http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 13 (12): 1945-1952, 2014
ISSN 1812-5638 / DOI: 10.3923/itj.2014.1945.1952

MSB Based Embedding with Integrity: An Adaptive RGB Stego on FPGA Platform

Sundararaman Rajagopalan, Pakalapati J.S. Prabhakar, Mucherla Sudheer Kumar,
N.V.M. Nikhil, Har Narayan Upadhyay, J.B.B. Rayappan and Rengarajan Amirtharajan
School of Electrical and Electronics Engineering, SASTRA University, India

Abstract: Information hiding as a field is extremely fast developing and one of the very few fields that derive its power from its inherent need for sophistication and complexity in its algorithms. Out of the ever growing list of information hiding methods, steganography has emerged as the front runner because of clandestine nature. Image steganography is the heart of steganography because of wide availability of images in local media, variable capacity and its ability to cloak the very existence of secret data in it. There are innumerable software methods available for steganography but there is a dearth of hardware implementations to serve this purpose. In this study, we have devised a FPGA implementation that randomises the volume of data embedded in each pixel, according to the same MSBs. This implementation also contains a novel method to check the integrity of the embedded data by providing a mechanism by which we can weed out any modifications in the secret data, to an extent that it can even pinpoint the exact pixel in which the change has occurred. This algorithm has been implemented on EP2C35F672C6 FPGA. The synthesis report and Timing analysis have also been discussed in this study.

Key words: Adaptive LSB substitution, information hiding, FPGA for security, random image steganography, pixel indicator method, hardware steganography

INTRODUCTION

Information exchange takes a crucial position in advancement associated with any field. In a way, information exchange and advancement are mutually inclusive i.e., rapid growth in a particular field increases the need for information and vice versa. Volatile nature of some kinds of information, like military secrets prevented its transfer over an insecure channel, as they could cause serious consequences if released. So, the need for security of the information transmitted over any channel grew (Stefan and Fabin, 2000). Information hiding rose to meet this demand by providing efficient methods to increase security of the transmitted information.

The first method that was used to provide security was cryptography. Cryptography (Schneier, 2007) basically provided algorithms to scramble data that was to be transmitted. The data to be transmitted was encrypted according to a particular algorithm and on reception the decryption algorithm was used to obtain the original secret data. This seemingly secure method did come with some pitfalls i.e., if the scrambled information that is being sent over the channel is intercepted by an eavesdropper, the tapped data will consist of a series of incoherent symbols which will instantly draw his attention and thereby increasing the probability of the information being decoded or changed. Or even worse, the attacker can modify the information thereby rendering it useless to the decryption algorithm. To address the issues that cryptography posed, steganography was invented. It is unparalleled in its performance because of its proactive nature. Steganography (Bender *et al*., 1996; Amirtharajan *et al*., 2012) also possessed another sterling quality. That is the data to be sent over the channel is embedded over an innocuous carrier in such a way that it is hidden from plain sight. So, even if the carrier is extracted from the channel the presence of covert data in it was not evident (Thenmozhi *et al*., 2012; Zhu *et al*., 2011). This inherent trait of steganography made it matchless. Steganography is also viewed as a hide and seek game, marking it as an art (Provos and Honeyman, 2003).

Image steganography (Cheddad *et al*., 2010; Thanikaiselvan *et al*., 2011, Zaidan *et al*., 2010, Zanganeh and Ibrahim, 2011) the crème da le crème of all the steganography methods, because of the easy availability of images and the ease with which we can embed data in them with very high imperceptibility. Two vast domains of Steganography do exist (spatial, frequency). The former involves the embedding of data in cover pixels straight (Petitcolas *et al*., 1999) whereas the latter involves embedding of data in transformed image pixels. Three main constituents of steganography are cover (entity employed to hide secret), stego (end result with buried secret) i.e. cover along with secret.

Corresponding Author: Sundararaman Rajagopalan, School of Electrical and Electronics Engineering, SASTRA University, India

Image based steganography (Petitcolas *et al*., 1999) has its origin in LSB, Pixel Indicator; to vary the intensiveness in images, pixel values or its intensities are modified or malformed (Chan and Cheng, 2004). LSB is one eminent modus operandi in steganography cover pixels of fixed length is replaced by the same of secret bits. Since it causes some noticeable deformation, OPAP comes to rescue (Amirtharajan and Balaguru, 2009; Amirtharajan and Rayappan, 2012a, b, c, d; Chan and Cheng, 2004). Unlike others, certain adaptive routines encourage embedding of variable bits with healthier visual quality (Amirtharajan *et al*., 2011). Nonetheless, it costs embedding capability.

Various RGB (Amirtharajan *et al*., 2010a, b) and grayscale images (Thanikaiselvan *et al*., 2011) steganographic techniques have been proposed Gutub *et al*. (2008) proposed high capacity information embedding on RGB image. Integrity checking along with adaptive embedding has been proposed in some steganography algorithms (Amirtharajan *et al*., 2011). Modulus function based stego systems (Thien and Lin, 2003) and spread spectrum based steganographic technique (Marvel *et al*., 1999; Thenmozhi *et al*., 2012) are present in the information hiding technique gallery.

Hardware based steganographic systems (Rajagopalan *et al*., 2012a) exist in the literature. A ConText steganographic technique has been implemented in Cyclone II FPGA (Gomez-Hernandez *et al*., 2008). Rajagopalan and Upadhyay (2011) have proposed LFSR based image steganographic system on FPGA. Quad block based hardware steganographic system on FPGA in Rajagopalan *et al*. (2012b) follows a guiding block for information hiding.

The hardware steganographic methods suggested earlier used approaches for hiding fixed number of secret bits in the grayscale cover image on FPGA. But our paper proposes an adaptive information hiding on RGB cover image stored in external SRAM connected to Cyclone II FPGA where PI decides the K value for each pixel. We also propose an innovative integrity check mechanism for verifying the integrity of the transmitted data and to pinpoint the location of a change in an embedded pixel if any.

PROPOSED METHODOLOGY

Among multitudinous Pixel indicator embedding schemes, three methods are adopted in the proposal.

Method 1: Of the three RGB planes available, we use RED plane as indicator channel, other two as data conduits. A k is determined by adding the MSB of R, G, B pixels. It decides how many pixels have to be embedded, Thus the name "MSB based embedding":

$$K = MSB(R)+MSB(G)+MSB(B)+1$$

If indicator's final two bits are "00" there will not be any embedding. If it is "01" the data will be embedded in BLUE channel. If the indicator returns a "10", the data will be embedded in GREEN channel and if it returns 11 then the data will be embedded in both GREEN and BLUE channels. The embedding is simple LSB substitution. The last row and column are used for integrity check mechanism instead of embedding data.

Method 2: This one is the same as first; except user has the privilege to select the indicator channel. Accordingly the other two channels become data channels.

Method 3: Unlike the other two methods, in this method there is no fixed indicator channel. It is like a cyclic process in which for starting pixel, RED is pointer; for the second pixel GREEN is the indicator and for the third pixel BLUE is the indicator channel. For the fourth pixel the RED will be indicator and process goes on.

Figure 1 shows the block diagram for the proposed RGB steganography methodology based on pixel indicator.

Integrity check mechanism: Taking 4 LSBs out of all pixels in entire rows cum columns but the last in two, it operates EXOR on that of 1st two pixels from which the result is again EXORed with that of next entity and so on. The final end outcome is infixed in 4 LSBs in concluding pixel which is left as mentioned above. It is of help in receiving end knowing whether there is any amendment. This integrity check mechanism will help you to know the exact position of the pixel where the attacker has changed the pixel values. Figure 2 shows the Integrity Check Calculations of a sample 4×4 Stego image block.

HARDWARE IMPLEMENTATION

For implementing this RGB stego methodology in hardware, we used Cyclone II EP2C35F672C6 FPGA. The implementation uses 512 KB external SRAM to store both the cover and stego output's pixel values. To display cover image plus stego image, we use VGA interface output ports attached to FPGA. The FPGA board contains 16-pin D-SUB connector pro VGA yield. It also consists of toggle switches in which few of them are used for various conditions like reset, authorization input. To make sure that only particular users can use the board the

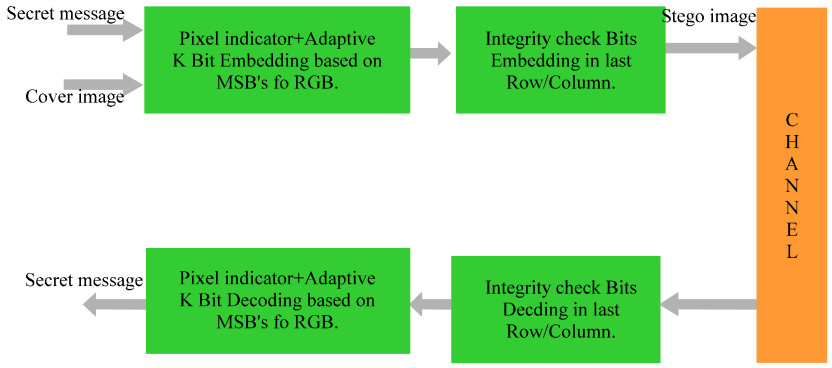

Fig. 1: Adaptive RGB steganography methodology block diagram

147	142	167	193
155	138	127	145
205	151	135	137
188	135	169	185

Stego image

147	142	167	202
155	138	127	158
205	151	135	141
188	135	169	185

Row wise operation

147	147	167	193
155	138	127	145
205	151	135	137
181	131	175	185

Column wise operation

147	147	167	202
155	138	127	158
205	151	135	141
181	131	175	176

Final element operation

Fig. 2: Integrity Check Calculations of a sample 4×4 Stego image block

inputs are taken only from 4 of those toggle switches which is the key to use the board. This makes the users authenticated. The design operates under 50 MHZ clock which can be enhanced with PLLs.

Architecture: Initially the message and cover pixels get accumulated in external SRAM memory of FPGA where each pixel value is addressable.

Reading from and writing to the SRAM memory is done with the signals namely Write Enable (WE), Output Enable (OE), Chip Enable (CE), LBO and UBO. The each cover image pixel consists of 24 bits, 8 bits each from R, G and B. Pixel values belonging to cover image are sent to logic for data embedding block. Based on the requirement for this block the message bits will be sent. If the logic demands that message bits should be embedded then new pixel values will be taken as output which is nothing but stego image. This stego upshot gets hoarded in third SRAM section. This hardware stego architecture with Cyclone II FPGA is shown in Fig. 3.

Block diagram in Fig. 4 shows the logic for data embedding in method I. Here, each pixel value of R, G plus B is stored into separate registers. MSB values of all the planes for a particular pixel are added with '1'. Resultant value from the adder and the message to be encoded are sent to embedding deciding block. Here, R plane is taken as indicator channel and G, B planes are taken as data channels. The value of two LSBs of R plane are compared with '00', '01', '10', '11'. If the value of two LSBs be '00'-no burying will be done in data channels; '01'-B plane is the data means and no embedding will be done in G plane. If it is '10' then G plane is the data channel and no embedding will be done in B plane. If it is '11' then both the data channels are embedded with message. The output from each comparator is given to their respective multiplexer as selection line. If the output from the comparator is '0' then don't care will be executed. If it is '1' then the mentioned bits will be LSB substituted with the message bits in the embedding deciding block. The output from this block will be a stego pixel with message embedded in it.

SYNTHESIS REPORT

Altera's Quartus II Design software Version 7.2 web edition has been used for hardware implementation. From the synthesis reports listed in Table 1, it is clear that when the images are stored in external memory, the hardware consumption is very less even for performing steganography on a 256×256×3 RGB image. At the same

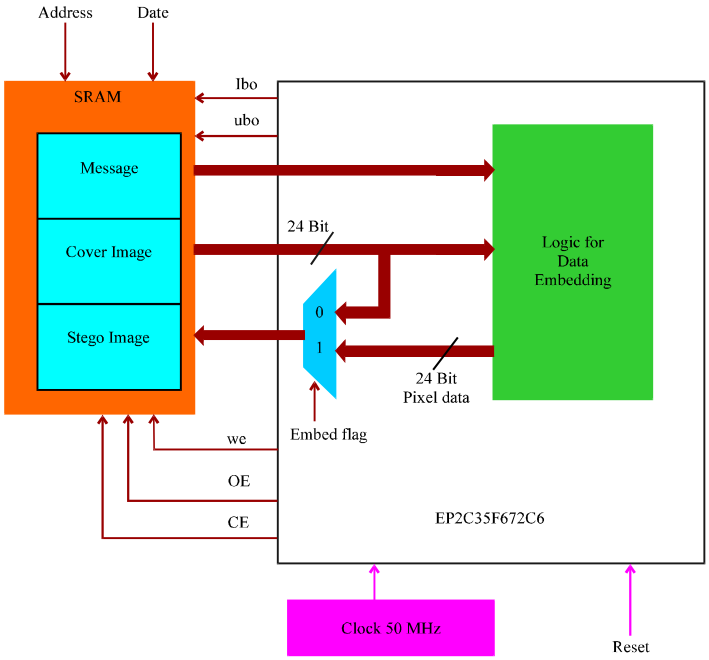

Fig. 3: FPGA based RGB steganography Architecture

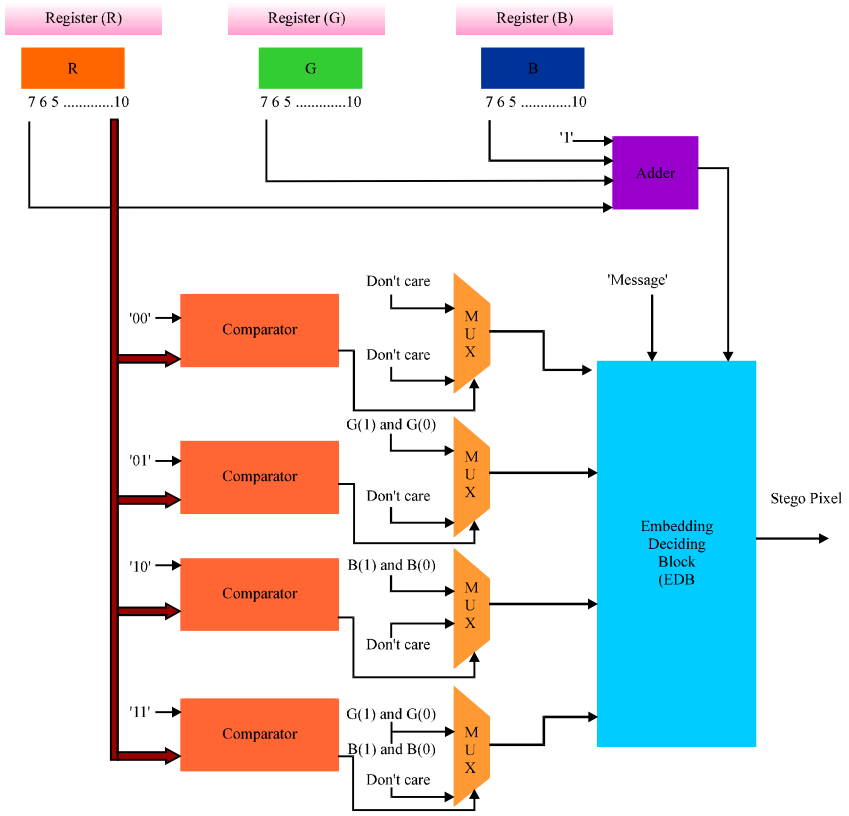

Fig. 4: Logic for data embedding in Method I

Table 1: Synthesis reports for stego algorithm with various cover images stored in external static RAM

Image size	Utilized logic elements	Utilized SRAM memory locations
16×16	142	384
32×32	153	1536
64×64	161	6144
128×128	167	24576
256×256	182	98304

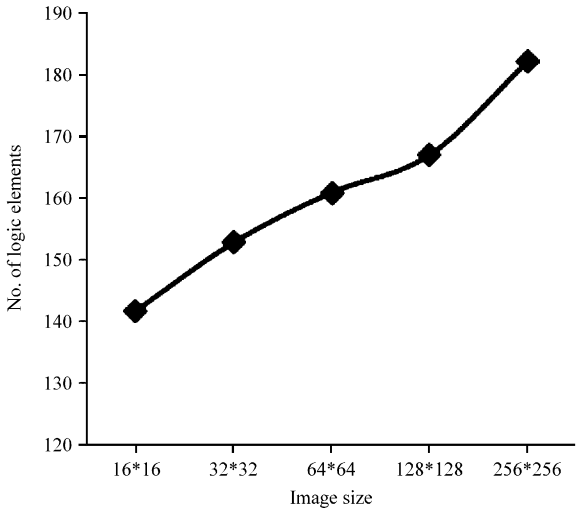

Fig. 5: Plot showing no. of logic elements utilized by various image sizes

time, when Block RAM is used for storage, more hardware is needed. The time taken for information hiding is analysed in detail in the next sub section where the dependency between message size and information hiding time can be studied. Figure 5 shows a plot for the relationship between Image Size and Logic Elements Utilization.

- FPGA Used: EP2C35F672C6
- Total Logic Elements: 33,216
- Total Registers: 33,216
- Total Pins: 475
- Total Memory Bits (External SRAM)
- 4194304 Bits
- Image Type: RGB image

Time taken to embed the secret information in cover image: This part discusses about the total time taken for hiding message in various size cover images. It has been inferred that the total time to hide message using the proposed methods in a 256×256 RGB image is 5.898 milli seconds. So an approximate 294900 clock cycles of 50 MHZ are needed to complete the embedding process. Figure 6 shows the timing analysis graph, which is an exponential curve

Table 2: Time taken to embed secret message in cover image and to store in External SRAM

RGB image size	Message embedding time in hardware in seconds
16×16	0.000023
32×32	0.000092
64×64	0.000368
128×128	0.001474
256×256	0.005898

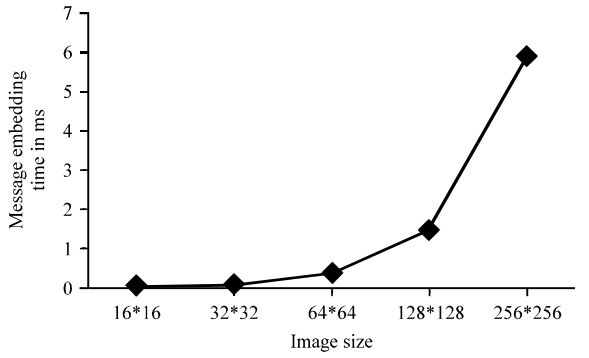

Fig. 6: Timing analysis

for different n times 16x16 RGB image pixels where n_{1,2,3,4}. The time consumption for message embedding in various size RGB images is shown in Table 2. Figure 6 shows the exponential curve representing the relationship between RGB Image sizes and Message embedding time.

DISPLAY SYSTEM DEVELOPMENT

So as to display the original image along with the end result, we use LCD monitor. The FPGA board has inbuilt 16-pin D-SUB connector meant for VGA (Video Graphics Adapter) o/p. Its synchronizing signals are lent from Cyclone II FPGA, ADV7123 triple 10-bit speedy DAC to generate analog signals (R, G, B) supporting pixel resolution of up to 1600×1200, on 100 MHZ. The image here is displayed for 640×480 resolution at 60 Hz refreshing rate. To display the image, internal clock frequency of 50 MHZ is divided to 25 MHZ. After embedding process is over and when the stego image is ready, a display program displays the cover image and stego image in the LCD monitor. This creates an option to choose the cover image for a specific algorithm and view the image without the use of software executable files only to display the image. A GUI developed on NI LabVIEW calculates the MSE, PSNR, Total Embedding capacity and bits per pixel for the RGB stego image. Figure 7 shows the cover and stego images display on the LCD monitor for proposed method-I.

Fig. 7: Display showing cover image and stego images of Method-I in LCD monitor

RESULTS AND DISCUSSION

In order to evaluate an algorithm we need some parameters to compare. In this case, we have taken the following four parameters. They are MSE, PSNR, Total Embedding Capacity, Bits per pixel (BPP).

MSE: This gives average value squares of variation amid the pixel values in cover and embedded stego. It tells us how much the latter deviates from the former. MSE of various images has been calculated and tabulated. MSE is premeditated by following expression:

$$MSE = \frac{1}{MN}\sum_{X=1}^{M}\sum_{Y=1}^{N}\left(S_{XY} - C_{XY}\right)^2$$

Here, M and N: pixels of horizontal and vertical facet, respectively, C_{XY} is original image pixel whereas S_{XY} is stego pixel.

PSNR: PSNR, a quality measure, is measured in terms of dB by:

$$PSNR = 10\log_{10}\left(\frac{c_{max}^2}{MSE}\right)$$

Here, C_{max} is maximum pixel intensity value. For an 8 bit colour image the C_{max} value is equal to 255.

Covers House, F-16, Boat and Tree of size 256×256×3 are in Fig. 8 a-d. Ensuing stego outputs are shown in Fig. 9 a-d. Table 3 shows the MSE, PSNR, Embedding capacity and Bits Per Pixel details for the four test images taken for data embedding using Method I. Table 4 shows the error metrics obtained with Method II implementation and Table 5 shows the error metrics for

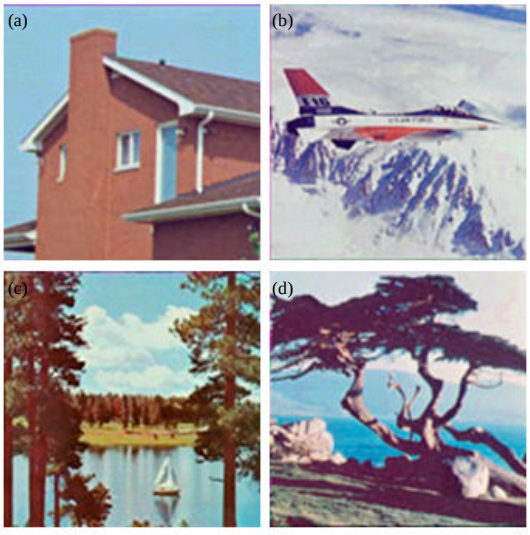

Fig. 8(a-d): Cover Images (a) House (b) F-16 (c) Boat and (d) Tree

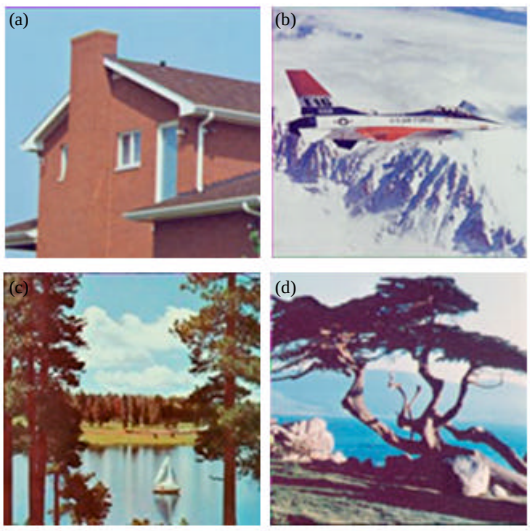

Fig. 9(a-d): Stego Images (a) House (b) F-16 (c) Boat and (d) Tree

Method III. Tentative results are tabled I, II, III, for Method I, II, III in that order. The algorithm is tested in 16×16, 32×32, 64×64, 128×128 and 256×256 size RGB cover images.

Because of the external SRAM used in this approach for storing the cover as well as stego images, the Cyclone II FPGA Logical elements consumption is very minimal for data embedding in even 256×256 image, which is 182 shown in Table 1. The linearity between Image size and number of logic elements used for stego

Table 3: Error metric and embedding capacity details for House, F-16, Boat and Tree stego images using Method-I based embedding technique

Image	Image plane	MSE	PSNR	Embedding Capacity	BPP
House	Red	0.0537	60.8302	167433	2.5548
	Green	2.4615	44.2188		
	Blue	3.8658	42.2585		
F-16	Red	0.0550	60.7253	233932	3.5695
	Green	6.0565	40.3086		
	Blue	6.0618	40.3048		
Boat	Red	0.0558	60.6659	157979	2.4105
	Green	3.1187	43.1911		
	Blue	3.2717	42.9831		
Tree	Red	0.0588	60.4387	162945	2.4863
	Green	3.4749	42.7213		
	Blue	3.4024	42.8129		

Table 4: Error metric and embedding capacity details for House, F-16, Boat and Tree stego images using Method-II based embedding technique

Image	Image Plane	MSE	PSNR	Embedding Capacity	BPP
House	Red	3.3491	42.8815	167433	2.5548
	Green	0.0518	60.9864		
	Blue	3.8434	42.2836		
F-16	Red	5.8916	40.4284	233932	3.5695
	Green	0.0548	60.7430		
	Blue	5.9585	40.3794		
Boat	Red	3.0993	43.2182	157979	2.4105
	Green	0.0528	60.9078		
	Blue	3.2759	42.9775		
Tree	Red	3.4222	42.7877	162945	2.4863
	Green	0.0527	60.9111		
	Blue	3.4410	42.7640		

Table 5: Error metric and embedding capacity details for House, F-16, Boat and Tree stego images using Method-III based embedding technique

Image	Image Plane	MSE	PSNR	Embedding Capacity	BPP
House	Red	2.3140	44.4872	165749	2.5291
	Green	1.5641	46.1882		
	Blue	2.6466	43.9038		
F-16	Red	3.9107	42.2082	230933	3.5237
	Green	4.0476	42.0589		
	Blue	3.9696	42.1434		
Boat	Red	2.0759	44.9587	156555	2.3888
	Green	2.0552	45.0022		
	Blue	2.2003	44.7060		
Tree	Red	2.2352	44.6377	160525	2.4494
	Green	2.3083	44.4978		
	Blue	2.3045	44.5051		

architecture can be understood more from Fig 5. As per the Table 2, which displays the message embedding time for various size images, an approximate four time increase in embedding time is observed between successive image sizes considered here. For example, an embedding time of 92 microseconds is needed for 32×32 image which is four times higher than 23 microseconds taken for 16×16 image. Similarly the embedding time of 256×256 image is approximately four times greater than 128×128 image. The time consumption exponential plot is at its peak for 256×256 RGB image which is showcased in Fig 6.

From the Table 3, 4 and 5, the following is inferred: In method 1 F-16 cover image possess elevated embedding capacity along with BPP (3.5695). Less embedding capability but with appreciable PSNR is witnessed as for as Boat is concerned. Method II and I look alike; when likened, first one offers flexibility. Method III promises enhanced imperceptibility in addition to embedding capacity. Of three, third is beneficial. Complexity for one bit embedding is 2^128*3*2*4/3*(8+8/3+8/3+8) which is higher than (Padmaa *et al*., 2011; Amirtharajan *et al*., 2011) which is 2^64*7*0.5*2*2.

CONCLUSION

Here, advised a dedicated hardware implementation of algorithm that randomises the data and embeds in the cover image providing security. The block diagram gives the hardware architecture and detail description of the logic for embedding data. MSB's of all the planes denote the number of bits in the channel plane thus providing more randomness of embedded data. The pixel indicator method provides further stochasticity of secret message. To furnish further authenticity, a key is provided so that only authorised users can perform the embedding and decoding in hardware. From all the three methods, it is noted that method III surmounts other methods with high randomness of embedding data.

ACKNOWLEDGMENTS

The authors wish to thank DRDO, New Delhi for supporting financially (ERIP/ER/1003836/M/01/1230). They also thank SASTRA University, Thanjavur as well for infrastructural support for this study. The Authors also appreciate Vivek Krishna Student ECE Department/ SEEE SASTRA University for his time, linguistic and technical support.

REFERENCES

Amirtharajan, R. and R.J.B. Balaguru, 2009. Tri-layer stego for enhanced security-a keyless random approach. Proceedings of the IEEE International Conference on Internet Multimedia Services Architecture and Applications, December 9-11, 2009, Bangalore, India, pp: 1-6.

Amirtharajan, R., D. Adharsh, V. Vignesh and R.J.B. Balaguru, 2010a. PVD blend with pixel indicator-OPAP composite for high fidelity steganography. Int. J. Comput. Appl., 7: 31-37.

Amirtharajan, R., S.K. Behera, M.A. Swarup, K.M. Ashfaaq and J.B.B. Rayappan, 2010b. Colour guided colour image steganography. Universal J. Comput. Sci. Eng. Technol., 1: 16-23.

Amirtharajan, R., R.R. Subrahmanyam, P.J.S. Prabhakar, R. Kavitha and J.B.B. Rayappan, 2011. MSB over hides LSB: A dark communication with integrity. Proceedings of the IEEE 5th International Conference on Internet Multimedia Systems Architecture and Application, December 12-14, 2011, Bangalore, Karnataka, India pp: 1-6.

Amirtharajan, R. and J.B.B. Rayappan, 2012a. An intelligent chaotic embedding approach to enhance stego-image quality. Inform. Sci., 193: 115-124.

Amirtharajan, R. and J.B.B. Rayappan, 2012b. Brownian motion of binary and gray-binary and gray bits in image for stego. J. Applied Sci., 12: 428-439.

Amirtharajan, R. and J.B.B. Rayappan, 2012c. Inverted pattern in inverted time domain for icon steganography. Inform. Technol. J., 11: 587-595.

Amirtharajan, R. and J.B.B. Rayappan, 2012d. Pixel authorized by pixel to trace with SFC on image to sabotage data mugger: A comparative study on PI stego. Res. J. Inform. Technol., 4:: 124-139.

Amirtharajan, R., J. Qin and J.B.B. Rayappan, 2012 Random image steganography and steganalysis: Present status and future directions. Inform. Technol. J., 11: 566-576.

Bender, W., D. Gruhl, N. Morimoto and A. Lu, 1996. Techniques for data hiding. IBM Syst. J., 35: 313-336.

Chan, C.K. and L.M. Cheng, 2004. Hiding data in images by simple LSB substitution. J. Pattern Recognit. Soc., 37: 469-474.

Cheddad, A., J. Condell, K. Curran and P.M. Kevitt, 2010. Digital image steganography: Survey and analysis of current methods. Signal Process., 90: 727-752.

Gomez-Hernandez, E., C. Feregrino-Uribe and R. Cumplido, 2008. FPGA hardware architecture of the steganographic context technique. Proceedings of the 18th International Conference on Electronics, Communications and Computers, March 3-5, 2008, Puebla, pp: 123-128.

Gutub, A., M. Ankeer, M. Abu-Ghalioun, A. Shaheen and A. Alvi, 2008. Pixel indicator high capacity technique for RGB image based steganography. Proceedings of the 5th IEEE International Workshop on Signal Processing and its Applications, March 18-20, 2008, Sharjah, UAE.

Marvel, L.M., C.G. Jr. Boncelet and C.T. Retter, 1999. Spread spectrum image steganography. IEEE Trans. Image Process., 8: 1075-1083.

Padmaa, M., Y. Venkataramani and R. Amirtharajan, 2011. Stego on 2^n: 1 Platform for users and embedding. Inform. Technol. J., 10: 1896-1907.

Petitcolas, F.A.P., R.J. Anderson and M.G. Kuhn, 1999. Information hiding-a survey. Proc. IEEE, 87: 1062-1078.

Provos, N. and P. Honeyman, 2003. Hide and seek: An introduction to steganography. IEEE Secur. Privacy, 1: 32-44.

Rajagopalan, S. and H.N. Upadhyay, 2011. Stego system on chip with LFSR based information hiding approach. Int. J. Comput. Appl., 18: 24-31.

Rajagopalan, S., R. Amirtharajan, H.N. Upadhyay and J.B.B. Rayappan, 2012a. Survey and analysis of hardware cryptographic and steganographic systems on FPGA. J. Applied Sci., 12: 201-210.

Rajagopalan, S., S. Janakiraman, H.N. Upadhyay and K. Thenmozhi, 2012b. Hide and Seek in Silicon-Performance Analysis of Quad Block Equisum Hardware Steganographic Systems. Procedia Eng., 30: 806-813.

Schneier, B., 2007. Applied Cryptography: Protocols, Algorithm and Source Code in C. 2nd Edn., Wiley, India.

Stefan, K. and A. Fabin, 2000. Information Hiding Techniques for Steganography and Digital Watermarking. Artech House, London, UK.

Thanikaiselvan, V., S. Kumar, N. Neelima and R. Amirtharajan, 2011. Data battle on the digital field between horse cavalry and interlopers. J. Theor. Applied Inform. Technol., 29: 85-91.

Thenmozhi, K., P. Praveenkumar, R. Amirtharajan, V. Prithiviraj, R. Varadarajan and J.B.B. Rayappan, 2012. OFDM+CDMA+Stego = Secure Communication: A Review. Res. J. Inform. Technol., 4: 31-46.

Thien, C.C. and J.C. Lin, 2003. A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function. Pattern Recog., 36: 2875-2881.

Zaidan, B.B., A.A. Zaidan, A.K. Al-Frajat and H.A. Jalab, 2010. On the differences between hiding information and cryptography techniques: An overview. J. Applied Sci., 10: 1650-1655.

Zanganeh, O. and S. Ibrahim, 2011. Adaptive image steganography based on optimal embedding and robust against chi-square attack. Inform. Technol. J., 10: 1285-1294.

Zhu, J., R.D. Wang, J. Li and D.Q. Yan, 2011. A huffman coding section-based steganography for AAC audio. Inform. Technol. J., 10: 1983-1988.