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): 1992-1998, 2014
ISSN 1812-5638 / DOI: 10.3923/itj.2014.1992.1998

Stego on Song-an Amalgam of vi and FPGA for Hardware Info Hide

[1]Sundararaman Rajagopalan, [2]K. Pravallika, [2]R. Radha, [1]Har Narayan Upadhyay,
[1]J.B.B. Rayappan and [1]Rengarajan Amirtharajan
[1]Faculty, School of Electrical and Electronics Engineering, SASTRA University, Thanjavur-61340, India
[2]Tata Consultancy Services, Chennai, India

Abstract: The inventions and innovations in the multimedia communication have revolutionized the infotainment world. With the advancements in multimedia , there is a growing demand to use the image, audio, video and text ingredients for information protection. In the field of steganography, the cover to carry over the secret information has been an image in many of the reported works due to high payload carrying capacity. However audio steganography on hardware platform provides the user an option to use multicarrier steganography on common chip. Here we propose a method of steganography which has been implemented on Cyclone II FPGA EP2C20F484C7 which houses an architecture with LSB substitution of Huffman encoded secret message on select parts of digitized audio signal. The experimental results show that a SNR of 127.7 dB has been obtained for 100 bits payload with embedding process done on every 2000th memory location of 16-bit digitized audio signal stored in External SRAM.

Key words: Audio steganography, hardware steganography, information hiding, FPGA, labVIEW

INTRODUCTION

The famous Alice-Bob pair has triggered the minds of everyone to follow certain inevitable things for information sharing in a secret manner. While information security has crossed miles in the road map of security index, an indication of depth and intensity of the security, a huge responsibility to carry out secure communication has always been on the cards just like a poet said 'Miles to go before I sleep and miles to go before I sleep'. Schemes based on cryptography offer randomization in terms of scrambling the data so that the information is hard to decrypt (Zaidan *et al*., 2010). Techniques working on watermarking have been workhorses for copyright protection. But one way of bringing absolute security for the information to be conveyed or carried is information hiding i.e., steganography (Al-Azawi and Fadhil, 2010; Al-Frajat *et al*., 2010; Zanganeh and Ibrahim, 2011). Steganography, a secret cover data hiding methodology (Cheddad *et al*., 2010) has been reported both in software as well as hardware platforms (Rajagopalan *et al*., 2012a, b)in literature. The flexibility in choosing the algorithm or designing an algorithm for spatial (Janakiraman *et al*, 2012a, b; Amirtharajan *et al*., 2012; Amirtharajan and Rayappan, 2012a-d; Thanikaiselvan *et al*, 2011) or transform domain steganography (Amirtharajan and Rayappan, 2012d) and some other important and interesting features are remarkable and that makes steganography to be on top of secure information transmission.

A number of hiding algorithms with image as cover have been proposed in software (Thenmozhi *et al*., 2012) as well as hardware platforms. Simple LSB substitution, Pixel Value differencing (Wu and Tsai, 2003), Random steganographic algorithms, Block based steganographic algorithms, Image processing based algorithms have been reported in software platforms. An image steganography method involving optimum MSE selection by random row-wise processing to improve MSE by a factor of 1.8 and to boost PSNR by 0.25 dB has been discussed by Amirtharajan and Rayappan (2012c). FPGA and Processor based stego implementations also exist in the literature. Rajagopalan and Upadhyay (2011) reported LFSR based information hiding on grayscale image stored in FPGA. A quad block steganographic system on FPGA and ARM has been discussed by Rajagopalan *et al*. (2012a). Secret key based stego architecture on Spartan FPGA has been reported by Farouk and Saeb (2004). A hardware architecture of the ConText steganographic technique implemented in Cyclone II FPGA of Altera family is given by Gomez-Hernandez *et al*. (2008) and Amirtharajan *et al*. (2010) discuss a method on 2D image processing based hardware stego.

On the other side, most of the implementations which use audio as cover have been worked out in software arena (Zhu *et al*., 2011; Qiao, *et al*., 2010) analysed

Corresponding Author: Sundararaman Rajagopalan, Faculty, School of Electrical and Electronics Engineering, SASTRA University, Thanjavur-61340, India

embedding strength when audio is used as cover to hide the information (Gopalan and Shi, 2010; Gopalan, 2003) discusses about bit modification based audio steganography. A high capacity audio stego algorithm based on DWT has been proposed by Shahadi and Jidin (2011). LSB substitution based approach with kth-bit embedding on audio exists in the literature (Djebbar *et al*., 2011; Asad *et al*., 2011). Spectrum spreading based information hiding on audio is proposed by Skopin *et al*. (2010) and Zamani *et al*. (2009) propose robust Audio secret hiding based on genetic algorithm. Lifting wavelet Transform has been the decider of data embedding in audio signal which is proposed by Pooyan and Delforouzi (2007). Integer wavelet Transform based audio steganography has been discussed by Delforouzi and Pooyan (2007). Minimum error replacement to increase the payload on audio is discussed by Cvejic and Seppanen (2002).

In our proposed method Cyclone II FPGA based audio steganography is discussed. FPGA increases security in terms of many aspects like hardware dependency, bit stream etc. Memories present inside and outside the FPGA offer advantages to keep the stego and cover audio. We have attempted Huffman encoding based information concealment method on the cover audio of 5.2 sec stored in external SRAM of FPGA and an SNR of 127.7 dB has been obtained for a payload of 100 bits which is a good value. The rest of the paper discusses the modules involved in FPGA based audio steganography.

PROPOSED METHODOLOGY

The Entire methodology is divided into modules as discussed below:

- Digitizing the audio
- Conversion of secret message to bit pattern by Huffman encoding
- Storing digitized audio samples and secret bits in the SRAM of FPGA board
- Creation of stego audio file by embedding the secret message in, specific places of the cover using, LSB substitution technique
- Retrieving the embedded message and Stego audio by processing the stego file

The hardware implementation of the stego architecture includes storing of the cover and stego files in the SRAM of FPGA and implementation of LSB steganographic technique using VHDL (Very High Speed IC Hardware Descriptive Language). The proposed stego architecture has been implemented in Altera FPGA EP2C20F484C7. Figure 1 shows the block diagram of proposed approach.

AUDIO DIGITIZING MODULE

The proposed system deals with an audio file of 5.2 sec in wave format as cover. The audio signal is

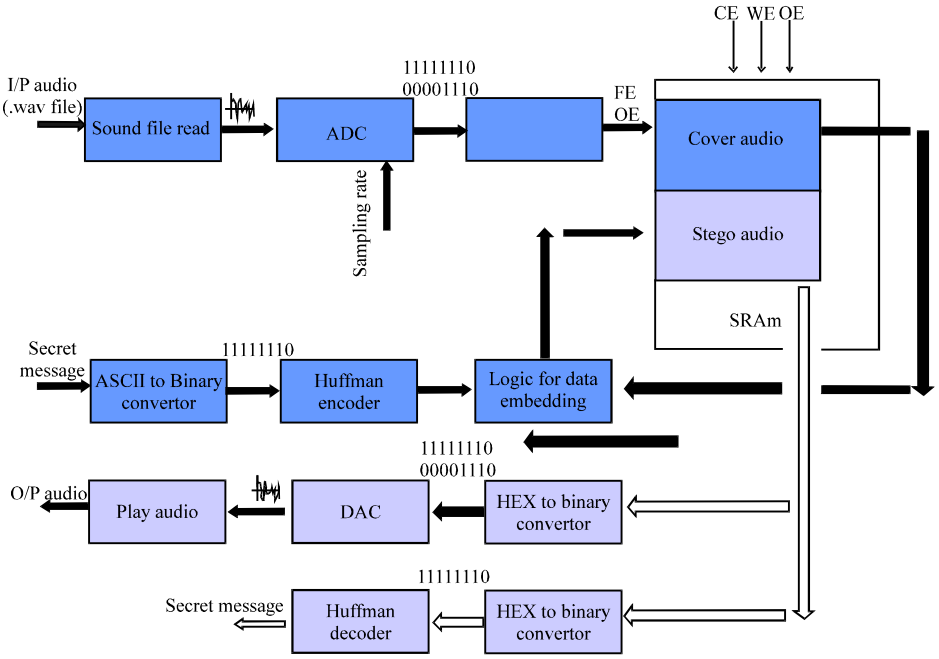

Fig. 1: Block diagram of audio steganography on FPGA

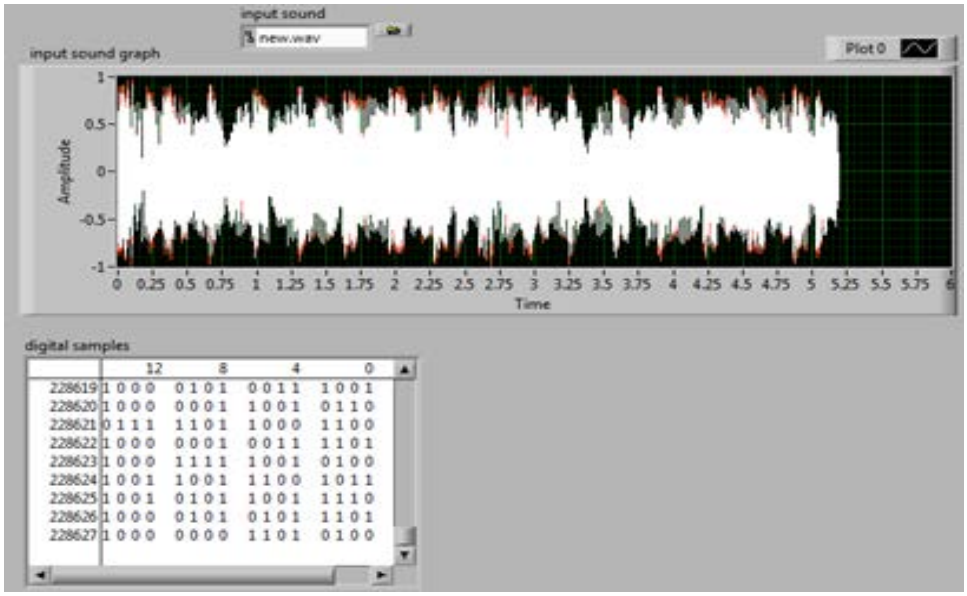

Fig. 2: Audio signal and its digital form from labVIEW

converted to digital form using LabVIEW based Virtual Instrumentation environment. This is taken as input to the system through sound file read.vi, through which the analog graph of the original sound is obtained. The analog data is then converted to digital data using DWDT analog to digital conversion.vi with 44100 Khz as sampling rate. A 226828 digital samples are obtained and these samples can be of 8/16/32 bits resolution. The sample values are then stored in text files which are then loaded into the SRAM. Figure 2 shows the audio digital conversion through LabVIEW.

SECRET MESSAGE ENCODING MODULE

The secret message is encoded using Huffman encoding. Huffman encoding is a simple compression algorithm which makes use of a binary tree to develop codes of varying lengths for the letters used in the original message. The procedure is as follows: First, the letters used in the secret message, including the "space" character, along with the frequency with which they occur in the message are listed. Considering each of these character/frequency pairs to be nodes, two nodes are picked with the lowest frequency and if there is a tie, randomly amongst those with equal frequencies are picked. A new node is made out of these two and two other are made its children. This new node is assigned the sum of the frequencies of its children. The process of combining the two nodes of lowest frequencies is continued until only one node, the root remains. Thus the

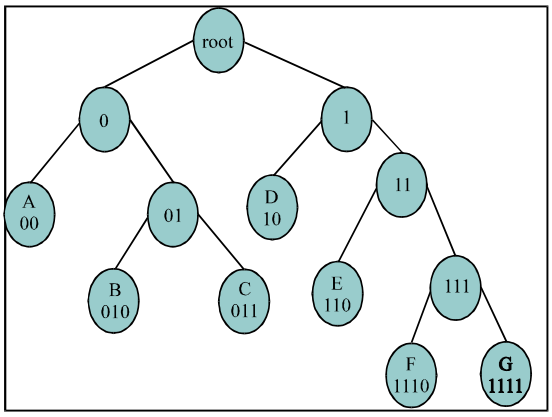

Fig.3: Sample Huffman encoding tree to assign unique coe

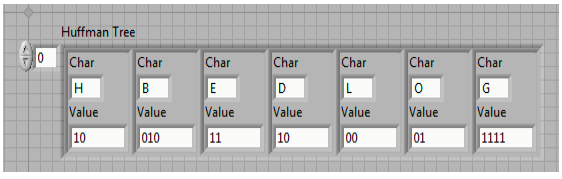

Fig. 4: Huffman table from LabVIEW Front panel

Secret Message is converted to a series of bits which are then converted to their Hexadecimal equivalent. The values are then stored in text files which are then loaded into the SRAM one by one for later embedding in the carrier. Space saving effected by Huffman encoding varies depending upon the data, anywhere from 70% over ASCII text and 25% over fixed-length codes. Figure 3 shows the

sample Huffman Tree for encoding few message characters. Figure 4 shows the Huffman Table obtained from LabVIEW.

COVER AUDIO AND SECRET MESSAGE STORAGE MODULE

The samples of cover audio and secret bits are stored to the SRAM of FPGA using the VHDL code. The SRAM external to FPGA considered here is of 512 kbytes with a 16 bit word representation in each address location. High speed asynchronous CMOS Static RAM IS61LV25616 has been employed in this SRAM architecture. Values can be written to it using the control signals like read enable, write enable, chip enable, upper byte enable and lower byte enable. The SRAM has an 18 bit address line. Each memory location in the SRAM is capable of storing 16 bits which means one 16 bit sample at a location. The pseudo code for storing the secret message cover audio is as follows:

Pseudocode:
Initialize a counter.
Check for the clock event.
FOR (i = 0 to I<228628+
length (message))
{
if (count = 0)
{
Provide the control signals and the address where the data is to be written.
}
Else if (count = 1)
{
Load the data that is to be written into the data lines.

Increment the address and the count of the array to load the next value.
}
Else
{
GOTO count = 0 and repeat the loop till all the values are loaded to the SRAM.
}

SECRET DATA EMBEDDING MODULE

The secret message is embedded in the cover (audio file) using simple LSB substitution algorithm. Samples from only one channel of audio data are used to embed the secret data. The secret bits and every (n*20000) th byte of cover data (audio samples) formed from the secret message are read from the SRAM by enabling read pin, where n ranges from 1 to the no of bits. Three different counts are used for reading, modification and writing back. Last bit of the cover data is replaced with a single bit of secret data. These modified bytes are written back to the SRAM of FPGA which forms the audio stego. The pseudo code for this process is given here.

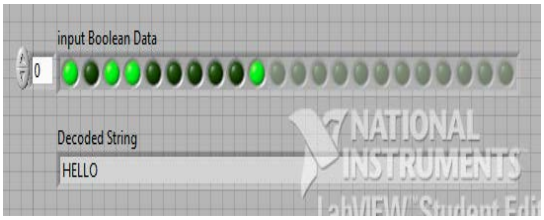

Fig. 5: Sample Embedded secret message

Pseudo code:
Initialize a counter.
Count the rising edge clock events.
{
if (count = 1)
{
Provide the address from which the data is to be read and read the data through the data lines.
}
Else if (count = 2)
{
Read the secret data and modify the content of cover data using LSB substitution.
}
Else if (count = 3)
{
Write the modified byte to the SRAM in other location which is allotted for stego audio.

Increment the address to read the next data.
}
Else
{
GOTO count = 0 and repeat the operations till all the bytes of cover data is modified.
}

Figure 5 shows the decoded message "HELLO" from the bitstream "1011000001".

RETRIEVING MODULE

This is done by specifying the required stego audio address from where the samples are fetched for further processing. The stego values are written to a text file form. The samples are read from the data lines after making write-enable to Logic high. The chip-enable and output-enable should be active low in order to have full access to the SRAM. The data hidden can be retrieved only if the corresponding extracting Huffman algorithm is implemented in the GUI. Based on LabVIEW. the initial values loaded into the Huffman table acts as key for extracting the secret message from the stego audio. From the stego file every 20000th byte is taken and its LSB is extracted to get back the message embedded. These bits are given as input to the Huffman decoder VI to get back the secret message.

The receiver who wants to get back the secret data should know the Huffman table by which encoding has been done, before hand to decrypt the message.

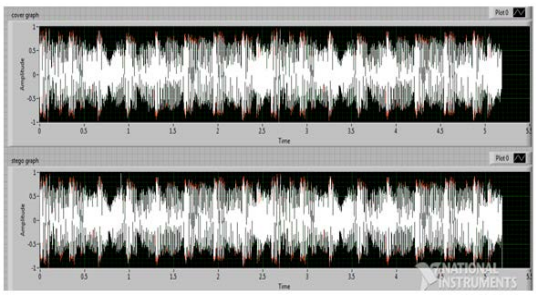

Fig. 6: Comparison of Cover and Stego Audio files

Thus this method adds security to the message because without the table the retrieved data is just a random stream of bits. This might reduce the security in the case of short messages and hence it is better to add some more characters other than those existing in the message to generate the Huffman code so that the message cannot be easily guessed using the Huffman table without having the stego audio. Along with the stego audio and Huffman table, the receiver should also know the places where the data is embedded. Here we have embedded in locations with constant intervals. This can be made even more complex by using key to generate random numbers and embedding in such locations. This adds security to the system.

The stego file which contains hex values are converted back to sound by adopting an extracting algorithm. The hex values are converted to binary values through mathscript and then to digital wavefrom through Boolean to Digital. vi. Digital waveform obtained is converted to analog waveform through Digital to Analog. vi and then to sound through playwaveform. vi. The modifications in the LSBs does not effect the audio to a large extent hence the stego file and cover file are almost undistinguishable. Figure 6 shows the cover and stego audio signals.

PERFORMANCE EVALUATION FRAMEWORK

Perceptual quality: Perceptual quality refers to the imperceptibility of embedded secret data within the host signal. In most applications, it is important that the watermark is undetectable to a listener or viewer. This ensures that the quality of the host signal is not perceivably distorted and does not indicate the presence or location of a data embedded. In this paper, the Signal-Noise Ratio (SNR) of the stego signal versus the host signal was used as a quality measure:

$$SNR = 10\log_{10}\frac{\sum_{n=0}^{N-1} S_0^{\,2}(n)}{\sum_{n=0}^{N-1}(S_w(n) - S_0(n))^2}$$

where, $s_0(n)$ is a host audio signal of length N samples and $s_w(n)$ is the stego audio signal.

The human auditory system can listen over even the smallest modifications. This puts a limit to the embedding capacity of audio steganographic system. Continuous embedding of bits results in very high noise at a particular interval. Initially 328 bits have been embedded continuously starting from the beginning in an audio with 55125 samples of 8 bit resolution. This resulted in a noise in the 1st few micro seconds. For a steganalyst it is very easy to identify. As a result, spreading the embedded bits over the cover file was done in regular intervals. To further reduce the noise effect, the resolution has been changed to 16 bits for an audio of length 5.2 sec. Figure 7 shows the relationship between Number of bits embedded in audio and Signal to Noise Ratio (SNR). The secret message bits are embedded in every nth location of the 228628 samples and SNR of this stego is observed in Fig. 7. SNR of 127.7 dB has been obtained for 100 bits payload.

Synthesis report: Altera's Quartus II Design software Version 7.2 web edition has been used for hardware implementation. FPGA Used : EP2C20F484C7 , Total Logic Elements in FPGA : 18,752. Table 1 shows the Synthesis report for various payload size.

RESULTS AND DISCUSSION

The proposed system offers better processing speed than the existing software based system. The time required for embedding is reduced as it uses dedicated hardware for processing. From the security point of view, without the particular bit file, or SRAM object file, the stego audio cannot be obtained. This aspect increases the information security by enabling particular hardware chip dependency in decrypting the message. The reduction in the embedding capacity is compensated by Huffman compression by compressing the data before embedding. Increase in security provided by the encryption makes this technique resistant to steganalytic attacks. Moreover the hardware device can be carried anywhere as it is light weight and portable. There is a disadvantage associated with the use of methods like LSB coding. The human ear is very sensitive and can often detect even the slightest bit of noise introduced into a sound file. If a sound file embedded with a secret message using LSB coding was re-sampled, the embedded information would be lost. The Fig. 7 showcased the relationship between Embedded bits in audio and SNR. Initially the secret message bits were

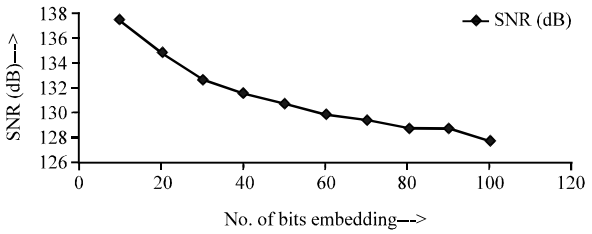

Fig. 7: No of bits embedded in digitized audio vs. SNR

Table 1: Synthesis report for various sizes of secret bits embedding in audio

No.of bits embedded in audio	Logic Elements used	sTotal Registers used	Pins used
10	78	72	41
20	78	73	41
50	81	74	41
100	82	75	41

embedded in every 2000th location of the 228628 samples. The SNR of this stego was observed to be 127.7 dB. Next the bits were embedded in every 10000th location, the SNR was found to be 135dB. Further, the bits were embedded in every 20000th location, the SNR was found to be 137dB. Hence Huffman encoding was employed by which length of secret message was reduced. For example , in order to embed a message "HELLO" by ASCII code, we require 40 bits (5 characters). In the Huffman encoding tree based approach, it was reduced to 10 bits. Also from the Table 1 , 82 Logic elements (less than 1% of total 18,752 LEs) were only used for embedding 100 bits in audio file stored in external SRAM connected to Cyclone II FPGA.

Future research direction is to explore the possibilities of improvements in audio steganographic systems on FPGA with respect to each technique of data hiding in audio. K-Bit embedding methodologies for audio needs more attention. One of the areas is to enhance the storage capacity of the system. This focuses on improving the maximum capacity of the audio signal to carry hidden data into it and making it robust to steganalysis. Further, the methods can be improved by applying mixed approaches, making the system more secure towards detection by using the combination of various techniques of data hiding in audio signals. For the real time implementation, it is better to find a threshold above which embedding should be done so that the addition of bits is not noticeable since speech might contain many silence samples.

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

Al-Azawi, A.F. and M.A. Fadhil, 2010. Arabic text steganography using kashida extensions with huffman code. J. Applied Sci., 10: 436-439.

Al-Frajat, A.K., H.A. Jalab, Z.M. Kasirun, A.A. Zaidan and B.B. Zaidan, 2010. Hiding data in video file: An overview. J. Applied Sci., 10: 1644-1649.

Amirtharajan, R., R.J.B. Balaguru and G. Vivek, 2010. Design and analysis of prototype hardware for secret sharing using 2-D image processing. Int. J. Comput. Applic., 4: 17-22.

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.

Asad, M., J. Gilani and A. Khalid, 2011. An enhanced least significant bit modification technique for audio steganography. Proceedings of the International Conference on Computer Networks and Information Technology, July 11-13, 2011, Abbottabad, Pakistan, pp: 143-147.

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.

Cvejic, N. and T. Seppanen, 2002. Increasing the capacity of LSB-based audio steganography. Proceedings of the 5th IEEE Workshop on Multimedia Signal Processing, Dec. 9-11, St. Thomas, VI, pp: 336-338.

Delforouzi, A. and M. Pooyan, 2007. Adaptive digital audio steganography based on integer wavelet transform. Proceedings of the 3rd IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Volume 2, November 26-28, 2007, Kaohsiung, Taiwan, pp: 283-286.

Djebbar, F., B. Ayady, H. Hamam and A.M. Karim, 2011. A view on latest audio steganography techniques. Proceedings of the International Conference on Innovations in Information Technology, April 25-27, 2011, Abu Dhabi, pp: 409-414.

Farouk, H.A. and M.M. Saeb, 2004. Design and implementation of a secret key steganographic micro-architecture employing FPGA. Proc. Des. Automat. Test Eur. Conf. Exhibit., 3: 212-217.

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.

Gopalan, K., 2003. Audio steganography using bit modification. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, April 6-10, IEEE Computer Society, Washington, DC. USA., pp: 412-424.

Gopalan, K. and Q. Shi, 2010. Audio steganography using bit modification-a tradeoff on perceptibility and data robustness for large payload audio embedding. Proceedings of the 19th International Conference on Computer Communications and Networks, August 2-5, 2010, Zurich, Switzerland, pp: 11-6.

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. Pixel forefinger for gray in color: A layer by layer stego. Inform. Technol. J., 11: 9-19.

Pooyan, M. and A. Delforouzi, 2007. LSB-based audio steganography method based on lifting wavelet transform. Proceedings of the IEEE International Symposium on Signal Processing and Information Technology, December 15-18, 2007, Giza, Egypt, pp: 600-603.

Qiao, M., A.H. Sung and Q. Liu, 2010. Predicting embedding strength in audio steganography. Proceedings of the 9th IEEE International Conference on Cognitive Informatics, July 7-9, 2010, Beijing, China, pp: 925-930.

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.

Shahadi, H.I. and R. Jidin, 2011. High capacity and inaudibility audio steganography scheme. Proceedings of the7th International Conference on Information Assurance and Security, December 5-8, 2011, Melaka, Malaysia, pp: 104-109.

Skopin, D.E., I.M.M. El-Emary, R.J. Rasras and R.S. Diab, 2010. Advanced algorithms in audio steganography for hiding human speech signal. Proceedings of the 2nd International Conference on Advanced Computer Control, Volume 3, March 27-29, 2010, Shenyang, China, pp: 29-32.

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.

Wu, D.C. and W.H. Tsai, 2003. A steganographic method for images by pixel-value differencing. Pattern Recogn. Lett., 24: 1613-1626.

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.

Zamani, M., H. Taherdoost, A.B.A. Manaf, R.B. Ahmad and A.M. Zeki, 2009. Robust audio steganography via genetic algorithm. Proceedings of the International Conference on Information and Communication Technologies, August 15-16, 2009, Pakistan, pp: 149-153.

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.