ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 2, August 2013

Guarding Images using a Symmetric key Cryptographic Technique: Blowfish Algorithm

Dr. J. Abdul Jaleel, Jisha Mary Thomas

Abstract*— In this modern century, we are living in, Internet has grabbed all the fields of human life and established its root deeply. Virtually every individual in the world today are connected to every other with the global acceptance of internet. People living in different location or even in different countries are able to exchange their data, news whether happiness or sadness, images, videos etc. Not simply sharing, but it enables them to see each other also. Internet has a wide role in strategic and non-strategic areas like educational institutions, banks, military and war fields, factories, medical and paramedical areas, companies etc where there are needs to sent their information, both confidential and general (data and images) through communication paths. Apart from all the advantages of internet, it has created a sense of insecurity for its users that the sender, after sending a data should fear until he gets an acknowledgment from the opposite side informing they have received the data safely, that too without any manipulation in its content. Its failure may result in great disasters. Thus, the confidentiality, authentication, integrity, non-repudiation of the information (data or image) should be ensured. These objectives can be met with cryptography which is simply the science of securing sensitive and confidential information as it is stored on media or transmitted through communication network paths. Here, in this work, images are considered with an aim to secure them during its storage and transmission. This is achieved using Blowfish Algorithm, a type of symmetric key cryptography. The two processes, encryption and decryption together form the cryptographic process. For ensuring security, the images are encrypted by the sender before transmitting them and are decrypted by the receiver after receiving them so that only the sender and the intended person can see the content in the image. Blowfish algorithm which uses a key of variable size up to 448 bits simply iterates the function 16 times (Feistel network). The work here, is done by considering two images, out of which, one is a gray scale image of Lena and the other is a color image of a cute baby. The image processing is done using MatLab and the Blowfish encryption-decryption is performed using the VHDL (Very Large Scale Integrated Circuits Hardware Description Language) platform Xilinx ISE 10.1.

***Index Terms*— Blowfish Algorithm, Cryptography, Decryption, Encryption, Feistel Network, Image Processing, MatLab, Symmetric Key Cryptography, VHDL.**

I. INTRODUCTION

In this modern world, data, images, documents are stored more in computers, hard disk, compact disks, etc and a very less amount as papers in files. These data and images stored in computers, when needed, are to be transmitted over short and long distances through both secure and insecure computer networks for various applications. During their transmissions, there are chances for these highly confidential data to fall into wrong hands, thus leading to dangerous situations. Cryptography provides a solution for this problem. Cryptography can be defined as the art of safeguarding documents and it makes sure that only the intended people are able to visualize its content. Every security system must provide a bundle of security functions that can assure the secrecy of the system. These functions are usually referred to as the goals of the security system. The five main goals behind using Cryptography include Confidentiality, Authentication, Integrity, Non-Repudiation, Service Reliability and Availability. These objectives ensures that the private data remains private, the data is not altered illegally and assures against a party denying a data or a communication that was initiated by them. There are basically two types of cryptography namely secret key cryptography and public key cryptography. In secret key cryptography, both the sender and receiver know the same secret code called key. Messages are encrypted by the sender using the key and the receiver decrypts it using the same key. In public key cryptography, sender and receiver uses different key for encryption and decryption. The sender encrypts the data using a public key and this key will be known by all the parties included in the communication. The receiver decrypts the data using a private key and it should be kept as a secret. The word key is unavoidable term which can be a word, number or a phrase. Knowing the algorithm without the key does not help the hacker untangle the information. Encryption is the process by which information is transformed to an unreadable form called cipher text where its contents are hidden to eavesdroppers. Any person who sees the cipher text will not be able to determine anything about the original message. An encryption scheme usually needs a key generation algorithm to randomly produce keys. Decryption is the process of retrieving the original data back from the cipher text. A decryption process is generally the reverse of encryption process. Both processes make use of corresponding key. Longer the encryption key is, the more difficult it is to decode.

II. SYMMETRIC KEY CRYPTOGRAPHY

Symmetric key cryptography is sometimes referred to as conventional cryptography or secret key cryptography. Here, the sender and the receiver will both have a common secret key. But both the parties must agree upon the key before any transmission begin, and nobody else should know about it. Sender and receiver only have to specify the shared key in the beginning and then they can begin to encrypt and decrypt messages between them using that key. The strength of modern secret key encryption methods rests in the secrecy of the encryption key, not in the algorithm being used. The two

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 2, August 2013

different types of secret key cryptography are discussed below:

A. Block Cipher

A block cipher is a deterministic algorithm which operates on fixed-length groups of bits, called blocks, with an un-varying transformation that is specified by a symmetric key. Block ciphers take a number of bits and encrypt them as a single unit, padding the plaintext, so that it is a multiple of the block size. Blocks of 64 bits have been commonly used.

B. Stream Cipher

A stream cipher is a symmetric key cipher where plaintext digits are combined with a pseudorandom cipher digit stream (key stream). In a stream cipher, each plaintext digit is encrypted one at a time with the corresponding digit of the key stream, to give a digit of the cipher text stream. An alternative name for stream cipher is state cipher, as the encryption of each digit is dependent on the current state.

III. BLOWFISH ALGORITHM

Blowfish Algorithm is a symmetric block cipher, designed by Bruce Schneier in 1993, that can be effectively used for encryption and safeguarding of data. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data. Blowfish Algorithm is a Feistel Network, iterating a simple encryption function 16 times. The block size is 64 bits, and the key can be of any length up to 448 bits. Although there is a complex initialization phase required before any encryption can take place, the actual encryption of data is very efficient on large microprocessors. The salient features of Blowfish algorithm which made select it in my work are as follows:

- It manipulates data in large blocks
- It has a 64-bit block size.
- It has a scalable key, from 32 bits to at least 256bits.
- It uses very simple operations like addition and XOR addition.
- It uses a design that is simple to understand. This facilitates analysis and increase the confidence in the algorithm.
- It is fast as this algorithm rate on a 32-bit microprocessor is 26 clock cycles per byte.
- It is compact as it can be executed in less than 5kb memory.

Blowfish symmetric block cipher algorithm encrypts block data of 64-bits at a time. It follows the Feistel network and this algorithm is divided into two parts namely Key Expansion part and Data Encryption part [1], [6].

C. Key Expansion

It converts a variable-length key of at most 56 bytes (448 bits) to several subkey arrays totalling 4186 bytes. Keys are generated before the image encryption and decryption process. There is a P array and four 32-bit S boxes. The P array contains 18 32-bit subkeys and out of four S boxes, each S box contains 256 entries. Fig.1 shows the schematic flow of key generation.

The generation of F function is described below.

- The input XL which is 32 bits divided into four quarters, each of 8 bit a, b, c, d.
- Each quarter enters S-boxes 1, 2, 3and 4 as shown in figure.
- The corresponding 32 bits outputs undergo the operation XOR and addition modulo 2^{32}.
- The final output is 32 bits key.

The generation of Key can be mathematically represented as,

$$F = \{(S1[a] + S2[b]) XOR (S3[c] + S4[d])\} \quad (1)$$

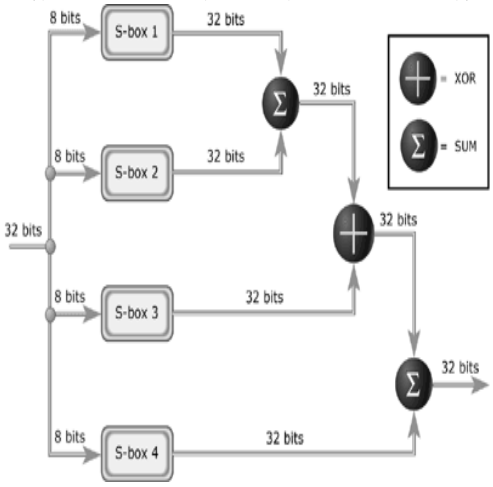

Fig. 1 Schematic Representation of Key Generation

The key so formed is then used to generate eighteen 32-bits subkeys and four 8×32 S-boxes containing 1024 bits (4186 bytes) entries. The subkeys are stored in P-array (Permutation box). There are four S-boxes (Substitution box) each with 256 32-bit entries. Thus, the user key is converted into P-array and s-array. These arrays need not be recomputed as long as the key is not changed. But they must be kept secret.

D. Data Encryption

Blowfish algorithm uses a Feistel network for data encryption which iterates the function 16 times. Each round includes a key dependent permutation and data dependent substitution. Fig.2 shows the overall process of data encryption and the different steps in encryption are described below:

- Split the 64 bit block into two equal blocks having 32 bits size each (XL and XR).
- The left block XL is XOR'd with first element of P-block, P_1 and thus obtained result is fed to the F function.
- In the F function block, substitution operation is carried out where the given 32 bit input is transformed into another 32 bit output.
- The output from F block is XOR'd with right half XR and the results obtained are swapped as shown in the Fig.2.
- After completing each round successfully, the so formed right half become the new left half or vice versa.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 2, August 2013

- These steps are continued up to 16 rounds.
- The final left and right halves are not swapped but XOR'd with seventeenth and eighteenth P box elements.
- So obtained result is the cipher text which is non understandable to outsiders and attackers.

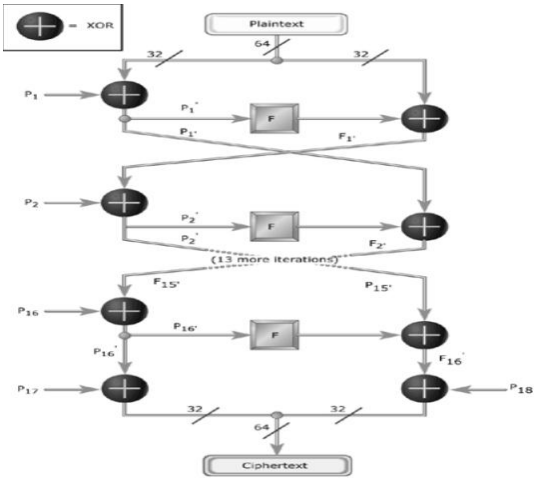

Fig.2 Feistel Structure of Blowfish Algorithm

The process of decryption is the reverse process of encryption. Because Blowfish is a symmetric algorithm, the same procedure is used for decryption as well as encryption. The only difference is that the input to the encryption is plaintext and output is cipher text, but for decryption, the input is cipher text and output must be the plaintext. During Decryption, the cipher text block of 64 bits is divided into two halves of 32 bit each. The subkeys are all used in the reverse order as in the encryption process. That is, the process starts with the last elements of P array (P_{17} and P_{18}) and ends with its first element (P_1).

IV. IMAGE ENCRYPTION AND DECRYPTION FROM PIXEL VALUES

The Image Processing part is done using MatLab software of version 7.5 (R2007b). It includes the visualization of images before and after encryption and decryption. The encryption and decryption part is done using VHDL platform, Xilinx ISE 10.1.

The various stages of work done are as follows:

- View the input image taken to encrypt.
- Collect the pixel values of the viewed image.
- Provide the pixel values to VHDL encryption and decryption coding.
- Collect the encrypted and decrypted pixel values.
- Provide these pixel values to visualize the encrypted and decrypted images.
- Verify that the decrypted image is an exact replica of the input image.

A. Viewing the input image using MatLab

The image processing part is done using MatLab of version 7.5.0 (R2007b). Here, the input to be encrypted is viewed. Two images are considered with an aim to perform encryption and decryption. It includes a gray scale image of Lena in PNG format and a color image of a cute little baby in JPEG format. The various steps performed during input visualization are as follows:

- Firstly, a color image of Lena is considered which is of dimension 220×220 pixels.
- This image is resized to a dimension of 64×64 pixels.
- The color image is then converted into gray scale image.
- Finally, it is transformed into an unsigned 8-bit integer.
- The pixel values of the input image are collected. Since the image was of size 64×64 pixels, the collected pixel values from MatLab window will be in a matrix form of dimension 64×64 (64 rows and 64 columns), thus totaling 4096 pixel values.

Fig.3 shows the visualization of the input image of Lena that is to be encrypted. The Fig.3 itself consists of four images in it and they are explained below:

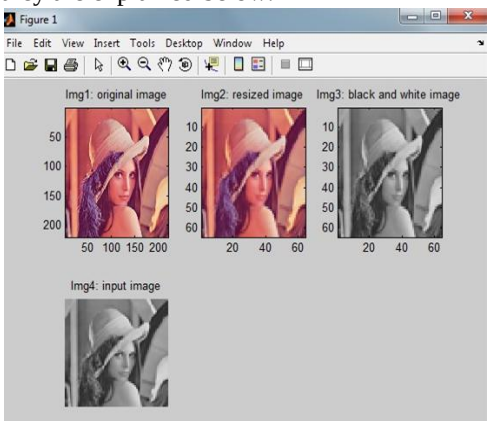

Fig. 3 Lena (input 1) Image Visualization using MatLab

Img1: This represents the original image of Lena of size220×220 pixels. This image is read from the memory using the MatLab command "imread".

Img2: This represents the resized image of size 64×64. This image will have pixel values in a matrix form with 64 rows and 64 columns. This is done using the command,

im= imresize (im, [64 64])

Img3: The image represents the gray scale image developed from its color image and is done using the command,

im = rgb2gray (im)

Img4: This represents the image formed when Img3 is passed through built-in-function, uint8 and the command is,

im = uint8 (im)

In all these above mentioned command of MatLab, "im" is used, because initially I used "im" while reading the image to memory and hence the pixel values of the original image was stored in "im". The same commands and the same MatLab code was used to visualize the second image (colour image of a baby) that I chose to encrypt and decrypt. The image chosen was a colour image. The result seen through MatLab when the second image is read is given by Fig.4 and it also has 3 images

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 2, August 2013

in it as explained before (but no colour to gray scale conversion).

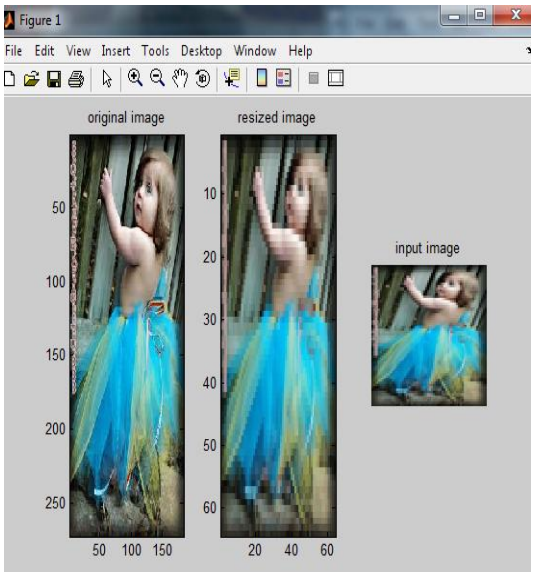

Fig. 4 Baby (Input 2) Image Visualization Using Matlab

B. Encryption and Decryption using VHDL

In blowfish algorithm, the input is processed as a group of bytes that are fixed in size (like 64, 128 or 256) long for encryption and decryption. Here, we consider the input as 64 bits long. In this work, the pixel values are encrypted one by one. Each pixel is 1 byte (8 bits). Hence in order to make it 64bits, we add 56 zeros to it before passing it to the Feistel network. The pixel values are temporarily read to a variable till 64 bits are read and each bytes or each pixel is encrypted. Since the taken image is a 64×64 sized image, this process is continued till 4096 pixel values are read. The flowchart representing the encryption process is given by Fig.5. The encryption process is coded according to the flowchart given (decryption process will be the reverse of this given encryption process where in decryption process, the elements of P-array are used from the last to the first i.e. it starts with (P_{17}, P_{18}) and ends with P_1. The encryption and decryption code is written in VHDL. Once we complete writing the code, we need to test its working capability. One method of testing it is by writing Test bench. Every entity must need a test bench and any design without test bench is useless. The output is seen through I Sim. The Fig. 6 shows the result of the VHDL coding for encryption and decryption of Lena's image (input 1) and is seen using ISim. This figure shows all the original input pixel values (red colour), each pixel value considered for encryption (pink colour), encrypted pixel values (yellow colour) and decrypted pixel values (blue colour).

The Fig. 7 shows the result of the VHDL coding for encryption and decryption of baby's image (input 2) and it also is seen using ISim. This figure shows all the original input pixel values (red colour), each pixel value considered for encryption (pink colour), encrypted pixel values (yellow colour) and decrypted pixel values (blue colour).

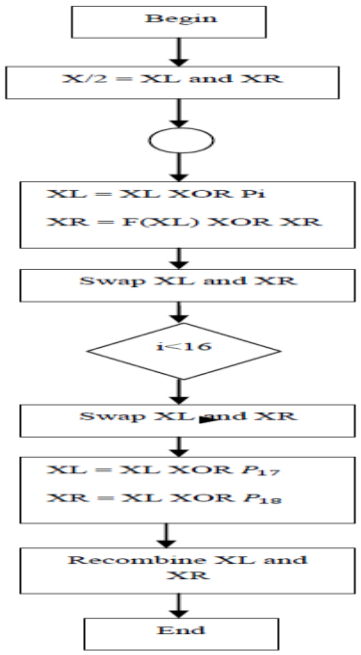

Fig.5 Flowchart for Blowfish Algorithm

C. Output Image Visualization using MatLab

The encrypted pixel values of Lena's image are collected directly from the ISim figure and are fed to the MatLab coding so as to view the encrypted image. Since there were a total of 4096 pixel values (from the input image) to be encrypted, the output will also have a total of 4096 pixel values. The encrypted pixel values seen through ISim were copied down to notepad as text for it is easy to read from it using MatLab command "text scan". The different steps performed during encrypted image visualization include:

- The pixel values are entered in a matrix form of size 64×64 pixels (64 rows and 64 columns) to MatLab in text form as strings.
- The pixel values are read using MatLab command.
- Since here the pixel values were written in hexadecimal format, it is being converted in to decimal form.
- The image is viewed as a 64×64 sized image.

Fig. 8 shows the encrypted image obtained when the input to MatLab coding was the encrypted pixel values of Lena's image (input 1) obtained from VHDL coding result.

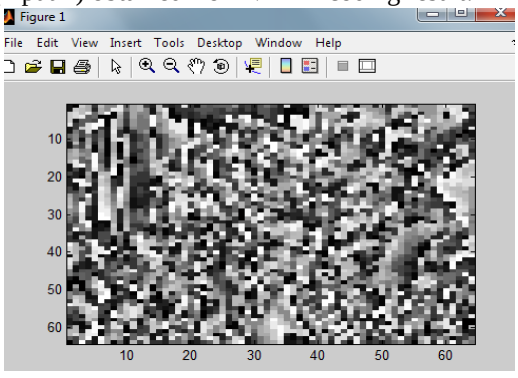

Fig. 8 Lena's (Image 1) Encrypted Image

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 2, August 2013

The same steps were carried out with the collected encrypted pixel values of the baby (image 2) so to obtain the Baby's encrypted image and the result is shown as in Fig.9

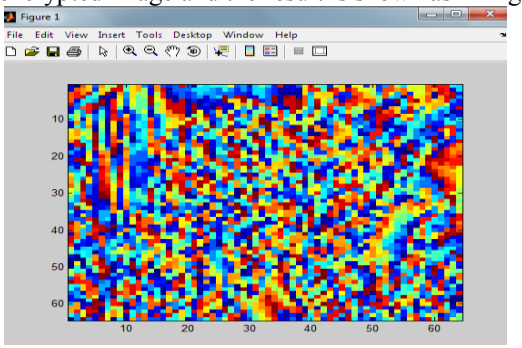

Fig. 9 Baby's (Image 2) Encrypted Image

The decrypted pixel values of Lena's image (image 1) are collected and fed to the MatLab coding so as to view the decrypted image. Since there were a total of 4096 pixel values (from the input image) to be decrypted, the output will also have a total of 4096 pixel values. The decrypted pixel values seen through ISim were copied down to notepad as text for it is easy to read from it using MatLab command "textscan". Almost the same steps were carried out to visualize the decrypted image also and Fig. 10 shows the decrypted image of Lena.

Fig. 10 Lena's (Image 1) Decrypted Image

Similarly the decrypted pixel values of Baby (Image 2) were collected and the same steps were followed to view the Baby's decrypted image and the result is shown by Fig. 11.

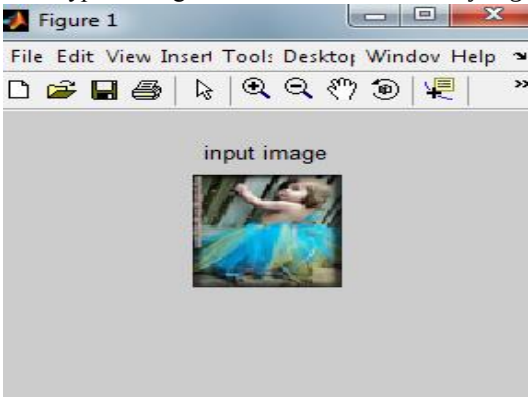

Fig. 11 Baby's (Image 2) Decrypted Image

V. CONCLUSION

The inputs selected with an aim to encrypt and decrypt include a gray scale image of Lena in PNG format and a color image of a cute baby in JPG format. The image processing part was done using MatLab and the encryption-decryption part was coded using VHDL. The results show that the encrypted image provides no information about the original image and the decrypted image is almost an exact replica of the input image. The Blowfish algorithm is strong and immune to hacking as it encrypts the data by a 16 round function iterating Feistel network.

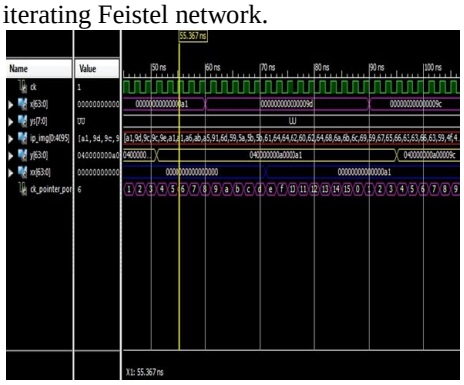

Fig. 6 VHDL Coding result for Lena's 1mage

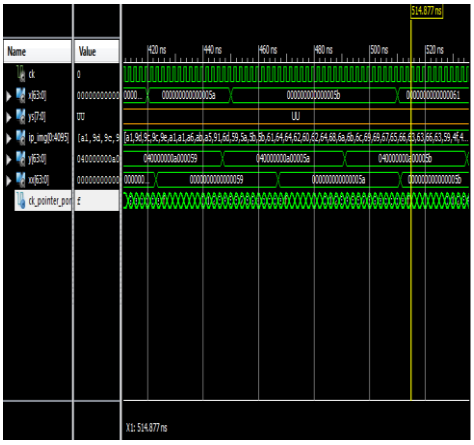

Fig.7 VHDL Coding Result for Baby's Image

VI. ACKNOWLEDGMENT

The authors would like to thank the management and the faculty members of the Department of Electrical and Electronics Engineering and Department of Electronics and Communication Department of TKM College of Engineering, Kollam, for many Insightful discussions, Timely suggestions and the facilities extended to us for the successful completion of the task.

REFERENCES

[1] Irfan Landge, Burhanuddin Contractor, Aamna Patel and Rozina Choudhary "Image encryption and decryption using Blowfish algorithm" Proceedings of the 2012 National Conference of Emerging Trends in Information Technology, Shirpur, Maharashtra, April 21 , 2012.

[2] Pooja Mishra, and Biju Thankachan, "A survey on Various Encryption and Key Selection Techniques", IJEIT, vol. 2, pp. 141-145, January 2013.

[3] Ratinder Kaur and V.K. Banga, "Image Security using Encryption based Algorithm", ICTEEP, Singapore 2012.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 2, August 2013

[4] Deepak Kumar Dakate and Pawan Dubey, "Blowfish Encryption: A Comparative Analysis using VHDL", IJAET, vol. 1, pp. 2249-8958, 2012.

[5] Ranjeet singh and Madan Kumar, "Linux based Encryption quality and Security valuation of Blowfish Algorithm and its modified version using Digital images", IJARCSSE, vol. 2, pp. 240-245, June 2012.

[6] Dr. V Ramaswamy and Krishnamurthy G N, "Performance Analysis of Blowfish and its Modified Version using Encryption quality, Key sensitivity", International Journal of Recent Trends in Engineering, vol. 1, pp, 1-4, May 2009.

AUTHOR'S PROFILE

Prof. Abdul Jaleel. J received the Bachelor degree in Electrical Engineering from University of Kerala, India in 1994. He received the M.Tech degree in Energetics from Regional Engineering College Calicut, Kerala, India in 2002, and PhD from WIU, USA in 2006.

He joined the EEE department of TKM College of Engineering as faculty member in 1990. He was with Saudi Aramco in 1996 to 1998 and worked in the field of power generation, transmission, distribution and instrumentation in the Oil and Gas sector of Saudi Arabia. He was with Water Supply department of Sultanate of Oman in 1985 to 1986 and worked with the maintenance of Submersible bore-well pumps and power supplies. He was with Saudi Electricity Company in 1979 to 1985 and worked in the Generation, Transmission and distribution fields. He worked with project management, Quality Management and he is a certified Value Engineer and Auditor for QMS. He is a consultant for Oztern_Microsoft, Techno Park, Kerala and Consultant for Educational Projects of KISAT and MARK Research and Education Foundation. He was P.G. Coordinator of M. Tech Programme in the TKM College of Engineering under University of Kerala and Director of Kerala Institute of Science and Technology. Currently he is working as Principal at Al Azhar College of Engineering & Technology, Kerala. His areas of interest are power system Control and optimization, power system reliability, voltage stability, computer aided design and analysis.

Jisha Mary Thomas received B.Tech Degree in Electronics and Communication from Caarmel Engineering College, Perunad, Pathanamthitta, India in 2011. Currently she is pursuing M.Tech in Industrial Instrumentation and Control at Thangal Kunju Musaliar College of Engineering, Kollam, India.