Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 5 (2): 160-170, 2013
ISSN 1815-7432 / DOI: 10.3923/rjit.2013.160.170

Captivating CODEC Stego (CCS): A Cover on Camouflage

Siva Janakiraman, Sundararaman Rajagopalan, K. Thenmozhi, Har Narayan Upadhyay, Janani Ramanathan, Swetha Varadarajan, J.B.B. Rayappan and Rengarajan Amirtharajan
School of Electrical and Electronics Engineering, SASTRA University, 613401, India

Corresponding Author: Siva Janakiraman, School of Electrical and Electronics Engineering, SASTRA University, 613401, India

ABSTRACT

One year's seeds, seven year's weeds' is precisely pointed out by the proverb to highlight the importance of doing things, not too late. Even though, this Anglo Saxon work ethic is directed towards the must attitude for workmanship, the importance of this proverb for information security is phenomenal. Secure information sharing needs novel ideas and techniques to impede the incoming threats of secret drop outs. Steganography, an information hiding approach has seen a variety of flavours in terms of domains, algorithms, platforms etc. We present an information hiding approach on hardware platform where Field Programmable Gate Array (FPGA) works out a spatial domain steganographic module with fixed k-bit embedding and adaptive picture element position p where -1<P<4 and gray scale change factor C falling between 0<C<3. This stego algorithm on programmable logic hides the secret message bits in the cover image stored in External SRAM through a convolution encoder which employs a triple XOR logic for encoded message creation to enhance the imperceptibility with higher embedding capacity.

Key words: Hardware steganography, convolution encoder, image steganography, spatial domain

INTRODUCTION

In today's scenario of fast paced communication and technology, where the exchange of information over the internet in the physical digital forms like video, audio image files etc has (Cheddad *et al*., 2010) ease of access to and is on the rise and thus, there is a dire need for high security (Petitcolas *et al*., 1999; Bender *et al*., 1996) of the scrutinized data. Hence, in this modern era, to cater to the impediment of authorization and potential hacking, Steganography (Provos and Honeyman, 2003; Zaidan *et al*., 2010) has gained a predominant stand in the realm of information security which revolves around factors like confidentiality, integrity. Steganography (Rabah, 2004) is a state-of-the-art terminology for hiding the clandestine information in a cover without any visible amendments to the cover. While Cryptography (Xia *et al*., 2009; Janakiraman *et al*., 2012c) encloses a key and scrambled data in its transmission, Steganography makes the very existence of the covert data is unperceivable. Of late, many techniques in steganography have been developed.

However, image steganography (Hmood *et al*., 2010; Amirtharajan and Rayappan 2012a, b, c, d) can be categorized into two namely: transform domain and spatial domain. While in the spatial domain, the clandestine data is hidden in the cover pixel using numerous methodologies like the least significant bit; pixel value indicator and pixel differencing, in transform domain technique, the cover pixel is primarily altered into other domains like the wavelet (Thanikaiselvan *et al*.,

2011b) or frequency domain (Kumar *et al.*, 2011) in which the covert data is then embedded in the corresponding cover pixel. However, spatial domain technique's (Amirtharajan and Rayappan, 2011) utility lies mainly in its improved embedding capacity and payload as compared to the transform domain's high robustness.

On a further note, steganography employed in spatial and transform domain could be carried out in raster or random scan (Amirtharajan and Balaguru, 2009; Amirtharajan and Rayappan, 2011; Amirtharajan *et al.*, 2012). Even though software methodologies may work on both spatial (Padmaa *et al.*, 2011) as well as transform domain, Steganography proposes multifarious data embedding techniques especially on hardware platform (Rajagopalan *et al.*, 2012b) comprising of various advantages such as high speed embedding, embedding capacity, imperceptibility, specific hardware dependency etc. (Janakiraman *et al.*, 2012b). Furthermore, a broad classification on image steganography can be performed on the basis of the methodologies implemented viz. distortion, spread spectrum (Thenmozhi *et al.*, 2012), Transform (Thanikaiselvan *et al.*, 2011a), substitution (Thanikaiselvan *et al.*, 2011b), statistical (Qin *et al.*, 2009) and cover generation methods.

RELATED WORKS

A secure spatial domain based steganography technique has been proposed by Ali *et al.* (2011) which cater to embed two least significant bits of covert information in a cover pixel, by transforming one bit from a one bit plane into another. However, the message which is embedded in a pixel is located in the second least significant bit or fourth least significant bit apart from the least significant bit (Mielikainen, 2006). The alterations so implemented in the cover pixel yields in a new stego pixel. In order to extract the clandestine data from the stego pixel, the author has premeditated upon a (3, 1) convolution decoder.

A mathematical formulae approach for embedding and retrieving by implementing simple LSB (least significant bit) has been propounded (Chan and Cheng, 2004). Further, it is also ascertained that there is a fair chance in reducing the Mean squared error (MSE) from 2^k to 2^{k-1}.

$St = Cv - \text{mod}(Cv, 2^k) + D$ where St is stego object (image), Cv is the cover object (image) and D is the decimal equivalent of the secret data to be hidden. Here, the extraction is also minimal: $C = \text{mod}(S, 2k)$.

LFSR based hardware steganography proposed (Sundararaman and Upadhyay, 2011) utilizes various LFSRs as address generators to perform secret concealment on image. A quad block guided steganography on chip architecture implemented on Cyclone II FPGA with a detailed analysis has been propounded (Rajagopalan *et al.*, 2012a). Further, the RISC processor chosen for their execution is the LPC 2136 which encloses in its core the ARM7TDMI-S having 32KB of on-chip SRAM which suffices amasses a maximum image size 128×128.

In image steganography based methodology (Janakiraman *et al.*, 2011) smart bit manipulation of embedding the covert pixels in the lower nibble of the cover pixel has been proposed which involves the convolution encoder to embed three bits by altering utmost two bits of the cover pixel and thus, enhancing the robustness of three bit embedding. In yet another similar approach by Janakiraman *et al.* (2012a), efforts were taken using the modified convolution encoder to embed 4 bits of information in each pixel with a constrain to adjust not more than two bits in cover pixel to produce the stego pixel.

In present study, a secure and stable method has been proposed on the basis of a block based image steganography algorithm with its implementation in the Cyclone II EP2C20F484C7 FPGA

in order to sensitize the creditability and performance of the stego based system. This image steganography (Xia *et al*., 2009) architecture inherits embedding of data block by block (Thien and Lin, 2003; Lu *et al*., 2009; Rajagopalan *et al*., 2012a) in a cover image divided into 4×4 blocks that is stored in external SRAM attached to reconfigurable hardware. A few objectives of this research are to perform extensive timing analysis, hardware consumption for data embedding and error metrics computation after implementation. This proposed methodology makes an optimistic effort to emphasize on the uniqueness to embed three data bits concurrently in two cover pixels and thereby, improving the randomness while embedding, by implementing convolution encoder and decoder for the process of embedding and retrieving. Hence, an aggressor would take long time to premeditate upon method to break the system.

PROPOSED METHODOLOGY

The main purpose of this paper is to improve the previous stated works in a way to provide an increased working efficiency by utilizing the advantage of con-current execution of embedding. This also aims in providing an alternative to the above mentioned method of sequential encoding to block encoding. The two major aspects involved in this study are:

- Embedding process based on the convolution encoder
- The hardware implementation carried out using the Cyclone II FPGA EP2C20F484C7. The LSB substitution done by using the convolution encoder allows for a maximum change in the lower nibble of the cover pixel by 2 bit and enabling the capacity of 3 bit embedding into the lower nibble. This method involves a change in either one of the 4 lower bits or a combination of 2 of the lower bits. The circuit of the convolution encoder is shown in Fig. 1

The circuit in Fig 1 works in a way so as to produce a new Stego image by altering the pixel value of the cover image. The input of the encoder is the lower nibble of the cover image (c4 c3 c2 c1) which produces an output (n1 n2 n3) through XOR operation as depicted in the figure.

The secret message bits (m1 m2 m3) to be embedded are compared with the decoded output and the decoder performs the changes in the cover pixel as shown below in the Table 1. Our approach travels on two variables namely P and C.

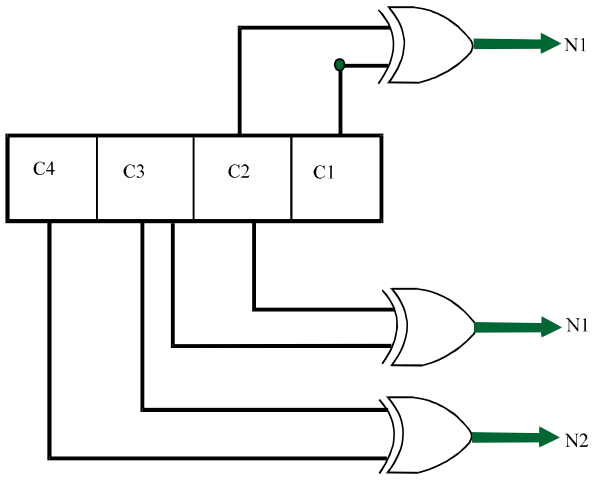

Fig. 1: Convolution encoder (3, 1)

Table 1: Sample encoder output Vs greyscale change in cover pixel

Bit(s) to change	Decoded output (N1 N2 N3)	Message bits (mL m2 m3)	Bit(s) to change	Decoded output (N1 N2 N3)	Message bits (m1 m2 m3)
Bit 1(c1)	000	100	Bits 1,4 (c1c4)	000	110
	001	101		001	111
	010	110		010	100
	111	111		011	101
	100	000		100	010
	101	001		101	011
	110	010		110	000
	111	011		111	001
Bit 2 (c2)	000	101	Bit 3 (c3)	000	011
	001	100		001	010
	010	111		010	001
	011	110		011	000
	100	001		100	111
	101	000		101	110
	110	011		110	101
	111	010		111	100
Bits 1,3(c1c3)	000	111	Bit 4 (c4)	000	010
	001	110		001	011
	010	101		010	000
	011	100		011	001
	100	011		100	110
	101	010		101	111
	110	001		110	100
	111	000		111	101
Bits 1,2(c1c2)	000	011	Bits 3,4 (c3c4)	100	101
	001	000		101	100
	010	011		110	111
	011	010		111	110

No change when (N1 N2 N3) = mL m2 m3

Let C_n = [b7 b6 b5 b4 b3 b2 b1 b0], Where C_n is the nth cover pixel and b7...b0 indicating the bits of cover pixel. The Picture element position variable P, which is adaptive in nature hes in the range -1<P<4, where 0 represents the LSB of a cover pixel (b0) and 3 representing the 3rd bit position from LSB (b3). The gray scale change factor C, which is given by 0<C<3, denoting the number of bits altered in the lower nibble of the cover pixel so as to generate stego pixel.

The retrieval procedure possesses a great challenge to detect the secret data. This is attributed due to the complexity involved in the decoder circuit as described. The recovery of the secret message is done by introducing a convolution decoder without a decision device at the end and simply obtaining the result. A grayscale image (128×128) is taken as a cover image. Thus, it has (128×128×8) bits that can be grouped into 1024 blocks of 4×4 matrixes that contain 16 bytes each. The pixels of the cover image thus arranged in blocks are arranged consecutively from left to right in every row. In this way, the message embedding is ensured in a block-wise manner rather than sequentially.

FPGA IMPLEMENTATION

The proposed convolution encoder based spatial domain stego system has been implemented on Cyclone II FPGA. Initially the gray scale cover Image of size 128×128 is stored in external high

speed asynchronous CMOS static RAM IS61LV25616 which has 2^{18} locations with each location capable of storing two bytes of data. A total of 16384 locations are needed to store both cover and stego images of size 128×128. Starting from location 1 to 8192 the cover image has been stored. This encoder based embedding algorithm has been implemented using a clock frequency of 100 MHZ, which is derived through PLL with a multiplication factor of 2 from the source of 50 MHZ external clock connected to FPGA. The hardware architecture is shown in Fig. 2.

The embedding process follows a continuous state machine, with five states per cycle. A state machine cycle performs the operations needed to read the cover image pixels in a SRAM location and to store back the stego pixels in external SRAM. The state machine cycle has six states namely Address, Secret, Triple XOR, Encode and Write. The state machine is shown in Fig 3. The description of each state is as follows:

State 1: Address: When storing the cover image on SRAM, the cover image has been stored in blocks. This means that the 128×128 cover image divided into 1024 4×4 blocks has been

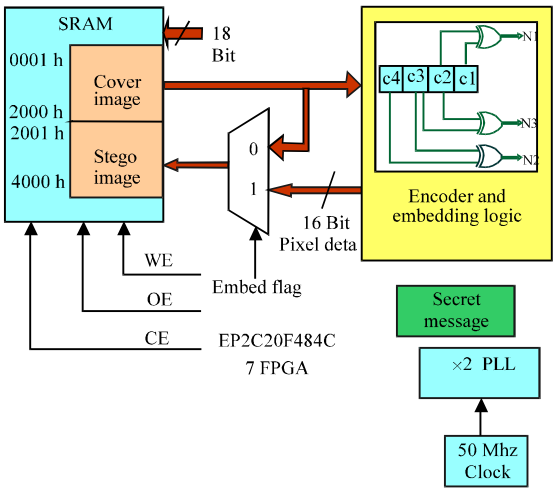

Fig. 2: Hardware architecture for CCS algorithm in FPGA

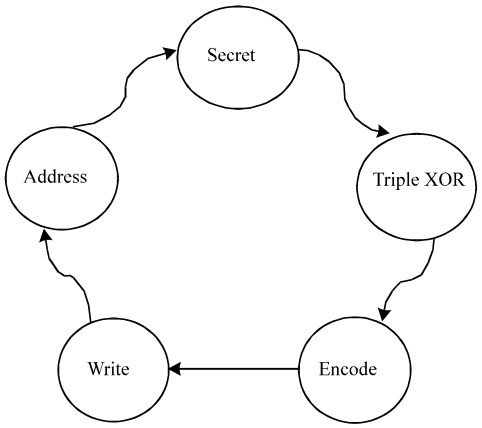

Fig. 3: State Machine structure for CCS on cyclone II FPGA

stored block wise continuously. For storing one 4×4 block 8 locations of SRAM is needed. The blocks are stored row wise (i.e.) first row 64 4×4 blocks find their place first and followed by next row 4×4 blocks and so on. The Address of SRAM is 18 bit data which is incremented with a counter starting from 1 to 8192 during each state machine cycle. This address state takes one clock cycle during which the Write Enable (WE) will be held at Logic High in order to read the two pixels as data bits width is 16

State 2: Secret: This state needs one clock cycle to select the embedding bits stored in FPGA registers. A total of 3 bits are required/pixel. Since we attempt to embed data in two pixels parallel, two 3-bit registers have been used to select and store the secret bits to be embedded. At the same time the data lines will be kept in high impedance state so that the two pixels in specified location will be transferred to 8-bit registers in FPGA namely a1 and a2. The pixels were transferred in such a way that a1 <= Lower Byte (location n) and a2 <= Upper Byte (location n)

State 3: Triple XOR state: During this third state the convolution encoder logic has been implemented on the two lower nibbles of the two pixels read from SRAM. The following bit wise operations will be performed in one clock cycle:

$$n1(1)<=a1(1)\oplus a1 \quad (2)$$

$$n2(1)<=a2(1)\oplus a2 \quad (2)$$

$$n1(3)<=a1(2)\oplus a1 \quad (3)$$

$$n2(3)<=a2(2)\oplus a2 \quad (3)$$

$$n1(2)<=a1(3)\oplus a1 \quad (4)$$

$$n2(2)<=a2(3)\oplus a2 \quad (4)$$

where, n1 (1), n1 (2) and n1 (3) generate the encoded bits using lower byte and n2 (1), n2 (2) and n2 (3) represent the encoded output using upper byte.

State 4: Encode: The output of the convolution circuit is compared with the three bit message input to be embedded on conditional basis in this state. It works in such a way that there will be an alteration in the lower nibble of the cover pixel which when passed through the circuit yields the same three bit message data. The conditions are grouped on the basis of the similarity in the alteration of the lower nibble of the cover pixels resulting in the three bit message data. Also during this state the lower nibble and upper nibble of the two pixels will be joined and stored in separate registers

State 5: Write: The stego pixels present in the two registers will be stored in specific location of stego memory which starts from location number 8193 in SRAM pointed out by stego location identifier.Thus one state machine cycle results in embedding 3 bits each in two pixels of the cover image. After this state machine cycle which takes five clock cycles gets over, the embedding process goes back to first state again to embed the data in all the pixels/number of pixels according to the size of secret message

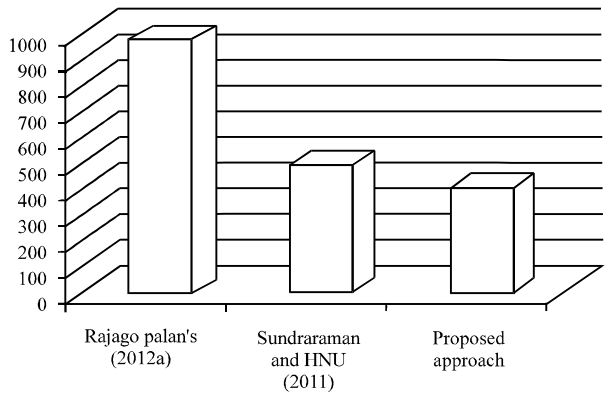

Fig. 4: Secret Embedding time in 128×128 grayscale images in various approaches

Timing analysis:

- FPGA Master clock frequency = 50 MHZ
- Stego process clock frequency = 100 MHZ (with PLL)
- Clock cycles required to embed secret bits in 1 SRAM location (2 cover pixels;6 message bits) =5 cycles of 100 MHZ
- Size of the image = 128×128 cover
- No of locations in SRAM = 8192
- Total number of clock cycles for hiding entire information = 8192×5 = 40960 clock cycles
- Total time required = 8192×5×(1/100MHz) = 409.6 microseconds

The time taken for embedding secret message (two bits/pixel) in a 128×128 grayscale image stored in External SRAM proposed in Rajagopalan *et al.* (2012b) is 983.04 micro seconds. As the time taken for embedding is not a function of the value of K (i.e.,) number of bits to be embedded in a pixel, same amount of time would have been consumed for 0<K<5. Our proposed method of hardware steganography on Cyclone II FPGA consumes 409.6 microseconds to hide information on the whole pixels of 128×128 grayscale image residing at SRAM, which is 573.44 microseconds less than earlier profound approach in Rajagopalan *et al.* (2012a). Also this approach surpassed the LFSR based information hiding approach (Sundararaman and Upadhyay, 2011) which resulted in 0.491 milliseconds (Approximately 80.4 microseconds less in proposed approach). The comparison is shown in the graph in Fig 4.

RESULTS

Synthesis report:

- **FPGA used:** EP2C20F484C7
- **Logic elements utilized:** 108 out of 18,752 (< 1%)
- **Logic registers used:** 82 out of 18,752 (< 1%)
- **Pins utilized:** 41 out of 315 (3%)
- **PLLs used:** 1 out of 4 (25%)

- **SRAM memory locations used:** 16384 Locations (8192 for cover Image and 8192 for stego image)

Error metrics:

- The error metrics i.e. the performance parameters to judge the quality of the Stego image are
- MSE(mean square error) calculated by the equation:

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

- PSNR(peak signal to noise ratio) calculated by the Equation:

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

The quality of the image and the PSNR values are directly proportional. Higher the value, improved quality can be obtained. The MSE and PSNR values obtained for the four different images using the LABVIEW code is tabulated in Table 2 for better understanding of the quality improvement in the proposed methodology.

Sample cover and Stego images:

- The cover images and the stego images are shown in Fig. 5 and 6. It is observed that the imperceptibility is enhanced as compared to the other methods. In addition to this, the robustness is improved due to extraction procedure which is not the simple LSB substitution

Table 2: Error metrics result

Image	MSE	PSNR (dB)
GANDHIJI	27.508	33.7362
PEPPE	27.4641	33.7431
USC* Test 5.2.10	26.9012	33.8331
USC* House	27.3018	33.7689

*USC: University of Southern California (Test Image)

Fig. 5(a-d): Cover image (a) Gandhiji, (b) Pepper, (c) USC 5.2.10 and (d) USC House

Fig. 6(a-d): Stego image (a) Gandhiji, (b) Pepper, (c) USC 5.2.10 and (d) USC House

CONCLUSION

A spatial domain hardware steganographic system on gray scale images has been proposed in this paper. Earlier works proposed in hardware architecture for stego systems on FPGA concentrated on traversal path, randomness with LFSR, shuffler etc.,. One of this work's main objectives are deciding the secret message to be embedded based on the secret bits itself using 4-bit convolution encoder (3,1). Also the proposed work parallely hides the secret bits in the two pixels stored in a SRAM location by accessing the two cover pixels in one clock cycle and storing back the stego pixels after modification and this concurrent hiding resulted in minimizing the total time taken to embed the secret information in entire cover image compared to the previous works which used techniques that operated on only one pixel per clock cycle. As varying the arrangement of XOR gates in convolution encoder and usage of LSBs of n-bit convolution encoder for the encoding purpose can find a place in camouflage, future work can be focussed on this direction. Also FPGA based image steganographic systems using various image processing operations needs attention.

ACKNOWLEDGMENT

The authors wish to express their sincere thanks to DRDO, New Delhi for their financial support (ERIP/ER/1003836/M/01/1230). They also wish to acknowledge SASTRA University, Thanjavur for extending infrastructural support to carry out the study.

REFERENCES

Ali, D., H. Aghaeinia and S.H. Seyedi, 2011. A more secure steganography method in spatial domain. Proceedings of the 2nd International Conference on Intelligent Systems on Modeling and Simulation, January 25-27, 2011, Kuala Lumpur, pp: 189-194.

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. 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. and R.J.B. Balaguru, 2011. Covered CDMA multi-user writing on spatially divided image. Proceedings of the Wireless ViTAE Conference, February 28-March 3, 2011, IEEE, Chennai, India, pp: 1-5.

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.

Hmood, A.K., B.B. Zaidan, A.A. Zaidan and H.A. Jalab, 2010. An overview on hiding information technique in images. J. Applied Sci., 10: 2094-2100.

Janakiraman, S., A.A. Mary, J. Chakravarthy, R. Amirtharajan, K. Thenmozhi, J. Bosco and B. Rayappan, 2011. Smart bit manipulation for K bit encoded hiding in K-1 pixel bits. Proceedings of the 3rd International Conference on Trendz in Information Sciences and Computing, December 8-9, 2011, IEEE.

Janakiraman, S., A.A. Mary, J. Chakravarthy, R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2012a. Pixel bit manipulation for encoded hiding-An inherent stego. Proceedings of the International Conference on Computer Communication and Informatics, January 10-12, 2012, IEEE Explore, USA., pp: 1-6.

Janakiraman, S., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2012b. Firmware for data security: A review. Res. J. Inform. Technol., 4: 61-72.

Janakiraman, S., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2012c. Pixel forefinger for gray in color: A layer by layer stego. Inform. Technol. J., 11: 9-19.

Kumar, P.P., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2011. Steg-OFDM blend for highly secure multi-user communication. Proceedings of the 2nd International Conference on Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology, February 28-March 3, 2011, IEEE, Chennai, India, pp: 1-5.

Lu, T.C., S.R. Liao, P.L. Chen, C.C.C. Chang and Z.H. Wang, 2009. Information hiding technology based on block-segmentation strategy. Proceedings of the ISECS International Colloquium on Computing on Communication, Control and Management, Volume 1, August 8-9, 2009, Sanya, pp: 500-506.

Mielikainen, J., 2006. LSB matching revisited. IEEE Signal Process. Lett., 13: 285-287.

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.

Qin, J., X. Sun, X. Xiang and Z. Xia, 2009. Steganalysis based on difference statistics for LSB matching steganography. Inform. Technol. J., 8: 1281-1286.

Rabah, K., 2004. Steganography. The art of hiding data. inform. Technol. J., 3: 245-269.

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.

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

Thanikaiselvan, V., P. Arulmozhivarman, R. Amirtharajan and J.B.B. Rayappan, 2011a. Wave (let) decide choosy pixel embedding for stego. Proceedings of the International Conference on Computer, Communication and Electrical Technology, March 18-19, 2011, India, pp: 157-162.

Thanikaiselvan, V., S. Kumar, N. Neelima and R. Amirtharajan, 2011b. 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.

Xia, Z., X. Sun, J. Qin and C. Niu, 2009. Feature selection for image steganalysis using hybrid genetic algorithm. Inform. Technol. J., 8: 811-820.

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.