*Journal of Embedded Systems, 2015, Vol. 3, No. 1, 11-15*
Available online at http://pubs.sciepub.com/jes/3/1/2
© Science and Education Publishing
DOI:10.12691/jes-3-1-2

# Multilevel Network Security Combining Cryptography and Steganography on ARM Platform

**Pallavi H. Dixit[1,*], Kamalesh B. Waskar[1], Uttam L. Bombale[2]**

[1]Electronics and Telecommunication, Bharati Vidyapeeth College of Engineering, Kolhapur, India
[2]Electronics, Shivaji University, Kolhapur, India
*Corresponding author: dixit.pallavi@rediffmail.com,

*Received November 18, 2014; Revised February 28, 2015; Accepted March 15, 2015*

**Abstract** This paper presents two level data security in Network system. Cryptographic algorithm BLOWFISH and Steganography algorithm List significant Bit (LSB) are used for data security. Confidential information is encrypted by BLOWFISH algorithm, and then encrypted data hide into image by LSB algorithm of Steganography. For more security we used iris image of authorized person to hide encrypted data. The keys required for BLOWFISH algorithm is generated from same iris image. These two algorithms implemented on 32 bit ARM 7. In the result of project include memory utilization, processing time for encryption and decryption etc. this project gives better security for embedded systems like mobile, smart card, ATM etc.

***Keywords:*** *network security, blowfish, cryptography, embedded system, list significant bit, steganography*

**Cite This Article:** Pallavi H. Dixit, Kamalesh B. Waskar, and Uttam L. Bombale, "Multilevel Network Security Combining Cryptography and Steganography on ARM Platform." *Journal of Embedded Systems*, vol. 3, no. 1 (2015): 11-15. doi: 10.12691/jes-3-1-2.

## 1. Introduction

Many embedded systems depend on obscurity to achieve e-mail from being read by someone other than the intended recipient, keep firmware upgrades out of devices they don't belong security, Modern embedded systems need data security more than ever before. Our PDAs store personal e-mail and contact lists; GPS receivers and, soon, cell phones keep logs of our movements and our automobiles record our driving habits. On top of that, users demand products that can be reprogrammed during normal use, enabling them to eliminate bugs and add new features as firmware upgrades become available.

Data security helps keep private data private. Secure data transmissions prevent contact lists and personal in, and verify that the sender of a piece of information is who he says he is. Data security techniques have a reputation for being computationally intensive, mysterious, and fraught with intellectual property concerns. Whereas some of this is true, straightforward public domain techniques that are both robust and lightweight do exist. One such technique, an algorithm called Blowfish, is perfect for use in embedded systems.

Cryptography and Steganography are widely used techniques that manipulate information in order to cipher or hide their existence. These techniques have many applications in computer science and other related fields. They are used to protect e-mail messages, credit card information, corporate data etc. Steganography is the art and science of communicating in a way which hides the existence of the communication [1]. A Steganography system thus embeds hidden content in unremarkable cover media so as not to arouse an eavesdropper's suspicion [2]. For example it is possible to embed a text inside an image or an audio file. On the other hand, cryptography is the study of mathematical techniques related to aspects of information security such as confidentiality, data integrity, entity authentication, and data origin authentication [3]. In this paper we will focus only on confidentiality. Cryptography and Steganography are cousins in the spy craft family: the former scrambles a message so it cannot be understood; the latter hides the message so it cannot be seen.

The aim of this paper is to describe a method for integrating together cryptography and Steganography through image processing. In particular, we present a system able to perform Steganography and cryptography at the same time.

In this paper, both Cryptography and Steganography methods are used for data security over the network. IRIS is considered to be the most trusted and unique feature of the person. Hence this project proposes a data encryption technique using IRIS biometric. IRIS images are taking from IRIS biometric database. ARM processor is used for processing Steganography and cryptography algorithms.

## 2. Related Work

"Iris Biometric Cryptography for Identity Document", this paper present an approach to generate a unique and more secure cryptographic key from iris template. The iris images are processed to produce iris template or code to be utilized for the encryption and decryption tasks. AES cryptography algorithm is employed to encrypt and decrypt the identity data [5].

Secondly" Two New Approaches for Secured Image Steganography Using Cryptographic Techniques and Type Conversions" This paper give information about Cryptography & Steganography, This paper introduces two new methods wherein cryptography and Steganography are combined to encrypt the data as well as to hide the encrypted data in another medium so the fact that a message being sent is concealed [6].

Next paper is "A New Image Steganography Technique" it includes various image Steganography techniques like Text-Based Steganography, Audio Steganography, Steganography in OSI Network Model, Image Steganography etc. [7]

"Multilevel Network Security Based on Iris Biometric", In this paper A novel security Mechanism is developed here for high security networks by combining IRIS biometric techniques with cryptographic and Steganography mechanisms [9].

## 3. Methodology

There have been many different encryption algorithms and public key cryptographic methods are being proposed to provide security to such data. All of these algorithms depend upon a user's key which he uses as the key for encryption. But these keys may be hacked by hacker, hence the only feature or data of a person that hackers cannot hack is their biometric features, hence this proposed project consider IRIS image of a user to generate secrete key for encryption.

For security, only encryption may not be enough, hence proposed project include combination of both cryptography and Steganography. The encrypted data hide into the image and then image is transmitted in the network.

There is some weakness in hiding information in images; that is adversary could easily detect the confidential message, by noticing the noise and clarity of the image's pixels, also by observing the difference between the embedded image and the original one if it is known to him. In the proposed project, here use Iris images instead of images that contain faces or natural scenes, because the only feature or data of a person that hackers cannot hack is their biometric features.

Block diagram:

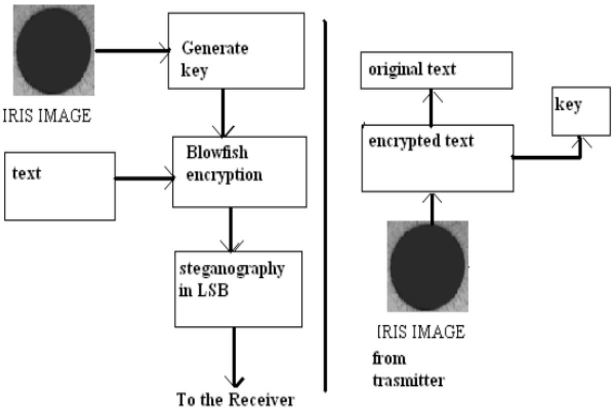


**Figure 1.** functional block diagram

Steps are

Generated key from iris image, we have taken only iris part of eye of person for more security. Key length is 128 bits.

Using Blowfish algorithm for encryption, the confidential information is encrypted.

$$\text{Cryptography} = \text{Text} + \text{key}$$

This encrypted text then hides into every pixel of iris image.

$$\text{Steganography} = \text{Text} + \text{image}$$

4. Iris image is transmitted to receiver, at the receiver side, hidden data removed from image and using same encrypted key, original data recovered from encrypted text.

# 4. Overview of algorithm

## 4.1. Image Definition

To a computer, an image is a collection of numbers that constitute different light intensities in different areas of the image [9]. This numeric representation forms a grid and the individual points are referred to as pixels. Most images on the internet consists of a rectangular map of the image's pixels (represented as bits) where each pixel is located and its color [10]. These pixels are displayed horizontally row by row. The number of bits in a color scheme, called the bit depth, refers to the number of bits used for each pixel [11]. The smallest bit depth in current color schemes is 8, meaning that there are 8 bits used to describe the color of each pixel [11].

## 4.2. Least Significant Bit Algorithm

Least significant bit (LSB) insertion is a common, simple approach to embedding information in a cover image [3]. The least significant bit in other words, the 8th bit of some or all of the bytes inside an image is changed to a bit of the secret message. When using a 24-bit image, a bit of each of the red, green and blue color components can be used, since they are each represented by a byte. In other words, one can store 3 bits in each pixel. An 800 × 600 pixel image, can thus store a total amount of 1,440,000 bits or 180,000 bytes of embedded data [7]. For example a grid for 3 pixels of a 24-bit image can be as follows:

$$\left(0010110\mathbf{1}0001110\mathbf{0}1101110\mathbf{0}\right)$$

$$\left(1010011\mathbf{0}1100010\mathbf{0}0000110\mathbf{0}\right)$$

$$\left(1101001\mathbf{0}1010110\mathbf{1}0110001\mathbf{1}\right)$$

When the number 200, which binary representation is 11001000, is embedded into the least significant bits of this part of the image, the resulting grid is as follows:

$$\left(0010110\mathbf{1}00011101\ 11011100\right)$$

$$\left(1010011\mathbf{0}11000101\ 00001100\right)$$

$$\left(1101001\mathbf{0}10101100\ 01100011\right)$$

Although the number was embedded into the first 8 bytes of the grid, only the 3 underlined bits needed to be changed according to the embedded message. On average, only half of the bits in an image will need to be modified to hide a secret message using the maximum cover size [7]. Since there are 256 possible intensities of each primary color, changing the LSB of a pixel results in small changes in the intensity of the colors. These changes cannot be perceived by the human eye - thus the message is successfully hidden. With a well-chosen image, one can even hide the message in the least as well as second to least significant bit and still not see the difference [3].

### 4.3. Blowfish

Blowfish is a symmetric encryption algorithm, meaning that it uses the same secret key to both encrypt and decrypt messages.

A graphical representation of the Blowfish algorithm appears in Figure 2. In this description, a 64-bit plaintext message is first divided into 32 bits. The "left" 32 bits are XORed with the first element of a P-array to create a value I'll call P', run through a transformation function called F, then XORed with the "right" 32 bits of the message to produce a new value I'll call F'. F' then replaces the "left" half of the message and P' replaces the "right" half, and the process is repeated 15 more times with successive members of the P-array. The resulting P' and F' are then XORed with the last two entries in the P-array (entries 17 and 18), and recombined to produce the 64-bit ciphertext.

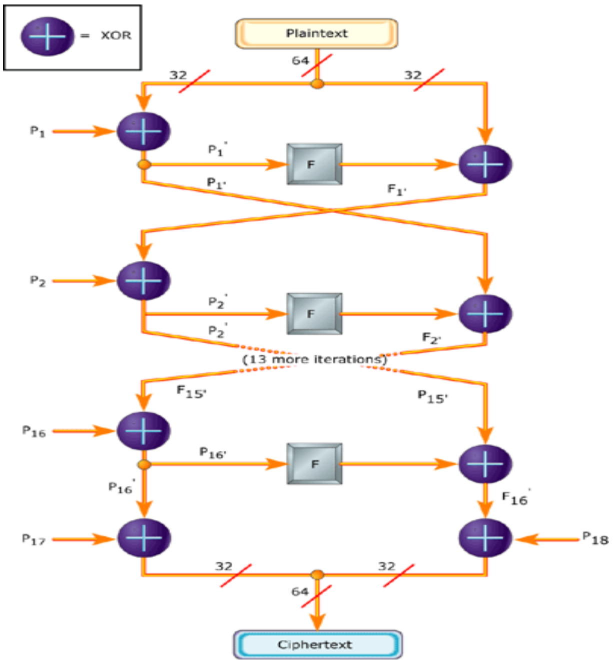


**Figure 2.** Blowfish algorithm

**Algorithm**:

```
The input is a 64-bit data element== x.
Divide x into two 32-bit halves: xL, xR.
Then,
for i = 1 to 16:
xL = xL XOR Pi
xR = F(xL) XOR xR
        Swap xL and xR
After the sixteenth round,
swap xL and xR again to undo the last swap.
Then,
                    xR = xR XOR P17 and
                    xL = xL XOR P18.
```

Finally, recombine xL and xR to get the ciphertext.

A graphical representation of F appears in Figure 2. The function divides a 32-bit input into four bytes and uses those as indices into an S-array. The lookup results are then added and XORed together to produce the output.

The P-array and S-array values used by Blowfish are precompiled based on the user's key. In effect, the user's key is transformed into the P-array and S-array; the key itself may be discarded after the transformation.

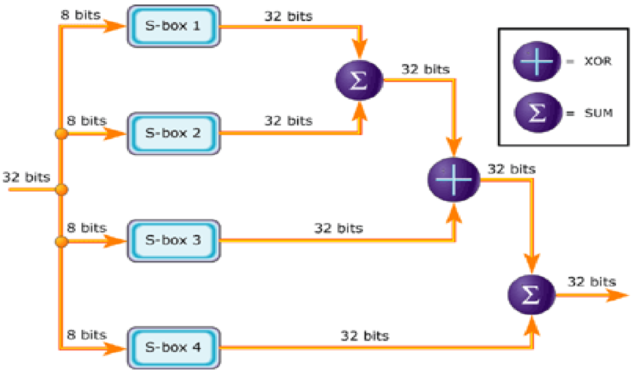


**Figure 3.** Graphical representation of F

### 4.4. Experimental Setup

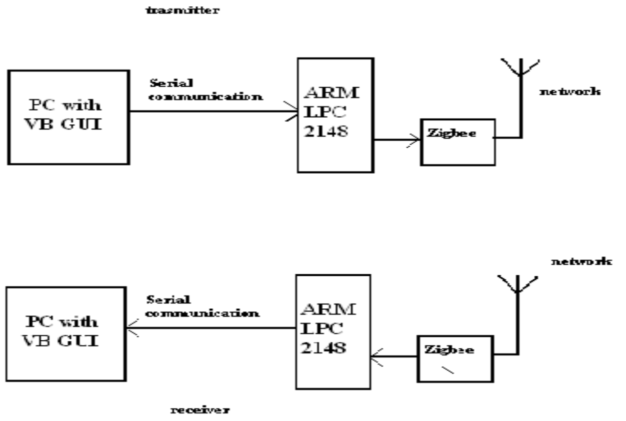


**Figure 4.** Experimental setup block diagram

PC must have Visual basic 6 software to run GUI. PC com port connected to ARM kit com Port. We used two UART port of ARM kit, one is connected to PC com port and second connected to Zigbee. So transmitter can acts as a receiver or receiver can acts as transmitter if requireed. As shown in Figure 4, for practical demonstration we required two PC or Laptops, two ARM kit, two zigbee module and two serial com cables.

## 5. Results

At transmitter side we are created GUI in Visual Basic 6, which can be used to transmit text and iris image to ARM kit. This GUI shown in Figure 7. After sending text and Image to microcontroller, LCD shows message Device is ready to receiver data from pc. Then Send text and image button pressed, then downloading of image and text done in RAM memory of ARM controller. When PC sends text and Image to Controller then controller is ready to receive data from computer.LCD display shows the message “receiving ready”.

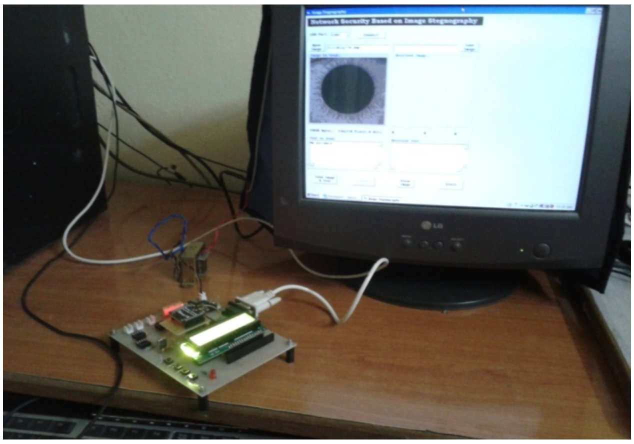
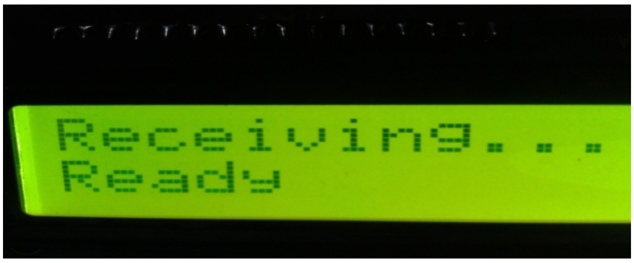


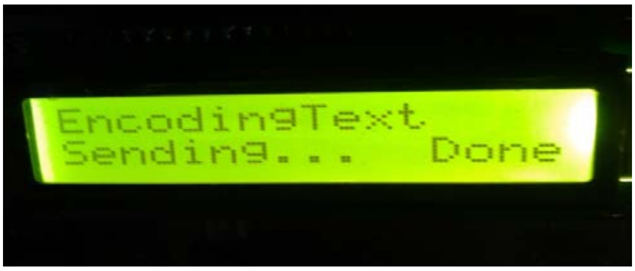
**Figure 5.** Experimental setup connection diagram transmitter side.



Receiving...
Ready

**Figure 6.** receiving status

After completion of successful reception of image and text to the controller. Then controller stat encoding and store encoded text in image. And stego image send to the UART1, where zigbee module is connected to controller. Whereas doing this process controller display message on LCD as below.



EncodingText
Sending... Done

**Figure 7.** Encoding status

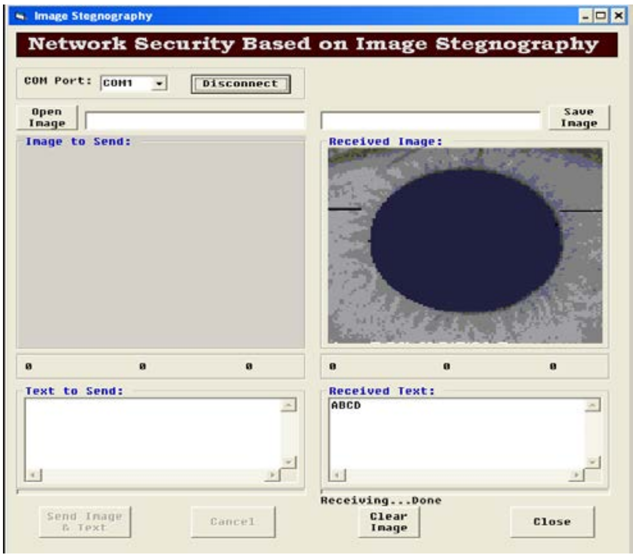


Image Stegnography

Network Security Based on Image Stegnography

COM Port: COM1

Image to Send:

Received Image:

Text to Send:

Received Text:
ABCD

Receiving...Done

**Figure 8.** Received image and text at receiver side

Reverse process takes place at the receiver side. Zigbee receive stego image and transit to the IC, decoding of image and encoding text take place then encoded text is converted into original text. Stego image display on received image block of GUI and original text is at bottom block.

Whole programming done in c code and compile in Keil 4, so timing analysis is possible. Timing analysis and memory utilization as shown in table below,

| | Blowfish algorithm | Least significant bit algorithm | Total |
|---|---|---|---|
| Encryption cycle | 1120 | 3227 | 4347 |
| Decryption cycle | 1119 | 3224 | 4343 |
| Memory utilization | 5KB | 18 KB | 23KB |

## 6. Conclusion

This paper is devoted to the problem and solution on security of small embedded system. Total ARM memory is utilized for processing of both algorithms. So this system can be used in small memory application like in smart cards, ATM machine etc. As the point of security, maximum security for text is possible so this system can be used in military application. Most confidential iris image of person consider for Steganography, so when iris image with hidden text is on network, and if hackers hack this image, then it is too difficult to catch the hidden data because iris image is unique identity for person, there is no another same image can be generated or captured. So this is the advantage.

Blowfish is a very secure algorithm. When we compared it with another algorithm AES then it is found that for embedded system security, blowfish is easier than AES. Blowfish required less processing time and memory utilization than AES. So it is a faster security algorithm for embedded system.

This project introduces two algorithms at a time for multiple securities, so maximum security can possible.

## References

[1] Johnson, Neil F. And Sushil Jajodia. “Exploring steganography: seeing the unseen.” IEEE computer, 32:2. 26-34. 1998.

[2] Proves, N. And Honeyman, P. “Hide and Seek: An Introduction to steganography.”,IEEE security &privacy, (2003).

[3] Menezes, A., Van Oorschot, P., and Vanstone, S. “Handbook of applied cryptography.” CRC Press, (1996).

[4] Hassan Mathkour, Batool AL-sadoon, ameur touir ” a new image steganography technique”.

[5] Sim hiew moi, nazeema binti abdul rahim,puteh saad, pang li sim, zalmiyah zakaria, subariah ibrahim, “iris biometric cryptography for identity document”, 2009 international conference of soft computing and pattern recognition.

[6] Sujay narayana1and gaurav prasad” two new approaches for secured image steganography using cryptographic techniques and type conversions” signal & image processing: an international journal (sipij) vol.1, no.2, december 2010.

[7] Mamta juneja 1, parvinder singh sandhu2 “designing of robust image steganography technique based on lsb insertion and encryption” 2009 international conference on advances in recent technologies in communication and computing.

[8] V.v.satyanrayanarayana tallapragada, dr. E.g.rajan, “multilevel network security based on iris biometric” 2010 international conference on advances in computer engineering.

[9] B. Schneier, applied cryptography, john wiley & sons, new york, 1994.

[10] B. Schneier, description of a new variable-length key, 64-bit block cipher (blowfish) fast software encryption, Cambridge security workshop proceedings (December 1993), springer-verlag, 1994, pp. 191-204.

[11] Zainul Abidin, Adharul Muttaqin, “A Simple Cryptography Algorithm for Microcontroller” in international journal of emerging technology and advanced engineering 2250-2459, iso 9001:2008 certified journal, volume 2, issue 12, December 2012

[12] Http://lcd-linux.sourceforge.net/pdfdocs/lcd1.pdf.

[13] http://lcd-linux.sourceforge.net/pdfdocs/lcd2.pdf.

[14] http://www.8051projects.net/lcd-interfacing/.