

INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY (IJCET)

ISSN 0976 – 6367(Print)

ISSN 0976 – 6375(Online)

Volume 4, Issue 3, May-June (2013), pp. 50-59

© IAEME: www.iaeme.com/ijcet.asp

Journal Impact Factor (2013): 6.1302 (Calculated by GISI)

www.jifactor.com



.....

BLOCK CIPHER ENCRYPTION FOR TEXT-TO-IMAGE ALGORITHM

Ahmad Salameh Abusukhon

(IT Dept., Al-Zaytoonah University of Jordan, Amman, Jordan)

ABSTRACT

The Internet is now providing many online services. These online services need both a client and a server to communicate with each other (this model is known as a client-server model). In this case, a client sends a request to the server and the server prepares the result and sends them back to the client. During the communication session, some sensitive data may be sent on both sides and thus it becomes necessary to protect the data from unauthorized users (known as hackers). One way to protect the data while sending them through the Internet is data encryption. The data encryption techniques are used to encrypt a given message into unreadable text using one or multiple encryption key(s). This way the user creates a secure path through the Internet making it difficult for hackers to guess the original text message. In previous work we proposed the Text-to-Image (TTIE) encryption algorithm and we analyzed the efficiency of this algorithm. In this paper, we propose the Block-Cipher TTIE (B-TTIE) algorithm.

Keywords: Block-cipher, Decryption, Encryption, Private-key, Secured-Communication.

I. INTRODUCTION

Cryptography techniques are used to protect the sensitive data from hackers while stored and transmitted. Cryptography or sometimes referred to as encipherment is used to convert the plain text to encode or make unreadable form of text [1].

The sensitive data are encrypted on the sender side in order to have them hidden and protected from unauthorized access and then sent via the network. When the data are received they are decrypted depending on an algorithm and zero or more encryption keys as described in "Fig.1".

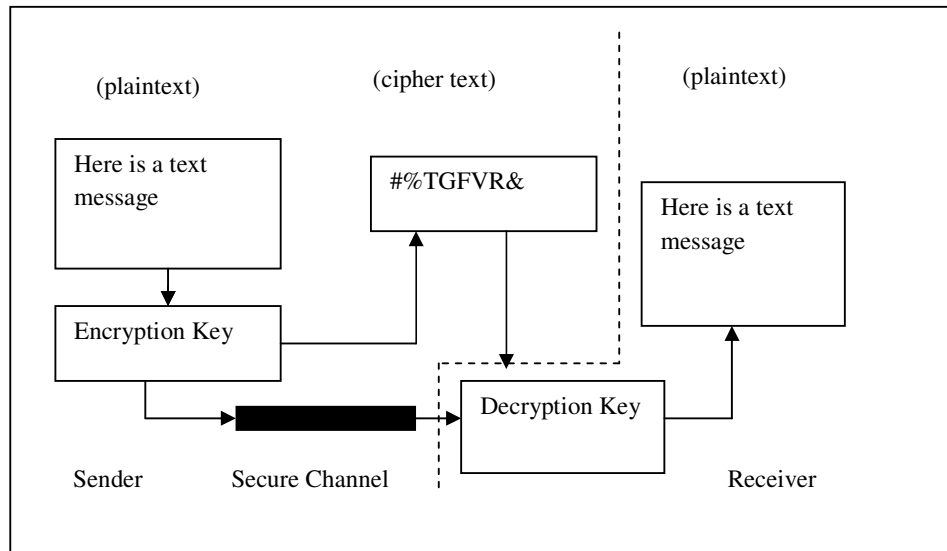


Figure 1. Encryption and decryption methods with a secure channel for key exchange

Decryption is the process of converting data from encrypted format back to their original format [2]. Data encryption becomes an important issue when sensitive data are to be sent through a network where unauthorized users may attack the network. These attacks include IP spoofing in which intruders create packets with false IP addresses and exploit application that use authentication based on IP and packet sniffing in which hackers read transmitted information.

Some of the techniques that are used to verify the user identity (i.e. to verify that a user sending a message is the one who claims to be) are the digital signature and the digital certificate [3]. Digital signature and digital certificate are not the focus of this research.

There are some standard methods which are used with cryptography such as private-key (also known as symmetric, conventional, or secret key), public-key (also known as asymmetric), digital signature, and hash functions [4].

In private-key cryptography, a single key is used for both encryption and decryption. This requires that each individual must possess a copy of the key and the key must be passed over a secure channel to the other individual [5].

Private-key algorithms are very fast and easily implemented in hardware. Therefore they are commonly used for bulk data encryption.

The main components of the symmetric encryption include - plaintext, encryption algorithm, secret key, cipher text and decryption algorithm. The plaintext is the text before applying the encryption algorithm. It is one of the inputs to the encryption algorithm. The encryption algorithm is the algorithm used to transfer the data from plaintext to cipher text. The secret key is a value independent of the encryption algorithm and of the plaintext and it is one of the inputs of the encryption algorithm. The cipher text is the scrambled text produced

as output. The decryption algorithm is the encryption algorithm run in reverse [6, 2, 7]. There are two main categories of private-key algorithms, namely block and stream encryption [8]. In stream encryption a given message is encrypted byte by byte or bit by bit at a time as in RC4, the one-time pad [5] and TTIE encryption algorithms [9]. In block cipher, a plaintext is divided into blocks of a certain length (m-bits) and then each block is encrypted (using a key) into cipher block. Thus, block-cipher algorithm operates on each block independently. Block-cipher techniques use different functions in order to perform text encryption such as XOR, multiplication, addition, bitwise shift etc. [10]. There are many examples on block-cipher algorithms such as Blowfish, IDEA, DES, FEAL, Twofish, RC6, Rijndael, and Mars[11]. In this paper, we propose to divide the text message into blocks and then encrypt each block (using an individual encryption key) into an image of type "png".

Public-key encryption uses two distinct but mathematically related keys - public and private. The public key is the non-secret key that is available to anyone you choose (it is often made available through a digital certificate). The private key is kept in a secure location used only by the user. When data are sent they are protected with a secret-key encryption that was encrypted with the public key. The encrypted secret key is then transmitted to the recipient along with the encrypted data. The recipient will then use the private key to decrypt the secret key. The secret key will then be used to decrypt the message itself. This way the data can be sent over insecure communication channels [6]. In this paper, we propose the B-TTIE encryption algorithm which is based on symmetric encryption technique.

II. RELATED WORK

Bh. P., et al. [12] proposed encoding and decoding a message in the implementation of Elliptic Curve Cryptography is a public key cryptography using Koblitz's method [13,14]. In their work, each character in a message is encoded by its ASCII code then the ASCII value is encoded to a point on the curve. Each point is encrypted to two cipher text points. Our work differs from their work. In their work they used public-key technique whereas in our work we use private key technique. They encoded each character by its ASCII value but we encode each character by one pixel (three integer values - R for Red, G for Green and B for Blue).

Singh and Gilhorta [5] proposed encrypting a word of text to a floating point number that lie in range 0 to 1. The floating point number is then converted into binary number and after that one time key is used to encrypt this binary number. In this paper, we encode each character by one pixel (three integer values R, G and B).

Kiran et al. [15] proposed a new technique of data encryption. Their technique is based on matrix disordering which was relied on generating random numbers used for rows or columns transformations. In their work, the original plaintext was ordered into a Tow-directional circular queue in a matrix A of order m x n. A number of column and row transformations were carried-out on the matrix and to do so a random function was used to generate positive integer say X and then X is converted to a binary number. Rows or columns transformation was made based on the values of the individual bits in the binary number resulted from the X value. Another random number was generated in order to determine the transformation operation. The random number was divided by three (as we have three types of transformation operations) and the modulus (0, 1, or 2) was used to determine the operation type. The operation type could be 0 (means circular left shift), 1 (means circular right shift) and 2 (means reverse operation on the selected rows). In case rows were selected to perform a transformation operation (the selection was made depending on the bit value of X) two

random numbers r1 and r2 were generated where r1 and r2 represent two distinct rows. Another two random numbers were generated c1 and c2 that represent two distinct columns. The two columns c1 and c2 were generated in order to determine the range of rows in which transformation had to be performed. After the completion of each transformation a sub-key was generated and stored in a file key which was sent later to the receiver to be used as decryption key. The sub-key format is (T, Op, R1, R2, Min, Max) where:

T: the transformation applied to either row or column

Op: the operation type coded as 0, 1, or 2, e.g., shift left array contents, shift right array contents, and reverse array contents.

R1 and R2: two random rows or columns

Min, Max: minimum and maximum values of range for two selected R1, R2.

Abusukhon, A., et al. [9] proposed the TTIE algorithm in which a given message is encrypted into an image of type ".png". Each character in the plaintext message is mapped into a pixel of three colors namely Red, Blue and Green. Each color has a value in the range from 0 to 255. Thus, to encrypt the letter "A", three integers, for example 1, 0, and 5, are created randomly in advance and then the letter "A" is mapped into a pixel (1, 0, 5) which is a mix of Red and Green colors (Note that the color value 0 means the color is missed). The result pixels are stored in a matrix and then the matrix is shuffled many times.

Abusukhon, A., et al. [16] studied the efficiency of the TTIE algorithm for multiple Gigabytes. They studied the effect of changing the memory size and the effect of changing the data collection size on the performance of the TTIE. They analyzed the encryption time by dividing this time into five basic times as described below:

RH: is the time required for reading the data collection from a hard disk.

SS: is the time required for a switch statement to be carried out. Switch statement is used to transfer the individual characters of a given message into pixels and filling them into a one-dimensional array.

FT: is the time required for filling the pixel's array into a two dimensional array. This is done in order to facilitate performing matrix scrambling (i.e. perform row or column swapping).

SM: is the time required for scrambling the matrix.

SI: is the time required for storing the result images (the encrypted text) on the hard disk.

They concluded that the most dominant time is the SI time. The work in [9, 16] is based on stream-cipher techniques. In this paper, we propose the Block-Cipher TTIE (B-TTIE) algorithm by which we divide the plaintext into blocks (rows or columns) where each block is assigned an individual key. Each block is then encrypted into a sub-image. All sub-images are joined together in order to produce the final image.

III. OUR TECHNIQUE

In a previous work [9], we proposed the TTIE algorithm which is based on stream cipher encryption. In this section, we describe our encryption algorithm B-TTIE developed for block-cipher encryption. The main steps of this algorithm are shown in "Fig. 2".

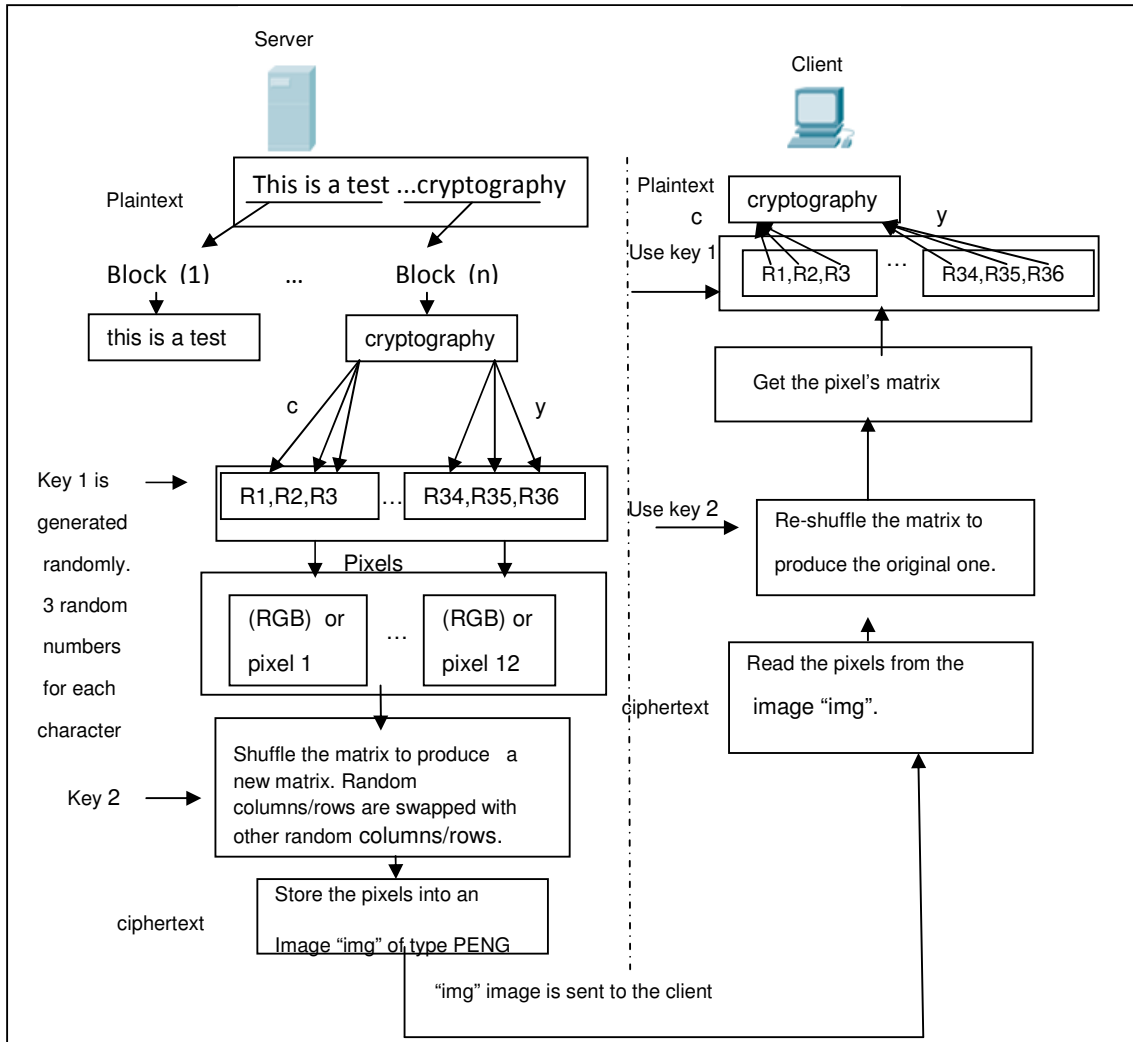


Figure 2. The main steps of the Block Text –To- Image encryption (B-TTIE). algorithm

In Fig. 2, we divide the plaintext into a number of blocks (block 1 to block n). Each block represents one row in the plaintext matrix. For each block, an individual key is generated (e.g. "key 1" in Fig. 2). This key contains three random integers for each individual character from A to Z. Thus, each character in this block is encrypted into one pixel. All ciphered blocks are joined together to form a matrix of pixels. This matrix is then shuffled (i.e. random rows/columns are swapped with other random rows/columns). The result of this process is "key 2" (as shown in Fig. 2). The ciphered blocks are stored as an image of type "PENG" and then sent to the other side of the network where reverse operations are performed. When the other network side receives the encrypted blocks it first re-order (re-shuffle) the matrix using "key 2", then it uses "key 1" to decrypt the ciphered blocks (note that multiple keys are used in this stage in order to decrypt the ciphered blocks: one key for each block. In Fig. 2, we follow up only one block) into plaintext blocks producing the original text message.

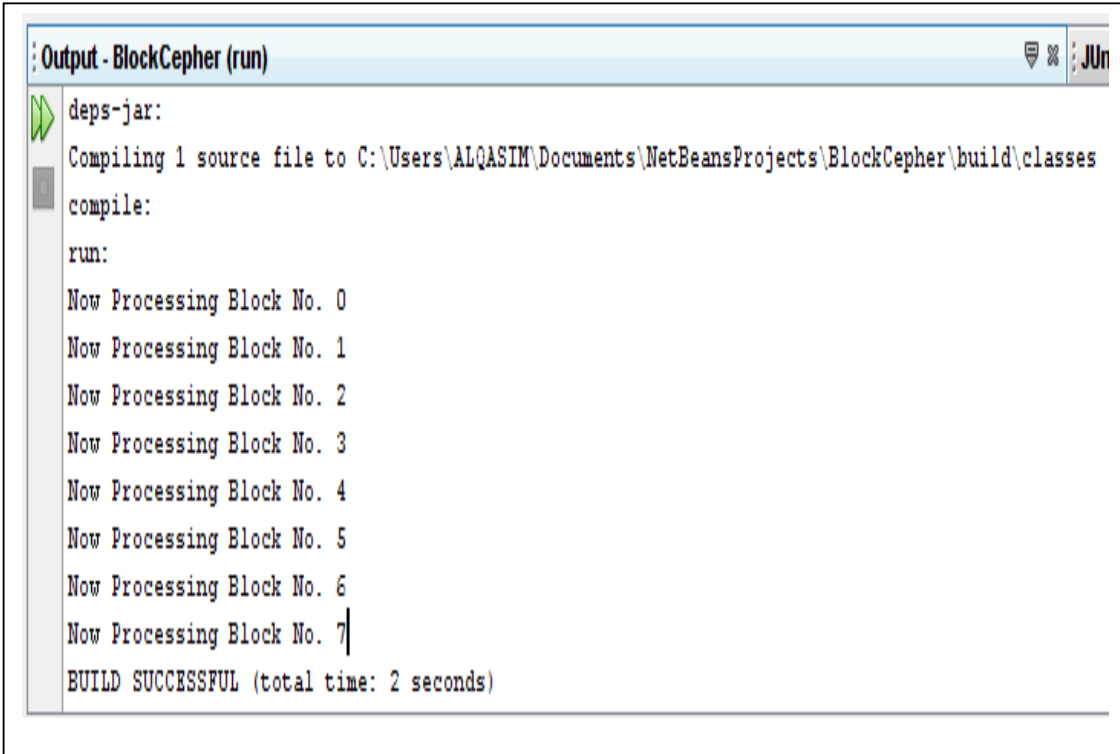
IV. EXPERIMENTS

We implemented our algorithm using Java NetBeans. We built the client's program, the server's program and the B-TTIE encryption and decryption algorithms. We used the following text message:

"cryptography is used to maintain the secrecy and integrity of information whenever it is exposed to potential attacks for example during transmission across networks" [17] encryption is the process of encoding a message in such a way as to hide its contents modern cryptography includes several secure algorithms for encrypting and decrypting messages. They are all based on the use of secrets called keys [9]".

The above message is divided into eight blocks as shown in Fig. 3. Each block is assigned an individual encryption key and encrypted into a cipher block.

"Fig. 4" describes part of the generated keys "key 0" to "key 7". Note that each three contiguous digits in each key represent one pixel (each individual pixel in a given key represents one character).



```
Output - BlockCepher (run)
deps-jar:
Compiling 1 source file to C:\Users\ALQASIM\Documents\NetBeansProjects\BlockCepher\build\classes
compile:
run:
Now Processing Block No. 0
Now Processing Block No. 1
Now Processing Block No. 2
Now Processing Block No. 3
Now Processing Block No. 4
Now Processing Block No. 5
Now Processing Block No. 6
Now Processing Block No. 7
BUILD SUCCESSFUL (total time: 2 seconds)
```

Figure 3. Dividing the plaintext into eight blocks.

"Fig. 5" describes the text after it is encrypted as an image. The image shown in "Fig. 5" is sent to the client. On the client side, we decrypt the cipher text into the original text message.

```

1#0#5#11#14#10#21#27#23#30#35#36#39#37#41#47#52#46#60#57#57#65#68#64#73#81#7
8#128#130#140#142#140#150#150#151#158#160#162#165#164#170#180#179#176#187#18
229#234#243#235#238#
8#0#2#15#15#12#19#22#25#35#33#29#37#39#38#51#48#54#58#55#61#67#68#68#81#75#7
3#130#132#144#137#137#145#150#151#159#157#162#168#167#169#175#175#174#183#18
230#233#241#238#242#
2#8#2#18#16#12#21#21#27#30#35#29#44#44#41#53#50#46#60#61#59#70#71#69#73#75#8
5#131#130#136#137#143#148#149#151#157#161#162#169#164#166#174#174#176#182#18
232#234#239#238#240#
4#0#4#12#14#16#19#19#22#34#30#34#43#43#42#47#50#54#63#55#59#66#68#71#73#74#7
4#135#129#144#143#138#145#146#146#159#154#160#164#166#169#180#173#178#183#18
227#228#241#241#238#
2#8#0#12#10#12#22#19#19#33#34#32#45#38#45#48#54#49#57#61#63#68#64#67#73#77#7
0#133#134#140#139#139#146#147#152#162#155#156#166#167#170#173#176#177#184#18
234#233#243#237#241#
0#8#0#18#10#12#22#19#25#31#30#33#38#38#45#49#52#54#61#55#59#64#70#68#78#76#7
0#129#135#142#137#144#145#152#150#155#154#155#164#167#165#177#174#172#183#18
231#234#242#242#240#
6#1#6#18#12#12#20#22#24#30#34#33#42#39#45#46#50#47#63#60#56#69#64#66#74#78#8
7#133#131#138#143#139#150#149#152#156#162#161#169#165#163#175#176#177#189#18
227#228#239#237#239#
8#5#5#13#10#14#22#19#24#32#28#31#40#43#38#52#53#51#55#58#63#66#65#71#76#81#7
8#128#131#142#142#136#146#147#153#162#159#156#170#171#170#179#174#173#183#18
234#229#238#240#241#
    
```

Figure 4. Encryption keys (key 0 to key 7)

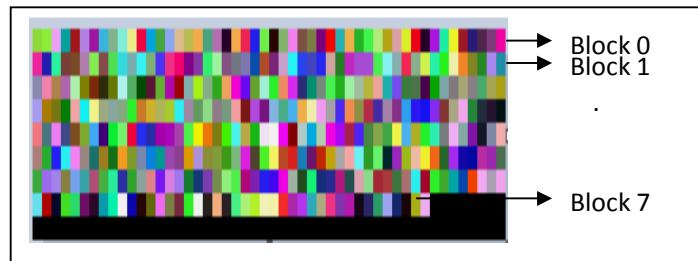


Figure 5. Ciphers blocks (block 0 to block 7)

V. ANALYSIS

In our algorithm (B-TTIE) a given plaintext message is divided into blocks. Let M represents the plaintext message, B represents a block of that message, NB represents the total number of blocks of that message, and K represents an individual encryption key. Let $M = \{B_0, B_1, B_2, \dots, B_n\}$. For each B a K key is assigned. Each K consists of 26 elements where each element consists of three random integers. Thus, the total number of possible permutations (P) to encrypt a given M is:

$$P = NB * ((256)^3)^{26} \quad (1)$$

Note that P is proportional to NB, thus when the number of blocks, NB, is increased, P is increased. The maximum value for NB could be the number of words in a given M. The

minimum value for NB is one (i.e. consider the whole message as one block and encrypt it using one key).

In a previous work Abusukhon, A., et al. [9] calculated the number of possible permutations for the TTIE encryption algorithm which is based on stream ciphering. They found that P for the TTIE is:

$$P = ((256)^3)^{26} \quad (2)$$

Thus, our algorithm increased P by $(NB-1) * (((256)^3)^{26})$, making it hard for hackers to guess the plaintext. To guess the plaintext hackers need to guess all encryption keys (key0 to key7). In addition when shuffling the matrix (here we swap the matrix rows (i.e. blocks)) we shuffle the blocks but not their keys; thus the shuffled blocks are assigned keys of other blocks. For example, in Fig. 6, block 0 is swapped with block n, but their keys (key 0 and key n) are not swapped; thus block n is assigned key 0 and block 0 is assigned key n. This makes it difficult (depending on the number of blocks, NB, and the number of swapping operations) for hackers to guess the key required for decrypting a given ciphered block.

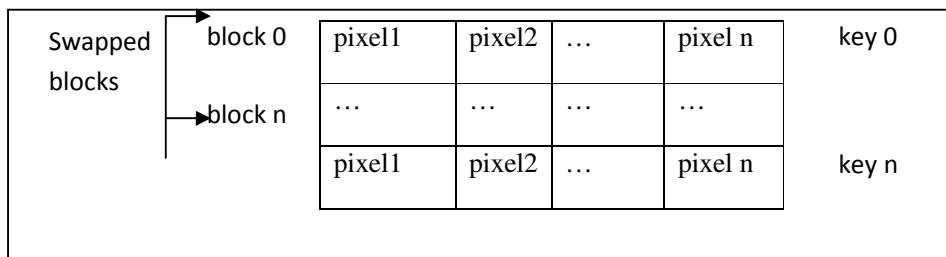


Figure 6. Matrix shuffling without swapping the blocks keys

VI. CONCLUSION AND FUTURE WORK

In this paper, we investigated using block-cipher technique with the TTIE encryption algorithm proposed by Abusukhon, A., et al. [9]. We proposed the B-TTIE algorithm which divide the plaintext message into blocks, generate a random encryption key for each block and then encrypt each block into a sub-image. All sub-images are collected together to create one image of type "png". The B-TTIE algorithm increased the number of key permutations by:

$$(NB-1) * (((256)^3)^{26})$$

making it difficult for hackers to guess the plaintext.

The B-TTIE can be used for encrypting the data stored on an individual machine (offline machine), the data sent via the Internet (create a secure path through the Internet), and it can be used for e-mail security where all text messages appear as images not as text.

In future, we propose to investigate the efficiency of the B-TTIE encryption algorithm for large-scale data collection (multiple Gigabytes) when a cluster of nodes are used. A server will distribute the data collection among the nodes. Each node receives a plaintext message will divide it into blocks, generates a random key for each block, and then encrypts all blocks in one image.

VII. ACKNOWLEDGEMENTS

Heartfelt gratitude to Al-Zaytoonah Private University of Jordan for their help and financial support to carry out this work successfully.

REFERENCES

- [1] K. Lakhtaria Protecting computer network with encryption technique: A Study. International Journal of u- and e-service, Science and Technology 4(2), 2011.
- [2] A. Chan, A Security framework for privacy-preserving data aggregation in wireless sensor networks. ACM transactions on sensor networks 7(4), 2011.
- [3] S. Goldwasser, S. Micali, and R. L. Rivest, A Digital signature scheme secure against adaptive chosen-message attacks, SIAM Journal of Computing 17(2), 1998, 281-308.
- [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 10(15), 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), 3(3), 2011.
- [6] W. Stallng, Cryptography and network security principles and practices, 4th edition Prentice Hall. Available at: <http://www.filecrop.com/cryptography-and-network-security-4th-edition.html>, Accessed on 1-Oct-2011.
- [7] C. E. Shannon, Communication Theory of secrecy systems. Bell System Technical Journal, 28(4), 1949, 656-715.
- [8] M. Bellare, J. Kilian, and P. Rogaway, The Security of cipher block chaining. In Proceedings of the Conference on Advances in Cryptology (CRYPTO'94). Lecture Notes in Computer Science, vol. 839, 1994.
- [9] A. Abusukhon, M. Talib, I. Ottoum, 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) 2012. 1(4), 2012.
- [10] R.H. Ismaeel, Apply Block Ciphers Using Tiny Encryption Algorithm (TEA). Baghdad Science Journal, 7(2), 2010.
- [11] A. T. Hishem, N. M. Hassen and E. M. Farhan, VHDL Implementation of Hybrid Block Cipher Method (SRC). Eng. And Tech. Journal, 28(5), 2010.
- [12] P. Bh, D. Chandravathi, P. Roja, Encoding and decoding of a message in the implementation of Elliptic Curve cryptography using Koblitz's method. International Journal of Computer Science and Engineering, 2(5), 2010.
- [13] N. Koblitz, Elliptic Curve cryptosystems, Mathematics of Computation, 48, 1987, 203-209.
- [14] N. Koblitz A Course in Number Theory and cryptography. 2nd edition. (Springer-Verlag, 1994).
- [15] M. Kiran Kumar, S. Mukthiyar Azam, and S. Rasool, Efficient digital encryption algorithm based on matrix scrambling technique. International Journal of Network Security and its Applications (IJNSA), 2(4), 2010.

- [16] 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), 3(11), 2012, 35 – 38.
- [17] G. Coulouris, J. Dollimore, and T. Kindberg, Distributed Systems Concepts and Design. 4th ed, (Addison Wesley, 2005).
- [18] Vismita Nagrale, Ganesh Zambre and Aamir Agwani, “Image Stegano-Cryptography Based on LSB Insertion & Symmetric Key Encryption”, International Journal of Electronics and Communication Engineering & Technology (IJCET), Volume 2, Issue 1, 2011, pp. 35 - 42, ISSN Print: 0976- 6464, ISSN Online: 0976 –6472.
- [19] Vikendra Singh and Sanjay Kumar Dubey, “Analysing Space Complexity of Various Encryption Algorithms”, International Journal of Computer Engineering & Technology (IJCET), Volume 4, Issue 1, 2013, pp. 414 - 419, ISSN Print: 0976 – 6367, ISSN Online: 0976 – 6375.