STATISTICAL ATTACK RESISTANT MULTI-BIT STEGANOGRAPHY USING MOBILE KEYPAD CHARACTER ENCODING

Uttiya Ghosh[1], Smritikana Maity[2], Imon Mukherjee[3]
Dept. of Computer Science and Engineering
St. Thomas' College of Engineering and Technology, Kolkata
{[1]uttiyaghosh, [2]smritikanamaity94, [3]mukherjee.imon}@gmail.com

Abstract
Communication over the internet has resulted in very fast communication over the years, but it has resulted in degradation of reliability and secrecy which brings in the concept of steganography. Many steganographic methods introduced with time has helped in overcoming this disadvantage by increasing the security during data communication. While least significant bit embedding steganographic algorithms in images helps to add security but this also results in low embedding capacity, multiple bit embedding adds on to the capacity of data that can be hidden. The process proposed in this paper uses a special mobile keypad character code for every character which is embedded in a randomized circular row major format in cover images. This code is based on the mobile keypad structure which not only adds to data security over algorithms using conventional binary code but also makes the proposed algorithm capable of withstanding visual attacks with almost no perceivable distortion in the stego images. Along with a high embedding efficiency and embedding capacity the proposed algorithm can withstand structural attacks as well as statistical attacks viz. Sample Pair test and chi-square test due to its randomized nature. Finally, the success of this method lies in its compilation with the standards set by Stirmark Benchmark.

Key words: *Mobile keypad character code, Data security, Embedding efficiency, Embedding capacity, Statistical attacks.*

1 Introduction and Motivation

Steganography is an art of transmitting message from sender to receiver in a secured manner. Secrecy has to be maintained in such a way that the cover media may not give any clue that the transmitting media is carrying any secret data. Mainly digital images are used as transmitting media known as cover image and the image with hidden data is known as stego image.

There are various techniques which have resulted in the advancement of this field. Mukherjee et al. [5] introduced a new technique where the potential pixel and spatial domain are used to embed the message which is selected by using the frequency domain of the cover image. Veron et al. [13] has shown the different ways in which algebraic coding theory can be used to define efficient steganographic methods. Aroukatos et al. [1] has used a union of Fibonacci numbers and Catalan numbers that form a set which satisfies Zeckendorf's theorem and thus helps in data encoding. Zhang et al. [16] has described a multi-bit steganography method for palette images where every gregarious color that has a close neighboring color in the palette is used for representing many secret bits. In the process introduced by Islam et al. [6] the message is embedded in the edge of the cover image. The use of weaker edges increases with the amount of embedded data. Mandal et al. [7] has used the Pixel Value Differencing (PVD) method, but for this process the pixel value of the stego image may vary beyond 0-255 which is solved through a special process. In the method used by Chang et al. [2] relocation and side matching is carried out that helps to propose an embedding scheme which is reversible for VQ-compressed images. Zhang et al. [15] has taken the last 3 bits for embedding the secret message. For the LSB embedding, the cover pixels that have to be modified by one are located. For second LSB embedding, determination of adding or subtracting one for located samples is done and then the samples to be modified by two are chosen and to embed the third bit, the samples that are to be added or subtracted by two are selected. Medeni et al. [8] has introduced the process where the cover image is divided into blocks of same size, and embedding of the message is done in the block edge based on the number of ones present in the left four bits of the pixels. Ravindranath et al. [12] has proposed a method where Fibonacci numbers are used for encryption and plane decomposition is used on weights of those numbers for benefits of cryptography. Yang et al. [14] has described a new technique to improve the histogram based reversible data hiding concept.

In this paper we have introduced a new way to embed secret data using mobile keypad character code, where every character is represented by a unique code word and this helps to increase the embedding efficiency. Moreover, data is embedded in a randomized way which helps in increasing the security and thus it becomes difficult for the attackers to retrieve the secret data.

2 Basic Ideas

This novel steganographic method provides a new scheme of representing characters in binary sequences based on a concept similar to that used in a mobile keypad. The entire character set is divided into 16 blocks and each block represents 8 characters just like a key in a mobile keypad which generates around 8 characters. Each character is represented by a block code (four MSBs) and a character code (three LSBs). The representation is shown in Table 1. Some spaces are intentionally left blank so that one may enter new characters in the vacant places if necessary.

Table 1. Total chart containing the block code and character code for every character

C.C. B.C.	000	001	010	011	100	101	110	111
0000	1	`	!	@		$	%	^
0001	a	b	c	A	B	C	2	
0010	d	e		D	E	F	3	*
0011	g	h	i	G	H	I		(
0100		k	l	J	K	L	5	)
0101	m	n		M	N		6	_
0110	p		r	s	P	Q	R	S
0111	t	u	v	"	U	V	8	7
1000	w	x	y	z	W	X	Y	
1001	~	9	0	-		=	{	}
1010	[	]		|	:	;	“	‘
1011	space		>	.	,	?	/	#
1100		’	‘	´	—	&	¥	€
1101	¤	¿	¡	new line	f	4		°
1110		©	O	q	T	Z	+	\
1111	®	<	”	j	o		£	carriage return

3 Proposed Method

A novel steganographic technique is proposed here which uses the mobile keypad character encoding concept to embed character in 24-bit color images in which embedding is done in randomized manner.

Embedding technique:

Consider a text message with l characters as shown in Table 2.

Table 2. Text message arrangement

T_1	T_2	…	T_i	T_{i+1}	…	T_l

Then convert the values of Table 2 to their corresponding mobile keypad character value with the help of Table 1 to generate Table 3.

Table 3. Mobile keypad character encoded text

M_1	M_2	…	M_i	M_{i+1}	…	M_l

Consider a cover image (of size $h \times w$) as shown in Table 4 in which Table 3 is to be embedded.

Table 4. $h \times w$ cover image matrix

$\omega_{1,1}$	$\omega_{1,2}$	…	$\omega_{1,w}$
$\omega_{2,1}$	$\omega_{2,2}$	…	$\omega_{2,w}$
…	…	…	…
$\omega_{h,1}$	$\omega_{h,2}$	…	$\omega_{h,w}$

In Table 4 every value of $\omega_{i,j}$ is a 24 bit value which is represented in Table 5 where r_i denotes the red bits of the pixel value with the MSB as r_7 and LSB as r_0. Similarly, g_i and b_i denotes the green bits and blue bits of the pixel value respectively.

Table 5. Binary representation of pixel values

r_7	r_6	…	r_0	g_7	g_6	…	g_0	b_7	b_6	…	b_0

Let us consider any particular value of Table 3 (say, M_k) which we will embed in Table 5 where x_i represents the Block code and y_i represents the character code. The binary representation of M_k is shown in Table 6.

Table 6. Binary representation of M_k

x_6	x_5	x_4	x_3	y_2	y_1	y_0

We calculate the value of α from Equation (3) where ∂_α and Δ_α is obtained from Equation (1) and Equation (2) respectively.

$$\partial_\alpha = abs\ \{(x_6 - r_3)*2^3 + (x_5 - r_2)*2^2 + (x_4 - r_1)*2^1 + (x_3 - r_0)*2^0\}, \text{ where } abs \text{ denotes absolute value.} \tag{1}$$

$$\Delta_\alpha = (x_6 + r_3)*2^3 + (x_5 + r_2)*2^2 + (x_4 + r_1)*\ 2^1 + (x_3 + r_0)*2^0. \tag{2}$$

$$\alpha = \begin{cases} x_6 * 2^3 + x_5 * 2^2 + x_4 * 2^1 + x_3 * 2^0\ , & \text{if } \partial_\alpha \leq abs\ (15 - \Delta_\alpha); \\ (x_6\ \Theta\ 0) * 2^3 + (x_5\ \Theta\ 0) * 2^2 + (x_4\ \Theta\ 0) * 2^1 + (x_3\ \Theta\ 0) * 2^0, \text{ otherwise.} & \end{cases} \tag{3}$$

We calculate the value of β from Equation (6) where ∂_β and Δ_β is obtained from Equation (4) and (5) respectively.

$$\partial_\beta = abs\ \{(y_2 - g_2)*2^2 + (x_1 - g_1)*2^1 + (x_0 - g_0)*2^0\}, \text{ where } abs \text{ denotes absolute value.} \tag{4}$$

$$\Delta_\beta = (y_2 + r_2)*2^2 + (y_1 + r_1)*2^1 + (y_0 + r_0)*2^0. \tag{5}$$

$$\beta = \begin{cases} y_2*2^2 + y_1*2^1 + y_0*2^0, & \text{if } \partial_\beta \leq abs\ (7 - \Delta_\beta); \\ (y_2\ \Theta\ 0)*2^2 + (y_1\ \Theta\ 0)*2^1 + (y_0\ \Theta\ 0)*2^0, \text{ otherwise.} & \end{cases} \tag{6}$$

The value of γ is obtained from Equation (7).

$$\gamma = \begin{cases} 0, \text{ if } \partial_\alpha \leq abs\ (15 - \Delta_\alpha); \\ 1, \text{ if } \partial_\alpha > abs\ (15 - \Delta_\alpha); \\ 2, \text{ if } \partial_\beta \leq abs\ (7 - \Delta_\beta); \\ 3, \text{ if } \partial_\beta > abs\ (7 - \Delta_\beta). \end{cases} \tag{7}$$

We begin the embedding process from pixel value $\omega_{1,1}$ of Table 4. Let ω_r denote the red component of the pixel value under consideration. We obtain $\mathbf{\Omega}_r$ from Equation (8) which is replaced in place of ω_r in the cover image to obtain the stego image.

$$\mathbf{\Omega}_r = \omega_r - \text{mod}\ (\omega_r, 16) + \alpha \tag{8}$$

Similarly, $\mathbf{\Omega}_g$ is obtained using Equation (9) and is used to replace ω_g (the green component of the pixel value under consideration).

$$\mathbf{\Omega}_g = \omega_g - \text{mod}\ (\omega_g, 8) + \beta \tag{9}$$

Finally, $\mathbf{\Omega}_b$ is obtained using Equation (10) which is used to replace ω_b (the blue component of the pixel value under consideration).

$$\mathbf{\Omega}_b = \omega_b - \text{mod}\ (\omega_{b,} 4) + \gamma \tag{10}$$

The next position of embedding is found by moving α + 1 pixels from the current position in a circular row major manner of Table 4 until the entire text gets embedded or there is a character embedded in every pixel value of the cover image.

Algorithm 1: Algorithm for embedding secret message bits into the cover image.

```
Input: An image of size $h×w$.
Output: The stego image component containing the secret data.

Convert the text message into mobile keypad character encoded data using Table 1;
Calculate $\alpha$, $\beta$ and $\gamma$ using Equation (3), (6) and (7) respectively;
while the entire text is not embedded or there are pixel values of the cover image with no
    secret information do
        Calculate $\Omega_r$, $\Omega_g$ and $\Omega_b$ from Equation (8), (9) and (10) respectively;
        Calculate the next pixel value position by moving $\alpha + 1$ pixels from the current
        position in a circular row major manner;
    end
Output the transformed image.
```

Retrieval technique:

The process of retrieving the embedded message from the stego image starts with the technique of getting the mobile keypad character code from the image, and then it is converted into the embedded text using Table 1.

Consider the stego image (of size $h×w$) as shown in Table 7 from which the mobile code has to be retrieved.

Table 7. $h×w$ stego image matrix

$\Omega^*_{1,1}$	$\Omega^*_{1,2}$	…	$\Omega^*_{1,w}$
$\Omega^*_{2,1}$	$\Omega^*_{2,2}$	…	$\Omega^*_{2,w}$
…	…	…	…
$\Omega^*_{h,1}$	$\Omega^*_{h,2}$	…	$\Omega^*_{h,w}$

From every value of $\Omega^*_{i,j}$ of Table 7 a 24 bit value is obtained where there are 8 bits each for red component (denoted as Ω^*_r), green component (denoted as Ω^*_g) and blue component (denoted as Ω^*_b). We will get the value of α^* from the red component of the pixel value (denoted as Ω^*_r) using the Equation (11).

$$\alpha^* = \mathrm{mod}\ (\Omega^*_r, 16) \tag{11}$$

Similarly, we will get β* from the green component of the pixel value (denoted as Ω^*_g) using the Equation (12).

$$\beta^* = \mathrm{mod}\ (\Omega^*_g, 8) \tag{12}$$

The value of γ^* is obtained from the blue component of the pixel value (denoted as Ω^*_b) using the Equation (13).

$$\gamma^* = \mathrm{mod}\ (\Omega^*_b, 4) \tag{13}$$

Using α^*, β^* and γ^* values we the block code (θ) is computed as shown in Equation (14) while the character code (Φ) is computed as shown in Equation (15).

$$\theta = \begin{cases} \alpha^*, & \text{if } \gamma^*=0; \\ 15 - \alpha^*, & \text{if } \gamma^*=1. \end{cases} \qquad (14)$$

$$\Phi = \begin{cases} \beta^*, & \text{if } \gamma^*=3; \\ 7 - \beta^*, & \text{if } \gamma^*=4. \end{cases} \qquad (15)$$

Finally, the mobile keypad character codes are obtained from Equation (16) and are arranged as shown in Table 8. Then they are converted into corresponding characters with the help of Table 1 to form the Table 9.

$$M_i = \theta \times 8 + \Phi \qquad (16)$$

Table 8. Mobile keypad character code obtained from stego image

M_1	M_2	...	M_i	...	M_l

Table 9. Text message retrieved

T_1	T_2	...	T_i	...	T_l

The next position for retrieving the hidden text is obtained by moving α^*+1 pixels from the current position in a circular row major manner of Table 7 until the hidden text is retrieved.

Algorithm 2: Algorithm for retrieving the secret data from the stego image.

```
Input: The stego image containing the secret data.
Output: The secret data retrieved from the stego image.

while the entire text is not retrieved do
    Calculate $\alpha^*$, $\beta^*$ and $\gamma^*$ values from the pixel value using Equation (11), (12) and (13) respectively;
    Calculate $\theta$ and $\Phi$ from Equation (14) and (15) respectively;
    Calculate mobile keypad character codes using Equation (16);
    Convert the codes into corresponding characters with the help of Table 1 to form the Table 9;
    Calculate the next pixel value position by moving $\alpha^*$ + 1 pixels from the current position in a circular row major manner;
end
Output the embedded secret message.
```

4 Experimental Result and Comparison with Existing Algorithms

Several attributes of the cover image and stego image obtained by the proposed technique is calculated for visual quality analysis viz. bit error rate (BER), mean square error (MSE), embedding efficiency and Peak Signal-to-Noise ratio (PSNR). This method is compared with various other existing methods. The fact that our algorithm is better can be seen from the comparative study shown in Table 10. It is seen that the PSNR (Peak Signal to Noise Ratio) of method B [8] and C [7] is better than our method but our method can embed a much higher amount of data claiming higher efficiency.

Table 10. Comparison of Capacity and PSNR with other methods: A [2], B [8], C [7], D [16] and E [14]

Method described in	Capacity (bits)	Average PSNR (dB)
A	<10738	<31.43
B	40017	42.68
C	1166296	42.26
D	<370000	<38.5
E	334589	30.25
Proposed Method	2097152	41.16084

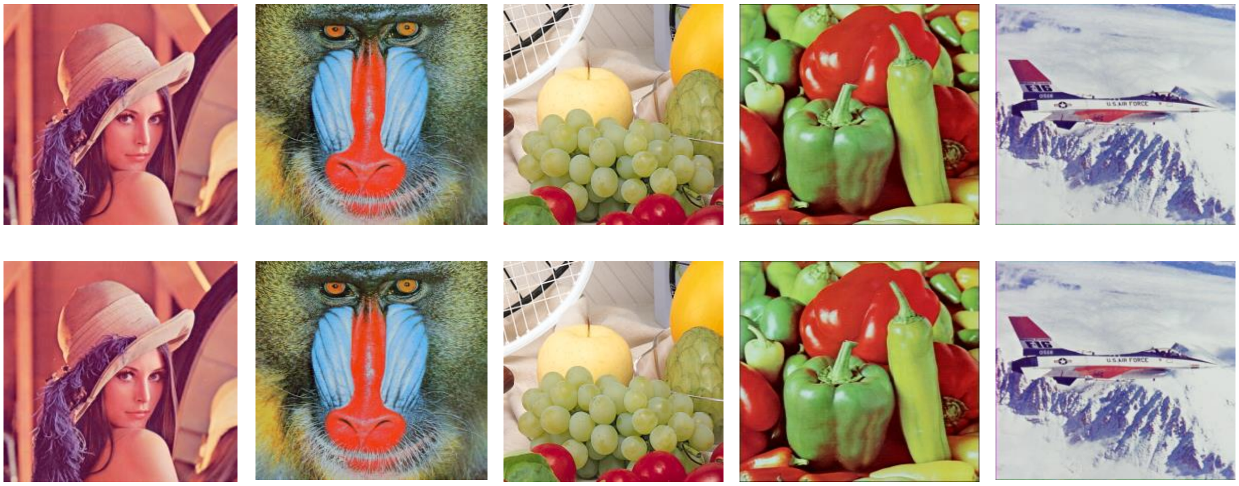

Fig. 1. Cover version (top) and stego version (bottom) of five images viz. Lenna, Baboon, Fruits, Peppers, Jet of size 512×512

Perceptibility Analysis

This novel method is widely tested over different standard test images using the first 2,62,144 characters of the story “Black Beauty” obtained from [17]. The experimental outcomes of such tests are really encouraging and are shown in Table 11. Visual analysis of the tested images can be seen in Figure 1 which shows that there is almost no visual distortion.

Table 11. MSE, PSNR and BER values of sample images.

Image Names	MSE	PSNR	BER
Lenna	5.0193	41.1243	0.1458
Baboon	4.6478	41.4583	0.15
Jet	5.042	41.1048	0.1455
Peppers	4.9949	41.1455	0.1449
Fruits	5.1994	40.9713	0.1458

Embedding Efficiency

The strength of a steganographic method can also be determined by embedding efficiency according to Fridrich et al. [4]. It can be defined as follows:

Definition 1. The Embedding efficiency of a particular steganographic technique is the expected number of data bits hidden per embedding change.

$$\Upsilon = \frac{\text{No. of bits embedded}}{\frac{\text{Distortion in proposed method}}{\text{Maximum possible Distortion}}}$$

In a LSB based steganographic technique, the embedding efficiency is 1/ (1/2) = 2. In the proposed method embedding is done in a 4-3-2 fashion to hide a single character i.e. 4 bits are embedded in the red, 3 in the green and 2 in the blue component. Although it may seem that 9 bits are embedded per 24 bit pixel actually only 8 bits are embedded (since one character is represented by 8 bits). In our method embedding efficiency for red component is 3, for blue component is 7, and for green component, it is 2. For a normal 3-3-2 embedding scheme an efficiency of 3, 3 and 2 is obtained. Hence, our method has a better embedding efficiency.

Color Frequency Test

Matching of color frequency distribution of the cover image with the stego image is done with the help of this test. According to [5, 9] this method can detect the presence of continuous distortion that occurs from the beginning of the image and cannot detect presence of data embedded in a random fashion. As it is evident from previous sections that in the proposed technique data is embedded in a random fashion which makes the proposed technique resistant to this attack.

Sample Pair Method

According to Dumitrescu et al. [3] the method works very well for LSB embedding scenarios. Since, this novel method embeds data in the multi-bit fashion therefore the proposed technique is capable of withstanding sample pair method.

StirMark Analysis.

Along with the efficiency, the robustness of all steganographic algorithms requires a proper analysis with respect to some standardized methods. These analyses were performed using StirMark 4.0 [10, 11]. The proposed technique was successful in providing some excellent results. The very minute deviation between the values of the cover and the stego image shown in Table 12 is an indication to the robustness of the proposed algorithm.

Table 12. StirMark analysis of proposed technique on cover and stego version of Lenna (512×512)

Test	**Factor**	**Cover (dB)**	**Stego (dB)**
Self Similarities	1	36.322	35.0116
Self Similarities	2	48.6309	48.5033
Self Similarities	3	32.3915	32.3163
PSNR	10	38.3482	38.3105
Add Noise	20	9.61381	9.63809
Add Noise	40	8.23211	8.22795
Add Noise	100	7.43846	7.47494
Small Random Distortions	0.95	17.8137	10.8045
Small Random Distortions	1.00	17.5981	10.5727
Small Random Distortions	1.05	17.4292	10.3843
ConvFilter	1.00	11.5344	11.497
ConvFilter	2.00	-6.92511	-6.90225
Median Cut	3.00	33.3908	33.0182

5 Conclusion

In this paper a novel steganographic technique has been discussed where we have used mobile keypad character code for the representation of data that is to be embedded in the cover image. Every character has a special code word of representation, which adds complexity as well as increased capacity of embedding to the data transmission and the randomization done while embedding results in the increased security and privacy. This process embeds the secret data in the cover image with almost no visual distortion. Thus it is better than any sequential conventional steganographic algorithms with respect to the point of privacy and embedding capacity.

References

1. Aroukatos, N., Manes, K., Zimeras, S., Georgiakodis, F.: Data Hiding Techniques in Steganography using Fibonacci and Catalan numbers, In: Ninth International Conference on Information Technology - New Generations (2012).
2. Chang, C.C., Lin, C.Y.: Reversible Steganography for VQ-Compressed Images Using Side Matching and Relocation, In: IEEE Transactions on Information Forensics and Security, Vol. 1, No. 4, pp. 493-501 (December 2006).
3. Dumitrescu, S., Wu, X., Wang, Z.: Detection of LSB steganography via Sample Pair Analysis, In: IEEE Transactions on Signal Processing, Vol. 51, No. 7 (July 2003).
4. Fridrich, J., Lisonek, P.: Grid Colorings in Steganography. In: IEEE Transactions on Information Theory, Vol. 53(4), pp. 1547-1549 (April 2007).

5. Mukherjee, I., Podder, A.: DCT Based Robust Multi-bit Steganographic Algorithm, In: 2nd International Conference on Advanced Computing, Networking and Informatics, Vol. 28 (June 2014).
6. Islam, S., Modi, M.R., Gupta, P.: Edge-based image steganography, In: EURASIP Journal on Information Security (2014).
7. Mandal, J.K., Das, D.: Colour Image Steganography Based on Pixel Value Differencing in Spatial Domain, In: International Journal of Information Sciences and Techniques (IJIST), Vol.2, No.4, pp. 83-93 (July 2012).
8. Medeni, M.B.O., Souidi, E.M.: A Generalization of the PVD Steganographic Method, In: International Journal of Computer Science and Information Security, Vol. 8, No. 8 (November 2008).
9. Provos, N.: Defending against Statistical Steganalysis. In: Tenth USENIX Security Symposium, pp. 325-335 (2001).
10. Petitcolas, F.A.P., Anderson, R.J., Kuhn, M.G.: Attacks on copyright marking systems, In: Second International Workshop on Information Hiding, IH98, Portland, Oregon, U.S.A., Proceedings, LNCS 1525, Springer-Verlag, ISBN 3-540-65386-4, pp. 219-239 (April 15-17, 1998).
11. Petitcolas, F.A.P.: Watermarking schemes evaluation, In: IEEE Signal Processing, Vol. 17, No. 5, pp. 5864 (September 2000).
12. Ravindranath, C.C., Bhatt, A.K., Bhatt, A.: Adaptive Cryptosystem for Digital Images using Fibonacci Bit-Plane Decomposition, In: International Journal of Computer Applications, Vol. 65, No.14 (March 2013).
13. Veron, P.: Code based cryptography and steganography, In: 5th International Conference on Algebraic Informatics (September 2013).
14. Yang, C.H., Tsai, M.H.: Improving Histogram-based Reversible Data Hiding by Interleaving Predictions, In: IET Image Processing (2009).
15. Zhang, X.: Efficient Data Hiding With Plus-Minus One or Two, In: IEEE Signal Processing Letters, Vol. 17, No. 7, pp. 635-638 (July 2010).
16. Zhang, X., Wang, S., Zhou, Z.: Multi-bit Assignment Steganography in Palette Images, In: IEEE Signal Processing Letters, Vol. 15 (2008).
17. http://pinkmonkey.com/dl/library1/digi105.pdf