A Secure Network Communication Protocol Based on Text to Barcode Encryption Algorithm

Abusukhon Ahmad
Department of Computer Networks,
Al-Zaytoonah University of Jordan,
Amman, Jordan

Bilal Hawashin
Department of CIS, Al-Zaytoonah University of Jordan,
Amman, Jordan

***Abstract*—Nowadays, after the significant development in the Internet, communication and information exchange around the world has become easier and faster than before. One may send an e-mail or perform money transaction (using a credit card) while being at home. The Internet users can also share resources (storage, memory, etc.) or invoke a method on a remote machine. All these activities require securing data while the data are sent through the global network.**

There are various methods for securing data on the internet and ensuring its privacy; one of these methods is data encryption. This technique is used to protect the data from hackers by scrambling these data into a non-readable form. In this paper, we propose a novel method for data encryption based on the transformation of a text message into a barcode image. In this paper, the proposed Bar Code Encryption Algorithm (BCEA) is tested and analyzed.

***Keywords*—Encryption, Decryption; Algorithm; Secured Communication; Private Key; Barcode Image**

I. INTRODUCTION

Nowadays, many applications on the web allow users from the whole world to interact with them. These applications rely on securing the channels between the client and the server while sending data through the global network.

Securing a channel between a server and a client is handled using authentication (i.e. a username and a password) and one of the encryption algorithms.

There are different methods for data encryption, which are used to protect data over a network and thus build a secure channel. These techniques can be classified based on the data type (e.g. text, image, sound) of the encrypted data into three categories; namely, text encryption, image encryption, and sound encryption. Fig. 1 describes the encryption process for private-key encryption. As shown in Fig.1, the data encryption system consists of a plain text (also could be an image or a sound), which is the data before running the encryption algorithm. The encryption algorithm is the algorithm used to transfer the original data (e.g. text message) into an unreadable or a hidden form [1]. The core of the encryption algorithm is a private key used by both encryption and decryption algorithms. The encryption key is used to encrypt and decrypt data.

The decryption algorithm is an algorithm used for transforming the encrypted data into the original data [2], or simply, it is the encryption algorithm working in reverse.

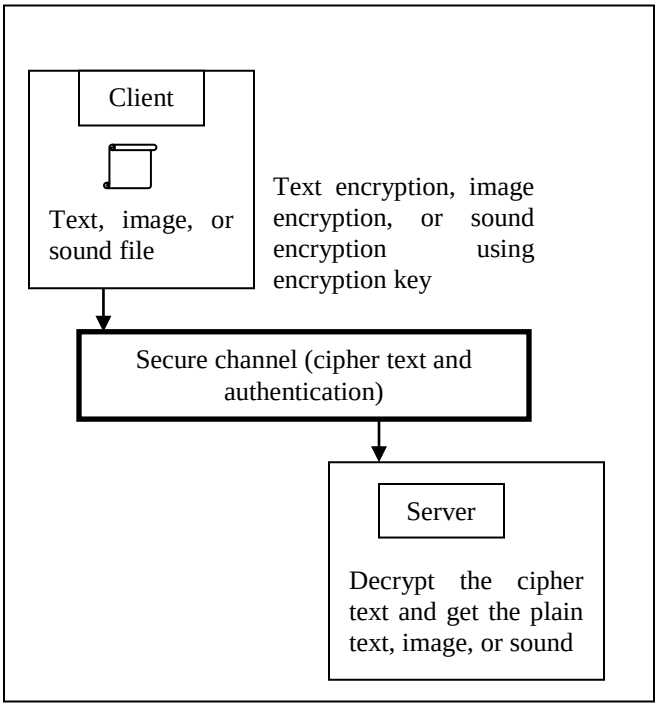

Fig. 1. Encryption process

The Internet is the richest area for hackers to perform their attacks. Hackers are unauthorized users who may attack sensitive data sent through the Internet and use false IP addresses to achieve various goals. Thus, in most of the Internet applications, verification and validation techniques are required to check the user's identity. These techniques include digital signature, and digital certificate [3]. Digital signature and digital certificate are not the focus of this research.

In general, the standard methods used for data encryption are private-key encryption (called symmetric encryption), public-key encryption (called asymmetric encryption), digital signature, and hash functions [4].

In private-key cryptography, both the sender and the receiver agree on a single key to be used for both encryption and decryption. This key is kept secret by sending it through a secure channel to the other side of the network [5].

This paper proposes a new encryption algorithm based on private-key techniques.

As it is mentioned earlier in this paper, an encryption key is used to encrypt images, text, and sounds. Image encryption techniques focus on encrypting a digital image in a specific format (e.g. png, bmp, etc.) into an unreadable image with the same image format using a specific encryption key.

One of the techniques used for image encryption is based on dividing the image into blocks and then encrypting those blocks using an encryption algorithm, the following are examples of this technique.

Nithin, Anupkumar, and Hegde [6] proposed and evaluated an image encryption algorithm (called FEAL) that is based on the DES encryption algorithm. The algorithm divides the original image into a number of blocks (16×16 blocks). Later encryption and decryption algorithms are performed using 12 keys of size 16-bit. Images used in this algorithm are gray scale images of size 256×256 resolution.

AliBaniYounes, and Janta [7] proposed an encryption algorithm based on dividing an image into blocks. These blocks are then rearranged into a transformed image (using their proposed transformation algorithm) and then the transformed image is encrypted using the Blowfish algorithm. Their work showed that increasing the number of blocks by decreasing the block's size resulted in a lower correlation and higher entropy.

Divya, Sudha, and Resmy [8] proposed a simple encryption algorithm based on dividing the image into 8 × 8 blocks. In their method, they proposed to encrypt a portion of a given image instead of encrypting the whole image to make the encryption process faster. In their algorithm, the resulting blocks are transformed from the spatial domain to frequency domain using the Discrete Cosine Transform (DCT). A selected DCT coefficients are then encrypted and XORed with random bits to make it difficult for hackers to guess the original message.

M.Mishra, P. Mishra, Adhikary, and Kumar [9] proposed a new method for image encryption based on Fibonacci and Lucas series.

Different techniques are used for encrypting text messages into an unreadable form. Examples of this technique are presented next.

Singh and Gilhotra [10] proposed an encryption algorithm based on the concept of arithmetic coding. In this algorithm, a given word in a text is transformed into a floating point between 0 and 1. The resulting floating number is then transformed into a binary number that is in turn encrypted to another binary number, and then the resulting binary number is converted to a decimal number.

Huang, Chi Lee, and Hwang [11] proposed a novel encryption algorithm. This algorithm generates n2+n common secret keys in one session. It is based on the difficulty of calculating discrete logarithms problem.

Torkaman, Kazazi, and Rouddini [12] proposed a hybrid cryptosystem which is a combination of public and private cryptography. Their technique is based on a combination of cryptographic and steganography techniques. This algorithm provides a secure communication while defeating the up to date attacks. In their work, steganography algorithm is based on DNA algorithm and is used to hide a secret key. This secret key is distributed among two parties once a network communication is established.

Krishna [13] proposed a new mathematical model in which the output of the Elliptic Curve Cryptography (EEC) algorithm, a variable value, and a dynamic time stamp are used to generate the cipher text. They compared the results from their proposed model with the results from RSA and ECC algorithms. The results from their work showed that the security strength of their proposed model is more than RSA and ECC's security strength.

Other techniques for text encryption are proposed. These techniques are used to encrypt text into musical notes. Examples of other techniques are presented next.

Dutta, Chakraborty, and Mahanti [14] proposed a novel method for encrypting a text into musical notes. In their work, they used MATLAB in which 26 alphabets and 0 to 9 numbers are considered as -12 to 23 as musical notes. A sender encrypts the text message into musical notes and sends it to a receiver. The receiver, when receiving the encrypted message, decrypts the musical notes into the original text message (i.e. the plain text).

Yamuna, Sankar, Ravichandran, and Harish [15] proposed an encryption algorithm based on the transformation of a text message into musical notes. The encryption algorithm consists of two phases; in the first phase, the text message is encrypted into a traditional Indian music. In the second phase of encryption, the Indian music notes are encrypted again into western music notes.

Dutta, Kumar, and Chakraporty [16] proposed an encryption algorithm that encrypts a text message into musical notes. The text characters of a message are replaced by mathematically generated musical notes. These musical notes and the seed value for encryption/decryption key are sent to the receiver using the RSA algorithm.

The reset of this paper is organized as follows. Section II presents the related work. Section III presents our work, including research methodology, experiments, and analysis of the proposed algorithm. Finally, section IV presents the conclusions and future work.

II. Related Work

Bh, Chandravathi, and PRoja [17] presented Koblitz's method and used it to map a message to a point in the implementation of Elliptic Curve Cryptography [18, 19]. A given character in a text is mapped into its ASCII code, and then this ASCII code is encrypted into a point on a curve.

Singh and Gilhorta [5] proposed an encryption algorithm which is based on the transformation of a word of text into a floating point number (n) where, $1 \geq n \geq 0$. The resulting floating point number (n) is then encrypted into a binary number (b), and then (b) is encrypted using an encryption key.

Kumar, Azam, and Rasool [20] proposed a new technique of data encryption. In this technique, three random numbers are generated, say (D1), (D2), and D3. The random number D1 is

used for rows transformation in a matrix (V). D2 is used for columns transformation, and D3 is converted into a binary number. Rows and columns transformation is based on the value of the individual bits of that binary number. Three operations are defined in order to perform the matrix transformation namely, circular left shift, circular right shift, and reverse operation.

Abusukhon and Talib [21], and Abusukhon, Talib, and Issa [22] proposed the Text-to-Image Encryption algorithm (TTIE). In their algorithm, a given text file is encrypted into an image. Each individual character in the text file is transformed into an individual pixel (a pixel with a specific color). Each pixel in the resulting image consists of three integers; namely, Red, Green, and Blue, and each integer represent a specific color density. Having a matrix of integers, they were able to perform columns and rows shuffling making it difficult for hackers to guess the plain text (i.e. the original text message).

Abusukhon [23] investigated using block cipher encryption with TTIE encryption algorithm. In their work, the plain text is divided into number of blocks say {b1, b2…bn}, and then each block is encrypted into an image. All images from all blocks are combined into one image. This image represents the plain text.

Abusukhon, Talib, and Nabulsi [24] analyzed the encryption time for the TTIE encryption algorithm. They divided the total time of their experiment into six parts. The results from their work showed that the most significant time is the time required to store the encrypted data into the hard disk.

Abusukhon, Talib, and Almimi [25] proposed the Distributed Text-to-Image Encryption Algorithm (DTTIE) in order to improve the speed of the TTIE algorithm when a large scale data collection is used. They proposed to distribute the Text-to-Image Encryption Algorithm (TTIE) proposed in [21, 22] among seven nodes, where each node encrypts a partition of the data collection. They evaluated the speed up of their system when a large data collection (5.77 Giga Bytes) is used.

Our work differs from the work presented in [21, 22, 23, 24, 25]. In their work, each letter in the plain text is encrypted and mapped into one colored pixel (for example, letter "'a" is represented as red pixel, letter "b" is represented as green pixel and so on). In this paper, each letter is encrypted into a black bar. Each black bar consists of a specific number of black pixels. In this paper we propose the Bar Code Encryption Algorithm (BCEA).

III. Our Work

In this paper, Java NetBeans is used as a vehicle to carry out our experiments. All algorithms are implemented in Java, and build from scratch including encryption and decryption algorithms, client code, and the server code.

A. Machine Specifications

Our experiments are carried out using a single machine with the following specifications; processor Intel (R) core (TM)2, Duo CPU T5870 @ 2.00GHz, installed memory (RAM) 2.00GB operating system Windows 7 Ultimate and hard disk 24.5 GB (free space).

B. Data Sample

The data sample is created and stored in a notepad file. The data sample is shown Fig. 2.

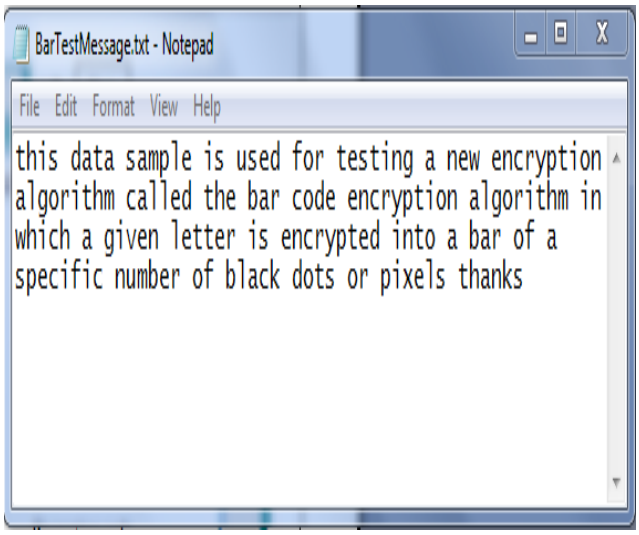

Fig. 2. Tested data

C. Research Methodology and Evaluation

The plain text shown in Fig. 2 is allocated at the client node. The client node encrypts the plain text using the proposed algorithm (BCEA), produces a bar code image, and then the resulting image is sent to the server. The server decrypts the received image and then displays the plain text message. To evaluate our system; the plain text message is checked and compared with the original one (i.e. the message sent by the client).

D. Our Experiment

In this experiment, encryption and decryption algorithms, a client code, and a server code are built from scratch using java. The system architecture is shown in Fig. 3.

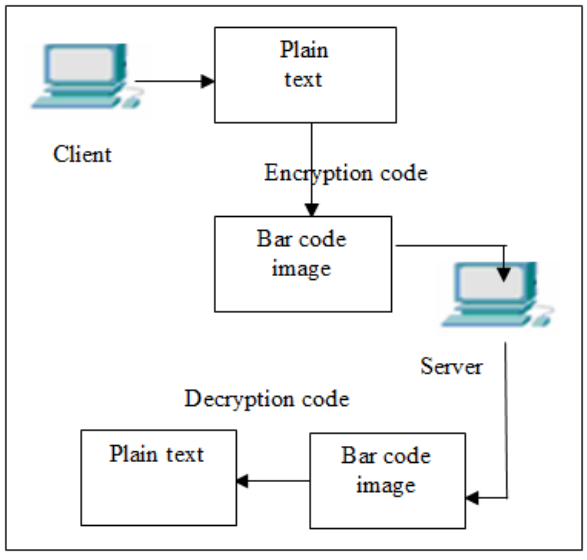

Fig. 3. The system architecture

In this experiment, the plain text shown in Fig. 2 is placed on the client side. The client uses the proposed encryption algorithm (BCEA) for encrypting the plain text. The output of the BCEA algorithm is an image of type ".png" as shown in Fig.4.

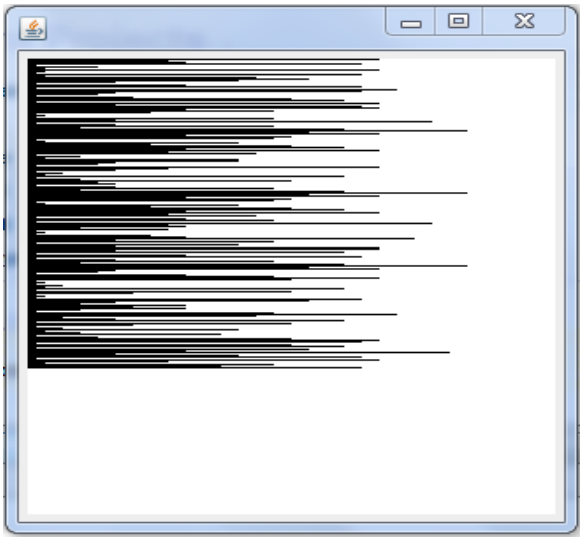

Fig. 4. The Barcode image results from running the BCEA algorithm

Using the proposed algorithm (BCEA), each letter from the plain text is encrypted into one bar. Each bar consists of a number of black pixels and has a specific length (the bar's length is measured in pixels). For example, in our experiment the letter "a" is encrypted as one bar of length = 10 black pixels. Letter "b" is encrypted as another bar of length = 20 pixels, and so on. We leave two white bars between each two black bars in order to clarify the bar code shape.

To verify our algorithm, the client encrypts the sample shown in Fig. 2, and then the encrypted text (.png file) is sent to the server. The server decrypts the .png file, and gets the original message shown in Fig. 5.

Encrypting the plain text into a bar code image makes it difficult for hackers to guess that each black bar in the image represents a specific letter from the plain text.

The main steps of encryption and decryption for BCEA algorithms are described in Fig. 6 (a) and (b).

In addition, we test the efficiency of our algorithm (BCEA) with respect to encryption time when different data collection sizes are used as shown in Fig. 7.

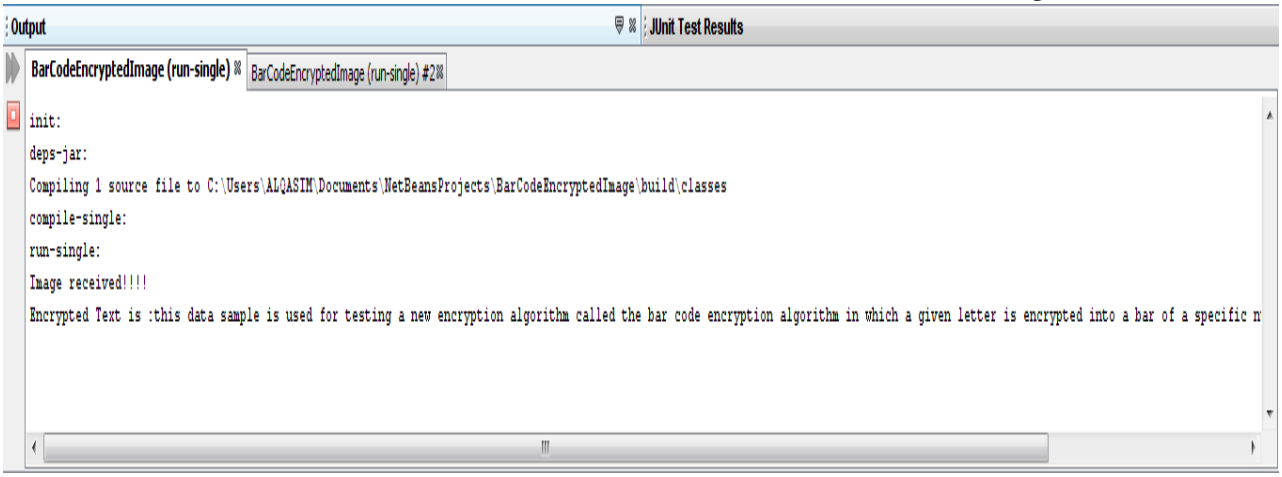

Fig. 5. Encryption algorithm running in reverse

(a) Encryption Algorithm

```
//      System set up
1. Determine the minimum size (X) and the maximum size (Y) of the bar.
2. Select an integer number between (X) and (Y) for each letter in the alphabet set [A to Z].
   //This number represents the bar length corresponding to a specific letter.
  // letter A →10 black Pixels, letter B → 20 black pixels and so on.

 // do the encryption
     3. Read the plain text and store it in an array of characters (chr)
     4. For (int i=1; i< = chr.length; i++)
       {
         Read chr [i]                  // read a letter (L) from chr
         Search for the bar length (L) correspond to the current letter
         Create a black bar whose length is (L)    // (see step 2)
         Draw the bar on the result image (. png)
         Draw two white bars on the image   // in order to separate the black bars from each other
        }
```

(b) Decryption Algorithm

```
1.  Read the image (the cipher text)
2.  Let the String  "OriginalMessage" = null
3.  While not the end of image     // determined by the image size
     {
       Extract a bar from the image

         If the extracted bar is a white bar then ignore   // discard white bars since they do not
                                                           //represent any letter from the plain text

        Else
         Calculate the bar length              // count the number of black pixels
         Search for the bar length and retrieve the corresponding letter
          OriginaMessage = OriginalMessage + the current letter      //    +  means concatenation

        }
```

Fig. 6. The main steps of encryption and decryption for the BCEA algorithm

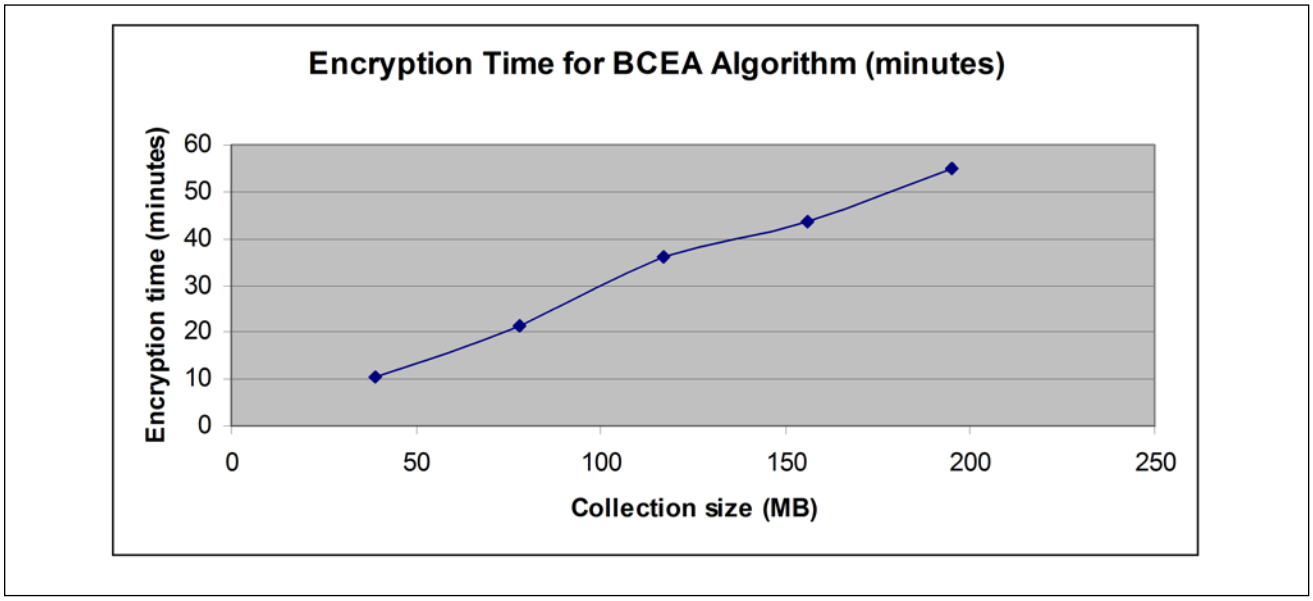

Fig. 7. Measuring the efficiency of (BCEA) encryption algorithm

To achieve our goal, five data collection are generated randomly. The first data collection consists of 100 groups and each group contains 100 text files. Each text file consists of 10 words and each word consists of seven characters generated randomly from the alphabet list. The second data collection consists of 200 groups, and the third data collection consists of 300 groups and so on. As shown in Fig. 7, the encryption time is proportional to collection size.

E. Analysis of the proposed algorithm (BCEA)

In this section, the maximum number of permutations (How many times a hacker may try before guessing the encryption key and getting the original text) is calculated.

The image resulting from the BCEA algorithm is a matrix of integers (M) having (R) rows and (C) columns. Suppose that the maximum row length (i.e. the number of columns in a row) is (mR), and the maximum number of rows is (nR); then the length of each bar in the image is limited by (mR). Each letter (L) from the plain text is encrypted as one bar and thus it allocates one row of the matrix (M) regardless the bar length.

The maximum number of key permutations (the range or the domain from which a key is picked out) is limited to (mR).

The number of letters (nL) in the plain text is limited to (nR). In other words, the number of letters in the plain text is limited by the maximum image size supported by Netbeans (in this paper). Thus, the maximum number of key permutations (P) provided by the BCEA is calculated as shown in (1).

$$P = \frac{mR!}{(mR - 26)!} \qquad (1)$$

Thus, suppose that mR = 1000 pixels, then P is calculated as follows.

P (1000, 26) = (1000×999×998…×974!) / 974!

= 7.2 e +77.

In our experiment, we use the key permutation where the letter "a" is represented as a bar of length equals 10 black pixels, the letter "b" is represented as a bar of length equals 20 black pixels and so on, Table 1 shows one of the key permutations.

TABLE I. POSSIBLE KEY PERMUTATION

Letter	a	b	c	d	…	z
Bar length (in pixels)	10	20	30	40	…	260

IV. CONCLUSIONS AND FUTURE WORK

In this paper, a novel encryption algorithm, the Bar Code Encryption Algorithm (BCEA), is proposed and tested. The BCEA is based on encrypting the plain text into a bar code image, where each letter in the plain text is encrypted into black bar consists of a specific number of black pixels.

The decryption algorithm is also tested where the plain text (the original message) is produced from the bar code image. Also, in section III-D, we measured the efficiency of the (BCEA) on encryption time, where different sizes of data collections are used.

Section III-E showed that the maximum number of key permutations is limited by the maximum row length (mR) of the resulting image.

The (BCEA) algorithm could be used for e-mail encryption, off-line data encryption, as well as online data encryption. For example, it can be used as a logistics barcode system (in packaging system), or as online Quick Response (QR) barcode for E-commerce.

In future, we propose to investigate the efficiency of the (BCEA) algorithm when a huge data size (multi Gigabytes) is used as well as to compare the efficiency of our proposed algorithm with the efficiency of other algorithms such as the TTIE algorithm with respect to the encryption time.

ACKNOWLEDGMENT

We would like to acknowledge and extend our heartfelt gratitude to Al-ZaytoonahUniversity of Jordan.

REFERENCES

[1] K..Lakhtaria"Protecting computer network with encryption technique: a study", International Journal of u- and e-service, Science and Technology, Vol. 4, No. 2, pp 43-52, 2011.

[2] A.Chan,"A security framework for privacy-preserving data aggregation in wireless sensor networks", ACM transactions on sensor networks, Vol. 7, No. 4, 2011. [Available online at: http://individual.utoronto.ca/aldar/paper/2011/cda- journal-tosn.pdf]. Accessedon 25-03-2015.

[3] S. Goldwasser, S.Micali, R. L.Rivest, "A digital signature scheme secure against adaptive chosen-message attacks", SIAM Journal of Computing Vol. 17, No.2, pp 281-308,1998.

[4] B. Zaidan, A.Zaidan, A. Al-Frajat, and H. Jalab, "On the differences between hiding information and cryptography techniques: an overview", Journal of Applied Sciences Vol. 10, No. 15, pp 1650-1655,2010.

[5] A. Singh, R. Gilhorta, "Data security using private key encryption system based on arithmetic coding", International Journal of Network Security and its Applications (IJNSA) Vol. 3, No. 3, pp. 58-67,2011.

[6] N. Nithin,M.B. Anupkumar , G. P. Hegde,"Image encryption based on FEAL algorithm". International Journal of Advances in Computer Science and Technology, Vol.2, No.3, pp 14-20,2013.

[7] M. Ali BaniYounes, A. Jantan, "Image encryption using block-based transformation algorithm". International Journal of computer science (IJCS). Vol.35 No. 1. pp 407-415, 2008.

[8] V.V Divya, S.K. Sudha, andV.R. Resmy, "Simple and secure image encryption". International Journal of Computer Science Issues (IJCSI). Vol. 9, No. 3, pp 286-289, 2012.

[9] M. Mishra, P. Mishra, M.C. Adhikary, S. Kumar, "Image encryption using Fibonacci-Lucas Transformation". International Journal on Cryptography and Information Security (IJCIS). Vol.2, No.3, pp 131-141, 2012.

[10] A. Singh, andR. Gilhotra, " Data security using private key encryption system based on arithmetic coding". International Journal of Network Security and its Applications (IJNSA). Vol. 3, No. 3, pp 58-67,2011.

[11] L. Huang, C. Chi Lee, and M. Hwang, "A n2+n MQV key agreement protocol". The International Arab Journal of Information Technology. Vol. 10, No. 2, pp 137-142,2013.

[12] M.R.N. Torkaman, N.S.Kazazi, and A. Rouddini, "Innovative approach to improve Hybrid Cryptography by using DNA steganography". International Journal on New Computer Architectures and Their Applications (IJNCAA). Vol.2 No. 1, pp 224, 235,2012.

[13] A.V. Krishna, "Time stamp based ECC encryption and decryption". The International Arab Journal of Information Technology. Vol. 11, No. 3. pp 276-281, 2014.

[14] S. Dutta, S. Chakraborty, and N.C. Mahanti, "A novel method of hiding message using musical notes". The International Journal of Computer Applications . Vol. 1, No. 16. pp 76-79, 2010.

[15] M. Yamuna, A. Sankar, S.Ravichandran, and V. Harish, "Encryption of a Binary String using music notes and graph theory". International Journal of Engineering and Technology (IJET). Vol. 5, No. 3. pp 2920-2925, 2013.

[16] S. Dutta, C. Kumar, and S. Chakraporty, "A Symmetric Key algorithm for cryptography using music". International Journal of Engineering and Technology (IJET). Vol. 5, No. 3. pp 3109- 3115,2013.

[17] P. Bh, D. Chandravathi, P.PRoja, "Encoding and decoding of a message in the implementation of Elliptic Curve cryptography using Koblitz's method", International Journal of Computer Science and Engineering, Vol. 2, No. 5, pp 1904-1907, 2010.

[18] N. Koblitz, "Elliptic Curve cryptosystems", Mathematics of computation Vol. 48, No. 177, pp 203-209, 1987.

[19] N. Koblitz,"A course in number theory and cryptography". 2'nd. ed. Springer-Verlag, 1994.

[20] K.M. Kumar, M.S.Azam, S.Rasool, "Efficient digital encryption algorithm based on matrix scrambling technique", International Journal of Network Security and its Applications (IJNSA) Vol. 2, No. 4, pp 30-41,2010.

[21] A. Abusukhon, M.Talib, "A novel network security algorithm based on Private Key encryption", International Conference on Cyber Security, Cyber Warfare and Digital Forensic. Kuala Lumpur, Malaysia, 2012.

[22] A. Abusukhon, M. Talib, and O. Issa, "Secure network communication based on text to image encryption", International Journal of Cyber-Security and Digital Forensics (IJCSDF), The Society of Digital Information and Wireless Communications (SDIWC) Vol. 1, No. 4, pp 263-271, 2012.

[23] A. Abusukhon, "Block cipher encryption for Text-to-Image Encryption algorithm", International Journal of Computer Engineering and Technology (IJCET) Vol. 4, pp 50-58, 2013.

[24] A. Abusukhon, M. Talib, and M. Nabulsi, "Analyzing the efficiency of Text-to-Image Encryption algorithm", International Journal of Advanced Computer Science and Applications (IJACSA) Vol. 3, No. 11, pp 35 – 38,2012.

[25] A. Abusukhon, M. Talib, and H. Almimi, "Distributed Text-to-Image Encryption algorithm", International Journal of Computer Applications Vol. 106, No. 1. [Available online at : http://research.ijcaonline.org/volume106/number1/pxc3899518.pdf]. Accessed on 25-03-2015,2014.