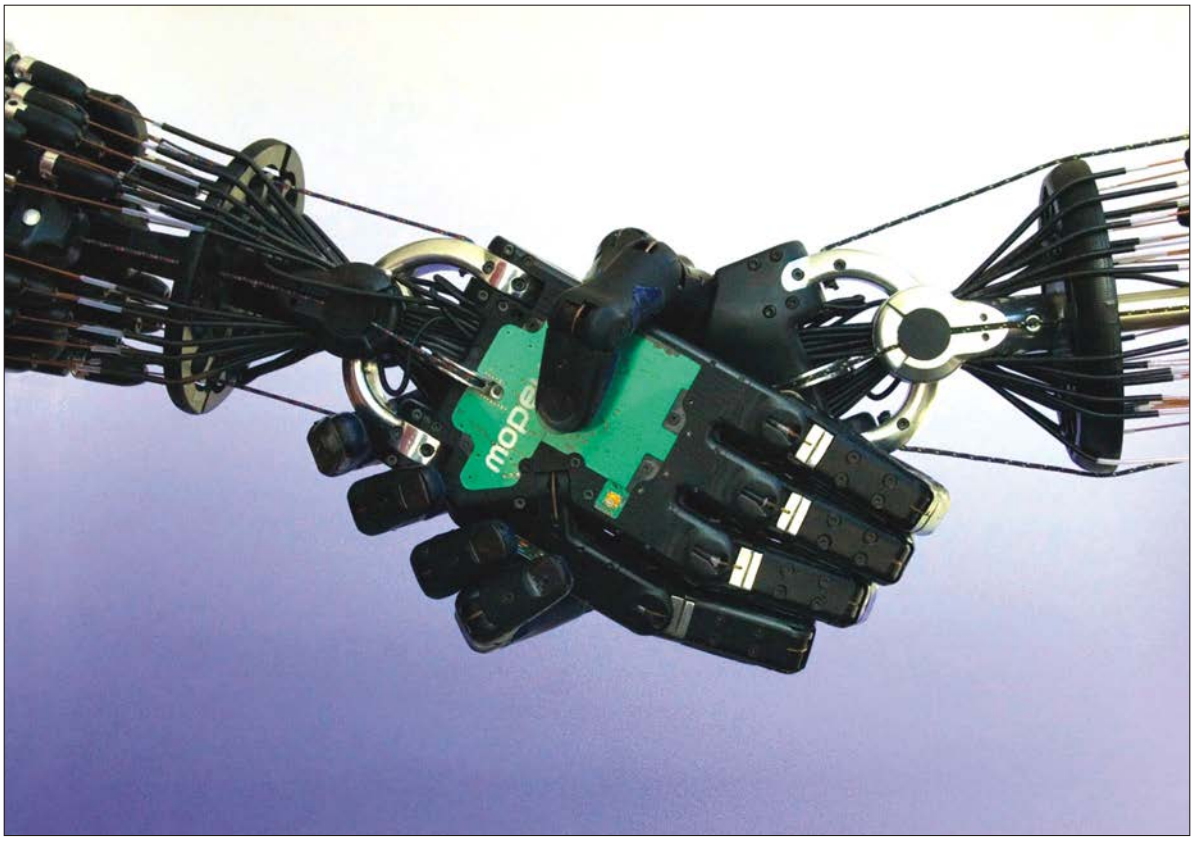

Journal of Artificial Intelligence

ISSN 1994-5450

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Artificial Intelligence 7 (2): 82-93, 2014
ISSN 1994-5450 / DOI: 10.3923/jai.2014.82.93

Audio Fingerprint Indicator in Embedded Platform: A Way for Hardware Steganography

Siva Janakiraman, K. Thenmozhi, John Bosco Balaguru Rayappan and Rengarajan Amirtharajan
Department of ECE, School of Electrical and Electronics Engineering, SASTRA University, India

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

ABSTRACT

The increased intimidation in pilfering copyrighted multimedia files demands the imprint of ownership identity called fingerprint in video/audio files. This would aid to indentify and accuse the imitators. Several hiding methods using image, audio, video or text as cover media have been found in literature with software based implementations like MATLAB. Some of these have also been reported in hardware platforms like FPGAs. As a trade-off between these implementations, embedded devices such as ARM core were used for image steganography on gray images to embed less payload in an attempt to abide the resource constrains on embedded devices. This study suggests an indicator based random LSB coding method that embeds fingerprint on a WAV audio file. The implementation was carried out on an embedded device LPC2148 with ARM7 core housed on MCB2140 evaluation board that supports audio play through on board amplifier and speaker. The original and embedded audio signal in their digitized form resides in FLASH and SRAM on-chip memories of LPC2148.

Key words: Audio fingerprint, hardware steganography, embedded, ARM7, LSB coding, steganography

INTRODUCTION

High quality in image, audio and video files are attainable in today's world with the modern word, digitization. The digital form makes it painless for an illicit person to make illegal copies of this digital content. This would distress the music, books, film and software development industries. These concerns over shielding the access to digital files and share out copies have given rise to noteworthy research that discover ways to warily conceal copyright information and serial numbers in digital files. These can help identify copyright violators and also to prosecute them. Data hiding techniques (Chan and Cheng, 2004; Amirtharajan and Rayappan, 2013) have been proposed and in exercise to carry secret digital information in out of sight over another digital media. These techniques at times may also use cryptography (Amirtharajan *et al*., 2013h, i; Janakiraman *et al*., 2014a, c; Salem *et al*., 2011; Zaidan *et al*., 2010) to scramble the data before the hiding takes place in spatial or transform domain (Praveenkumar *et al*., 2012a, b, 2013a, b, 2014a-j; Thenmozhi *et al*., 2012; Thanikaiselvan *et al*., 2012a, b, 2013a, b).

Depending on the reason behind the information to be carried, this hidden communication technique gets broadly classified in to steganography (Al-Azawi and Fadhil, 2010; Cheddad *et al*., 2010; Amirtharajan and Rayappan, 2012a, b, d; Amirtharajan *et al*., 2013e) and watermarking. A digital watermark is a sort of indication hidden in a noise-tolerant signal like audio

(Gopalan, 2003), image (Thien and Lin, 2003; Rajagopalan *et al.*, 2014a-d) or video data (Al-Frajat *et al.*, 2010). It is usually used to imply actual ownership or copyright in addition to authenticity or integrity verification of the content in digital media. Digital watermarks may or may not be visible. That is, they can be made perceptible occasionally at times after the use of some algorithm and imperceptible at all other times. Ideally, the watermark should not disrupt the original signal. Distorting the carrier information in a perceptible manner through digital watermark is not worthy. Conventional watermarks can be found inside visible media (like images or video) whereas; for digital watermarking and cover signal can also be audio or texts.

A special case of watermark is called 'digital fingerprint' where a specific data or information embedded in a digital media file that can uniquely identify every digital file. Digital fingerprints are compact unique information carefully embedded into the original digital content which may be audio, image or video. This fingerprint represents the contents' characteristics and has enough details to identify a content variant from various different sources and can effectively authenticate a file upon comparison with a database. Ideally, digital fingerprints should uniquely identify every audio or video files even when subject to modification like compression, re-sampling or even content degradation. While a digital watermark identifies ownership and traces the copyright infringements, digital fingerprints are intended to prosecute the defaulters. Fingerprint generation and identification forms an integral part of the content distributor's media workflow with ability to identify, track, monitor and monetize their content. It empowers publishers with technology to prevent copyright infringement, provides means to extend the due financial benefits to rightful content owners and obviate themselves from legal liabilities arising due to unlicensed spread of copyrighted material.

High speed, expensive solutions with reconfigurable devices such as FPGAs (Janakiraman *et al.*, 2013; Rajagopalan *et al.*, 2012a, b; Ramalingam *et al.*, 2014a, b; Rajagopalan *et al.*, 2014b) and low cost compact solutions using ARM processors (Janakiraman *et al.*, 2012a) are also reported in literature for information hiding with digital images. In addition, a few implantations on hardware reported the use of audio carrier for hiding information (Rajagopalan *et al.*, 2014a, b). Steganography with audio as cover media (Gopalan and Shi, 2010) uses several methods at various complexity levels ranging from simple LSB coding (Asad *et al.*, 2011; Cvejic and Seppanen, 2002, 2004), parity coding, phase coding, spread spectrum and echo data hiding (Bandyopadhyay *et al.*, 2008). Audio fingerprinting (Cano *et al.*, 2002) has many real world applications in the music and television industry.

A method called Pixel Indicator Technique (PIT) was proposed for RGB image steganography in order to strengthen the LSB algorithm (Amirtharajan *et al.*, 2013c, d, j). Janakiraman *et al.* (2012b) suggested two variants for the use of PIT method on grey scale images (Janakiraman *et al.*, 2012a). This study uses the simplest embedding technique on audio file namely the LSB coding combined with PIT considering a 16-bit audio sample as both indicator and channel for random embedding (Amirtharajan *et al.*, 2012; Amirtharajan and Rayappan, 2012c; Amirtharajan *et al.*, 2013a, b, f, g; Janakiraman *et al.*, 2014b; Rajagopalan *et al.*, 2014a, b) of fingerprint.

In this study, a finger print embedding algorithm implemented on ARM 7-LPC2148 device with audio file in WAV (Waveform Audio) file format is discussed. The device can be operated at 3.3V and runs up to the maximum frequency of 60 MHz. Its on-chip memory of 512 KB FLASH and 32 KB of SRAM puts a limitation on the size of audio file to be used in-turn, limiting the amount of data to embed. The fingerprint algorithm manages a balance between ideal amount of data embedded to enable comparison while keeping fingerprints lightweight for manageable access, indexing, search and storage. It can also be used to prosecute the copyright infringements legally.

A typical digital fingerprinting process involves content owners registering their content for fingerprinting and creating reference digital representation of their content in a database which is used for future comparisons. Spatial domain is more suitable for hiding techniques in embedded processors like ARM (Daniela Stanescu *et al.*, 2009).

METHODOLOGY

Audio Fingerprint Indicator (AFI) method described here two bits uses from MSBs (b_{15} to b_2) in the selected 16-bit audio samples to indicate the presence of a fingerprint bit in the LSB position (b_0) of that audio sample. Initially, the algorithm selects the indicator bits (b_n,b_m) based on which the fingerprint bit is to be embedded in audio samples. Based on the size of cover audio file (A_{max}) and size of fingerprint (Fp_{max}), the minimum embedding interval (EI_{min}) between the audio samples is selected to distribute the fingerprint bits all over the cover. In every audio sample A_i separated by the interval EI_{min}, the fingerprint embedding is done using LSB coding based on the value of indicator bits b_n, b_m. The AFI choices for embedding are given in Table 1.

The implementation of embedding algorithm detailed in Fig. 1 is done on an embedded device LPC2148 comprising ARM7 core. The digitized audio input file is in WAV format, which is a 16-bit Pulse Code Modulated (PCM) version of original analog audio signal for the duration of about 0.9 sec, sampled and quantized at 128 bits per second. This is first converted to 16-bit hexadecimal format (HEX386 or H86) using a DOS application utility program called Bin to Hex converter that runs on 32-bit Windows systems producing 14428 samples of each 16-bit in length. The hex file created using the above said utility program is loaded directly in to the 512 KB of on-chip FLASH memory.

The algorithm receives a 32-bit fingerprint value in the beginning from the user through asynchronous serial communication with a baud rate of 115.2K bits using the on-chip USART of LPC2148. During the course of embedding process, the embedded and non-embedded audio samples are stored in on-chip SRAM of 32KB provided by LPC2148. The algorithm interacts with the user during demonstration through the USART of LPC2148. Once, the device sends the embedding completion message, the user can select their options to play the embedded file or non-embedded audio files from SRAM and FLASH memories, respectively. Alternatively to verify the embedding, the user can extract the fingerprint. These options can be given as input to the algorithm through USART.

To facilitate the testing of proposed AFI algorithm in real time, in this study, MCB2140 evaluation board from KEIL is used. The board supports access to the FLASH memory of LPC2148 via., either by In System Programming (ISP) or by JTAG based debugger units like KEIL ULINK2. To play the audio files, the audio samples are sent to the 10-bit on-chip DAC of LPC2148 the output of which is amplified by a low power amplifier that drives the on board speaker module. During the play of audio signals through the speaker, the speaker volume can be adjusted using a potentiometer that controls the analog input applied to the 10-bit successive approximation based on-chip ADC of LPC2148. The block diagram of the entire hardware setup is revealed in Fig. 2.

Table 1: Embedding operation based on AFI

Value of AFI (b_n, b_m)	Operation
00	No embedding
01	Embed complement of fingerprint bit, FP_j'
10	Embed fingerprint bit, FP_j
11	No embedding

The absolute process done by the embedded software is described in the pseudo code given below.

Pseudo code describing the entire process done by the embedded software:

```
Get the number of samples in Audio cover file, Amax
Initialize the baudrate for USART of LPC2148
Transmit message ("Enter the Fingerprint to be embedded (max-4 chars)")
Get the number of Finger Print bits to be embedded, FPmax
Select the Minimum Embedding Interval between Audio Samples, EImin
Select the Indicator bits (bn, bm) that decides Embedding in Audio Samples
Initialize Audio sample count, i = 0
Initialize Finger print bit count, j = 0
while (i<Amax && j<FPmax)
{
    Read audio sample from FLASH memory, Ai
    Extract finger print bit from SRAM memory, FPj
    if (Ai % EImin == 0)
    {
        if (Ai (bn,bm) == 01)
        {
                Fpj´ = Complement (FPj)
                Embed FPj´ in LSB of Ai
                Update Embedded bit count, j = j+1
        }
        if(Ai(bn,bm) == 10)
        {
                Embed FPj in LSB of Ai
                Update Embedded bit count, j = j+1
        }
    }
    Update audio sample count, i = i+1
    Store Ai in SRAM
}
Transmit message ("Embedding completed successfully")
Transmit message ("Press P to play the Original audio file,
                E to play the Finger print embedded audio file and
                D to extract the fingerprint")
Execution mode = Receive command input character
Switch (Execution mode)
{
case P: Read volume level from the output ADC
        Read Original audio samples from FLASH memory and send it to DAC
        Transmit message ("Playing original audio from Flash memory")
case E: Read volume level from the output ADC
        Read Finger print embedded audio samples from SRAM memory and send it to DAC
        Transmit message ("Playing Finger print embedded audio from SRAM memory")
case D: Extract Finger print from embedded audio file in SRAM memory, FP
        Transmit message ("The Finger print embedded in audio file is:")
        Transmit (Extracted finger print, FP)
}
```

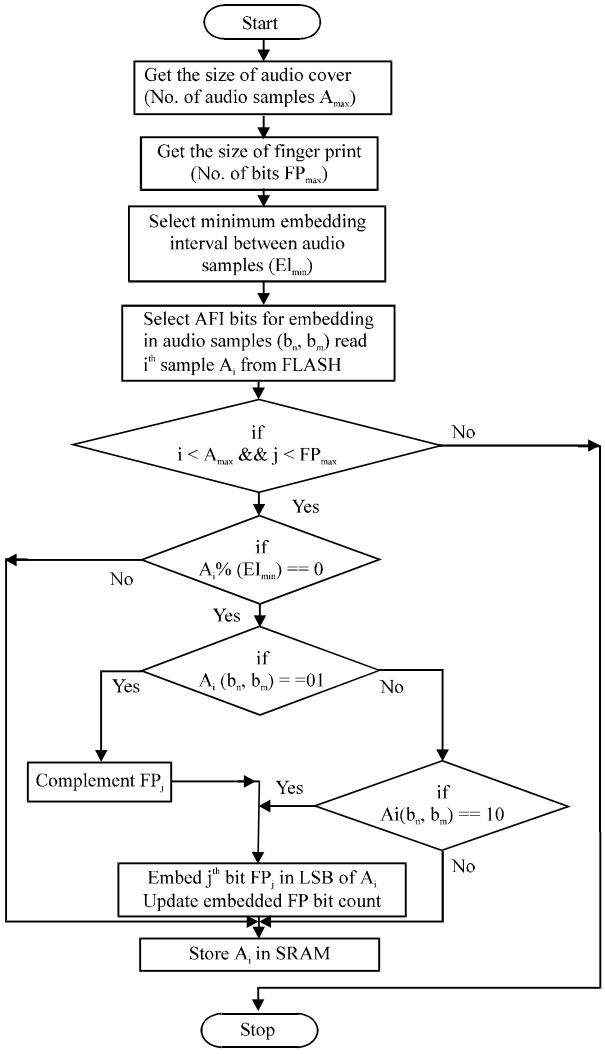

Fig. 1: Flowchart for Audio Fingerprint Indicator (AFI) embedding process

RESULTS AND DISCUSSION

The embedded software for this Audio Fingerprint Indicator (AFI) algorithm was developed using KEIL mdk, an Integrated Development Environment (IDE) for ARM devices. Here, we embedded a 32-bit fingerprint comprising of 4 characters PIT! [0×50, 0×49, 0×54, 0×21]. The embedding process was mainly carried out using the undemanding LSB coding method which is amalgamated here with an indicator based embedding technique that is usually seen with image steganography.

The proposed AFI method reinforces the conventional audio LSB coding method by the introduction of two algorithmic parameters namely the minimum Embedding Interval (EI_{min}) and the embedding indicator bits (b_n, b_m). Though the minimum interval between the samples getting embedded with fingerprint is ensured by EI_{min}, the average of actual distribution interval (EI_{avg}) varies based on the indicator (AFI) bit values. Still choosing the value of EI_{min} large enough helps

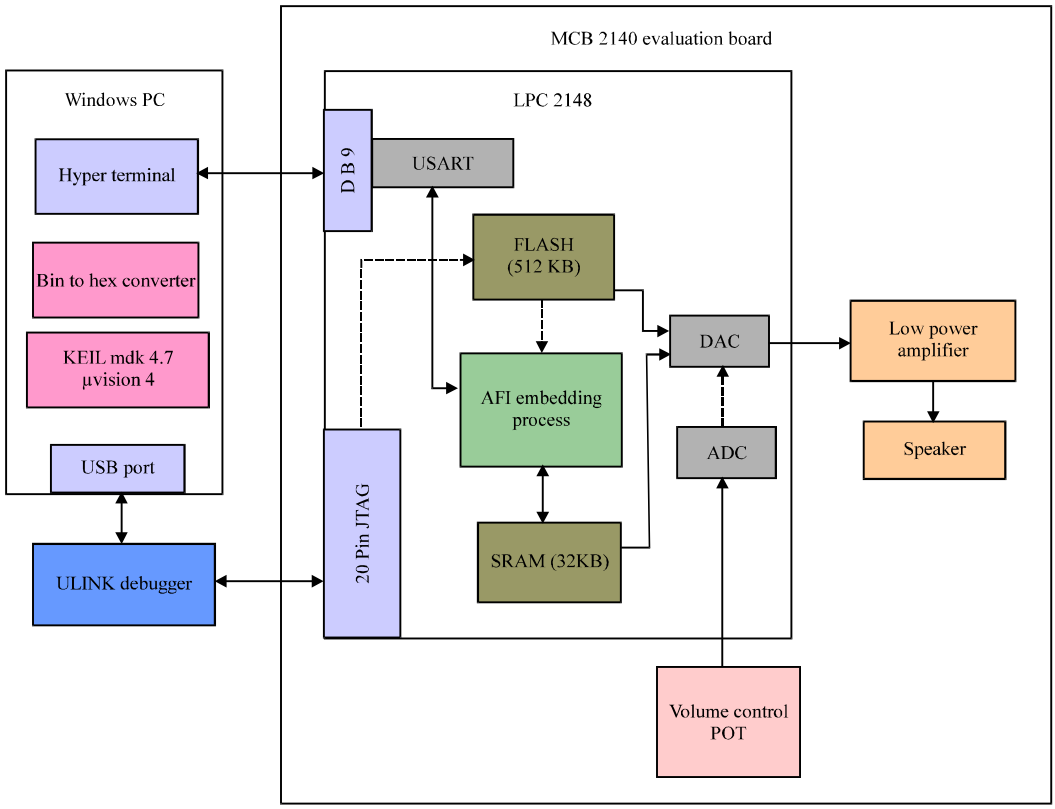

Fig. 2: Block diagram for complete hardware setup

spreading the fingerprint over large duration of audio signal measured as coverage of distribution in terms of percentage. The quality of fingerprint embedded audio signal against the real audio signal is measured by the factor Signal to Noise Ratio (SNR) given by:

$$SNR = 10 \log_{10} \frac{\sum_{i=1}^{A_{max}} S_o^2(i)}{\sum_{i=1}^{A_{max}} (S_e(i) - S_o(i))^2}$$

where, S_o and S_e are the original and embedded sample values of audio file with length A_{max}. The value of SNR, coverage of fingerprint distribution and average embedding interval against the selection values for algorithmic parameters EI_{min} and b_n, b_m are tabularized in Table 2.

The inference from Table 2 shows that for the AFI values (3, 2), (7, 6) and (9, 8) when Ei_{min} is taken as 20 coincidentally brings out equal distribution percentage and EI_{avg} while their SNR values are slightly differing. The graph in Fig. 3 shows the randomness between the audio samples selected for fingerprint embedding based on AFI values.

The obtained results showed no perceptible difference in the audio quality between embedded and original files when played at maximum volume selected through the volume control POT. The snapshots of partial audio signals with and without embedding taken during their play from logic analyzer of KEIL mdk 4.7 are shown in Fig. 4 and 5 correspondingly.

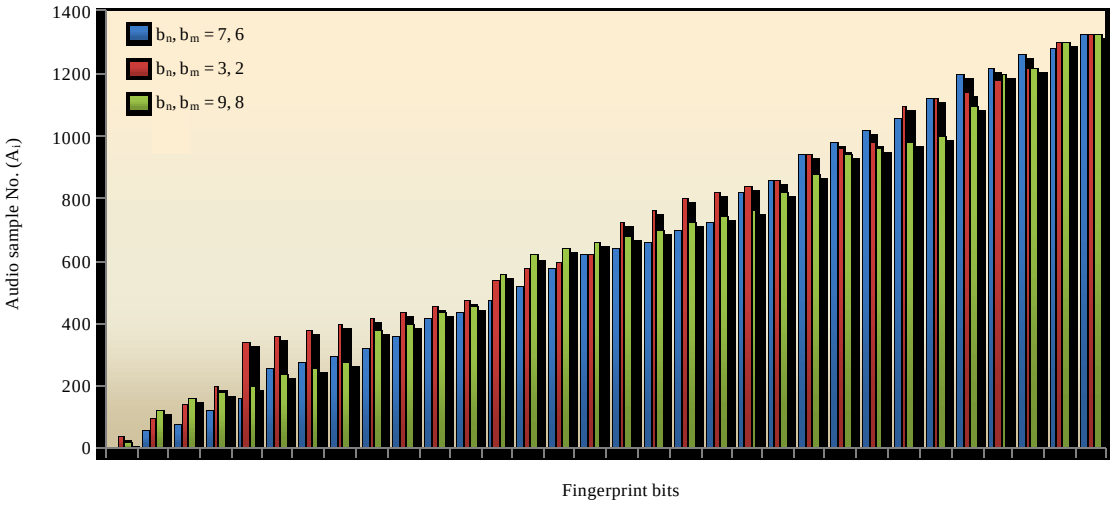

Fig. 3: Randomness in distribution of fingerprint based on selection of AFI bits

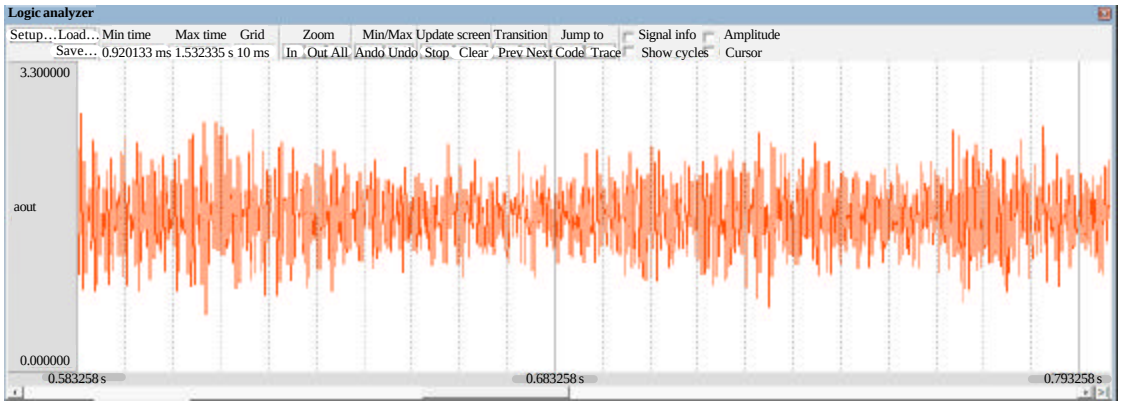

Fig. 4: Original audio signal

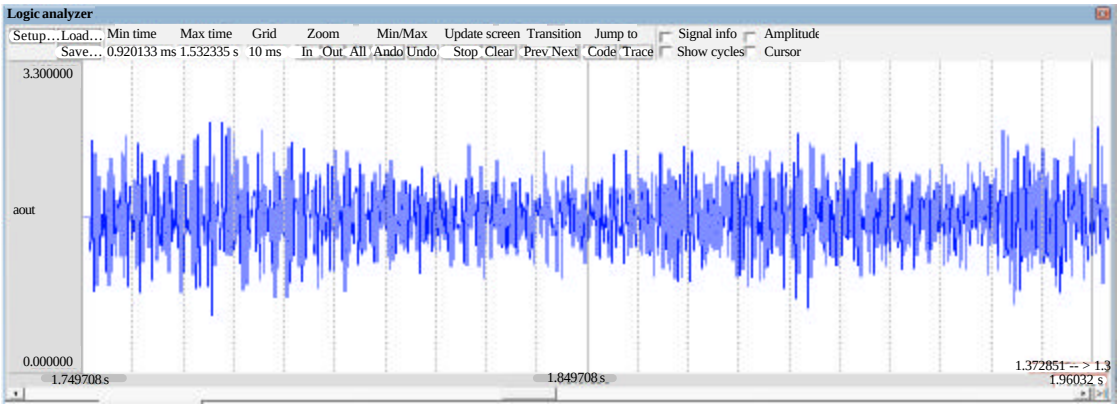

Fig. 5: Fingerprint embedded audio signal

Table 2: Analysis of SNR and randomness based on AFI algorithm parameters

Minimum embedding interval EI_{min}	Selected AFI b_n, b_m	Coverage of distribution (%)	Average embedding interval EI_{avg}	SNR (dB)
20	3,2	18.29	66	79.93
50	3,2	42.97	62	81.17
100	3,2	92.87	67	80.15
20	5,4	16.63	60	80.15
50	5,4	44.35	64	81.17
100	5,4	74.85	54	80.63
20	7,6	18.29	66	81.79
50	7,6	42.27	61	81.47
100	7,6	83.17	60	79.93
20	9,8	18.29	66	81.79
50	9,8	38.81	56	80.63
100	9,8	EER*	EER*	EER*
20	11,10	20.79	75	81.47
50	11,10	51.28	74	80.63
100	11,10	EER*	EER*	EER*
20	13,12	19.41	70	82.14
50	13,12	45.74	66	80.63
100	13,12	85.94	62	80.63
20 / 50 / 100	15,14	EER*	EER*	EER*

*EER (Embedding Error) occurs when required numbers of samples are not available in audio cover with the value of selected AFI bits that leads to incomplete embedding of fingerprint

The real time values of execution time on MCB2140 evaluation board was measured using the hardware debugger unit ULINK2. The embedding of 32-bit fingerprint takes about 65.89 ms whereas the extraction (decoding) takes 65.07 ms when LPC2148 is operated at 60 MHz.

CONCLUSION

The extended features of the ARM cores make it suitable for processing audio signals. Though the original input signals can be housed in larger FLASH area, the available SRAM for the storage of processed information limits the input size (cover). By embedding 32-bit finger over 14428 samples, this study achieved tolerable SNR in the range of 79.7dB-82.1dB. During the audibility test carried out by playing the original and embedded audio files by adjusting the volume control POT to different levels, it was found that the audio quality of embedded file degrades proportionally for the reduction in volume whilst, the original audio file quality remains unaffected. This can be considered as a severe glitch that makes the job of spotting the difference between original and embedded audio files effortless. On the other side, AFI method serves as pseudo random generator in the selection of audio samples for fingerprint embedding and thus strengthens the simple LSB coding.

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

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

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

Amirtharajan, R. and J.B.B. Rayappan, 2013. Steganography-time to time: A review. Res. J. Inform. Technol., 5: 53-66.

Amirtharajan, R., K. Karthikeyan, M. Malleswaran and J.B.B. Rayappan, 2013a. Kubera kolam: A way for random image steganography. Res. J. Inform. Technol., 5: 304-316.

Amirtharajan, R., M.V. Abhiram, G. Revathi, J.B. Reddy, V. Thanikaiselvan and J.B.B. Rayappan, 2013b. Rubik's cube: A way for random image steganography. Res. J. Inform. Technol., 5: 329-340.

Amirtharajan, R., K.M. Ashfaaq, A.K. Infant and J.B.B. Rayappan, 2013c. High performance pixel indicator for colour image steganography. Res. J. Inform. Technol., 5: 277-290.

Amirtharajan, R., R. Subrahmanyam, J.N. Teja, K.M. Reddy and J.B.B. Rayappan, 2013d. Pixel indicated triple layer: A way for random image steganography. Res. J. Inform. Technol., 5: 87-99.

Amirtharajan, R., S.D. Roy, N. Nesakumar, M. Chandrasekar, R. Sridevi and J.B.B. Rayappan, 2013e. Mind game for cover steganography: A refuge. Res. J. Inform. Technol., 5: 137-148.

Amirtharajan, R., V. Rajesh, P. Archana and J.B.B. Rayappan, 2013f. Pixel indicates, standard deviates: A way for random image steganography. Res. J. Inform. Technol., 5: 383-392.

Amirtharajan, R., G. Devipriya, V. Thanikaiselvan and J.B.B. Rayappan, 2013g. High capacity triple plane embedding: A colour stego. Res. J. Inform. Technol., 5: 373-382.

Amirtharajan, R., P. Archana and J.B.B. Rayappan, 2013h. Why image encryption for better steganography. Res. J. Inform. Technol., 5: 341-351.

Amirtharajan, R., S. Sulthana and J.B.B. Rayappan, 2013i. Seeing and believing is a threat: A visual cryptography schemes. Res. J. Inform. Technol., 5: 435-441.

Amirtharajan, R., P.S. Priya and J.B.B. Rayappan, 2013j. Pixel indicated user indicator: A muxed stego. Res. J. Inform. Technol., 5: 73-86.

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.

Bandyopadhyay, S.K. D. Bhattacharyya, P. Das, D. Ganguly and S. Mukherjee, 2008. A tutorial review on steganography. Proceedings of the International Conference on Contemporary Computing, August 7-9, 2008, Noida, India, pp: 105-114.

Cano, P., E. Batle, T. Kalker and J. Haitsma, 2002. A review of algorithms for audio fingerprinting. Proceedings of the Workshop on Multimedia Signal Processing Lausanne, December 9-11, 2002, Switzerland, pp: 169-173.

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

Cheddad, A., J. Condell, K. Curran and P. Mc 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, December 9-11, 2002, St. Thomas, VI., pp: 336-338.

Cvejic, N. and T. Seppanen, 2004. Increasing robustness of LSB audio steganography using a novel embedding method. Proceedings of the International Conference on Information Technology: Coding and Computing, April 5-7, 2004, Sunderland, UK., pp: 533-537.

Daniela, S., V. Stangaciu, I. Ghergulescu and M. Stratulat, 2009. Steganography on embedded devices. Proceedings of the 5th International Symposium on Applied Computational Intelligence and Informatics, May 28-29, 2009, Timisoara, pp: 313-318.

Gopalan, K., 2003. Audio steganography using bit modification. Proceedings of the International Conference on Acoustics, Speech and Signal Processing, April 6-10, 2003, 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., K.V.S.K. Kumar, R.R.K. Reddy, A. Srinivasulu, R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2014a. Humming bird with coloured wings: A feedback security approach. Inform. Technol. J., 13: 2022-2026.

Janakiraman, S., K. Thenmozhi, S. Rajagopalan, H.N. Upadhyay, J.B.B. Rayappan and R. Amirtharajan, 2014b. Space filling curve for data filling: An embedded security approach. Res. J. Inform. Technol., 6: 188-197.

Janakiraman, S., J. Chakravarthy, B. Radhakrishnan, K. Thenmozhi, J.B.B. Rayappan and R. Amirtharajan, 2014c. Cover as key and key as data: An inborn stego. Inform. Technol. J., 13: 1969-1976.

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

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., S. Rajagopalan, K. Thenmozhi, H.N. Upadhyay and J. Ramanathan *et al.*, 2013. Captivating CODEC Stego (CCS): A cover on camouflage. Res. J. Inform. Technol., 5: 160-170.

Praveenkumar, P., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2012a. Regulated OFDM-role of ECC and ANN: A review. J. Applied Sci., 12: 301-314.

Praveenkumar, P., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2012b. Phase for face saving-a multicarrier stego. Procedia Eng., 30: 790-797.

Praveenkumar, P., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2013a. Can we reduce PAPR? OFDM+PTS+SLM+STEGO: A novel approach. Asian J. Sci. Res., 6: 38-52.

Praveenkumar, P., M. Nagadinesh, P. Lakshmi, K. Thenmozhi, J.B.B. Rayappan and R. Amirtharajan, 2013b. Convolution and viterbi EN(DE)coders on OFDM hides, rides and conveys message-A neural STEGO. Proceedings of the International Conference on Computer Communication and Informatics, January 4-6, 2013, Coimbatore, pp: 1-5.

Praveenkumar, P., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2014a. Sub carriers carry secret: An absolute stego approach. J. Applied Sci., 14: 1728-1735.

Praveenkumar, P., R. Amirtharajan, K. Thenmozhi and J.B.B. Rayappan, 2014b. Double layer encoded encrypted data on multicarrier channel. J. Applied Sci., 14: 1689-1700.

Praveenkumar, P., K. Thenmozhi, J.B.B. Rayappan and R. Amirtharajan, 2014c. Purposeful error on OFDM: A secret channel. Inform. Technol. J., 13: 1985-1991.

Praveenkumar, P., G.S. Hemalatha, B. Reddy, K. Thenmozhi, J.B.B. Rayappan and R. Amirtharajan, 2014d. Secret link through simulink: A stego on OFDM channel. Inform. Technol. J., 13: 1999-2004.

Praveenkumar, P., K. Thenmozhi, J.B.B. Rayappan and R. Amirtharajan, 2014e. Stego in multicarrier: A phase hidden communication. Inform. Technol. J., 13: 2011-2016.

Praveenkumar, P., K. Thenmozhi, J.B.B. Rayappan and R. Amirtharajan, 2014f. Inserted embedding in OFDM channel: A multicarrier stego. Inform. Technol. J., 13: 2017-2021.

Praveenkumar, P., K. Thenmozhi, J.B.B. Rayappan and R. Amirtharajan, 2014g. Data puncturing in OFDM channel: A multicarrier stego. Inform. Technol. J., 13: 2037-2041.

Praveenkumar, P., R. Deepak, K. Thenmozhi, J.B.B. Rayappan and R. Amirtharajan, 2014h. Reversible steganography on OFDM channel: A role of cyclic codes. Inform. Technol. J., 13: 2047-2051.

Praveenkumar, P., K. Thenmozhi, J.B.B. Rayappan and R. Amirtharajan, 2014i. Reversible steganography on OFDM channel-a role of RS coding. Inform. Technol. J., 13: 2052-2056.

Praveenkumar, P., K. Thenmozhi, J.B.B. Rayappan and R. Amirtharajan, 2014j. Spread and hide-a stego transceiver. Inform. Technol. J., 13: 2061-2064.

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.

Rajagopalan, S., K. Pravallika, R. Radha, H.N. Upadhyay, J.B.B. Rayappan and R. Amirtharajan, 2014a. Stego on song-an amalgam of VI and FPGA for hardware info hide. Inform. Technol. J., 13: 1992-1998.

Rajagopalan, S., S. Janakiraman, B. Swaminath, H.N. Upadhyay, K. Thenmozhi and R. Amirtharajan, 2014b. LabVIEW based PIN hider on ATM cards: A transform domain secret concealment approach. Res. J. Inform. Technol., 6: 154-165.

Rajagopalan, S., Y. Ravishankar, H.N. Upadhyay, J.B.B. Rayappan and R. Amirtharajan, 2014c. Modeling combo PR Generator for Stego Storage Self Test (SSST). Inform. Technol. J., 13: 1936-1944.

Rajagopalan, S., P.J.S. Prabhakar, M.S. Kumar, N.V.M. Nikhil, H.N. Upadhyay, J.B.B. Rayappan and R. Amirtharajan, 2014d. MSB based embedding with integrity: An adaptive RGB Stego on FPGA platform. Inform. Technol. J., 13: 1945-1952.

Ramalingam, B., R. Amirtharajan and J.B.B. Rayappan, 2014a. LCC-LSB-FPGA stego-A reconfigurable security. J. Applied Sci., 14: 2139-2148.

Ramalingam, B., R. Amirtharajan and J.B.B. Rayappan, 2014b. Stego on FPGA: An IWT approach. Sci. World J. 10.1155/2014/192512

Salem, Y., M. Abomhara, O.O. Khalifa, A.A. Zaidan and B.B. Zaidan, 2011. A review on multimedia communications cryptography. Res. J. Inform. Technol., 3: 146-152.

Thanikaiselvan, V., P. Arulmozhivarman, R. Amirtharajan and J.B.B. Rayappan, 2012a. Wavelet Pave the Trio Travel for a Secret Mission-A Stego Vision. In: Global Trends in Information Systems and Software Applications, Krishna, P.V., M.R. Babu and E. Ariwa (Eds.). Springer, USA., ISBN: 9783642292156, pp: 212-221.

Thanikaiselvan, V., P. Arulmozhivarman, R. Amirtharajan and J.B.B. Rayappan, 2012b. Horse riding and hiding in image for data guarding. Procedia Eng., 30: 36-44.

Thanikaiselvan, V., P. Arulmozhivarman, S. Subashanthini and R. Amirtharajan, 2013a. A graph theory practice on transformed image: A random image steganography. Sci. World J. 10.1155/2013/464107

Thanikaiselvan, V., K. Santosh, D. Manikanta and R. Amirtharajan, 2013b. A new steganography algorithm against chi square attack. Res. J. Inform. Technol., 5: 363-372.

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 Recognit., 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.