International Journal of Computer Science and Security (IJCSS)

ISSN : 1985-1553

VOLUME 3, ISSUE 6

PUBLICATION FREQUENCY: 6 ISSUES PER YEAR

Editor in Chief Dr. Haralambos Mouratidis

International Journal of Computer Science and Security (IJCSS)

Book: 2009 Volume 3, Issue 6
Publishing Date: 30-01-2010
Proceedings
ISSN (Online): 1985-1553

IJCSS Journal is a part of CSC Publishers
http://www.cscjournals.org

Typesetting: Camera-ready by author, data conversation by CSC Publishing Services – CSC Journals, Malaysia

CSC Publishers

Table of Contents

Volume 3, Issue 6, January 2010.

Pages

A wavelet - Based Object Watermarking System for MPEG4 Video

A.Essaouabi abdessamad1977@yahoo.fr
Department of physics, LIMIARF Laboratory,
Mohammed V University
Rabat, Morocco

E.Ibnelhaj ibnelhaj@inpt.ac.ma
Institut National of Posts and Telecommunications,
Rabat, Morocco

F.Regragui regragui@fsr.ac.ma
Department of physics, LIMIARF Laboratory,
Mohammed V University
Rabat, Morocco

Abstract

Efficient storage, transmission and use of video information are key requirements in many multimedia applications currently being addressed by MPEG-4. To fulfill these requirements, a new approach for representing video information which relies on an object-based representation, has been adopted. Therefore, object-based watermarking schemes are needed for copyright protection. This paper presents a novel object based watermarking solution for MPEG4 video authentication using the shape adaptive-discrete wavelet transform (SA-DWT). In order to make the watermark robust and transparent, the watermark is embedded in the average of wavelet blocks using the visual model based on the human visual system. Wavelet coefficients n least significant bits (LSBs) are adjusted in concert with the average. Simulation results show that the proposed watermarking scheme is perceptually invisible and robust against many attacks such as lossy compression (e.g. MPEG1 and MPEG2, MPEG-4, H264).

Keywords: *Watermark*, *Visual model*, Robustness, Shape adaptive-discrete wavelet transform.

1. INTRODUCTION

With the emergence of new multimedia standards such as Mpeg-4, the notion of video-object or image object is more and more widespread [1][2]. Consequently, protecting the different objects of an image or a video appeared necessary. Therefore several object-based watermarking techniques, those consist at introducing an invisible signal known as a digital watermark into an image or video sequence, aim at solving this type of problem. Wu and al. [3] proposed a multiresolution object watermarking approach based on 2D and 3D shape adaptive wavelet transforms. The advantage of the multiresolution watermarking method is its robustness against image/video compression and computational saving. However, the main disadvantage is that original image/video object is required for watermark detection. Kim and al. [4] proposed an object-based video watermarking method using the shape adaptive-discrete cosine transforms

(SA-DCT). The SA-DCT method is superior to all other padding methods in terms of robustness against the image deformations. Yet, the watermark can be damaged by a wavelet-based image codec in the quantization stage. Therefore, this method limits their applications in the context of JPEG2000 and MPEG-4 due to the fact that the wavelet transform is playing an important role in JPEG2000 and MPEG-4. Piva and al. [5] propose an object watermarking system for MPEG-4 streams. Since this method applies the discrete wavelet transform (DWT) to the whole image and the watermark is embedded in all the wavelet coefficients belonging to the three detail bands at level 0, this may lead to loss of the watermark which is embedded in the region outside the object. Barni and Bartolini [6] proposed a method that consists in embedding a watermark in each video object of an MPEG-4 coded video bit-stream by imposing specific relationships between some predefined pairs of quantized DCT middle frequency coefficients in the luminance blocks of pseudo-randomly selected macroblocks. The quantized coefficients are recovered from the MPEG-4 bit-stream, they are modified to embed the watermark and then encoded again. The main drawback of this technique is that, since the code is directly embedded into the compressed MPEG-4 bit-stream, the copyright information is lost if the video file is converted to a different compression standard, like MPEG-2. In order to be robust against format conversions, the watermark has to be inserted before compression, i.e. frame by frame.

In order to satisfie the previous requirements we propose in this paper an object based watermarking solution for MPEG4 video authentication based on in place lifting SA-DWT . The watermark signal is embedded in the wavelet coefficients n LSBs before MPEG4 encoding and not embedded in the region outside object. Unlike most watermark schemes, watermark embedding is performed by modulating the average of the wavelet coefficients instead of the individual coefficients in the wavelet block. Visual model is employed to achieve the best tradeoff between transparent and robustness to signal processing. Watermark detection is accomplished without the original. Experimental results demonstrate that the proposed watermarking scheme is perceptually invisible and robust against unintentional and intentional attacks such as lossy video compression (e.g. MPEG2 and MPEG-4,MPEG1 and H264).

The rest of this paper is organized as follows:
In section 2, we briefly describe the SA-DWT, section 3 will describe the basic functionalities of watermarking embedding and extraction procedure, section 4 will give the simulations results, finally section 5 will give the conclusion.

2. IN-PLACE LIFTING SHAPE ADAPTIVE-DISCRETE WAVELET TRANSFORM

Given an arbitrarily shaped object with shape mask information, with in place lifting SA-DWT[8][9], the number of transformed coefficients is equal to the number of pixels in the arbitrarily shaped segment image, and the spatial correlation across subbands is well preserved. Fig. 1 illustrates the result of one-level wavelet decomposition of an arbitrarily shaped object.

The in-place lifting DWT implementation has special implications for the SA-DWT[9], which can best be understood visually as shown in Fig. 1. As the SA-DWT is performed, the spatial domain shape mask remains intact with no requirement to derive a shape mask for each subband. How the subbands are arranged in this pseudo-spatial domain arrangement is shown in Fig. 2(a). Each subband can in fact be extracted from the interleaved subband arrangement using the lazy wavelet transform (LWT) [10]. After the one-level SA-DWT is performed, the LL1 subband can be extracted using a coordinate mapping from the interleaved subband coordinates (i,j) to the LL1 subband coordinates (i_{LL1},j_{LL1}) as follows:

$$(i_{LL1}, j_{LL1}) \leftarrow ([i/2], [j/2]) \qquad (1)$$

Similarly, the mapping for the HL1 subband is $(i_{HL1}, j_{HL1}) \leftarrow ([i/2]+1,[j/2])$; for the LH1 subband $(i_{LH1}, j_{LH1}) \leftarrow ([i/2],[j/2]+ 1)$; and for the HH1 subband $(i_{HH1}, j_{HH1}) \leftarrow ([i/2]+1,[j/2]+1)$.

After the first level of the SA-DWT, the interleaved subband arrangement is made up of 2 × 2 basic blocks of coefficients. As shown in the left side of Fig. 2 (b), the top-left coefficient of each block is an LL1 subband coefficient, the top-right coefficient is an HL1 subband coefficient, and so on The second level SA-DWT is performed by first extracting the LL1 subband using the coordinate mapping (1) and then performing the one-level SA-DWT using the LL1 subband as the new input. The output is the four interleaved subbands, LL2, HL2, LH2, and HH2. This is then placed back into the original interleaved subband arrangement where the LL1 coefficients were extracted from. This creates a two-level interleaved subband arrangement. As shown in the middle of Fig.2(b), the two-level interleaved subband arrangement is made of a basic 4×4 coefficient block, with the top-left coefficient of each block being an LL2 coefficient. The coordinate mappings to extract the second and subsequent level subbands are simply derived by applying the one level coordinate mappings iteratively to the LL subband coordinate mapping from the previous level.

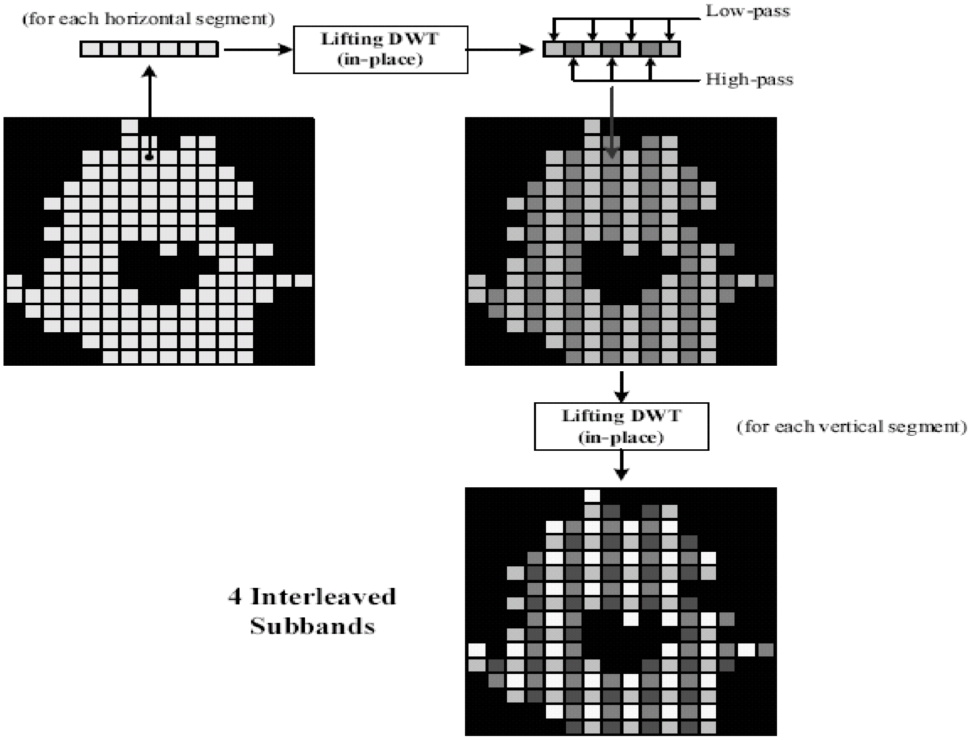
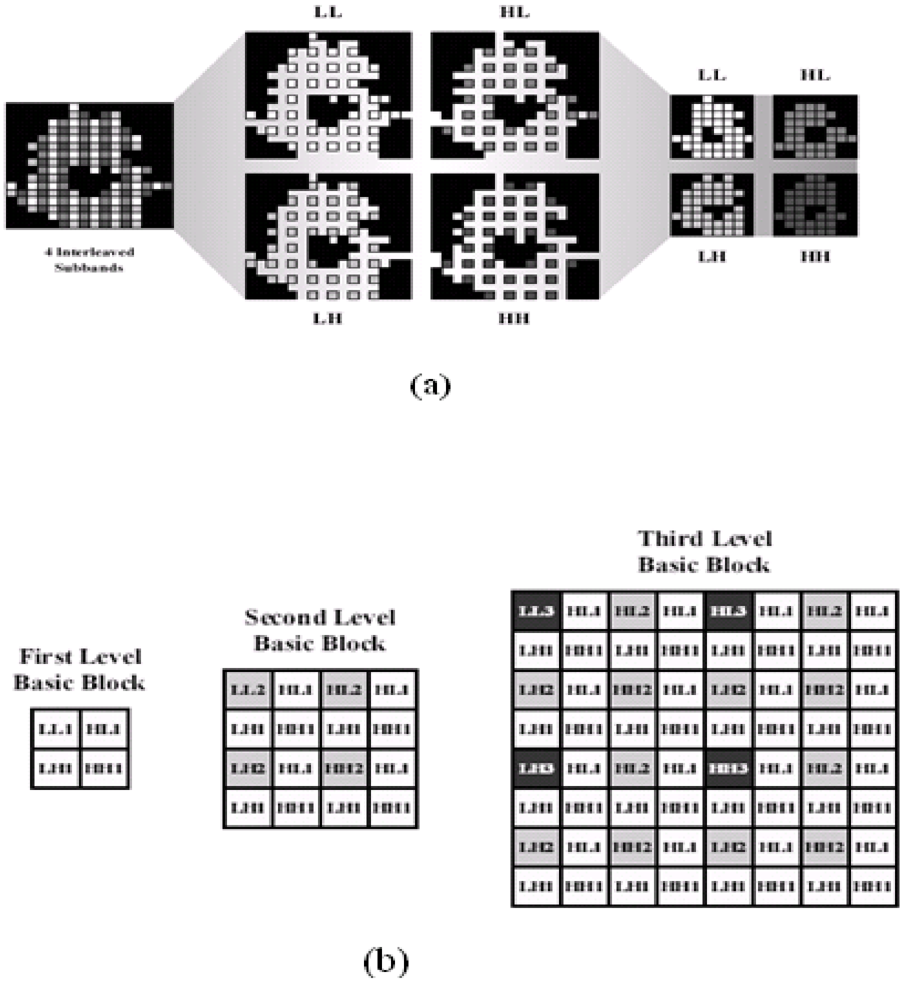


FIGURE 1: One-Level Two-Dimensional SA-DWT Using In-Place Lifting DWT Implementation



First Level Basic Block

LL1	HL1
LH1	HH1

Second Level Basic Block

LL2	HL1	HL2	HL1
LH1	HH1	LH1	HH1
LH2	HL1	HH2	HL1
LH1	HH1	LH1	HH1

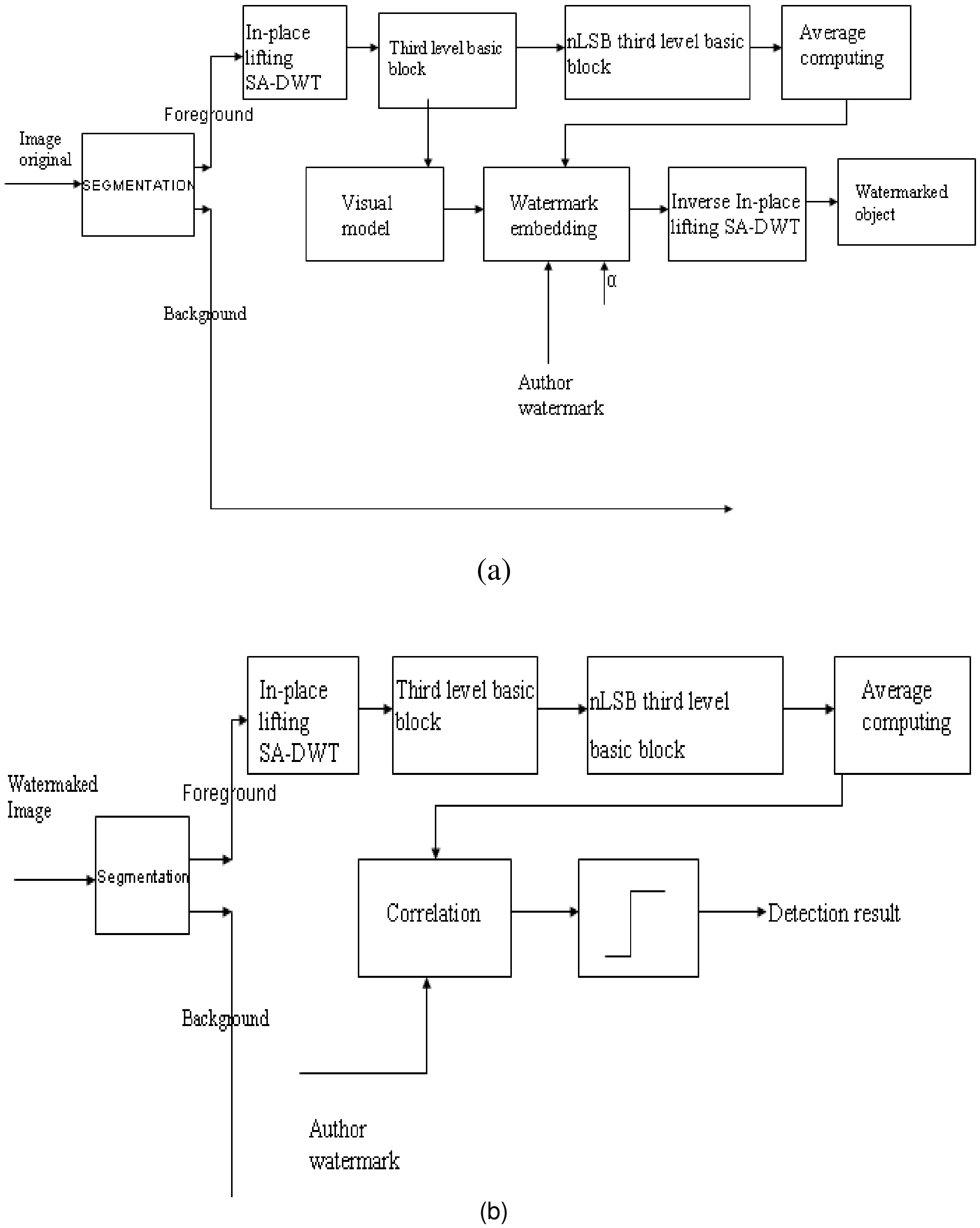
Third Level Basic Block

LL3	HL1	HL2	HL1	HL3	HL1	HL2	HL1
LH1	HH1	LH1	HH1	LH1	HH1	LH1	HH1
LH2	HL1	HH2	HL1	LH2	HL1	HH2	HL1
LH1	HH1	LH1	HH1	LH1	HH1	LH1	HH1
LH3	HL1	HL2	HL1	HH3	HL1	HL2	HL1
LH1	HH1	LH1	HH1	LH1	HH1	LH1	HH1
LH2	HL1	HH2	HL1	LH2	HL1	HH2	HL1
LH1	HH1	LH1	HH1	LH1	HH1	LH1	HH1

FIGURE 2: (a) Interleaved Subband Abstraction (b) Basic Group of Coefficients for Each Level of In-Place DWT

3. PROPOSED WATERMARKING SCHEME

A content-based watermarking system for content integrity protection is illustrated in Fig. 3.



(a)

(b)

FIGURE 3 : Block Diagrams for The Proposed Watermarking Scheme. (a) Watermark Embedding (b) Watermark Detection.

3.1 Watermark Embedding

Fig.3 (a) shows the watermarking embedding procedure. Firstly, the MPEG-4 coded video bit stream is decoded obtaining a sequence of frames. Each frame is segmented into foreground (objects) and background and we apply the three levels SA-DWT to foreground object frame by frame. Then, we apply the algorithm scheme at each third level basic block (see fig 2(b)). NxN is the size of the matrix wavelet block and $I_i(k)$ is the ith wavelet coefficient in the kth wavelet block where i Є [1, N×N].

The rest of the watermarking embedding procedure is presented in and resumed in the following. The *n* LSBs of $I_i(k)$ is defined as :

$$\hat{I}_i(k)=\mathrm{mod}(I_i(k),2^n) \qquad (2)$$

The average of the wavelet block is defined as follows:

$$Average\ (k)=\frac{\sum_{i=1}^{N\times N}\hat{I}_i(\mathrm{k})}{N\times N} \qquad (3)$$

In the proposed watermarking, we choose the blocks with an average value different from zero. If a few of I_i (k) are changed by *Ω* due to some distortions, the average of the wavelet block will only have a small change. Assuming that I'_i(k) is the ith wavelet coefficient in the kth wavelet block after the watermark embedding, $\hat{I}_{i'}$ (k) is the *n* LSBs of I'_i (k) and Average′(k) is the average of $\hat{I}_i$ ′(k) in the kth wavelet block accordingly. The watermark W, consisting of a binary pseudo random sequence, *W(k)* Є {–1, 1}, is embedded by adjusting the average of wavelet blocks in this way :

$$Average'(k)\in\begin{cases}\left[0,2^{n-1}\right),if \quad W(k)=-1\\ \left[2^{n-1},2^{n}\right),if \quad W(k)=1\end{cases} \qquad (4)$$

To adapt the watermark sequence to the local properties of the wavelet block, we use the model based on HVS in the watermark system. The model is similar to that proposed in [7], but it is developed independently. The visual model takes into account the brightness sensitivity and texture sensitivity of the wavelet block to noise. The visual model function Vm(k) is defined as:

$$Vm(k)=brightness\ (k)x\ texture(k)^{\beta} \qquad (5)$$

where

$$texture(k)=\frac{\sum_{i=1}^{N\times N}\left[brightness(k)-I_i(k)\right]^2}{N\times N}$$

$$brightness(k)=\frac{\sum_{i=1}^{N\times N}I_i(k)}{N\times N}$$

β is a parameter used to control the degree of texture sensitivity. This visual model function indicates that the human eye is less sensitive to noise in the highly bright and the highly textured areas of the image. Hence, the wavelet blocks are divided into two parts depending on the value of Vm(k): high activity wavelet block and low activity wavelet block. For simplicity, the threshold Tc is set to the average of Vm(k). The following function can be applied to distinguish high or low activity wavelet block:

$$T(k)=\mathrm{sign}(Vm(k)-Tc) \qquad (6)$$

Considering the tradeoff between robustness and transparency, the proposed watermark embedding algorithm can be formulated as follows:

$$I'_i(k)=I_i(k)+\alpha W(k)F_i(k)[2^{n-2-S(k)}+T(k)x2^{n-3}] \qquad (7)$$

where α is a scaling factor used to control the strength of the inserted watermark. The flag function is defined as follows:

$$F_i(k)=\text{sign}((2^{n-1}-I_i(k))xW(k)) \tag{8}$$

where

$$sign(x)=\begin{cases} 1 \quad if \quad x\geq 0 \\ -1 \quad if \quad x<0 \end{cases}$$

The strength function is defined as follows:

$$S(k)=\text{sign}(X(k)) \tag{9}$$

Where

$$X(k)=(2^{n-1}-\text{Average}(k))xW(k)$$

Details concerning the flag function and the strength function are described in table 1.

W(k)	$2^{n-1}-\hat{I}_i(k)$	2^{n-1}-Average(k)	$F_i(k)$	S(k)
-1	>0	>0	-1	-1
-1	≤0	≤0	1	1
1	>0	>0	1	1
1	≤0	≤0	-1	-1

TABLE 1: The Detailed Results of $F_i(K)$ and S(K)

In light of the above, the *n* LSBs of wavelet coefficients have been adjusted by using equation (7). Naturally, their average has been updated depending on the requirement of *W(k)* as show in equation (4). In other word, the watermark has been embedded.

3.2 Watermark Extraction and Detection

The watermark sequence can be extracted without the original object. From the process of the watermark embedding, we can obtain the watermarked objects by applying the function of equation (3). Thus, for a given watermarked object, the watermark can be extracted as:

$$W'(k)=\begin{cases} -1, \quad if \quad Average(k)\in\left[0,2^{n-1}\right) \\ 1, \quad if \quad Average(k)\in\left[2^{n-1},2^{n}\right) \end{cases} \tag{10}$$

In order to detect the watermark *W'* extracted from the watermarked object, we firstly evaluate the detector response (or similarity of *W'* and *W*) as :

$$\rho(W',W)=\frac{\sum_{k=1}^{L}W'(k)\times W(k)}{\sum_{k=1}^{L}\left\|W'(k)\right\|^{2}}=\frac{\sum_{k=1}^{L}W'(k)\times W(k)}{L} \qquad (11)$$

where, L is the length of the watermark signal. The Threshold Tρ is set so as to minimize the sum p of the probability of error detection and the probability of false alarm. If $\rho \geq$ Tρ, we considered the watermark is present, otherwise absent.

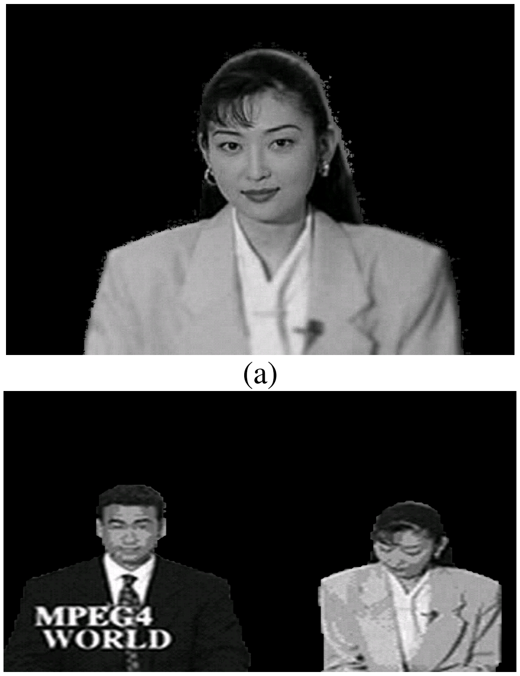
4. EXPERIMENTS RESULTS

We tested our scheme on a number of video (“akiyo”, ”news”, ”sean”) as show in Fig.4, we only report in detail results for ‘Akiyo’. In our experiments, the parameters considered are : the threshold Tρ = 0.1, β = 0.318, n = 5, N=8 ,wavelet-level = 3, wavelettype ='haar', L=1600 and scaling factor α Є [0.1, 0.5].

In order to test the performance of the proposed watermarking scheme, 200 watermarks were randomly generated.
The PSNR result between the original object and the watermarked object is 39.26 dB, As shown in Fig. 5, the watermark is perceptual invisible and the object with watermark appears visually identical to the object without watermark.

In Fig. 6 the absolute difference between the original object and the watermarked one, it is evident that there is no watermark embedded in the region outside the object. Fig. 7 shows the response of the watermark detector to 200 randomly generated watermarks of which only one matches the watermark present. The response to the correct watermark (i.e. number 100) is much higher than the responses to incorrect watermarks.

Added experiment results are described in details in the following and added experiment for other video are listed in table 2.


(a)

(b)

(c)

FIGURE 4: (a)Original Video (Object) Akiyo, (b)Original Video(Object) News, (c)Original Video(Object) Scene.

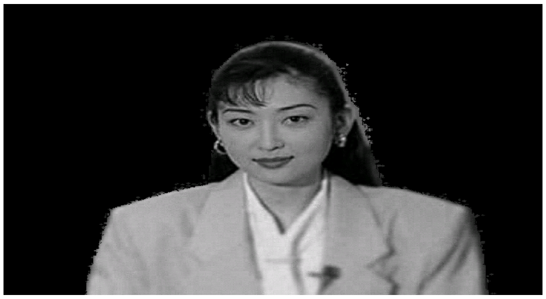


FIGURE 5: Watermarked Object Akiyo (PSNR=39.26 dB)

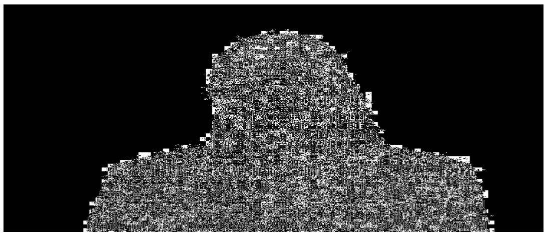


FIGURE 6: Absolute Difference Between the Original Object and The Watermarked

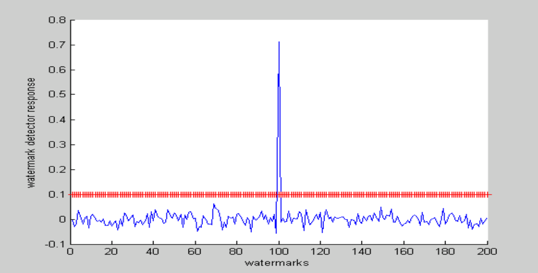


FIGURE 7: Detector Response of the Watermarked Object Akiyo for 200 Randomly Generated Watermark

4.1 MPEG-4 Compression

The watermarks are embedded in video objects, frame by frame, using MPEG-4 video object watermarking scheme, as shown in Fig. 3(a). The MPEG-4 video stream is next decompressed and two different objects are obtained as shown in Figs 8,9and 10, where the watermark detection process is applied, as shown in Fig. 3(b). The watermark detector responses of the decoded foreground objects of akiyo sequence are 0.4735 (foreground) and 0.8834(background), as shown in Fig. 11 and 13. The responses are well above the threshold Tρ and indicate that our proposed watermarking scheme is robust to MPEG-4 compression.

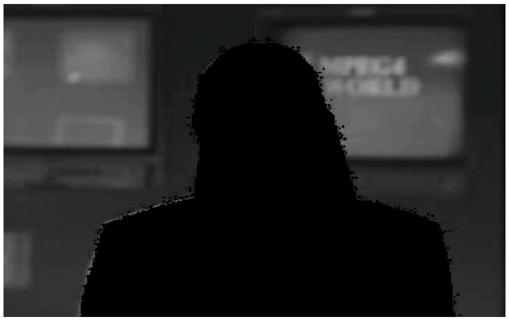


FIGURE 8: A Frame of The Video Sequence 'Akiyo' The Video Object 3 'Background'.

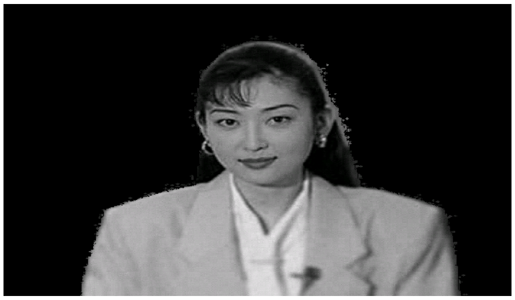


FIGURE 9: A Frame of The Video Sequence 'Akiyo' The Video Object 3 'Foreground'.

FIGURE 10 : A Frame of The Video Sequence 'Akiyo' The Video Object 3.

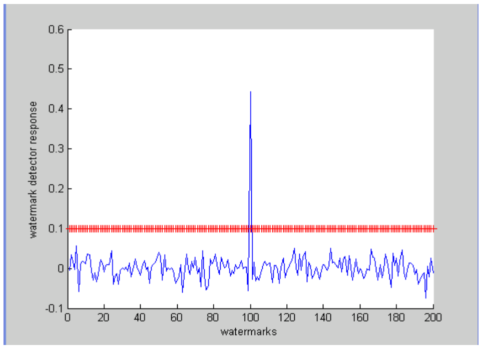


FIGURE 11: Watermark Detection Response Relating To The Video Object 3(Foreground) After MPEG-4 Compression.

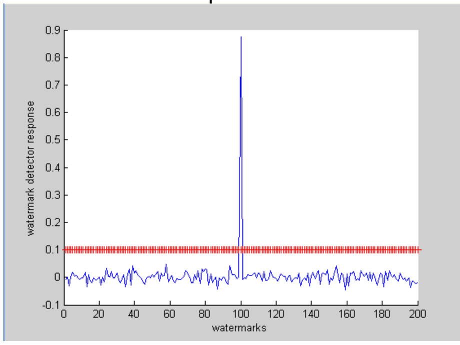


FIGURE 12 : Watermark Detection Response Relating To The Video Object 3(Background) After MPEG-4 Compression.

4.2 Format Conversion from MPEG-4 to MPEG-2

The watermarked MPEG-4 video bitstream is decompressed and frames are obtained. These frames are compressed MPEG-2 coded video bitstream using AVS video converter 6.2 and Easy video converter V.4.2. Next, the MPEG-2 coded video bitstream is decompressed, and each frame is separated so different objects are obtained, where the watermark detection process is applied. As shown in Fig.13 and 14, both watermarks embedded in the two objects are easily detected which indicates that the proposed scheme is robust to conversion from MPEG-4 to MPEG-2.

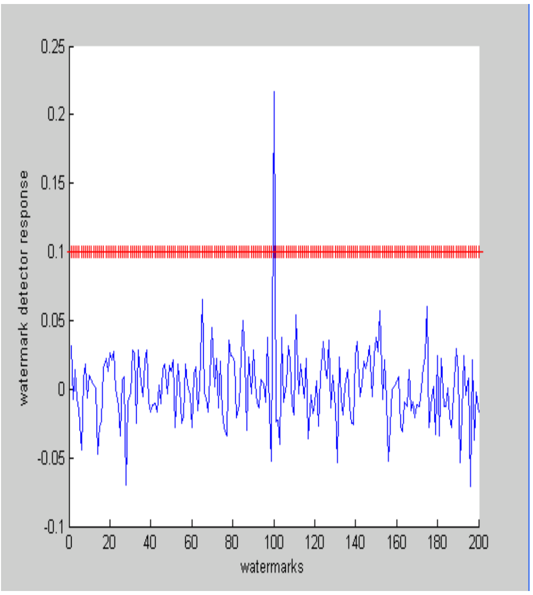


FIGURE 13 : Watermark Detection Response Relating To The Video Object 0 (Foreground) After Format Conversion From MPEG-4 To MPEG-2.

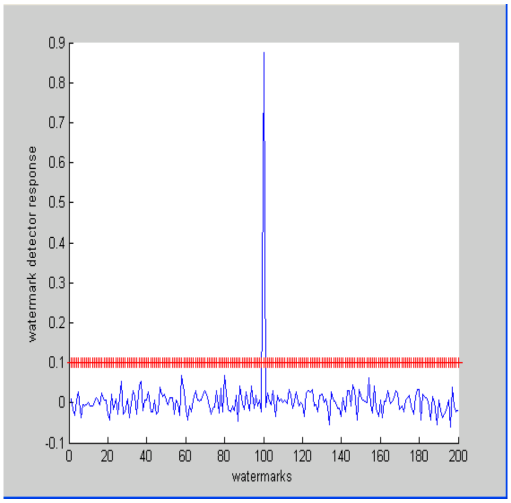


FIGURE 14: Watermark Detection Response Relating To The Video Object 1 (Background) After Format Conversion From MPEG-4 To MPEG-2.

Detector responses	akiyo	News	Sean
No attack(foreground)	0.7101	0.7212	0.7131
MPEG-4 Compression (foreground)	0.4506	0.5535	0.5944
MPEG-4 Compression (background)	0.8723	0.8077	0.8606
MPEG-4 to MPEG-2 (foreground)	0.2376	0.3253	0.3962
MPEG-4 to MPEG-2 (background)	0.8491	0.8652	0.8245
MPEG-1(foreground)	0.4231	0.4179	0.5481
MPEG-1(background)	0.8296	0.8019	0.8022
H264(foreground)	0.4673	0.4798	0.4022
H264(background)	0.8024	0.8266	0.8070

TABLE 2: Watermark Detector Responses After Attacks

5. CONSLUSION

In this article, a novel blind object watermarking scheme for MPEG4 video streams using the SA-DWT has been proposed. To make the watermark robust and transparent, we embed it in the average of the wavelet blocks using visual model. The visual model takes into account sensitivity to brightness and texture. Efficiency of the method is revealed on the basis of the following results:

(1) The average has a smaller change than that of individual coefficient. Thus, unlike most watermarking schemes, the watermark is not embedded by just an individual wavelet coefficient but by modulating the average of the wavelet blocks.

(2) Visual model allowed to achieve the best tradeoff between transparency and robustness.

(3) Watermark detection is accomplished without the original.

(4) Many parameters can be used as private key to that they are unknown to public.

6. REFERENCES

1. MPEG Requirements Group. "*MPEG-4 requirements*". Doc. ISO/IEC JTC1/SC29/WG11 N1595, Sevilla MPEG meeting, February 1997

2. F. Hartung and M. Kutter. "*Multimedia watermarking techniques*". In Proceedings of the IEEE, vol. 87, no. 7,pp. 1079–1107, july 1999.

3. X. Wu, W. Zhu, Z. Xiong and Y. Zhang. "Object-*based multiresolution watermarking of images and video*". In Proceedings of ISCAS'2000, Geneva, Switzerland, pp. 212–215, May 23–31, 2000

4. G.Y. Kim, J. Lee and C.S. Won. "*Object-based video watermarking*". In Proceedings of ICCE'99, pp. 100–101, June 22–24, 1999.

5. A. Piva, R. Caldelli and A.D. Rosa. "*A DWT-based object watermarking system for MPEG-4 video streams*". In Proceedings of ICIP'2000,Vancouver, Canada, September 2000, vol. III, pp. 5–8, 2000.

6. M. Barni, F. Bartolini, V. Capelini, and N. Checacci. "*Object Watermarking for MPEG-4 video streams copyright protection*". In Proceedings of IST/SPIE's : Security and Watermarking of Multimedia Content II, SPIE Proceedings, San Jose, CA, vol 3971, pp. 465-476, 2000

7. K. Xiangwei, L. Yu, L. Huajian and Y. Deli. “*Object watermarks for digital images and video*”. Image and Vision Computing, Vol.22, No.8, pp.583-595, Aug.2004.

8. M .Karl, L.Rastislav and, N. P. Konstantinos. “ SPIHT-based Coding of the Shape and Texture of Arbitrarily-Shaped Visual Objects”. Circuits and Systems for Video Technology, IEEE Transactions on Volume 16, Issue 10, pp:1196 – 1208 Oct. 2006

9. S. Li and W. Li. “*Shape-adaptive discrete wavelet transforms for arbitrarily shaped visual object coding*”. In Proceedings of IEEE Trans. Circuits Syst. Video Technol., vol. 10, pp. 725–743, Aug. 2000.

10. W. Sweldens. “*The lifting scheme: A new philosophy in biorthogonal wavelet constructions*”. In Wavelet Applications in Signal and Image Processing III, A. F. Laine and M. Unser, Eds. Proc. SPIE 2569, pp. 68–79, 1995.

HIGH CAPACITY AND SECURITY STEGANOGRAPHY USING DISCRETE WAVELET TRANSFORM

H S Manjunatha Reddy manjunathareddyhs @rediffmail.com

Dept. of Electronics and Communication
Global Academy of Technology, Bangalore, India-560098

K B Raja raja_kb@yahoo.com

Dept. of Computer Science and Engg
University Visvesvarya College of Engg,
Bangalore University, Bangalore-01

Abstract

The secure data transmission over internet is achieved using Steganography. In this paper High Capacity and Security Steganography using Discrete wavelet transform (HCSSD) is proposed. The wavelet coefficients of both the cover and payload are fused into single image using embedding strength parameters alpha and beta. The cover and payload are preprocessed to reduce the pixel range to ensure the payload is recovered accurately at the destination. It is observed that the capacity and security is increased with acceptable PSNR in the proposed algorithm compared to the existing algorithms

Keywords: Steganography, Wavelet Fusion, Security, Embedding strength parameters, Imperceptibility.

1. INTRODUCTION

The development in technology and networking has posed serious threats to obtain secured data communication. This has driven the interest among computer security researchers to overcome the serious threats for secured data transmission. One method of providing more security to data is information hiding. The approach to secured communication is cryptography, which deals with the data encryption at the sender side and data decryption at the receiver side. The main difference between steganography and cryptography is the suspicion factor. The steganography and cryptography implemented together, the amount of security increases. The steganography make the presence of secret data appear invisible to eaves droppers such as key loggers or harmful tracking cookies where the users keystroke is monitored while entering password and personal information. The Steganography is used for secret data transmission. Steganography is derived from the Greek word steganos which means “covered” and graphia which means “writing”, therefore Steganography means “covered writing”. In steganography the secret image is embedded in the cover image and transmitted in such a way that the existence of information is undetectable. The digital images, videos, sound files and other computer files can be used as carrier to embed the information. The object in which the secret information is hidden is called covert object. Stego image is referred as an image that is obtained by embedding secret image into covert image. The hidden message may be plain text, cipher text or images etc. The steganography method provides embedded data in an imperceptible manner with high payload capacity. Encrypting data provides data confidentiality, authentication, and data integrity.

Steganography, copyright protection for digital media and data embedding are the data hiding techniques. Steganography is a method of hiding secret information using cover images. Copyright marking classified into watermarking and fingerprinting. Watermarking is the process of possibly irreversibly embedding information into a digital signal. The signal may be audio, pictures or video etc. Fingerprinting attaches a serial number to the copy of digital media. Copyright protection prevents illegal transfer of data. In data embedding systems the receiver will know about the hidden message and the task is to decode the message efficiently. The main aspect of steganography is to achieve high capacity, security and robustness. Steganography is applicable to (i) Confidential communication and secret data storing, (ii) Protection of data alteration, (iii) Access control system for digital content distribution, (iv) Media Database systems etc.

The various steganographic techniques are: (i) Substitution technique: In this technique only the least significant bits of the cover object is replaced without modifying the complete cover object. It is a simplest method for data hiding but it is very weak in resisting even simple attacks such as compression, transforms, etc. (ii)Transform domain technique: The various transform domains techniques are Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and Fast Fourier Transform (FFT) are used to hide information in transform coefficients of the cover images that makes much more robust to attacks such as compression, filtering, etc. (iii) Spread spectrum technique: The message is spread over a wide frequency bandwidth than the minimum required bandwidth to send the information. The SNR in every frequency band is small. Hence without destroying the cover image it is very difficult to remove message completely. (iv)Statistical technique: The cover is divided into blocks and the message bits are hidden in each block. The information is encoded by changing various numerical properties of cover image. The cover blocks remain unchanged if message block is zero. (v) Distortion technique: Information is stored by signal distortion. The encoder adds sequence of changes to the cover and the decoder checks for the various differences between the original cover and the distorted cover to recover the secret message.

Steganalysis is the science of detecting hidden information. The main objective of Steganalysis is to break steganography and the detection of stego image is the goal of steganalysis. Almost all steganalysis algorithms rely on the Steganographic algorithms introducing statistical differences between cover and stego image. Steganalysis deals with three important categories: (a) Visual attacks: In these types of attacks with a assistance of a computer or through inspection with a naked eye it reveal the presence of hidden information, which helps to separate the image into bit planes for further more analysis. (b) Statistical attacks: These types of attacks are more powerful and successful, because they reveal the smallest alterations in an images statistical behavior. Statistical attacks can be further divided into (i) Passive attack and (ii) Active attack. Passive attacks involves with identifying presence or absence of a covert message or embedding algorithm used etc. Mean while active attacks is used to investigate embedded message length or hidden message location or secret key used in embedding. (c) Structural attacks: The format of the data files changes as the data to be hidden is embedded; identifying this characteristic structure changes can help us to find the presence of image.

2. RELATED WORK

Neil F. Johnson and sushil jajodia et al., [1] have provided several characteristics in information hiding methods to identify the existence of a hidden messages and also identify the hidden information. The images are reviewed manually for hidden messages and steganographic tool to automate the process. The developed tool is to test robustness of information hiding techniques in images such as warping, cropping rotating and blurring. Lisa M. Marvel and Charles T. Retter [2] have presented a method of embedding information within digital images, called Spread Spectrum Image Steganography (SSIS). SSIS conceals a message of substantial length with in digital images while maintaining the original image size and dynamic range. A hidden message can be recovered using the appropriate keys without any knowledge of the original image. Giuseppe Mastronardi et al., [3] have studied the effects of Steganography in different image formats (BMP, GIF, JPEG and DWT) and proposed two different approaches for lossless and lossy image. They are based on the creation of an “adhoc” palette for BMP and GIF images. LUI Tong and QIU Zheng-ding [4] have proposed a Quantization-based Steganography scheme. In this method the secret message is hidden in every chrominance component of a color image and the hiding capacity is higher than that of the popular Steganography software. Since the

Quantization-based hiding method is free from the interference and simulation results the hidden message can be extracted at low BER and our scheme is robust to common attacks.

Jessica Fridrich et al., [5] have proposed a new higher-order Steganalytic method called Pairs Analysis for detection of secret messages embedded in digital images. Although the approach is in principle applicable to many different Steganographic methods as well as image formats, it is ideally suited to 8-bit images, such as GIF images, where message bits are embedded in LSBs of indices to an ordered palette. The Ezstego algorithm with random message and optimized palette order is used as an embedding archetype on which we demonstrate Pairs Analysis and compare its performance with the chi-square attacks. Jessica Fridrich and David Soukal [6] have presented two approaches to matrix embedding for large payloads suitable for practical steganographic schemes – one based on family of codes constructed from simplex codes and the second one based on random linear codes for small dimension .The embedding efficiency of the proposed methods is evaluated with respect to theoretically achievable bounds. Yuan-Yu Tsai and Chung-Ming Wang [7] have proposed a novel data hiding scheme for color images using a BSP tree. This method shows high capacity with little visual distortion. Furthermore, there is an advantage of the tree data properties to improve the security of embedding process, making it difficult to extract the secret message without the secret key provided. Jun Zhang et al., [8] have proposed detection of steganographic algorithms based on replacement of the Least Significant Bit (LSB) plane. Since LSB embedding is modeled as an additive noise process, detection is especially poor for images that exhibit high-frequency noise.

M. Mahdavi et al., [9] presented a steganalysis method for the LSB replacement. The method is based on the changes that occur in histogram of an image after the embedding of data. It is less complex and more accurate than the RS steganalytic method for the images which are acquired directly from scanner without any compression. The RS method needs to count the number of regular and singular groups twice and also require LSB flipping for the whole image. This method has better average and variance of error comparing to RS steganalytic method. Shilpa p. Hivrale et al., [10] have presented various statistical measures and PMF based method of detection. It uses the frequency count of the pixel intensities in the image to test for the detection of stego image or not. Here LSB embedding technique is used. K. B. Raja et al., [11] have proposed a novel image adaptive stegnographic technique in the integer wavelet transform domain called as the Robust Image Adaptive Steganography using Integer Wavelet Transform. According to information theoretic prescriptions for parallel Gaussian models of images, data should be hidden in low and mid frequencies ranges of the host image, which have large energies. Jan Kodovsky and Jessica Fridrich [12] worked out the specific design principles and elements of steganographic schemes for the JPEG format and their security. The detect ability is evaluated experimentally using a state of art blind steganalyser. L.Y. Por et al., [13] have proposed a combination of three different LSB insertion algorithms on GIF image through stegcure system. The unique feature about the stegcure is being able to integrate three algorithms in one Steganography system. By implementing public key infrastructure, unauthorized user is forbidden from intercepting the transmission of the covert data during a communication because the stego-key is only known by the sender and the receiver. Gaetan Le Guelvoit [14] proposed a work which deals with public- key Steganography in presence of passive warden. The main aim is to hide the secret information within cover documents without giving the warden any clue and without any preliminary secret key sharing. This work explores the use of trellis coded quantization technique to design more efficient public key scheme.

Mohammad Ali Bani Younes and Aman Jantan [15] have proposed a steganographic approach for data hiding. This approach uses the least significant bits (LSB) insertion to hide data within encrypted image data. The binary representation of the data is used to overwrite the LSB of each byte within the encrypted image randomly. The hidden data will be used to enable the receiver to reconstruct the same secret transformation table after extracting it and hence the original image can be reproduced by the inverse of the transformation and encryption processes. Chang-Chu Chen and Chin-Chen Chang [16] have proposed that data hiding scheme is a modification of the LSB-based steganography using the rule of reflected gray code. The embedding ability and distortion level of our novel method are similar to those of the simple LSB substitution scheme. The difference is that the LSBs of stego-image are not always the same as the secret bits while the simple LSB substitution keeps them equally. Babita Ahuja and, Manpreet Kaur [17] have presented LSB based steganography algorithm with high data hiding capacity, as four LSB's are used to hide data, high confidentiality as distortions which can cause suspiscions for the intruders, are removed through filtering techniques and two level high security is applied. Debnath Bhattacharyya et al., [18] a security model is proposed which imposes the concept of secrecy over privacy for text messages. The proposed model combines cryptography, steganography and along with an extra layer of security has been imposed in between them. Chin-Chen Chang et al.,[19] proposed a scheme embeds a larger-sized secret image while maintaining acceptable image quality of the stego-image and also improved image hiding scheme for grayscale images based on wet paper coding.

3 MODEL

The definitions, Wavelet Transform and HCSSD model are described in this section.

Definitions:

- *Cover Image*: It is defined as the original image into which the required secret message is embedded. It is also termed as innocent image or host image. The secret message should be embedded in such a manner that there are no significant changes in the statistical properties of the cover image. Good cover images range from gray scale image to colored image in uncompressed format.
- *Payload:* It is the secret massage that has to be embedded within the cover image in a given Steganographic model. The payload can be in the form of text, audio, images, and video.
- *Stego image:* It is the final image obtained after embedded the payload into a given cover image. It should have similar statistical properties to that of the cover image.
- *Hiding Capacity*: The size of information that can be hidden relative to the size of the cover without deteriorating the quality of the cover image.
- *Robustness:* The ability of the embedded data to remain intact if the stego image undergoes transformation due to intelligent stego attacks.
- *Security*: This refers to eavesdropper's inability to detect the hidden information.
- *Mean Square Error (MSE)*: It is the measure used to quantify the difference between the initial and the distorted or noisy image. Let *Pi* represents the pixel of one image of size N and Q_i that of the other.

$$MSE=\sum_{i=1}^{allpixels}\sum_{i=1}^{allpixels}\frac{\left(Cover(i,j)-stego(i,j)\right)^2}{N\times N} \quad (1)$$

From MSE we can find Peak Signal to Noise Ratio (PSNR) to access the quality of the Stego image with respect to cover image given by

$$PSNR=20\log_{10}\frac{255}{\sqrt{MSE}} \quad (2)$$

- Haar Wavelet: It is a piecewise wavelet that provides orthogonal decomposition given as

$$\psi(t)=\begin{cases}+1, & if\ 0\le t\le 1/2\\ -1, & if\ 1/2\le t\le 1\\ 0, & otherwise\end{cases} \quad (3)$$

- Wavelet Transform: It converts an image from time or spatial domain to frequency domain. It provides a time-frequency representation. The Wavelet Transform is obtained by repeated filtering of the coefficients of the image row-by-row and column-by-column.
- *Approximation Band*: It is the band having the lower frequency coefficients of the image in the wavelet domain. It contains all the significant features of the image.
- *Detail Band*: It has high frequency components of the image in the wavelet domain and consists of insignificant features of the image.
- *Payload Encryption*: Encryption of payload is done not only to protect data frame theft or alteration, but can also be used for authentication and increase security level. Secret key cryptography is used, wherein the same key is used for both encryption and decryption.

- *Inverse Wavelet Transform*: It is applied over the stego image to convert it from frequency domain to spatial domain. Hence it is frequency-time representation.
- *Fusion*: It is the process of adding the wavelet coefficients of both the Cover Image and Payload.
- *Cover-escrow*: The scheme in which the original Cover image is required at the extraction model to get the Payload.
- *Normalization*: It is the division of all the pixel values of an image in the spatial domain with the maximum pixel value of the image. For gray scale image the maximum value of any pixel is 255.
- *Preprocessing*: All the pixels of an image in spatial domain are multiplied by embedding strength factors alpha or beta.

3.1 *Wavelet Transform*:

Wavelet transform is used to convert a spatial domain into frequency domain. The use of wavelet in image stenographic model lies in the fact that the wavelet transform clearly separates the high frequency and low frequency information on a pixel by pixel basis. Discrete Wavelet Transform (DWT) is preferred over Discrete Cosine Transforms (DCT) because image in low frequency at various levels can offer corresponding resolution needed. A one dimensional DWT is a repeated filter bank algorithm, and the input is convolved with high pass filter and a low pass filter. The result of latter convolution is smoothed version of the input, while the high frequency part is captured by the first convolution. The reconstruction involves a convolution with the synthesis filter and the results of this convolution are added. In two dimensional transform, first apply one step of the one dimensional transform to all rows and then repeat to all columns. This decomposition results into four classes or band coefficients.

The Haar Wavelet Transform is the simplest of all wavelet transform. In this the low frequency wavelet coefficient are generated by averaging the two pixel values and high frequency coefficients are generated by taking half of the difference of the same two pixels. The four bands obtained are approximate band (LL), Vertical Band (LH), Horizontal band (HL), and diagonal detail band (HH). The approximation band consists of low frequency wavelet coefficients, which contain significant part of the spatial domain image. The other bands also called as detail bands consists of high frequency coefficients, which contain the edge details of the spatial domain image.

Research into human perception indicates that the retina of the eye splits an image into several frequency channels, each spanning a bandwidth of approximately one octave. The single in these channels is processed independently. Similarly in a multilevel decomposition, the image is separated into bands of approximately equal bandwidth on a logarithmic scale. It is therefore expected that use of the DWT will allow independent processing of the resulting components without significant perceptible interaction between them, and hence makes the process imperceptibility marking more effective. For this reason the wavelet decompositions is commonly used for the fusion of images. Fusion technique include the simple method of pixel averaging to more complicated methods such as principal component analysis and wavelet transform fusion. Several approaches to image fusion can be distinguished; depending on whether the image is fused in the spatial domain or any other domains, and their transform fused. Image fusion is a process that produces a single image from a set of input images. The fused image contains more complete information, than any individual input. Since this is a sensor-compresses information problem, it follows that wavelets, classically useful for human visual processing, data compression and reconstruction are useful for such merging. Other important applications of the fusion of images include medical imaging, microscopic imaging, remote sensing, computer vision and robotics.

3.2 *High Capacity and Security Steganography using Discrete wavelet transform model (HCSSD)*

(i) HC*SSD Encoder:* Figure 1 shows the block diagram of the embedding algorithm. The main idea behind the proposed algorithm is wavelet based fusion. It involves merging of the wavelet decomposition of the normalized version of both the cover image and the payload into a single fused result. Normalization is done so that the pixel range of the image lies between 0.0 to 1.0 instead of the integer range (0, 255). Hence we convert the integer range (0, 255) of pixels into floating point values between 0.0 and 1.0. This normalized pixel values is fed as input to the floating point filters which results in reconstruction of the transformed image with better accuracy compared to direct integer values of the pixels as input. Normalization is a process on both the cover image and the payload in order to guarantee pixel values do not exceed their maximum value of one due to modifying corresponding coefficients of the cover image and payload during fusion. Both cover image and payload is convert into DWT domain. Further, apply DWT on the payload in order to increase the security level. The single fused resultant matrix is obtained, by the addition of wavelet coefficients of the respective sub-bands of the cover image and payload is given by the Equation (4).

$$F\ (x, y) = \alpha C(x, y) + \beta P(x, y) \tag{4}$$

$$\alpha + \beta = 1 \tag{5}$$

Where F is modified DWT coefficients, C is the original DWT coefficients and P is the approximation band DWT coefficients of the payload. Also alpha and beta are the embedding strength factors. Since alpha and beta are chosen such that the payload is not predominantly seen in the Stego

image obtained in the spatial domain and also for full utilization of the bandwidth of both the Cover Image and the payload. Once fusion is done, we apply Inverse Discrete Wavelet Transform (IDWT) followed by renormalization to get the Stego image in the spatial domain.

ii) HCSSD Decoder: Figure 2 shows the block diagram for retrieval of payload from the Stego image. The Stego image is normalized, and then DWT is taken. The extraction process involves subtracting the DWT coefficients of the original cover image from the DWT coefficients of the Stego image. It is then followed by decryption of the subtracted coefficients. Then first step of IDWT on these coefficients is applied followed by second IDWT only with respect to the approximation band of the first IDWT coefficients of the payload. Finally, denormalization is done to get back the payload in spatial domain.

4. ALGORITHM

4.1 Problem definition

Given a cover image *c* of size (n * m) and payload *p* of size (2n * 2m).

The objectives are:

(i) to embed the Payload into the Cover image in the wavelet domain.

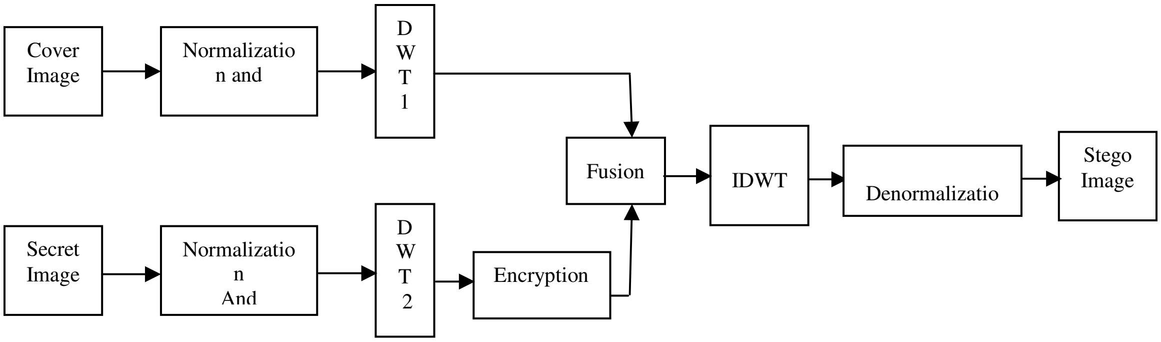


Fig: 1. HCSSD Encoder

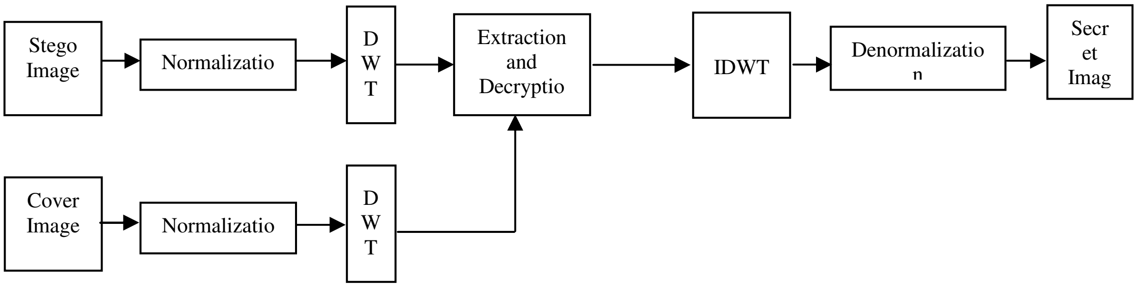


Fig: 2. HCSSD Decoder

(ii) to increase the embedding capacity into the Cover image.

(iii) to ensure reasonable PSNR of the Stego mage.

Assumptions:

Cover and payload images are grayscale uncompressed images, i.e., color images are converted into grayscale images.

(ii) Haar wavelet Transform is used to convert spatial domain image to wavelet domain.

Table 1 gives the HCSSD Encoder algorithm. The algorithm gives high security as encrypting the wavelet coefficients of Payload before embedding. The payload being double the size of the cover, the capacity is high due to the fact that we are embedding only the approximation band of the payload coefficients. For a cover image of size (n * m), when we apply two level DWT we get matrix dimension of size (n/2 * m/2), and we take a Payload of size (2n * 2m) and apply twice two level DWT, it gives matrix dimension of (n/2 *m/2). Since the matrix dimension of cover image and payload are same we are able to add their respective coefficients. Table II gives HCSSD Decoder to retrieve the Payload from the Stego image.

5. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

For performance analysis we considered the Cover Images (CI) such as Lady, Aero plane, Players, Cow boys and Flower. Payload images (PL) are Flower, Bank text, Astronauts, Dog and Elephant. The payload is embedded into the cover image to derive the Stego image at the sending end. The payload is recovered from the Stego image at the destination with minimum distortion. Fig. 3(a), 4(a), 5(a), 6(a) and 7(a) are the Cover Images (CI). Fig. 3(b), 4(b), 5(b), 6(b) and 7(b) are the Payload images (PL). Fig. 3(c), 4(c), 5(c), 6(c) and 7(c) are the Stego Images (SI). Fig. 3(d), 4(d), 5(d), 6(d) and 7(d) are the Retrieved Payload images (RPL). Table III shows the experimental results of the proposed HCSSD algorithm where the MSE, PSNR and Entropy between the cover image and Stego image are computed. The PSNR, MSE and Entropy are dependent on

image formats and sizes of the cover and Stego image. The Entropy values approximately equal to zero which indicates that the security of the payload is high. Since all the bits in the pixel of the cover image are used for fusion purpose, the embedding capacity reaches its maximum that is 8 bit for pixel for a gray scale cover image The size of the payload is twice to that of the cover image. Table IV shows the experimental results of existing Wavelet Based Fusion (WBF) algorithm, wherein the MSE, PSNR and Entropy between the cover image and Stego image are computed. From Table III and IV, we observed that the PSNR values of proposed algorithm are within the acceptable range along with higher capacity and highly secure as the Entropy value is approximately zero.

Table: 1 **ALGORITHM OF DATA EMBEDDING**

- Input: Cover Image *c* and Payload image *p*.
- Output: Stego image *s*.

1. Normalize *c* and *p*, so that the wavelet coefficient varies between 0.0 and 1.0.
2. Preprocessing on *c* and *p*
3. Transform *c* and *p* into 2 levels of decomposition using Haar Wavelet.
4. Apply 2 levels DWT on the approximate band of the payload obtained.
5. Encrypt the DWT coefficients obtained.
6. Wavelet fusion of DWT coefficients of *c* and *p*.
7. Inverse transform of all the subbands of the fused image.
8. Denormalize the Fused image.
9. Stego image *s* is generated.

Table: II **ALGORITHM OF DATA EXTRACTION**

- Input: Stego Image s.
- Output: Payload *p*.

1. Normalize Stego Image *s*.
2. Transform *s* in to 2 levels of wavelet decomposition.
3. Subtract DWT coefficients of *c* from DWT coefficients of *s* to get DWT coefficients of only *p*.
4. Decrypt the DWT coefficients of *p* obtained.
5. Apply IWT of all the sub bands of *p*.
6. Apply IWT of payload obtained with respect to approximate band.
7. Denormalize the resultant of step 6.
8. Payload Image is obtained *p*.

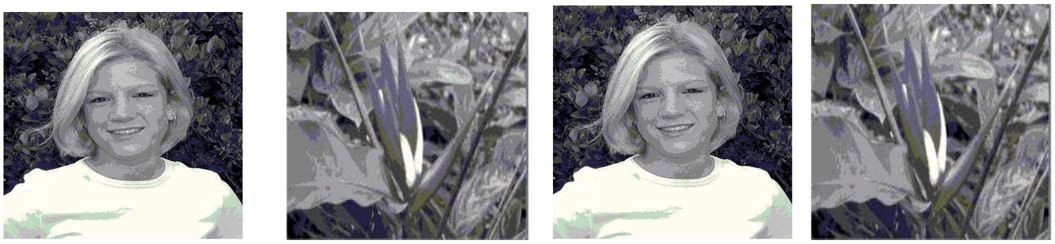
Table: III **MSE, PSNR and Entropy of Cover and Stego image of HCSSD**

Images	Type	Size	MSE	PSNR	Entropy
Lady Flower	JPEG JPEG	346×396 240×240	0.17	55.6	0.00019
Aero plane Bank Text	TIFF PNG	400×300 810×400	2.76	43.7\	0.0000
Player Astronauts	JPEG PNG	400×300 200×200	0.9	48.1	0.0004
Cow Boys Dog	JPEG TIFF	186×100 436×600	0.17	55.58	0.0000
Flower	JPEG	200×150	0.98	48.20	0.0000

Elephant	JPEG	335×219			

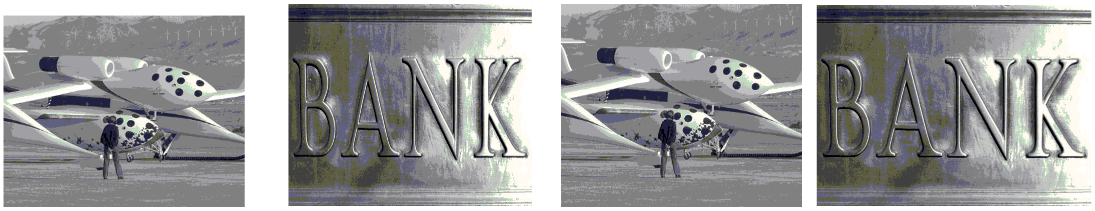
Table: IV **MSE, PSNR and Entropy of Cover and Stego image WBF**

Images	Type	Size	MSE	PSNR	Entropy
Lady Flower	JPEG JPEG	346×396 240×240	2.10	44.9	0.004
Aero plane BankText	TIFF PNG	400×300 810×400	0.41	51.9	0.0010
Player Astronauts	JPEG PNG	400×300 200×200	0.49	51.1	0.0060



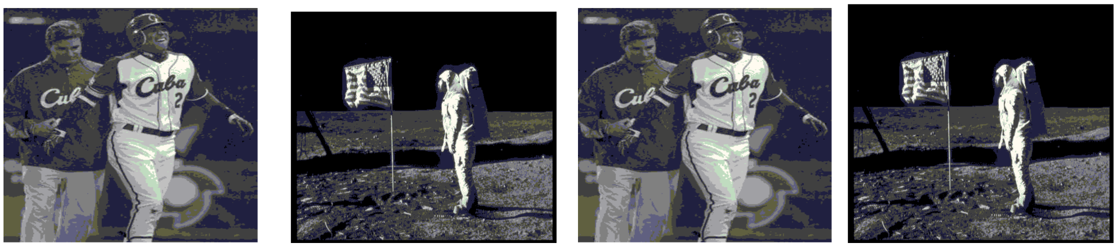
(a) Cover image (b) Payload Image (c) Stego Image (d) Retrieved Payload image

Fig. (3). Lady and Flower images



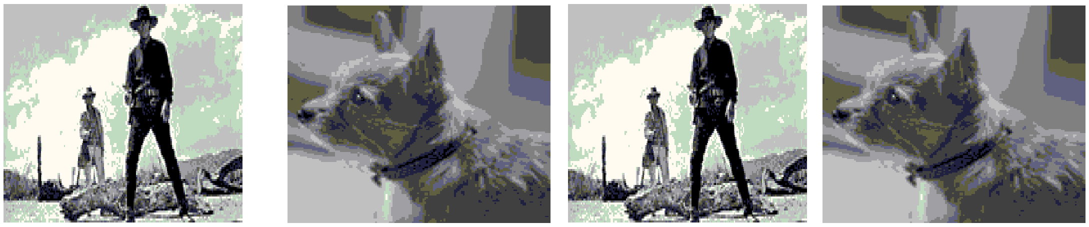
(a) Cover Image (b) Payload image (c) Stego Image (d) Retrieved Payload

Fig.4. Aero plane and Bank Text Images



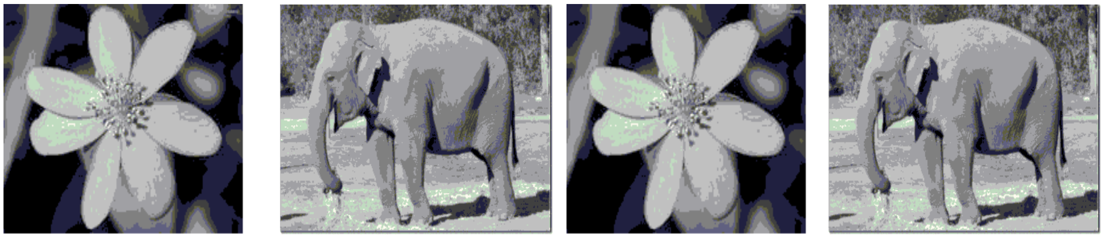
(a) Cover image (b) Payload Image (c) Stego Image (d) Retrieved payload

Fig.5. Player and Astronauts Images



(a) Cover image (b) Payload Image (c) Stego Image (d) Retrieved Payload image

Fig.6 Cow Boys and Dog Images



(a) Cover image (b) Payload Image (c) Stego Image (d) Retrieved Payload image

Fig.7 Flower and Elephant Images

6. CONCLUSIONS

The Steganography is used for secrete communication. In this paper High Capacity and Security Steganography using Discrete wavelet transform algorithm is proposed. The cover and payload are normalized and the wavelet coefficient is obtained by applying discrete wavelet transform. The approximation band coefficient of payload and wavelet coefficient of cover image are fused based on strength parameters alpha and beta. The capacity of the proposed algorithm is increased as the only approximation band of payload is considered. The Entropy, MSE and Capacity are improved with acceptable PSNR compared to the existing algorithm. In future the algorithm can be tested with curvelet transform and other transform techniques.

Contributions: In this paper the two level wavelet transform is applied as cover and payload. The payload wavelet coefficients are encrypted and fused with wavelet coefficients of cover image to generate stego coefficients based on the embedding strength parameters alpha and beta.

Acknowledgements : The first author thank full to the management and Principal Dr..Narendra Viswanath to support the entire research work.

7. REFERENCES

[1] Neil F. Johnson and Sushil Jajodia, "Steganalysis: The Investigation of Hidden Information," *IEEE conference on Information Technology*, pp. 113-116, 1998.

[2] Lisa M.Marvel and Charles T. Retter, "A Methodlogy for Data Hiding using Images," IEEE *conference on Military communication*, vol. 3, Issue. 18-21, pp. 1044-1047, 1998.

[3] Giuseppe Mastronardi, Marcello Castellano, Francescomaria Marino, "Steganography Effects in Various Formats of Images. A Preliminary Study," International Workshop on Intelligent data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 116-119, 2001.

[4] LIU Tong, QIU Zheng-ding "A DWT-based color Images Steganography Scheme" IEEE International Conference on Signal Processing, vol. 2, pp.1568-1571, 2002.

[5] Jessica Fridrich, Miroslav Goijan and David Soukal, "Higher-order statistical steganalysis of palette images" *Proceeding of SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia ContentsV*, vol. 5020, pp. 178-190, 2003.

[6] Jessica Fridrich and David Soukal, "Matrix Embedding for Large Payloads" *SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents* , vol. 6072, pp. 727-738. 2006.

[7] Yuan-Yu Tsai, Chung-Ming Wang "A novel data hiding scheme for color images using a BSP tree" *Journal of systems and software*, vol.80, pp. 429-437, 2007.

[8] Jun Zhang, Ingemar J. Cox and Gwenael Doerr.G "Steganalysis for LSB Matching in Images With High-frequency Noise" *IEEE Workshop on Multimedia Signal Processing*, issue 1-3, pp.385- 388, 2007.

[9] M. Mahdavi, Sh. Samavi, N. Zaker and M. Modarres-Hashemi, "Steganalysis Method for LSB Replacement Based on Local Gradient of Image Histogram," *Journal of Electrical and Electronic Engineering*, vol. 4, no. 3, pp. 59-70, 2008.

[10]Shilpa P. Hivrale, S. D. Sawarkar, Vijay Bhosale, and Seema Koregaonkar "Statistical Method for Hiding Detection in LSB of Digital Images: An Overview *World Academy of Science, Engineering and Technology*, vol. 32, pp. 658-661, 2008.

[11]K. B. Raja, S. Sindhu, T. D. Mahalakshmi, S. Akshatha, B. K. Nithin, M. Sarvajith, K. R. Venugopal, L. M. Patnaik, "Robust Image Adaptive Steganography using Integer Wavelets" *International conference on Communication Systems Software*, pp. 614-621, 2008.

[12]Jan Kodovsky, Jessica Fridrich "Influence of Embedding Strategies on Security of Steganographic Methods in the JPEG Domain" *Proceedings of SPIE, the International Society for Optical Engineering*, vol. 6819, pp. 681902.1-681902.13, 2008.

[13] L. Y. Por, W. K. Lai, Z. Alireza, T. F. Ang, M. T. Su, B. Delina, "StegCure: A Comprehensive Steganographic Tool using Enhanced LSB Scheme," *Journal of WSEAS Transctions on Computers*, vol. 8, pp. 1309-1318, 2008.

[14]Gaetan Le Guelvouit, "Trellis-Coded Quantization for Public-Key Steganography," *IEEE International conference on Acostics, Speech and Signal Processing*, pp.108-116, 2008.

[15]Mohammed Ali Bani Younes and Aman Jantan, "A New Steganography Approach for Images Encryption Exchange by Using the Least Significant Bit Insertion," *International Journal of Computer Science and Network Security*, vol. 8, no. 6, pp.247-257, 2008.

[16]Chang-Chu Chen, and Chin-Chen Chang, "LSB-Based Steganography Using Reflected Grey Code," *The Institute of Electronics, Information and communication Engineers Transaction on Information and System,",* vol. E91-D (4), pp. 1110-1116, 2008.

[17]Babita Ahuja and, Manpreet Kaur, "High Capacity Filter Based Steganography," *International Journal of Recent Trends in Engineering*, vol. 1, no. 1, pp.672-674, May 2009.

[18]Debnath Bhattacharyya, Poulami Das, Samir kumar Bandyopadhyay and Tai-hoon Kim, "Text Steganography: A Novel Approach," *International Journal of Advanced Science and Technology,* vol.3, pp.79-85, February2009.

[19]Chin- Chen Chang, Yung- Chen Chou and Chia- Chen Lin, "A steganography scheme based on wet paper codes suitable for uniformly distributed wet pixels," *IEEE International Symposium on circuits and Systems,* pp. 501-504, 2009.

Hybrid Compression Encryption Technique for Securing SMS

Tarek M. Mahmoud Tarek@minia.edu.eg
Faculty of science/ Department
of Computer Science
Minia University
El Minia, Egypt

Bahgat A. Abdel-latef Dr_bahgat2005@yahoo.com
Faculty of science/ Department
of Computer Science
Minia University
El Minia, Egypt

Awny A. Ahmed awny_ahmed70@yahoo.com
Faculty of science/ Department
of Computer Science
Minia University
El Minia, Egypt

Ahmed M. Mahfouz AhmedMahfouz@minia.edu.eg
Faculty of science/ Department
of Computer Science
Minia University
El Minia, Egypt

Abstract

Mobile communication devices have become popular tools for gathering and disseminating information and data. When sensitive information is exchanged using SMS, it is crucial to protect the content from eavesdroppers as well as ensuring that the message is sent by a legitimate sender. Using an encryption technique to secure SMS data increases its length and accordingly the cost of sending it. This paper provides a hybrid compression encryption technique to secure the SMS data. The proposed technique compresses the SMS to reduce its length, then encrypts it using RSA algorithm. A signature is added to the encrypted SMS for signing it to differentiate it from other SMS messages in SMSINBOX. The experimental results which are based on Symbian OS show that the proposed technique guarantees SMS data security without increasing its size.

Keywords: Mobile Communication Devices, Short Message Service, compression, encryption, Symbian Operating System

1. INTRODUCTION

Mobile communication devices have become commonplace during the past few years, integrating multiple wireless networking technologies to support additional functionality and services. One of the most important developments that have emerged from communications technology is SMS. It was designed as part of Global System for Mobile communications (GSM), but is now available on a wide range of network standards such as the Code Division Multiple Access (CDMA) [1]. Although SMS was originally meant to notify users of their voicemail messages, it has now become a popular means of communication by individuals and businesses. Banks worldwide are using SMS to conduct some of their banking services. For example, clients are able to query their bank balances via SMS or conduct mobile payments. Also, people sometimes exchange confidential information such as passwords or sensitive data amongst each other [2].

SMS technology suffers from some risks such as vulnerabilities, eavesdroppers and unauthorized access [3]. So, we need to find a solution to ensure that these SMS messages are secure and their contents remain private, without increasing their lengths.

This paper provides a solution to this SMS security problem. Our approach is to secure the SMS message using Hybrid Compression Encryption (HCE) system. The proposed technique compresses the SMS to reduce its length, then encrypts it using RSA algorithm. A signature is added to the encrypted SMS for signing it to differentiate it from other SMS messages in SMSINBOX.

This paper is structured as follows: Section 2 gives an overview of Short Message Service (SMS). Section 3 provides some details of SMS security. The Proposed Technique used for Securing SMS is introduced in section 4. Section 5 shows our experimental results. Finally, conclusion and future work are presented in section 6.

2. Short Message Service (SMS)

SMS is a communication service standardized in the GSM mobile communication systems; it can be sent and received simultaneously with GSM voice, data and fax calls. This is possible because whereas voice, data and fax calls take over a dedicated radio channel for the duration of the call, short messages travel over and above the radio channel using the signaling path [4]. Using communications protocols such as Short Message Peer-to-Peer (SMPP) [5] allow the interchange of short text messages between mobile telephone devices as shown in Figure 1 that describe traveling of SMS between parties.

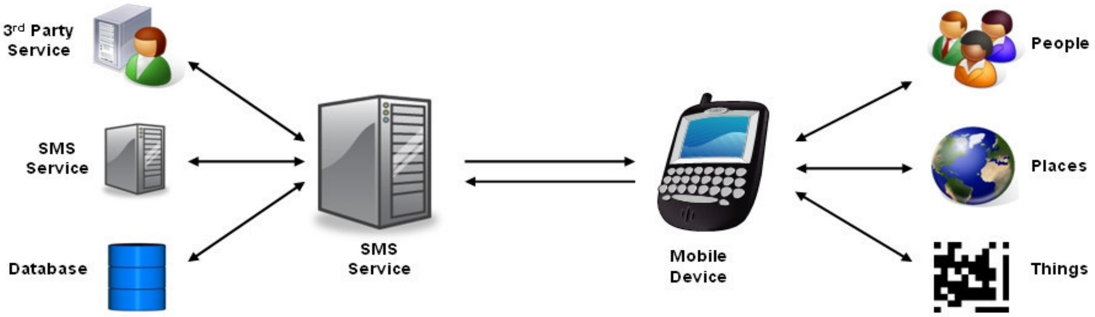


FIGURE 1: The basic of SMS system.

SMS contains some meta-data [6]:

- Information about the senders (Service center number, sender number)
- Protocol information (Protocol identifier, Data coding scheme)
- Timestamp

SMS messages do not require the mobile phone to be active and within range, as they will be held for a number of days until the phone is active and within range. SMS are transmitted within the same cell or to anyone with roaming capability. The SMS is a store and forward service, and is not sent directly but delivered via an SMS Center (SMSC). SMSC is a network element in the mobile telephone network, in which SMS is stored until the destination device becomes available. Each mobile telephone network that supports SMS has one or more messaging centers to handle and manage the short messages [4].

SMS message packets are simple in design. The structure of SMS packet is shown in Figure 2 [2].

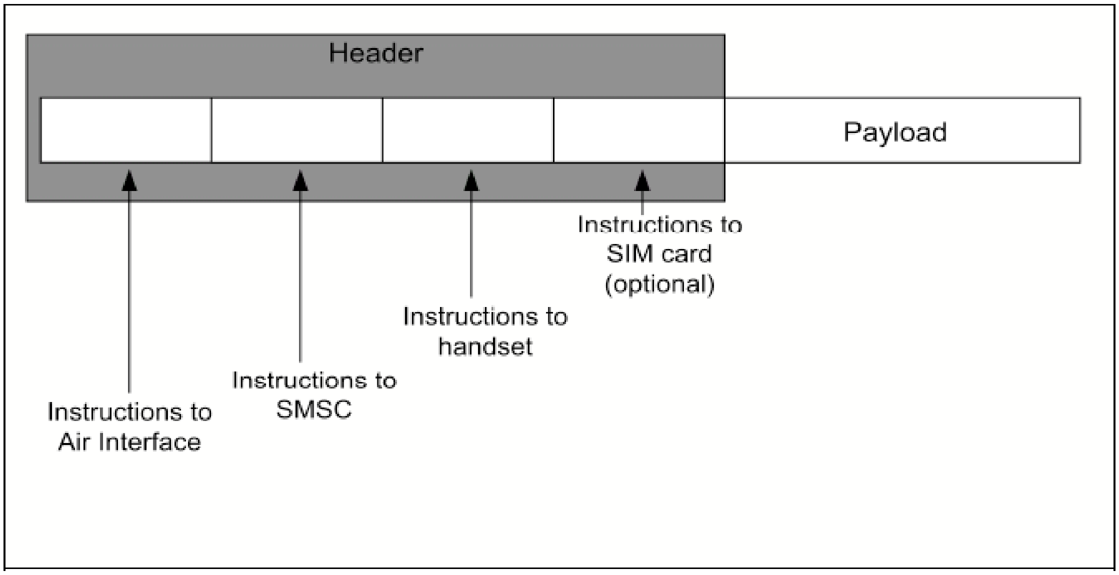

Header
Payload
Instructions to SIM card (optional)
Instructions to handset
Instructions to SMSC
Instructions to Air Interface

FIGURE 2: SMS Message structure

An SMS comprises of the following elements, of which only the user data is displayed on the recipient's mobile device:

- Header - identifies the type of message:
  - Instruction to Air interface
  - Instruction to SMSC
  - Instruction to Phone
  - Instruction to SIM card
- User Data - the message body (payload).

As shown in Table 1, each SMS is up to 140 bytes, which represents the maximum size of SMS, and each short message is up to 160 characters in length when Latin alphabets are used, where each character is 7 bits according to the 7-bit default alphabet in Protocol Data Unit (PDU) format, and 70 characters in length when non-Latin alphabets such as Arabic and Chinese are used, where 16-bit messages are used [7] [8].

Coding scheme	Text length per message segment
GSM alphabet, 7 bits	160 characters
8-bit data	140 octets

USC2, 16 bits	70 complex characters

TABLE 1: Relation between coding scheme and text length.

3. SMS security

SMS travels as plain text and privacy of the SMS contents cannot be guaranteed, not only over the air, but also when such messages are stored on the handset. The contents of SMS messages are visible to the network operator's systems and personnel. The demand for active SMS based services can only be satisfied when a solution that addresses end-to-end security issues of SMS technology is available, where primary security parameters of authentication, confidentiality, integrity and non-repudiation are satisfied [9,13].

Authentication is concerned with only specific users with specific combination of device, application, memory card, and SIM card that are allowed to access corporate data. This way the users or unauthorized persons cannot change any part of the combination to obtain access to sensitive data. Confidentiality is about ensuring that only the sender and intended recipient of a message can read its content. Integrity is concerned with ensuring that the content of the messages and transactions not being altered, whether accidentally or maliciously. Non-repudiation is about providing mechanisms to guarantee that a party involved in a transaction cannot falsely claim later that he/ she did not participate in that transaction[14].

An end-to-end key based encryption technology for SMS plugs the gaps in transit security of SMS. Authentication added for resident SMS security access together with encryption, addresses the confidentiality issue of SMS technology. Added features of message integrity and digital signing of SMS address integrity and Non Repudiation for SMS technology[15].

4. The Proposed Technique for Securing SMS

In this section, we describe the proposed technique used to secure SMS without increasing its length. The two main steps of this technique are the compression and encryption processes. SMS Compression is the process of encoding SMS information using fewer bits than an unencoded representation. The purpose of this step in the proposed technique is reducing the consumption of expensive resources and reducing SMS length. SMS encryption is the art of achieving security by encoding messages to make them non-readable.

The steps of the proposed technique can be described as follows:

Step 1: Get SMS.
Step 2: Determine the SMS recipient.
Step 3: Compress the SMS.
Step 4: Check the compressed SMS length.
    4.1 If it is greater than 145 characters then divide it into more than one according to its length such that each message is 145 characters to satisfy the message length limit imposed by the proposed technique.
Step 5: Encrypt the compressed SMS using RSA algorithm.
Step 6: Add signature to the SMS.
Step 7: Send the SMS.

In Step 4, restricting the SMS length in the proposed technique to 145 characters is necessary for the encryption process. We have conducted many experiments to determine the length of SMS cipher (encrypted) text. Table 2 illustrates the experimental results for the relation between the RSA Modulus bits, maximum number of SMS plain text and length of output encrypted characters. According to these results, we selected the RSA Modulus size to be 1248 bits as optimal value for the proposed technique, so the output cipher text will be 156 characters and the maximum input characters will be 145. As mentioned in section 2, the standard SMS length is 160 characters.

RSA Modulus Size (bits)	Number of Input Characters Range	Length of Output Encrypted Character
256	1 – 21	32
512	1 – 53	64
1024	1 – 117	128
1248	1 – 145	156
2048	1 – 245	256

TABLE 2: The relation between RSA Modulus bits, maximum number of Input characters and length of output encrypted characters

In step 5, encrypting the SMS is based on RSA algorithm [10] [11].
The steps of this algorithm can be described as follows:

Step 1: choose two large primary numbers P and Q
Step 2: calculate N=P*Q
Step 3: select the public key (i.e. the encryption key) E, such that it is not a factor of (P-1) and (Q-1)
Step 4: Select the private key (i.e. the decryption key) D, such that the following equation is true (D*E) mod (P-1) * (Q-1) =1
Step 5: For encryption, calculate the cipher text CT from the plain text PT as follows CT=PT^E mod N
Step 6: Send CT as the cipher text to the receiver
Step 7: For decryption, calculate the plain text PT from the cipher text CT as follows PT=CT^D mod N

Figure 3 illustrates the SMS format after applying the proposed technique. It contains 4 characters as a signature and 156 characters as encrypted SMS data.

4 Signature	156 Cipher text

FIGURE 3: SMS Format after applying the proposed technique.

5. Experimental Results

This section presents the results of evaluating the efficiency of the proposed technique that is based on Symbian OS [12]. We consider the SMS length as a criterion to evaluate the performance of the proposed technique. The main purpose of the proposed technique is to secure SMS. We achieved this by compressing the SMS data to reduce its length then encrypting it to guarantee its security.

Table 3 shows a comparison between SMS length before and after the compression step. The 1st column contains some SMS samples, the 2nd column represents the total number of SMS characters before the compression process, and the 3rd column contains the total number of SMS characters after compression.

SMS Sample	Total number of SMS characters before compression	Total number of SMS characters after compression
#There are "men" like mountains "high" friend "honor" comradely "warranty" communicate with them "right and duty of the length of time" forgotten "impossible".	160	125
Source, Name : ahmed Mahfouz Password : 02034112 Card Number : 2400139 Account Number : 0111149 Operation : withdrawal Value : 1000$ Destination, Name : MobiTech Account Number : 0111133	185	142
Dear Sir this data are important for you so take your precautions-- --name : ahmed Muhammad balance : 100000 your password : 02710101 --	225	119

TABLE 3: Comparison between SMS length before and after Compression

Table 4 illustrates the results obtained after applying the proposed technique. The 1st column contains SMS samples, the 2nd column represents the total number of SMS characters before the encryption process, the 3rd column contains SMS length after the compression process, the 4th column contains the percentage of compression phase, and the 5th column contains message length using the proposed technique.

Message	Length of original Message	Length of compressed Message	Percentage of compression phase	Message length using the proposed technique
#There are "men" like mountains "high" friend "honor" comradely "warranty" communicate with them "right and duty of the length of time"	160	125	22%	156

forgotten "impossible".				
Source, Name : ahmed Muhamed Password : 02034112 Card Number : 2400139 Account Number : 0111149 Operation : withdrawal Value : 1000$ Destination, Name : MobiTech Account Number : 0111133	185	142	23%	156
Dear Sir this data are important for you so take your precautions -- name : ahmed Muhammad balance : 100000 your password : 02710101 --	225	119	47%	156
Your account 'Save 1' was credited with $999.98 on Wed 22 Nov 2006 Ref.2390809CR Call 800800 for assistance, if required. Thank you for SMS Banking with ABC Bank.	165	145	12%	156
Salary has been credited to your A/C BAL A/C NO. Balance in A/C xxxxx3329 as of 06 Aug 2009 is INR /908.8/. Thank you for SMS Banking with ABC Bank.	150	135	10%	156
xyzBank, user test Account Number:9820209954 Available balance in A/C xx310 On 04-Nov 2008 05:30 Is Rs. 50000 Thank you for SMS Banking with ABC Bank	158	141	11%	156
Peace be upon you Dear,Muhammed Key 1 : A2HBN - 3SJKL - 7HBN6 - OIKML - YPL9N - OPF8V - TRDCV - 7HJ4D Key 2 : K8DFF - BN4KI - KSLOM - QPOCD - AOPED -\x01\x33IOMN - 8GVFD	166	144	13%	156
Peace be upon you Name : Ahmed Muhamed Account Number : 056789034 Operation type : withdrawal Balance : 60000 Value : 10000 Outstanding Account : 40000	241	163	32%	Split into two messages

on 15-Aug 2009 06:45 --- SMS services center.				

TABLE 4: Comparison between SMS lengths using compression and the proposed technique

It is clear from Table 4 that using the proposed technique for securing SMS messages caused a considerable reduction in their lengths equal 21% approximately on average. Also, the length of compressed message depends on its contents. It should be noted that the last message in this table has been split into two messages because its length is greater than 145 characters.

6. Conclusion and future work

In this paper a new hybrid technique for securing SMS is introduced. The proposed technique combines the compression and encryption processes. The proposed technique compresses the SMS data using a lossless algorithm. After this step the compressed SMS data is encrypted using RSA algorithm. The advantage of this technique is achieving the protection criteria such as confidentiality and authenticity between two communication parties and at the same time decreasing the message lengths. The experimental results show that SMS length does not exceed the standard SMS length using the proposed technique compared with the technique that uses only the RSA encryption process to secure SMS. Future work is required to apply the proposed technique to other mobile operating systems and services.

REFERENCES

1. SMS document, Nokia, (2009, June). Available:http://wiki.forum.nokia.com/index.php/SMS
2. J. Li-Chang Lo, J. Bishop and J. Eloff. *"SMSSec: an end-to-end protocol for secure SMS"*, Computers & Security, 27(5-6):154-167, 2007.
3. P. Traynor, W. Enck, P. McDaniel and T. La Porta. *"Mitigating Attacks on Open Functionality in SMS-Capable Cellular Networks"*, IEEE/ACM Transactions on In Networking, 17(1):40-53, 2009
4. GSM document, Short Message Service, (2009, July). Available: http://www.gsmfavorites.com/documents/sms/
5. SMS peer-to-peer protocol, Wikipedia, (2009, May). Available: http://en.wikipedia.org/wiki/Short_message_peer-to-peer_protocol
6. PDU-encode-decode, thought works, (2009, July). Available: http://twit88.com/home/utility/sms-pdu-encode-decode
7. N. Croft and M. Olivier, *"Using an approximated One Time Pad to Secure Short Messaging Service (SMS)"*, In Proceedings of the Southern African Telecommunication Networks and Applications Conference. South Africa, 2005
8. G. Le Bodic, "*Mobile Messaging Technologies and Services SMS, EMS and MMS*", 2nd ed., John Wiley & Sons Ltd, (2005).
9. SMS vulnerabilities and XMS technology, Network Security Solutions, (2009, July). Available: http://www.mynetsec.com/files/xms_mobile/SMS_Vulnerabilities_XMS_Technology_White_Paper.pdf
10. Atul Kahate, "*Cryptography and network security*", 3rd ed., Tata McGrawHill, (2003).
11. David Pointcheval, RSA Laboratories' CryptoBytes, "*How to Encrypt Properly with RSA*", Volume 5, No.1, Winter/Spring 2002, pp. 9-19.
12. Symbian developer library, Symbian Software Ltd, (2006, January). Available: https://developer.symbian.com/main/documentation/sdl/;jsessionid=D059D9E1944BD96B3FAA3A61E42E7FD7.worker1
13. Anita & Nupur Prakash, *“Performance Analysis of Mobile Security Protocols: Encryption and Authentication”*, International Journal of Security, Volume (1) : Issue (1), June 2007.
14. Bhadri Raju MSVS, Vishnu Vardhan B, Naidu G A, Pratap Reddy L & Vinaya Babu A, *"A Noval Security Model for Indic Scripts - A Case Study on Telugu"*, International Journal of Computer Science and Security, (IJCSS) Volume (3) : Issue (4), August 2009.

15. Jayaprakash Kar & Banshidhar Majhi, *"An Efficient Password Security of Multi-Party Key Exchange Protocol based on ECDLP",* November 2009.

Manuscript Preparation Guidelines for International Journal of Computer Science and Security

Omessaad Hamdi ohamdi@labri.fr
LABRI Laboratory,
Bordeaux 1, France.

Ammar Bouallegue ammar.bouallegue@enit.rnu.tn
SYSCOM Laboratory,
Ecole Nationale d'ingénieurs
De Tunis, Tunisia

Sami Harari harari@univ-tln.fr
USTV,
Toulon France

Abstract

We discuss the chained randomized linear code and their use in cryptography. We show that the adoption of randomized chained codes in the framework of McEliece cryptosystem expose the cryptosystem to some new attacks.

Key Words: Cryptography, Chained Codes, Attack, Complexity

1. INTRODUCTION

In this paper, a new variant of cryptographic schemes based on error coding is studied. Random based techniques allow to design large families of chained codes. Therefore, in principle, such codes can substitute Goppa codes, originally used by McEliece [2].The McEliece cryptosystem is a public key cryptosystem based on coding theory that has successfully resisted cryptanalysis [1] for thirty years. The original version, based on Goppa codes, is able to guarantee a high level of security, and is faster than computing solutions, like RSA.

Despite this, it has not been considered in practical applications, due to the major drawbacks like the large size of the public key, the low transmission rate. Moreover, there is no efficient signature scheme based on error coding.

Several attempts have been made for overcoming such drawbacks, but the adoption of most families of codes has not been possible without compromising the system security [2], [8], [9]. Chained codes are a particular class, able to join low complexity decoding techniques. One idea consists in adopting this family of codes in some signature schemes.

Recently, however, new attacks have been found that are able to exploit the flaw in the transmission from the private key to the public one [10]. Such attack seems to be effectively countered by changing some constituent matrices like introducing some random vectors.

This works gives an overview of the chained code and weakness related to their structure. A recent randomized version can be considered and its ability to counter the currently known attacks is discussed.

To counter this weakness, we concatenate random rows to the generator matrix. This new structure avoids minimum codewords. However, it does not modify the dual code. Consequently, other attacks can be generated.

The details of chained code design are given in section 2.In sections 3 and 4, a digital signature scheme using chained code and its security are discussed. In section 5, we introduce a digital signature using randomized chained code and before concluding we study its security.

2. CHAINED CODE

A chained code C is defined as a direct sum of γ elementary codes $C_i(n_i, k_i)$. This code is of length $N = \sum_{i=1}^{\gamma} n_i$ and of dimension $K = \sum_{i=1}^{\gamma} k_i$.

$$C = \bigoplus_{i=1}^{\gamma} C_i = \{(u_1,...,u_\gamma); u_1 \in C_1,...,u_\gamma \in C_\gamma\}$$

To encode an information $m = (m_1,...,m_\gamma)$, where m_i is k_i bits, we simply multiply it by the generator matrix to obtain the codeword $u = m.G = (u_1,...,u_\gamma)$ with u_i is the n_i bits codeword obtained from m_i using the elementary code C_i. So, G is a diagonal matrix in blocs and whose diagonal is formed by elementary generator matrices G_i of the code C_i.

We assume that we have an efficient decoding algorithm for each elementary code C_i. To decode $u = (u_1,...,u_\gamma)$, we apply for each codeword u_i its correspondent decoding algorithm $dec_{C_i}(\ )$. The decoded word is $m = (m_1,...,m_\gamma)$ with $m_i = dec_{C_i}(u_i)$.

We define the support of a non zero word $x = (x_1,...,x_n)$, denoted $\sup(x)$, as the set of its non zero positions. $\sup(x) = \{i \in \{1,..,n\}, x_i \neq 0\}$ and the support of a set $S = \{y_1,...,y_\gamma\}$ as the union of the supports of its words $\sup(S) = \bigcup_{y_i \in S} \sup(y_i)$. So the support of a code $C(N,K)$ is the union of its 2^k codeword supports.

Two words x and y are said to be connected if their supports are not disjoints i.e $\sup(x) \cap \sup(y) = \Theta$ and two sets I and J are said to be disjoints if there is no connection subset between them.

A non zero codeword x of C is said to be minimal support if there is no codeword $y \in C$ such that $\sup(y) \subset \sup(x)$.

Two codes $C(N,K)$ and $C'(N,K)$ are said to be equivalents if there is a permutation σ of $\{1,..,N\}$ such as: $C' = \sigma(C) = \{c_{\sigma(1)},..,c_{\sigma(N)}\}$. In other words, C and C' are equivalents if there is a permutation matrix such as for any generator matrix G of C, the matrix $G' = G.P$ is a generator matrix of C'.

3. Chained codes and Cryptography

As we mentioned in the introduction, the drawback of the unique digital signature scheme based on error coding is the high signature complexity which is due to Goppa decoding algorithm. One idea to counter this drawback consists in replacing Goppa code by chained code which have faster decoding algorithm.

Generally, the secret key of a cryptographic scheme based on error coding is the code itself, for which an efficient decoding algorithm is known, and the public key is a transformation of the generator or parity check matrices. We consider a digital signature scheme based on chained code, and then we develop an algorithm to discover the private key from public key. This attack is applicable for each cryptographic scheme since it is a structural attack.

Secret key:

- S is a random $(K \times K)$ non singular matrix called the scrambling matrix.
- G is a $(K \times N)$ generator matrix of a chained code
- P is a random $(N \times N)$ permutation matrix

Public key:

- $G' = S.G.P$ is a randomly scrambled et permuted generator matrix. It is a generator matrix of an equivalent non structured code to the chained code $\sum_i c_i$ is the completed correction capacities calculated as [3].
- $h(\ )$ is a hash function.

Signature:

The signer, first, calculates $y = h(M).P^{-1}$, where $h(M)$ is the N bit message, P^{-1} is the inverse of P. Then he uses the completed decoding algorithm [3] for the original chained code C to obtain $x = S.\sigma$. Finally, the receiver obtains the signature by computing $\sigma = S^{-1}.x$ where S^{-1} is the inverse of S.

Verification:

The verifier calculates $\rho' = \sigma.G'$ and $\rho = h(M)$

The signature is valid if $d(\rho, \rho') < \sum_i c_i$

To avoid exhaustive attack, we use at least five different elementary codes and to avoid attack by information set, we use a chained code with length at least equal to 1500 bits.

After developing a digital signature scheme, we discovered a weakness in this scheme. This weakness is due to the fact that chained codes have an invariant. Code equivalence means that one generator matrix is a permutation of the other, because matrix S does not change the code but only performs a modification on the basis of the linear subspace. Canteaut showed that the matrix S may be important to hide the systematic structure of the Goppa codes, therefore having an important security role [6]. However, Heiman was the first to study this point and states that the random matrix S used in the original McEliece scheme serves no security purpose concerning the protection [7]. We confirm this argument and we show that the random matrix S has no security role for cryptographic schemes based on linear codes. We state also that disjoint elementary code supports is an invariant by permutation.

The attack explores the characteristics of the code transformation in order to identify its building blocks. Its input is a generating matrix G' of a randomly permuted chained code of length N and dimension K . Its output is a structured chained code. The algorithm's steps are:

- Apply a Gauss elimination to the rows of the matrix G' to obtain the systematic form $G_0 = (I_d, Z)$.

Sendrier shows that rows of any systematic generator matrix of a code C are minimal support codewords of C and that any minimal support codeword of C is a row of a systematic generator matrix of C [4]. So, the systematic chained code support is formed by disjoint sets. Each set represents the support of an elementary code. The transformation of any randomly permuted chained code generator matrix into a systematic matrix by linear algebraic algorithms will allow us to find these supports and thus elementary codes.

- Search the disjoint sets of rows of the systematic matrix G_0. Each set forms the elementary code support. Use elementary decoding algorithms to decode every message. As application of these codes, regular LDPC codes which represent chained repetition codes. Next sections represent the proprieties of these codes.

The complexity of this attack is less than 2^{45} even with so long codes (see FIGURE 1).

4. Randomized chained linear codes

To counter the attack introduced in previous section, one idea consists in concatenating random vectors to the generator matrix. In this section, first, we define randomized chained codes then we introduce a cryptographic scheme based on these codes.

4.1 Random vectors

The randomized chained linear code concatenates random vectors of length N to the chained code. Using Information Theory, a N bit random binary vector is of weight closely to $N/2$ and the distance between two random vectors is of order $N/4$. These approximations are more precise when N is large.

4.2 Construction of randomized chained codes

Lets consider a chained linear code generator matrix G_{CL} as described in section 2. Each elementary linear code is of length n_i and of size k_i. Chained linear code is of length $N = \sum_{i=1}^{\gamma} n_i$ and of dimension $K = \sum_{i=1}^{\gamma} k_i$.

Lets consider a matrix G_r formed by K random rows of length N .

The generator matrix G of the system using randomized linear chained code has the following form: $G = (G_{CL}, G_r)$.

The weight of a row of the systematic generator matrix is about $N/2 + p_i$ where p_i is the weight of i^th^ row of the chained code generator matrix G_{CL} .

4.2.1 Encoding

m is a word of length K to be encoded. The codeword is obtained by multiplying m by the generator matrix G of the randomized chained linear code.

$$c = m.G$$

4.2.2 Decoding

r is the word to be decoded.

$$r = c + e = m.G_{CL} + e_1, m.G_r + e_2$$

Note by $dec_{CL}(\ )$ the chained linear decoding algorithm. Thus, $m = dec_{CL}(m.G_{CL} + e_1)$. The codeword closest to r is $c = m.G$.

5. DIGITAL SIGNATURE USING RANDOMIZED CHAINED LINEAR CODES

5.1 Key generation

- Generate a sequence γ linear codes. Each code is of length n_i and of dimension k_i.
- Build the chained linear code generator matrix G_{CL}. This matrix is of size $N = \sum_{i=1}^{\gamma} n_i \times K = \sum_{i=1}^{\gamma} k_i$
- Generate K random vectors v_i of length N. These vectors will be stored in a matrix G_r of size $K \times N$.

  The obtained code is of length $2N$ and size K. It has the following generator matrix's form $G = (G_{CL}, G_r)$

To hide the code structure, we also generate

- A random invertible matrix S of size $((2.N) - N) \times ((2.N) - K)$.
- A permutation matrix P of size $((2.N) \times (2.N))$
- Determine the check parity matrix H as follows $H.(G.P)^t = 0$

Thus, the private key is formed by

- The generator matrix G of size $K \times 2.N$
- The random matrix S of size $((2.N) - N) \times ((2.N) - K)$.
- The permutation matrix P of size $((2.N) \times (2.N))$.

The public key is formed by the hidden and permuted parity check Matrix $H^{'} = S.H$ of size $(2.N - K) \times (2.N)$

5.2 Signature algorithm

Let m be a message to be signed. The signer has the private key formed by G, S and P and the hash function $h(\ )$ whose result is of length $2.N$.

- Compute $\rho^{'} = h(m)$ of length $2.N$
- Compute $\rho = \rho^{'}.P^{-1}$.
- Divide ρ in two parts ρ_1 and ρ_2, each one is of length N.

$$\rho = \rho_1 \| \rho_2$$

- Decode ρ_1 using the decoding algorithm of chained linear code to obtain information m of length K .
- Compute $v = m.G$ which is a codeword.
- Compute $e^{'} = \rho + v$ the error related to the secret code which is closer to $N/2$ .This error has the same syndrome as ρ .
- Compute the error $e = e^{'}.P$ and its weight $p = w(e)$. The error e has the same syndrome as $\rho^{'} = h(m)$relatively to the public code generated by $G.P$

The signature of m is formed by $\sigma = (e, p)$.

5.3 Verification Algorithm

- The verifier has the matrix H and the hash function $h(\ )$, the message m and the signature σ .
- he checks that $w(e) = p$
- he computes $\rho^{'} = h(m)$.
- he computes $x_1 = H^{'}.e$
- he computes $x_2 = H^{'}.\rho^{'}$

The signature is valid if

$$x_1 = x_2$$

5.4 Soundness

$x_1 = H^{'}.e = H^{'}.(\rho + v).P = H^{'}.\rho.P = x_2$ since $v.P$ is a codeword of the permuted code having $G.P$ as generator matrix.

5.5 Parameters

Forging a signature consists in determining the signature $\sigma = (e, p)$ message from m or retrieving the secret key. An attacker who has the parity check matrix of size $(2.NK) \times 2.N$, may proceed as follows:

- he transforms $H^{'}$ a systematic matrix $H_0 = (R^t, I_{(2.N-K),(2.N-K)})$
- he guess the corresponding matrix G_0 of size $K \times 2.N$:

$$G_0 = (I_K, R)$$

- he computes $\rho = h(M) = (\rho_1, \rho_2)$ with $|\rho_1| = K$ and $|\rho_2| = 2N - K$
- he search the closest codeword $c = (c_1, c_2)$ of length $2.N$ to ρ .

So, he will obtain

- $d(C_1, \rho_1) = 0$
- $d(C_2, \rho_2) = (2.N - K)/2$

To build a secure algorithm, the difference k between p

and $(2.NK)/2$ should be large enough. The table 1 shows parameters for a signature scheme based on randomized chained code. From Table 1, we show that is necessary that used code must have a length $2.N$ greater than 1350.

N	990	1080	1170	1260	1350	1440	1530	1520	1710	1800	1890	1980
K	253	276	299	322	345	368	391	414	437	460	483	506
K	44	48	52	56	60	64	68	72	76	80	84	88

Table 1: Signature parameters

Table 2 shows performances of randomized chained code in terms of execution complexity and public key size.

Signature	Signature with randomized code
Public key size (ko)	123
Signature complexity	2^{20}
Verification Complexity	2^{13}

Table 2: Performance of signature based on randomized chained codes

5.6 Solidity

The strength of the scheme depends on the choice of parameters. There are two types of attacks on asymmetric systems.

The starting point was to hide the structure of the chained codes. Possible attack of the new structure consists in enumerating all matrices of size $(2.N-K)\times 2.N$ and test their equivalences with $H^{'}$. The code is formed by γ elementary codes and K random vectors. So, the number of randomized chained code is $\frac{\left(N!/(N/2!)^{2}\right)!}{K!}2^{v}$ which is very large considering chosen parameters in section 5. The concatenation of random vectors avoid minimal codewords attack since a codeword is at least of weight $N/2$. Moreover, the new structure avoids support disjunction since the distance between two codewords is in order of N/4.

However, this new structure hides a weakness related to the dual code. In fact, concatenated vectors do not modify the dual code. Consequently, an attacker may proceed as follows:

- Transform $H^{'}$ in a systematic matrix $H_0=\left(R^{t},I_{2.N-K}\right)$.

- Search minimal codewords of elementary linear codes which have weight smaller than those of random vectors.

- Use the algorithm introduced in section 3 to recover dual code.

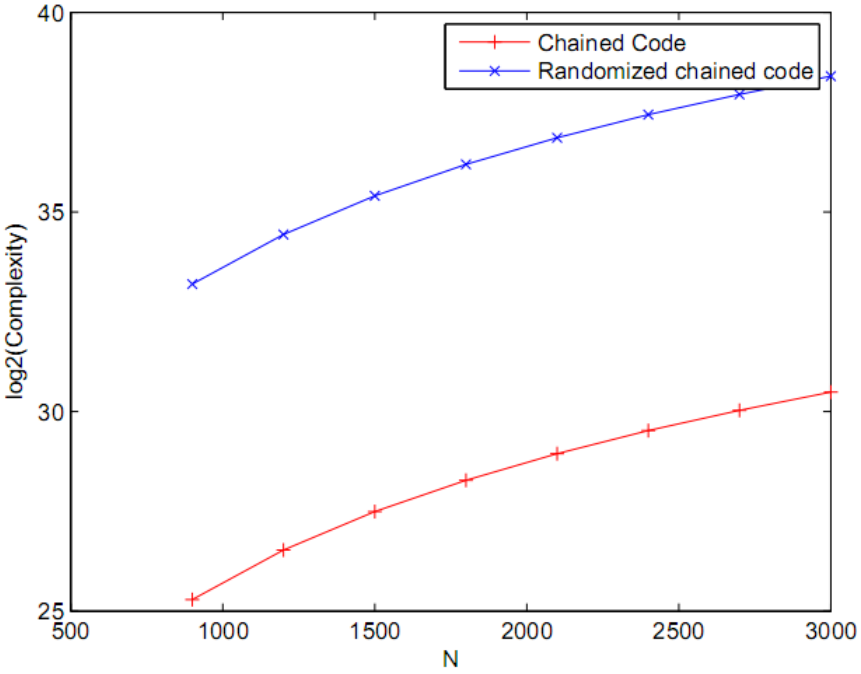


FIGURE 1: Attack Complexity

The security of cryptographic schemes based on error coding is highly dependent on the class of used codes. Some class of codes reveal their characteristics even when they go through the permutation used to construct the public code. It is the case with chained codes and randomized chained codes. The starting point was the observation that any systematic matrix of a chained code is formed by small weight codeword and that the code contains so many minimal support codewords. These two properties lead to a structural attack of digital signature scheme based on chained code.

We have tried to counter this attack by concatenating some random vectors to the generator matrix. However, the added vectors avoid this attack but they do not modify the dual code. Consequently, we discover another structural weakness related to this kind of codes.

Figure 1 shows the complexity of the attacks of some cryptosystems using chained codes and randomized chained code. The complexity is always less than 2^{45} even with so long codes $(N = 3000)$. This complexity prohibits using chained code in cryptography.

6. Conclusion

In this paper, we discussed the structure of a randomly permuted chained code. We explored potential threats from systematic generator matrices that have particular structure. Chained code generator matrices have the properties of disconnected elementary code supports. We have tried to hide this property by concatenating some random vectors to the generator matrix. Unfortunately, these vectors avoid attack by minimum codeword in the code itself. However, they do not modify the dual code which makes weakness on cryptographic scheme based on chained codes. This property is invariant by permutation, which make this kind of code useless in cryptography.

7. REFERENCES

1. *E.R. Berlekamp, R.J. McEliece, and H.C.A. van Tilborg, "On the inherent intractability of certain coding problems", IEEE Transactions on Information Theory, Vol.24, No.3,1978, pp.384-386.*

2. *R.J. McEliece, "A public-key cryptosystem based on algebraic coding theory"; DSN Prog. Rep., Jet Propulsion Laboratory, California Inst. Technol., Pasadena, CA, pp. 114-116,January 1978.*

3. *D. J. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. In Post-Quantum Cryptography, volume 5299 of Lecture Notes in Computer Science, pages 31-46. Springer Berlin Heidelberg, 2008.*

4. *N. Courtois, M. Finiasz, and N. Sendrier, "How to achieve a McEliece-based digital signature scheme", In C. Boyd, editor, Asiacrypt 2001, volume 2248 of LNCS, pages 157-174. Springer-Verlag, 2001.*

5. *N.Sendrier, "On the structure of a linear code"AAECC, Vol.9, n3, 1998, pp.221-242.*

6. *A. Canteaut "Attaques de cryptosystemes a mots de poids faible et construction de fonctions t-resilientes" PhD thesis, Universite Paris 6, October 1996.*

7. *R. Heiman "On the security of Cryptosystems Based on Linear Error Correcting codes" MSc. Thesis, Feinberg Graduate School of the Weizmann Institute of Science. August 1987.*

8. *M. Baldi and F. Chiaraluce. Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes. In Proc. IEEE International Symposium on Information Theory (ISIT 2007), pages 2591-2595, Nice, France, June 2007.*

9. *A. Otmani, J. P. Tillich, and L. Dallot. Cryptanalysis of two McEliece cryptosystems based on quasi- cyclic codes. In Proc. First International Conference on Symbolic Computation and Cryptography (SCC 2008), Beijing, China, April 2008.*

10. *O. Hamdi, A. Bouallegue, S.Harari, Weakness on Cryptographic Schemes based on Chained Codes, The First International Workshop on Wireless and Mobile Networks Security (WMNS-2009) in conjunction with NSS 2009, October 19~21 2009, Gold Coast, Australia.*

A Parallel Framework for Multilayer Perceptron for Human Face Recognition

Debotosh Bhattacharjee debotosh@indiatimes.com
*Reader,
Department of Computer Science and Engineering,
Jadavpur University,
Kolkata- 700032, India.*

Mrinal Kanti Bhowmik mkb_cse@yahoo.co.in
*Lecturer,
Department of Computer Science and Engineering,
Tripura University (A Central University),
Suryamaninagar- 799130, Tripura, India.*

Mita Nasipuri mitanasipuri@gmail.com
*Professor,
Department of Computer Science and Engineering,
Jadavpur University,
Kolkata- 700032, India.*

Dipak Kumar Basu dipakkbasu@gmail.com
*Professor, AICTE Emeritus Fellow,
Department of Computer Science and Engineering,
Jadavpur University,
Kolkata- 700032, India.*

Mahantapas Kundu mkundu@cse.jdvu.ac.in
*Professor,
Department of Computer Science and Engineering,
Jadavpur University,
Kolkata- 700032, India.*

Abstract

Artificial neural networks have already shown their success in face recognition and similar complex pattern recognition tasks. However, a major disadvantage of the technique is that it is extremely slow during training for larger classes and hence not suitable for real-time complex problems such as pattern recognition. This is an attempt to develop a parallel framework for the training algorithm of a perceptron. In this paper, two general architectures for a Multilayer Perceptron (MLP) have been demonstrated. The first architecture is All-Class-in-One-Network (ACON) where all the classes are placed in a single network and the second one is One-Class-in-One-Network (OCON) where an individual single network is responsible for each and every class. Capabilities of these two architectures were compared and verified in solving human face recognition, which is a complex pattern recognition task where several factors affect the recognition performance like pose variations, facial expression changes, occlusions, and most importantly illumination changes. Both the structures were implemented and tested for face recognition purpose and experimental results show that the OCON structure performs better than the generally used ACON ones in term of training convergence speed of the network. Unlike the conventional sequential approach of training the neural networks, the OCON technique may be implemented by training all the classes of the face images simultaneously.

Keywords: Artificial Neural Network, Network architecture, All-Class-in-One-Network (ACON), One-Class-in-One-Network (OCON), PCA, Multilayer Perceptron, Face recognition.

1. INTRODUCTION

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze [1]. This proposed work describes the way by which an Artificial Neural Network (ANN) can be designed and implemented over a parallel or distributed environment to reduce its training time. Generally, an ANN goes through three different steps: training of the network, testing of it and final use of it. The final structure of an ANN is generally found out experimentally. This requires huge amount of computation. Moreover, the training time of an ANN is very large, when the classes are linearly non-separable and overlapping in nature. Therefore, to save computation time and in order to achieve good response time the obvious choice is either a high-end machine or a system which is collection of machines with low computational power.

In this work, we consider multilayer perceptron (MLP) for human face recognition, which has many real time applications starting from automatic daily attendance checking, allowing the authorized people to enter into highly secured area, in detecting and preventing criminals and so on. For all these cases, response time is very critical. Face recognition has the benefit of being passive, nonintrusive system for verifying personal identity. The techniques used in the best face recognition systems may depend on the application of the system.

Human face recognition is a very complex pattern recognition problem, altogether. There is no stability in the input pattern due to different expressions, adornments in the input images. Sometimes, distinguishing features appear similar and produce a very complex situation to take decision. Also, there are several other that make the face recognition task complicated. Some of them are given below.

a) Background of the face image can be a complex pattern or almost same as the color of the face.
b) Different illumination level, at different parts of the image.
c) Direction of illumination may vary.
d) Tilting of face.
e) Rotation of face with different angle.
f) Presence/absence of beard and/or moustache
g) Presence/Absence of spectacle/glasses.
h) Change in expressions such as disgust, sadness, happiness, fear, anger, surprise etc.
i) Deliberate change in color of the skin and/or hair to disguise the designed system.

From above discussion it can now be claimed that the face recognition problem along with face detection, is very complex in nature. To solve it, we require some complex neural network, which takes large amount of time to finalize its structure and also to settle its parameters.
In this work, a different architecture has been used to train a multilayer perceptron in faster way. Instead of placing all the classes in a single network, individual networks are used for each of the

classes. Due to lesser number of samples and conflicts in the belongingness of patterns to their respective classes, a later model appears to be faster in comparison to former.

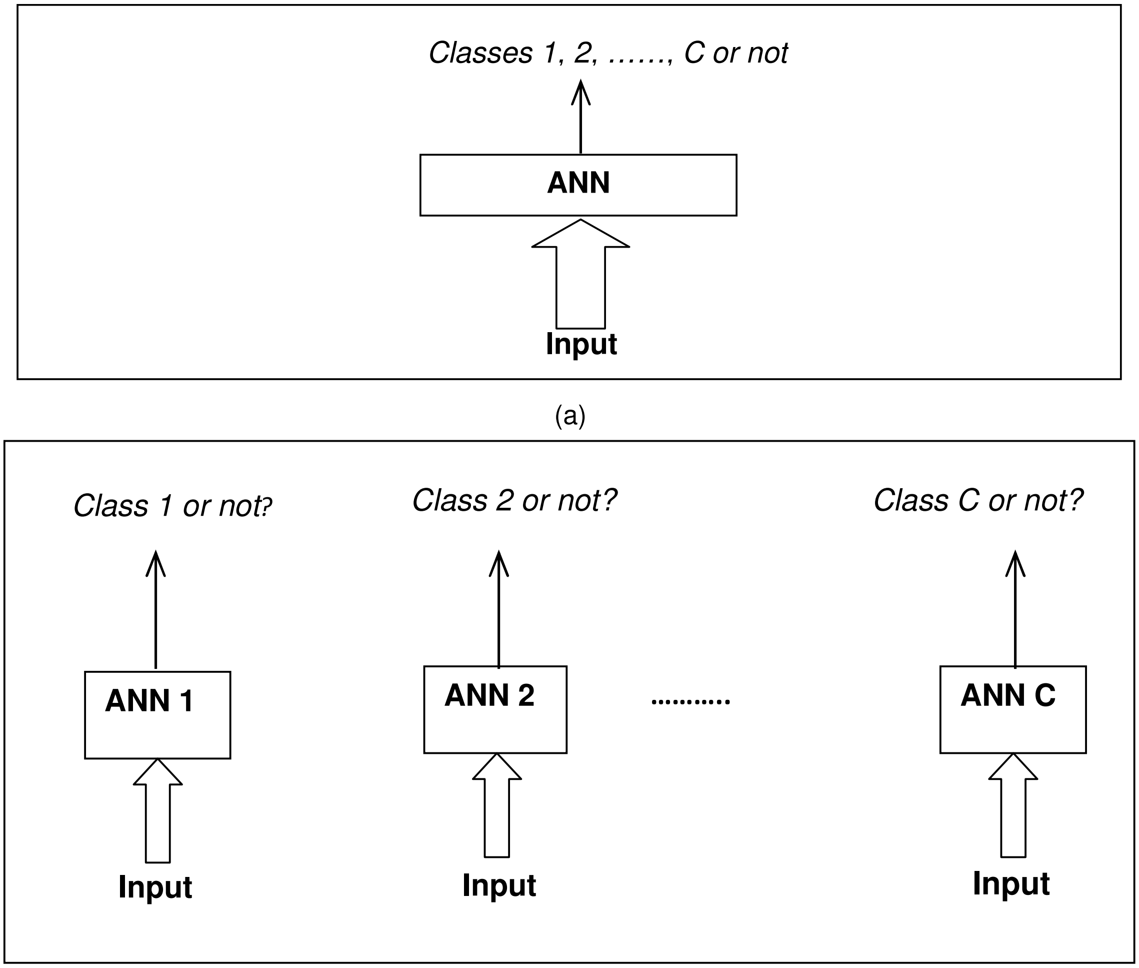
2. ARTIFICIAL NEURAL NETWORK

Artificial neural networks (ANN) have been developed as generalizations of mathematical models of biological nervous systems. A first wave of interest in neural networks (also known as connectionist models or parallel distributed processing) emerged after the introduction of simplified neurons by McCulloch and Pitts (1943).The basic processing elements of neural networks are called artificial neurons, or simply neurons or nodes. In a simplified mathematical model of the neuron, the effects of the synapses are represented by connection weights that modulate the effect of the associated input signals, and the nonlinear characteristic exhibited by neurons is represented by a transfer function. The neuron impulse is then computed as the weighted sum of the input signals, transformed by the transfer function. The learning capability of an artificial neuron is achieved by adjusting the weights in accordance to the chosen learning algorithm. A neural network has to be configured such that the application of a set of inputs produces the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to train the neural network by feeding it teaching patterns and letting it change its weights according to some learning rule. The learning situations in neural networks may be classified into three distinct sorts. These are supervised learning, unsupervised learning, and reinforcement learning. In supervised learning, an input vector is presented at the inputs together with a set of desired responses, one for each node, at the output layer. A forward pass is done, and the errors or discrepancies between the desired and actual response for each node in the output layer are found. These are then used to determine weight changes in the net according to the prevailing learning rule. The term supervised originates from the fact that the desired signals on individual output nodes are provided by an external teacher [3]. Feed-forward networks had already been used successfully for human face recognition. Feed-forward means that there is no feedback to the input. Similar to the way that human beings learn from mistakes, neural networks also could learn from their mistakes by giving feedback to the input patterns. This kind of feedback would be used to reconstruct the input patterns and make them free from error; thus increasing the performance of the neural networks. Of course, it is very complex to construct such types of neural networks. These kinds of networks are called as auto associative neural networks. As the name implies, they use back-propagation algorithms. One of the main problems associated with back-propagation algorithms is local minima. In addition, neural networks have issues associated with learning speed, architecture selection, feature representation, modularity and scaling. Though there are problems and difficulties, the potential advantages of neural networks are vast. Pattern recognition can be done both in normal computers and neural networks. Computers use conventional arithmetic algorithms to detect whether the given pattern matches an existing one. It is a straightforward method. It will say either yes or no. It does not tolerate noisy patterns. On the other hand, neural networks can tolerate noise and, if trained properly, will respond correctly for unknown patterns. Neural networks may not perform miracles, but if constructed with the proper architecture and trained correctly with good data, they will give amazing results, not only in pattern recognition but also in other scientific and commercial applications [4].

2A. Network Architecture

The computing world has a lot to gain from neural networks. Their ability to learn by example makes them very flexible and powerful. Once a network is trained properly there is no need to devise an algorithm in order to perform a specific task; i.e. no need to understand the internal mechanisms of that task. The architecture of any neural networks generally used is All-Class-in-One-Network (ACON), where all the classes are lumped into one super-network. Hence, the implementation of such ACON structure in parallel environment is not possible. Also, the ACON structure has some disadvantages like the super-network has the burden to simultaneously satisfy all the error constraints by which the number of nodes in the hidden layers tends to be large. The structure of the network is All-Classes-in-One-Network (ACON), shown in Figure 1(a) where one single network is designed to classify all the classes but in One-Class-in-One-Network

(OCON), shown in Figure 1(b) a single network is dedicated to recognize one particular class. For each class, a network is created with all the training samples of that class as positive examples, called the class-one, and the negative examples for that class i.e. exemplars from other classes, constitute the class-two. Thus, this classification problem is a two-class partitioning problem. So far, as implementation is concerned, the structure of the network remains the same for all classes and only the weights vary. As the network remains same, weights are kept in separate files and the identification of input image is made on the basis of feature vector and stored weights applied to the network one by one, for all the classes.



(a)

(b)

Figure 1: a) All-Classes-in-One-Network (ACON) **b)** One-Class-in-One-Network (OCON).

Empirical results confirm that the convergence rate of ACON degrades drastically with respect to the network size because the training of hidden units is influenced by (potentially conflicting) signals from different teachers. If the topology is changed to One Class in One Network (OCON) structure, where one sub-network is designated and responsible for one class only then each sub-network specializes in distinguishing its own class from the others. So, the number of hidden units is usually small.

2B. Training of an ANN

In the training phase the main goal is to utilize the resources as much as possible and speed-up the computation process. Hence, the computation involved in training is distributed over the system to reduce response time. The training procedure can be given as:

(1) Retrieve the topology of the neural network given by the user,
(2) Initialize required parameters and weight vector necessary to train the network,
(3) Train the network as per network topology and available parameters for all exemplars of different classes,
(4) Run the network with test vectors to test the classification ability,
(5) If the result found from step 4 is not satisfactory, loop back to step 2 to change the parameters like learning parameter, momentum, number of iteration or even the weight vector,
(6) If the testing results do not improve by step 5, then go back to step 1,
(7) The best possible (optimal) topology and associated parameters found in step 5 and step 6 are stored.

Although we have parallel systems already in use but some problems cannot exploit advantages of these systems because of their inherent sequential execution characteristics. Therefore, it is necessary to find an equivalent algorithm, which is executable in parallel.

In case of OCON, different individual small networks with least amount of load, which are responsible for different classes (e.g. k classes), can easily be trained in k different processors and the training time must reduce drastically. To fit into this parallel framework previous training procedure can be modified as follows:

(1) Retrieve the topology of the neural network given by the user,
(2) Initialize required parameters and weight vector necessary to train the network,
(3) Distribute all the classes (say k) to available processors (possibly k) by some optimal process allocation algorithm,
(4) Ensure the retrieval the exemplar vectors of respective classes by the corresponding processors,
(5) Train the networks as per network topology and available parameters for all exemplars of different classes,
(6) Run the networks with test vectors to test the classification ability,
(7) If the result found from step 6 is not satisfactory, loop back to step 2 to change the parameters like learning parameter, momentum, number of iteration or even the weight vector,
(8) If the testing results do not improve by step 5, then go back to step 1,
(9) The best possible (optimal) topology and associated parameters found in step 7 and step 8 are stored,
(10) Store weights per class with identification in more than one computer [2].

During the training of two different topologies (OCON and ACON), we used total 200 images of 10 different classes and the images are with different poses and also with different illuminations. Sample images used during training are shown Figure 2. We implemented both the topologies using MATLAB. At the time of training of our systems for both the topologies, we set maximum number of possible epochs (or iterations) to 700000. The training stops if the number of iterations exceeds this limit or performance goal is met. Here, performance goal was considered as 10^{-6}. We have taken total 10 different training runs for 10 different classes for OCON and one single training run for ACON for 10 different classes. In case of the OCON networks, performance goal was met for all the 10 different training cases, and also in lesser amount of time than ACON. After the completion of training phase of our two different topologies we tested our both the network using the images of testing class which are not used in training.

2C. Testing Phase

During testing, the class found in the database with minimum distance does not necessarily stop the testing procedure. Testing is complete after all the registered classes are tested. During testing some points were taken into account, those are:

(1) The weights of different classes already available are again distributed in the available computer to test a particular image given as input,
(2) The allocation of the tasks to different processors is done based on the testing time and inter-processor communication overhead. The communication overhead should be much less than the testing time for the success of the distribution of testing, and
(3) The weight vector of a class matter, not the computer, which has computed it.

The testing of a class can be done in any computer as the topology and the weight vector of that class is known. Thus, the system can be fault tolerant [2]. At the time of testing, we used total 200 images. Among 200 images 100 images are taken from the same classes those are used during the training and 100 images from other classes are not used during the training time. In the both topology (ACON and OCON), we have chosen 20 images for testing, in which 10 images from same class those are used during the training as positive exemplars and other 10 images are chosen from other classes of the database as negative exemplars.

2D. Performance measurement
Performance of this system can be measured using following parameters:
(1) resource sharing: Different terminals remain idle most of the time can be used as a part of this system. Once the weights are finalized anyone in the net, even though not satisfying the optimal testing time criterion, can use it. This can be done through Internet attempting to end the “tyranny of geography”,
(2) high reliability: Here we will be concerned with the reliability of the proposed system, not the inherent fault tolerant property of the neural network. Reliability comes from the distribution of computed weights over the system. If any of the computer(or processor) connected to the network goes down then the system works Some applications like security monitoring system, crime prevention system require that the system should work, whatever may be the performance,
(3) cost effectiveness: If we use several small personal computers instead of high-end computing machines, we achieve better price/performance ratio,
(4) incremental growth: If the number of classes increases, then the complete computation including the additional complexity can be completed without disturbing the existing system. Based on the analysis of performance of our two different topologies, if we see the recognition rates of OCON and ACON in Table 1 and Table 2 OCON is showing better recognition rate than ACON. Comparison in terms of training time can easily be observed in figures 3 (Figure 3 (a) to (k)). In case of OCON, performance goals met for 10 different classes are 9.99999e-007, 1e-006, 9.99999e-007, 9.99998e-007, 1e-006, 9.99998e-007,1e-006, 9.99997e-007, 9.99999e-007 respectively, whereas for ACON it is 0.0100274. Therefore, it is pretty clear that OCON requires less computational time to finalize a network to use.

3. PRINCIPAL COMPONENT ANALYSIS

The Principal Component Analysis (PCA) [5] [6] [7] uses the entire image to generate a set of features in the both network topology OCON and ACON and does not require the location of individual feature points within the image. We have implemented the PCA transform as a reduced feature extractor in our face recognition system. Here, each of the visual face images is projected into the eigenspace created by the eigenvectors of the covariance matrix of all the training images for both the ACON and OCON networks. Here, we have taken the number of eigenvectors in the eigenspace as 40 because eigenvalues for other eigenvectors are negligible in comparison to the largest eigenvalues.

4. EXPERIMENTS RESULTS USING OCON AND ACON

This work has been simulated using MATLAB 7 in a machine of the configuration 2.13GHz Intel Xeon Quad Core Processor and 16 GB of Physical Memory. We have analyzed the performance of our method using YALE B database which is a collection of visual face images with various poses and illumination.

4A. YALE Face Database B

This work has been simulated using MATLAB 7 in a machine of the configuration 2.13GHz Intel Xeon Quad Core Processor and 16 GB of Physical Memory. We have analyzed the performance of our method using YALE B database which is a collection of visual face images with various poses and illumination. This database contains 5760 single light source images of 10 subjects each seen under 576 viewing conditions (9 poses x 64 illumination conditions). For every subject in a particular pose, an image with ambient (background) illumination was also captured. Hence, the total number of images is 5850. The total size of the compressed database is about 1GB. The 65 (64 illuminations + 1 ambient) images of a subject in a particular pose have been "tarred" and "gzipped" into a single file. There were 47 (out of 5760) images whose corresponding strobe did not go off. These images basically look like the ambient image of the subject in a particular pose. The images in the database were captured using a purpose-built illumination rig. This rig is fitted with 64 computer controlled strobes. The 64 images of a subject in a particular pose were acquired at camera frame rate (30 frames/second) in about 2 seconds, so there is only small change in head pose and facial expression for those 64 (+1 ambient) images. The image with ambient illumination was captured without a strobe going off. For each subject, images were captured under nine different poses whose relative positions are shown below. Note the pose 0 is the frontal pose. Poses 1, 2, 3, 4, and 5 were about 12 degrees from the camera optical axis (i.e., from Pose 0), while poses 6, 7, and 8 were about 24 degrees. In the Figure 2 sample images of per subject per pose with frontal illumination. Note that the position of a face in an image varies from pose to pose but is fairly constant within the images of a face seen in one of the 9 poses, since the 64 (+1 ambient) images were captured in about 2 seconds. The acquired images are 8-bit (gray scale) captured with a Sony XC-75 camera (with a linear response function) and stored in PGM raw format. The size of each image is 640(w) x 480 (h) [9].

In our experiment, we have chosen total 400 images for our experiment purpose. Among them 200 images are taken for training and other 200 images are taken for testing purpose from 10 different classes. In the experiment we use total two different networks: OCON and ACON. All the recognition results of OCON networks are shown in Table 1, and all the recognition results of ACON network are shown in Table 2. During training, total 10 training runs have been executed for 10 different classes. We have completed total 10 different testing for OCON network using 20 images for each experiment. Out of those 20 images, 10 images are taken form the same classes those were used during training, which acts as positive exemplars and rest 10 images are taken from other classes that acts as negative exemplars for that class. In case of OCON, system achieved 100% recognition rate for all the classes. In case of the ACON network, only one network is used for 10 different classes. During the training we achieved 100% as the highest recognition rate, but like OCON network not for all the classes. For ACON network, on an average, 88% recognition rate was achieved.

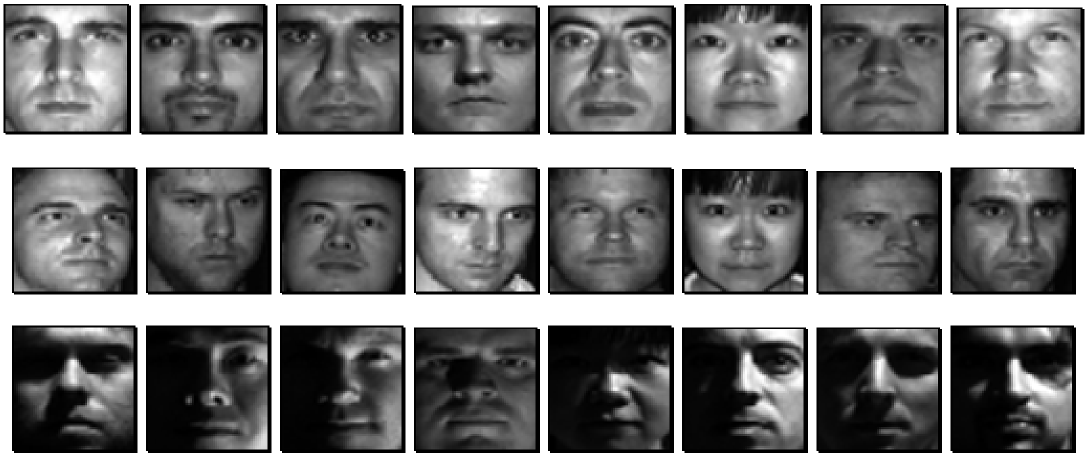


Figure 2: Sample images of YALE B database with different Pose and different illumination.

Class	Total number of testing images	Number of images from the training class	Number of images from other classes	Recognition rate
Class-1	20	10	10	100%
Class-2	20	10	10	100%
Class-3	20	10	10	100%
Class-4	20	10	10	100%
Class-5	20	10	10	100%
Class-6	20	10	10	100%
Class-7	20	10	10	100%
Class-8	20	10	10	100%
Class-9	20	10	10	100%
Class-10	20	10	10	100%

Table 1: Experiments Results for OCON.

Class	Total number of testing images	Number of images from the training class	Number of images from other classes	Recognition rate
Class - 1	20	10	10	100%
Class - 2	20	10	10	100%
Class - 3	20	10	10	90%
Class - 4	20	10	10	80%
Class - 5	20	10	10	80%
Class - 6	20	10	10	80%
Class - 7	20	10	10	90%
Class - 8	20	10	10	100%
Class - 9	20	10	10	90%
Class-10	20	10	10	70%

Table 2: Experiments results for ACON.

In the Figure 3, we have shown all the performance measure and reached goal during 10 different training runs in case of OCON network and also one training phase of ACON network.

We set highest epochs 700000, but during the training, in case of all the OCON networks, performance goal was met before reaching maximum number of epochs. All the learning rates with required epochs of OCON and ACON networks are shown at column two of Table 3.

In case of the OCON network, if we combine all the recognition rates we have the average recognition rate is 100%. But in case of ACON network, 88% is the average recognition rate i.e.

we can say that OCON showing better performance, accuracy and speed than ACON. Figure 4 presents a comparative study on ACON and OCON results.

Total no. of iterations	**Learning Rate (lr)**	**Class**	**Figures**	**Network Used**
290556	lr > 10^{-4}	Class – 1	Figure 3(a)	OCON
248182	lr =10^{-4}	Class – 2	Figure 3(b)	
260384	lr =10^{-5}	Class – 3	Figure 3(c)	
293279	lr < 10^{-4}	Class - 4	Figure 3(d)	
275065	lr =10^{-4}	Class - 5	Figure 3(e)	
251642	lr =10^{-3}	Class – 6	Figure 3(f)	
273819	lr =10^{-4}	Class – 7	Figure 3(g)	
263251	lr < 10^{-3}	Class – 8	Figure 3(h)	
295986	lr < 10^{-3}	Class – 9	Figure 3(i)	
257019	lr > 10^{-6}	Class - 10	Figure 3(j)	
Highest epoch reached (7, 00, 000)	Performance goal not met	For all Classes (class -1,…,10)	Figure 3(k)	ACON

Table 3: Learning Rate vs. Required Epochs for OCON and ACON.

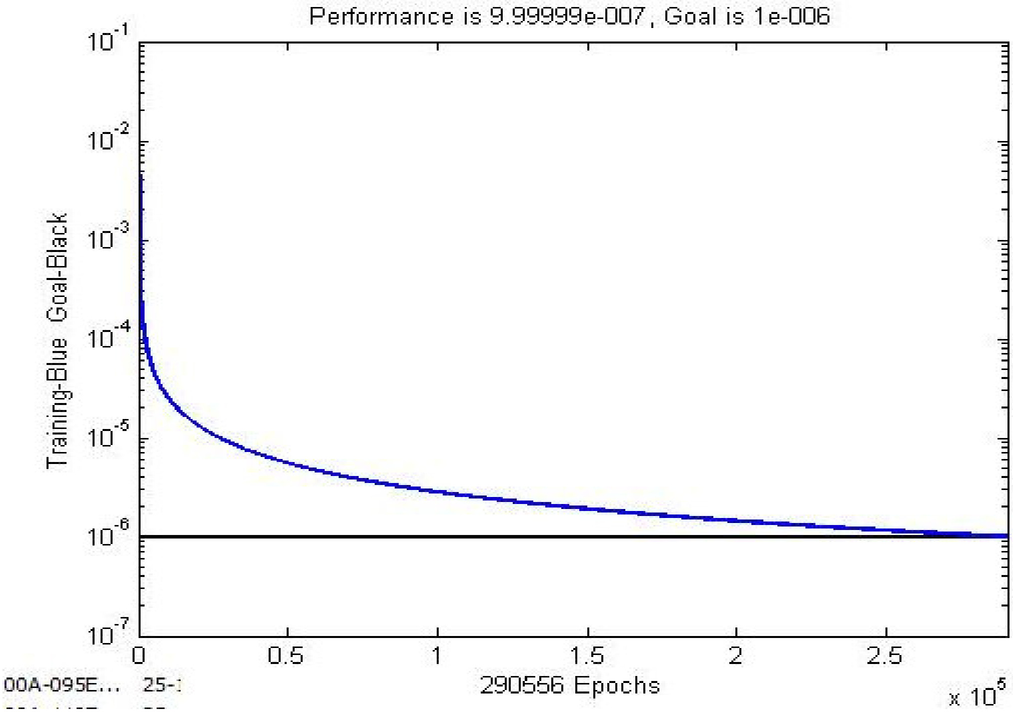


Figure 3 (a) Class – 1 of OCON Network.

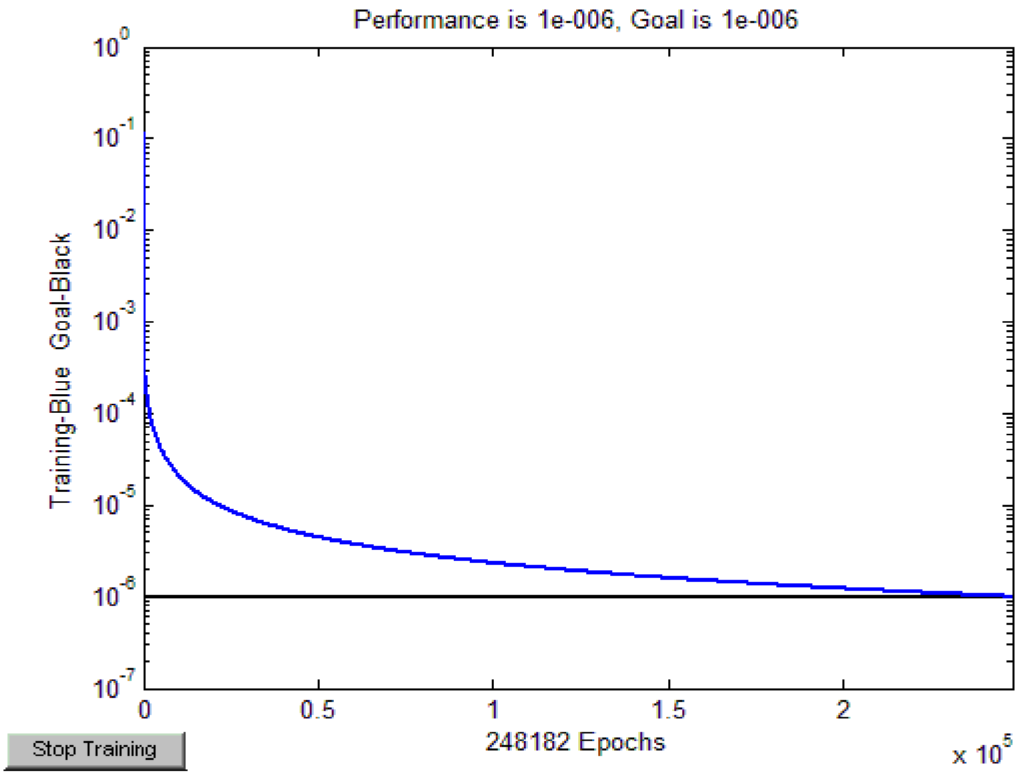


Figure 3 (b) Class – 2 of OCON Network.

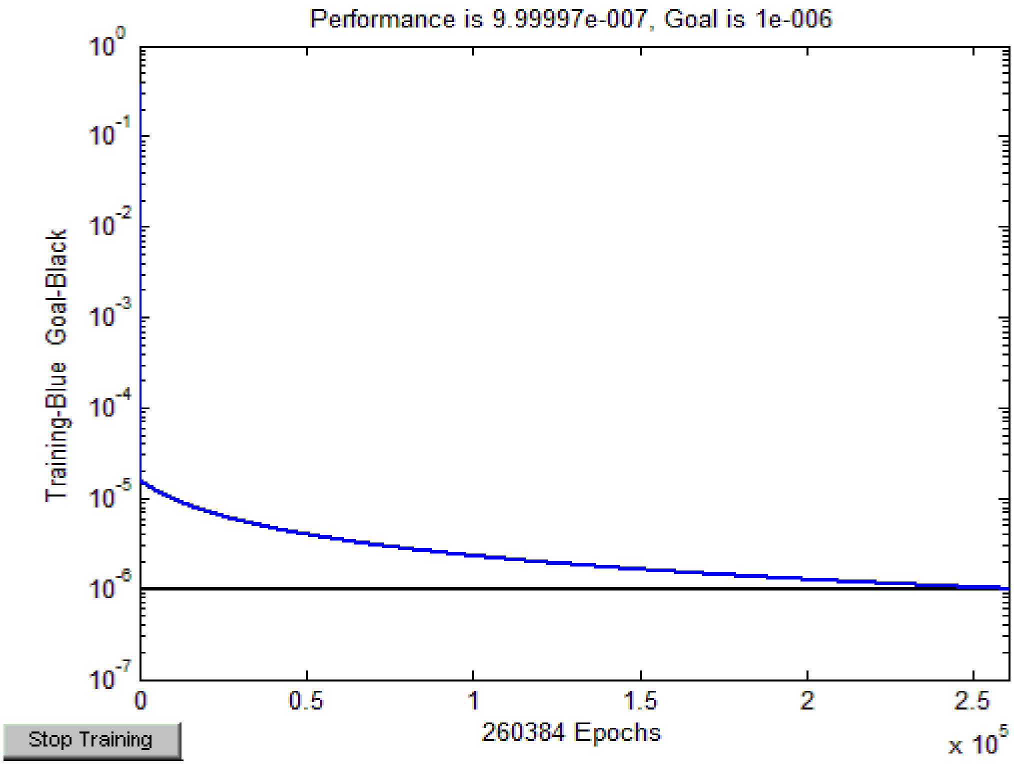


Figure 3 (c) Class – 3 of OCON Network.

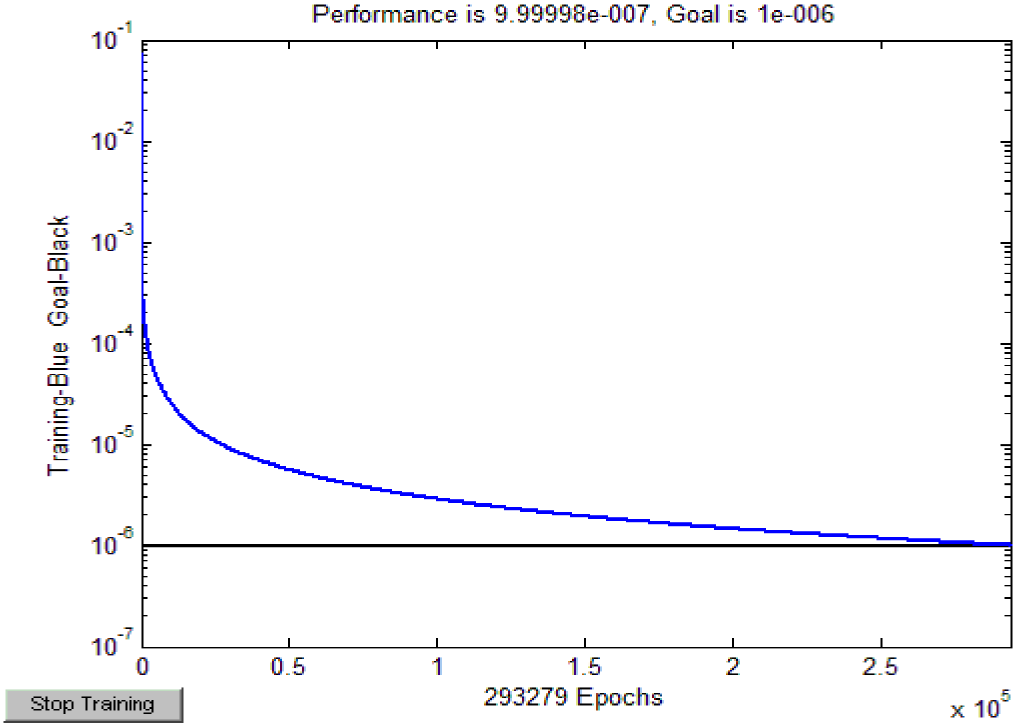


Figure 3 (d) Class – 4 of OCON Network.

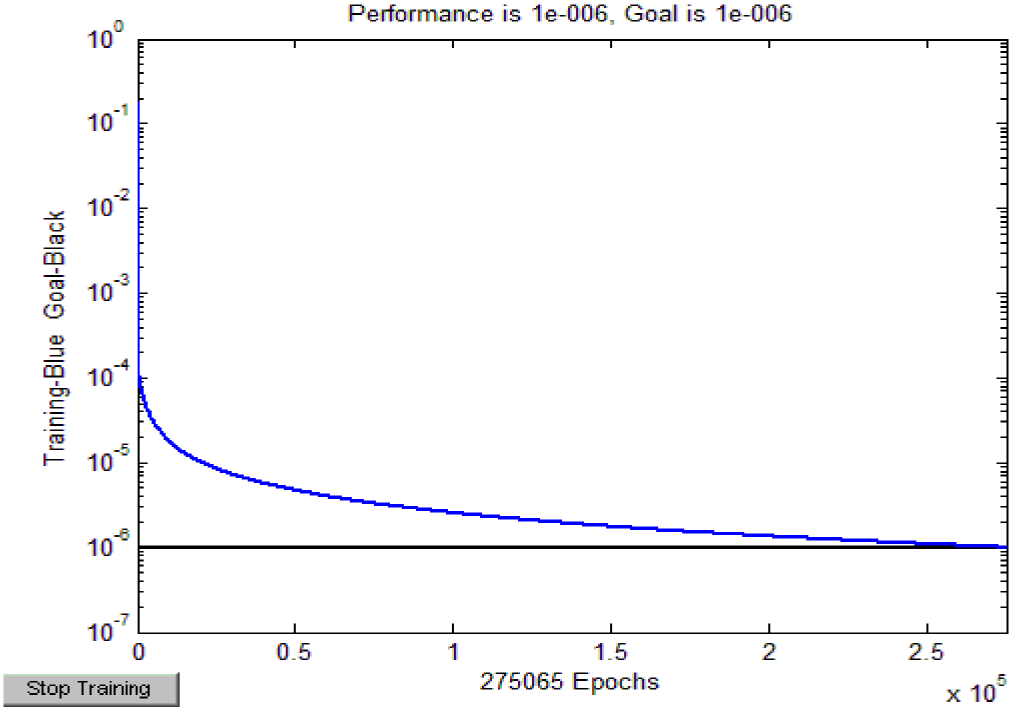


Figure 3 (e) Class – 5 of Ocon Network.

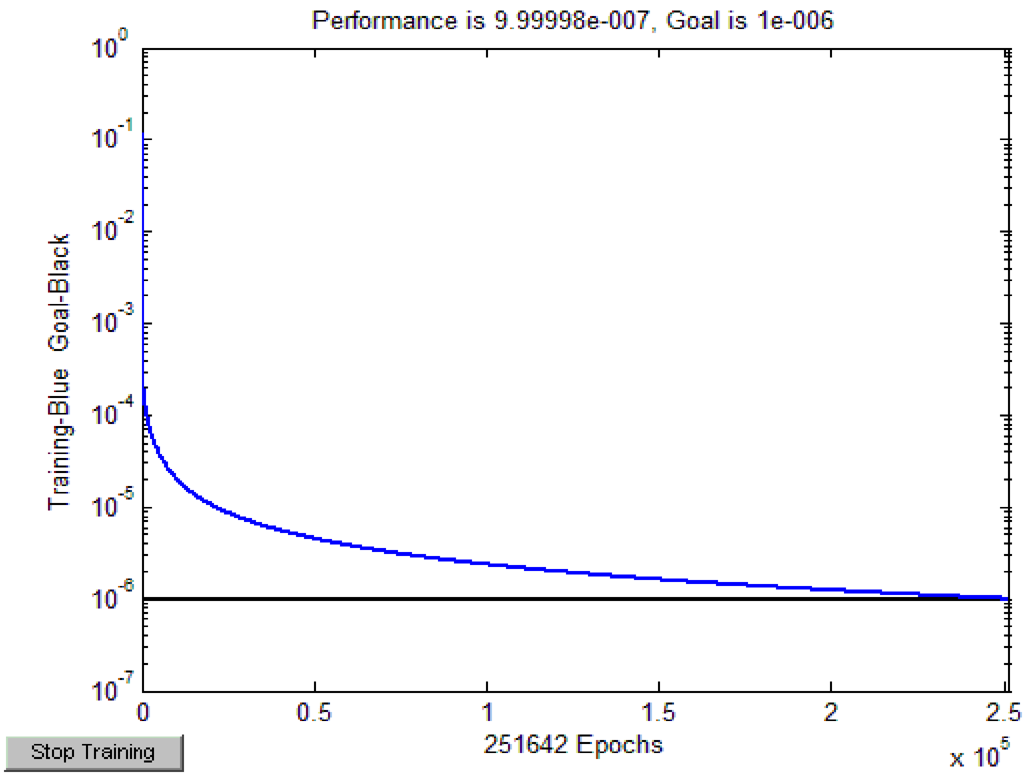


Figure 3 (f) Class – 6 of OCON Network.

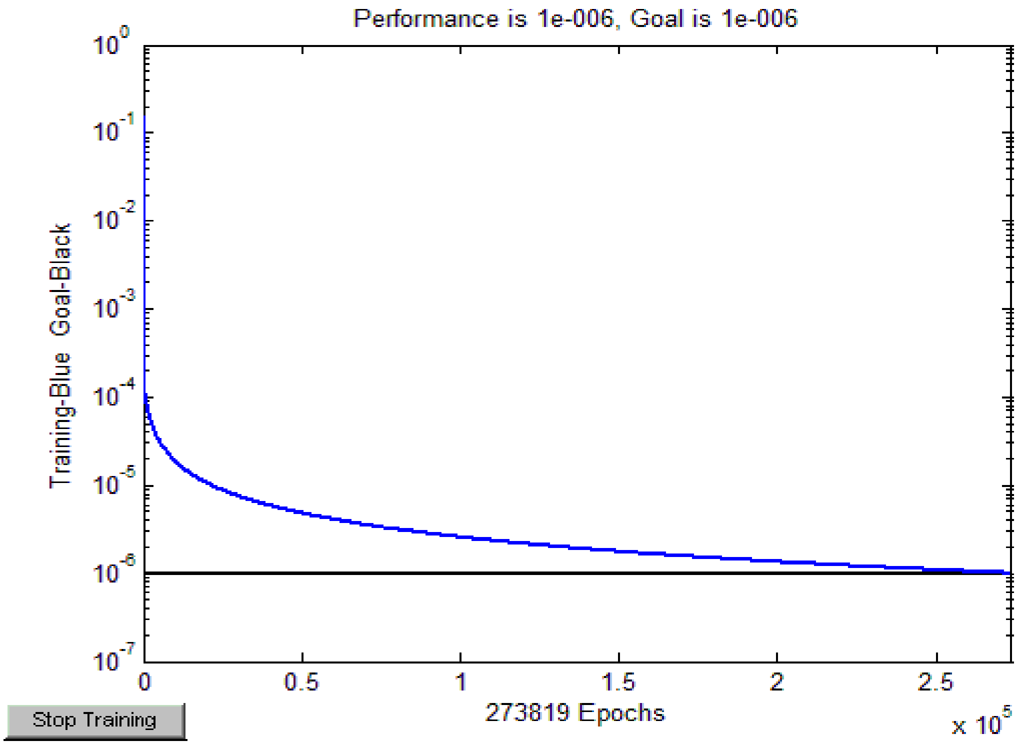


Figure 3 (g) Class – 7 of OCON Network.

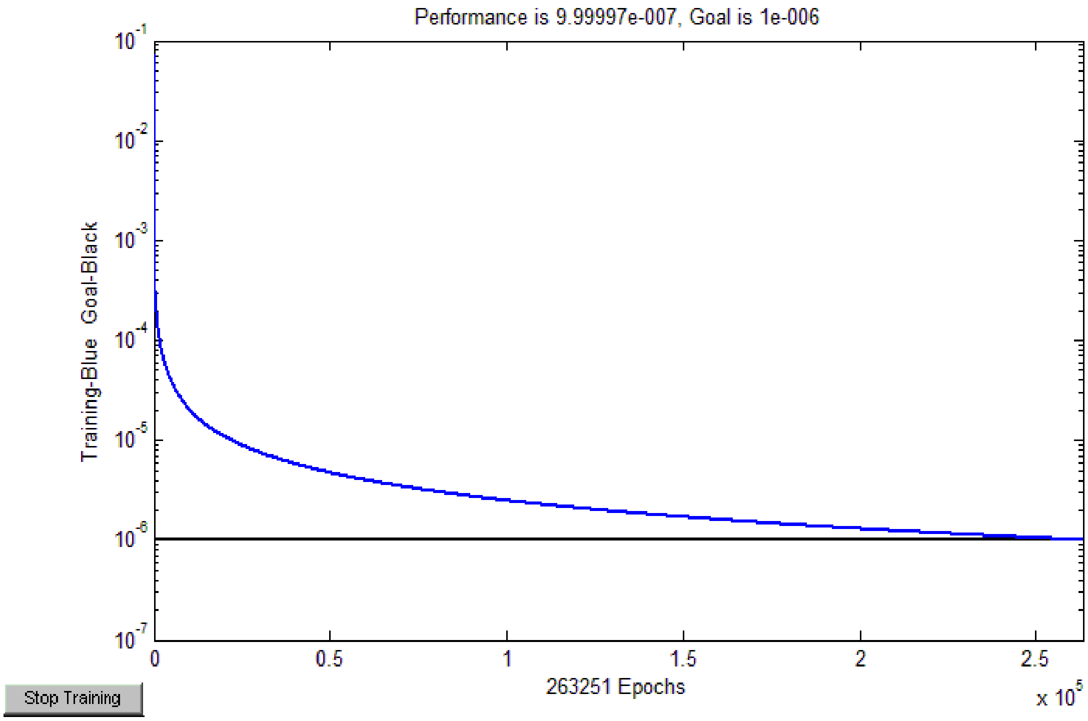


Figure 3 (h) Class – 8 of OCON Network.

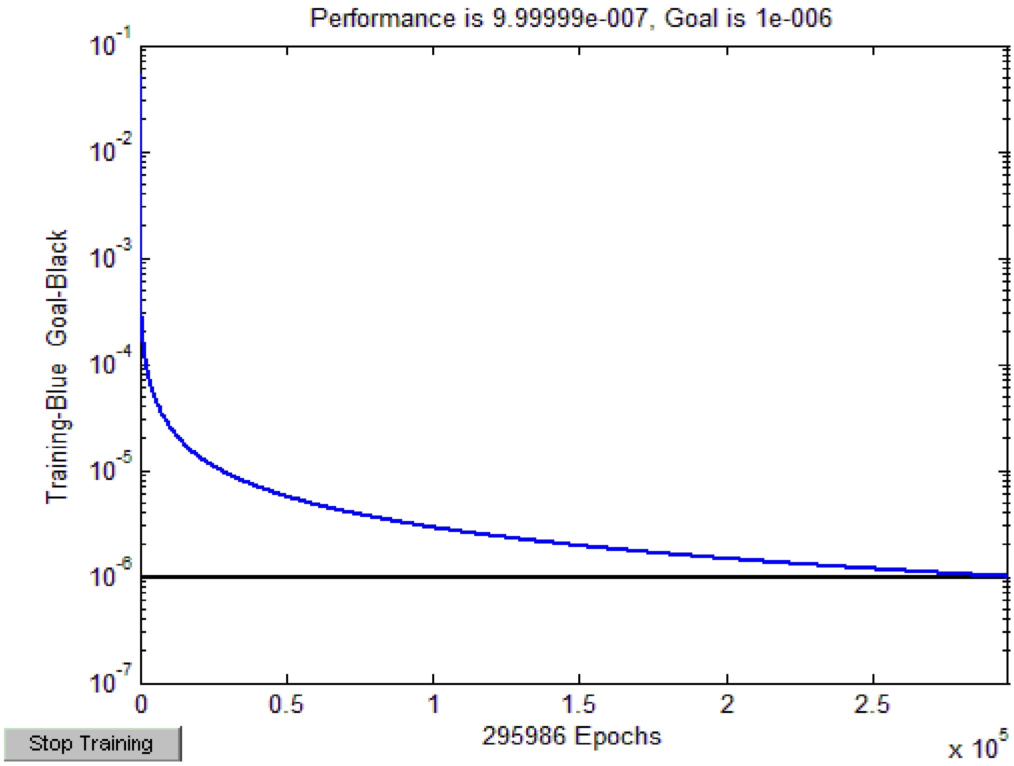


Figure 3 (i) Class – 9 of OCON Network.

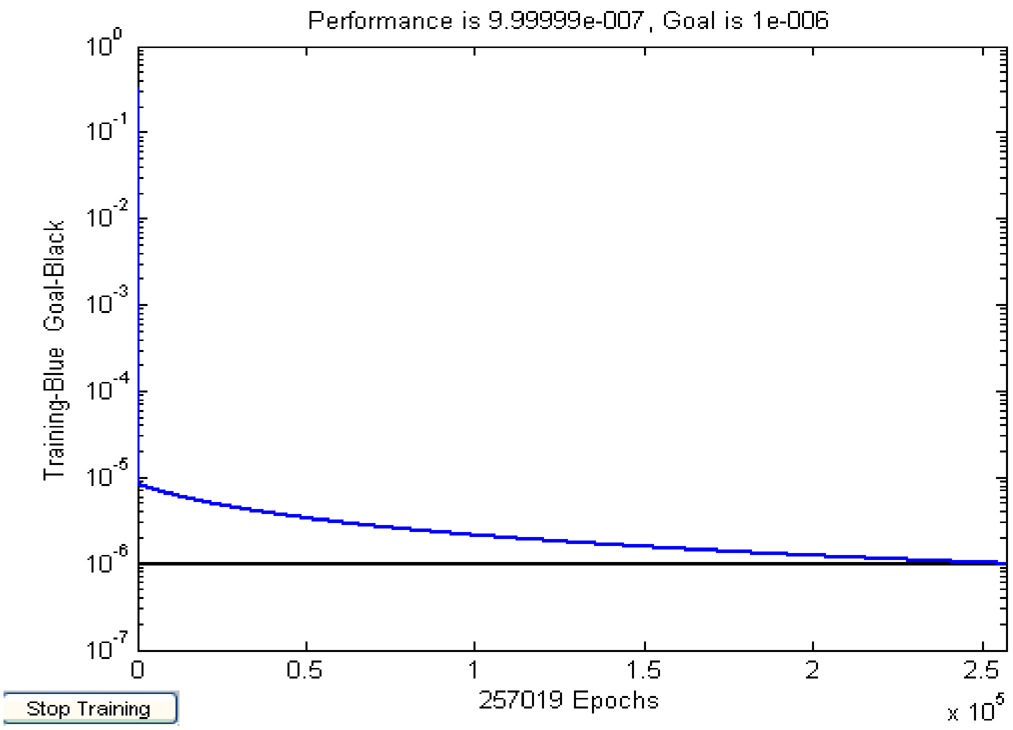
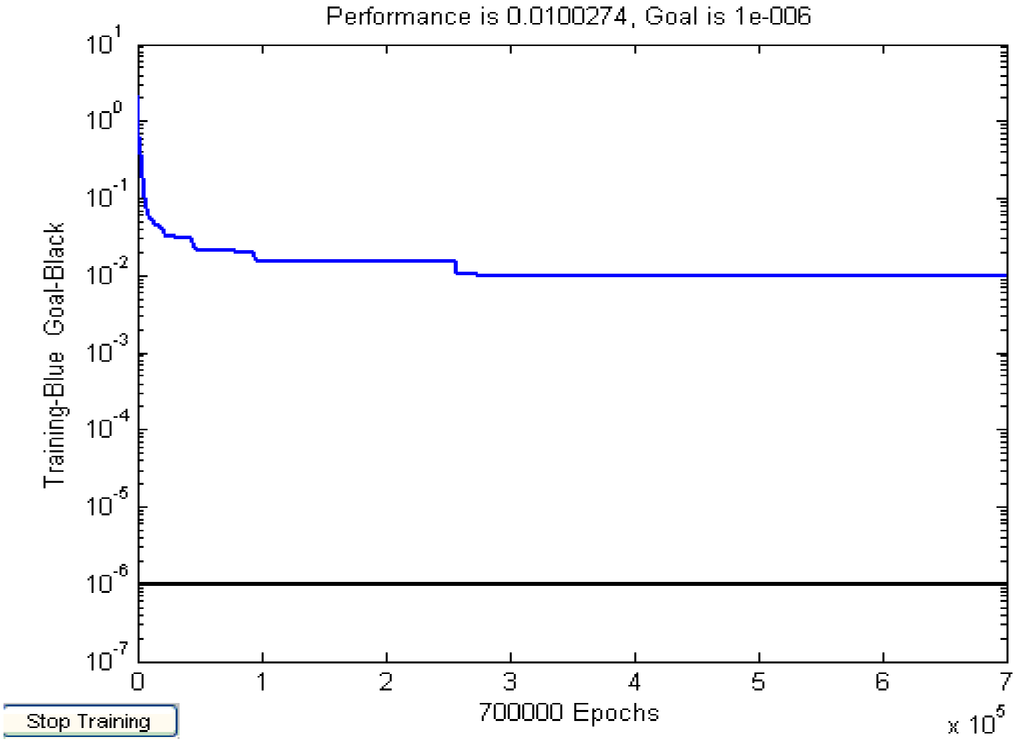


Figure 3 (j) Class – 10 of OCON Network.



3 (k) of ACON Network for all the classes.

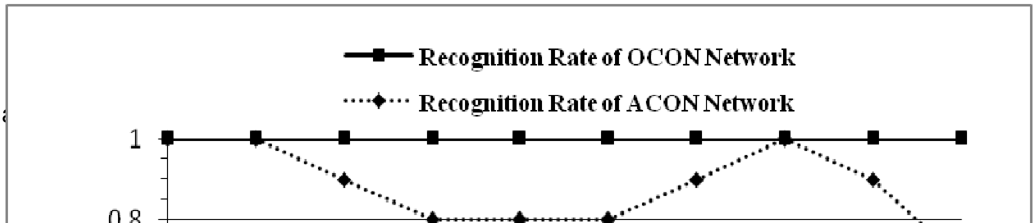


Figure 4: Graphical Representation of all Recognition Rate using OCON and ACON Network.

The OCON is an obvious choice in terms of speed-up and resource utilization. The OCON structure of neural network makes it most suitable for incremental training, i.e., network upgrading upon adding/removing memberships. One may argue that compared to ACON structure, the OCON structure is slow in retrieving time when the number of classes is very large. This is not true because, as the number of classes increases, the number of hidden neurons in the ACON structure also tends to be very large. Therefore ACON is slow. Since the computation time of both OCON and ACON increases as number of classes grows, a linear increasing of computation time is expected in case of OCON, which might be exponential in case of ACON.

5. CONCLUSION

In this paper, two general architectures for a Multilayer Perceptron (MLP) have been demonstrated. The first architecture is All-Class-in-One-Network (ACON) where all the classes are placed in a single network and the second one is One-Class-in-One-Network (OCON) where an individual single network is responsible for each and every class. Capabilities of these two architectures were compared and verified in solving human face recognition, which is a complex pattern recognition task where several factors affect the recognition performance like pose variations, facial expression changes, occlusions, and most importantly illumination changes. Both the structures were implemented and tested for face recognition purpose and experimental results show that the OCON structure performs better than the generally used ACON ones in term of training convergence speed of the network. Moreover, the inherent non-parallel nature of ACON has compelled us to use OCON for the complex pattern recognition task like human face recognition.

ACKNOWLEDGEMENT

Second author is thankful to the project entitled “Development of Techniques for Human Face Based Online Authentication System Phase-I” sponsored by Department of Information Technology under the Ministry of Communications and Information Technology, New Delhi-110003, Government of India Vide No. 12(14)/08-ESD, Dated 27/01/2009 at the Department of Computer Science & Engineering, Tripura University-799130, Tripura (West), India for providing the necessary infrastructural facilities for carrying out this work. The second author is also grateful to the Honorable Vice Chancellor Prof. A. Saha and Prof. B.K. De, Dean of Science of Tripura University (A Central University), Tripura, India for their kind support to carry out this research work.

REFERENCES

1. M. K. Bhowmik, ”Artificial Neural Network as a Soft Computing Tool – A case study”, In Proceedings of National Seminar on Fuzzy Math. & its application, Tripura University, November 25 – 26, 2006, pp: 31 – 46.

2. M. K. Bhowmik, D. Bhattacharjee and M. Nasipuri, “Topological Change in Artificial Neural Network for Human Face Recognition”, In Proceedings of National Seminar on Recent Development in Mathematics and its Application, Tripura University, November 14 – 15, 2008, pp: 43 – 49.

3. A. Abraham, “Artificial Neural Network”, Oklahoma Tate University, Stillwater, OK, USA.

4. http://www.acm.org/ubiquity/homepage.html.

5. I. Aleksander and H. Morton, “An introduction to Neural Computing,” Chapman & Hall, London, 1990.

6. R. Hecht-Nielsen, “Neurocomputing,” Addison-Wesley, 1990.

7. M. Turk and A. Pentland, “Eigenfaces for recognition”, Journal of Cognitive Neuro-science, March 1991. Vol. 3, No-1, pp. 71-86.

8. L. Sirovich and M. Kirby, “A low-dimensional procedure for the characterization of human faces,” J. Opt. Soc. Amer. A 4(3), pp. 519-524, 1987.

9. A. S. Georghiades, P. N. Belhumeur and D. J. Kriegnab, “From Few to Many: Illumination Cone Models for face Recognition under Variable Lighting and Pose”, IEEE Trans. Pattern Anal. Mach. Intelligence, 2001, vol. 23, No. 6, pp. 643 – 660.

10. D. Bhattacharjee, “Exploiting the potential of computer network to implement neural network in solving complex problem like human face recognition,” Proc. of national Conference on Networking of Machines, Microprocessors, IT, and HRD-need of the nation in the next millennium, Kalyani Engg. College, Kalyani, West Bengal, 1999.

11. Pradeep K. Sinha, “Distributed Operating Systems-Concepts and Design,” PHI, 1998. [8] M. K. Bhowmik, D. Bhattacharjee, M. Nasipuri, D. K. Basu and M. Kundu; “Classification of Fused Face Images using Multilayer Perceptron Neural Network”, proceeding of International Conference on Rough sets, Fuzzy sets and Soft Computing, Nov 5–7, 2009, organized by Department of Mathematics, Tripura University pp. 289-300.

12. M.K. Bhowmik, D. Bhattacharjee, M. Nasipuri, D.K. Basu and M. Kundu, “Classification of Polar-Thermal Eigenfaces using Multilayer Perceptron for Human Face Recognition”, proceedings of the 3rd IEEE Conference on Industrial and Information Systems (ICIIS-2008), IIT Kharagpur, India, Dec 8-10, 2008, pp. 118.

13. M.K. Bhowmik, D. Bhattacharjee, M. Nasipuri, D.K. Basu and M. Kundu, “Classification of Log-Polar-Visual Eigenfaces using Multilayer Perceptron for Human Face Recognition”, proceedings of The 2nd International Conference on Soft computing (ICSC-2008), IET, Alwar, Rajasthan, India, Nov 8–10, 2008, pp.107-123.

14. M.K. Bhowmik, D. Bhattacharjee, M. Nasipuri, D.K. Basu and M. Kundu, “Human Face Recognition using Line Features”, proceedings of National Seminar on Recent Advances on Information Technology (RAIT-2009), Indian School of Mines University, Dhanbad, Feb 6-7,2009, pp. 385-392.

15. P. Raviram, R.S.D. Wahidabanu Implementation of artificial neural network in concurrency control of computer integrated manufacturing (CIM) database, International Journal of Computer Science and Security (IJCSS), Volume 2, Issue 5, pp. 23-25, September/October 2008.

16. Teddy Mantoro, Media A. Ayu, “Toward The Recognition Of User Activity Based On User Location In Ubiquitous Computing Environments,” International Journal of Computer Science and Security (IJCSS)Volume 2, Issue 3, pp. 1-17, May/June 2008.

17. Sambasiva Rao Baragada, S. Ramakrishna, M.S. Rao, S. Purushothaman , “Implementation of Radial Basis Function Neural Network for Image Steganalysis,” International Journal of Computer Science and Security (IJCSS) Volume 2, Issue 1, pp. 12-22, January/February 2008.

A Novel Method for Quantitative Assessment of Software Quality

Neelam Bawane neelambawane@yahoo.com
Assistant Prof., Dept. of MCA
PES Institute Technology
Bangalore, 560085, India

C. V. Srikrishna cvsrikrishna@yahoo.co.in
Prof. & Head, Dept. of MCA
PES Institute Technology
Bangalore, 560085, India

Abstract

This paper deals with quantitative quality model that needs to be practiced formally through out the software development life cycle at each phase. Proposed quality model emphasizes that various stakeholders need to be consulted for quality requirements. The quality goals are set through various measurements and metrics. Software under development is evaluated against expected value of set of metrics. The use of proposed quantitative model is illustrated through a simple case study. The unaddressed quality attribute reusability in ISO 9126 is also discussed.

Keywords- quality assurance, quality metrics, quality model, stakeholders

1. INTRODUCTION

Quality can not be added to the system as an afterthought, instead it must be built into the system from the beginning. This paper proposes a quantitative software quality model which can evaluate requirements engineering phase of system development rigorously. The objective of this paper is to identify the need of a software quality model to be used as a foundation to Software Quality Engineering. The paper illustrates the use of the proposed model during system analysis & design and demonstrates usefulness through a case study. Software quality related literature in section II and a brief description of existing quality models in section III presents background knowledge about the topic considered in this paper. In Section IV, authors propose a quality model where need of early quality analysis and importance of collecting the requirements from various stakeholders are shown. A case study to demonstrate the application of proposed model is presented in Section V.

2. RELATED LITERATURE

According to Gordon [12], software quality is an important characteristic affecting overall system development lifecycle cost, performance and useful life. Increasing demands from the marketplace for a greater emphasis on quality in software products are promising to revolutionize good practice of software engineering [4].

It is now well established that production is meaningless without assessment of product quality. "Quality is a complex and multifaceted concept." Garvin [10] describes quality from five different perspectives: transcendental view, user view, manufacturers view, product view, and value-based view.

Kitchenham and Pfleeger [7] have recently discussed Garvin's [9] approach in the context of software product quality, accordingly Garvin's model is a useful starting point not as a quality model in its own right but rather as a specification of a set of requirements for quality models or alternatively as a set criteria for evaluating product quality models.

The fact "quality must be monitored from the early phase such as requirements analysis and design" provides need of Software Quality Assurance (SQA) [5]. The aim of the SQA organization is to assure that the standards, procedures and policies used during software development are adequate to provide the level of confidence required for the process or product. SQA is defined as "a planned and systematic pattern of all actions necessary to provide adequate confidence that the item or product conforms to established technical requirements".

With increasing importance placed on standard quality assurance methodologies by large companies and government organizations, software companies have implemented rigorous quality assurance (QA) processes to ensure that these standards are met [20]. Various software quality assurance models have been developed by different organizations to ensure that specific standards are met and to give guidelines on achieving these standards [15].

Bourque [27] also suggests that the implementation of quality in a software product is an effort that should be formally managed throughout the Software Engineering lifecycle. Such an approach to the implementation of quality leads to Software Quality Engineering (SQE). Suryn [34] has suggested that SQE is an application of a continuous, systematic, disciplined, quantifiable approach to the development and maintenance of quality of software products and systems. Georgiadou [10] demonstrated that more mature the process and its underlying lifecycle model, earlier the identification of errors in the deliverables.

Thus, to achieve quality in software processes and products, it is necessary to develop systematic measurement programs, compatible with the organizational objectives and tailored to the quality aspects that are being considered [30]. There is a need to establish baselines of performance for quality, productivity and customer satisfaction by the organizations. These baselines are used to document current performance and improvements by showing deviations from the baseline. In order to establish a baseline, a model must be established [26]. A quality model is a schema for better explanation of our view on quality. Some existing quality models can predict fault-proneness with reasonable accuracy in certain contexts. Few standard models are discussed in next section.

3. STANDARD MODELS

In the past years the scientific and industrial communities have proposed many QA standards and models. According to Moore [20] "there are more than 300 standards developed and maintained by more than 50 different organizations." Most popular model is ISO/IEC 9126 [16], which specify requirements for a quality management system within an organization.

The metrics listed in ISO/IEC TR 9126-2 are not intended to be an exhaustive set. Users can select or modify and apply metrics and measures from ISO/IEC TR 9126-2:2003 or may define application-specific metrics for their individual application domain. Software metrics are the only mechanized tools for assessing the value of internal attributes [17]. Software metrics are defined as "standard measurements, used to judge the attributes of something being measured, such as quality or complexity, in an objective manner" [24].

ISO/IEC 9000:2005 [13] provides guidance for the use of the series of International Standards named Software product Quality Requirements and Evaluation (SQuaRE). Software Quality in the Development Process (SQUID) [18] allows the specification, planning, evaluation and control of software quality through the software development process. SQUID uses external and internal quality measures defined in ISO 9126. Although the existence of documentation is a key requirement of a functional ISO 9001 Quality Management System (QMS), it is not in itself sufficient. To develop and implement a fully functional ISO 9001 QMS, it is essential that a small/medium-sized enterprises correctly identifies the initial state of its QMS and the path it will follow to achieve the desired state [1].

Capability Maturity Model (CMM) proposed by Software Engineering Institute (SEI) provides a framework for continuous software process improvement [31]. The key notion is that CMM provides guidelines for conducting audits, testing activities, and for process improvement. The CMM approach classifies the maturity of the software organization and practices into five levels describing an evolutionary process from chaos to discipline [31] as Initial (chaotic), Repeatable (project management), Defined (institutionalized), Managed (quantified), Optimizing (process improvement).

McCall's model [19] of software quality incorporates 11 criteria encompassing three main perspectives for characterizing the quality attributes of a software product. These perspectives are Product revision (ability to change), Product transition (adaptability to new environments), and Product operations (basic operational characteristics)

Boehm's model [8] is based on a wider range of characteristics and incorporates 19 criteria. At the highest level of his model, Boehm defined three primary uses (basic software requirements), these three primary uses are as-is utility, the extent to which the as-is software can be used (i.e. ease of use, reliability and efficiency), Maintainability, ease of identifying what needs to be changed as well as ease of modification and retesting, Portability, ease of changing software to accommodate a new environment

FURPS developed by Hewlett-Packard takes five characteristics of quality attributes - Functionality, Usability, Reliability, Performance and Supportability. When the FURPS model is used, two steps are considered: setting priorities and defining quality attributes that can be measured [21]. One disadvantage of this model is that it does not take into account the software product’s portability [25].

Dromey [29] proposes a working framework for building and using a practical quality model to evaluate requirement determination, design and implementation phases. Dromey includes high-level quality attributes: functionality, reliability, efficiency, usability, maintainability, portability, reusability and process mature. In comparing to ISO 9126, additional characteristics like process maturity and reusability are noticeable.

Georgiadou [11] developed a generic, customizable quality model (GEQUAMO) which enables any stakeholder to construct their own model depending on their requirements. In a further attempt to differentiate between stakeholders, Siaka et al [22] studied the viewpoints of users, sponsors and developers as three important constituencies/stakeholders and suggested attributes of interest to each constituency as well as level of interest. More recently, Siaka and Georgiadou [23] reported the results of a survey amongst practitioners on the importance placed on product quality characteristics. Using their empirical results they extended ISO 9126 by adding two new characteristics namely Extensibility and Security which have gained in importance in today’s global and inter-connected environment.

Basili and Rombach [33] define a goal-based measurement program. The concept of the goal/question/metric paradigm is to identify goals, translate into the questions that need to be answered to determine if one is meeting or moving towards these goals, and then selecting metrics that provide information to help answer these questions.

The criteria in all above models are not independent. They interact with each other and often cause conflict, especially when software providers try to incorporate them into the software development process. There are a number of difficulties in the direct application of any of the above models. The models are static since they do not describe how to project the metrics from current values to values at subsequent project milestones. It is important to relate software metrics to progress and to expected values at the time of delivery of the software [3].

To formulate the requirements of a quantitative quality model for software development, three issues must be addressed: the different interest groups need to be identified; the intended applications of the model need to be spelled out; and it is necessary to establish the quality needs or perspectives/views of the different interest groups [28].

4. PROPOSED QUALITY MODEL

According to Dromey [28] "The first task in building a software product quality model is to identify what the intended applications of the model are and to address the needs of the different interest groups that will use the model in different applications." The attributes of a quality model should be sufficient to meet the needs of all interest groups associated with the software.

Proposed quantitative model (Figure 1) keeps quality attributes a central consideration during application analysis and design. There are often a number of stakeholders involved in the design process with each having different quality goals. The model suggests the method of analyzing the stakeholders' quality requirements and computes the relative priorities among the quality attributes and their subcharacteristics.

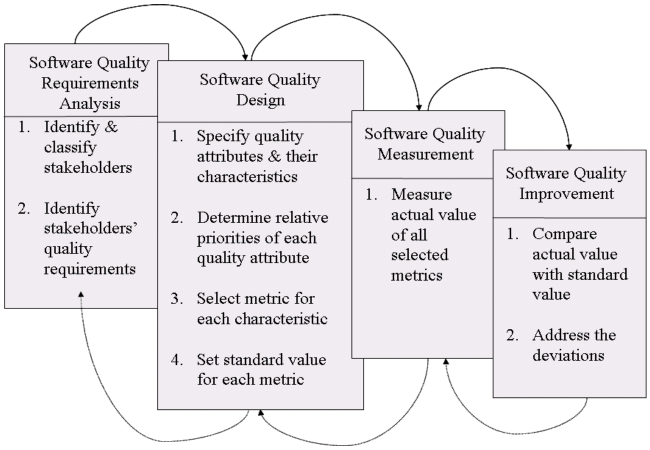

Software Quality Requirements Analysis
1. Identify & classify stakeholders
2. Identify stakeholders' quality requirements
Software Quality Design
1. Specify quality attributes & their characteristics
2. Determine relative priorities of each quality attribute
3. Select metric for each characteristic
4. Set standard value for each metric
Software Quality Measurement
1. Measure actual value of all selected metrics
Software Quality Improvement
1. Compare actual value with standard value
2. Address the deviations

FIGURE1: Quantitative Quality Model

An integral part of an effective user-centered development is the collection of requirements and the identification of common user tasks. A number of methods can be used to gather the requirements and to identify the task groups. For quantifying the relative priorities of quality attributes and their subcharacteristics of software design, the constant sum pair wise comparison method of Analytical Hierarchy Process (AHP) [2, 32] is employed. Proposed quality model has four major phases which are discussed in following sections:

4.1 Software Quality Requirements Analysis (SQRA)

In managing customer's quality expectations, relevant views and attributes need to be reviewed first, because different quality attribute may have different levels of importance to different customers and users [16]. For example, reliability is a prime concern for the business and commercial software systems because of people reliance on them and the substantial financial loss if they malfunction. SQRA is customized by software category, development phase and users' requirements.

Considering all requirements, quality factors are chosen from the set of factors given by McCall quality model [19] and ISO 9126. Each quality factor is further defined by a set of attributes, called criteria, which provide the qualitative knowledge about customers' quality expectations. This qualitative knowledge helps to quantify the goals in the software quality design (SQD). SQRA can be carried out as follows:

1) Identification & classification of stakeholders

Authors emphasize focus on identification of important stakeholders. Interview is a common method that can be employed in requirements analysis. Each stakeholder will have an idea of their expectation and visualization of their requirements. Various stakeholders may be grouped based on similar characteristics. The proposed model classifies stakeholders based on their jobs as these are directly related to quality preferences.

2) Identification of stakeholders' quality requirements
Stakeholders' quality requirements can be gathered through existing requirements gathering techniques such as Win Win requirement gathering method [6] and the goal oriented requirement gathering method [30]. External review team reviews the requirements and may add a set of quality requirements. Requirements of various groups are tabulated (see case study).

4.2 Software Quality Design (SQD)

Once all the quality requirements are listed, each attribute is quantified by individual metric as measurement is essential if quality is to be achieved. Various metrics are available for different quality characteristics. Requirements are customized to products and users, thus expected values of the corresponding metrics are determined for the product under development. Necessary steps can be followed to determine priorities of quality characteristics, related metrics and expected values which are as follows:

1) Specify quality attributes and their corresponding characteristics to satisfy stakeholders' quality requirements
Quality attributes and their subcharacteristics are specified for each stakeholder group based on ISO 9126. It is not easy to translate a user requirement (informal quality requirements) into a set of quality characteristics (formal specification of the software quality as defined in ISO/IEC 9126)

2) Determine the relative priorities of subcharacteristics of each quality attribute
The relative priorities of quality characteristics are computed for each stakeholders group. 100 points are allocated between each pair of quality attribute at a time based on their preferences for the attributes to make comparative judgment. The AHP [32] is applied to transform judgment values to normalized measures. The use of AHP to obtain the initial values is systematic quantification of the stakeholders' expectations, as it is the subjective judgments of the stakeholders.

3) Select the metrics for each characteristic
Metrics can be selected for each of the characteristics based on the concept of goal/question/metric.

4) Set the standard value for each metric
Based on the priorities and standards defined by ISO/IEC TR 9126, a standard value is associated with each metric to achieve the characteristic expected by stakeholders' groups. Once specific quality goals, expectations, measurements and metrics are set, related tools, techniques and procedures to achieve these goals can be selected.

4.3 Software Quality Measurement

The measurement offers visibility into the ways in which the processes, products, resources, methods, and technologies of software development relate to one another. According to the ISO/IEC 15939 Software Engineering – Software Measurement Standard decision criteria are the “thresholds, targets, or patterns used to determine the need for action or further investigation or to describe the level of confidence in a given result”.

1) Measure the actual value
During development life cycle, in each phase, the values of selected metrics can be measured, compared and analyzed with respect to standard set values.

4.4 Software Quality Improvement

The feed back on measurement step enables the software engineers to assess the quality at each development stage and in turn helps to improve whenever violation from set goals is found.

1) Compare the actual value with standard value for each metric
Deviations may be encountered in the project plan, process description, applicable standards or technical work. Recommendations derived from the interpretation of actual value and established of the metrics are provided to the development engineers for improvement.

2) Address the deviations
The product can be adjusted to make things better and improved. It is important to establish a follow up procedure to ensure that items on the issues list have been properly corrected.

5. CASE STUDY

The case study considered in this paper is related to an automated application for job consultancy firm. A survey is carried out for various stakeholders based on their knowledge, interest and job responsibilities. Stakeholders are classified in three groups as given in table-1.

No.	Group
1	Manager
2	Developer
3	User

TABLE 1: STAKEHOLDERS GROUPS

The quality requirements for the system under development, identified by different stakeholders' groups are tabulated in table-2.

No	Group	Quality requirements		
1	Manager	Team size	Cost	Delivery time
2	Developer	Maintainability	Portability	Reusability
3	User	Reliability	Usability	Efficiency

TABLE 2: QUALITY REQUIREMENTS OF STAKEHOLDERS GROUPS

Quality attributes listed in table-2 under group developer are maintainability, portability, reusability that is further divided into subcharacteristics. This division follows quality analysis framework given by ISO 9126. Quality attributes and their subcharacteristics for the developer group are shown in table-3 as a sample. However reusability is not part of quality framework given by ISO 9126 and is addressed in this paper.

Maintainability (QA1)	Analyzability (SC11)
	Changeability (SC12)
	Stability (SC13)
	Testability (SC14)
Portability (QA2)	Adaptability (SC21)
	Installability (SC22)
	Conformance (SC23)
	Replaceability (SC24)
Reusability (QA3)	Coupling (SC31)
	Comprehensibility (SC32)
	Interoperability (SC33)

TABLE 3: QUALITY ATTRIBUTES AND SUBCHARACTERISTICS FOR DEVELOPER GROUP

A total of 100 points is distributed between each two attributes. This distribution shows their ratio scale for prioritizing quality attributes and their corresponding subcharacteristics. The relative priorities are computed by AHP constant sum method. Three experts from the developer group are identified for prioritization considering three related attributes and corresponding subcharacteristics (refer appendix A). Average and relative of the priorities of quality attributes and their subcharacteristics are computed.

5.1 Average Priorities

QA1 = Maintainability, QA2 = Portability, QA3=Reusability

QA1	60	QA1	45	QA2	45
QA2	40	QA3	55	QA3	55

SC11 = Analyzability, SC12 = Changeability, SC13 = Stability, SC14 = Testability

SC11	60	SC11	60	SC11	60
SC12	40	SC13	40	SC14	40
SC12	40	SC12	40	SC13	50
SC13	60	SC14	60	SC14	50

SC21 = Adaptability, SC22 = Installability, SC23 = Conformance, SC24 = Replaceability

SC21	55	SC21	60	SC21	55
SC22	45	SC23	40	SC24	45
SC22	50	SC22	45	SC23	40
SC23	50	SC24	55	SC24	60

SC31 = Coupling, SC32 = Comprehensibility, SC33 = Interoperability

SC31	55	SC32	45	SC31	60
SC32	45	SC33	55	SC33	40

5.2 Relative Priorities

QA = {0.354, 0.271, 0.375}
SC1 = {0.332, 0.180, 0.244, 0.244,
SC2 = {0.302, 0.223, 0.202, 0.273}
SC3 = {0.403, 0.289, 0.308}

For each required characteristic, appropriate metrics are chosen and their expected values are set based on the priorities calculated. Few relevant metrics are shown in table-4 (Source: ISO9126).

Maintainability (QA1)	Analyzability (SC11)	Reliability index
		Comment percentage
		Cyclomatic complexity
	Changeability (SC12)	Code duplication
		Maximum number of references violation
	Stability (SC13)	Correlation of complexity / size
		Global variables usage
	Testability (SC14)	Cyclomatic complexity
Portability (QA2)	Adaptability(SC21)	Mean efforts to adapt
	Installability (SC22)	Installation efforts in Man-months
		Parameter change ratio
	Conformance (SC23)	Standard conformance ration
	Replaceability (SC24)	Function change ratio
		Source code change ratio
Reusability (QA3)	Coupling (SC31)	Cohesion and coupling metrics (Fan-in & Fan-out)
	Comprehensibility (SC32)	Comment percentage
	Interoperability (SC33)	Size of domain independent part

TABLE 4: METRICS AND THEIR EXPECTED VALUES FOR QUALITY SUBCHARACTERISTICS

Similarly, other users' quality requirements are analyzed and specified. All the metrics are measured during development as and when required, and compared with the expected values for the conformance of quality characteristics expected by various stakeholders. On occurrence of any deviation, desired changes are made that shows improvement in product quality.

6. CONCLUSION

Aim of this research paper is to provide the model to establish the quality requirements expected by various stakeholders and to incorporate these requirements in the product under development. Proposed quantitative quality model takes a set of quality requirements as input for the development of a software application. The model is dynamic and allows product deliverables to be compared with set goals by various stakeholders through measurements and metrics throughout the development life cycle. The case study validates the suitability and usefulness of the proposed model. The quality attribute reusability is discussed in addition to other quality attributes of ISO 9126.

ACKNOWLEDGEMENT

The authors acknowledge the support extended by the management of PES Institute during the course of research.

7. REFERENCES

[1] Andres Sousa-Poza, Mert Altinkilinc, Cory Searcy, “Implementing a Functional ISO 9001 Quality Management System in Small and Medium-Sized Enterprises”, International Journal of Engineering, v. 3 Issue 3, 2009

[2] Arun Sharma, Rajesh Kumar, P.S. Grover, "Dependency Analysis for Component-Based Software Systems", ACM SIGSOFT Software Engineering Notes, v.34 n.4, July 2009 [DOI: 10.1145/1543405.1543424]

[3] Arun Sharma, Rajesh Kumar, P.S. Grover, “Managing Component-Based Systems With Reusable Components”, International Journal of Computer Science and Security, v. 1 Issue 2, 2007

[4] Ashley Williams, “The documentation of quality engineering: applying use cases to drive change in software engineering models”, SIGDOC '04: Proceedings of the 22nd annual international conference on Design of communication: The engineering of quality documentation, October 2004

[5] Avadhesh Kumar, Rajesh Kumar, P.S. Grover, “An Evaluation of Maintainability of Aspect-Oriented Systems: a Practical Approach”, International Journal of Computer Science and Security, v. 1 Issue 2, 2007

[6] B. Boehm et al., “Win Win requirements negotiation process: A multi project analysis”, Proceedings of the International Conference on Software Process, Chicago, 1998

[7] B. Kitchenham, S. L. Pfleeger, “Software Quality: The Elusive Target”, IEEE Software, vol. 13(1), pp.12-21, 1996

[8] B. W. Boehm, J. R. Brown, H. Kaspar, M. Lipow, G. McLeod, and M. Merritt, “Characteristics of Software Quality”, North Holland, (1978)

[9] D. Garvin, "What Does 'Product Quality' Really Mean?" Sloan Management Review, Fall, pp 25-45, 1984

[10] E. Georgiadou “Software Process and Product Improvement: A Historical Perspective”, International Journal of Cybernetics, Volume 1, No1, pp172-197, Jan 2003

[11] E. Georgiadou, “GEQUAMO– A Generic, Multilayered, Customisable, Software Quality Model”, International Journal of Cybernetics, Volume 11, Number 4 , pp 313-323 November 2003

[12] Gordon W. Skelton, “Integrating total quality management with software engineering education”, ACM SIGCSE Bulletin, Volume 25 Issue 2, June 1993

[13] ISO 9000:2005 Quality management systems Fundamentals and vocabulary, 2005

[14] ISO 9001:2000 Quality management systems Requirements, 2001

[15] ISO 9004:2000 Quality management systems Guidelines for performance improvement, 2000

[16] ISO/IEC, IS 9126-1, “Software Engineering – Product Quality – Part 1: Quality Model”, Geneva Switzerland: International Organization for Standardization, 2001

[17] ISO: ISO/IEC 14598-1. International Standard Information technology software product evaluation, 1999

[18] J. Bøegh, S. DePanfilis, B. Kitchenham, A. Pasquini, “A Method for Software Quality Planning, Control and Evaluation”. IEEE Software, 69-77, March/April 1999

[19] J.A. McCall, P.K. Richards and G. F. Walters, "Factors in software quality", Griffiths Air Force Base, N.Y. : Rome Air Development Center Air Force Systems Command, 1977

[20] J.W. Moore, "Software Engineering Standards: A User's Road Map", IEEE Computer Society, Los Alamitos, CA, 1998

[21] K. Khosravi, & Y.G. Gueheneuc, "A quality model for design patterns", http://www.yann_gael.gueheneuc.net/work/Tutoring/Documents/041021+Khosravi+Technical+Report. doc.pdf, 2004

[22] K.V. Siaka, E. Berki, E. Georgiadou, C. Sadler, "The Complete Alphabet of Quality Software Systems: Conflicts and Compromises", 7th World Congress on Total Quality & Qualex 97, New Delhi, India, 17-19, February 1997

[23] K.V. Siaka, E. Georgiadou, "PERFUMES: A Scent of Product Quality Characteristics", SQM, UK, March 2005

[24] M. Lorenz, J. Kidd, "Object-Oriented Software Metrics", 1st Ed. Prentice Hall,1994

[25] M. Ortega, M. Perez, & T. Rojas, "Construction of a systemic quality model for evaluating a software product", Software Quality Journal 11: 219-242, 2003

[26] N. E. Fenton, and M. Neil, "A Critique of Software Defect Prediction Models", IEEE Transactions on Software Engineering (25:5), pp.675-689, September/October 1999

[27] P. Bourque, R. Dupuis, A. Abran, J.W. Moore, L.L. Trippet S. Wolff, "Fundamental Principles of Software Engineering - A Journey", Journal of Systems and Software, 2000

[28] R. G. Dromey, "A Model for Software Product Quality", IEEE Trans. Soft. Eng., pp 146-162, 1995

[29] R. G. Dromey, "Software product quality: Theory, model and practice. Software Quality Institute," Griffith University, Brisbane, Technical Report, 1999

[30] S. Godbole, "Software Quality Assurance: Principles and Practices", Alpha Science International Ltd., 2004

[31] Software Engineering Institute, "The Capability Maturity Model: Guidelines for Improving the Software Process", MA: Addison-Wesley, 1994

[32] T. L. Saaty, "The Analytic Hierarchy Process", McGraw Hill, Inc., New York NY, 1980

[33] V.R. Basili and H.D. Rombach, "The TAME projects: Towards improvement-oriented software environment", IEEE Transactions in Software Engineering, 14, no.6, Nov 1988

[34] W. Suryn, "Course notes SYS861", École de Technologie Supérieure, Montréal, 2003

APPENDIX

Developer group

a) Preferences of stakeholder 1

QA1 = Maintainability, QA2 = Portability, QA3 = Reusability

QA_1	60	QA_1	50	QA_2	45
QA_2	40	QA3	50	QA3	55

SC11 = Analyzability, SC12 = Changeability, SC13 = Stability, SC14 = Testability

SC_{11}	65	SC_{11}	60	SC_{11}	55
SC_{12}	35	SC_{13}	40	SC_{14}	45
SC_{12}	40	SC_{12}	35	SC_{13}	50
SC_{13}	60	SC_{14}	65	SC_{14}	50

SC21 = Adaptability, SC22 = Installability, SC23 = Conformance, SC24 = Replaceability

SC_{21}	55	SC_{21}	60	SC_{21}	50
SC_{22}	45	SC_{23}	40	SC_{24}	50
SC_{22}	50	SC_{22}	45	SC_{23}	45
SC_{23}	50	SC_{24}	55	SC_{24}	55

SC31 = Coupling, SC32 = Comprehensibility, SC33 = Interoperability

SC_{31}	60	SC_{32}	45	SC_{31}	60

SC_{32}	40	SC_{33}	55	SC_{33}	40

b) Preferences of stakeholder 2

QA1 = Maintainability, QA2 = Portability, QA3 = Reusability

QA_1	65	QA_1	55	QA_2	45
QA_2	35	QA3	45	QA3	55

SC11 = Analyzability, SC12 = Changeability, SC13 = Stability, SC14 = Testability

SC_{11}	60	SC_{11}	65	SC_{11}	55
SC_{12}	40	SC_{13}	35	SC_{14}	35
SC_{12}	45	SC_{12}	40	SC_{13}	55
SC_{13}	55	SC_{14}	60	SC_{14}	45

SC21 = Adaptability, SC22 = Installability, SC23 = Conformance, SC24 = Replaceability

SC_{21}	60	SC_{21}	55	SC_{21}	55
SC_{22}	40	SC_{23}	45	SC_{24}	45
SC_{22}	50	SC_{22}	50	SC_{23}	40
SC_{23}	50	SC_{24}	50	SC_{24}	60

SC31 = Coupling, SC32 = Comprehensibility, SC33 = Interoperability

SC_{31}	55	SC_{32}	50	SC_{31}	55
SC_{32}	45	SC_{33}	50	SC_{33}	45

c) Preferences of stakeholder 3

QA1 = Maintainability, QA2 = Portability, QA3 = Reusability

QA_1	55	QA_1	40	QA_2	45
QA_2	45	QA3	60	QA3	55

SC11 = Analyzability, SC12 = Changeability, SC13 = Stability, SC14 = Testability

SC_{11}	55	SC_{11}	55	SC_{11}	60
SC_{12}	45	SC_{13}	45	SC_{14}	40
SC_{12}	45	SC_{12}	40	SC_{13}	50
SC_{13}	55	SC_{14}	60	SC_{14}	50

SC21 = Adaptability, SC22 = Installability, SC23 = Conformance, SC24 = Replaceability

SC_{21}	50	SC_{21}	65	SC_{21}	50
SC_{22}	50	SC_{23}	35	SC_{24}	50
SC_{22}	55	SC_{22}	40	SC_{23}	40
SC_{23}	45	SC_{24}	60	SC_{24}	60

SC31 = Coupling, SC32 = Comprehensibility, SC33 = Interoperability

SC_{31}	55	SC_{32}	40	SC_{31}	60
SC_{32}	45	SC_{33}	60	SC_{33}	40

Hierarchical Non-blocking Coordinated Checkpointing Algorithms for Mobile Distributed Computing

Surender Kumar ssjangra20@rediffmail.com
Deptt. of IT,
H.C.T.M
Kaithal (HRY), 136027, INDIA

Parveen Kumar pk223475@yahoo.com
Deptt. of CSA,
M.I.E.T
Meerut (U.P), INDIA

R.K. Chauhan rkc.dcsa@gmail.com
Deptt. of CSA,
K.U.K
Kurukshetra (HRY), INDIA

Abstract

Mobile system typically uses wireless communication which is based on electromagnetic waves and utilizes a shared broadcast medium. This has made possible creating a mobile distributed computing environment and has brought us several new challenges in distributed protocol design. So many issues such as range of transmission, limited power supply due to battery capacity and mobility of processes. These new issue makes traditional recovery algorithm unsuitable. In this paper, we propose hierarchical non blocking coordinated checkpointing algorithms suitable for mobile distributed computing. The algorithm is non-blocking, requires minimum message logging, has minimum stable storage requirement and produce a consistent set of checkpoints. This algorithm requires minimum number of processes to take checkpoint.

Keywords: Co-ordinated Checkpointing, fault tolerant, Non-blocking approach, Mobile Computing System.

1. INTRODUCTION

The market of mobile handheld devices and mobile application is growing rapidly. Mobile terminal are become more capable of running rather complex application due to the rapid process of hardware and telecommunication technology. Property, such as portability and ability to connect to network in different places, made mobile computing possible. Mobile computing is the performance of computing tasks whiles the user in on the move, or visiting place other than their usual environment. In the case of mobile computing a user who is away from his “home” environment can still get access to different resources that are too computing or data intensive to reside on the mobile terminal [4].Mobile distributed systems are based on wireless networks that are known to suffer from low bandwidth, low reliability, and unexpected disconnection [3].

Checkpointing / rollback recovery strategy has been an attractive approach for providing fault tolerant to distributed applications [1] [16]. Checkpoints are periodically saved on stable storage and recovery from a processor failure is done by restoring the system to the last saved state. So the system can avoid the total loss of the computation in case of the failure. In a distributed system, since the processes in the system do not share memory, a global state of the system is defined as a set of local states, one from each process. An orphan message is a message whose receive event is recorded, but its sent event is lost. A global state is said to the "consistent" if it contains no orphan message and all the in-transit messages are logged. To recover from a failure, the system restarts its execution from a previous consistent global state saved on the stable storage during fault-free execution. This saves all the computation done up to the last checkpoint state and only the computation done thereafter needs to be redone [7], [12], [13]. Synchronous and asynchronous are two fundamental approaches for checkpointing and recovery [2].

In uncoordinated or independent checkpointing, processes do not coordinate their checkpointing activity and each process records its local checkpoint independently [8], [14], [15]. After a failure, a consistent global checkpoint is established by tracking the dependencies. It may require cascaded rollbacks that may lead to the initial state due to domino-effect [11], [12], [13].

In coordinated of synchronous checkpointing, processes take checkpoints in such a manner that the resulting global state is consistent. Mostly it follows two-phase commit structure [9], [10], [11]. In the first phase, processes take tentative checkpoints and in the second phase, these are made permanent. The main advantage is that only one permanent checkpoint and at most one tentative checkpoint is required to be stored. In case of a fault, processes rollback to last checkpointed state. A permanent checkpoint can not be undone.

Coordinated checkpointing algorithms can be blocking and non blocking [3]. A primitive is blocking if control returns to the invoking process after the processing for the primitive completes but in case of non-blocking control return back to the invoking process immediately after invocation, even though the operation has not completed [1].

The objective of the present work is to design a checkpoint algorithm that is suitable for mobile computing environment. Mobile computing environment demands efficient use of the limited wireless bandwidth and the limited resources of mobile machines, such as battery power, memory etc. Therefore in the present work we emphasize on eliminating the overhead of taking temporary checkpoints. To summarize, we have proposed a hierarchical non-blocking checkpointing algorithm in which processes take permanent checkpoints directly without taking temporary checkpoints and whenever a process is busy, the process takes a checkpoint after completing the current procedure.

This paper organized as follows. In section 3 we state the system model considered in this work. In section 4 we have stated the algorithm. In section 5, we have the suitability of our proposed algorithm in the mobile computing environment. Finally section 6 shows the extension of the algorithms.

2. System Model

The system consists of collection of N processes, P_1, P_2….P_n, that are connected by channels. There is no globally shared memory and processes communicate solely by passing messages. There is no physical global clock in the system. Message send and receive is asynchronous.

3. Data Structure

Root is the initiator who starts a new consistent checkpoint by taking a tentative checkpoint. All child process take their checkpoint after receiving the checkpoint request (chk_req) message from their parent process, forward request message to its child node and increment to its checkpoint integer number (cin). Each process counts the number of messages it sent and received in the sr_counter (sent/received counter) variable. Every time a message is sent, the sr_counter is incremented. When a message is received, sr_counter is decremented. When a process receives an chk_tkn request, it adds the sr_counter value from that message to its own sr_counter. When it has received the chk_tkn reply from all its children, it sends the chk_tkn message to its parents. When the root process receives a chk_tkn reply from all its children, and its sr_counter is zero, root broadcasts a commit request (commit_req) message to its children.

When root process receives an update message, it increment in its sr_counter value till the sr_counter value not become zero. When a process receives a commit request it makes its tentative checkpoint permanent and discards the previous permanent checkpoint and propagates the message to its children and wait for the commit acknowledge.

4. Hierarchical Non-blocking Checkpoint Algorithms:

At any instant of time one process act as a checkpoint coordinator called the initiator or root process. Each process maintain one permanent checkpoint, belongs to the most recent consistent checkpoint. During each run of the protocol, each process takes a tentative checkpoint, which replaces the permanent one only if the protocol terminates successfully [6]. In this algorithm if any process is busy with other high priority job, it takes the checkpoint after the job ends. Otherwise it takes a checkpoint immediately. Each process stores one permanent checkpoint. In addition each process can have one tentative checkpoint, and are either discarded or made permanent after some time. Each process maintains a checkpoint integer number (cin), and it is incremented by one in every checkpoint session. Here we use the word checkpoint for tentative checkpoint.

Root process P_i:

There is only one checkpoint initiator or root process which initiates a checkpointing session.
When P_j receives a message from processes P_j, P_k…, P_i takes the tentative checkpoint. After that if it receives any other chk_req it will discard the request.

```
1. Check direct dependency node ddn$_i$ [] vector.
2. Sends chk_req message to its entire dependent or child processes.
3. Increment in cin$_i$ (cin$_i$ ++).
4. Every time a message is sent, the sr_counter is incremented. When a message is
   received, sr_counter is decremented.
5. while (sr_counter != 0)
     if receives a chk_tkn response including sr_counter value from all its children it adds the
     value of sr_counter in its own sr_counter value.
5. if sr_counter = 0
        Send commit_req to all processes to make tentative checkpoint
        permanent and wait for commit_ack.
```

For Any child processes P_j j! =i and $_1$<=j<= (n-$_1$)

On receipt of checkpoint request:

```
if P$_j$ receives a checkpoint request
    if P$_j$ has not already participated in checkpoint process
        Take a tentative checkpoint
        Do chkpt_process ()
    else
        If (received cin) > (current cin)     /*Compare both received cin and current cin.*/
           Take a new tentative checkpoint in place of old one.
           Do chkpt_process ();
        else
            Discard the chk_req and continue normal operation.
```

On receipt of piggyback application message:

```
If $P_j$ receives a piggyback application message
    If (received cin > (current cin)            /* Compare both received cin and current cin */
        Take tentative checkpoint before processing the message.
        Do chkpt
    else
        Ignore the request and continue normal operation.
```

Procedure chkpt_process ()

```
    If      $ddn_j$[] = = Null    /* for leaf node */
            Increment in $cin_j$
            Sends chk_tkn response including sr_counter value to its parent process.
    else
            /* If $ddn_j[] \neq$ Null */
            Check $ddn_j$[] vector.
            Send chk_req to its entire dependent or child node.
            Increment $cin_j$.
            sr_counter= own sr_counter value + received sr_counter value. /*When Receives
                        chk_tkn response including sr_counter value from its child node*/
               If receives any update message
                   Update sr_counter value and sends this updated message to its
                    Parent process $P_i$.
               If $P_j$ receives chk_tkn response from all its children processes
                   Send chk_tkn response including sr_counter to its parent process $P_i$ .
 End procedure
```

An example

The basic idea of the algorithm is illustrated by the example shown in figure 1. We assume that process P_1 initiates the algorithm. It is also called the root, coordinator or initiator process. First process P_1 takes the tentative checkpoint $Ck_{1,2}$. After that it check its direct dependency node ddn_1[] vector which is { P_1,P_2, P_3}. This means that process P_1 has receive at least one message from P_2, P_3, and P_4. After that P_1 send chk_req to P_2, P_3, P_4 and increment its checkpoint integer number cin 1 to 2 and work as usual. Each time it sends a message, it increase sr_counter and decrease when it receives the message. So in given example sr_counter= -3 which shows that it has received three messages. If sr_counter =0 it meant that it received chk_tkn message from all the processes. Then it sends the commit messages to all its coordinator to convert the tentative checkpoint in to permanent. When it receives the sr_counter from its dependent or child process, it adds this in to its own sr_counter. If it receives any updated message from coordinated or child process it will decrease the sr_counter value and continue this process until or unless the sr_counter ≠ 0. On receiving the chk_req from P_1, process P_2 first take tentative checkpoint $Ck_{2,2}$. After that it check its direct dependent node ddn_2[] vector which is null. It indicates that is a leaf node. So it will take tentative checkpoint and increment in its cin_2 from 1 to 2.

After receiving the chk_req from P_1 process P3 first takes a tentative checkpoint $Ck3_{,2}$ and check its direct dependency node ddn3[] vector which is {P_1, P_5}. Here we are assuming that message $M_{6,2}$ are the late message and process P3 does not receive this message till now. So first process P3 send chk_req message to P_1 and P_5 and after that it increase its checkpoint integer number cin3 from 1 to 2. Similarly process P_4 first take checkpoint $Ck_{4,2}$ and check its ddn_4[] which is {P_6} . Hence P_4 sends a chk_req message to P_6 and increment its cin_4 from 1 to 2. Same process is repeated by the processes P_1 and P_5.

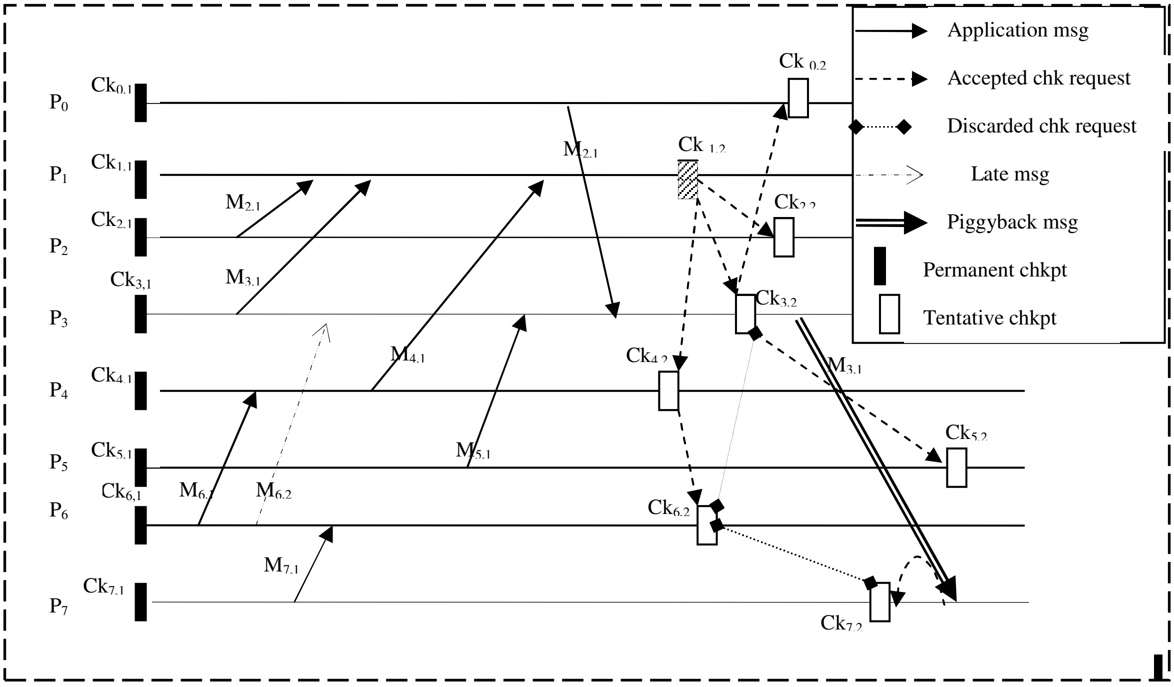


Figure 1: An example of checkpointing approach

Process P_6 receives chk_req from process P_4 first. So it will take checkpoint $Ck_{6,2}$. It is a non blocking checkpointing algorithms. Processes are not blocked after taking checkpoint and free to communicate to other process. Suppose that process P_3 sends an application message $M_{3,1}$ to process P_7. As we know that it is the first application message send by process P_3 after taking its checkpoint $Ck_{3,2}$. So process P_3 send piggyback application message to process P_7 which contain cin value with the message. Now process P_7 compare received cin with current cin which is 1. It finds that received cin 2 is grater than the current cin. So process P_7 takes the checkpoint $Ck_{7,2}$ before processing the message $M_{3,1}$. and increments its cin number from 1 to 2. After that process P_6 receives the message from process P_7. So process P_6 sends a chk_req to process P_7 and increments its cin_6 to 2. It is the second chk_req for process P_7 because it has already taken a checkpoint. In such case process P_7 first compare its current cin_7 with the received cin_6 which is 1. It finds that current cin is greater than the received cin. So it ignores the new checkpoint request.

A leaf process sends chk_tkn message including sr_counter to its parent process after that parent process adds sr_counter in its own sr_counter and when it receives chk_tkn message from all its children it sends to its parent process. This process will be continued until the root process does not receive all messages.

In figure 2 dependency tree sr_counter are shows in brackets. Firstly process P_2 sends its chk_tkn message and sr_counter which is 1 to the root process directly. So the sr_counter of root become -2. Now process P_1, P_5 sends the same to its parent process P_3 receives the same and adds the sr_counter of these processes in its own sr_counter. Now the sr_counter value of the P_3 become 1. As it receive the chk_tkn message and sr_counter value from all its dependent processes. So it sends the chk_tkn message including sr_counter to the initiator process P_1 and P_1 adds the sr_counter in its own sr _counter. Now the sr_counter of initiator process become -1. Then process P_7 sends chk_tkn message including sr_counter which is 1 to its parents process P_6 and after that sr_counter value of P_6 become 2 and then sends the chk_tkn message to process P_4 and after that sr_counter value of process P_4 become 2 and P_4 forward this to the initiator process. Now the sr_counter value of initiator process becomes 1. So root process

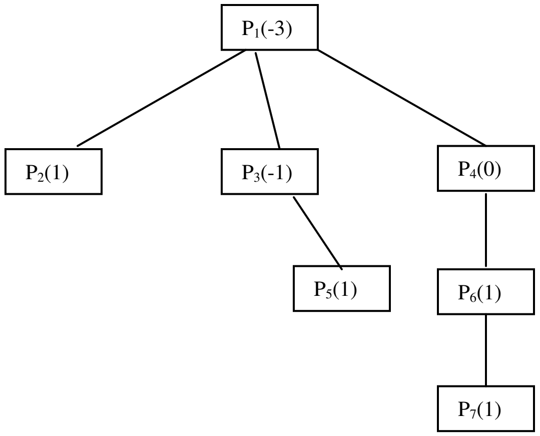


Figure2: Dependency Tree of all processes with sr_counter value before receiving late message send by the process P_6.

receives the chk_tkn message from all the process and its sr_counter value is 1. It shows the inconsistent global state and wait for update message and when the process P_3 receive a message sent by process P_6 its sr_counter value will become -2 and it will send this update message to the root process.

Root process receives the update message from the process P_3 and decrement its sr_counter by 1. So now the sr_counter value of root process become 0(zero) .Root process send the commit_req to the entire child node. When a process receives a commit_req message, it makes its tentative checkpoint permanent and discards the previous permanent checkpoint.
On the other side when the process P_6 receive the chk_req sent by process P_3 it compare its current checkpoint integer number cin_5 with the received checkpoint integer number cin_3. It finds that current cin_6 2 is greater than the received cin_3 which is 1. So it discards the request.

5. Suitability for Mobile Computing Environment

Consider a distributed mobile computing environment. In such an environment, only limited wireless bandwidth for communication among the computing processes. Besides, the mobile hosts have limited battery power and limited memory. Therefore, it is required that, any distributed application running. It is required that, any mobile distributed application running in such an environment must make efficient use of the limited wireless bandwidth, and mobile hosts' limited battery power and memory. Below we show that the proposed algorithm satisfies all the above three requirements.

a) This algorithm, processes neither take any useless and unnecessary checkpoints which help in better utilization of the mobile host limited memory.
b) This algorithm uses the minimum number of control messages. It definitely offers much better bandwidth utilization.

6. Extension of the Algorithms

The algorithms so far discussed, considers that there is only one checkpoint initiator. In case there are multiple concurrent initiators, each process has to handle multiple checkpoint sessions concurrently, and also maintain synchronization among them. A comparative study can also be done with other existing algorithms.

Reference:

[1] Kshemkalyanl Ajay D, Singhal, M.: Distributed Computing Principals, Algorithms, and System

[2] Singhal, M. , Shivaratri, N.-G.: Advanced Concept in Operating System. McGraw Hill,(1994)

[3] Cao, G. and, Singhal, M "Mutable checkpoints: a new checkpointing approach for mobile computing systems,"IEEE Transactions on Parallel and Distributed Systems, vol. 12, Issue 2,pp. 157-172, Feb 2001.

[4] Coulouris, G., Dollimore, J., Kindberg, T., Distributed System Concepts and Design, 3rd edition, Addison- Weslely,(2001), 772p

[5] Elnozahy,E.N, Johnson, D.B. and Zwaenepoel, W. "The Performance of Consistent" Proceedings of 11th Symp. On Reliable Distributed Systems, pp. 86-95, October 1992, Houston.

[6] Koo R. and Toueg. S, "Checkpointing and Rollback-Recovery for distributed System," IEEE Trans. Software Eng., SE-13(1):23-31, January 1987.

[7] Ziv Avi and Bruck Jehoshua "Checkpointing in Parallel and Distributed Systems", Book Chapter from Parallel and Distributed Computing Handbook edited by Albert Z. H. Zomaya, pp. 274-320, Mc Graw Hill, 1996.

[8] Bhargava B. and Lian S.R., "Independent Checkpointing and Concurrent Rollback for Recovery in Distributed System -An Optimistic Approach," Proceeding of 17th IEEE Symposium on Reliable Distributed System, p. 3-12, 1988.

[9] Chandy K.M. and Lamport L., "Distributed Snapshots: Determining Global State of Distirbuted Systems,"ACM Transaction on Computing Systems, vol. 3 No. 1, pp. 63-75, Feb. 1985.

[10] Elnozahy E.N., Alvisi L., wang Y.M. and Johnson D.B., "The Performance of Consistent Checkpointing," Proceedings of the 11th Symposium on Reliable Distributed Systems, pp. 39-47, October 1992.

[11] Koo R. and Toueg S., "Checkpointing and Roll-Back Recovery for Distributed System," IEEE Trans.on Software Engineering, vol. 13, no. 1, pp. 23-31, January 1987.

[12] Randall, B, " System structure for Software Fault Tolerance", IEEE Trans.on Software Engineering, Vol.1,No.2,pp220-232, 1975.

[13] Russell, D.L., "State Restoration in System of Communication Processes", IEEE Trans. Software Engineering, Vol.6,No.2pp 183-194, 1992.

[14] Sistla,A.P. and Welch,J.L., "Optimistic Recovery in Distributed Systems", ACM Trans. Computer System, Aug, 1985, pp. 204-226.

[15] Wood, W.G., " A Decentralized recovery Control Protocol", IEEE Symposium on Fault Tolerant Computing. 1981.

[16] Gupta Bidyut .el "A low-Overhead Non block Checkpointing Algorithm for Mobile Computing Environment" springer-Verlag Berlin Heidelberg 2006 pp. 597-608.

Implementation of Agent based Dynamic Distributed Service

Prof. I.Ravi Prakash Reddy irpreddy@gnits.ac.in
IT Dept.
G.Narayanamma Institute of Tech & Science
Hyderabad-500008

Dr. A.Damodaram adamodaram@jntu.ac.in
Prof., Dept. of CSE
JNTU College of Engg.Hyderabad

Abstract

The concept of distributed computing implies a network / internet-work of independent nodes which are logically configured in such a manner as to be seen as one machine by an application. They have been implemented in many varying forms and configurations, for the optimal processing of data.

Agents and multi-agent systems are useful in modeling complex distributed processes. They focus on support for (the development of) large-scale, secure, and heterogeneous distributed systems. They are expected to abstract both hardware and software vis-à-vis distributed systems.

For optimizing the use of the tremendous increase in processing power, bandwidth, and memory that technology is placing in the hands of the designer, an agent based Dynamically Distributed Service (DDS, to be positioned as a service to a network / internet-work) is proposed. The service will conceptually migrate an application on to different nodes**.** In this paper, we present the design and implementation of an inter-mobility (migration) mechanism for agents. This migration is based on FIPA[1] ACL messages. We also evaluate the performance of this implementation by using a Distributed framework.

Keywords: Distributed Systems, Agents, Agent Migration

1. INTRODUCTION

Over the last two decades, the concept of distributed computing has been implemented in varying configurations and on diverse platforms. In current context, a distributed system implies a networked system of nodes in which a single application is able to execute transparently (and concurrently) on data that is, (or may be) spread across heterogeneous (hardware & operating system) platforms. The salient payoffs of distributed computing may be listed as:

- Enhanced performance (in respect of both speed up and scale up).
- Resource sharing (in respect of data as well hardware and software processing elements).
- Fault tolerance and enhanced reliability.
- Serve as the basis for grid computing.

Several other relevant issues while assessing the relevance of distributed computing vis-à-vis the current computing environment and this paper are:

- Interoperability in a heterogeneous environment will continue to be the dominating theme in future applications.
- Communication technology advances will continue to fuel the need for more bandwidth and enhanced Quality of Service specifications.
- The rate of increase in data processing and storage is greater than that of data transfer.
- Most users are reluctant to switch platforms, as are developers to switch technology paradigms.
- The individual behavior of the vast number of interconnected nodes based on individual workstations mandate that any service acting upon them universally must be dynamic in nature.
- In many computer installations /complexes, a lot of state of the art hardware is not used round the clock. There are times when it is idle, or is under-utilized. When this happens, it may be used by other applications for processing purposes remotely. Networking enables this. Inter-networking further emphasizes the same.

In view of the above, it is forecast that distributed processing of applications and data will no longer be restricted to high end research and scientific applications, but will become as normal as any other service provided over an inter-network. The Internet and the Web themselves are a viable distributed computing domains. Distributed computing however, has yet to gain the type of proliferation mandated by enhanced rates of data processing as well as transfer.

To further the optimization of internet-works (including networks), by the use of distributed computing concepts, a Dynamically Distributed Service, analogous to e-mail, FTP, Voice over IP, etc, is proposed, which can be made available on demand, in an intranet/inter-network. The service will conceptually migrate an application on to different nodes. In this paper, we have presented a proposal for the mobility of agents between various agencies, based on the agent communication language (ACL) proposed by FIPA. This Dynamically Distributed Service (DDS) is at variance with distributed paradigms used till date, though *no changes to the underlying hardware or OS are proposed* in its implementation.

The efficient utilization of network resources is an important issue. The problem is hard due to the distributed nature of computer networks, high communication demands and the desire for limited communication overheads. One solution to such challenges is to design efficient, decentralized and fault-tolerant data propagation model which accomplishes tasks with no access to global network information. Mechanisms of agent propagation are useful because agents can be organized into efficient configurations without imposing external centralized controls. Operation without a central coordinator eliminates possible bottlenecks in terms of scalability and reliability.
In the section 5,we evaluate the performance of DDS by applying it to distributed calculation of Prime numbers.

2. DISTRIBUTED AGENTS

The research areas of multi-agent systems and distributed systems coincide, and form the research area of *distributed agent computing*. Multi-agent systems are often distributed systems, and distributed systems are platforms to support multi-agent systems[2].

Agents are considered to be autonomous (i.e., independent, not-controllable), reactive (i.e., responding to events), pro-active (i.e., initiating actions of their own volition), and social (i.e., communicative). Sometimes a stronger notion is added (beliefs, desired, intentions) realizing intention notions for agents. Agents vary in their abilities; e.g. they can be static or mobile, or may or may not be intelligent. Each agent may have its own task and/or role. Agents, and multi-agent systems are used as a metaphor to model complex distributed processes.

Both distributed systems and agents share the notion of 'distributedness'. The area of multi-agent systems addresses distributed tasks; distributed systems addresses supporting distributed information and processes.

The area of *distributed agent computing* is the area in which both approaches intersect. Both can be synergized to further optimality. Mobile agents transfer code, data, and especially authority to act on the owner’s behalf on a wide scale, such as within the entire Internet. Because of this advantage, we have decided to use mobile agents for migration.

3. PROPOSED MODEL

The platform chosen for implementing migration was JADE[3], because it is a widely adopted platform within the software agent development and research communities. It is open source and complies with FIPA specifications.

3.1 The JADE platform

The JADE platform is divided into a large number of functional modules, which can be placed into three categories in general terms:

Core. The core of the platform is formed by all components providing the necessary execution environment for agents' functioning.

Ontologies and Content Languages administration. This consists of the agency's mechanisms for carrying out information processing in ACL messages, and the internal structures that the agency and agents will use to represent this content.

Message transport mechanisms. Mechanisms and protocols used to send and receive messages at both intra-agency and inter-agency level.

At the core of the JADE platform is the concept of the container, which is the minimum execution environment necessary for an agent to operate. Each container in JADE is executed in a different Java virtual machine, but they are all interconnected by RMI (Remote Method Invocation).

Containers do not only enable groups of agents to be separated into different execution groups, but agencies may also be distributed in various machines so that each has one or several of them. One of the different existing containers is the principal, which represents the agent itself and which gives orders to all the others. JADE also provides mobility between containers. For this reason, if the agency is distributed in various machines, agents can move between them. However, accepting this type of mobility as migration could be considered a mistake." Satellite" containers are highly dependent on the principal and many operations carried out by the agents within them end up passing through the central node. Furthermore, the connections between them (carried out by RMI) must be permanent, as if not, many errors due to the loss of link may be generated. As we can see, using this type of mobility as a typical migration ends up making mobile agent systems' scalability disappear because a certain type of operations is centralized in a single node. However, it may be very useful to use the diagram of containers to distribute the processing of agencies that have to bear a heavy load or to isolate some agency types within a single agency for security reasons.

These details lead to the necessity for inter-agency migration, which is carried out through a non-permanent channel and makes a system of mobile agents available that is much more scalable, and in which agencies are totally independent units. This independence is not only desirable from the point of view of fault tolerance, but also because of privacy.

3.2 Our proposal for migration using ACL

The idea of creating a migration using ACL messages came from FIPA's specification regarding mobility, where this type of migration is proposed. However, as mentioned above, this specification only gives a general outline of the ontologies, the protocol, and the life cycle of a mobile agent. It has not been updated due to the lack of implementations and has become obsolete within the FIPA specifications. For these reasons, we have found that there is a need to propose extensions to the specification to cover situations that it does not deal with.

The design for a migration using ACL means that transmission of mobile agents between two agencies will be carried out using the message system between agents. In other words, the agent (both the code forming it as well as the state that it is in) will travel as the content of a message between two agents. Specifically, the agent will travel in the message between the AMS agents of each of the agencies involved in a migration.

Because the agencies have mechanisms for sending and receiving messages, using a parallel transmission protocol is not necessary. This is an advantage in interoperability terms and enables agents to be transmitted using the various message transport protocols (HTTP, HOP, SMTP, etc). Furthermore, this is achieved in a totally transparent way.

The first logical step in this process is to design the ontology and the protocol that will be used in the exchange of messages between the two agencies. This protocol has the movement of the agent as its final purpose. Defining an ontology basically consists of defining the elements that will form the content of an ACL message to give a common interface between the two parties when extracting the information of the message.

The two possible migration models that are proposed in the initial FIPA specification deal with one migration directed by the agent and another directed by the agency. In our implementation, we have decided to adopt the migration directed by the agency, which is more robust, as it enables the agency to decide which migration requests are accepted and which are not.

The ontology initially specified by FIPA is made up of seven elements: five concepts ("mobile-agent-description", "mobile-agent-profile", "mobile-agent-system", "mobile-agent-language", and "mobile-agent-os") and two actions ("Move" and "Transfer").

Of these concepts, we only use the first one, "mobile-agent-description". This is because it is very difficult to develop systems that enable agents with different architectures to migrate with total interoperability, at the current level of agent technological maturity. These agents could have been written in different languages or executed in different operating systems. For this reason, these concepts are never used, assuming that mobile agents which migrate move between the same agencies. Obviously, if the agency has been developed in a language like Java, a migration between agencies lodged in different operating systems is possible. However, this is a characteristic of this language which is transparent to the agency, and therefore does not involve the need to use the concept "mobile-agent-os", for example. In any case, although we do not use it, we maintain these concepts to ensure compatibility in case it is possible to make agents migrate between agencies implemented in a different way or with different languages.

The concept "mobile-agent-description", on the other hand, is highly useful to us. Within it are several fields that define the characteristics of the mobile agent in question. Among others, these include the characterization of the code and its data.

Of the two actions specified, we have only implemented "Move", for migrations directed by the agency. We do not take the "Transfer" action into consideration, which can be used for migrations directed by the agent, although it is supervised by the agency.

Once the content of the ACL messages was described, we moved on to enumerating the protocol by which the migration process is carried out. A diagram of the messages exchanged during the migration process can be seen in Figure1.

- Firstly, the agent wishing to migrate starts a conversation with the AMS agent of the local platform to make a request for migration. This request is the first step in the standard FIPA-Request protocol and consists of a Request-type message with a Move action and a *MobileAgentDescription (MobileAgentDescription* is the name of the class that implements the concept of "mobile-agent-description"), in which the code and data fields are empty.
- When the AMS agent receives the request for migration from a mobile agent, the first thing it does is to decide whether to accept it or not according to a given criterion. If the migration is accepted, the AMS agent sends *din Agree* message to the agent, or if not, a *Refuse* message.
- If the migration is accepted, the first thing that the AMS agent does is to obtain the code and data (serialized instance) of the agent that made the request and fills in the code and data fields of'the*MobileAgentDescription.*
- The next step is to make contact with the AMS belonging to the agency to which the mobile agent wishes to migrate. To this end, the local AMS must start a parallel conversation to that between the agent and the remote AMS.
- When the remote AMS receives the request, the agent's code and data travel within the *MobileAgentDescription* that the local AMS has prepared.
- Following its own criteria, the remote platform decides whether to accept the in coming agent. If so, it responds with a *Agree* message, and if the agent does not meet the requirements specified by the agency to execute it, it responds with a *Refuse* message.
- The remote AMS loads the agent's class, deserializes its instance and restores its execution. Once this entire process has been successfully completed, the standard FIPA-Request protocol is completed by sending an *Inform* message to the local AMS.
- The final step in the protocol consists of informing the agent that started the process. If the process has been successfully completed, the original agent is destroyed.

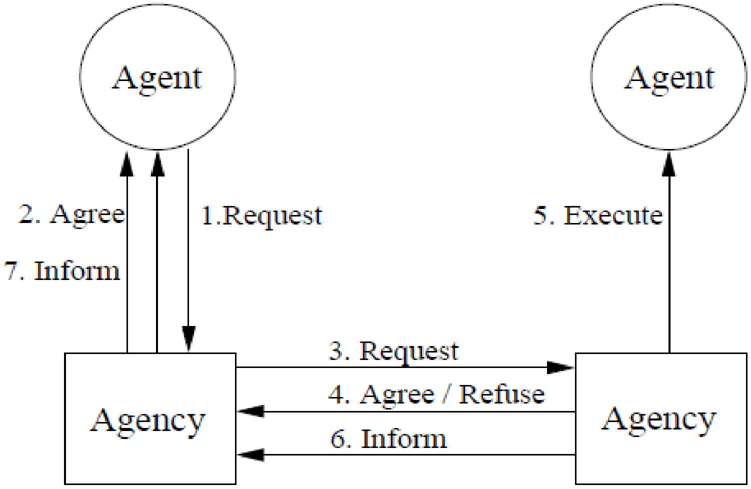


FIGURE 1: Exchange of ACL messages in the migration protocol

4. DESIGN COMPONENTS

After describing the ontology and the protocol used, the components necessary to carry out the entire process are listed below.

Migration Manager: The Migration Manager component is responsible for carrying out all operations enabling the execution of an agent to be suspended in the local agency. Also, it enables its state and code to be processed. These operation allow to insert the agent in an ACL message. In the remote agency, it is responsible for undoing the messages that contain agents, decoding them and acting together with the agency to restore the agent's execution.

Class Loader: The class loader is the component that constructs the representation of the class in the memory so that it can be used in the code. The bytecode of the class is extracted from the ACL message and loaded during the deserialization time of an incoming agent. At the same time, each classloader provides a space of different names from the others, so that two agents created with one class with the same name do not conflict within a single platform if they are loaded by different classloaders.

Mobile Agent: We have created the *MobileAgent* class, inheriting the properties of a basic agent and adding the functionality of being able to migrate to it. This functionality provides it with the *doMigrate(dest)* method, which starts the migration protocol when invoked.

Conversation Modules: These modules were implemented using the behaviours of the JADE model [4]. Behaviors represent agent tasks and are also a useful tool for designing the protocols that govern conversations using ACL. There are basically two components of this type developed to provide mobility. The first is the agent's behavior. This component is launched when the function *doMigrate(dest)* is invoked and is responsible for supervising the migration from the agent's point of view. The second was implemented within the AMS agent to help it administer its part of the migration protocol. This behavior has a double functionality as it was designed for playing the roles of the local AMS and the remote AMS. The most complex part of the implementation of this component is its functioning as a local AMS: parallel conversations (AMS-Agent and AMS-AMS) which depend on each other must be taken into consideration.

5. PERFORMANCE EVALUATION

Providing agents with abilities to migrate continues to be a highly challenging problem[5]. We conduct a set of experiments in two environments: 4 heterogeneous computers and 45 (almost) homogeneous computers. Specifically, we are looking for a way to find optimal configurations for migration in both environments. The conclusion from this work is that introducing propagation to a system in a form of agent migration in both networks could considerably decrease the execution time according to used algorithms and established assumptions.

5.1 The Problem

The prime number generation problem was selected for testing the DDS. We assume that we generate a given number of primes above a given start number and write the results into the file. All computers search prime numbers and as soon as possible they send the solution to the one, which is responsible for saving the results in a file. Prime number generator was selected to be solved in a distributed way because of its computation-focused nature. A range of numbers to examine is given and from the result point of view, it does not matter, on which machine the computation is running. The most important aspect is that there is a possibility to move agents from the slow computers to the faster ones.

The algorithm for prime number is simple. For each x from given range [a,b] calculate square root c = sqrt(x) and check if any number from range [2,c] divides x without reminder; if no then add x to the list of prime numbers. Of course, there are more efficient ways to calculate prime numbers. The goal is not to calculate them as fast as possible, but to show how distribution could be managed using agent migration.

This approach has several advantages. One of them is that the range to search the primes can be divided, and the results can be merged at the end giving the complete solution to the problem.

In order not to create a bottle-neck, partial results are being sent continuously to be saved into the file.

5.2 DDS Framework Architecture

In this architecture, in general the DDS framework consists of two kinds of units: nodes and a broker. A node is a component, on which agents reside, which is supposed to search prime numbers. A broker is a component, which distributes the task into the framework and saves the results into the file. It is not destined to calculate primes.

We assume the following algorithm for distributing the task over the nodes. When the broker gets the task order it knows how many nodes are available, because when a node enters the network, it sends a ready message to the broker. We do not consider any changes after the broker has received the task order. At the beginning there is no information about the available resources, so primes search range is divided equally by the number of nodes and then sent as a task request. While processing, nodes also send found primes to the broker by portions.

5.3 Main Process

The main process is the core of the framework[6]. Its main goal is to distribute task and afterwards collect all messages concerning reporting in order to display the final results. All activities used in the diagram are agent services.

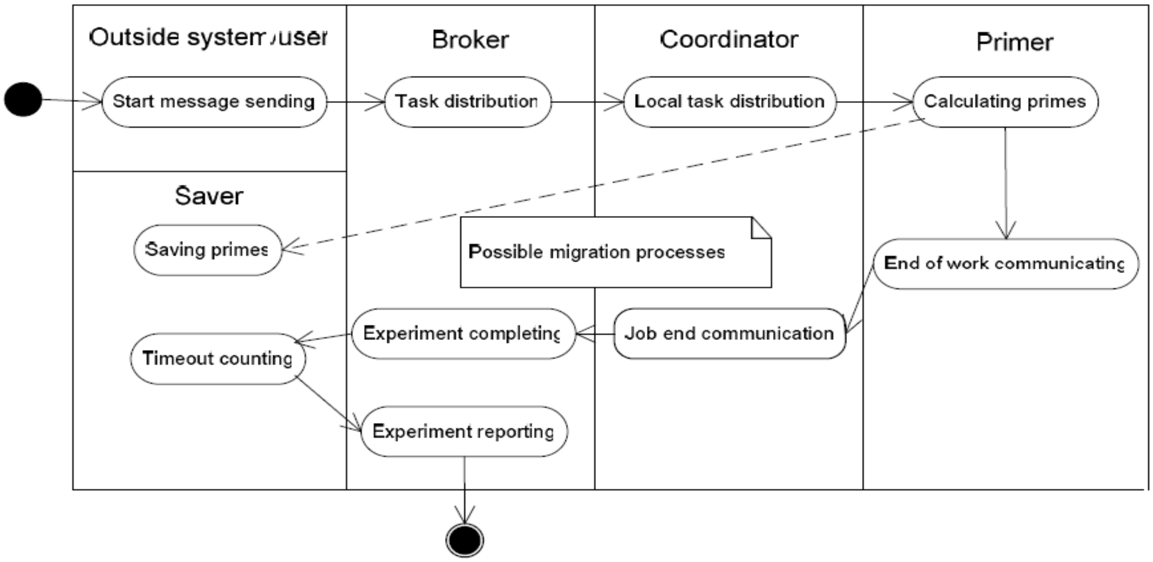


FIGURE 2: Main process diagram

There are 4 types of agents participating in the main process: Broker, Saver, Coordinator and Primer. This process, depicted on Figure 2, starts outside of the system, when a user or agent from another system sends a message to the Broker that includes the range of numbers to search primes from. The next step is done by the Broker, which distributes task to Coordinators. The diagram shows the control flow only for one Coordinator and one Primer, with actually many Coordinators and many Primers.

Task distribution from the Broker goes to the local level and finally each Primer gets its task. Searching primes consists of many single search processes after which results are sent to the Saver agent. This is shown in figure by a dashed arrow. The box "Possible migration processes" denotes the place in the main process flow when migration process is possible. They take place after the initial task distribution and before end of work.

The last phase of the main process starts when Primers finish searching the whole range of numbers they got. Each of them sends a message to the Coordinator (a Coordinator "commanding" the location where Primer resides) and when all Primers report back to Coordinators, agents send job end message to the Broker. When Broker gets all job end messages, it expects the main process to be nearing completion. Because there are possible message asynchronisms, Saver agent starts a timeout counting in order to receive all messages from Primers, whose results have not been written into the file so far. This possibility exists mainly because each result message includes large data to write. When timeout is over, Saver agent confirms that there are most probably no other result messages (also called save messages). When Broker gets this message, it displays information about the experiment. This ends the main process.

5.4 Migration Calculation

This process assesses the ability of a node to perform a task. The key point here is that the estimation is based on the previous efficiency of a node when performing a task. It is compliant to the assumption that at the beginning of an experiment the ability of nodes to perform the task is unknown.

When a Coordinator gets all progress reports it estimates remaining time till the end of experiment based on the data it has. This is an important moment in the whole algorithm. Time till the end is estimated based on a sample from the previous change in the environment for the node - from the last migration or the beginning of the experiment. Basically, when Primers are reporting, they deliver two parameters - what is the range of numbers to examine and how many numbers have already been checked. All Primers report that there are two sums being calculated: a sum of numbers to check and a sum of numbers that already have been examined. Based on the new and old report there is a quantity of numbers calculated that have been examined. Then, the time from the last change is calculated. Based on these two values the speed for node i (1) is calculated.

After the Migration Coordinator gets all reports form Coordinators it starts the calculation. During that time all Coordinators wait for messages. When the agent network is being created at the beginning of an experiment, Migration Coordinator creates a list of objects describing nodes. This list is used for migration calculation but also for remembering names, locations in the network and for information if a node has to report back after migration finishing. As the reports arrive the information in the list is being refreshed.

$$speed_i = \frac{currently\_checked\_numbers_i - last\_checked\_numbers_i}{current\_time_i - last\_change\_time_i} \tag{1}$$

$$estimated\_time_i = \frac{number\_of\_numbers\_to\_check_i}{speed_i} \tag{2}$$

$$agent\_value_i = \frac{estimated\_time_i}{number\_of\_agents_i} \tag{3}$$

$$agent\_change_i = \frac{average\_active\_time - estimated\_time_i}{agent\_value_i} \tag{4}$$

$$proposed\_agents_i = number\_of\_agents_i + agent\_change_i \tag{5}$$

$$norm\_proposed\_agents_i = proposed\_agents_i * \frac{current\_agents\_sum}{sum\_of\_proposed\_agents} \tag{6}$$

For each node we have data on the estimated time (2) and the list is sorted beginning with the shortest time till the end of experiment. Then for each node three values are calculated. The agent value for node i (3) is a measure of how much time from the estimated time till the end falls to one agent. The next value is a bit more complicated (4). In this algorithm there is such a value as an average active time. The term active time applies to those nodes only, where migration can take place or in other words, which have the estimated time higher than the node threshold parameter . So the goal is to calculate how many agents on node i should have the time as close to average as possible. The assumption is that an agent (Primer) represents some work to do and if there was a certain number of agents, then the time would be close to average. Having such simple assumption, the number of proposed agents for each node is calculated (5). If for example a node has a time lower than average - then there should be more agents and the agent value change is greater than zero. If not then some agents should migrate from this node. But after calculating the proposed agent number there is a possibility that there should be more or less agents than currently is, so it is necessary to correct this number on each node by sum of agents divided by sum of proposed agents (6).

After this process agents are distributed according the resources (agents) available in the system. But there is a possibility that still the sums of agents and proposed agents are not equal, so there is correction algorithm, that makes these sums equal by adding or subtracting proposed agents for each nodes starting from those that have the biggest number of proposed agents.After executing this algorithm a list of proposed migrations is created. Building this list is based on making equal the number of agents and proposed agents on a node (in the node information list) possessed by Migration Coordinator. Agents always migrate from the node that has the biggest estimated time to the node that has the lowest estimated time.

5.5 Performance Measurements

The experiment is conducted in order to test the implemented system using big number of computers. Because the College environment is homogeneous, there was a heterogeneity introduced by running more instances on one computer. Tests were conducted on 15 computers in which on five of them there was one JADE container running. On the next five there were two containers running and on the rest 3 containers. The number of containers equals 30. In this experiment the main goal is to find the optimal parameters for the described configuration. Without migration the experiment took 1000.5 seconds and the first computer reported after 317.6 seconds.

5.6 Discussion of Results

The goal is to find optimal configuration for two different environments: home and university. In order to do this there has been a set of experiments conducted. Other main purpose is to show, that migration helps to improve efficiency of task completion when a network is composed of computers, which are not homogeneous - they have different configuration or/and different computing power at the moment (they can be busy because of other programs running

We spent countless hours making multiple runs in order to assure ourselves that the results were due to inherent randomness and not model errors. Table 1 provides the best values from these runs. Note that the agent migration increases the efficiency in task execution.

Parameter	**Heterogeneous computers**	**Homogeneous computers**
Profit in execution time	58%	35.4%
Primes range search unit	150	150-200
Report time	20s	30s
Agents on a node	5-20	10-15

TABLE 1: Optimal configurations for heterogeneous and homogeneous computers

Moreover, we can combine the results of all of above cases and draw the following conclusions. The more agents in the system, the more dynamic the environment is and also the more migrations take place. The most important parameter in the system is primes range search unit, because it is able to balance the calculations efficiency and the communication velocity. The optimal configuration is when all computers finish their tasks in time close to each other. The closer the migration phases are in time, the more migrations there are in the system. To make the system more stable migration phases have to be distant in time. This also impacts the number of messages in the system - the more stable the system is the lower communication costs. The lower number of migrations the shorter the execution time assuming there is enough agents in the system, that are able to cover differences in computer efficiency (the optimal number of agents seems to be around 10 or 15).

When there are migrations in the system, there is a possibility of cycle migration phenomenon. This has a negative impact on nodes efficiency and it was proven using full experiments report. The cause of this lies in the accuracy of time till end estimation for a node and the number of agents. There are two ways of limiting it: low number of agents or limited migration, but there is always a danger that after a sudden event in the system (like one node efficiency collapse) it could not handle changes in a short time (slightly higher execution time).

When the number of agents in the system is small, a limited migration could function more efficiently. The overall conclusion here is that within a dynamic environment the key is to find a balance between covering differences in computer efficiency and unexpected events (the more agents, the more accurate it is) and limiting migrations and migration cycles (the less agents the better).

6. CONCLUSION & FUTURE WORK

The performance results show that a lot of work must be done in the transport area, defining fast Message Transport Protocols, and using lightweight content languages in order to make ACL based migration more competitive in terms of performance.

The upgrades to DDS framework could be connected with two key points: algorithm for migration and time estimation. More advanced algorithm for migration calculation could be more concentrated on avoiding migration cycles (something like limited migration) with parameters regarding how the node was handling the task previously. The time till end estimation could be more influenced by historical efficiency based on assumption that it does not change so often

The DSS proposed has attempted to introduce dynamically distributed service as inherent to an inter-network as FTP, TELNET, e-mail, chat etc. The rationale for - and the main features of the basic scheme have been described. Mechanisms of DDS are useful because agents can be organized into efficient configurations without imposing external centralized controls. Operation without a central coordinator eliminates possible bottlenecks in terms of scalability and reliability. Process intensive applications will be the main beneficiaries of the scheme.

7. REFERENCES

[1] FIPA. Foundation for Intelligent Physical Agents, http://www.fipa.org

[2] AgentCities.NET. European Commission funded 5th Framework IST project. November 2001. http://www.agentcities.net

[3] JADE, Java Agent DEvelopment Framework, http://jade.tilab.com

[4] F.Bellifemine, G.Caire, T.Trucco, G.Rimassa. JADE Programmers Guide, July 2002.

[5] Hmida, F.B., Chaari, W.L., Tagina, M.: Performance Evaluation of Multiagent Systems: Communication Criterion, In Proc. KES-AMSTA 2008, LNAI 4953, 2008, 773-782.

[6] Bernon, C., Chevrier, V., Hilaire, V., Marrow, P.: Applications of Self-Organising Multi-Agent Systems: An Initial Framework for Comparison, Informatica,30,2006, 73-82.

A Havoc Proof For Secure And Robust Audio Watermarking

K.Vaitheki email: vaidehi.balaji@gmail.com
Assistant professor, Department of Computer Science
Pondicherry University
Puducherry, India - 605014

R.Tamijetchelvy email:narendra_naren_lucky@yahoo.co.in
Assistant professor, Department of Electronics and communication
Perunthalaivar Kamarajar institute of Technology,Karaikal
Puducherry, India – 605107

Abstract

The audio watermarking involves the concealment of data within a discrete audio file. Audio watermarking technology affords an opportunity to generate copies of a recording which are perceived by listeners as identical to the original but which may differ from one another on the basis of the embedded information. A highly confidential audio watermarking scheme using multiple scrambling is presented Superior to other audio watermarking techniques; the proposed scheme is self-secured by integrating multiple scrambling operations into the embedding stage. To ensure that unauthorized detection without correct secret keys is nearly impossible, the watermark is encrypted by a coded-image; certain frames are randomly selected from the total frames of the audio signal for embedding and their order of coding is further randomized. Adaptive synchronization is improves the robustness against hazardous synchronization attacks, such as random samples cropping/inserting and pitch-invariant time stretching. The efficient watermarking schemes make it impossible to be detected and robust even though the watermarking algorithm is open to the public.

Keywords: Audio watermarking, Information hiding, Copyright protection, Multiple Scrambling.

1. INTRODUCTION

The digital media have opened the door to an information marketplace where the true value of the product (digital content) is dissociated from any particular physical medium. It also enables a greater degree of flexibility in its distribution and a lower cost, the commerce of disembodied information raises serious copyright issues. Indeed, digital data can be duplicated and re-distributed at virtually no cost, potentially turning piracy into a simple "click and drag" process. Cryptography has been clearly established as a technology of fundamental importance for securing digital transfers of data over unsecured channels. By providing encryption and authentication of digital data, cryptography enables trustworthy point-to-point information exchange and transactions to be achieved. Hence, once the recipient validates and decrypts the data, the product can be subsequently stripped from any content identification, proof-of-ownership or other descriptive information and any further duplication and re-distribution can leave the rights holders powerless and royalty-less. While such re-distributions may not represent a serious threat when the content consists of proprietary information that has a short life span, such piracy could have catastrophic implications for the entertainment industry, whose content has a very long life span.

Cryptography provides an easy way to see how digital sub-codes and other proprietary digital formats can fail in similar ways, since they are only volatile representations of a medium. The true value of the product (the content) can still be transferred effortlessly onto different formats and media. Any attempt to secure the identities of content's rights holder's calls for a technology that enables some secure auxiliary information, or watermark, to travel with the content through any channel, format or medium where the content's value remains. A properly designed audio watermarking technology provides the means to do this in the context of audio content, while preserving the integrity of the original recording. Unlike sub-codes, encryption or audio compression, a watermark should not rely on any specific format. In order to travel along with the content it protects, the watermark must be carried by the content itself. Embedding a watermark is an active modification of the audio waveform. Subsequent to this process, the watermarked content becomes a message carrier regardless of the format of medium it lives on.

2. SCOPE OF AUDIO WATERMARKING

Digital watermarking is techniques of embedding information into a signal. The host signal that carries the watermark is also called a cover signal. When the cover signal is an audio signal, the embedding technique is called audio watermarking. The purposes, types and requirements of audio watermarking are presented after the research.

Purposes

There are various purposes for audio watermarking. The original intention of watermarking is for copyright protection. he most obvious purposes are the needs for proof of ownership and the enforcement of usage policy. In addition, watermarking can also be used for fingerprinting and additional features to a media

Proof of Ownership

A watermark can represent a code of information to identify the owner of a digital signal. This application is similar to the function of international standard book number (ISBN) for book identification. The watermark must be correctly presented to proof an ownership in a court of law.

Enforcement of Usage Policy

Watermark can be used to provide copyright information to consumer devices. The usage of audio information will be limited or stopped by the devices if certain requirement is not fulfilled by the user. However, this function of watermark has posted a difficulty in actual application. This is because in order for a consumer device to recognize a watermark, the watermark or the secret key for watermark generation has to be kept by the device. Attackers can use reverse engineering to obtain the watermark or disable the watermark verifying function in a device.

Fingerprinting

The usage of an audio file can be recorded by a fingerprinting system. When a file is accessed by a user, a watermark, or called fingerprint in this case, is embedded into the file. The usage history can be traced by extracting all the watermarks that were embedded into the file.

Additional Features

A watermark can also provide additional information to a file. For instance, the lyrics can be embedded into a song and extracted when it is played. Furthermore, the watermark can be a special label for convenient search function in databases.

3. REQUIREMENTS OF A WATERMARK

For a scheme to fulfill the purposes of watermark, a number of requirements have to be satisfied. The most significant requirements are perceptibility, reliability, capacity and speed performance.

Perceptibility

The most important requirement is that the quality of the original signal has to be retained after the introduction of watermark. A watermark cannot be detected by listeners.

Reliability

Reliability involves the robustness and detection rate of the watermark. A watermark has to be robust against intentional and unintentional attacks. The detection rate of watermark should be perfect whether the watermarked signal has been attack or not. Otherwise, the watermark extracted is not useful for proof of ownership. Secure digital music initiative (SDMI), an online forum for digital music copyright protection, has summarized a list of possible attacks to evaluate the robustness of watermarking schemes. These attacks include digital-to-analog, analog-to-digital conversions, noise addition, band-pass filtering, time-scale modification, addition echo and sample rate conversion. If the quality of the watermarked signal after the attacks is not significantly distorted, the watermark should not be removed by these attacks.

Capacity

The amount of information that can be embedded into a signal is also an important issue. A user has to be able to change the amount embedded to suit different applications. An example can be seen in real-time application. If a watermark is spread across an audio signal, the complete signal has to be presented first. This is not possible in streaming over the Internet.

Speed

Watermarking may be used in real-time applications, such as audio streaming mentioned before. The watermark embedding and extracting processes have to be fast enough to suit these applications.

4. TYPES OF AUDIO WATERMARKS

Audio watermarks are special signals embedded into digital audio. These signals are extracted by detection mechanisms and decoded. Audio watermarking schemes rely on the imperfection of the human auditory system. However, human ear is much more sensitive than other sensory motors. Thus, good audio watermarking schemes are difficult to design. Even though the current watermarking techniques are far from perfect, during the last decade audio watermarking schemes have been applied widely. These schemes are sophisticated very much in terms of robustness and imperceptibility. Robustness and imperceptibility are important requirements of watermarking. There are two types of audio watermarks, Non-blind watermarking and blind watermarking

Non- blind watermarking

Non-blind watermarking schemes are theoretically interesting while they are conflicting each other. It requires double storage capacity and double communication bandwidth for watermark detection. These non-blind schemes may be useful as copyright verification mechanism in a copyright dispute.

Blind watermarking

The blind watermarking scheme can detect and extract watermarks without use of the unwatermarked audio. Hence it requires only a half storage capacity and half bandwidth compared with the non-blind watermarking scheme. Blind audio watermarking schemes are mostly used in practice. The blind watermarking methods need self detection mechanisms for detecting watermarks without unwatermarked audio.

5. REQUIREMENTS FOR AUDIO WATERMARKING ALGORITHMS

The relative importance of a particular property is application dependent, and in many cases the interpretation of a watermark property itself varies with the application.

Perceptual Transparency

The watermark-embedding algorithm has to insert additional data without affecting the perceptual quality of the audio host signal. The fidelity of the watermarking algorithm is usually defined as a perceptual similarity between the original and watermarked audio sequence. However, the quality of the watermarked audio is usually degraded, either intentionally by an adversary or unintentionally in the transmission process, before a person perceives it. It is more adequate to define the fidelity of a watermarking algorithm as a perceptual similarity between the watermarked audio and the original host audio at the point at which they are presented to a consumer.

Watermark Bit Rate

The bit rate of the embedded watermark is the number of the embedded bits within a unit of time and is usually given in bits per second (bps). Some audio watermarking applications, such as copy control, require the insertion of a serial number or author ID, with the average bit rate of up to 0.5 bps. For a broadcast monitoring watermark, the bit rate is higher, caused by the necessity of the embedding of an ID signature of a commercial within the first second at the start of the broadcast clip, with an average bit rate up to 15 bps. In some envisioned applications, for example hiding speech in audio or compressed audio stream in audio, algorithms have to be able to embed watermarks with the bit rate that is a significant fraction of the host audio bit rate, up to 150 kbps.

Robustness

The robustness of the algorithm is defined as an ability of the watermark detector to extract the embedded watermark after common signal processing procedures. Applications usually require robustness in the presence of a predefined set of signal processing modifications, so that watermark can be reliably extracted at the detection side. For example, in radio broadcast monitoring, an embedded watermark need only to survive distortions caused by the transmission process, including dynamic compression and low pass filtering, because the watermark is extracted directly from the broadcast signal. On the other hand, in some algorithms, robustness is completely undesirable and those algorithms are labelled *fragile audio watermarking* algorithms.

Blind or Informed Watermark Detection

A detection algorithm may use the original host audio to extract a watermark from the watermarked audio sequence (informed detection). It often significantly improves the detector performance, in that the original audio can be subtracted from the watermarked copy, resulting in the watermark sequence alone. However, if blind detection is used, the watermark detector does not have access to the original audio, which substantially decreases the amount of data that can be hidden in the host signal. The complete process of embedding and extracting of the watermark can be modelled as a communications channel where the watermark is distorted due to the presence of strong interference and channel effects. A strong interference is caused by the presence of the host audio, and channel effects correspond to signal processing operations.

5. WATERMARKING SCHEME EVALUATION

Digital watermarking has been presented as solutions for protection against illegal copying of multimedia objects and dozens algorithms. The requirements, tools and methodologies to assess the current technologies are al- most inexistent. The lack of benchmarking of current algorithms is blatant. This confuses rights holders as well as software and hardware manufacturers and prevents them from using the solution appropriate to their needs. Digital watermarking remains a largely untested field and only very few large industrial consortiums have published requirements against which watermarking algorithms should be tested. Even though number of claims has been made about robustness of watermarking, the growing number of attacks against such systems has shown that far more research is actually required to improve the quality of existing watermarking methods.

Using benchmarking authors and software providers would just need to provide a table of results which would give a reliable summary of the performances of the proposed scheme. So the users can check whether their requirements are satisfied the industry can properly evaluate risks associated to the use of a particular solution by knowing which level of reliability can be achieved by each contender. The idea to evaluate watermarking schemes is to implement an automated benchmark server for digital watermarking schemes and to allow users to send a binary library of their scheme to the server which in turns runs a series of tests on this library and keeps the results in a database accessible to the scheme owner or to all ‘water-markers’ through the Web.

The service has a simple interface with existing watermarking libraries (only three functions must be provided). It also takes into account the application of the watermarking scheme by proposing different evaluation profiles (sets of tests and images) and strengths.Each type of watermarking scheme needs different evaluation profiles without having to recompile the profile .Evaluation of profiles is not an easy task and the choice of these profiles does not affect the design of the evaluation service. The main function that should be done here is to evaluate the permeability of scheme, its capacity, its reliability (robustness to at- tacks and false alarm rate) and its performances (mainly the speed of execution). For each of these set of tests we have implemented ad-hoc libraries which are built easily on top of the core libraries. Perceptibility characterizes the amount of distortion introduced during by the watermarking scheme itself. The problem here is very similar to the evaluation of compression algorithms. The capacity of a scheme is the amount of information one can hide. In most applications the capacity will be a fixed constraint of the system so robustness test will be done with a random payload of given size. Our benchmark provide a test that help to analyze this trade-off by drawing different graphs.

The robustness can be assessed by measuring the detection probability of the mark and the bit error rate for a set of criteria that are relevant to the application, which is considered. Finally, related to speed our test just computes the average of the time required on a particular given platform to watermark and image depending on its size. The evaluation service only requires three functions to be exported from the watermarking library supplied by the user. All possible cases are captured and it ended up with a solution where several parameters are provided but not all of them are mandatory. They include the original medium, the watermarked medium, and the embedding key, the strength of the embedding, the payload, the maximum distortion tolerated and the certainty of extraction.

6. EXISTING SYSTEM

In an audio watermarking technology which is based on chaotic map and modified Patchwork algorithm, a chaotic sequence is introduced in the embedding process to ensure the security. And a novel Patchwork algorithm is projected in DWT domain. A portion of DWT coefficients in two patches have been modified in different ways according to bit code, thus, their statistical characteristics shift to opposite directions. But the disadvantage in this algorithm is degradation of host audio.

An algorithm based on Gammatone Filter bank which has remarkable resistance against common manipulations and attacks such as adding noise, low-pass filtering, resampling, lossy compression , random sampling etc. Gammatone filter bank (GTF) is a bank of overlapping band-pass filters, which mimics the characteristics of the human cochlea. Even though it has many merits, the demerit here is the values of SNR and BER are pretty high**.** A novel audio watermarking scheme based on the statistical feature manipulation in wavelet domain combined with error correction coding technique is used in this method. Here, a physical feature insensitive to attacks based on the idea of Invariant watermark is found and the watermark is embedded by modifying them directly. Robustness can be increased by using repetition codes and BCH codes. But only under the condition that BER is below 10%, the use of BCH codes makes sense. These attributes are overcome by a highly confidential audio watermarking scheme that is proposed.

7. PROPOSED WATERMARKING SCHEME

Audio watermarking is a promising solution to copyrights protection for digital audio and multimedia products. To achieve its objectives, a qualified audio watermarking scheme should possess excellent imperceptibility for transparent perception, high-level security for preventing authorized detection, and strong robustness against various attacks, such as noise addition, MPEG compression, reverberation, random samples cropping/inserting, time stretching and pitch shifting. Previously implemented audio watermarking schemes have excellent capabilities for the purpose of copyrights protection. In this system, performance on security is improved using multiple scrambling. Every scrambling operation has its independent secret key; a pseudorandom sequence, the detection can be only conducted properly when all the keys are known. This means that we can be able to revive the watermark even from the attacked audio files with loss of synchronization.The proposed system is a Havoc free multiple audio watermarking is a secure audio watermarking scheme which uses multiple scrambling. This new scheme is self-secured by integrating multiple scrambling operations into the embedding stage.

Firstly, a pre-selection is applied on the host audio signal to determine the embedding segments. Only the regions whose power exceeds a certain threshold will be chosen for watermarking. Before embedding the actual watermark in those embedding segments, the watermark is converted into the coded data because it can be identified visually, which is a kind of ownership stamp. Then the coded image is processed for encryption. The first scrambling operation is to encrypt the coded image watermark into incomprehensible ciphers, where one secret key is used. After the image is encrypted, the image bits are randomized in their order of encoding and then it is embedded in the host audio signal. Instead of using all the frames, we randomly select certain frames out of the total frames and randomize their orders of encoding. Since the secret keys are shared only between the embedder and authorized detectors, the goal of copyrights protection is really achieved. The proposed system uses DC watermarking scheme which hides watermark data in lower frequency components of the audio signal, that are below the perceptual threshold of the human auditory system. Security can be improved by using this multiple scrambling method. The proposed scheme can be further improved by embedding multiple watermarks. Multiple watermarks, each of which has different characteristics are embedded in an audio signal.

The characteristics of the various watermarks are chosen so that each of the watermarks will be affected in a different manner if the audio signal is subsequently copied and reproduced. Thus, the audio watermarking is a promising solution to copyrights protection for digital audio and multimedia products. Thus its objectives can be achieved by using this havoc free multiple audio watermarking schemes. Audio watermarking involves the concealment of data within a discrete audio file. Audio watermarking technology thus affords an opportunity to generate copies of a recording which are perceived by listeners as identical to the original but which may differ from one another on the basis of the embedded information.

Multiple Scrambling

The proposed scheme is self-secured by integrating multiple scrambling operations into the embedding stage. Along with the random settings on the amount and positions of slots assigned to each watermark bit, anyone without all the secret keys rarely has the possibility to find out the watermark. Since the secret keys are shared only between the embedder and authorized detectors, the goal of copyrights protection is really achieved.

DC Watermarking Scheme

The DC watermarking scheme can be used to hide auxiliary information within a sound file. The watermarking scheme provides an overview of techniques which are common to all digital audio watermarking schemes. The DC watermarking scheme hides watermark data in lower frequency components that are perceptible to the human auditory system of the audio signal and that are below the perceptual threshold of the human auditory system. From the spectral analysis of each frame, the low frequency (DC) component F(1), can now be removed by subtraction from each frame using the following formula:

$$f(n) = \sum_{k=1}^{3969n} f(k) - F(1) \qquad n = 1, 2, \ldots, N$$

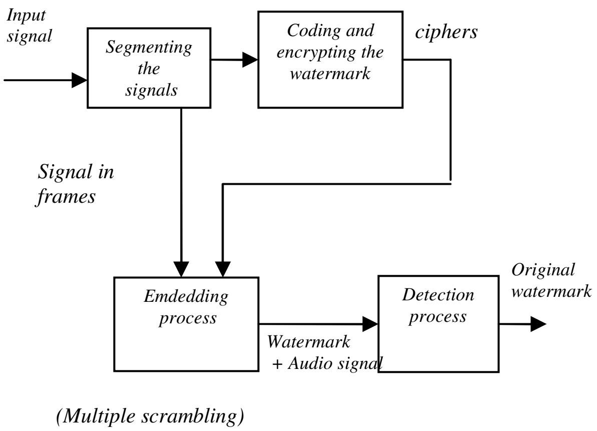


FIGURE 1. Multiple Scrambling Operations of Audio watermarking Scheme

Segmenting The Audio Signal

The audio file is portioned into frames which are 90 milliseconds in duration. This frame size is chosen so that the embedded watermark does not introduce any audible distortion into the file. With a 90 ms frame size, our bit rate for watermarked data is equal to 1 / 0.09 = 11.1 bits per second.

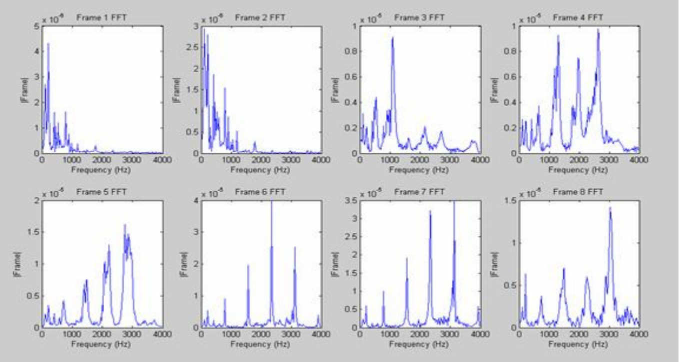


FIGURE 2. Sample spectrum of signal frames

Coding And Encrypting The Watermark

The image to be embedded is converted into a coded binary image with bits '1' and '0' as visual watermark, instead of meaningless pseudorandom or chaotic sequence. Not only because coded binary image can be identified visually, which is a kind of ownership stamp indeed, but also post processing on the extracted watermark could be done to enhance the binary image and consequently the detection accuracy will increase. Image de-noising and pattern recognition are examples of post processing techniques for automatic character recognition. Thus, on top of the bit error rate, coded binary image provides a semantic meaning for reliable verification. Next the coded binary image is encrypted for the security purpose, which involves a secret key.

Watermark Embedding Process

Multiple Scrambling

To increase the level of security, multiple scrambling can be used in the embedding. The first scrambling operation is to encrypt the coded image watermark into incomprehensible ciphers, where one secret key is used. Furthermore, instead of using all the subbands, we randomly select some frames out of total frames and randomize their orders of encoding, where two secret keys are employed. Along with the random settings on the amount and positions of frames assigned to each watermark bit, anyone without all the secret keys rarely has the possibility to find out the watermark. Since the secret keys are shared only between the embedder and authorized detectors, the goal of copyrights protection is really achieved.

Embedding Process

The process of embedding a watermark into an audio file is divided into four main processes. An original audio file in wave format is fed into the system, where it is subsequently framed, analyzed, and processed, to attach the inaudible watermark to the output signal.

Framing

As with the insertion process, the audio file is partitioned into frames which are 90 milliseconds in duration. With a 90 ms frame size, we expect an extracted watermark data rate equal to 11.1 bits per second.

Spectral Analysis

Subsequent to the framing of the unprocessed audio signal, spectral analysis is performed on the host audio signal, consisting of a fast Fourier transform (FFT), which allows us to calculate the low frequency components of each frame, as well as the overall frame power. The FFT processing is accomplished in Mat lab, using the following equation:

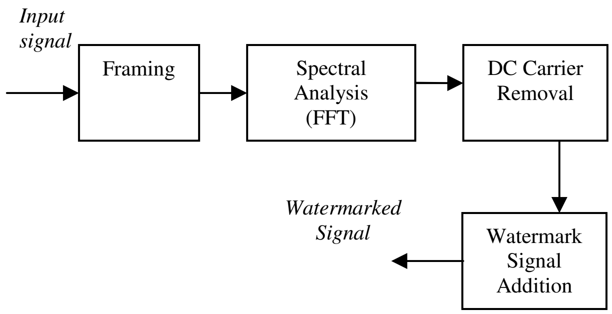


FIGURE 3- Watermark Embedding Process

$$F(k) = \sum_{n=1}^{N} f(n) e^{\frac{-j2\pi(n-1)(k-1)}{N}} \qquad k = 1, 2, \quad , N$$, N denotes the last frame in the audio file.

With a standard 16 bit CD quality audio file having a sampling rate, Fs = 44,100 samples per second, a frame consists of 3969 samples. If we perform a FFT on a frame of this size

with N = 3969, we end up with a frequency resolution as follows:

$$\frac{44{,}100 \ Hz}{2 \times \frac{3969}{2} \ samples} = 5.6 \ Hz \ resolution$$

From the FFT, we are now able to determine the low frequency (DC) component of the frame F(1), as well as the frame spectral power. To calculate the frame power, we use the sum of amplitude spectrum squared:

$$P_{Frame}(n) = \frac{1}{\left(\frac{3969}{2}+1\right)} \sum_{k=1}^{3969+1} F(k)^2 \qquad n = 1, 2, .., N$$

DC Removal

From the above spectral analysis of each frame, the low frequency (DC)

$$P_{Frame}(n) = \frac{1}{\left(\frac{3969}{2}+1\right)} \sum_{k=1}^{3969+1} F(k)^2 \qquad n = 1, 2, \ldots . N$$

From the above spectral analysis of each frame, the low frequency (DC) component F(1) is calculated, which can now be removed by subtraction from each frame using the following formula:

$$f(n) = \sum_{k=1}^{3969n} f(k) - F(1) \qquad n = 1, 2, .., N$$

Watermark Signal Addition

From the spectral analysis completed previously, we calculated the spectral power for each frame, which is now utilized for embedding the watermark signal data. The power in each frame determines the amplitude of the watermark which can be added to the low frequency spectrum. The magnitude of the watermark is added according to the formula:

$$f(n) = \sum_{k=1}^{3969n} f(k) + K_s \times w(n) \times P_{frame}(n) \qquad n = 1, 2, ..., N$$

Where Ks is the scaling factor, which ensures the watermark is embedded below the audibility threshold, and w(n) represents the watermark signal data, which is binary, having a value of 1, or -1.The f (n) function has now been watermarked with the above process, and is ready for storage, testing, and watermark extraction.

Watermark Detection Process

Watermark Extraction

The process of extracting the digital watermark from the audio file is similar to the technique for inserting the watermark. The computer processing requirements for extraction are slightly lower. A marked audio file in wave format is fed into the system, where it is subsequently framed, analysed, and processed, to remove the embedded data which exists as a digital watermark.

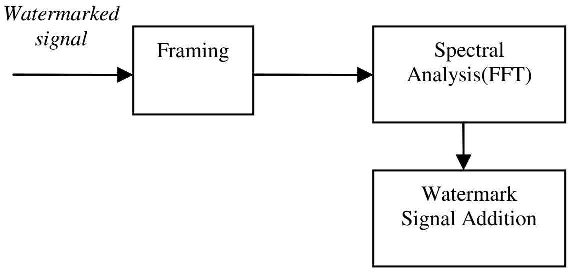


FIGURE 4 - Watermark Extraction Process

Watermark Signal Extraction

From the spectral analysis completed previously, we calculated the spectral power for each frame, which allows us to examine the low frequency power in each frame and subsequently extract the watermark, according to the following formula:

$$w(n) = \begin{cases} 1 & if\, F_n(1) \geq 0 \\ 0 & if\, F_n(1) \leq 0 \end{cases} \qquad n = 1, 2, \ldots, N$$

where, N denotes the last frame in the audio file.

The extracted watermark signal, w (n), should be an exact replica of the original watermark, providing the original audio file has enough power per frame to embed information below the audible threshold, and above the quantization floor.

8. EXPERIMENTAL RESULTS

The implementation phase begins with the process of dividing the audio signal into a certain number of frames such that each frame is 90 milliseconds in duration. This frame size is chosen so that the embedded watermark does not introduce any audible distortion into the file.

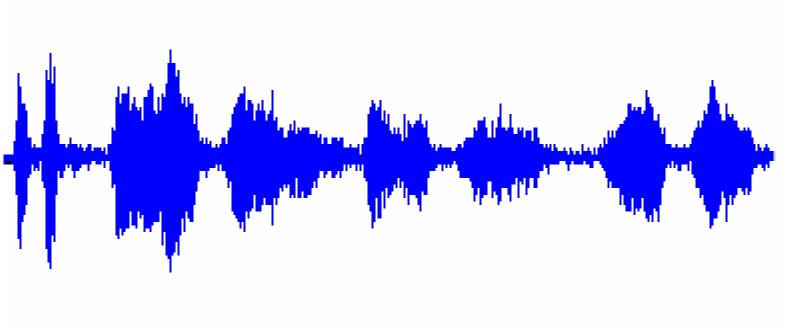


FIGURE 5. Input Audio Signal

The sample spectrum of the original audio signal is shown below .

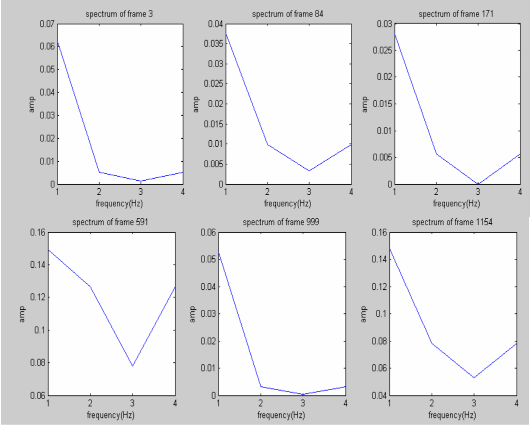


FIGURE 6 Sample Spectrum Of the Original Audio Signal

The image to be watermarked is chosen and is binary coded as shown in figure 7.

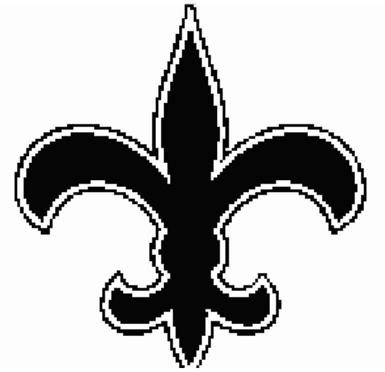


FIGURE 7 Image to be watermarked

After coding the image watermark, it is embedded using Multiple Scrambling and DC watermarking scheme. Following the process of embedding, the original audio signal now consists of the image watermark. It is shown as in figure 8. The embedded watermark is then detected from the watermarked signal in the detection process.

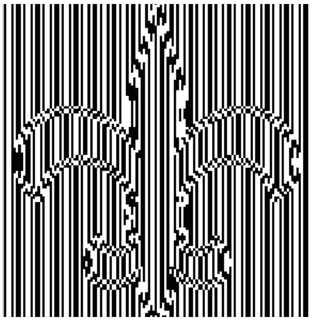


FIGURE 8. Encrypted image

After encrypting the image watermark, it is embedded using Multiple Scrambling and DC watermarking scheme. Following the process of embedding, the original audio signal now consists of the image watermark. The sound is played using the sound viewer.

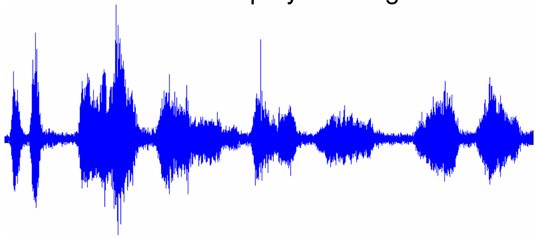


FIGURE 9 Watermarked Audio Signal

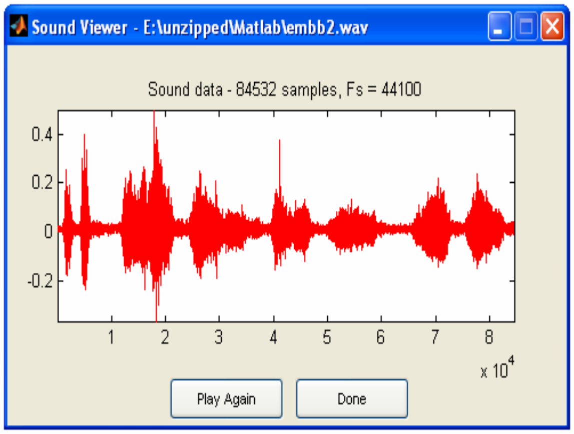

Sound Viewer - E:\unzipped\Matlab\embb2.wav
Sound data - 84532 samples, Fs = 44100
0.4
0.2
0
-0.2
1
2
3
4
5
6
7
8
x 10^4
Play Again
Done

FIGURE 10 Sound Viewer

9. CONCLUSION AND FUTURE WORK

Audio watermarking can also be used for fingerprinting and additional features to audio contents besides the functions as copyright protection. a secure and robust audio watermarking scheme using coded-image watermark, multiple scrambling and adaptive synchronization is proposed .The report has also found that in order to achieve these functionalities, a watermarking scheme has to meet the requirements of perceptibility, reliability, capacity and speed. The coded image can further improve the watermark detection by using image processing techniques and pattern matching analysis Compared with digital image and video watermarking technologies, audio watermarking technology provides a special challenge because the human auditory system is extremely more sensitive than human visual system. Audio watermarking is a persistent data communication channel within an audio stream. It should survive through various format changes. Watermarked Audio Signal and manipulations (either legitimate or not) of the audio material, as long as the content retains some commercial potential. Additionally, it should do so without introducing any perceivable audio artifacts. Most of the audio watermarking technology includes many demerits such as degradation of host audio signal, high Bit Error Ratio and Signal to Noise Ratio of the host signal, less security and etc. which can be completely overcome by the proposed audio watermarking technology. Also with the help of multiple scrambling, the scheme is strictly self-protected and any attacker without all the secret keys is impossible to ascertain or destroy the watermark embedded without noticeably degrading the signal. The experimental results prove that the system is secure and robust. The work can be extended for security against collusion attacks.

ACKNOWLEDGMENT

We would like to acknowledge all the persons who have contributed to a great extent towards the initialization, development and success of this paper. With great respect and obedience, We would like to thank our parents who were the backbone behind our deeds.

REFERENCES

[1] F.A.P. Petitcolas, "Watermarking schemes evaluation", IEEE Signal Processing Magazine, vol. 17, no. 5, pp. 58-64, 2000.

[2] Rangding Wang, Qian Li and Diqun Yan ," A High Robust Audio Watermarking Algorithm", CKC Software Lab, University of Ningbo,2006.

[3] R. Tachibana, "Improving audio watermarking robustness using stretched patterns against geometric distortion", IEEE Pacific-Rim Conference on Multimedia (PCM'02), pp. 647-654, 2002.

[4] Tong Won Seok and Jin Woo Hong, "Audio watermarking for copyright protection of digital audio data", Electronics Letters, Vol. 37, No. 1,2001.

[5] Y.Q. Lin, W.H. Abdulla, "Robust audio watermarking for copyrights protection", Technical Report (No. 650), Dept. of Electrical & Computer Engineering, The University of Auckland, 2006. http://www.ece.auckland.ac.nz/~wabd002/Publications.

[6] Y.Q. Lin, W.H. Abdulla, "Robust audio watermarking technique based on Gammatone filterbank and coded image", International Symposium on Signal Processing and Its Application (ISSPA'07), 2007.

[7] Eric Metois, Ph.D. "Audio Watermarking and Applications", ARIS Technologies, Inc. – September 1999.

[8] A. Gurijala, J.R. Jr. Deller, "Robust algorithm for watermark recovery from cropped speech",IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP'01), vol. 3, pp. 1357-1360, 2001.

[9] W. Li, X. Xue, "An audio watermarking technique that is robust against random cropping", Computer Music Journal, vol. 27, no. 4, pp. 58-68, 2003.

[10]http://www.ece.uvic.ca/~aupward/w/watermarking.htm.

[11] Yiqing Lin Waleed H. Abdulla "A Secure and Robust Audio Watermarking Scheme Using Multiple Scrambling and Adaptive Synchronization" International Symposium on Signal Processing and Its Application, 2007

[12] W. Li, X. Xue, "An audio watermarking technique that is robust against random cropping", Computer Music Journal, vol. 27, no. 4, pp. 58-68, 2003.

[13] EBU, "SQAM - Sound Quality Assessment Material", http://sound.media.mit.edu/ mpeg4/ audio/ sqam/

[14] Essaouabi Abdessamad ,E.Ibneihaj, F.Regragui," A Wavelet- based object watermarking system for MPEG4 Video ", International journal of Image Processing,(IJIP),volume (3): issue(6).

Distributed Co-ordinator Model for Optimal Utilization of Software and Piracy Prevention

Vineet Kumar Sharma vineet_sharma@kiet.edu
Associate Professor, Department
of Computer Science & Engg.
Krishna Institute of Engineering & Technology
Ghaziabad, 201206, U.P., India

Dr. S.A.M. Rizvi samsam_rizvi@yahoo.com
Associate Professor, Department
of Computer Science
Jamia Millia Islamia, central university
New Delhi, 110025, India

Dr. S.Zeeshan Hussain szhussain@rediffmail.com
Asst Professor, Department
of Computer Science
Jamia Millia Islamia, central university
New Delhi, 110025, India

Abstract

Today the software technologies have evolved it to the extent that now a customer can have free and open source software available in the market. But with this evolution the menace of software piracy has also evolved. Unlike other things a customer purchases, the software applications and fonts bought don't belong to the specified user. Instead, the customer becomes a licensed user — means the customer purchases the right to use the software on a single computer, and can't put copies on other machines or pass that software along to colleagues. Software piracy is the illegal distribution and/or reproduction of software applications for business or personal use. Whether software piracy is deliberate or not, it is still illegal and punishable by law. The major reasons of piracy include the high cost of software and the rigid licensing structure which is becoming even less popular due to inefficient software utilization. Various software companies are inclined towards the research of techniques to handle this problem of piracy. Many defense mechanisms have been devised till date but the hobbyists or the black market leaders (so called "software pirates") have always found a way out of it. This paper identifies the types of piracies and licensing mechanisms along with the flaws in the existing defense mechanisms and examines social and technical challenges associated with handling software piracy prevention. The goal of this paper is to design, implement and empirically evaluate a comprehensive framework for software piracy prevention and optimal utilization of the software.

Keywords: End User License Agreements, User Datagram Protocol, Distributed software and License key management.

1. INTRODUCTION

Most retail programs are licensed for use at just one computer site or for use by only one user at any time. By buying the software, the customer becomes a licensed user rather than an owner. Customers are allowed to make copies of the software for backup purposes, but it is against the law to give copies to friends and colleagues. Software piracy is all but impossible to stop, although software companies are launching more and more lawsuits against major infractions'. Originally, software companies tried to stop software piracy by copy-protecting their software. This strategy failed, however, because it was inconvenient for users and was not 100 percent foolproof. Most software now requires some sort of registration, which may discourage would-be pirates, but doesn't really stop software piracy.

An entirely different approach to software piracy, called shareware, acknowledges the futility of trying to stop people from copying software and instead relies on people's honesty. Shareware publishers encourage users to give copies of programs to friends and colleagues but ask everyone who uses a program regularly to pay a registration fee to the program's author directly. Commercial programs that are made available to the public illegally are often called "warez".

Software piracy now cost in the range $15-20 annually to the companies. The main objective is to prevent intellectual property of individuals and organizations. Software piracy prevention is important because the legal actions are lengthy and it is difficult to change the moral standards of the people.

It is a matter of history that with the introduction of IBM PC in the early 1980's a revolution began. Some famous software applications like "word star","lotus123","dBase" were used with IBM PC. Hardware was relatively more expensive than software, and often inclusions of the latest version of these software packages with new system were very common and routinely expected. Upgrades to the software packages were available usually directly from the developers, or as often was the case through resellers, but always incurred additional fees. Tragically, this also led to underground trade on these applications. Ironically, the more popular the application was, the greater its appeal in the so called "black market" and its traders, the "software pirates".

Among the various approaches that have been explored recently to counteract the problem of software piracy, some are of legal, ethical and technical means. Legal means are based on the fear of consequences of violating piracy law. But while most software piracy cases legal means are available, prosecution on a case by case basis is economically unviable. Furthermore, it is conceived as bad publicity and can take a long time. Ethical measures relate to making software piracy morally unappealing. While the intentions are laudable, it takes even more time to change the moral standards of a larger group of people. The technical means include the static measures of defense which incorporates in itself the protection mechanism that is built into the distributed database. Once the system is broken then the static protection techniques are not satisfactory at all.

The concept of software license[8] was developed by the software industry since its early inspection. Software license is that the software publisher grants a license to use one or more copies of software, but that ownership of those copies remains with the software publisher. One consequence of this feature of proprietary software licenses is that virtually all rights regarding the software are reserved by the software publisher. Only a very limited set of well-defined rights are conceded to the end-user. Most licenses were limited to operating systems and development tools. Enforcement of licenses was relatively trivial and painless. Any software customer had certain rights and expectations from the software and developer. Some software was licensed only to one user or one machine, while some software may have been licensed to a site specifying the maximum number of machines or concurrent instances of the program in execution (processes). These are also known as "End User License Agreements" (EULAs). The terms of each EULA may vary but the general purpose is the same – establish the terms of contract between software developer and user of software product.

2. TYPES OF SOFTWARE LICENSES

1) Individual licenses (Single-user) This license type allows the software to be used on only one computer which is not accessed by other users over a network. The other users are not allowed to use software while connected to your computer. Types of individual licenses are:[10]

a) Perpetual license: allows the customer to install and use the software indefinitely without any limitation.

b) Subscription license: Allows the user to use the software for a specific time period. At the end of the term the user has several options: (1) renew the subscription; or (2) purchase a perpetual license at a discounted cost; or (3) remove the software from the computer.

c) Trial license: Allows software vendors to release the software for a period time like a month as marketing tool. The end user can only use this software for only one month.

d) Evaluation license: This license allows the expensive software to be evaluated with certain period.

e) Demo license: This license allows end users to demonstrate the software with partial function or certain period.

f) Feature based license: This license allows end users to use partial function of the software to save money than to use the whole package.

g)Time limited rental license: This allows the end user to pay per time. The end user pre-pay a period of time that fits their needs. The end user can also renew before the license expired.

2) Network /Multiuser licenses:

a) Server (Network): Licensed per server – This license type requires that you have a single copy of the software residing on the file server.

b) Per Seat (Machine): Licensed per machine/seat – This license requires that you purchase a license for each client computer and/or device where access to services is needed. This license is typically used in conjunction with a network license.

3) Add-on's to existing or new licenses:

a) Upgrade: This license is acquired when a user has a previously acquired software license and would like to move up to a newer version.

b) Student use: This allows students to use the software as long as they are students of the institutions. Students are required to uninstall software upon leaving the University.

c) Secondary use: Allows the licensed end user to use the software on a second computer.

d) Work-at-home rights: Allows Faculty/Staff to use software at home. This is effective for as long as the primary work computer is licensed and as long as the person is an employee. Termination of employment also terminates this benefit.

2.1 How piracy is done?

The pirates can be called as cryptanalysts who decrypt the patches from certain software CD's and the license key available with that CD. Each CD of the software has a unique license key available with it. The pirates take up few CD's along with their license key and find out the patch which is consistent with that particular CD. Then they make copies of such CD's along with the same patch and sell it in the market illegally.

2.2 Types of piracy

If you're computer savvy, you're probably aware that copying and distributing copyrighted software is illegal, but you may be surprised to know that there are actually many ways you can unintentionally pirate software. It seems that illegal software is available anywhere, to anyone, at any time. From shopping malls, to the unscrupulous computer systems retailers a few blocks down the street, pirated programs are sold for a pittance. Becoming familiar with the different types of software piracy can help protect you from purchasing pirated software [7]. Many computer users have found themselves caught in the piracy trap, unaware they were doing anything illegal. To avoid such unpleasant surprises, it may be helpful to know the basic ways one can intentionally or unintentionally pirate software:

1) Cracks and serials: Cracks and serials are forms of software piracy that consists of legally obtaining an evaluation version and subsequently entering a copied license code or applying a generic patch that undergoes a copy protection. This is a widespread form of piracy. It is so popular because of small amount of information that needs to be exchanged illegally distribute and obtain a license code or a patch than a complete program.

2) Softlifting: Softlifting occurs when a person (or organization) purchases a single licensed copy of a software program and installs it onto several computers, in violation of the terms of the license agreement. Typical examples of softlifting include, "sharing" software with friends and co-workers and installing software on home/laptop computers if not allowed to do so by the license. In the corporate environment, softlifting is the most prevalent type of software piracy - and perhaps, the easiest to catch.

3) Unrestricted Client Access: Unrestricted client access piracy occurs when a copy of a software program is copied onto an organization's servers and the organization's network "clients" are allowed to freely access the software in violation of the terms of the license agreement. This is a violation when the organization has a "single instance" license that permits installation of the software onto a single computer, rather than a client-server license that allows concurrent server-based network access to the software. A violation also occurs when the organization has a client-server license but is not enforcing the user restrictions outlined in the license agreement. This occurs, for instance, when the license places a restriction on the number of concurrent users that are allowed to access the software but the organization is not enforcing that number. Unrestricted client access piracy is similar to softlifting, in that it results in more employees having access to a particular program than is permitted under the license for that software. Unlike softlifting though, unrestricted client access piracy occurs when the software is installed onto a company's server - not on individual machines - and clients are permitted to access the server-based software application through the organization's network.

4) Hard-Disk Loading: Hard-disk loading occurs when an individual or company sells computers preloaded with illegal copies of software. Often this is done by the vendor as a means of encouraging the consumer to buy certain hardware. If you buy or rent computers with preloaded software, your purchase documentation and contract with the vendor must specify which software is preloaded and that these are legal, licensed copies. If it does not and the vendor is unwilling to supply you with the proper documentation, do not deal with that vendor.

5) OEM Piracy/Unbundling: OEM (original equipment manufacturer) software is software that is only legally sold with specified hardware. When these programs are copied and sold separately from the hardware, this is a violation of the distribution contract between the vendor and the software publisher. Similarly, the term "unbundling" refers to the act of selling software separately that is legally sold only when bundled with another package. Software programs that are marked "not for resale" are often bundled applications.

6) Unauthorized Use of Academic Software: Many software companies sell academic versions of their software to public schools, universities and other educational institutions. The price of this software (and sometimes the functionality) is often greatly reduced by the publisher in recognition of the educational nature of the institutions. Using academic software in violation of the software license is a form of software piracy. Acquiring and using academic software hurts not only the software publisher, but also the institution that was the intended recipient of the software.

7) Counterfeiting: Counterfeiting is the duplication and sale of unauthorized copies of software in such a manner as to try to pass off the illegal copy as if it were a legitimate copy produced or authorized by the software publisher. Much of the software offered for bargain sale at computer trade shows and on auction and classified ads sites is counterfeit software. The price and source are often indicators as to whether software is counterfeit. For example, if a particular piece of software normally retails for $1399 but is being sold or auctioned for $199 then red flags should go up. Likewise, if a particular software vendor only sells its products though certain authorized channels, then a purchaser would be wise not to purchase the software at a trade show.

8) CD-R Piracy: CD-R piracy is the illegal copying of software using CD-R recording technology. This form of piracy occurs when a person obtains a copy of a software program and makes a copy or copies and re-distributes them to friends or for re-sale. Although there is some overlap between CD-R piracy and counterfeiting, with CD-R piracy there may be no attempt to try to pass off the illegal copy as a legitimate copy - it may have hand-written labels and no documentation at all. With technological advancements making it relatively easy and inexpensive to copy software onto a CD-R, this form of piracy has escalated. As with counterfeit CDs, CD-R piracy is rampant on auction and classified ad sites.

9) Download Piracy: Download piracy is the uploading of software onto an Internet site for anyone. Anyone who uploads or downloads the software is making an illegal copy and is therefore guilty of software piracy. Examples of this include the offering of software through a website, P2P network or share hosting site. Incidences of Internet piracy have risen exponentially over the last few years.

10) Manufacturing Plant Sale of Overruns and 'Scraps': Software publishers routinely authorize CD manufacturing plants to produce copies of their software onto CD-ROM so that they can distribute these CD-ROMs to their authorized vendors for resale to the public. Plant piracy occurs when the plant produces more copies of the software than it was authorized to make, and then resells these unauthorized overruns. Piracy also occurs when the plant is ordered by the publisher to destroy any CDs not distributed to its vendors, but the plant, in violation of these orders, resells those CDs that were intended to be scrapped. While most plants have compliance procedures in place, there have been several instances of this type of piracy.

11) Renting: Renting software for temporary use, like you want to watch a movie, was made illegal in the United States by the Software Rental Amendments Act of 1990 and in Canada by a 1993 amendment to the Copyright Act. As a result, rental of software is rare.

The types of piracy identified above are not mutually exclusive. There is often overlap between one type of piracy and another. OEM software is unbundled by the pirate in order to be re-sold. Not only does the pirate sell the OEM software, but he also makes numerous illegal copies of the OEM software and sells them as counterfeits.

3. OPTIMAL USE OF SOFTWARE

For optimal use of the software a model in an organization is used which tries to keep the information about the specified software on a single machine (considered as co-ordinator) and the complete management of the dynamic distribution of that software and its license is to be done on the same machine. The selection of the co-ordinator is done arbitrary or by executing the election algorithms. If in any case the co-ordinator goes down than any other machine is voluntary elected as the co-ordinator to provide uninterrupted functioning for dynamic or electronic distribution of the software license. Here the software and license key management is done dynamically by the co-ordinator machine. The co-ordinator machine is responsible to make an account for all those machines which are executing the software. In this methodology the organization cannot use the software on the number of computers, exceeding the number of license purchased but this methodology provides an ethical way for optimal uses of the software on the network of an organization. Therefore it prevents organizational piracy and supports optimal use of software on the network of an organization[4], for e.g. there are 500 users on a network and software is used by at most 300 users at a time then it is better to take 300 licenses and use it with the prevention of piracy. In this scheme a machine known as co-ordinator is dedicated for dynamic software and license management.

3.1 Distributed Co-ordinator Model: The approach of “single coordinator model” can be enhanced in terms of efficiency and performance. In single coordinator model there is only one coordinator which manages the distribution of software license keys among the client machines when there is the requirement of execution of the software by that client. Now in the present approach the concept of “Super Coordinator” is used.

Now in this approach of “Super Coordinator” a tree based approach is used. In this-“tree based approach” the root is the super coordinator and its children are the sub coordinators. Various client machines communicate with their corresponding sub coordinators only. Here, sub coordinator takes up the job of providing the license keys to the client machines. The super coordinator performs the crucial task of assigning the particular number of license keys to the sub coordinators as per their requirement. For example, super coordinator has 500 license keys and then it assigns 100 license keys to sub coordinator 1, 150 license keys to sub coordinator 2, 100 license keys to sub coordinator 3, and 80 license keys to sub coordinator 4. The remaining 70 license keys are kept with the super coordinator itself which can be used in

case of need and can be assigned to a particular sub coordinator on demand. In case sub coordinator 3 is finished executing the software and is in no need of more than 40 license keys then the remaining license keys (that is, 60) is returned back to the super coordinator and can be used in future need by any of the sub coordinator.
The operation of this system of approach is quite similar as that of "single coordinator" approach. If a client machine needs a license key then communication between the client and its sub coordinator occurs. If the sub coordinator has a free license key (free license key is the key that is not being used by any machine for the execution of the software) then it is been sent to the client and the client starts its execution of the software. On the contrary, if the sub coordinator does not have any free license key then it requests the super coordinator to provide it with the license key which it could provide to its client. If the super coordinator has a free license key then it is provided to the sub coordinator and then to the client machine, otherwise the sub coordinator along with the client machine goes into waiting queue and waits for the license key.

3.2 Role of sub-coordinator: When a user wants to execute the software application it broadcasts port specific UDP message in its sub network to the sub coordinator (this is the only sub coordinator which is listening request messages on that port). On receiving the request the sub coordinator checks for the available license keys in the data bank, if keys are available then it is delivered to the client and makes its entry in the active client list. On the contrary if the sub coordinator does not have requisite number of keys then it maintains a waiting queue and if it crosses a particular "threshold value" then it sends request message to the super coordinator. Now there arises two cases: Case:1- If the super coordinator has available number of keys then it grants those keys to the sub-coordinator. Case:2-If the super coordinator possess lesser keys or does not posses any available key messages are transmitted to each sub-coordinators and then the sub-coordinators surrender their unused keys to the super coordinator.

3.3 Fault tolerance On peaceful termination of the client the keys available with it are surrendered back to its sub-coordinator along with the deletion of its entry from the active client list available with other clients. In case of abnormal termination of the client there is a procedure of dual check, in this process an "Is-Alive" message is repetitively sent to each client by the sub coordinators, if a response is received from the client then it means that the client is functioning. On the other hand if there is no response from the client then the sub coordinator checks it for second time by sending the message again. If this request is also without the response then the client is supposed to be "dead" and it results in surrendering of its keys to the sub coordinator and deletion of its entry from the active client list of every alive clients. Now there arises a case of sub-coordinator crash, in such cases there occurs a voluntary selection of a client as the sub-coordinator and then this information is passed to all the available clients by updating the active client list of every client.

3.4 Algorithm for key distribution

A hierarchical approach is used consisting of three entities at different levels Super-Coordinator (level 0), Sub-Coordinator (level 1) and the clients (level 2).

1. Clients, who want to execute the software, first of all broadcast the *Search UDP Port Specific Message* in its own subnetwork. This message packet is captured by the sub-coordinator working in the same subnetwork. Sub-coordinator sends the response packet back to the same client. Because UDP packet contains the ip address of both sender and receiver, the client gets the ip address of the sub-coordinator.
2. Now the same client unicasts the *Request UDP Port Specific Message* to its sub-coordinator requesting for a license key for execution of the software.
3. The sub-coordinator receives the request message and check out the availability of the license keys.

3.1. If license keys are available with the sub-coordinator then it is provided to the requesting client and its entry is being made in the current active list of the sub-coordinator.

3.2. If the sub-coordinator does not have any available key then it ask the client whether it is ready to wait or want to quit.

3.2.1. If the client is ready to wait to get a license key then the sub-coordinator makes its entry in the waiting client list. If the length of the waiting client list exceeds the threshold limit (T_H), the sub-coordinator sends a Request Message to the super-coordinator demanding the T_H license keys.

3.2.1.1. The super-coordinator checks the availability of the license keys.

3.2.1.2. If the super-coordinator possess more than or equal to T_H license keys then super-coordinator transmit T_H keys to the sub-coordinator.
3.2.1.3. Else the super-coordinator unicasts a Surrender Back UDP Message to each sub-coordinator except the one for which super-coordinator is intending to provide license keys, requesting to surrender the unused license keys
3.2.1.4. Each sub-coordinator receives this message and returns back some of the unused license keys to the super-coordinator.
3.2.1.5. Now the availability of the license keys is checked by the super-coordinator and if it is again lesser than T_H then step 3.2.1.3 is repeated until available keys becomes greater than or equal to the T_H or available keys do not increase at all.
3.2.1.6. These keys are transmitted to the sub-coordinator.
3.2.1.7. Sub-coordinator servers these keys to the clients of the waiting client list based on the FIFO principal.
3.2.2. Else the client quits.

3.5 Termination Algorithm

3.5.1 Peaceful Termination of a client

1. On the peaceful termination of its own software application the client sends a message to sub-coordinator.
2. Sub-coordinator deletes the entry of the specific client form its active client list and make one more license key available.

3.5.2 Abnormal termination of a client

1. Sub-coordinator sends a regular Is-Alive UDP Message to the clients which are in the active client list of the sub-coordinator.
2. When clients receive these messages they respond back to the sub-coordinator indicating their active state.
3. If sub-coordinator does not receive the response message from any specific client it performs the dual check by repeating the step 2.
   3.1. If this time again the same client does not respond back to the sub-coordinator, the sub-coordinator considers it as an abnormal termination of that client and deletes its entry from the current active client list and increases the available license key by one.

3.5.3 Abnormal termination of a sub-coordinator

1. Sub-coordinator periodically transmits the active& waiting client list, available unused license keys and the ip address of the super-coordinator to all the active clients. By this way all the active clients keep all the latest information which the sub-coordinator is bearing.
2. In case of the abnormal termination of the sub-coordinator, the active clients will not receive the Is-Alive messages. On the expiration of the timer which works at the client machines, the client can get an idea about the abnormal termination of the sub-coordinator.
3. The clients getting the information about the absence of the sub-coordinator will wait for a random amount of time and then they transmit the message packets to all other active clients indicating itself as the new sub-coordinator. Because the active client whose timer is expired, have to wait for a random time the contention of election of multiple coordinator is very less. But still if two or more clients take part in process of becoming the new coordinator, they settle this issue by the voting technique.
4. When a new coordinator is elected, it sends a *New Sub-coordinator UDP Message* to all the current and waiting clients. The new sub-coordinator also updates the super-coordinator about itself.

4. EVALUATION AND PERFORMANCE

This approach is better than the single coordinator approach in terms of efficiency. The efficiency of such systems is measured in terms of number of messages that are passed between the communicating entities (that is, client machine and the coordinator). More the number of messages transmitted between the communicators, less is the efficiency and vice versa. Therefore, efficiency is inversely proportional to the number of messages transmitted between the communicators.

In this approach the number of messages to be communicated is decreased to a great extent. As the technique used is a tree based approach, a client demanding the license key broadcasts the request messages in its own subnetwork only. This request packet is captured by the sub coordinator working in the same subnetwork. In case of a single coordinator the broadcast packets transmitted by a client are destined to all the machines of the entire network and this is how number of messages is reduced in the hierarchical coordinator environment. Thus the overhead of messages on a single coordinator is distributed on many sub coordinators. Similarly, the messages received and sent by super coordinator are reduced as no client machine communicates directly with the super coordinator but it communicates with the sub coordinator which in turn communicates with the super coordinator.

The major aspect of efficiency of this system is based on its fault tolerance mechanism. This approach is very viable in terms of failure of the coordinator. If a sub coordinator fails then the client machines which falls under its scope becomes functionless while other client machines under other sub coordinators work without any disturbance. This is how the complete system is prevented against absolute failure. On the other hand if the super coordinator fails then the sub coordinators are still functioning along with their respective client machines which are under their scope.

5. TIGHTER SECURITY BY COMBINING H/W AND S/W TOKENS

This invention relates to security mechanisms which prevent unauthorized use of the software, and in particular to mechanisms which prevent the unauthorized use of software on more than one computer. Various security mechanisms have been devised for preventing the use of software without authorization of the software supplier. These have included hardware security devices, which must be attached to a computer before the software can run on the computer. Typically, the software that is to run includes an inquiry which looks for an indication that the hardware device has been installed. Such hardware security devices assure that the software will only execute on one computer at any one time. These hardware devices[6] can, however, be relatively expensive and moreover need to be adaptable to various types of computers in which they are to be attached.

One of the best example to quote for this tight security provided by hardware and software together is seen in HSBC banks. The HSBC banks provide its users with software security of login and password and along with it, the bank also provides with a hardware key which produces a random number each time it is turned on. The user is supposed to use this identification number generated by the hardware key along with the login and password in order to access the account. The numbers generated by the key follow some particular sequence according to some algorithm which is been checked by the bank and then it allows the user to use his account.

5.1 How hardware based software license key works?

In order to improve the relationships between vendors and customers as well as grow revenue, software pricing and licensing policies are made more flexible. Hardware keys are made to feature secure software licensing options and offers multiple licensing modes locked into the hardware key to supply flexible licensing protection. Such as HASP from Aladdin, Sentinel from Safenet, UniKeyfromSecutech. Hardware key or software protection dongle is the hardware-based protection and licensing management tool. It is a USB key with memory that protects software against piracy and illegal use by allowing access and execution of the protected software only when the key is connected to the computer. The hardware keys [6] provide licensing method with envelope-base automatic implementation and API-based automatic implementation.

The only thing that would make hardware key better for software license management is if there were an industry-standard way to store multiple software licenses on one physical dongle (and transfer them securely if need be). Public computers could have those software installed on them at no cost to the business providing the computer, but they would only work when someone presented a valid license via their dongle. And the end user would never have to think about moving the licenses to a new computer or losing valuable software when they dispose of the old one. Other security mechanisms include software devices that look for an identification of the computer in which the software is to be installed rather than the presence of any hardware device [12]. Such software devices often require complex algorithms to generate a unique association of the program to be run with the identification of the host computer. These software devices may still require that other hardware devices be connected to the host computer in order to establish the unique association of the program to be run with the target computer. It is an object of the invention to provide a software security mechanism which authorizes run time execution of certain software only after generating an association of the software to be run with a single computer in a manner that does not require extremely complex algorithms or the attachment of any additional devices to the target computer.

6. CONCLUSION

Now a day the businesses of software companies are dependent on flexible software applications for changing market environments but the rigid licensing structures for software distribution, as used with most legacy systems, is becoming hurdle in it. The paper presents the types of piracy and how the piracy is done. Besides this it provides a new and ethical technique for the optimal utilization of the software resources of an organization in its own network environment by achieving a better degree of prevention of software piracy. Its strength is the dynamic distribution of software license keys to the clients with the help of hierarchical coordinators. The chance of piracy is eliminated up-to a remarkable position because no static measures to prevent piracy are used. In terms of efficiency the technique of hierarchical coordinator is better than the single coordinator technique.

7. REFERENCES

[1] C. Collberg and C. Thomborson. Software watermarking: Models and dynamic embeddings. In *Principles of Programming Languages*, pages 311–324, 1999.

[2] Mukesh Singhal & Niranjan G. Shivratri. Voting and Election Algorithms. In Advanced concept in operating Systems pages 209 & 343, 2002

[3] George Coulouris, Jean Dollimore & Tim Kindberg. Election Algorithm, Bully Algo & Ring based algo. In Distributed Systems page 445-448, 2006

[4] Leili Noorian & Mark Perry. Autonomic Software License Management System: an implementation of licensing patterns. 2009 Fifth International Conference on Autonomic and Autonomous Systems. IEEE 978-0-7695-3584-5/09

[5] Mathias Dalheimer and Franz-Josef Pfreundt. License Management for Grid and Cloud Computing Environments. 9th IEEE/ACM International Symposium on Cluster Computing and the Grid. IEEE 978-0-7695-3622-4/09

[6] Mikhail J. Atallah, Jiangtao Li. Enhanced Smart-card based License Management. IEEE International Conference on E-Commerce (CEC'03)0-7695-1969-5/03 2003 IEEE

[7] Yawei Zhang, Lei Jin, Xiaojun Ye Dongqing Chen.Software Piracy Prevention: Splitting on Client. 2008 International Conference on Security Technology IEEE 978-0-7695-3486-2/08

[8]Daniel Ferrante. Software Licensing Models:What's Out There? 1520-9202/06/ © 2006 IEEE

[9] Dinesh R. Bettadapur.Software Licensing Models in the EDA Industry 0-7803-4425-1/98/$10.00 1998 IEEE

[10] Sathiamoorthy Manoharan and Jesse Wu. Software Licensing: A Classification and Case Study. Proceedings of the First International Conference on the Digital Society (ICDS'07) 0-7695-2760-4/07 $20.00 © 2007 IEEE

[11] Zhengxiong Hou, Xingshe Zhou, Yunlan Wang Software License Management Optimization in the Campus Computational Grid Environment. Third International Conference on Semantics, Knowledge and Grid 0-7695-3007-9/07 © 2007 IEEE

[12] Petar Djekic & Claudia Loebbecke Software Piracy Prevention through Digital Rights Management Systems Proceedings of the Seventh IEEE International Conference on E-Commerce Technology (CEC'05) 1530-1354/05 $20.00 © 2005 IEEE

Knowledge Discovery through Data Visualization of Drive Test Data

Shalini Saxena *shalinisaxena1@gmail.com*
Dept. of Computer Science and Engineering
Florida Atlantic University
777 Glades Road, Boca Raton, FL - 33431

Abhijit S. Pandya *pandya@fau.edu*
Dept. of Computer Science and Engineering
Florida Atlantic University
777 Glades Road, Boca Raton, FL - 33431

Robert Stone *rstone@fau.edu*
Dept. of Computer Science and Engineering
Florida Atlantic University
777 Glades Road, Boca Raton, FL - 33431

Saeed Rajput *rajput@nova.edu*
Farquhar College of Arts & Sciences
Nova Southeastern University
Fort Lauderdale, FL - 33134

Sam Hsu *sam@cse.fau.edu*
Dept. of Computer Science and Engineering
Florida Atlantic University
777 Glades Road, Boca Raton, FL - 33431

ABSTRACT

This paper focuses on the analysis of a large volume of drive test data and the identification of adverse trends or aberrant behavior of a mobile handset under test by means of drive test data visualization. The first target application was to identify poor mobility decisions that are made by the handsets in calls. The goal was to compare a set of behaviors from a baseline unit (one accepted to generally operate well). We were able to identify a particular call that was exhibiting a different path (talking to a different cell than expected or taking longer to move to a new cell). In this paper we develop a mobility tool that evaluates the handset's performance by means of mapping the handoffs on the Google Maps. The mapping of the handoffs by means of the Google Maps were very powerful in identifying the above mentioned mobility patterns.

Keywords: Mobility Patterns, Hand-offs, Drive Test, Mobile phones.

1. INTRODUCTION

In wireless communications bandwidth is always divided into smaller sub-bands. The limited availability of electromagnetic spectrum or frequency band for transmitting voice call led to the development of the cellular radio networks [1]. It essentially increases the number of simultaneous conversations (called user capacity) for mobile radio telephone service by frequency reuse [1]. In cellular networks numerous lower-power transmitters each with shorter coverage are strategically deployed to cover a large geographic area. Each cell is served by a base station that consists of an antenna, a number of transmitters, a receiver and a control unit. Within any given cell, multiple frequency bands are assigned. The number of frequency bands depends on the traffic expected. Adjoining cells are assigned different group of frequencies to avoid interference and crosstalk. However, cells that are sufficiently far apart can reuse the same frequencies since radio signals strength diminishes with distance [1]. At any given instance, a number of mobile units are active and moving in a cell, communicating with the base station. Each base station is connected to a mobile switching center, which serves multiple base stations. The mobile switching center routes the calls depending on the location of the mobile unit, assigns voice channel to each call, performs handoffs, and monitors the call for billing information [1]. We will call this mechanism mobile call management.

Handoff is an important aspect of mobile call management. During a connection when the mobile unit moves from one coverage area of a base station to another it crosses the cell boundaries. The mobile unit must switch the traffic channel assigned to the old base station to the new base station as it crosses over its cell boundary. This process is called handoff and is performed ubiquitously. The Received Signal Strength Indicator (RSSI) gets weaker as the mobile unit moves away from the base station. When a neighboring site is stronger than the serving/current cell, the mobile unit requests a handover to another site [1]. The signal strength is allocated a level from 0 to 14 with 6 dB separation between the consecutive levels [2], with 0 being the quietest and 14 being the least quiet. Therefore after the handover, the RSSI typically sees at least a 6 dB improvement.

Handoff is an important feature of mobile call management because the continuity of a call is maintained through it when the mobile moves from one cell area to another. However, cell-dragging may occur when a mobile handset moves a considerable distance into the neighboring cell area without making a handoff, resulting in an increased level of system interference [3]. A handoff scheme that utilizes two adaptive algorithms in combination; one using a relative threshold and the other an absolute threshold, has been proposed in [3]. This scheme aims at minimizing cell-dragging.

Earlier research has emphasized greatly on proposing, developing and comparing various handoff algorithms. The performance of handoff algorithms based on relative signal strength measurements has been evaluated in [4], [5], [6], [7] and [8]. In [9] a handover decision algorithm’s performance is evaluated by means of a simulation that calculates expected number of handovers that occur as a user moves between two base stations. The performance of handover algorithms used in microcellular urban mobile systems is investigated in [10] by means of an analytical model that is able to take Frequency Hopping, modulation and coding schemes, fading and interference due to a multi-cell environment into account.

A new discrete-time approach has been introduced in [11] to analyze the performance of handoff algorithms based on pilot signal strength measurements. In [12] a local averaging technique for processing the received pilot signal strength, which can significantly improve handoff performance in cellular networks, has been proposed. In literature numerous papers have been published that have developed models to analyze the performance of handover algorithms [13] and [14].

Several handover algorithms have been proposed for improving the handoff performance. In [15], [16], [17] and [18] new handover algorithms and techniques to optimize the handoff algorithms’ performance in cellular networks have been proposed. In [19] a survey of various channel assignment schemes has been conducted to analyze their effects on the performance of the handover algorithms. [20] provides an in depth overview of implementation of handoff schemes and analysis of their performance.

For efficient utilization of the radio spectrum, a frequency reuse scheme that is consistent with the objectives of increasing capacity and minimizing interference is required [1]. Several channel assignment strategies have been developed to achieve these objectives. In [21] an aggressive channel allocation scheme to reduce call loss due to failed hand over requests and call blocking in a multiple call hand-off context has been proposed. In [22] cellular radio channel assignment problem is addressed. They proposed a new algorithm that provides better results using the modified discrete Hopfield network.

The performance of cellular mobile communication systems such as handover priority system, with overlaying macro-cell system are evaluated and compared to that of standard micro-cellular system in [23] and [24]. In [25] for the CDMA network the effects of specific mobility parameters on base station selection using handoff algorithms are examined. They examine user mobility in the context of base station selection to study the effect of mobility parameters on connectivity and transmission quality.

Previous works [26] and [27] have proposed models to configure cellular networks based on subscriber mobility between cells. In [28] they have proposed an algorithm to dynamically determine neighboring cell lists i.e. handover candidates and their associated broadcast control channels for each cell in the system. In [29] they use knowledge of the cell terrain, the mobile trajectory, and the vehicular movements in a cellular network to predict handoff rates.

In this paper the drive test field data available for our research uses proprietary handoff algorithms for the handoff phenomenon. We do not try to evaluate or compare the performance of various handoff algorithms, but evaluate how various handsets are implementing a particular algorithm. Little has been done till date to actually analyze the performance of any of such handoff algorithms for various handsets from the user's perspective. In this paper we develop a mobility tool that measures the degree to which any handset is implementing the handoff design and it evaluates the handset's performance independent of the handoff algorithm implemented by means of mapping the handoffs on the Google Maps. The focus of evaluation is on the behavior of the handset irrespective of what handoff algorithm the service is using.

2. STRATEGY

In a drive test radios are taken in a car and calls are made while driving. Data such as the latitude, longitude, color code and RSSI is captured and collected during the duration of the call. Each day the drive test team takes a fixed route performing field test and recording the data. They have technical equipment that records the data when they make calls while driving. Separate tests are conducted for both directions – clockwise and counter-clockwise. The data collected from a baseline unit (one that is generally expected to behave well) is used to prepare a model. A test unit is the product under test whose data is compared with the baseline unit.

From the drive test data we first examine the data from the baseline unit to list all the handoffs and categorize them based on the call types which are phone, dispatch and idle. Similarly we examine the drive test data for the product under test to prepare a list of all the handoffs. These handoffs are also classified and separated based on the call types. This aggregated data of handoffs is used to identify the locations where the product under test exhibits aberrant or abnormal behavior in the sense that it communicates with a base station that serves a lower RSSI when a closer base station which can serve a better RSSI is available. A good mobility decision would be to handover to the new stronger tower as a cell phone moves from a weaker tower site to a stronger tower site.

The comparison of mobility decisions made by the product under test to the baseline unit's mobility decision is realized by means of Zones. Every time a handoff occurs in the model (data from the baseline unit) from one cell site to another cell site we construct a zone. For example, when a mobility decision is taken to handoff from color code A (radio frequency associated with the sample) to color code B, a zone is created. Similarly when the handoff occurs from Color code B to Color Code A another zone is created. Mobility decisions taken by the product under test are plotted using Google Maps but due to lack of presence of model preclude statistical evaluation. However, they are available for visual inspection on the Google Maps.

For our experiment we have primarily two phones under observation – a baseline unit and a product under test. We have learning data from the baseline unit that we use to prepare models and we have test data from the product under test.

2.1 **Statistical Measure**

2.1.1 Data Preparation

We will examine the drive test data collected from a baseline unit to prepare models. Drive tests produce approximately one record every second. Each record contains latitude, longitude, color code, Signal Quality Estimation (SQE) and RSSI data captured at that instance while the phone is in the call as well as when the phone is not in a call. The color code and GPS location are extracted from each record. Fig. 1 shows the mobility of a handset between two cell sites 1 and 2 that use color code A and B respectively.

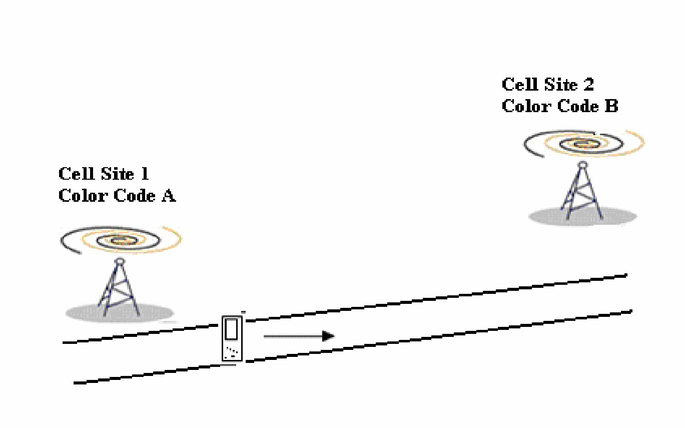


FIGURE 1 : A mobile handset moving from cell site 1 towards cell site 2

We first examine the data for the baseline unit to aggregate all the handoffs and categorize them based on the call types which are phone, dispatch and idle. Every time a handoff occurs in the model from one cell site to another cell site we construct a zone. For a handoff where the handover occurred between two cell sites such that the old and new cell sites have been recorded and a zone exists previously, then that handoff is assigned to that zone. Fig. 2 shows a portion of the route where the handoffs for the model data occurred from cell site 1 to cell site 2 such that the color code changed from A to B. Hence handoffs P, Q, R, S, T, U and V that handed-over from color code A to color code B are aggregated into one zone. In every zone using the color code and GPS data (latitude and longitude) we calculate the distance of each handoff from every other handoff. For example, the distance of P from Q, P from R, P from S, P from T, P from U and P from V is calculated and similarly the distance of each other handoff Q, R, S, T, U and V from every other handoff is calculated.

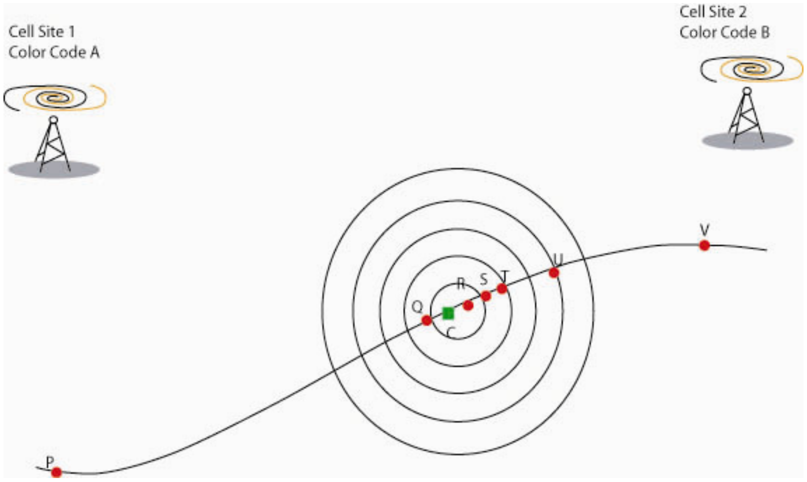


FIGURE 2 : Handoff aggregation into a Zone for the Baseline Unit

For each handoff we find the mean distance of its distance from every other handoff. After calculating the mean distance for each handoff we find the mean of all the mean distances. A standard deviation of 1.96 is calculated on the data set of means of all the means for the handoffs. Using the standard deviation we eliminate the handoffs that are outliers. An outlier is an observation that is numerically distant from the rest of the data. In Fig. 2 handoff P and V are examples of outliers. The center of the handoffs that excludes the outliers, called the center of the zone is now calculated. In Fig. 2 C is the center of the zone. We now calculate the distance of each handoff for the model from the center of the zone. For each call type of the model we calculate the

Standard Deviation for all the handoffs from the center of the zone. In Fig. 2, each concentric circle indicates the distance from the center of the zone in 0.5, 1, 1.5, 2 and 2.5 standard deviations respectively.

Similarly we examine the drive test data for the product under test for the same zone to list all the handoffs and categorize it into three different handoff lists based on the call types - phone, idle and dispatch. Fig. 3 shows a, b, c, d, e and f as the handoffs for the product under test. C1 is the center for these handoffs. Fig. 4 shows the product under test's handoffs aligned in the zone along with the baseline unit's handoffs. The distance of each handoff of the product under test from the center of the zone C is calculated. Then we use the standard deviation from the model for the model as well as the product under test so we can represent the distance of the handoffs in terms of the model. Then the distance of each product under test handoff from the center of the zone is compared to this Standard Deviation of the model. For example if 1 standard deviation from the model was equivalent to 50m and a handoff for the product under test is 91m away from the center of the zone we marked it as within 2 Standard Deviation. We do not use the standard deviation of the product under test, because we need to compare to the model and how does it perform in reference to the model. If we take the standard deviation of the product under test into consideration we will not be able to compare it to anything. By comparing to the model it compensates if the model performed poorly.

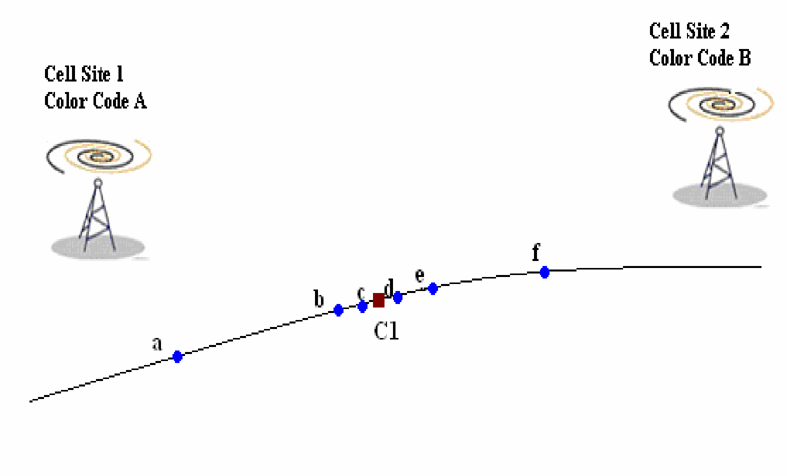


FIGURE 3: Handoffs for the Product under Test

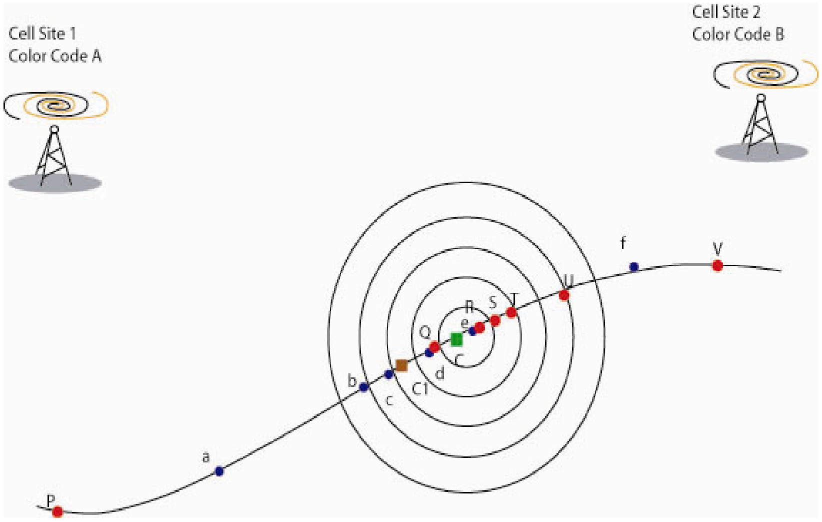


FIGURE 4: Handoffs for the Product under Test within a Zone

2.1.2 Standard Deviation as a method for comparing the behavior of product under test with baseline unit

Standard Deviation is a simple way of describing the range of a variation (usually denoted by the Greek letter sigma: σ). For our experiment we measure the deviation of the handoffs for each call type of the test unit in comparison to the handoffs for the corresponding call type in the baseline unit. Standard Deviation is the squared root of the variance. Variance (S2) is computed by taking the average squared deviation of the mean distance of all handoffs from the center of the zone (X') from the distance of individual handoff from center of the zone (Xi). For example, in a zone for the call type Idle we examine the handoffs for the model data to compute the variance and then the standard deviation. The Variance is calculated as below:

$$S^2 = \Sigma \frac{(X_i - X')^2}{n} \qquad (1)$$

Here n is the total number of handoffs for the call type Idle. Standard Deviation can now be calculated as the square-root of the variance.

For data that is “normally distributed" we expect that about 68.3% of the data will be within 1 standard deviation of the mean (i.e., in the range X' ± σ). In a normal distribution there is a relationship between the fraction of the included data and the deviation from the mean in terms of standard deviations (Table 1).

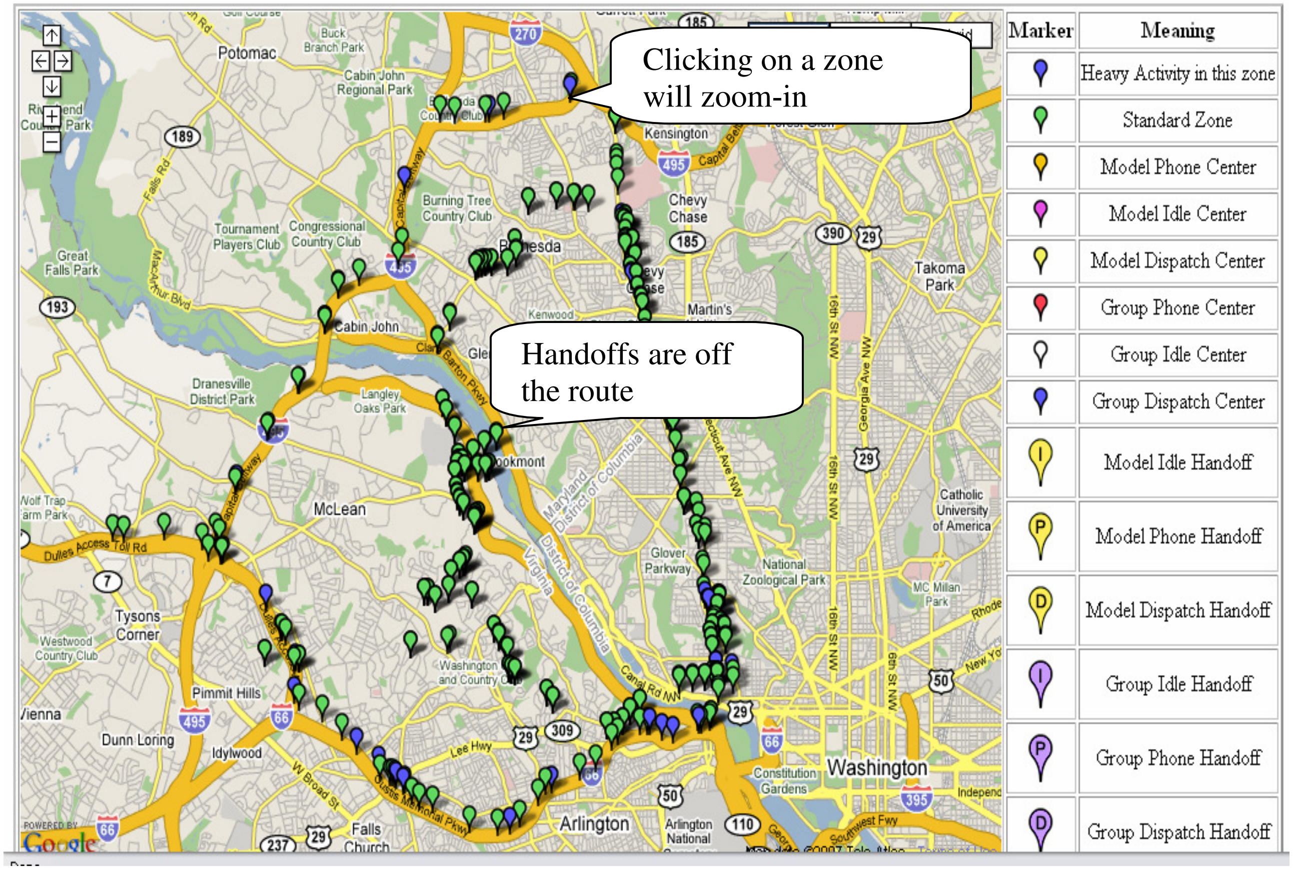
Fraction of Data	Number of Standard Deviations from Mean
50.0%	.674
68.3%	1.000
90.0%	1.645
95.0%	1.960
95.4%	2.000
98.0%	2.326
99.0%	2.576
99.7%	3.000

TABLE 1: Relationship between the fraction of the included data and the deviation from the mean in terms of standard deviations

Thus we should expect that 95% of the handoffs would be within 1.96 standard deviations of X' (i.e., in the range X' ± 1.96σ). This is called a 95% confidence interval for the sample.

3. DATA VISUALIZATION

The drive test data for the baseline unit is used to make a model. We select data from a day when the drive test team followed the route either in a clockwise or counter-clockwise direction. Similarly we select data from the product under test that we like to compare with the model for the same direction. The handoffs for the model and the product under test that have been aggregated into zones are mapped on the Google Map for visual inspection. Fig. 5 shows a map of the zones for the drive test data for the baseline unit and product under test in the counter-clockwise direction. From the map we can clearly see that a large number of handoffs occurred off the route.



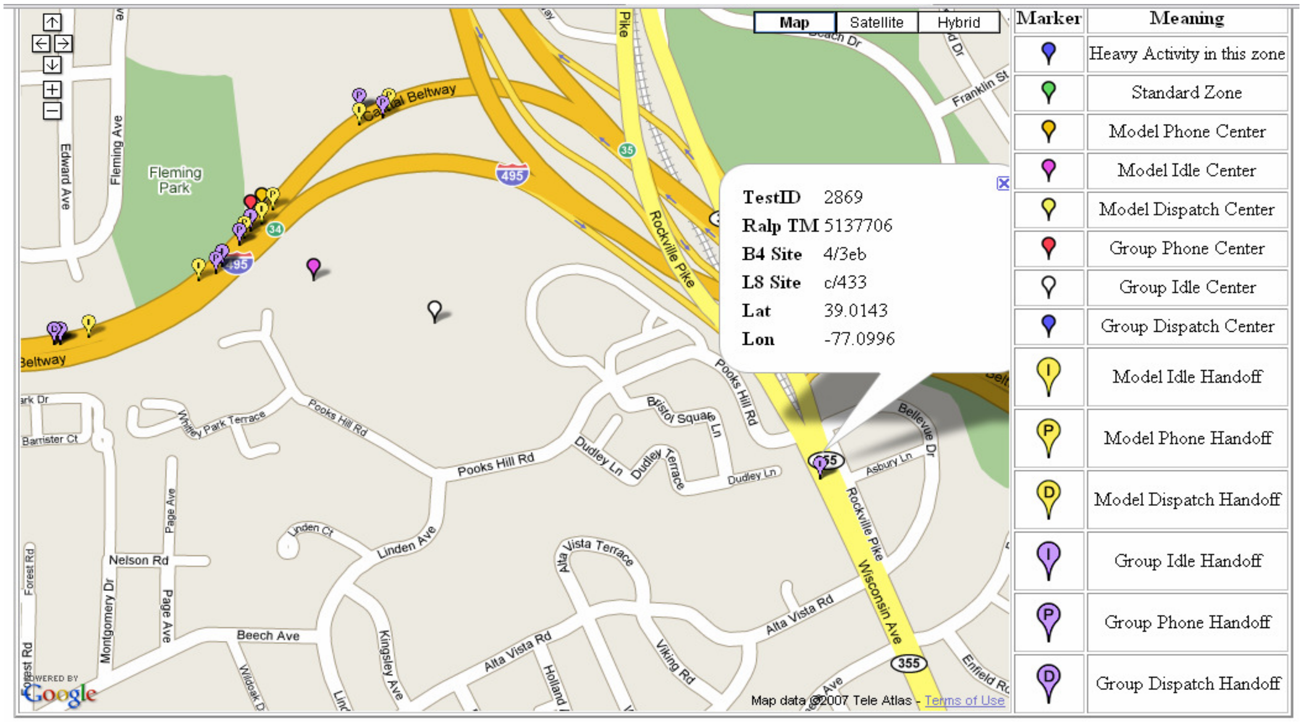
Clicking on a zone will zoom-in

Handoffs are off the route

Marker	Meaning
🔵	Heavy Activity in this zone
🟢	Standard Zone
🟠	Model Phone Center
🟣	Model Idle Center
🟡	Model Dispatch Center
🔴	Group Phone Center
⚪	Group Idle Center
🔵	Group Dispatch Center
I	Model Idle Handoff
P	Model Phone Handoff
D	Model Dispatch Handoff
I	Group Idle Handoff
P	Group Phone Handoff
D	Group Dispatch Handoff

FIGURE 5 : Handoff mapping for the baseline unit and product under test on the Google Map

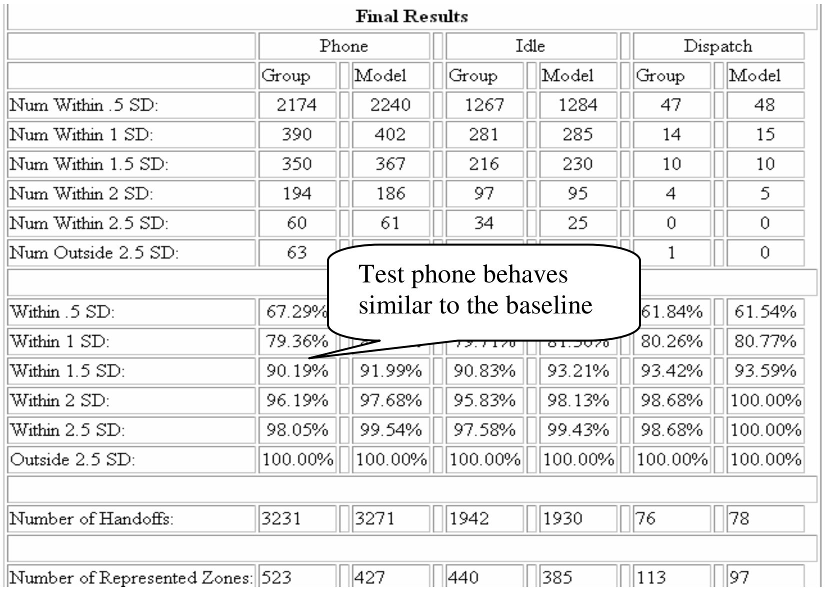
Zooming into a heavy activity zone (Fig. 6) we can clearly see that the product under test's handoffs for not all call types are aligned along the route with the baseline unit. This implies that the product under test has dissimilar behavior compared to the baseline. The idle-handoff for Test-ID 2869 is clearly not close to the model handoffs for idle call type. This implies that this handoff data requires further inspection. On further analysis we found that this data was mislabeled having counter-clockwise directionality. This drive test data was in actuality from a clock-wise route. The statistical analysis that is used to aggregate the handoffs into zones is represented along with the map (Fig. 7). The data for the product under test is indicated as Group and baseline unit data is indicated as Model.



TestID 2869
Ralp TM 5137706
B4 Site 4/3eb
L8 Site c/433
Lat 39.0143
Lon -77.0996

Marker	Meaning
●	Heavy Activity in this zone
●	Standard Zone
●	Model Phone Center
●	Model Idle Center
●	Model Dispatch Center
●	Group Phone Center
○	Group Idle Center
●	Group Dispatch Center
I	Model Idle Handoff
P	Model Phone Handoff
D	Model Dispatch Handoff
I	Group Idle Handoff
P	Group Phone Handoff
D	Group Dispatch Handoff

FIGURE 6: Zoomed in view of a Zone



Final Results

	Phone		Idle		Dispatch	
	Group	Model	Group	Model	Group	Model
Num Within .5 SD:	2174	2240	1267	1284	47	48
Num Within 1 SD:	390	402	281	285	14	15
Num Within 1.5 SD:	350	367	216	230	10	10
Num Within 2 SD:	194	186	97	95	4	5
Num Within 2.5 SD:	60	61	34	25	0	0
Num Outside 2.5 SD:	63	[illegible]	[illegible]	[illegible]	1	0
Within .5 SD:	67.29%	[illegible]	[illegible]	[illegible]	61.84%	61.54%
Within 1 SD:	79.36%	[illegible]	79.71%	81.30%	80.26%	80.77%
Within 1.5 SD:	90.19%	91.99%	90.83%	93.21%	93.42%	93.59%
Within 2 SD:	96.19%	97.68%	95.83%	98.13%	98.68%	100.00%
Within 2.5 SD:	98.05%	99.54%	97.58%	99.43%	98.68%	100.00%
Outside 2.5 SD:	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
Number of Handoffs:	3231	3271	1942	1930	76	78
Number of Represented Zones:	523	427	440	385	113	97

Test phone behaves similar to the baseline

FIGURE 7 : Final Result representation for the handoffs

The mapping of handoffs on Google Maps is very powerful in visually evaluating the data for errant handovers, cell drags, directionality and other mobility decisions. It separates the handoffs into different call types which allows for examination of each call type individually. The aggregated results generated using standard deviation provides a measure of how similar the test unit is behaving to the baseline unit.

4. CONCLUSION

The purpose of our research using data visualization of drive test data through Google Maps yielded results that helped identify aberrant mobility decisions of a product under test. This study used large volumes of real drive test data from the industry to project the behavior of the radios. Meaningful data was extracted from this enormous drive test data using data-mining and using mathematical models different call-types were analyzed for hand-over and mobility patterns. Similar work was done in [25] for CDMA networks to study the effect of mobility parameters on connectivity and transmission quality.

Other related works [26] have proposed models to configure cellular networks to study the dynamics of the mobility between single and multiple cells. In [28] they proposed an algorithm to dynamically determine neighboring cell lists i.e. handover candidates for each cell in the system. On the other hand this paper proposes a dynamic model using the data from the proprietary system iDEN (Integrated Dispatch Enhanced Network) for a baseline unit to study the existing network layout, handoff rates and mobility decisions of various call types available in this network. The Google Maps graphs were very powerful in highlighting cells drags and hand-off activity. The mobility tool developed to examine the mobility decisions of test units is technology independent and can be applied to other technologies like CDMA, GSM or WiMAX.

Unlike the previous studies which did not verify their theoretical models [27], our work utilized volumes of industrial data to suggest a model that was implemented for analyzing other products under test. In [29] they make use of knowledge of the cell terrain, the mobile trajectory, and the vehicular movements in a cellular network to predict handoff rates. However, our research is independent of the cell topography (i.e. road layout, street orientation and network layout) and is very dynamic in analyzing the mobility patterns irrespective of the network architecture and terrain layout.

Little research has been done so far to actually analyze the performance of any handoff algorithms for various handsets from the user's perspective. In this paper the drive test field data available for our research used proprietary handoff algorithms for the handover. We did not try to evaluate or compare the performance of various handoff algorithms, but evaluated how various handsets implemented a particular algorithm. We analyzed the data collected by the drive testers wherein they made calls and recorded the data pertaining to the calls almost each second. The focus was to analyze the large volume of drive test field data and develop a method that can measure the degree of adverse trends or aberrant behavior of a product under test. Data visualization of the handoffs by means of Google Maps was very effective in comparing a set of behaviors from a baseline unit and identifying a particular call that is exhibiting a different path (talking to a different cell than expected or taking longer to move to a new cell site). Hence this data visualization method was very effective in providing a method to evaluate the performance of handoffs of the mobile phones.

The mobility behavior of nodes is an important issue as discussed in [30] and in future we plan to incorporate path accumulation during the route discovery process using the Optimized-AODV protocol to attain extra routing information. Networks are being overloaded in terms of their capacity and probability of blocking being high day-by-day and in future we plan to extend our work to include a share loss analysis of internet traffic when two operators are in competition in respect of quality of service as discussed in [31].

5. REFERENCES

[1] T. S. Rapport. “The Cellular Concept- System Design Fundamentals”. Wireless Communications: Principles and Practice, 3: 57-58, 2006

[2] R. S. Saunders and N. Q. Tat. “Handoff by monitoring of received signal strengths among base stations” United States, Nokia Mobile Phones, Ltd. (Salo, Finland), U.S. Patent, Patent #: 5896570, Issue Date : April 20, 1999 http://www.patentstorm.us/patents/5896570-description.html

[3] G. Senarath, A. Abu-Dayya, and R. Matyas. “Adaptive Handoff Algorithms Using Absolute and Relative Thresholds for Cellular Mobile Communication Systems”. In the Proceedings of the 48th IEEE Conference on Vehicular Technology. May 1998, 1603-1607

[4] A.J.M. Ransom. “Handoff Considerations in Microcellular System Planning”. In the Proceedings of the 6th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 'Wireless: Merging onto the Information Superhighway'. September 1995, 804-808

[5] N. Zhang and J.M. Holtzman. “Analysis of Handoff Algorithms Using both Absolute and Relative *Measurements”.* IEEE Transactions on Vehicular Technology. 45(1): 174-179, February 1996

[6] G.N. Senarath and D. Everitt. “Controlling handoff performance using signal strength Prediction schemes & hysteresis algorithms for different shadowing environments”. In the Proceedings of the 46th IEEE Conference on Vehicular Technology, 'Mobile Technology for the Human Race'. May 1996, 1510-1514

[7] Xinrong Li. “RSS-Based Location Estimation with Unknown Path-loss Model”. IEEE Transactions on Wireless Communications. 5(12): 3626 – 3633, December 2006

[8] Ming-Hsing Chiu and M.A. Bassiouni. “Predictive schemes for handoff prioritization in cellular networks based on mobile positioning”. IEEE Journal on Selected Areas in Communications, 18(3): 510 – 522, March 2000

[9] G.P. Pollini. “Handover Rates in Cellular Systems: Towards a Closed Form Approximation”. In the Proceedings of IEEE Global Telecommunications Conference. November 1997, 711 -715

[10] M. Chiani, G. Monguzzi, R. Verdone, and A. Zanella. “Analytical Modeling of Handover Algorithm Performance in a Multi-cell Urban Environment with Frequen*cy Hopping”.* In the Proceedings of the 48th IEEE Conference on Vehicular Technology. May 1998, 1059-1063

[11] A. E. Leu and B.L. Mark. “A discrete-time approach to analyze hard handoff performance in cellular networks, IEEE Transactions on Wireless Communications”. 3(5): 1721- 1733, September 2004

[12] B.L. Mark and A.E. Leu. “Local Averaging for Fast Handoffs in Cellular Networks”. IEEE Transactions on Wireless Communications. 6(3), 866-874, March 2007

[13] P.M. Jung-Lin Pan Djuric and S.S. Rappaport. “A Simulation Model of Combined Handoff Initiation and Channel Availability in Cellular Communications”. In the Proceedings of the 46th IEEE Conference on Vehicular Technology, Mobile Technology for the Human Race. 3, May 1996, 1515-1519

[14] S. Agarwal and J.M. Holtzman. “Modeling and Analysis of Handoff Algorithms in Multi-Cellular Systems”. In the Proceedings of the 47th IEEE Conference on Vehicular Technology. 1, May 1997, 300-304

[15] N. Benvenuto and F. Santucci. “A Least Squares Path-Loss Estimation Approach to Handover Algorithms”. IEEE Transactions on Vehicular Technology. 48(2): 437-447, March 1999

[16] B.L. Lim and L.W.C. Wong. “Hierarchical Optimization of Microcellular Call Handoffs”. IEEE Transactions on Vehicular Technology. 48(2): 437-447, March 1999

[17] R. Prakash and V.V. Veeravalli. “Adaptive Hard Handoff Algorithms”. IEEE Journal on Selected Areas in Communications. 18(11): 2456-2464, November 2000

[18] M. Akar and U. Mitra. “Soft handoff algorithms for CDMA cellular networks”. IEEE Transactions on Wireless Communications. 2(6): 1259-1274, November 2003

[19] I. Katzela and M. Naghshineh. “Channel assignment schemes for cellular mobile telecommunication systems: a comprehensive survey”. IEEE Personal Communications. 3(3): 10 – 31, June 1996

[20] N.D. Tripathi, J.H. Reed and H.F. VanLandinoham. “Handoff in cellular systems”. IEEE Personal Communications. 5(6): 26-37, December 1998

[21] D. Zeghlache. “Aggressive Handover Algorithm for Mobile Networks”. In the Proceedings of the 44th IEEE Conference on Vehicular Technology. June 1994, 87-90

[22] Jae-Soo Kim, Sahng Ho Park, P.W. Dowd and N.M. Nasrabadi. “Cellular radio channel assignment using a modified Hopfield network”. IEEE Transactions on Vehicular Technology. 46(4): 957-967, November 1997

[23] M. Inoue, H. Morikawa and M. Mizumachi. “Performance Analysis of Microcellular Mobile Communication System”. In the Proceedings of the 44th IEEE Conference on Vehicular Technology. June 1994, 135 – 139

[24] H. Viswanathan and S. Mukherjee. “Performance of cellular networks with relays and centralized scheduling”. IEEE Transactions on Wireless Communications, 4(5): 2318- 2328, September 2005

[25] S. Sharma and B. Jabbari. “Mobility Effects on Base Station Selection in Wireless CDMA Networks”. In the Proceedings of the 60th IEEE Conference on Vehicular Technology. September 2004, 4310 – 4314

[26] B. Gavish and S. Sridhar. “The Impact of Mobility on Cellular Network Configuration”. Springer Wireless Networks. 7(2): 173 – 185, March 2001

[27] R. Sankar and N. Savkoor. “A Combined Prediction System for Handoffs in Overlaid Wireless Networks”. In the Proceedings of the IEEE International Conference on Communications. June 1999, 760 – 764

[28] S. Magnusson and H. Olofsson. “Dynamic neighbor cell list planning in a micro cellular network”. In the Proceedings of the 6th IEEE International Conference on Universal Personal Communications Publication. October 1997, 223-227

[29] D. Bansal, A. Chandra, R. Shorey, A. Kulshreshtha and M. Gupta. “Mobility models for cellular systems: cell topography and handoff probability”. In the Proceedings of the 49th IEEE Conference on Vehicular Technology. July 1999, 1794 – 1798

[30] A. Goel and A. Sharma. “Performance Analysis of Mobile Ad-hoc Network Using AODV Protocol”. International Journal of Computer Science and Security (IJCSS), 3(5): 334-343, 2009

[31] D. Shukla, at el. “Share Loss Analysis of Internet Traffic Distribution in Computer Networks”. International Journal of Computer Science and Security (IJCSS), 3(5): 414-427, 2009

A Lexisearch Algorithm for the Bottleneck Traveling Salesman Problem

Zakir H. Ahmed **zhahmed@gmail.com**
Department of Computer Science,
Al-Imam Muhammad Ibn Saud Islamic University,
P.O. Box No. 5701, Riyadh-11432
Kingdom of Saudi Arabia

Abstract

The bottleneck traveling salesman problem (BTSP) is a variation of the well-known traveling salesman problem in which the objective is to minimize the maximum lap (arc length) in a tour of the salesman. In this paper, a lexisearch algorithm using adjacency representation for a tour has been developed for obtaining exact optimal solution to the problem. Then a comparative study has been carried out to show the efficiency of the algorithm as against an existing exact algorithm for some TSPLIB and randomly generated instances of different sizes.

Keywords: Bottleneck traveling salesman, Lexisearch, Bound, Alphabet table.

1. INTRODUCTION

The bottleneck traveling salesman problem (BTSP) is a variation of the benchmark traveling salesman problem (TSP). It can be defined as follows:

A network with n nodes (or cities), with 'node 1' (suppose) as 'headquarters' and a cost (or distance, or time etc.) matrix C=[c_{ij}] of order n associated with ordered node pairs (i,j) is given. Let {1=α_0, α_1, α_2,....,α_{n-1} , α_n=1} ≡ {1→α_1→α_2→..... →α_{n-1}→1} be a tour, representing irreducible permutations interpreted as simple cycle. The tour value is defined as $\max \left\{ c_{\alpha_i, \alpha_{i+1}} : i = 0, 1, 2, ..., n-1 \right\}$. The objective is to choose a tour which has minimum tour value.

Both TSP and BTSP are well known NP-hard problems. Vairaktarakis [1] considered a polynomially solvable TSP and showed that the corresponding BTSP is strongly NP-complete. The BTSP finds application in the area of workforce planning. A commonly used objective in workforce leveling (or range) is to minimize the difference between the maximum and minimum number of workers required by any worker schedule. The objective leads to level worker schedules that smooth the workforce fluctuations from one production period to the next. Such schedules are particularly useful in automobile assembly because they help to preserve overall smoothing of operations [1]. Another application of the BTSP is in minimizing makespan in a two-machine flowshop with no-wait-in-process which is a building block for more general no-wait production system [2].

Gilmore and Gomory [3] introduced the BTSP, and discussed a specific case of the problem. Definitely, the BTSP has not been as well researched as the TSP. There are a few exact algorithms available in the literature for the BTSP. An algorithm based on branch and bound (BB) is developed by Garfinkel and Gilbert [4] for solving the general BTSP, and discussed an application of the problem in the context of machine scheduling. Carpaneto et al. [5] also developed an algorithm based on BB that uses a heuristic search to find a Hamiltonian circuit containing only arcs whose cost is not greater than the current lower bound. Ramesh [6] reported this problem as min-max TSP, and developed a lexisearch algorithm, using path representation for a tour of the salesman, to obtain exact optimal solution to the problem, and computational experiments were reported for the randomly generated problems of sizes up to 30.

There are some heuristic algorithms in the literature which are reported to be good for the general BTSP [7, 8, 9, 10]. Also, there are some algorithms in the literature which were developed for some special case of the problem [2, 11]. In this paper, we are not considering any special case of the problem, rather the general BTSP. A lexisearch algorithm, using adjacency representation for a tour of the salesman, is developed to obtain exact optimal solution to the problem. Finally, the efficiency of our algorithm is compared with the algorithm of Ramesh [6] for some TSPLIB and randomly generated instances of different sizes.

This paper is organized as follows: Section 2 presents some definitions that are required for the lexisearch algorithm. A lexisearch algorithm with an illustrative example is presented in Section 3. Computational experiments for two algorithms have been reported in Section 4. Finally, Section 5 presents comments and concluding remarks.

2. SOME DEFINITIONS

2.1. Alphabet table

Alphabet matrix, A=[a(i,j)], is a square matrix of order n formed by the positions of the elements of the cost matrix of order n, C=[c_{ij}]. The ith row of the matrix A consists of the positions of the elements in the ith row of the matrix C when they are arranged in the non-decreasing order of their values. If a(i,p) stands for the pth element in the ith row of A, then a(i,1) corresponds to the smallest element in the ith row of the matrix C. That is,

$$\min{}_i [c_{ij}] = c_{i,a(i,1)}.\ So,\ if\ \ p < q,\ then\ \ c_{i,a(i,p)} \le c_{i,a(i,q)}.$$

Thus, the ith row of A is [a(i,1), a(i,2), …., a(i,n)]. Clearly,

$$c_{i,a(i,1)} \le c_{i,a(i,2)} \le \dots\dots\dots \le c_{i,a(i,n)}$$

The words can be generated by considering one element in each row as follows:

$$1 \rightarrow \{a(1,j) = \alpha_1\} \rightarrow \{a(\alpha_1, k) = \alpha_2\} \rightarrow \dots\dots \rightarrow \alpha_{n-2} \rightarrow \{a(\alpha_{n-2}, m) = \alpha_{n-1}\} \rightarrow \{\alpha_n = 1\}$$

where j, k,…, m are some indices in the alphabet matrix.

Alphabet table "$[a(i,j) - c_{i,a(i,j)}]$" is the combination of elements of matrix A and their values. For example, a cost matrix and its 'alphabet table' are shown in Table 1 and Table 2 respectively.

Node	1	2	3	4	5	6	7
1	999	75	99	9	35	63	8
2	51	999	86	46	88	29	20
3	100	5	999	16	28	35	28
4	20	45	11	999	59	53	49
5	86	63	33	65	999	76	72
6	36	53	89	31	21	999	52
7	58	31	43	67	52	60	999

TABLE 1: The cost matrix.

Node	N - V	N - V	N - V	N - V	N - V	N - V	N - V
1	7-8	4-9	5-35	6-63	2-75	3-99	1-999
2	7-20	6-29	4-46	1-51	3-86	5-88	2-999
3	2-5	4-16	5-28	7-28	6-35	1-100	3-999
4	3-11	1-20	2-45	7-49	6-53	5-59	4-999
5	3-33	2-63	4-65	7-72	6-76	1-86	5-999
6	5-21	4-31	1-36	7-52	2-53	3-89	6-999
7	2-31	3-43	5-52	1-58	6-60	4-67	7-999

TABLE 2: The alphabet table (N is the label of node, and V is the value of the node).

2.2. Incomplete word and block of words

$W = (\alpha_0, \alpha_1, \alpha_2, ...\alpha_m), m < n,$ represents an incomplete word. An incomplete word (partial tour) consists of some of the nodes. Incomplete word represents the block of words with this incomplete word as the leader of the block. If F(.) is the objective function and *W* is an incomplete word, then for a complete word *S* whose leader is *W*, we have *F(S) ≥ F(W).*

For the BTSP, each node is considered as a letter in an alphabet and each tour can be represented as a word with this alphabet. Thus the entire set of words in this dictionary (namely, the set of solutions) is partitioned into blocks. A block B with a leader (α_0, α_1, α_2,) of length three consists of all words beginning with (α_0, α_1, α_2,) as string of first three letters. The block A with the leader (α_0, α_1) of length 2 is the immediate superblock of B and includes B as one of its sub-blocks. The block C with leader (α_0, α_1, α_2, β) is a sub-block of block B. The block B consists of many sub-blocks (α_0, α_1, α_2, β_k), one for each $\beta_{k.}$ The block B is the immediate super-block of block C.

By structure of the problem it is often possible to get lower bound for the block to the values of all words in a block by examining its leader. Hence, by comparing the bound with the 'best solution value' found so far, one can

(i) ‘go' into the sub-block by concatenating the present leader with an appropriate letter; if the block-bound is less than the 'best solution value',

(ii) ‘jump over’ to the next block; if no word in the block can be better in value than the 'best solution value', or

(iii) ‘jump out’ to the next super-block, if the current block, which is to be jumped over, is the last block of the present superblock.

Further, if value of the current leader is already greater than or equal to the 'best solution value' found so far, then no need for checking subsequent blocks within this super-block, and we 'jump out' to the next supper-block.

Let a, b, c, d be the four nodes in a network. The words starting with ‘a’ constitute a ‘block’ with ‘a’ as its leader. In a block, there can be many sub-blocks; for instance ‘ab’, ‘ac’ and ‘ad’ are leaders of the sub-blocks of block ‘a’. There could be blocks with only one word; for instance, the block with leader ‘abd’ has only one word ‘abdc’. All the incomplete words can be used as leaders to define blocks. For each of blocks with leader ‘ab’, ‘ac’ and ‘ad’, the block with leader ‘a’ is the immediate super-block. For example, 'go' into the sub-block for ‘db’ leads to ‘dba’ as augmented leader, 'jump over' the block for ‘abc’ is ‘abd’, and 'jump out' to the next higher order block for ‘cdbe’ is ‘cde’.

3. A LEXISEARCH ALGORITHM FOR THE BTSP

The lexicographic search derives its name from lexicography, the science of effective storage and retrieval of information. This search (lexisearch, for short) is a systematic branch and bound approach, was developed by Pandit [12], which may be summarized as follows:

The set of all possible 'solutions' to a combinatorial optimization problem is arranged in hierarchy- like words in a dictionary, such that each 'incomplete word' represents the block of words with this incomplete word as the 'leader'. Bounds are computed for the values of the objective function over these blocks of words. These are compared with the 'best solution value'. If no word in the block can be better than the 'best solution value', jump over the block to the next one. However, if the bound indicates a possibility of better solutions in the block, enter into the sub-block by concatenating the present leader with appropriate 'letter' and set a bound for the new (sub) block so obtained.

This procedure is very much like looking for a word in a dictionary; hence the name 'lexi(cographic) search'. The basic difference with the branch and bound approach is that lexisearch approach is one-pass, implicitly exhaustive, search approach, avoiding the need for book-keeping involved in storing, in active memory, the bounds, at various branching nodes at various levels and related backtracking procedures, which can be expensive in terms of memory space and computing times.

There are mainly two ways of representing salesman's path in the context of lexisearch. For example, let {1, 2, 3, 4, 5} be the labels of nodes in a 5 node instance and let path to be represented be {1→3→4→2→5→1}. Adjacency representation of this path is usual representation of corresponding permutation, namely, $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{pmatrix}$, indicating that the edges 1→3, 2→5, …., 5→1 constitute the tour. The path representation just lists the sequence of the tour as (1, 3, 4, 2, 5). The following subsections discuss the lexisearch algorithm by considering adjacency representation for solving the BTSP and its illustration through an example.

3.1. The algorithm

Ramesh [6] used path representation for a tour to obtain exact optimal solution to the problem. As reported, the algorithm shows a large variation in solution times. So, we present another lexisearch algorithm using adjacency representation for a tour. A preliminary version of this algorithm is reported in Ahmed [8]. The algorithm is presented below:

Let C=[c_{ij}] be the given n x n cost matrix and c_{ij} be the cost of visiting of node j from node i, and let 'node 1' be the starting node.

Step 0: - Form the 'alphabet table'. Initialize the 'best solution value' to a large number, and set l = 1.

Step 1: - With the partial tour of length (l -1) take as leader; consider the first 'legitimate and unchecked' node. Compute the lower bound as discussed in section 3.2, and go to step 2. If there is no any 'legitimate and unchecked' node, go to step 5.

Step 2: - If the lower bound is less than the 'best solution value', go to step 3, else go to step 5.

Step 3: - If there is a sub-tour, go to step 1, else go to step 4.

Step 4: - Go to sub-block, i.e., augment the current leader; concatenate the considered node to it, lengthening the leader by one node, and compute the current tour value. If the current tour is a complete tour, then replace the 'best solution value' with the current solution value, and go to step 5. If the current tour is not a complete tour, then go to step 1.

Step 5: - Jump this block, i.e., decrement l by 1 (one), rejecting all the subsequent tours from this block. If l<1, go to step 6, else go to step 1.

Step 6: - Current tour gives the optimal tour sequence, with 'best solution value' as the optimal cost, and stop.

3.2. Lower bound

The objective of lower bound is to skip as many subproblems in the search procedure as possible. A subproblem is skipped if its lower bound exceeds the 'best solution value' found so far in the process. The higher the lower bound the larger the set of subproblems that are skipped. Some algorithms in the literature calculate overall lower bound for the BTSP instance and develop algorithms based on relaxation and subtour elimination scheme [4, 5, 9]. Our lexisearch algorithms do not follow this way for solving the BTSP instances. In our algorithm, we are not setting lower bound for an instance, rather setting lower bound for each leader on the value of objective function for the instance as follows:

Suppose the present permutation for the partial tour is $\begin{pmatrix} 1 & 2 & 3 \\ \alpha_1 & \alpha_2 & \alpha_3 \end{pmatrix}$ and the node α_4 is selected for concatenation. Before concatenation, we check the bound for the leader $\begin{pmatrix} 1 & 2 & 3 & 4 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix}$. For that, we start our computation from 5th row of the 'alphabet table' and traverse up to the nth row, check the value of first 'legitimate' node (the node that is not present in the partial tour) in each row. Maximum among the values of first 'legitimate' nodes and the leader value is the lower bound for the leader $\begin{pmatrix} 1 & 2 & 3 & 4 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{pmatrix}$.

3.3. Illustration

Working of the above algorithm is explained through an example of the seven-node instance given in Table-1. Table 3 gives the ‘search table’. The symbols used therein are listed below:

GS: Go into the sub-block, i.e., attach the first ‘free’ letter to the current leader.

JB: Jump over the block, i.e., go to the next block of the same order i.e., replace the last letter of the current block by the letter next to it in the alphabet table.

JO: Jump out to the next, higher order block, i.e., drop out the last letter of the current leader and then jump the block.

BS: Best solution value.

ST: Sub-tour.

As illustration of the example, we consider BS = 9999 and 'partial tour value (Sol)' = 0. We start from 1st row of the 'alphabet table'. Here, a(1,1) = 7 with 'present node value (Val)' = c_{17} = 8. Since Max{Sol, Val} = 8 < BS, we go for bound calculation for the present leader $\begin{pmatrix} 1 \\ 7 \end{pmatrix}$. The bound will guide us whether the node 7 will be accepted or not.

$$\begin{aligned} Bound &= Max\ \{Sol\ ,Val\ ,c_{2,a(2,4)}, c_{3,a(3,1)}, c_{4,a(4,1)}, c_{5,a(5,1)}, c_{6,a(6,1)}, c_{7,a(7,1)}\} \\ &= Max\ \{0, 8, c_{2,6}, c_{3,2}, c_{4,3}, c_{5,3}, c_{6,5}, c_{7,2}\} \\ &= Max\ \{0, 8, 29, 5, 11, 33, 21, 31\} = 33 \end{aligned}$$

Since Bound<BS, we accept the node 7 that leads to the partial permutation $\begin{pmatrix} 1 \\ 7 \end{pmatrix}$ with Sol=8. Next we go to 2th row of the 'alphabet table'. Since a(2,1) = 7 is repeated, we consider the next element of the row, i.e., a(2,2) =6 with Val = c_{26} = 29. Since Max{Sol, Val} = 29 < BS, we go for bound calculation for the present leader $\begin{pmatrix} 1 & 2 \\ 7 & 6 \end{pmatrix}$.

Leaders							Bound	Best Solution Value	Remarks
1	2	3	4	5	6	7			

7-8							33	9999	GS
	6-29						33	9999	GS
		2-5					33	9999	GS
			3-11				65	9999	GS
				4-65			65	9999	GS
					5-21		65	9999	ST
					1-36		65	9999	GS
						5-52	65	9999	GS
							BS =	65	JB, JO
			1-20				43	65	GS
				3-33			52	65	GS
					5-21		67	65	ST
					4-31		52	65	GS
						5-52	52	65	GS
							BS =	52	JB, JO
			5-59				59	52	JO
		4-16					33	52	GS
			3-11				63	52	ST
			1-20				33	52	GS
				3-33			33	52	GS
					5-21		33	52	GS
						2-31	33	52	GS
							BS =	33	JB, JO
			2-45				45	33	JO
		5-28					33	33	JB
		6-35					35	33	JO
	4-46						46	33	JO
4-9							33	33	JB
5-35							35	33	STOP

TABLE 3: The search table.

$$\begin{aligned}
Bound &= Max\ \{Sol\ ,Val\ ,c_{3,a(3,1)}, c_{4,a(4,1)}, c_{5,a(5,1)}, c_{6,a(6,1)}, c_{7,a(7,1)}\} \\
&= Max\ \{8,\ 29\ ,\ c_{3,2},\ c_{4,3},\ c_{5,3},\ c_{6,5},\ c_{7,2}\} \\
&= Max\ \{8,\ 29\ ,\ 5,\ 11\ ,\ 33\ ,\ 21\ ,\ 31\ \}\ =\ 33
\end{aligned}$$

Since Bound < BS, we accept the node 6 that leads to the partial permutation $\begin{pmatrix} 1 & 2 \\ 7 & 6 \end{pmatrix}$ with Sol=29. Proceeding in this way, we obtain the 1st complete permutation $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 6 & 2 & 3 & 4 & 1 & 5 \end{pmatrix}$ for the tour {1→7→5→4→3→2→6→1} with Sol= 65. Since Sol<BS, so we replace BS = 65. Now, we jump out to the next higher order block, i.e., $\begin{pmatrix} 1 & 2 & 3 \\ 7 & 6 & 2 \end{pmatrix}$ with Sol = 29, and try to compute another complete tour with lesser tour value. Proceeding in this way, we obtain the optimal tour {1→7→2→6→5→3→4→1} that is given by the permutation $\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 7 & 6 & 4 & 1 & 3 & 5 & 2 \end{pmatrix}$ with optimal solution value = 33.

4. COMPUTATIONAL EXPERIMENT

Our lexisearch algorithm (LSA) has been encoded in Visual C++ on a Pentium 4 personal computer with speed 3 GHz and 448 MB RAM under MS Windows XP. Also, for the comparison lexisearch algorithm by Ramesh [6], named as RA, is encoded and run in the same environment. Both the algorithms (RA and LSA) are tested for some TSPLIB instances and randomly generated instances of different sizes drawn from different uniform distribution of integers.

Table 4 gives the results for nine asymmetric TSPLIB instances of size from 17 to 70. We report optimal solution values and solution times (in second) for solving the instances by both RA and LSA. To the best of our knowledge, no literature presents experimental solutions to the asymmetric TSPLIB instances. The instance br17 of size 17 could be solved within only 1.59 seconds by RA, which could be solved by LSA within 185.99 seconds. For the remaining instances, reported in the table, LSA is found to be better. We do not report the solution of other asymmetric TSPLIB instances, because we could not solve them by any algorithm within one hour. Table 4 also reports the computational time when the optimal solution is seen for the first time. In fact, a lexisearch algorithm first finds an optimal solution and then proves the optimality of that solution, i.e., all the remaining subproblems are discarded. Table 4 shows that, on average solution time, RA found optimal solution within 41% of the total solution time, whereas LSA found the optimal solution within only 6% of the total solution time. That is, RA spent 59% and LSA spent 94% of total time on proving optimality of the solutions. Therefore, for these asymmetric TSPLIB instances, RA spends a relatively large amount of time on finding an optimal solution compared to our LSA, and hence, a small number of subproblems are thrown by RA.

Instances	n	Optimal Solution	Solution time		Solution is seen first	
			RA	LSA	RA	LSA
br17	17	8	1.59	185.99	0.39	0.00
ftv33	34	113	0.16	1.32	0.16	0.55
ftv35	36	113	0.34	0.48	0.34	0.42
ftv38	39	113	0.02	0.05	0.02	0.05
p43	43	5008	0.02	0.02	0.05	0.02
ftv44	45	113	57.88	49.13	57.88	40.23
ft53	53	977	2960.56	1970.3	0.00	0.00
ftv64	65	104	812.03	0.00	789.23	0.00
ft70	70	1398	1452.36	156.45	1278.93	94.32
Mean			**587.22**	**262.64**	**236.33**	**15.07**

TABLE 4: Solution times for asymmetric TSPLIB instances

Instances	n	Optimal Solution	Solution time		Solution is seen first	
			RA	LSA	RA	LSA
burma14	14	418	0.08	0.03	0.01	0.00
ulysses16	16	1504	15.21	0.03	0.00	0.00
gr17	17	282	105.36	75.98	0.00	0.02
gr21	21	355	293.21	315.06	0.00	0.02
ulysses22	22	1504	325.09	270.13	0.00	0.00
gr24	24	108	121.56	29.08	78.88	0.03
fri26	26	93	87.98	70.32	14.05	8.94
bayg29	29	111	2972.05	108.06	2089.32	0.01
bays29	29	154	2145.32	75.32	0.00	0.00
swiss42	42	67	**----**	2148.5	-----	40.17
bier127	127	7486	**----**	3541.2	0.03	0.01
Mean			**673.98**	**603.06**	**218.23**	**4.47**

TABLE 5: Solution times for symmetric TSPLIB instances

Table 5 gives the results for eleven symmetric TSPLIB instances of size from 14 to 127. Recently, Ramakrishnan et al. [9] developed a very good heuristic algorithm for the general BTSP and reported results for some symmetric TSPLIB instances only. Since, the nature of our algorithm is not same as their algorithm; we do not to carry out comparison with the algorithm in terms of solution times. However, solutions reported there are found to be same as our solutions. Out of eleven instances two instances could not be solved within one hour by RA. Of course, we saw the optimal solution for one of them within 0.03 seconds. On the basis of average solution times, the table concludes that LSA is better than RA. It is to be noted that while we calculate average solution time, we do not consider the instances which were not solved optimally within one hour. For these symmetric TSPLIB instances also, RA spends a relatively large amount of time on finding an optimal solution compared to LSA. For these case also, we do not report the instances which could not be solved by any algorithm within one hour.

Randomly generated asymmetric and symmetric instances of different sizes are drawn from uniform distribution of integers in the intervals [1, 100] and [1, 10000]. Fifteen different instances were generated for each size. Table 6 reports mean and standard deviation of solution times by RA and LSA for asymmetric instances. On the basis of the average solution times and standard deviation, Table 6 shows that LSA is better than RA for both intervals. Of course, instances generated from the interval [1, 10000] are found to be more difficult than the instances generated from [1, 100].

We report mean and standard deviation of solution times by RA and LSA for symmetric instances in the Table 7. For these instances also, LSA is found to be better than RA. Also, the instances generated from [1, 10000] are found to be more difficult than the instances generated from [1, 100]. It is also observed that symmetric instances are more difficult than the asymmetric instances.

n	$1 \le c_{ij} \le 100$				$1 \le c_{ij} \le 10000$			
	RA		**LSA**		**RA**		**LSA**	
	Mean	**Std dev**	**Mean**	**Std dev**	**Mean**	**Std dev**	**Mean**	**Std dev**
30	24.58	39.33	10.12	25.85	27.18	40.32	13.25	29.18
35	141.93	198.96	65.06	119.38	158.21	201.15	75.06	146.16
40	457.95	578.08	163.28	319.3	495.92	518.15	196.86	347.01
45	735.14	952.32	275.03	502.32	802.21	1103.87	289.50	562.23
50	959.21	1014.23	317.92	516.39	967.32	1201.14	321.15	596.27

TABLE 6: Solution times for random asymmetric instances.

n	$1 \le c_{ij} \le 100$				$1 \le c_{ij} \le 10000$			
	RA		**LSA**		**RA**		**LSA**	
	Mean	**Std dev**	**Mean**	**Std dev**	**Mean**	**Std dev**	**Mean**	**Std dev**
30	29.58	43.12	15.22	20.15	32.16	47.52	14.99	21.35
35	189.21	209.56	76.98	101.76	190.46	229.13	79.32	98.21
40	507.19	618.54	182.35	357.97	535.78	761.76	166.28	375.21
45	805.98	987.32	352.67	547.12	854.36	1087.25	398.52	601.24
50	1020.01	1321.45	573.87	602.95	1223.01	1532.78	1052.32	1524.21

TABLE 7: Solution times for random symmetric instances.

5. CONCLUSION & FUTURE WORK

We presented a lexisearch algorithm using adjacency representation method for a tour for the bottleneck traveling salesman problem to obtain exact optimal solution to the problem. The performance of our algorithm is compared with the lexisearch algorithm of Ramesh [6] for some TSPLIB instances and two types of randomly generated instances of different sizes. The computational experiment shows that our lexisearch algorithm is better. Between asymmetric and symmetric TSPLIB as well as random instances, symmetric instances are found to be hard.

In this present study, it is very difficult to say that what moderate sized instance is unsolvable by our lexisearch algorithm, because, for example, br17 of size 17 could be solved within 185.99 seconds and dantzig42 of size 42 could not be solved within one hour, whereas, ftv64 of size 65 could be solved within only 0.00 seconds by our algorithm. It certainly depends upon the structure of the instance. So a closer look at the structure of the instances and then developing a data-guided module may further reduce the solution time. Also, it is seen that the optimal solution is seen for the first time very quickly, which suggests that applying a tight lower bound method may reduce the solution time drastically, which are under our investigations.

Acknowledgements

The author wishes to acknowledge Prof. S.N. Narahari Pandit, Hyderabad, India for his valuable suggestions and moral support. The author also thanks the reviewer for his constructive suggestions.

6. REFERENCES

[1] G.L. Vairaktarakis. "*On Gilmore-Gomory's open question for the bottleneck TSP*". Operations Research Letters 31, pp. 483–491, 2003.

[2] M.-Y. Kao and M. Sanghi. *"An approximation algorithm for a bottleneck traveling salesman problem"*. Journal of Discrete Algorithms, doi: 10.1016/j.jda.2008.11.007 (2009)

[3] P.C. Gilmore and R.E. Gomory. *"Sequencing a one state-variable machine: a solvable case of the traveling salesman problem"*. Operations Research 12, pp. 655–679, 1964.

[4] R.S. Garfinkel and K.C. Gilbert. *"The bottleneck traveling salesman problem: Algorithms and probabilistic analysis"*. Journal of ACM 25, pp. 435-448, 1978.

[5] G. Carpaneto, S. Martello and P. Toth. *"An algorithm for the bottleneck traveling salesman problems"*. Operations Research 2, pp. 380-389, 1984.

[6] M. Ramesh. *"A lexisearch approach to some combinatorial programming problems"*. PhD Thesis, University of Hyderabad, India, 1997.

[7] Z.H. Ahmed, S.N.N. Pandit, and M. Borah. *"Genetic Algorithms for the Min-Max Travelling Salesman Problem"*. Proceedings of the Annual Technical Session, Assam Science Society, India, pp. 64-71, 1999.

[8] Z.H. Ahmed. "*A sequential constructive sampling and related approaches to combinatorial optimization*". PhD Thesis, Tezpur University, India, 2000.

[9] R. Ramakrishnan, P. Sharma and A.P. Punnen. *"An efficient heuristic algorithm for the bottleneck traveling salesman problem".* Opsearch 46(3), pp.275-288, 2009.

[10] J. Larusic, A.P. Punnen and E. Aubanel. *"Experimental analysis of heuristics for the bottleneck traveling salesman problem".* Technical report, Simon Fraster University, Canada, 2010.

[11] J.M. Phillips, A.P. Punnen and S.N. Kabadi. "*A linear time algorithm for the bottleneck traveling salesman problem on a Halin graph*". Information Processing Letters 67, pp. 105-110, 1998.

[12] S.N.N. Pandit. "*Some quantitative combinatorial search problems*". PhD Thesis, Indian Institute of Technology, Kharagpur, India, 1963.

[13] TSPLIB, http://www.iwr.uni-heidelberg.de/iwr/comopt/software/TSPLIB95/

An Analysis Of Fraudulence In Fuzzy Commitment Scheme With Trusted Party

D.B.Ojha* ojhdb@yahoo.co.in
Deparment of Mathematics, R.K.G.I.T. , Ghaziabad (INDIA),
Ajay Sharma ajaypulast@rediffmail.com
Department of Information Technology, R.K.G.I.T., Ghaziabad (INDIA)
Ramveer Singh ramveersingh_rana@yahoo.co.in
Department of Information Technology, R.K.G.I.T., Ghaziabad (INDIA)
Shree Garg garg.shree@rediffmail.com
Department of Mathematics, S.I.E.T. , Saharanpur (INDIA)
Awakash Mishra awakashmishra@gmail.com
Deparment of M.C.A., R.K.G.E.C., Ghaziabad (INDIA)
Abhishek Dwivedi dwivediabhi@gmail.com
Deparment of M.C.A., R.K.G.E.C., Ghaziabad (INDIA)

Abstract

This paper attempt has been made to elaborate the possible cases of dishonesty between the two communicating parties under fuzzy commitment scheme. However there could be many instances where the transmission involves complete security, even if it contains errors arising purely because of the factors over which either sender or receiver have complete control. The concept itself is illustrated with the help of simple situations.

Keywords: Cryptography, Error correcting code, Fuzzy logic and commitment scheme, Error correction, Honesty.

1. INTRODUCTION:

Commitment schemes are an essentials ingredient of many cryptographic protocols. Commitments schemes are the process in which the interest of the party involves in a process are safeguarded and the process itself is made as fair as possible. Parties which perform according to the prescribed rules and aimed to achieve the protocol objective are called 'honest' [1]. Fuzzy commitment scheme was firstly introduced by Juels and Martin, fuzziness was introduced later for generating cryptography key [2, 3, 4].

The impression of commitment scheme is indispensable for the construction of modern cryptographic protocols. Since security violation is usual phenomena hence the need of commitment scheme in cryptographic protocol cannot be ruled out. Now a days, dishonesty between communicating parties emerges as salient problem. The vital role of 'fuzzy decision making' under fuzzy commitment scheme makes assure about appropriateness of communication between two parties, even after this assurance dishonesty may play their role.

In this paper, we elaborate possible cases that are the treacherous role of communicating parties. The organization of the paper is as follows: Section 2 gives some definitions and notation that will be used in the sequel, Crisp commitment scheme, Hamming distance, error correction function, measurement of nearness, fuzzy membership function, Commitment scheme, Fuzzy Commitment scheme and fuzzy decision making. In section 3, we analyze here, three possible cases in commitment scheme with trusted party.

2. PRELIMINARIES:

2.1. CRISP COMMITMENT SCHEMES:

In a commitment scheme, one party A (sender) aim to entrust a concealed message ‘m’ to the second party B (receiver), intuitively a commitment scheme may be seen as the digital equivalent of a sealed envelope. If A wants to commit a message ‘m’, he just puts it into the sealed envelope, so that whenever A wants to reveal the message to B, A facilitate to open the envelope. First of all the digital envelope should hide the message from B and it should be able to learn ‘m’ from the commitment. Second, the digital envelope should be bind, which means that A cannot change his mind about ‘m’, and by checking the opening of the commitment one can verify that the obtained value is actually the one A had in mind originally[5].

2.2 Definition: Let $C\{0,1\}^n$ be a code set which consists of a set of code words c_i of length n. The distance metric between any two code words c_i and c_j in C is defined by $dist(c_i, c_j) = \sum_{r=1}^{n} |c_{ir} - c_{jr}| \qquad c_i, c_j \in C$

This is known as Hamming distance [6].

2.3 Definition: An error correction function f for a code C is defined as $f(c_i) = \{c_j \, / \, dist(c_i, c_j) \text{ is the minimum, over } C - \{c_i\}\}$. Here, $c_j = f(c_i)$ is called the nearest neighbor of c_i [3].

2.4 Definition: The measurement of nearness between two code words c and c' is defined by $\text{nearness } (c, c') = dist(c, c') / n$, it is obvious that $0 \leq \text{nearness } (\text{c}, \text{c}') \leq 1$ [3].

2.5 Definition: The fuzzy membership function for a codeword c' to be equal to a given c is defined as[3]

$$\begin{aligned} FUZZ(c') &= 0 \qquad && \text{if nearness(c,c')} = \text{z} \leq \text{z}_0 < 1 \\ &= \text{z} && \text{otherwise} \end{aligned}$$

2.6 Definition : ***Commitment scheme[1]*** is a tuple***{P, E,M }*** Where ***M*** ={0,1}n is a message space, ***P*** is a set of individuals , generally with three elements A as the committing party, B as the party to which Commitment is made and TC as the trusted party , ***E*** = { (t_i, a_i) } are called the events occurring at times t_i, i = 1,2,3 , as per algorithms a_i , i = 1,2,3. The scheme always culminates in either acceptance or rejection by A and B.

The environment is setup initially, according to the algorithm *Setupalg* (a_1) and published to the parties A and B at time t_1. During the Commit phase,

A uses algorithm *Commitalg* (a_2), which encapsulates a message m□M, along with secret string S□R$\{0,1\}^k$ into a string C. The opening key (secret key) could be formed using both m and S. A sends the result C to B (at time t_2). In the Open phase, A sends the procedure for revealing the hidden Commitment at time t_3, and B uses this. *Openalg* (a_3): B constructs C’ using *Commitalg*, message m and opening key, and checks weather the result is same as the commitment C

Decision making:
If (C = C')
Then A is bound to act as in ‘m’
Else he is free to not act as ‘m’

2.7 Definition : ***Fuzzy Commitment scheme[2]*** is a tuple ***{P, E, M, f }*** Where ***M***□$\{0,1\}^k$ is a message space which consider as a code, ***P*** is a set of individuals, generally with three elements A as the committing party, B as the party

to which Commitment is made and TC as the trusted party , ***f*** is error correction function (def. 2.3) and ***E*** = { (t_i, a_i) } are called the events occurring at times t_i , i = 1,2,3 , as per algorithms a_i , i = 1,2,3. The scheme always culminates in either acceptance or rejection by A and B.
In the setup phase, the environment is setup initially and public commitment key K generated, according to the algorithm *Setupalg* (a_1) and published to the parties A and B at time t_1.

During the Commit phase, Alice commits to a message m□M according to the algorithm
Commitalg (a_2) into string C.
In the Open phase, A sends the procedure for revealing the hidden Commitment at time t_3 and B use this.*Openalg* (a_3): B constructs C' using *Commitalg*, message t(m) and opening key, and checks weather the result is same as the received commitment t(C), where t is the transmission function.
Fuzzy decision making:
If (nearest (t(C),*f*(C'))$\leq z_0$)
Then A is bound to act as in 'm'
Else he is free to not act as 'm'

3. ANALYSS OF A FUZZY COMMITMENT SCHEME:

This section presents an analysis of possible attacks against a fuzzy commitment scheme.

Let our analysis mainly consider a tuple [7],

{P,E,M,K,g(w,m),C,S,V(v,w),f,α_i}. ……………………. (1)

Where P is a set of individuals, generally with three elements A as the committing party, B as the party to which commitment is made and TC as the trusted party, E = {(t_i, a_i)} are called the algorithms occurring at times t_i, i=1,2,3 , as per algorithms a_i, i=1,2,3 , M $\subseteq \{0,1\}^k$ is a message space which consider as a code, K is the public commitment key according to the algorithm setupalg (a_1) and publish to the parties A and B at time t_1, g_w is an encoding function with key w, C is the image set under g is a code set, which satisfies the closure property under K operation , S is a element of set C, V is the set of verifier's tags for key w with value v, f is error correction function (def. 2.3) , α_i are possible attacks.

3.1 With trusted party:

Now equation (1) will become a tuple

{P,E,M,K,g(w,m),C,V(v,w),f,α_i}. ……………………. (1)

Where P is a set of individuals, generally with three elements A as the committing party, B as the party to which commitment is made and TC as the trusted party, E = {(t_i, a_i)} are called the algorithms occurring at times t_i, i=1,2,3 , as per algorithms a_i, i=1,2,3 , M $\subseteq \{0,1\}^k$ is a message space which consider as a code, K is the public commitment key according to the algorithm setupalg (a_1) and publish to the parties A and B at time t_1, g_w is an encoding function with key w, C is the image set under g is a code set, which satisfies the closure property under K operation , V is the set of verifier's tags for key w with value v, f is error correction function (Def. 2.3) , α_i are possible attacks where i = 1, 2, 3

CASE α_1: Dishonesty of A for g with w

During the commit phase at time t_2:

Let g_w: M → C, $\forall$ m $\in$ M, $V v_{:\,w}$→ {0, 1}, $\forall$ w.

In this case we represent an attack where the committer ignore his key, here the trusted party TC gives a key w to A for hide the commitment and a verifier tag v to B which B can verify the key that A will reveal later.

In this attack, A commit a value 'm', compute g_w (m) and send this value to B. Now to open the commitment A sends w' to B and since every g_w is injective, knowing w' B can compute inverse $g_{w'}^{-1}(m) = m'$. To verify that w'=w and therefore m'=m, B computes his verifying function V (v, w').
Now if V (v, w') =1 than A can cheat to B successfully and B accept the commitment else B reject accordingly.

CASE α_2: Dishonesty of A for g regards v

During the commit phase at time t_2:

Let g_w: M → C, ∀ m ∈ M, $Vv_{:\,w}$→ {0, 1}, ∀ w .

In this case, A attack like this, he try to compute the set V_v of all the tags that B have. He than pick the tag v_0 □ V_v that maximizes Pr [V =v | w=w_0]. Let α_2= Pr [V =v | w=w_0]. By an averaging argument $\alpha_2 \geq 1/ |V_v|$. Now A picks two value m=m' and compute g_w(m). But by concealing property, there is another key w' such that $g_{w'}$ (m') which is equal to g_w (m) and V (v, w') =1 which allow A to cheat B successfully.

CASE α_3: Denial of Service

Parties which act accordingly to the prescribe rule and aimed to achieve the protocol objective are called 'honest'. When at least one honest party is involved, the protocol succeeding despite the objective not having being achieved is a infringe or contravene (discussed earlier) of security.

The protocol is expected to fail is some of the parties act dishonestly – thus it is never in the interest of the dishonest party to perform an action that is guaranteed to lead to the protocol failing.

Here we disregarding the kind of 'Denial of service' attack where dishonest party start up protocol runs but intentionally never complete them.

Conclusion:

Perfidious behavior of one or both communicating parties still in existence, even after having strong cryptographic protocol, which maligns the soul of commitment scheme and shows the failure of it's objective.

Reference:

[1]. M. Blum, "coin flipping by telephone" ,"Advances in Cryptology-A report on CRYPTO'81, pp.11-15, 1981.

[2]. A.Jules and M. Wattenberg. "A fuzzy commitment scheme " in proceedings of the sixth ACM Conference on computer & communication security, pages 28-36,November 1999.

[3]. A.A.Al-saggaf , H.S.Acharya,"A Fuzzy Commitment Scheme" IEEE International Conference on Advances in Computer Vision and Information Technology,28-30,November,2007 – India.

[4]. Xavier boyen "Reusable cryptography fuzzy extractors " in proceedings of the eleventh ACM Conference on computer & communication security, pages82-91,ACM Press 2004.

[5]. Alawi A. Al-Saggaf and Acharya H. S. "A Generalized Framework for Crisp Commitment Schemes "eprint.iacr.org/2009/202.

[6]. V.Pless, “ Introduction to theory of Error Correcting Codes”, Wiley , New York 1982.

[7]. Alexandre Pinto, Andr´e Souto, Armando Matos, Luis Antunes “Galois Field Commitment Scheme” eprint.iacr.org, November 2006.

Maximizing Lifetime of Homogeneous Wireless Sensor Network through Energy Efficient Clustering Method

Asfandyar Khan asfand43@yahoo.com
Computer & Information Sciences Department
Universiti Teknologi PETRONAS
Bandar Seri Iskander, 31750 Tronoh,
Perak, Malaysia

Azween Abdullah azweenabdullah@petronas.com.my
Computer & Information Sciences Department
Universiti Teknologi PETRONAS
Bandar Seri Iskander, 31750 Tronoh,
Perak, Malaysia

Nurul Hasan nurul_hasan@petronas.com.my
Chemical Engineering Department
Universiti Teknologi PETRONAS
Bandar Seri Iskander, 31750 Tronoh, Perak, Malaysia

Abstract

The main purpose of this paper is to develop a mechanism to increase the lifetime of homogeneous sensor nodes by controlling long distance communication, energy balancing and efficient delivery of information. Energy efficiency is a very important issue for sensor nodes which affects the lifetime of sensor networks. To achieve energy balancing and maximizing network lifetime we divided the whole network into different clusters. In cluster based architecture, the role of aggregator node is very crucial because of extra processing and long range communication. Once the aggregator node becomes non functional, it affects the whole cluster. We introduce a candidate cluster head node on the basis of node density. We introduce a modified cluster based model by using special nodes called server nodes (SN) that is powerful in term of resources. These server nodes are responsible for transmitting the data from cluster head to the base station. Our proposed algorithm for cluster head selection based on residual energy, distance, reliability and degree of mobility. The proposed architecture is more scalable and proposed algorithm is robustness against even/uneven node deployment.

Keywords: *Server node (SN), cluster head (CH), Network lifetime.*

1. INTRODUCTION

A Wireless sensor network (WSN) is composed of large numbers of tiny low powered sensor nodes and one or more multiple base stations (sinks). These tiny sensor nodes consist of sensing, data processing and communication components. The sensor nodes sense, measure and collect ambient environment conditions, they have the ability to process the data and perform simple computation and send the processed information to the base station either directly or through some intermediate point called gateway. Gateway can be used for fusion and removing the anomalies and to get some conclusion from the collected data over a period of time. Wide range of application can be found in [1, 2].

Sensor nodes are resource constrained in term of energy, processor and memory and low range communication and bandwidth. Limited battery power is used to operate the sensor nodes and is very difficult to replace or recharge it, when the nodes die. This will affect the network performance. Energy conservation and harvesting increase lifetime of the network. Optimize the communication range and minimize the energy usage, we need to conserve the energy of sensor nodes [1, 2].Sensor nodes are deployed to gather information and desired that all the nodes works continuously and transmit information as long as possible. This address the lifetime problem in wireless sensor networks. Sensor nodes spend their energy during transmitting the data, receiving and relaying packets. Hence, designing routing algorithms that maximize the life time until the first battery expires is an important consideration.

Designing energy aware algorithms increase the lifetime of sensor nodes. In some applications the network size is larger required scalable architectures. Energy conservation in wireless sensor networks has been the primary objective, but however, this constrain is not the only consideration for efficient working of wireless sensor networks. There are other objectives like scalable architecture, routing and latency. In most of the applications of wireless sensor networks are envisioned to handled critical scenarios where data retrieval time is critical, i.e., delivering information of each individual node as fast as possible to the base station become an important issue. It is important to guarantee that information can be successfully received to the base station the first time instead of being retransmitted.

In wireless sensor network data gathering and routing are challenging tasks due to their dynamic and unique properties. Many routing protocols are developed, but among those protocols cluster based routing protocols are energy efficient, scalable and prolong the network lifetime [3, 4].In the event detection environment nodes are idle most of the time and active at the time when the event occur. Sensor nodes periodically send the gather information to the base station. Routing is an important issue in data gathering sensor network, while on the other hand sleep-wake synchronization are the key issues for event detection sensor networks.

In the clustered environment, the data gathered by each sensor is communicated either by single hop or multihop to base station. In cluster based architectures, at times, each cluster has their own leader node that collects the aggregated data from the non leader nodes and is responsible for the data transmission to the base station. Clustering approach increases the network life time, because each node do not directly communicate with the base station and hence overcome the problem of long range communication among the sensors nodes. To improve the overall network scalability, cluster based architecture can share the traffic load equally among all the nodes in various clusters, due to this the end to end delay between the sensor nodes and command node can be reduced [5]. Many critical issues associated with clustering architecture system because of the non-uniformly distribution of the sensors in the field. Some cluster heads may be heavily loaded than others, thus causing latency in communication, decreasing the life time of the network and inadequate tracking of targets or events.

Self organization of the nodes with in cluster for a randomly deployed large number of sensors was considered in recent years emphasizing the limited battery power and compact hardware organization of each sensor module. To send the information from a high numbers of sensors nodes to base station, it is necessary to be a cost effective and group all the nodes in cluster. It is necessary to examine a list of metrics that determine the performance of a sensor network. In this paper, we propose a strategy to select an efficient cluster head for each cluster on the basis of different parameters. Our proposed strategy will be better in terms of data delivery and energy balancing as shown in Fig 1.

The focus in this paper is to assess the role of strategic sensor node placement (clustering, cluster head selection) on the data delivery of a network. In this regard, Section 2 discusses related work. Section 3 describes the methodology, section 4 describes the analytical discussion, and section 5 presents conclusion and future works.

2. RELATED WORK

In wireless sensor network energy efficient communication is a matter of survival; as a result research mainly focused towards energy efficient communication, energy conservation, prolonging network lifetime. Various routing techniques have been introduced, but clustering architecture is one of the most dominant, and scalable. In [6], node should become a cluster head by calculating the optimal probability, in order to minimize network energy consumption has been proposed. Heinzelman et al [7] introduced a hierarchical clustering algorithm for sensor networks called LEACH. It uses distributed cluster formation for a randomized rotation of the cluster-head role with the objective of reducing energy consumption that increases the network life time and data delivery. LEACH uses TDMA/CDMA, MAC to reduce inter and intra cluster collisions. Uppu et al [8] introduced a technique of backup heads to avoid re-clustering, secondary membership heads to eliminate redundant transmission and optimum distance hopping to achieve network efficiency in terms of life time and data delivery.

Balanced cluster architecture are used to maximize the life time of the network by forming balanced cluster and minimize the total energy consumed in communication and also the number of hop is not fixed and depends upon the spatial location of sensors[An energy constrained multi-hop clustering algorithm].L. Ying et al [9] developed an algorithm for cluster head selection based on node residual energy and required transmission energy. In [10], uneven load in network is minimized by cluster size adaptation using cluster ID based routing scheme. Cluster head maintain information of a node with maximum residual energy in its cluster. It ensures the location of cluster head (CH) approximately at center of cluster. In case of uneven deployment, it is not necessary that most of the nodes are in the center of the cluster. In this way, the distance between CH and sensor nodes will be increased.

A cluster-based routing protocol called Energy Efficient Clustering Routing (EECR) [11] select cluster head on the basis of weight value and leads to a more uniform distribution evenly among all sensor nodes. In energy constrained case, the traffic pattern and remaining energy level condition the routing scheme may be adoptive. Cluster heads selection is an NP-hard problem [12]. Thus, in the literature, the solution is based on heuristic approaches. Efficient clustering algorithms do not require regular topology re-construction and this will lead to regular information exchange among the nodes in the network. HEED [13] is another distributed clustering approach. In this approach, the cluster heads are selected periodically on the basis of two parameters, the residual energy and cost incurred during the intra clustering communication.

In [14] some resource rich nodes called gateways are deployed in the network and performing fusion to relate sensor reports. Member nodes of each cluster communicate with base station only through gateway nodes. MECH [15] avoids the uneven member distribution of cluster and reduced the long range communication between cluster head and base station. MECH suffer by Energy consumption in each round.

TEEN [16] based on LEACH, the transmission is less frequently and the sensor nodes sense the media contiguously. After cluster formation and cluster head selection process, CH broadcasts two thresholds value, called hard threshold and soft threshold as the sense attributes. Hard threshold is the minimum possible value of the sense attribute that triggers the nodes to switch on its transmitter and transmit. Thus, the hard threshold reduces the number of transmission and allows the nodes to transmit only when the sensed attribute is in the range of interest. The soft threshold further reduces the number of transmissions when there is little or no change in the value of sensed attributes. The number of packets transmission is controlled by setting the soft threshold and hard threshold. However the main disadvantage of this scheme is that when periodic reports are needed and if the threshold is not received, the user will not get any data from the network at all.

3. METHODOLGY

3.1 System Model and Problem Statement

The sensor nodes are highly resource constrained. Energy is one of the major issues of the sensor nodes. In wireless sensor network most of the energy is consumed during transmission and it is further increased with the distance, as energy consumption is directly proportional to the square of the distance among the nodes. In order to minimize energy consumption and increase lifetime of the network, we must keep the distance under consideration and it is possible by the architecture design of the network and efficient routing schemes. Scalability is also another issue, as they may contain hundreds or thousands of nodes and this issue are addressed in cluster based architecture particularly in LEACH [7]. There are few areas in LEACH [7] that can be improved and to make it more energy efficient and scalable. Avoiding the creation of new routing table and selection of cluster head in each round significantly reduces the amount of energy consumed. In cluster based architectures, cluster head are over loaded with long range transmissions to the base station [10] and with additional processing responsibility of data aggregation. Due to these responsibilities, cluster heads nodes are drained of their energy quickly. It is unsuitable in the case of homogenous wireless sensor networks that cluster heads are regular nodes but they communicate for longer distance with the base station and also the cluster head that are near to the base station are drained of their energy quickly because of inter cluster communication. This leads to energy imbalance among the clusters. For a broad analysis of our proposed scheme, we have developed a system model based on the energy consumption during transmission. In most of short range applications, the circuit energy consumption is higher than transmission energy. Energy efficient communication techniques mainly focus on minimizing the transmission energy, while in long range applications the transmission energy is dominant in the total energy consumption. The transmission energy generally depends on the transmission distance. The findings by [23] that different performance parameters are designed to minimize the energy consumption and to prolong the network lifetime. These parameters are described as follows:

3.1.1 Distance (D) to Base Station

"D" is the summation of all distances from sensor nodes to the BS. This distance is defined as follows:

$$D = \sum_{i=1}^{m}(x_{1s} + x_{2s} + \dots + x_{is})$$

i.e.

$$D = \sum_{i=1}^{m}(x_{is}) \qquad (1)$$

Where "x_{is}" is the sum of distance from the node "i" to the Base Station. For a larger network, try to keep this distance minimum because most of the energy will be wasted. However, for a smaller network the nodes near to base station directly send the information may be an acceptable option.

3.1.2 Cluster Distance (C)

C is the summation of the distances from the member nodes to the cluster head (CH) and the distance from the cluster head to the server node (SN). For a cluster with k member nodes, the cluster distance C is defined as follows:

$$C = \sum_{i=1}^{k}(x_{1h} + x_{2h} + \cdots + x_{ih}) + (x_{h1sn} + x_{h2sn} + \cdots + x_{hisn})$$

i.e.

$$C = \sum_{i=1}^{k}(x_{ih} + x_{ihsn}) \qquad (2)$$

Where “x_{ih}” is the distance from node “i” to the cluster head (CH) and “x_{ihsn}” is the distance from the cluster head to the server node (SN). For a cluster that has large number of spreads nodes, the distance among the nodes ‘i’ to cluster head (CH) will be more and the energy consumption will be higher. So keep the cluster size small to reduce energy dissipation and C should not be too large. This metric will keep control the size of the clusters. The Equation.3 shows the total distances “T_{dist}” from cluster member ‘i’ to cluster head and from cluster head (CH) to server node (SN) and from server Node to Base station (BS).

$$T_{dist} = \sum_{i=1}^{m} (x_{ih} + x_{ihsn} + \ldots + x_{snb}) \qquad (3)$$

3.1.3 Total Dissipated Energy (E)

The total dissipated energy “E_{total}” shows the energy dissipated to transfer the aggregated messages from the cluster to the BS. For a cluster with k member nodes, the total dissipated energy can be calculated follows:

$$E_{total} = \sum_{i=1}^{m} \left(E_{T_x ih} + E_{T_x hsn} + K \times E_R + E_{T_{snb}} \right) \qquad (4)$$

The fist part of Equation.4 show the energy consumed to transmit messages from member nodes to the cluster head. The second part shows the energy consumed to transmit aggregated messages by cluster head to server node SN to receive messages from the member nodes. Finally, the fourth term “ETsnb” resents the energy needed to transmit from the SN to the BS.

3.1.4 Residual energy

The energy dissipated in previous round by the cluster head preferably les then residual energy of a node. Equation for residual energy of node i is described in [17]. The individual node energy (lifetime of node) can be calculated by the formula as presented in [18] is:

$$Ti = \frac{E_b}{\sum_{j \in Si} eij \sum qij} \qquad (5)$$

Where ‘Ti’ is the node lifetime, ‘E_b’ is the initial energy of the battery ‘eij’ is energy required for transmission of one information unit, ‘qij’ is the rate at which information is generated at node i, ‘S_i’ is the set of neighbor nodes of node i. Based on the node lifetime, the network lifetime can be computed by:

$$Tsys = \max_{i \in N} Ti = \max_{i \in N} \frac{E_b}{\sum_{j \in Si} eij \sum qij} \qquad (6)$$

Where ‘T_{sys}’ is the system life time, ‘N’ is the number of nodes in the network.

3.2 Proposed Cluster Based WSN Architecture

We have introduced a resource reachable node called server node (SN). It has the ability to cover long transmission range. Server node (SN) is deployed in a location where all the nodes of each cluster are easily reachable. If it is not reachable, it is recommended to add another server node (SN). Due to extra processing capability sever node (SN) are responsible for selecting cluster head from candidate nodes. The purpose of introducing SN is to closely monitor the operation of sensor nodes in a cluster and command them for specific operations as shown in Fig 1.

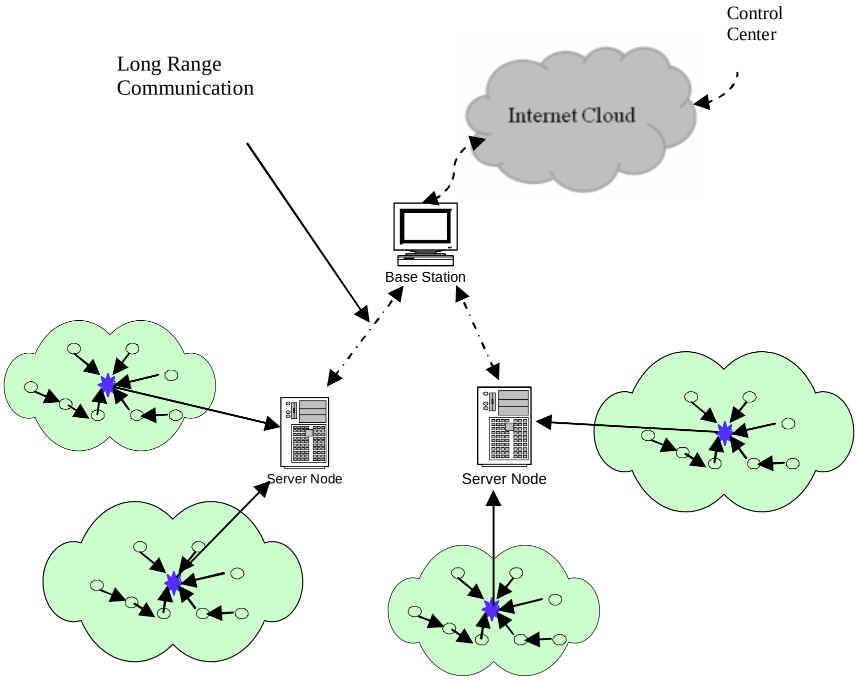


FIGURE 1: Modified Cluster Based Architecture

3.3 Cluster Formation and Node Deployment

The main purpose of clustering the sensor nodes is to form groups so as to reduce the overall energy spent in aggregation, communicating the sensed data to the cluster head and base station. There are various methods proposed for clustering, but K-mean [19] is found to be the most efficient for clustering. Fig 1 shows the formation of different clusters and the assignment of nodes to each cluster. K-mean clustering technique assigns nodes to the cluster having the nearest centroid. K-mean algorithm uses Euclidean distance formula to calculate distance between centroid and other objects Fig. 2.

$$D_{i,j} = \sqrt{\left|x_{i1} - x_{j2}\right|^2 + \left|x_{i2} - x_{j2}\right|^2 + \cdots \left|x_{ip} - x_{jp}\right|^2} \quad (7)$$

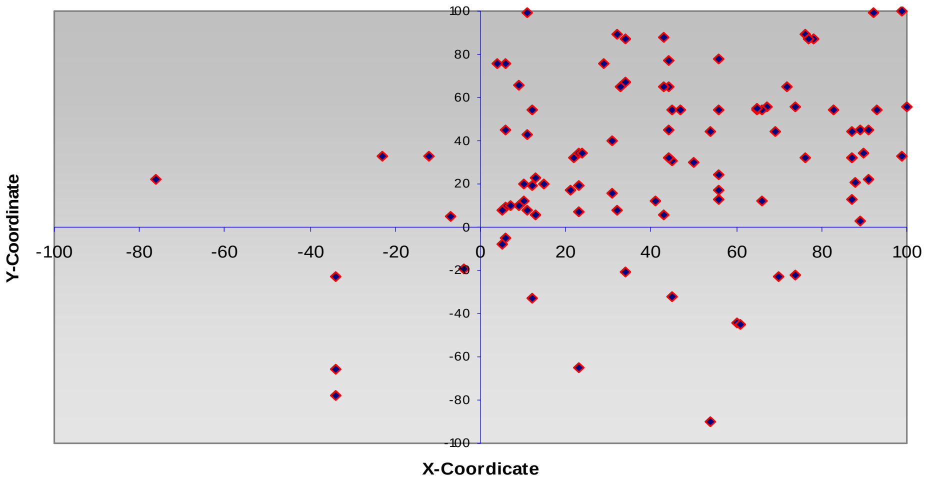


FIGURE 2: 100 Nodes Random Network

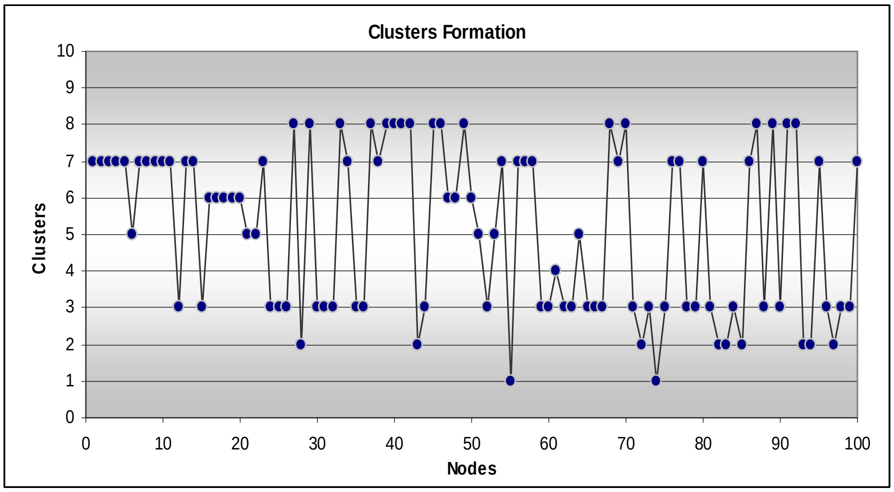


FIGURE 3: Cluster Formation

K-mean is computationally efficient and does not require the user to specify many parameters. The number of clusters and the nodes assigned to each cluster is shown in Table 1.

Cluster	Nodes
1	2
2	9
3	31
4	1
5	6
6	8
7	26
8	17
Valid	100
Missing	0

Table I: Number of Cases in each Cluster

	Clusters							
	1	2	3	4	5	6	7	8
X	-34.00	99.00	91.00	-76.00	-34.00	54.00	34.00	9.00
Y	-78.00	100.00	22.00	22.00	-23.00	-90.00	-21.00	66.00

Table II: Cluster Centroids

3.4 Selection of Candidate Cluster Head

The selection of cluster head becomes highly challenging when there is an uneven distribution of sensor nodes in clusters. In order to make the cluster head (CH) selection algorithm more accurate, we first identify the candidate sensor nodes for cluster head and then select the best among them. In order to select candidate cluster heads from each cluster, we use the K-theorem. The philosophy behind the K-theorem is to select a candidate CHs based on the bunch of sensor nodes as there is an uneven distribution Fig. 4.

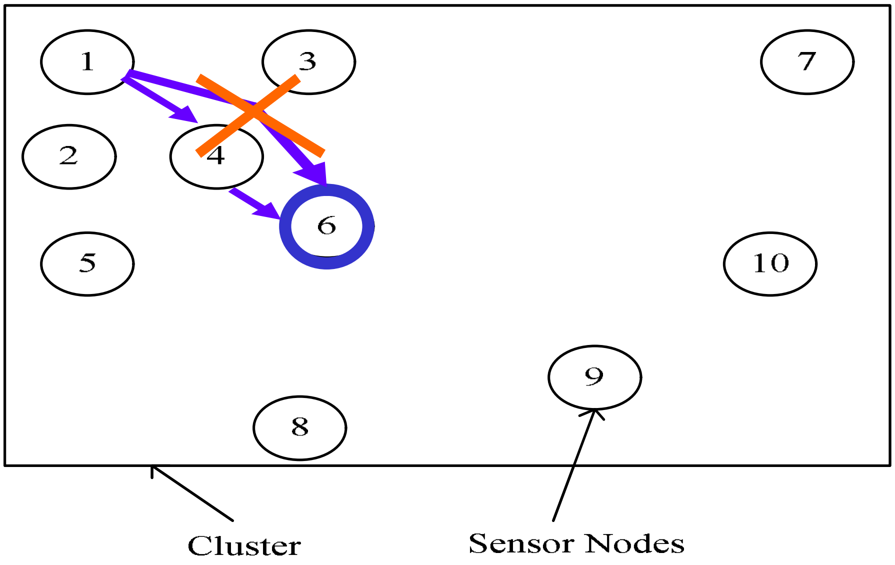


FIGURE 4: Selection of Candidate Cluster Head

The server node (SN) set the value of ki for each cluster. The value of ki is relative to the node density in a cluster and ratio (i.e. r) of the cluster heads in a WSN. It is the product of the number of nodes in a cluster (i.e. ni) and ratio r. The value of r can vary from 0.01 to 0.99 but it should not be more than 0.50. The lesser the value of ki, the greater the probability of getting a local optimal is. The value of ki determines the ki number of best sensor nodes as candidate CHs.

For each sensor node deployed in the cluster, we choose its ki nearest neighbors based on distance. The distance between sensor nodes can be calculated through received signal strength indicator (RSSI) that is described in detail in [20] or any other localization technique [21, 22]. When the number of immediate (1-hop) neighbors is less than ki and distance is greater than the transmission range then multihop route is preferred. Multihop route is preferred over direct because of less energy consumption.

The server node (SN) adopts the procedure detailed below in order to select candidate CHs for each cluster. The server node (SN) maintains a table for each cluster, listing all the sensor nodes present in the cluster. It maintains ki nearest neighbors for each node and the frequency of occurrence of each node is maintained in the table. The ordered list of sensor nodes based on their frequency is shown in Table. III i.e. Si. The minimum frequency required in cluster i to be the CH (i.e. Ki) is calculated based on weighted mean of frequencies and 1 is added for better result. Weighted mean is calculated by product of each frequency of occurrence into number of sensor nodes having that frequency. The value of Ki is rounded to the nearest integer if required. The sensor nodes having frequency Ki or greater are identified as candidates for CH i.e. Ci. The best candidates for cluster head would always be equals to value of ki i.e. 3 in this case.

Node ID	$k_i = 3$ List of Terminals with its K-Nearest Neighbor	Frequency of Occurrence
1	1) 2, 3, 4	3
2	2) 1, 4, 5	4
3	3) 1, 4, 6	5
4	4) 2, 3, 6	6
5	5) 2, 4, 6	4
6	6) 3, 4, 5	7
7	7) 3, 9, 10	2
8	8) 5, 6, 9	2
9	9) 6, 8, 10	4
10	10) 6, 7, 9	3

Table III: List of nodes with their K-nearest neighbors and their frequency of occurrence

Sorting {Si = Ordered list of sensor nodes, where i is the frequency of occurrence}

S2 = (7, 8), S3 = (1, 10), S4 = (2, 5, 9), S5 = (3)

S6 = (4), S7 = (6)

Ki = [Weighted Mean] + 1

= [(2*2)+(3*2)+(4*3)+(5*1)+(6*1)+(7*1) / 10] + 1

= [(4 + 6 + 12 + 5 + 6 + 7) / 10] + 1

= [(40) / 10] + 1➔ 4 + 1

Ki = 5

So, the best nodes for candidate cluster head in cluster are: Ci = {3, 4, 6} when ki = 3.

3.5 Cluster Head Selection

We describe a cluster head selection algorithm which is energy efficient in Fig.5. Suitable cluster-head selection makes the network efficient and increases the life time and data delivery of the networks.

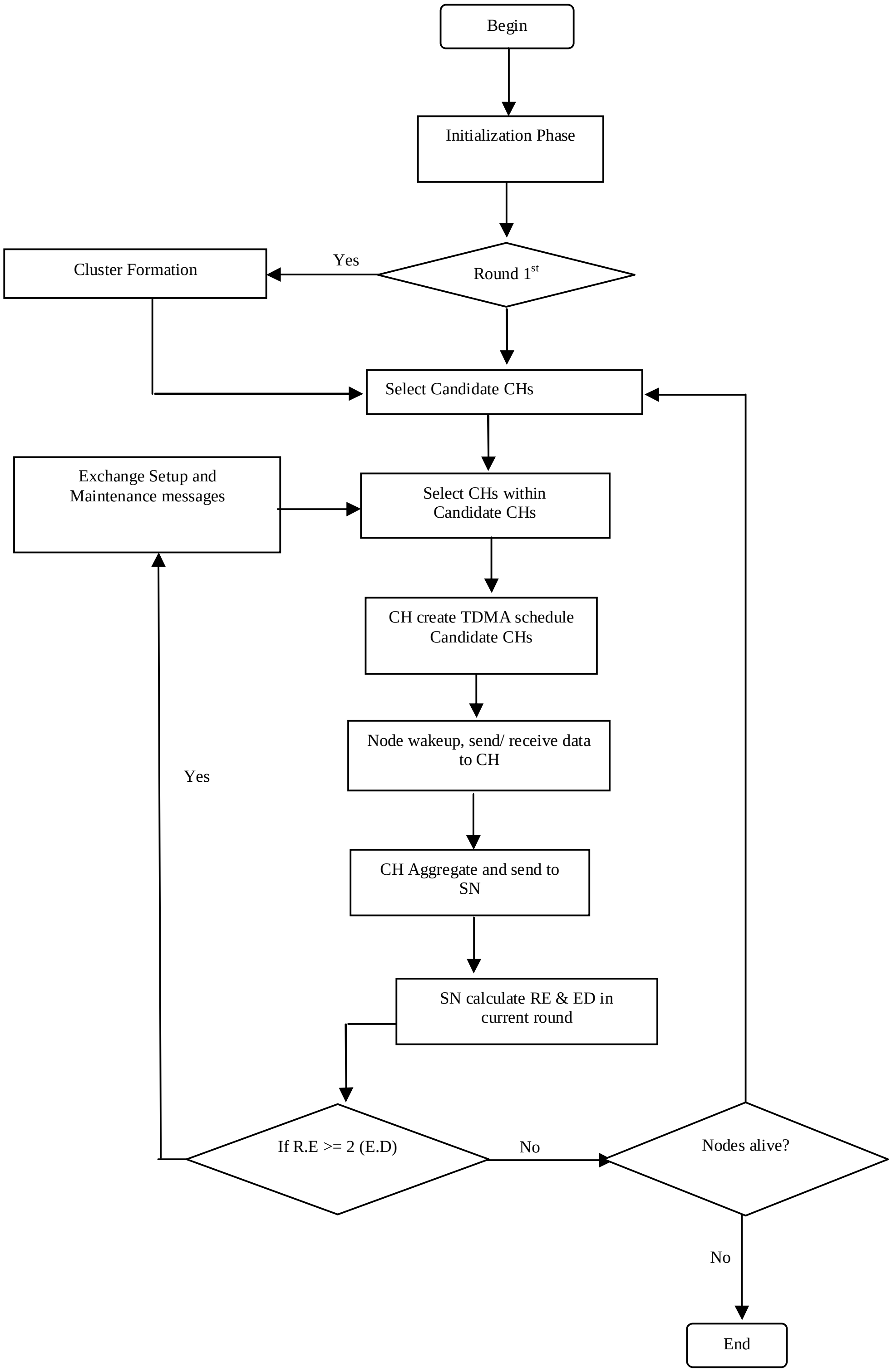

Begin
Initialization Phase
Cluster Formation
Yes
Round 1st
Select Candidate CHs
Exchange Setup and Maintenance messages
Select CHs within Candidate CHs
CH create TDMA schedule Candidate CHs
Node wakeup, send/ receive data to CH
Yes
CH Aggregate and send to SN
SN calculate RE & ED in current round
If R.E >= 2 (E.D)
No
Nodes alive?
No
End

FIGURE 5: Flow Chart of Cluster Head

4. ANALYTICAL DISCUSSION

In our methodology the idea is to distribute the load of long range transmission from CH to the SN. This will conserve energy at CH and ultimately lead to increase in the life time of wireless sensor network. The problem is when CHs are burdened with extra processing by gathering information from non cluster heads and as a result the CHs are drained of their energy quickly. There is no need to have inter-cluster multihop communication because each cluster head is able to directly reach the SN. In this way, the CHs near the BS or SN does not have to take extra responsibility of transmitting other clusters information. They just have to transmit their own data. This will lead to energy balancing among the clusters that was previously a problem. The modified architecture can be more scalable as we just need to add extra SN if network density or distance is high.

The proposed algorithm describes the selection process of CHs. The benefit of using K-Theorem is that it minimizes the communication and reduces the long range intra-cluster communication by selecting the best nodes for cluster heads from the location where network density is higher.

5. CONSLUSION & FUTURE WORK

Energy efficiency is the most important design consideration for wireless sensor networks and its optimum utilization is a challenge in its own regard. We achieved energy efficiency through efficient cluster head selection algorithm. We divide sensor nodes into clusters through by using K-Mean. Our model is simple, efficient and less costly and can scale well to large networks. The findings of this research can be summarized as follows. We believe that this will not only minimize the communication cost but will also increase the reliability of the network.

- A good clustering technique is one that achieves an energy balance in the network and maximum data delivery.
- The main factor for energy balance and data delivery is the efficient clustering and cluster head selection.
- Optimization of data delivery depends upon the optimum level of energy of CH neighbor's nodes. Our work can be extended to analyze the performance for other parameters. i.e. latency, throughput and efficient routing techniques.

6. REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, "Wireless sensor networks: a survey", *Computer Networks: The International Journal of Computer and Telecommunications Networking*", v.38 n.4, p.393-422, 15 March *2002*

[2] M. Younis, K. Akkaya, M. Eltoweissy, A. Wadaa, "On Handling QoS Traffic in Wireless Sensor Networks", *Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04)* - Track 9, p.90292.1, January 05-08, 2004

[3] K. Akkaya, M. F. Younis, "A survey on routing protocols for wireless sensor networks", *Ad Hoc Networks",* 3(3): 325-349 (2005)

[4] A. A. Abbasi, M. F. Younis, "A survey on clustering algorithms for wireless sensor networks", *Computer Communications:* 30(14-15): 2826-2841 (2007)

[5] C. P. Low, C. F. J. M. Ng, N.H.Ang, "Efficient Load-Balanced Clustering Algorithms for Wireless Senser Networs", *Elsevier Computer communications*", pp 750-759,2007

[6] H. Yang, B. Sikdar, "Optimal Cluster Head Selection in the LEACH Architecture", *In the proceedings of the 26th IEEE International Performance and Communications Conference", IPCCC2007*, April 11-13, 2007, New Orleans, Louisiana, USA

[7] W. Heinzelman, A. Chandrakasan, H. Balakrishnan, “Energy-efficient communication protocols for wireless microsensor networks”, *Proceedings of the Hawaii International Conference on Systems Sciences,* Jan. 2000

[8] N. Uppu, B. V. S. S. Subrahmanyam, R. Garimella, “An Energy Efficient Technique to prolong Network Life Time of Ad Hoc Sensor Networks (ETPNL)”, IEEE Technical Review Vol 25,2008

[9] L. Ying, Y. Haibin, “Energy Adaptive Cluster-Head Selection for Wireless Sensor Networks”, *In Proceedings of the 6th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT’05)”,* pp. 634--638, 2005

[10] I. Ahmed, M. Peng, W. Wang, ”A Unified Energy Efficient Cluster ID based Routing Scheme for Wireless Sensor Networks-A more Realistic Analysis”, *Proc of IEEE Third International conference on Networking and Services (ICNS’07),* pp-86, 2007

[11] L. D. Shu-song, W. Xiang-ming, ”An energy Efficient clustering routing algorithm for wireless sensor networks”, *The journal of China Universities of posts and Telecommunications,* Volume 13, Issue 3, September 2006

[12] S. Basagni, I. Chlamtac, A. Farago, “A Generalized Clustering Algorithm for Peer-to-Peer Networks”, *In: Workshop on lgorithmic Aspects of Com., Bologna,* Italy (July 1997)

[13] O. Younis, S. Fahmy, Distributed Clustering for Scalable, Long-Lived Sensor Networks”, *Purdue University, Technical Report,* CSD TR-03-026 (2003)

[14] G. Gupta, M. Younis, “Load-Balanced Clustering in Wireless Sensor Networks”, *In:Proceedings of the Int. Conference on Communication, Anchorage,* AK (2003)

[15] R. S. Chang, C. J. Kuo, ‘‘ An energy efficient routing mechanism for wireless sensor networks, ’’ *Advanced Information Networking and Applications, 2006. AINA 2006. 20th International Conference,* Volume 2, 18-20 April 2006 Page(s):5 pp

[16] A. Manjeshwar, D. P. Agarwal, "TEEN: a routing protocol for enhanced e±ciency in wireless sensor networks," *In 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing,* Apri 2001

[17] Tillapart. P., S. Thammarojsakul, T. Thumthawatworn, and P. Santiprabhob, “An Approach to Hybrid Clustering and Routing in Wireless Sensor Networks”, *Proc of IEEE Aerospace Conference, 2005*

[18] H. Kar, A. Willing, “Protocols and Architecture for Wireless Sensor Networks”, John Wiley and sons, 2005

[19] D. Arthur, S. Vassilvitskii,” How Slow is the k-Means method?”, *SCG’06, June 5–7, 2006, Sedona, Arizona, USA*. Copyright 2006 ACM 1595933409/06/0006

[20] A. A. Minhas, “Power Aware Routing Protocols for Wireless ad hoc Sensor Networks”, *Ph. D Thesis, Graz University of Technology, Graz, Austria,* March 2007

[21] L. Hu, D. Evans, “Localization for mobile sensor networks”, *ACM International Conference on Mobile Computing and Networking,* (MobiCom 2004)

[22] J. Hightower, G. Borriello, “Location systems for ubiquitous computing”, *IEEE Computer* 34 (8) (2001) 57–66

[23] A. Matin, “Energy-Efficient Intelligent hierarchical Cluster-Based Routing for Wireless Sensor Networks”, *MSc Thesis, Acadia University* 2006

COMPUTER SCIENCE JOURNALS SDN BHD
M-3-19, PLAZA DAMAS
SRI HARTAMAS
50480, KUALA LUMPUR
MALAYSIA