzed according to different k and n values in experimental studies. The performance of the proposed algorithm is evaluated according to maximum embedding capacities with the parameters k (k=1, 2, 3) and n (n=2, 3, 4, 5, 6, 7). If the k value is more than 3, the PSNR value of the stego image is lower than 30 dB.

The PSNR, SSIM, EC and P results are analyzed according to different k and n values in experimental studies. The performance of the proposed algorithm is evaluated according to

TABLE 1. Comparisons of SSIM and PSNRs ($K=1$, between $N=2$ AND $N=7$).

	n=2		n=3		n=4		n=5		n=6		n=7	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	43.47	0.982	43.11	0.982	43.07	0.983	42.93	0.982	42.87	0.983	42.86	0.982
Fruits	43.48	0.981	43.16	0.980	43.11	0.981	42.94	0.980	42.89	0.981	42.91	0.981
Pepper	43.45	0.983	43.15	0.983	43.09	0.984	42.92	0.983	42.89	0.984	42.89	0.983
Barbara	43.44	0.988	43.11	0.988	43.09	0.989	42.92	0.988	42.86	0.989	42.86	0.988
Goldhill	43.47	0.986	43.10	0.986	43.04	0.987	42.95	0.986	42.86	0.987	42.86	0.986
Baboon	43.47	0.994	43.12	0.994	43.05	0.995	42.95	0.994	42.89	0.995	42.86	0.994
Airplane	43.47	0.981	43.11	0.980	43.06	0.981	42.95	0.981	42.84	0.982	42.87	0.981
House	43.46	0.986	43.14	0.985	43.08	0.986	42.98	0.986	42.87	0.986	42.87	0.986
Average	43.46	0.985	43.13	0.985	43.07	0.986	42.87	0.985	42.87	0.986	42.87	0.985

TABLE 2. Comparisons of SSIM and PSNRs ($K=2$, between $N=2$ and $N=7$).

	n=2		n=3		n=4		n=5		n=6		n=7	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	37.27	0.930	36.92	0.929	36.88	0.933	36.78	0.930	36.70	0.933	36.65	0.930
Fruits	37.28	0.926	36.92	0.923	36.90	0.927	36.80	0.924	36.73	0.927	36.69	0.925
Pepper	37.27	0.934	36.96	0.933	36.87	0.937	36.79	0.934	36.68	0.937	36.69	0.934
Barbara	37.29	0.955	36.95	0.954	36.90	0.956	36.75	0.954	36.69	0.956	36.68	0.955
Goldhill	37.28	0.945	36.93	0.944	36.87	0.947	36.77	0.945	36.63	0.948	36.68	0.945
Baboon	37.26	0.978	36.95	0.976	36.88	0.978	36.80	0.977	36.69	0.978	36.69	0.977
Airplane	37.30	0.928	36.96	0.924	36.92	0.929	36.74	0.926	36.67	0.928	36.65	0.926
House	37.30	0.947	36.99	0.944	36.96	0.947	36.80	0.945	36.71	0.947	36.74	0.946
Average	37.28	0.943	36.95	0.941	36.90	0.944	36.78	0.942	36.69	0.944	36.68	0.942

TABLE 3. Comparisons of SSIM and PSNRs ($K=3$, between $N=2$ and $N=7$).

	n=2		n=3		n=4		n=5		n=6		n=7	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Lena	31.29	0.788	30.89	0.783	30.82	0.788	30.69	0.783	30.61	0.788	30.64	0.784
Fruits	31.24	0.777	30.95	0.769	30.90	0.778	30.80	0.769	30.67	0.777	30.74	0.771
Pepper	31.22	0.794	30.87	0.788	30.83	0.797	30.73	0.790	30.66	0.796	30.58	0.791
Barbara	31.13	0.857	30.81	0.853	30.84	0.859	30.67	0.854	30.65	0.860	30.56	0.855
Goldhill	31.25	0.828	30.95	0.823	30.83	0.827	30.72	0.822	30.54	0.829	30.61	0.823
Baboon	31.21	0.918	30.85	0.916	30.82	0.921	30.71	0.918	30.63	0.921	30.60	0.918
Airplane	31.11	0.785	30.82	0.777	30.88	0.781	30.68	0.774	30.64	0.781	30.68	0.776
House	31.17	0.832	30.85	0.826	30.88	0.834	30.71	0.830	30.58	0.836	30.73	0.827
Average	31.20	0.822	30.87	0.817	30.85	0.823	30.71	0.818	30.62	0.824	30.64	0.818

maximum embedding capacities with the parameters k ($k=1, 2, 3$) and n ($n=2, 3, 4, 5, 6, 7$). If the k value is more than 3, the PSNR value of the stego image is lower than 30 dB.

To appreciate the achievement of our algorithm in different cover images, Tables 1 to 3 compares according to SSIM and PSNR values under $k = 1$ to 3, and $n = 2$ to 7, respectively. Even if a lot of secret bits is hidden in the cover-image, the stego-image quality appears to be acceptable. In Table 1 presents PSNR value of the proposed algorithm provides about 43 dB when $k = 1$ and n between 2 and 7, which indicates that the stego-image quality is good. Also, the SSIM mean value is 0.985, it means that the cover and stego images are highly similar.

According to Table 2, PSNR and SSIM mean values are calculated as 36.88 and 0.984, respectively, when $k = 2$ and n between 2 and 7. According to Table 2, the stego image has little distortion and the human eye cannot perceive it, also the two images are visually indistinguishable.

In Table 3, PSNR value of the proposed algorithm provides higher than 30 dB when $k = 3$ and n between 2 and 7,

which shows that the stego image quality is acceptable. When Table 3 is examined, both high data embedding capacity and higher PSNR value (31.20) are obtained for $n = 2$ compared to other n values. The SSIM average value for all cover images and n values is obtained about 0.82, which emphasizes that the original and result images are highly similar.

Table 4 indicates comparisons of payload for different data hiding algorithm. The proposed algorithm can be used to hide data securely at high capacity in cover images. Table 4 shows that it has higher capacity when compared with EMD [34], GEMD [35], SMSD [36], enhanced GEMD [37], and EMSD [38]. However, in the CRT-EMD study [18], n values (except $n = 2$ and 3) are not suitable because of the poor stego image quality and lower PSNR value (<30 dB). The proposed algorithm offers high payload for $k=1$, 2.107 - 2.404 (bpb), for $k = 2$, 3.107 - 3.404 (bpb), and for $k=3$, 4.107 - 4.404 (bpb), respectively. In this paper shows that the proposed algorithm for $k = 1$ value is utilized if better stego image quality is needed and for $k = 2$ or $k = 3$ will be employed if higher embedding capacity is required.

TABLE 4. Comparisons of payload (P) for different data hiding algorithm.

	EMD [34]	GEMD [35]	SMSD [36]	Enhanced GEMD [37]	CRT-EMD [23]	EMSD [38]	Proposed algorithm		
	$\log_2(2n + 1)$	$n + 1$	$\log_2(Tn)$	$n + 2$	$n + 3$	$\log_2(Mn)$	k=1	k=2	k=3
	n	n	n	n	2	n	n x k + $\log_2(Mn)$		
n = 2	1.161	1.500	1.161	2.000	2.500	1.404	2.404	3.404	4.404
n = 3	0.936	1.333	1.153	1.667	3.000	1.302	2.302	3.302	4.302
n = 4	0.793	1.250	1.098	1.500	-	1.131	2.131	3.131	4.131
n = 5	0.692	1.200	1.085	1.400	-	1.156	2.156	3.156	4.156
n = 6	0.617	1.167	1.068	1.333	-	1.074	2.074	3.074	4.074
n = 7	0.558	1.143	1.060	1.286	-	1.107	2.107	3.107	4.107

TABLE 5. Comparisons of embedding capacity (EC) for different data hiding algorithm.

	EMD [34]	GEMD [35]	SMSD [36]	Enhanced GEMD [37]	CRT-EMD [23]	EMSD [38]	Proposed algorithm		
							k=1	k=2	k=3
n = 2	304349	393216	304349	524288	655360	368050	630194	892338	1154482
n = 3	245367	349438	302252	436994	786432	341311	603455	865599	1127743
n = 4	207880	327680	287834	393216	-	296485	558629	820773	1082917
n = 5	181404	314573	284426	367002	-	303038	565182	827326	1089470
n = 6	161743	305922	279970	349438	-	281543	543687	805831	1067975
n = 7	146276	299631	277873	337117	-	290193	552337	814481	1076625

TABLE 6. Comparisons of embedding capacity (EC) and PSNR values for different data hiding algorithm.

	SMSD [36]		CRT-EMD [23]		EMSD [38]		Proposed algorithm					
							k=1		k=2		k=3	
	EC	PSNR	EC	PSNR	EC	PSNR	EC	PSNR	EC	PSNR	EC	PSNR
n = 2	304349	52.11	655360	41.60	368050	51.80	630194	43.46	892338	37.28	1154482	31.20
n = 3	302252	51.86	786432	32.50	341311	51.62	603455	43.13	865599	36.95	1127743	30.87
n = 4	287834	52.32	-	-	296485	52.21	558629	43.07	820773	36.90	1082917	30.85
n = 5	284426	52.32	-	-	303038	51.82	565182	42.94	827326	36.78	1089470	30.71
n = 6	279970	52.33	-	-	281543	52.30	543687	42.87	805831	36.69	1067975	30.62
n = 7	277873	52.33	-	-	290193	52.26	552337	42.87	814481	36.68	1076625	30.64

Table 5 presents comparisons of embedding capacity between the proposed algorithm and different data hiding algorithm. Embedding capacity (EC) offers the total amount of secret bits hidden in a 512 x 512 grayscale cover image in table 5. The proposed algorithm has a higher embedding capacity compared to other algorithms when k = 2 and k = 3. The CRT-EMD algorithm embedding capacity is better for k=1, but at other k values, the proposed algorithm has higher embedding capacity. It has more than twice embedding capacity EMD, GEMD, SMSD, and EMSD algorithms. Especially, the embedding capacity of the proposed algorithm is about 262,144 to 786,432 bits larger than that of EMSD algorithm which bases on the proposed algorithm.

Figure 4 presents the relationship between the number of pixels and the embedding capacity for different algorithms, including EMD [34], GEMD [35], SMSD [36], Enhanced GEMD [37], CRT-EMD [23], EMSD [38], and the proposed algorithm. When the pixel number value (n) is increased for

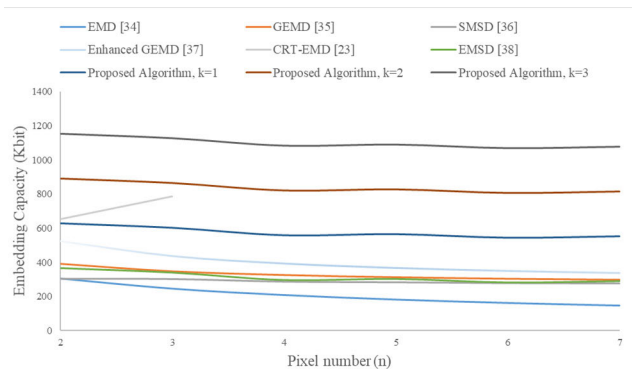


FIGURE 4. Comparisons of embedding capacity (EC) for different data hiding algorithm.

all algorithms (except CRT-EMD), the maximum embedding capacity decreases. In this study, the maximum embedding capacity is given as Kbit in figure 4. It is seen in figure 4 that

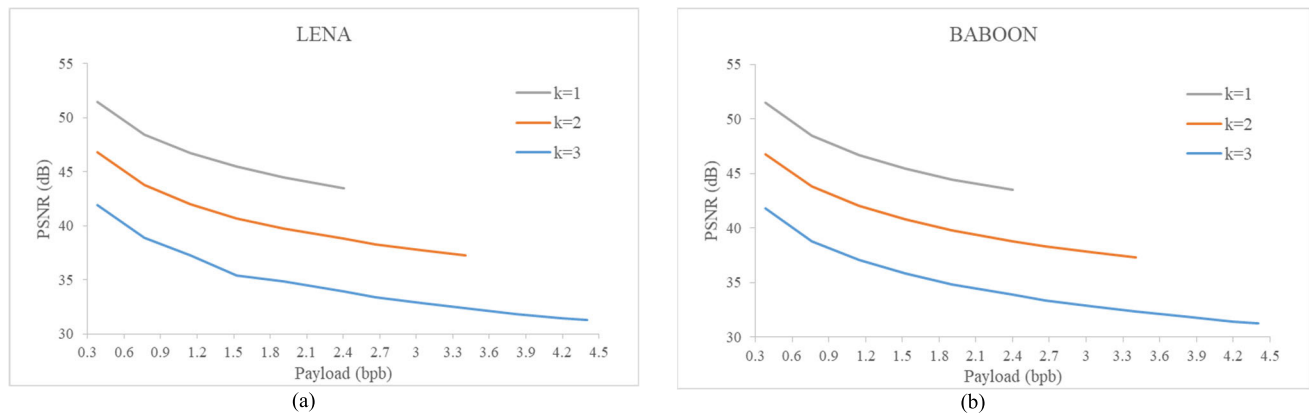


FIGURE 5. Comparisons of PSNR for different k values, (a) Lena, (b) Baboon.

the algorithms used for comparison in experimental studies have embedding capacity between 150 Kbit and 500 Kbit. Also, it is seen that the maximum embedding capacity varies between 600 Kbit and 1200 Kbit with the proposed algorithm.

Figure 5 presents the relationship between payload and PSNR for different ' k ' values of the proposed algorithm. In the study, the payload between 0.3 and 4.5 (bbp) is embedded in cover images and PSNR values are obtained. All cover images used in experimental studies give similar PSNR values at the same payload, so the graphics of Lena and Baboon cover images are presented in Figure 5. When the graph is examined, it is seen that PSNR values are acceptable, even if the payload is increased.

Table 6 shows that the comparisons of embedding capacity and PSNR values between different data hiding algorithms and the proposed algorithm. SMSD and EMSD algorithms appear to have higher PSNR values according to the proposed algorithm but these algorithms have a much lower embedding capacity. Although approximately twice as much data is embedded secret data compared to SMSD and EMSD algorithms, it is achieved a PSNR value above 43 dB by the proposed algorithm. In addition, the proposed algorithm has both more data hiding capacity and higher PSNR values than the CRT-EMD algorithm.

V. CONCLUSION

High embedding capacity and acceptable visual image quality are the most basic features of image steganography. In this article, a new hybrid data hiding technique based on least significant bit (LSB) substitution and enhanced modified signed digit (EMSD) algorithms is proposed to embed secret data. The purpose of our algorithm is to achieve high embedding capacity when acceptable visual stego-image quality. The experimental results indicate that PSNR values above 43dB while embedding secret data of approximately 630Kbit, PSNR value above 37dB while embedding secret data of 900Kbit, and PSNR value over 31dB while hiding 1150Kbit secret data. Especially, the embedding capacity of the proposed algorithm is about 262,144 to 786,432 bits larger

than similar algorithms as the EMSD. Also, the hybrid use of the EMSD and LSB substitution algorithms provides more security for secret data, and get suitable stego-image quality.

REFERENCES

- [1] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. IT-22, no. 6, pp. 644–654, Nov. 1976.
- [2] P. Patil, P. Narayankar, N. D. G., and M. S. M., "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and blowfish," *Procedia Comput. Sci.*, vol. 78, pp. 617–624, Jan. 2016.
- [3] M. Z. Konyar and S. Öztürk, "Reed Solomon coding-based medical image data hiding method against salt and pepper noise," *Symmetry*, vol. 12, no. 6, p. 899, Jun. 2020.
- [4] T. Aydoğan and C. Bayılmış, "A new efficient block matching data hiding method based on scanning order selection in medical images," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 25, no. 1, pp. 461–473, 2017.
- [5] R. Das and I. Das, "Secure data transfer in IoT environment: Adopting both cryptography and steganography techniques," in *Proc. 2nd Int. Conf. Res. Comput. Intell. Commun. Netw. (ICRCICN)*, Sep. 2016, pp. 296–301.
- [6] S. Arunkumar, S. Vairavasundaram, K. S. Ravichandran, and L. Ravi, "RIWT and QR factorization based hybrid robust image steganography using block selection algorithm for IoT devices," *J. Intell. Fuzzy Syst.*, vol. 36, no. 5, pp. 4265–4276, May 2019.
- [7] A. K. Sahu, G. Swain, and E. S. Babu, "Digital image steganography using bit flipping," *Cybern. Inf. Technol.*, vol. 18, no. 1, pp. 69–80, Mar. 2018.
- [8] M. Hussain, A. W. A. Wahab, Y. I. B. Idris, A. T. Ho, and K. H. Jung, "Image steganography in spatial domain: A survey," *Signal Process., Image Commun.*, vol. 65, pp. 46–66, Jul. 2018.
- [9] M. Z. Konyar, O. Akbulut, and S. Öztürk, "Matrix encoding-based high-capacity and high-fidelity reversible data hiding in HEVC," *Signal, Image Video Process.*, vol. 14, no. 5, pp. 897–905, Jul. 2020.
- [10] S. Solak and U. Altınışık, "Image steganography based on LSB substitution and encryption method: Adaptive LSB+3," *J. Electron. Imag.*, vol. 28, no. 4, Aug. 2019, Art. no. 043025.
- [11] S. Solak and U. Altınışık, "A new approach for Steganography: Bit shifting operation of encrypted data in LSB (SED-LSB)," *Bilişim Teknolojileri Dergisi*, vol. 12, no. 1, pp. 75–81, 2019.
- [12] S. M. Masud Karim, M. S. Rahman, and M. I. Hossain, "A new approach for LSB based image steganography using secret key," in *Proc. 14th Int. Conf. Comput. Inf. Technol. (ICCIT)*, Dec. 2011, pp. 286–291.
- [13] W. Luo, F. Huang, and J. Huang, "Edge adaptive image steganography based on LSB matching revisited," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 201–214, Jun. 2010.
- [14] A. K. Sahu and G. Swain, "A novel n-Rightmost bit replacement image steganography technique," *3D Res.*, vol. 10, no. 1, Mar. 2019.
- [15] S. Prasad and A. K. Pal, "An RGB colour image steganography scheme using overlapping block-based pixel-value differencing," *Roy. Soc. Open Sci.*, vol. 4, no. 4, Apr. 2017, Art. no. 161066.

- [16] P.-H. Kim, E.-J. Yoon, K.-W. Ryu, and K.-H. Jung, "Data-hiding scheme using multidirectional pixel-value differencing on colour images," *Secur. Commun. Netw.*, vol. 2019, pp. 1–11, Oct. 2019.
- [17] Z. Li and Y. He, "Steganography with pixel-value differencing and modulus function based on PSO," *J. Inf. Secur. Appl.*, vol. 43, pp. 47–52, Dec. 2018.
- [18] C. H. Yang, C. Y. Weng, H. K. Tso, and S. J. Wang, "A data hiding scheme using the varieties of pixel-value differencing in multimedia images," *J. Syst. Softw.*, vol. 84, no. 4, pp. 669–678, 2011.
- [19] D.-C. Wu and W.-H. Tsai, "A steganographic method for images by pixel-value differencing," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1613–1626, Jun. 2003.
- [20] C.-F. Lee, C.-C. Chang, and K.-H. Wang, "An improvement of EMD embedding method for large payloads by pixel segmentation strategy," *Image Vis. Comput.*, vol. 26, no. 12, pp. 1670–1676, Dec. 2008.
- [21] Y.-X. Liu, C.-N. Yang, Y.-S. Chou, S.-Y. Wu, and Q.-D. Sun, "Progressive (k,n) secret image sharing scheme with meaningful shadow images by GEMD and RGEMD," *J. Vis. Commun. Image Represent.*, vol. 55, pp. 766–777, Aug. 2018.
- [22] C.-C. Wang, W.-C. Kuo, Y.-C. Huang, and L.-C. Wu, "A high capacity data hiding scheme based on re-adjusted GEMD," *Multimedia Tools Appl.*, vol. 77, no. 5, pp. 6327–6341, Mar. 2018.
- [23] Y. Liu, C. C. Chang, P. C. Huang, and C. Y. Hsu, "Efficient information hiding based on theory of numbers," *Symmetry*, vol. 10, no. 19, pp. 1–17, 2018.
- [24] H.-S. Leng and H.-W. Tseng, "Generalize the EMD scheme on an n-dimensional hypercube with maximum payload," *Multimedia Tools Appl.*, vol. 78, no. 13, pp. 18363–18377, Jul. 2019.
- [25] K. A. Darabkh, A. K. Al-Dhamari, and I. F. Jafar, "A new steganographic algorithm based on multi directional PVD and modified LSB," *Inf. Technol. Control*, vol. 46, no. 1, pp. 16–36, Apr. 2017.
- [26] G. Swain, "Very high capacity image steganography technique using quotient value differencing and LSB substitution," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 2995–3004, Apr. 2019.
- [27] M. He, Y. Liu, C.-C. Chang, and M. He, "A mini-sudoku matrix-based data embedding scheme with high payload," *IEEE Access*, vol. 7, pp. 141414–141425, 2019.
- [28] W. C. Kuo and S. Y. Chang, "Hybrid GEMD data hiding," *J. Inf. Hiding Multimedia Signal Process.*, vol. 5, no. 3, pp. 420–430, 2014.
- [29] A. Pradhan, K. R. Sekhar, and G. Swain, "Digital image steganography using LSB substitution, PVD, and EMD," *Math. Problems Eng.*, vol. 2018, pp. 1–11, Sep. 2018.
- [30] A. K. Sahu and G. Swain, "High fidelity based reversible data hiding using modified LSB matching and pixel difference," *J. King Saud Univ.-Comput. Inf. Sci.*, to be published, doi: [10.1016/j.jksuci.2019.07.004](https://doi.org/10.1016/j.jksuci.2019.07.004).
- [31] G. Swain and A. K. Sahu, "Data hiding using adaptive LSB and PVD technique resisting PDH and RS analysis," *Int. J. Electron. Secur. Digit. Forensics*, vol. 11, no. 4, p. 458, 2019.
- [32] A. K. Sahu and G. Swain, "Pixel overlapping image steganography using PVD and modulus function," *3D Res.*, vol. 9, no. 3, p. 40, Sep. 2018.
- [33] A. K. Sahu and G. Swain, "An optimal information hiding approach based on pixel value differencing and modulus function," *Wireless Pers. Commun.*, vol. 108, no. 1, pp. 159–174, Sep. 2019.
- [34] X. Zhang and S. Wang, "Efficient steganographic embedding by exploiting modification direction," *IEEE Commun. Lett.*, vol. 10, no. 11, pp. 781–783, Nov. 2006.
- [35] W.-C. Kuo and C.-C. Wang, "Data hiding based on generalised exploiting modification direction method," *Imag. Sci. J.*, vol. 61, no. 6, pp. 484–490, Jul. 2013.
- [36] W.-C. Kuo, C.-C. Wang, and H.-C. Hou, "Signed digit data hiding scheme," *Inf. Process. Lett.*, vol. 116, no. 2, pp. 183–191, Feb. 2016.
- [37] Y. Liu, C. Yang, and Q. Sun, "Enhance embedding capacity of generalized exploiting modification directions in data hiding," *IEEE Access*, vol. 6, pp. 5374–5378, 2018.
- [38] Y.-X. Liu, C.-N. Yang, Q.-D. Sun, S.-Y. Wu, S.-S. Lin, and Y.-S. Chou, "Enhanced embedding capacity for the SMSD-based data-hiding method," *Signal Process., Image Commun.*, vol. 78, pp. 216–222, Oct. 2019.



SERDAR SOLAK received the B.S. and M.S. degrees from the Department of Computer Engineering, Kocaeli University, Turkey, in 2002 and 2008, respectively, and the Ph.D. degree in electronics-computer education from Kocaeli University, in 2016. He is currently an Assistant Professor with the Department of Information System Engineering, Kocaeli University. His research interests include mobile robots, computer vision, embedded systems, image-steganography, data-hiding, and distance learning.

...