Journal of Information Hiding and Multimedia Signal Processing ©2014 ISSN 2073-4212
Ubiquitous International Volume **5**, Number **2**, April **2014**

High-capacity Steganographic Method based on Division Arithmetic and Generalized Exploiting Modification Direction

Wen-Chung Kuo
Department of Computer Science and Information Engineering
National Yunlin University of Science & Technology
simonkuo@yuntech.edu.tw
The corresponding author

Yi-Heng Chen and Chen-Tsun Chuang
Department of Computer Science and Information Engineering
National Formosa University

Received February, 2013; revised October, 2013

Abstract. *Data hiding by using LSB replacement is a common and straightforward embedding method. However, the attacker can reveal the secret easily by analyzing the bitplane attack. In this paper, we will propose a simple and high-capacity steganographic method based on division arithmetic and generalized exploiting modification direction method which is not susceptible to bitplane analysis. The experimental results show that our proposed method not only maintains the advantages of the LSB replacement technique but also enhances security of the secret data.*
Keyword: Steganography, steganalysis, data hiding, LSB replacement

1. **Introduction.** From the rapid growth of computer and internet technology, digital multimedia content such as images, audio, and video are distributed faster than before. How to prevent digital content from being intercepted by unauthorized parties is a very interesting topic for information security. In general, there are two common methodologies for data security: cryptography and steganography. Specifically, steganography hides personal data behind a meaningful image so an unintended observer will not be aware of the existence of the hidden secret message. Previously, many data hiding schemes based on different embedding methods (such as direct embedding or indirect embedding) have been proposed [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14].

For direct embedding, the most common data hiding technique is the least significant bit replacement method (LSB). This scheme is very simple, fast and has good stego image quality, but it is not secure by using the bit-plant attack. Alternatively, indirect embedding employs an embedding function, such as the data hiding method based on the Exploiting Modification Direction (EMD) proposed by Zhang and Wang [14] in 2006. Many EMD-type schemes [6, 7, 9] were proposed to improve the embedding capacity or enhance embedded data security. However, the embedding capacity is at most 1.5 bpp (bits per pixel) for these EMD schemes.

In order to enhance both the embedding capacity and confidentiality, a hybrid LSB block data hiding scheme based on division arithmetic (DA) and General Exploiting

Modification Direction (GEMD) [7] method will be proposed in this paper. According to experiment results, our proposed scheme keeps the advantages of LSB substitution and enhances the embedding capacity but also prevents disclosure from bitplane attack [13].

The rest of paper is organized as follows: In Section 2, the LSB and GEMD data hiding schemes are reviewed. In Section 3, a new data hiding method based on DA and GEMD (DA-GEMD) method is described. Experimental results and secure analysis are provided in Section 4. Finally, the conclusion is given in Section 5.

2. **Data hiding scheme review.** In this section, we will review two data hiding schemes from two different embedding approaches. For direct embedding, the least significant bit replacement method is detailed For indirect embedding using the extraction function, the GEMD data hiding scheme will be described.

2.1. **LSB data hiding.** In the LSB data hiding scheme, we convert the secret image into binary bit-stream form and also convert the pixel values of the cover image from decimal to binary. Then, we replace the k-rightmost bits of each pixel sequentially with the binary data of the secret stream.

Algorithm LSB (Embedding Algorithm for LSB replacement method)
Input: cover image I_C and binary secret data stream M
Output: stego image I_S

(LSB-1): For each pixel i, embedded k binary bits and calculate the secret $m_i = \sum_{j=0}^{k-1} m_j \times 2^{k-1-j}$, where $m_j \in \{0, 1\}$..

(LSB-2): Compute $y_i = x_i - (x_i \bmod 2^k) + m_i$ where y_i is the i_{th} stego pixel of I_S.

Example 2.1. *Given four pixels* (202, 197, 196, 203) *and secret data as* $M =$ [1, 1, 0, 1, 0, 0, 1, 0], *embedded two bits for each pixel to get the stego pixels* (203, 197, 196, 202) *from following steps:*

(Step 1): Calculate $M^* = [m_1, m_2, m_3, m_4] = [3, 1, 0, 2]$.

(Step 2): By using the LSB algorithm, we can get the stego pixels (203, 197, 196, 202).

The LSB replacement hiding technique is simple and fast, and has good imperceptibility (PSNR) and good capacity. However, attackers can determine the secret easily by analyzing the bitplane attack. For example, the data stream easily obtained from the 1st bitplane. The 1st bitplane is composed of the rightmost LSB of the pixels in the stego image.

2.2. **The data hiding based on GEMD.** Recently, Kuo and Wang [7] proposed a GEMD data hiding scheme to improve the embedding capacity of EMD method [14]. According to Kuo-Wang scheme, there is a new extraction function proposed, shown as Eq.(1).

$$f_g(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{n} x_i \times (2^i - 1) \bmod 2^{n+1}, \quad (1)$$

where x_i is the i_{th} pixel and n is the pixel number. In particular, Lee *etal.*'s scheme [9] is the special case of [7].

The function $O_{GEMD}(\cdot)$ can obtain all n-tuples $(x_1, x_2, \ldots, x_n)$ from partitioning the image I_C into non-overlapping n-pixel blocks by scanning from left to right per row and then top down, as shown in Fig.1. $O_{GEMD-S}(\cdot)$ can obtain (2^{n+1})-ary data m from partitioning the secret data stream M for each block.

Algorithm GEMD (Embedding Algorithm for Kuo-Wang Scheme)

Input: cover image I_C and binary secret data stream M
Output: stego image I_S

(GEMD-1): Obtain all n-pixel blocks $(x_1, x_2, \ldots, x_n)$ from I_C and $O_{GEMD}(I_C)$ and secret data m from $O_{GEMD-S}(M)$.
(GEMD-2): For each block, calculate $t = f_g(x_1, x_2, \ldots, x_n)$.
(GEMD-3): Calculate the difference $D_g = m - t$.
(GEMD-4): If $D_g \leq 2^n$, then $D'_g = D_g$ and go to (GEMD-5), else let $D'_g = 2^{n+1} - D_g$ and go to (GEMD-6).
(GEMD-5): If $D'_g = 2^n$, then $x'_n = x_n + 1$, $x'_1 = x_1 + 1$;
else transform D'_g to $(V_n V_{n-1} \ldots V_1 V_0)_2$ and
For $i = n$ to 1 do
{if ($V_i = 0$ & $V_{i-1} = 0$) or ($V_1 = 1$ & $V_{i-1} = 1$) then $x'_i = x_i$;
else if ($V_i = 0$ & $V_{i-1} = 1$) then $x'_i = x_i + 1$;
else if ($V_i = 1$ & $V_{i-1} = 0$) then $x'_i = x_i - 1$.}
Go to (GEMD-7).
(GEMD-6): Transform D'_g to $(V_n V_{n-1} \ldots V_1 V_0)_2$;
For $i = n$ to 1 do
{If ($V_i = 0$ & $V_{i-1} = 0$) or ($V_1 = 1$ & $V_{i-1} = 1$) then $x'_i = p_i$;
else if ($V_i = 0$ & $V_{i-1} = 1$) then $x'_i = x_i - 1$;
else if ($V_i = 1$ & $V_{i-1} = 0$) then $x'_i = x_i + 1$.}

(GEMD-7): End.

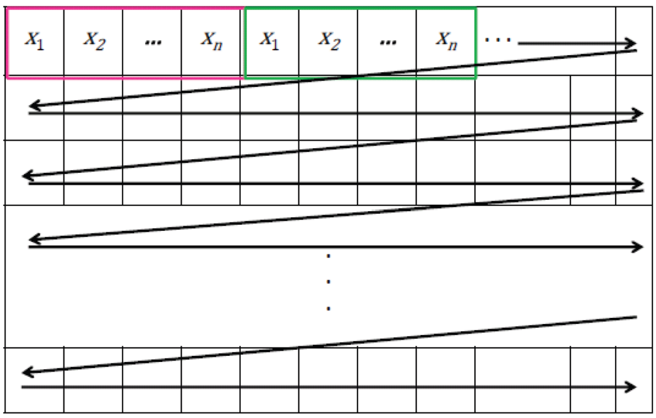


FIGURE 1. The embedding data sequence by GEMD

Example 2.2. *Given three cover image pixels* $(x_1, x_2, x_3) = (152, 155, 157)$ *and secret data* $m_2 = (0101)_2 = 5$, *we obtain output stego image pixels* $(x'_1, x'_2, x'_3) = (153, 155, 157)$ *using the GEMD embedding procedure*

The embedding steps are shown as following:

(Step 1): Compute $t = f_g(x_1, x_2, x_3) = f_g(152, 155, 157) = (1 \times 152 + 3 \times 155 + 7 \times 157) mod 16 = 4$.
(Step 2): Compute $D = (m_2 - t) \bmod 16 = (5 - 4) \text{mod} 16 = 1 = (0001)_2$.
(Step 3): By using the GEMD algorithm, we can compute $x'_1 = 152 + 1 = 153$, $x'_2 = 155 - 0 = 155, x'_3 = 157 + 0 = 157$.

So, the secret data is recovered by using $m_2 = f(x'_1, x'_2, x'_3) = f(153, 155, 157) = 5 = (0101)_2$.

3. **Proposed data hiding scheme.** For raising entropy and maximum capacity, we will propose a data hiding scheme based on DA-GEMD. The proposed hiding framework of DA-GEMD embedding process is shown as Fig.3.

Some notations are defined to assist introduction of the DA-GEMD-scheme:

I_C**:** Grayscale cover image.

$O_{DA-GEMD}(\cdot)$**:** Obtains all 9-tuples $(x_1, x_2, \ldots, x_9)$ from partitioning the image I_C into non-overlapping 3×3 pixels for each block by scanning from left to right side and from top to bottom, as shown in Fig.2.

$O_{DA-GEMD-S}(\cdot)$**:** Obtains n-bit binary data m from partitioning the secret data stream M for each block.

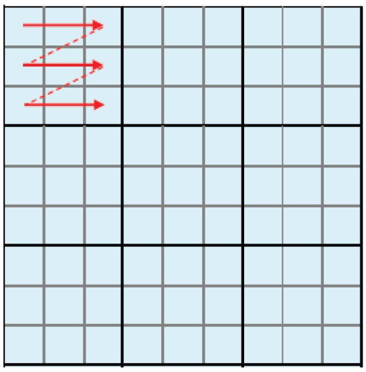


FIGURE 2. The raster scan in each block

Algorithm DA-GEMD (Embedding Algorithm for DA-GEMD Scheme)

Input: cover image I_C and binary secret data stream M

Output: stego image I_S

(DA-GEMD-1): Obtain all 3×3-pixels for each block $(x_1, x_2, \ldots, x_9)$ from I_C and $O_{DA-GEMD}(I_C)$.

(DA-GEMD-2): For each block,

1. Calculate the quotient set Q and residue set RR:

$$Q = \{q_i = [x_i/4], for \quad i = 1, 2, \cdots, 9\} \tag{2}$$

$$RR = \{r_i = x_i mod 4, for \quad i = 1, 2, \cdots, 9\}. \tag{3}$$

2. Find the median of Q as the unique EP_I.
3. Calculate the difference d_i between q_i and EP_I.
4. According to following two cases, we can embed the n-bit secret message into x_i for $for \quad i = 1, 2, \cdots, 9$.

Case A: $d_i = 0$,

Using 2-LSB to replace two least-significant bits of ei with binary secret bits.

Case B: $d_i \neq 0$,

if $r_i = 0$, then $r'_i = r_i + 1$,

else if $r_i = 3$, then $r'_i = r_i - 1$, else $r'_i = r_i$.

Let $x_i^\star = q_i \times 4 + r'_i$ and embed the secret bit by using GEMD-algorithm.

Below is a simple example to explain the proposed embedding procedure in this section.

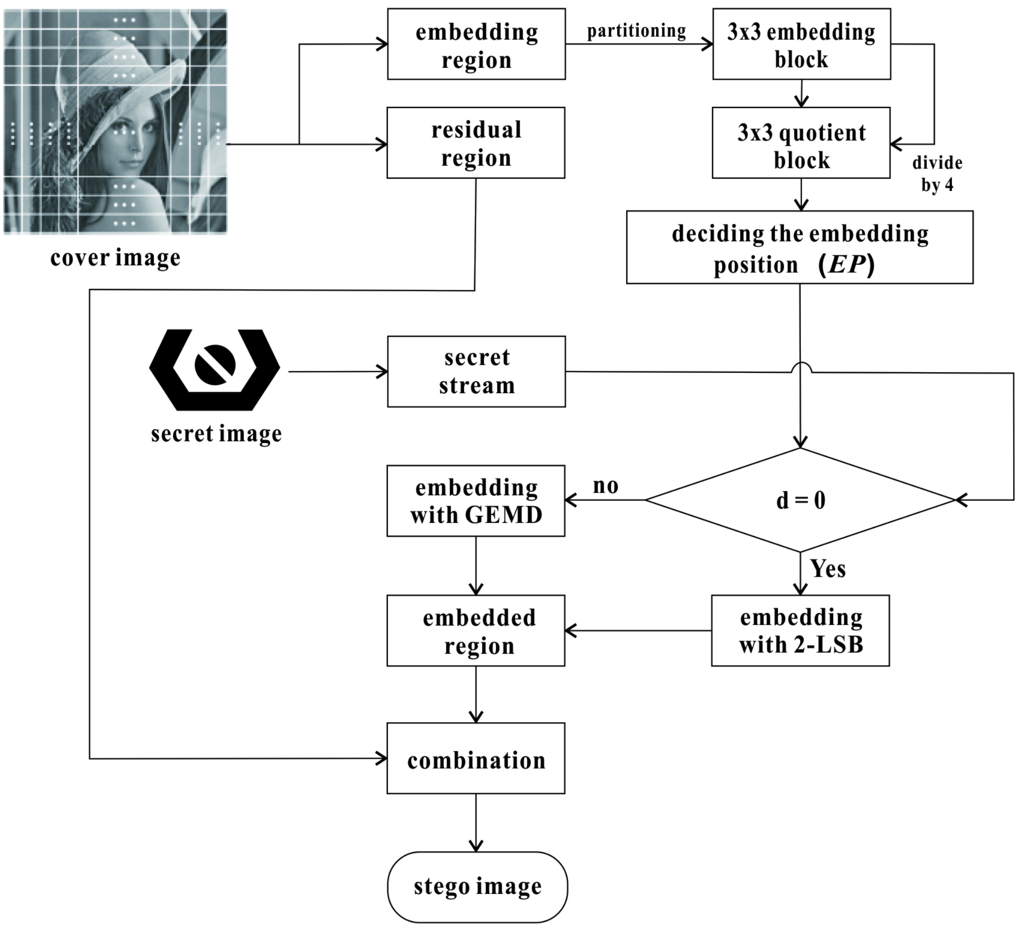


FIGURE 3. The framework of DA-GEMD embedding process

Example 3.1. *If there is a* 3×3 *cover block* $\mathbf{C} = \begin{bmatrix} 97 & 103 & 92 \\ 98 & 105 & 93 \\ 101 & 107 & 86 \end{bmatrix}$ *and the secret bitstream is* $m = (101001101001)_2$. *We get the stego block* $\mathbf{S} = \begin{bmatrix} 98 & 101 & 93 \\ 98 & 105 & 93 \\ 102 & 106 & 85 \end{bmatrix}$ *by using the following steps.*

(Step 1): Divide the block by 4 and obtain a quotient set $Q = 24, 25, 23, 24, 26, 23, 25, 26, 21$. Therefore, the median value is $EP_I = [(24 + 25)/2] = 24$.

(Step 2): Calculate the difference d_i between EP_I and q_i, which results in:

$$\begin{bmatrix} |24-24| & |25-24| & |23-24| \\ |24-24| & |26-24| & |23-24| \\ |25-24| & |26-24| & |21-24| \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 2 & 1 \\ 1 & 2 & 3 \end{bmatrix}.$$

(Step 3): Embed secret bit-stream $(101001101001)_2$. Divide into two parts $(1010)_2$ and $(01101001)_2$ which is dependent on $d_i = 0$ or $d_i \neq 0$, respectively. That is to say, for $d_i = 0$, $(1010)_2$ will be embedded into x_1 and x_4 by using 2-LSB method directly, and, for $d_i \neq 0$, $(01101001)_2$ will be embedded by GEMD (embedding process shown as Table 1.

TABLE 1. The conversion table from x_i to x'_i ($n = 7$)

Terms	Parameter	The block's pixels by using GEMD data hiding						
		$i=2$	$i=3$	$i=5$	$i=6$	$i=7$	$i=8$	$i=9$
Element	x_i	103	92	105	93	101	107	86
Residue	r_i	3	0	1	1	1	3	2
New Residue	r_i'	2	1	1	1	1	2	2
Transitional Element	$x_i^\star$	102	93	105	93	101	106	86
Adjusting factor	a_i	-1	0	0	0	1	0	-1
Element of Stego Block	x_i'	101	93	105	93	102	106	85

(Step 4): Obtain the stego block $\mathbf{S} = \begin{bmatrix} 98 & 101 & 93 \\ 98 & 105 & 93 \\ 102 & 106 & 85 \end{bmatrix}$

3.1. **Extraction Procedure.** The receiver can recover the secret data from the stego image using the DA-GEMD-E algorithm.

Algorithm DA-GEMD-E (Extracting Algorithm for DA-GEMD Scheme)
Input: stego image I_S
Output: binary secret data stream M

(DA-GEMD-E-1): Obtain all 3×3-pixels for each block $(x_1, x_2, \ldots, x_9)$ from I_S and $O_{DA-GEMD}(I_S)$.

(DA-GEMD-2): For each block,

1. Calculate the quotient set $QE = \{qe_i = [x_i/4]$, for $i = 1, 2, \cdots, 9\}$ and residue set $RRE = \{re_i = x_i \bmod 4$; for $i = 1, 2, \cdots, 9\}$.
2. Find the median of QE as the unique EPE_I.
3. Calculate the difference de_i between qe_i and EPE_I.
4. According to following two cases, we can recover n bits secret message from x_i for $i = 1, 2, \cdots, 9$.

   Case A: $d_i = 0$,
   Recover two secret binary bits from the two least-significant bits of e_i.

   Case B: $d_i \neq 0$,
   Use the GEMD method to recover the secret message.

Therefore, $m' = (m_{CaseA}||m_{CaseB})_2$.

(DA-GEMD-E-3): Concatenate secret data m' from each block to form M.

Example 3.2. *If there is a* 3×3 *stego block* $\mathbf{S}' = \begin{bmatrix} 98 & 101 & 93 \\ 98 & 105 & 93 \\ 102 & 106 & 85 \end{bmatrix}$. *Then, we can recover the secret bit-stream* $(101001101001)_2$ *by using the following steps.*

(Step 1): Divide the block elements by 4 and obtain a quotient matrix $Q = \begin{bmatrix} 24 & 25 & 23 \\ 24 & 26 & 23 \\ 25 & 26 & 21 \end{bmatrix}$.

Therefore, the median value is $EP_I = [(24 + 25)/2] = 24$.

(Step 2): Subtracting the median value from each element in Q results in matrix $\mathbf{D}' = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 2 & 1 \\ 1 & 2 & 3 \end{bmatrix}$.

(Step 3): Recover secret data based on d_i.

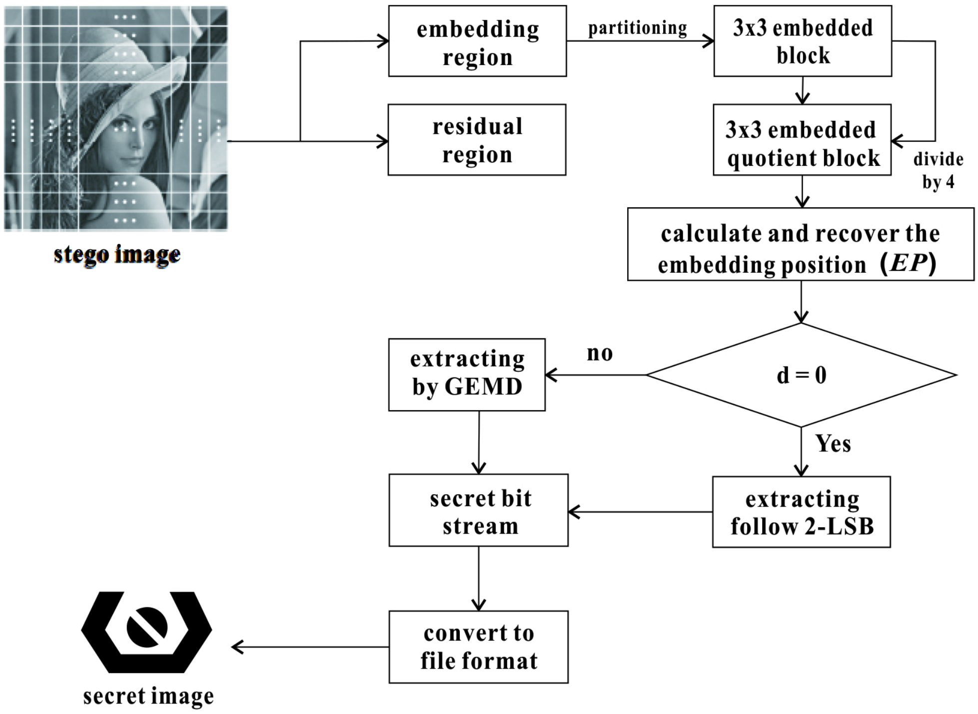


FIGURE 4. The framework of DA-GEMD extracting process

Case A: $d_i = 0$,

Recover two secret binary bits from the two least-significant bits of matrix D' corresponding to x_1 and x_4,

$$\begin{bmatrix} 98 \bmod 4 & 101 & 93 \\ 98 \bmod 4 & 105 & 93 \\ 102 & 106 & 85 \end{bmatrix}$$

.i.e., the secret data for Case A is $m_{CaseA} = (1010)_2$.

Case B: $d_i \neq 0$,

Use the GEMD method to recover the secret message.

$f_g(101, 93, 105, 93, 102, 106, 85) = (101 \times 1 + 93 \times 3 + 105 \times 7 + 93 \times 15 + 102 \times 31 + 106 \times 63 + 85 \times 127) \bmod 256 = 105 = (01101001)_2$, i.e., $m_{CaseB} = (01101001)_2$

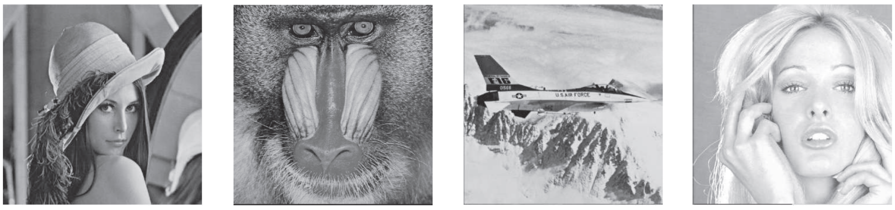
(Step 4): The secret data $m = (m_{CaseA}||m_{CaseB}) = (101001101001)_2$ is obtained from concatenating the results from Case A and Case B.

4. **Simulation and result.** In this section, we use our proposed scheme for simulations and show their results. For our experiment, the hardware environment is a personal computer with an Intel Core Duo 2 E4600 2.4 (GHz) CPU with 2G RAM. The operating system is Windows XP Professional and the experimental software is MATLAB R2007a. Four gray-scale 512 × 512 pixels cover images Lena, Baboon, F-16 and Tiffany were used as shown as Fig.5(a)-(d).

A binary secret mediaimage Fig.5(e) is embedded into the cover images with the resulting stego images shown in Fig.6. Bitplane attack [13] is used to 2-LSB and our proposed method and then the simulation results are shown as Fig.7 and Fig.8, respectively. Table 2 compares the experimental results for encoding our four test images using 1-LSB, 2-LSB and our proposed method. According to the results of Fig.8, bitplane analysis does not reveal any secret information from the stego image because the data hiding rate is not

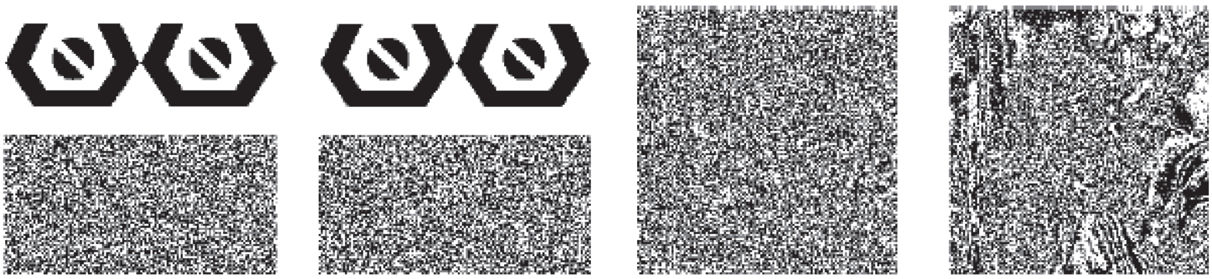
(a) Lena (b) Baboon (c) F-16 (d) Tiffany (e)The secret data

FIGURE 5. The cover images(a)-(d) and the secret media(e)



(a) Lena (b) Baboon (c) F-16 (d) Tiffany

FIGURE 6. The stego images



(a)1^{st} bitplane (b)2^{nd} bitplane (c) 3^{rd} bitplane (d)4^{th} bitplane

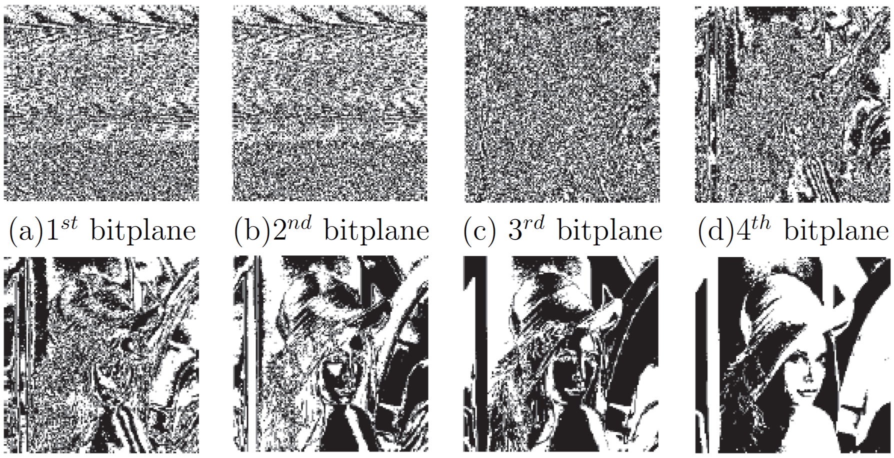
(e)5^{th} bitplane (f)6^{th} bitplane (g)7^{th} bitplane (h)8^{th} bitplane

FIGURE 7. The bit-plane attack is used to data hiding based on 2-LSB

random in each block. Consequently, the data hiding method based on division arithmetic can be a viable approach to protect embedded data from bitplane analysis.

Both the capacity and stego image quality of proposed scheme fall between the 2-LSB and 1-LSB methods. Table 3 compares the capabilities of the 1-LSB, 2-LSB and DAGEMD schemes for cover image Lena.

5. **Conclusions.** In this paper, a new LSB block data hiding method was proposed based on division arithmetic(DA) and GEMD. DA-GEMD is shown to be more secure than simple LSB substitution against Bitplane attack. The proposed scheme compares well to LSB in terms of quick execution, good PSNR, high capacity, and easy to implement.



(a)1^{st} bitplane (b)2^{nd} bitplane (c) 3^{rd} bitplane (d)4^{th} bitplane

(e)5^{th} bitplane (f)6^{th} bitplane (g)7^{th} bitplane (h)8^{th} bitplane

Figure 8. The bitplane attack is used to our proposed scheme

Table 2. The scheme comparison of experimental results

Stego image	Method	Max Capacity	PSNR(dB)
Lenna	1-LSB	262,144	51.2
	2-LSB	524,288	45.7
	DA-GEMD	409,826	46.8
Baboon	1-LSB	262,144	51.2
	2-LSB	524,288	45.7
	DA-GEMD	363,071	47.3
F-16	1-LSB	262,144	51.2
	2-LSB	524,288	45.7
	DA-GEMD	427,881	46.6
Tiffany	1-LSB	262,144	51.2
	2-LSB	524,288	45.7
	DA-GEMD	420,201	46.7

PSNR: peak signal to noise ratio

Table 3. Comparison results for 1-LSB,2-LSB, DA-GEMD

Item	1-LSB	2-LSB	DA-GEMD
Prevent bitplane attack	No	No	Yes
Easy to implement	Yes	Yes	Yes
Bits per pixel (bpp)	1.00	2.00	1.56
Max. embedding capacity(bits)	262,144	524,288	409,826
Secret image(NFU)	51.2 dB	45.7 dB	46.8 dB
Secret image(random)	51.2 dB	44.2 dB	45.9 dB

Acknowledgement. The authors extend appreciation to the anonymous reviewers for providing invaluable comments during the revision process. This work was supported by National Science Council Taiwan grant(NSC 102-2218-E-150 -001).

REFERENCES

[1] A. Benhocine, L. Laouamer, L. Nana, and A. C. Pascu, New images watermarking scheme based on singular value decomposition, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 1, pp. 9-18, 2013.

[2] H. C. Huang, and F. C. Chang, *Journal of Expert Systems with Applications*, vol. 40, no. 1, pp. 34-43, 2013.

[3] H. C. Huang, and W. C. Fang, Authenticity preservation with histogram-based reversible data hiding and quadtree concepts, *Journal of Sensors*, vol. 11, no. 10, pp. 9717-9731, 2011.

[4] T. D. Kieu, and C. C. Chang, A steganographic scheme by fully exploiting modification directions, *Journal of Expert Systems with Applications*, vol. 38, no. 8, pp. 10648-10657, 2011.

[5] C. S. Kim, D. K. Shin, D. G. Shin, and X. P. Zhang, Improved steganographic embedding exploiting modification direction in multimedia communications, *Journal of Communications in Computer and Information Science*, vol. 186, Springer, pp. 130-138, 2011.

[6] W. C. Kuo, L. C. Wuu, and S. H. Kuo, The high embedding steganographic method based on general multi-EMD, *Proc. of International Conference on Information Security and Intelligence Control*, pp. 288-291, 2012.

[7] W. C. Kuo, and C. C. Wang, Data hiding based on generalized exploiting modification direction method, *Journal of The Imaging Science Journal*, vol. 61, no. 6, pp. 484-490, 2013.

[8] A. Latif, An adaptive digital image watermarking scheme using fuzzy logic and tabu search, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4. no. 4, pp. 250-271, 2013.

[9] C. F. Lee, Y. R. Wang, and C. C. Chang, A steganographic method with high embedding capacity by improving exploiting modification direction, *Proc. of The 3rd International Conference on International Information Hiding and Multimedia Signal Processing*, pp. 497-500, 2007.

[10] J. Mielikainen, LSB matching revisited, LSB matching revisited, *Journal of IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285-287, 2006.

[11] F. A. P. Petitcolas, R. j. Anderson, and M. G. Kuhn, Information hiding - A Survey, *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062-1078, 1999.

[12] S. Weng, S. C. Chu, N. Cai, and R. Zhan, Invariability of mean value based reversible watermarking, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 4, no. 2, pp. 90-98, 2013.

[13] A. Westfeld, and A. Pfitzmann, *Attacks on steganographic systems*, LNCS 1768, Springer, pp. 61-76, 2000.

[14] X. Zhang, and S. Wang, Effcient steganographic embedding by exploiting modification direction, *Journal of IEEE Communications Letters*, vol. 10, no. 11, pp. 781-783, 2006.

[15] Advanced Encryption Standard, NIST FIPS PUB 197, National Institute of Standard and Technology, 2001.

[16] Data Encryption Standard, NIST FIPS PUB 46-2, National Institute of Standard and Technology, 1993.