IACSIT International Journal of Engineering and Technology, Vol.2, No.4, August 2010
ISSN: 1793-8236

Novel Use of Steganography for Both Confidentiality and Compression

Fahad Ullah, Muhammd Naveed, Mohammad Inayatullah Babar and Faisal Iqbal

Abstract—**Text is one of the most important data that is transmitted on today's communication networks, in the form of html WebPages, commands to access devices remotely using ssh or telnet, email etc. Confidentiality is one of the fundamental requirements for secure communication on an untrusted channel and compression is also required to conserve the bandwidth of the channel. In this study, steganography is used in a completely novel way that is different from the traditional use of steganography. Confidentiality and compression of large text using steganography is presented. In our approach text i.e. message is encoded using a grayscale bitmap image. The image acts as a steganographic carrier for the text, the carrier is never transmitted across the untrusted channel. Only the compressed index array that contains the indices for our data hidden in the image is transmitted. The image also acts as a shared key between sender and receiver, which is used for confidentiality and also to extract the desired text from the image. Encoding text into the image not only makes it secure but we achieve good amount of compression of the message that is to be sent across the channel. Results for different images and texts are compared and for every image, the length of the text is found to be directly proportional to the amount of compression.**

Index Terms—**Steganography, Confidentiality, Compression, Index Array, Shared image as a key**

I. Introduction

From the very beginning in communication, text has been the most important form of information exchanged across the ends. Even after the introduction of audio and vivid media across internet including picture and video, textual information has still its importance in data communication. Due to the unsecure nature of Internet, information can be easily eavesdropped. To avoid such problems, various security methods are used in which the most popular one is the encryption of the information to make it unintelligible for the eavesdropper. Steganography [1] is a method in which information is hidden in an image and the image itself is transmitted across the channel. In contrast to common encryption methods, steganography is better because eavesdropper doesn't know if there is any confidential data, and also if he knows in advance he doesn't know where to look for the message because the image appears to be common information rather an encrypted message to attract the attention of the eavesdropper. In this paper, instead of using traditional steganographic approach, a new method is devised in which steganography is used in a novel and non-traditional manner and it is used to convert the text data into indices of an image and then this converted index array is sent over the channel. The text is encoded in the image but the image is never sent across the channel and is used by both the parties as a key. The text is mapped to the image for corresponding values of the pixels shared between both the image and the text and the indices of the image containing the characters are saved in an index array. The index array, above certain threshold of text data size, exhibits compression with respect to the original text size. Larger the text size better is the compression. The index array is further compressed through a 3rd party compressor [8]. Also, because of the change of pixels is expected in the image after encoding text in it, this pixel information must be sent to the intended receiver so that the key i.e. the picture can be modified at the location where pixel change was observed after encoding. The array is decompressed and decoded at the receiver end and the key i.e. image is modified using information from pixel change and then the message is extracted back from the image.

The basic aim of this study is to use Steganography not only for confidentiality but also for text compression. This paper also analyses the behaviour of compression with respect to the text file size and other factors including encoding density, compression ratio, best possible compression and the impact of using a 3rd party compressor on index array for further compression.

Section II presents some related work, Section III details the proposed approach, Section IV includes the simulation of the model in MATLAB, Section V shows and explains the results of the proposed approach, Section VI explains possible future work on this research, Section VII concludes the study and Section VIII presents the image set used.

II. Related Work

A lot of research is being done in the field of steganography and compression. KB Raja et. al proposed a high capacity wavelet steganography(HCWS) algorithm[2]. The cover image in this model is transformed to wavelet domain and the payload is encrypted using a random technique to increase its security. Tuomas Aura et. al proposed that using gray scale images for cover is the best approach [3]. He proposed a new method for pseudorandom hiding bit selection in random access covers. Juneja et. al proposed a robust image steganography technique based on Least Significant Bit insertion and RSA encryption technique[4]. They used the method of ranking a set of images in a library based on their suitability to be used as a cover or carrier. Weifeng Sun Nan Zhang et. al proposed StarNT, a dictionary-based fast and lossless text transform algorithm[5]. Bernhard Balkenhol Stefan Kurtz et. al provided an analysis of BWT(burrows-wheeler transformation) from the aspect of information theory[6].

Gutman P.C et. al proposed a hybrid approach to text compression using both symbol wise method for its good compression results and dictionary method for its high speed[7]. A lot of work is done on lossless compression algorithms including Context Tree Weighting method (CTW) [10], LZ77 [11], LZW [12].

III. Proposed Approach

A. Steganography for Confidentiality

Steganography, just like cryptography is a method [1] for ensuring confidentiality of a message or information to be sent across an untrusted channel and it is, unlike cryptography, more effective art because it doesn't attract the attention of the attacker or eavesdropper. In this paper, steganography is not used for its intended purpose rather it is only used to secure the text in the image and the image is used as a shared key between the sender and receiver rather actually transmitted across the channel as shown in Fig. 1. Using image as a key has many plus points. Due to no limit on the size of image, it is actually a key of indefinite size. Larger the image size, powerful is the key. Also the text has a lot of redundant characters. A single character, wherever it lies in the text, is mapped to a single pixel value where it actually lies in the image in its integer form. For most cases, the image is modified very little by the text after mapping and the average pixel change in both original image and modified image is in range of 2-5 pixels per total number of pixels in the image.

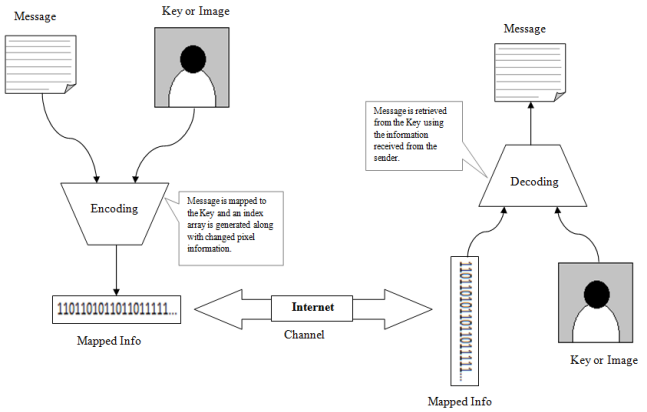

Fig. 1 Encoding and Decoding Process.

The encoding density (d) is given by;

$$d = \frac{\text{Number of altered pixels}}{\text{Total number of pixels}} \text{ x } 100$$

For 512x512 grayscale image, the encoding density is in range of $7.6 \text{x} 10^{-4}$ which shows that the modified image is almost the same as the original one. This little amount of pixel change has two advantages.

1. The overhead generated due to the requirement of sending pixel change information to the receiver is quite less, hardly 100 bytes, because of the small encoding density. This helps in compression which is discussed in next section.
2. If the key was to send across some other channel to some other receiver, the pixel change is small enough to make it almost impossible for an eavesdropper to consider the image suspicious even if the key is some known image, for instance an image from standard test images.

As pixel change and index array(after compression) are the only information required to recover the message back from the key ,that is the image, so these two quantities are sent to the receiver. Across the channel, even if an eavesdropper can grab these values, it's not possible to reconstruct the message from the index array and the pixel change information because the shared image is still unknown to the eavesdropper and it is impossible to draw the image from the available information acquired by the eavesdropper.

B. Steganography for Compression

In this paper, steganography is also used for the purpose of compression. A text file, due to high redundant data, can be compressed to smaller size using many lossless compression algorithms including Context Tree Weighting method (CTW) [10], LZ77[11], LZW[12] etc. Our approach is to use steganography for text compression. After encoding text in the image, an index array is obtained which contains the image pixel locations where the text is mapped. For some initial values of the text file size, it is observed that the index array size exceeds the original text but after a certain threshold, the text file size surpasses the index array size and hence compression is achieved in the index array in contrast to the original text. The compression ratio keeps improving until reaching some saturation level where the change in it is almost indiscernible. The amount of compression appears to have a direct relationship with the text size after the threshold point until the saturation is achieved.

IV. Simulation

MATLAB is used for the simulation purpose. Encoding is done at the sender end using Encoding module and the information is sent to the receiver, on the basis of which the receiver uses Decoding Module to retrieve the original message from the Key.

A. Encoding Module

The encoding process is explained in the flow chart shown in Fig. 2. As clear from the chart, the algorithm is simple. Every character in text, converted to their respective 8 bit integer values, is searched in the image. If found than the resulted index (only first one found in the image) is stored in an index array. If not found, then Approx() function is used to approximate the current character value to the image pixel values, the image is searched for a value at some location nearest to the character value and the index where it lies in the image is returned. Now the image can be altered at that location for the current character in the loop. The index array, after normalization, is sent to dzip() [8] function for compression. Pixels altered in the modified image are found and the values along with their indices are saved in another variable. Now the information including altered pixels and compressed index array are sent to the receiver.

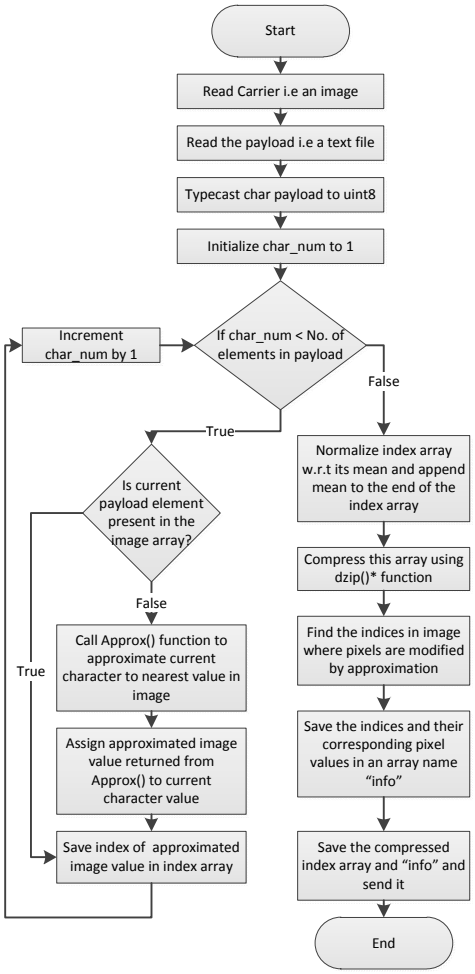

Fig. 2 Encoding Module

The Approx() function, as clear from its explanation, takes two inputs that is the image and the current character in the loop and return index of the approximated value.

1) Approx() function

- It takes 2 inputs, the Carrier and the Current Character and returns the Index where the character can be approximated in the carrier.
- In this function, an array is generated which contains the absolute difference between every element of Carrier and the Current Character.
- Minimum of the difference array is found and as this array shares the same indexing and size with carrier, so the location where minimum is found is actually the index required; where the character value can replace the pixel value.
- A check is performed to avoid the mapping of the current character to an index already altered by some previous character.
- If this happens then the character is approximated with a value second closest to it in the image in order to avoid loss of previous character mapping info in the image.

B. Decoding Module

The decoding process is explained in the flowchart shown in Fig. 3.

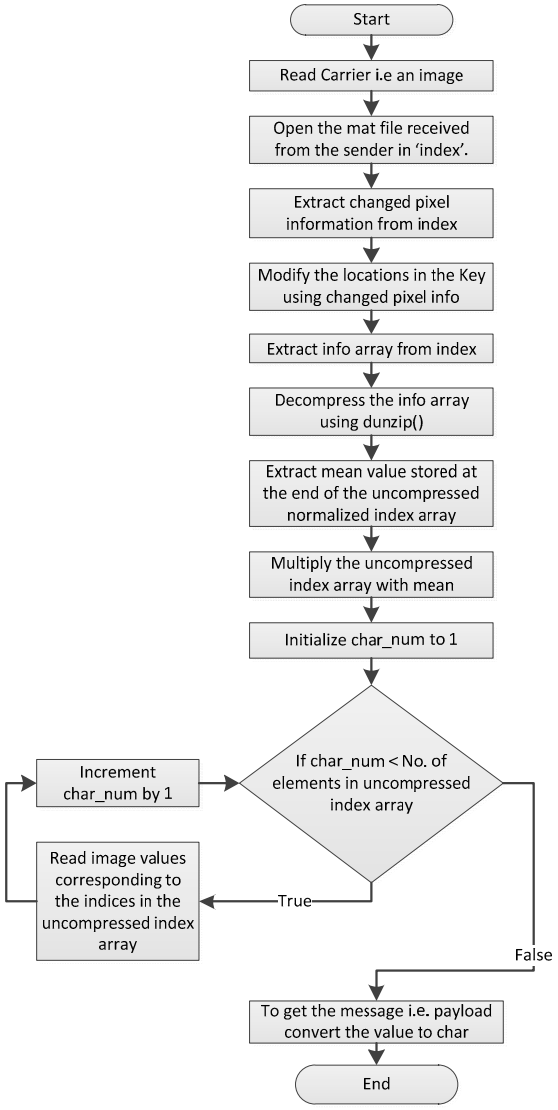

Fig. 3 Decoding Module

In this process, the image i.e. the key is read and is saved in a variable. The received variables containing the changed pixel values and the compressed index array information are also read into variables. Now the index array is decompressed using dunzip() [8] function. Pixel values at the corresponding locations are changed on the basis of changed pixel information. After that, the message is read from the key using the index array, which contains the corresponding indices for each character of the message in the key, and decoding process completes.

1) Lossless compression and decompression (dzip () and dunzip() functions).

dzip() and dunzip() [8] are user-made MATLAB functions which uses ZLIB Deflate algorithm[13] for lossless data compression and decompression of most of MATLAB variables. Index array, though already exhibiting compression is further compressed using dzip() function and sent to the receiver which uses dunzip() to recover the array.

V. RESULTS

For simulation, the following sources were used.

A. Image or Key:

Fig. 6 displays the set of images used in the simulation. The first four images are taken from standard test images while the last one, bitArray.bmp is 16x16 matrix whose values ranges from 0 to 255 and every value is distinct. bitArray.bmp is used to achieve maximum compression but using this array has its limitations which will be discussed later in this section.

TABLE I
RESULTS OF DIFFERENT IMAGES

Key or Image	Image Resolution in pixels	Payload Size (bytes)	Receiver Info Size without using dzip (bytes)	Receiver Info Size using dzip(bytes)	Compression Ratio		Encoding Density (d)
					Without dzip	With dzip	
Airplane.bmp	512x512	16536	11309	11120	0.6839	0.6724	1.1444e-003
Lena.bmp	512x512	16536	13753	11122	0.8317	0.6725	1.1444e-003
Zelda.bmp	512x512	16536	13643	11083	0.8250	0.6702	3.8147e-004
Mandrill.bmp	512x512	16536	13391	11202	0.8098	0.6774	3.8147e-003
bitArray.bmp	16x16	16536	10694	11022*	0.6467	0.6665	0
bitArray.bmp	16x16	163660	103868	100447*	0.6281	0.6074	0

- Lena.bmp, 512x512 grayscale-Bitmap image (bit depth=8bit).
- Airplane.bmp, 512x512 grayscale-Bitmap image (bit depth=8bit).
- Zelda.bmp, 512x512 grayscale-Bitmap image (bit depth=8bit).
- Mandrill.bmp, 512x512 grayscale-Bitmap image (bit depth=8bit).
- bitArray.bmp, 16x16 grayscale-Bitmap image (bit depth=8bit). *

B. Text or Payload:

- "***Eloisa to Abelard* by Alexander Pope**" [9] in text file 'payload.txt'. Full size =16536 bytes.

Compression Ratio is given by:

$$C/Ratio = \frac{\text{Compressed Size}}{\text{Uncompressed Size}}$$

Using a 512x512 bitmap image as a key means a very high end confidentiality. Predicting the key or image from the decoding information sent across the channel is very hard or impossible. Table 1 shows different images used as carrier for a fixed text size of 16536 bytes (except for the last row). Among those, Airplane.bmp has better results for both cases that are with and without dzip. Zelds.bmp results with using dzip are good because of the very low encoding density associated with it which means the pixel information overhead in the decoding information is smaller. bitArray.bmp is standard 16x16 pixel image with all 8 bit characters it can cover. The limitations associated with it are explained later in this section.

The graph in Fig. 4 represents text size vs. compression ratio curve for different carriers used in the simulation without using the index array compression by dzip() function. In Fig. 5, the graph is similar except the compression ratio calculations are made using dzip() function for the compression of index array at encoding side.

The graphs have following characteristics:

- In either graphs or clearly in graph of Fig. 4, the best compression is achieved using the bitArray.bmp but the shortcomings are there using it and are explained later.
- In both graphs of Fig. 4 and Fig. 5, the compression ratio passes through some threshold, which is 1 for the vertical axis and varies for horizontal axis mostly less than 2000 bytes. Before this, the index array actually expands and after it the actual compression starts.
- The graph in Fig. 5, that is using dzip() function for further compression, shows that the compression curves for all images are almost the same and hence the different curves are hard to distinguish.
- The use of dzip function works better on larger text and it can be seen from comparing both graphs. For lower text size, the compression ratio is better without using dzip function but for higher text size, for instance over 15,000 bytes, the ratio gets remarkably better using dzip function for most of the images except the bitArray.bmp.

C. bitArray Characteristics

* bitArray is actually a user generated image array of size 256 bytes whose values range from 0 to 255. It is saved in a 16x16 bitmap picture. Because it covers all text characters within 8 bit range, that is why the encoding density is 0. Using bitArray.bmp for encoding process, the compression is at its best. Using this bitmap as a key has its limitation because it is easily predictable in comparison to other images due to its small size. Also it is more suspicious if sent across the channel. The achievable compression using this array is also not remarkably huge when compared to all other bitmaps used in the simulation especially Airplane.bmp. Also due to some constraints associated with dzip() function [8], the index array generated after using bitArray.bmp expands as obvious from the Table 1 unless text size reaches a proper threshold. The highlighted row in Table 1 shows that for a larger text, dzip can actually compress the index array generated by using bitArray.bmp.

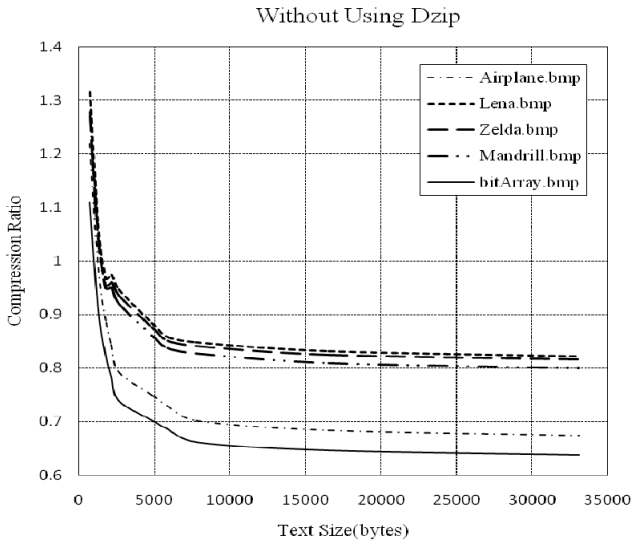

Fig. 4 Text Size vs. Compression Ratio Without Using dzip

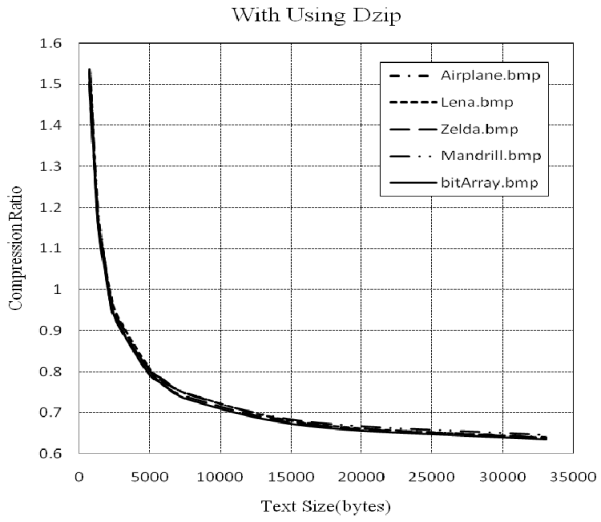

Fig. 5 Text Size vs. Compression Ratio Using dzip

VI. Future Work

The future work may include the processing time optimization for the process of compression and steganography as in this paper, the processing resources associated with compression process aren't considered. Also like a 3rd party compressor is used for further compression, that is dzip() function, some another compression method can be devised to get better results meaning better compression ratio. Besides, this novel approach to steganography can be used on color images with higher color depths e.g. 16 or 24 bpp instead of using grayscale images as used in this research.

VII. Conclusion

In this paper, steganography is used in a novel way that is not just for the sake of confidentiality, but, also for the compression of the text message. Instead of following traditional steganographic methods, that is sending the carrier across the channel, the image is used only to encode text and the index array generated after encoding is sent along with the overhead associated with the image used, with overall size much less than the actual text size. Hence the image works as a key shared between the two ends that is the sender and the receiver. Text size associated with the message is an important factor and after a threshold of the size value, compression is achieved between the secure message that is the index array and the text message sizes. Based on the simulation results, it can be concluded that for large bulk of text, the compression achieved is better in some limits.

VIII. Image Set

The following images were used for the simulation:

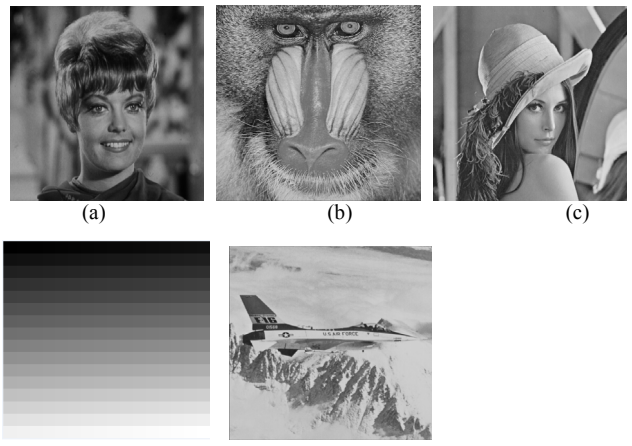

Fig.6 (a) Zelda (b) Mandrill (c) Lena (d) bitArray (magnified) (e) Airplane

References

[1] Artz D., "Digital steganography: hiding data within data", Internet Computing, IEEE, vol. 5, Issue: 3, pp. 75-80, May/Jun 2001.

[2] Raja K.B., Vikas, Venugopal K.R. and Patnaik L.M., "High capacity lossless secure image steganography using wavelets," Advanced Computing and Communications, 2006, pp. 230-235, Dec. 2006.

[3] Aura T, "Practical invisibility in digital communication," In Information Hiding: First International Workshop. Lecture Notes in Computer Science, Vol. 1174. Springer-Verlag, Berlin Heidelberg New York 1996, pp. 265-278.

[4] Juneja M. and Sandhu P.S., "Designing of robust image steganography technique based on LSB insertion and encryption," Advances in Recent Technologies in Communication and Computing, 2009, pp. 302-305, Oct. 2009.

[5] Weifeng Sun, Nan Zhang and Mukherjee, A., "Dictionary-based fast transform for text compression," Information Technology: Coding and Computing [Computers and Communications], 2003, pp. 176-182, April 2003.

[6] Balkenhol B. and Kurtz, S., "Universal data compression based on the Burrows-Wheeler transformation: theory and practice," Computers, IEEE Transactions on, vol. 49, Issue: 10, pp. 1043-1053, Oct 2000.

[7] Gutmann P.C. and Bell T.C., "A hybrid approach to text compression," Data Compression Conference, 1994, pp. 225-233, Mar 1994.

[8] Michael Kleder, "Rapid Lossless Data Compression" , MATLAB Function, http://www.mathworks.com/MATLABcentral/fileexchange/8899 .

[9] Text Source, Alexander Pope's "*Eloisa to Abelard"*, http://www.monadnock.net/poems/eloisa.html .

[10] Sadakane K., Okazaki T. and Imai H., "Implementing the context tree weighting method for text compression," Data Compression Conference 2000, pp. 123-132, 2000.

[11] Chi-Hung Chi, "Study on mutli-lingual LZ77 and LZ78 text compression," Data Compression Conference, 1998, pp. 533, Mar/Apr 1998.

[12] Ling Sun Tan, Sei Ping Lau, Chong Eng Tan, "Optimizing LZW text compression algorithm via multithreading programming," Communications (MICC), pp. 592-596, Dec 2009.

[13] ZLIB, DEFLATE Algorithm "An Explanation of the Deflate Algorithm", http://www.zlib.net/feldspar.html .

Fahad Ullah was born in Karak, Pakistan in 1989. He has completed his B.Sc degree in Electrical Engineering from University of Engineering and Technology (UET), Peshawar, Pakistan. He has worked as a research internee for over a year on a project funded by Daimler Chrysler, USA. MOL (Hungarian Oil and Gas, PLC) has given him MOL Technical Scholarship 2008-2010. He is a member of IEEE and National Space Society (NSS). His research interests include image processing and compression, video compression and radio astronomy.

Email: ddspliting@gmail.com

Muhammad Naveed was born in Kohat, Pakistan in 1988. He has completed his B.Sc degree in Electrical Engineering (with majors in communication) from University of Engineering and Technology (UET), Peshawar, Pakistan in 2010. He has several internationally recognized certifications. He is CCNA (Cisco Certified Network Associate) and CCNAS (Cisco Certified Network Associate – Security) and was a student of Cisco Networking Academy to earn these certifications. He also has "Juniper Networks" associate and specialist certifications: JNCIA-ER, JNCIS-ER, JNCIA-EX, JNCIA-JUNOS, JNCIS-ES and JNCIS-SEC.

He was part of a research and development project titled "Pilot Research Study on Zero Flow Power Generation System" with Directorate of Science and Technology, Khyber Pukhtoonkhwa, Peshawar, Pakistan – a government organization. The project was aimed to relieve energy problems in the region. His research paper titled "Low Cost Crypto Core" is published in Pakistan Higher Education commission recognized X Category Science/Multidisciplinary journal – Sarhad Journal of Agriculture. One of his papers titled "Network Intrusion Prevention by Configuring ACLs on the Routers, based on Snort IDS alerts" is submitted for review. His research interests include information security, cryptography, cryptanalysis, steganography and computer networks.

Mr. Muhammad Naveed is the member of IEEE, IEEE Computer Society, IEEE Information Theory Society, IEEE Communication Society, Association for Computing Machinery (ACM), International Association of Computer Science and Information Technology (IACSIT), Sun Microsystems Open Source University Meetup (OSUM) and American Society for Mechanical Engineers (ASME). He is also the recipient of MOL (Hungarian Oil and Gas, PLC) Technical Scholarship 2008 – 2010.

Email: mnaveed29@gmail.com, mnaveed@ieee.org

Dr. Mohammad Inayatullah Babar received his Master and doctorate degrees from the School of Engineering, George Washington University, Washington DC, USA in 2005. His primary doctoral research was based on issues in Mobile Ad Hoc Networking including Quality of Service and Security. Dr. Mohammad Inayatullah Babar did his B.Sc Electrical Engineering from NWFP UET Peshawar in 1997. Due to his excellent academic credentials as he secured first positions in all four years of Engineering, he received Presidential award "Aizaz E Sabqat" in Year 2000. He also received University Gold Medal as Best Graduate and Siemens Gold Medal as Best Engineering Graduate from NWFP in Year 1998.

During his PhD in George Washington University, he was involved in number of research projects in the area of Mobile Ad Hoc Networks. Due to his research contribution in the area of Mobile Networks and Bio-Informatics, he received "Youngest Researcher Award" from MCOS Foundation in Washington DC in 2003. He also taught Telecommunication Engineering Courses at Graduate Level in School of Engineering, Stratford University, Virginia USA.

Dr. Mohammad Inayatullah Babar has more than thirty six publications in Engineering and Computing Conferences and Journals of International Repute. He is also a member of ACM USA and IEEE USA and has the honour to chair a conference Session in International ACM conference in USA in Year 2004.

Currently, he is working as Associate Professor in Department of Electrical Engineering and also as a Project Director of Information Service Center, UET, Peshawar.

Email: babar@nwfpuet.edu.pk

Faisal Iqbal was born in Karak, Pakistan, in 1987. He received his B.Sc in Electrical Engineering (with majors in communication) from University of Engineering and Technology, Peshawar (UET), Pakistan in 2009. He is currently pursuing his M.Sc in Electrical Engineering (with majors in communication) from the same university. He has worked as a lecturer in Faculty of Electrical Engineering at Sarhad University of Science & Information Technology (SUIT), Peshawar, Pakistan. He is currently serving as a lecturer in Department of Electrical Engineering, University of Engineering and Technology, Peshawar, Pakistan. His research interests are computer networks, information security, image processing and digital electronics.

Email: faisaliqbal@nwfpuet.edu.pk