A Simultaneous Approach for Compression and Encryption Techniques Using Deoxyribonucleic Acid

Dilovan Asaad Zebari
School of Computing
Faculty of Engineering
University Teknologi Malaysia
Johor, Malaysia
damzebari2@live.utm.my

Habibollah Haron
Department of Computer Science
Faculty of Computing
Universiti Teknologi Malaysia
Johor, Malaysia
habib@utm.my

Diyar Qader Zeebaree
School of Computing
Faculty of Engineering
University Teknologi Malaysia
Johor, Malaysia
dqszeebaree2@live.utm.my

Azlan Mohd Zain
School of Computing
Faculty of Engineering
University Teknologi Malaysia
Johor, Malaysia
azlanmz@utm.my

***Abstract*— The Data Compression is a creative skill which defined scientific concepts of providing contents in a compact form. Thus, it has turned into a need in the field of communication as well as in different scientific studies. Data transmission must be sufficiently secure to be utilized in a channel medium with no misfortune; and altering of information. Encryption is the way toward scrambling an information with the goal that just the known receiver can peruse or see it. Encryption can give methods for anchoring data. Along these lines, the two strategies are the two crucial advances that required for the protected transmission of huge measure of information. In typical cases, the compacted information is encoded and transmitted. In any case, this sequential technique is time consumption and computationally cost. In the present paper, an examination on simultaneous compression and encryption technique depends on DNA which is proposed for various sorts of secret data. In simultaneous technique, both techniques can be done at single step which lessens the time for the whole task. The present work is consisting of two phases. First phase, encodes the plaintext by 6-bits instead of 8-bits, means each character represented by three DNA nucleotides whereas to encode any pixel of image by four DNA nucleotides. This phase can compress the plaintext by 25% of the original text. Second phase, compression and encryption has been done at the same time. Both types of data have been compressed by their half size as well as encrypted the generated symmetric key. Thus, this technique is more secure against intruders. Experimental results show a better performance of the proposed scheme compared with standard compression techniques.**

***Index Terms*—Security, Compression, Encryption, DNA, Standard Compression Techniques.**

I. INTRODUCTION

Due to the expansion in transmission of data through the Internet and its constrained transfer speed, and time taken by the data to achieve the goal additionally expanded [1]. Data compression is one of the most far reaching applications in computer technology which proves to be useful in this scenario as it lessens the asset use, for example, data storage space or the capacity of transmission [2]. The underlying feature of data compression is to change over a series of characters into another arrangement of characters which comprises of same data however whose length is as little as could be allowed [3]. Compression is thinkable in light of the reality that the large priority of this present fact data is exceptionally excessed and can decrease the measure of data by expelling pointless data and deception to store and transmit the required bits of data [4]. Data compression has turned into a need while handling data that possesses colossal memory [5]. The conspicuous advantages that can be accomplished incorporate lessening stockpiling necessity, data transfer capacity, encoding fewer bits, less time requests for transmission, compelling channel usage and so on [6]. Algorithms and techniques that are utilized depended firmly on the kind of data, for example, regardless of whether the data is static or dynamic, and on the substance, that can be any mix of content, images, numeric data or unhindered binary data [2]. Different space-time exchanges, for example, compression ratio, the time taken for encoding and decoding, measure of storage space required and so forth can be determined for each compression method [1]. In alike way, decompression is an opposite technique to compression which gives back the data to its one of kind structures as it is shown in Figure 1.

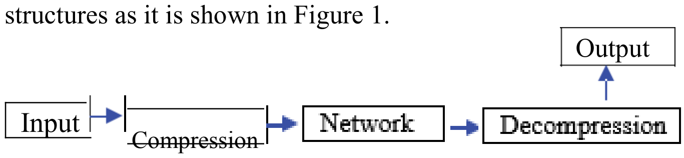

Figure 1: Compression and Decompression Process

In theory, compression and cryptography are two contradicting techniques. Cryptography techniques can give a protected method to exchange secure data. Enhancing the security and dependability of data can be received dependent on the nature of DNA [12]. Conversely, a compression technique tries to decrease the extent of transmit or put away data by discovering and omitting duplicate parts patterns of data. Nevertheless, data compression and cryptographic framework are profoundly associated and commonly helpful that they are fit for being utilized together. The points are to create a littler size of data; to guarantee a quality of data amid reconstruction; to accelerate data transmission; to decrease transfer speed prerequisite, and to guarantee its condition [22]. Utilizing a data compression technique together with an encryption technique, in the right request, bodes well for three reasons. To begin with, compression data before encryption diminishes the redundancies that can be exploited by cryptanalysts to recoup the first data. Second, the encryption procedure works faster after compression. Third, whenever encrypted data are transmitted in a computer system, the transmission capacity is better used. Information must be compressed before encryption. On the off chance that it was the contrary case, the aftereffect of the cryptographic task would be indecipherable data and no pattern or redundancy would be available, prompting exceptionally poor or no compression by any means [7].

Recently, Deoxyribonucleic Acid (DNA) assumes an incredible job in enhancing applications by means of joining

978-1-7281-2741-5/19/$31.00 ©2019 IEEE

with computer science. In this paper, we proposed a hybrid algorithm among compression and encryption for various data which are plaintext and image based upon DNA.

A. Data Compression Techniques

Essentially data compression is defined as a method which can be used in several methods to decrease the size of different data such as audio, image, video, and text in order to decrease the size of memory as well as the transformation time over the network. In general, lossless and lossy are considered as two main types of compression techniques [8]. Lossless compression is the process of converting the original data with compressed data becomes more concise without reducing the loss of information [9]. Huffman coding, and Shannon-Fano coding, Run Length Encoding (RLE), Arithmetic encoding, and Lempel Ziv families which are LZ77 and LZ78 are different lossless algorithms. These algorithms are typically used for text also can be used for programs, images and sound. The major benefit of this type could be preserved the quality whilst the size of data could be reduced less than lossy compression [8]. Lossy compression algorithms, on the other hand, remove unnecessary data permanently and so the original data couldn't be completely regenerated. Several techniques are used for lossy compression such as DCT, DWT, Rectangle Segmentation, Transform coding and Sparse Matrix Storage (RSSMS). This type of compression is utilized for sound, image, and video despite of no preserving the quality, but it can cause big reduction in size [10].

B. DNA Background

DNA is a molecule that represents the genetic material for all living organisms. It is the information carrier of all life forms and considered as the genetic drawing of living or existing creatures. DNA consists of tow chains which are twisted around each other to compose a double- stranded helix with four different nucleotides on the inside [11]. A single strand of DNA molecule consists of four chemical bases representing its building blocks, named as nucleotides. Each nucleotide consists of a phosphate, a sugar, and a nucleotide together. Nitrogenous base causes to initialize types of nucleotide which are purine or pyrimidine base. Guanine (G) and Adenine (A) are considered as purine bases whereas cytosine (C) and thymine (T) are pyrimidines bases, where G pairs with C and T pairs with A. Any DNA sequence is represented by a combination of these 4 bases, encoding the genetic information. Each three nucleotides are called a codon there are 64 codons since there are (43) letter combinations. Finally, amino acid can be created by codon translation. The structure and function of protein would be dictated after arrangement of the amino acids [29].

II. LITERATURE REVIEW

By seeking out previous works; the procedure type of literature works is done of lossless data compression and hybrid data compression encryption mapped. Large amount of data must be stored and handle in such a way using efficient algorithm usually succeed in compression. In literature, many compression techniques are proposed for compress large data [13]. Entropy and dictionary are the two main techniques which have been done in the literature. Entropy based several techniques have been proposed; for example; Huffman coding [2], Modified Huffman Coding [5], RLE [15], improved RLE [17], Shanon-Fano coding [14], and Arithmetic coding by exploiting the statistical allocation of input data and implementing the variable length encoding [18]. LZ77 and LZ78 are data compression techniques based on dictionary. This type compression of data can be achieved by substituting duplicated occurrence of data with indicates to elements in the dictionary [19]. Utilizing only one technique of both types is called single compression techniques. These techniques may have benefits in some specific bit-streams while cannot consider as suitable for all streams. Thus, a presumption has been made which can make a combination from these two types and utilize them in a right order can collect their benefits and enlarge the suitable range. In [20] Lempel Ziv Welch (LZW) is an algorithm from LZ78 family and a combination has been made with RLE where the results showed that this combination obtained better results than single RLE, single LZW.

Because of enlarging need for data transforming safely and quickly, researches on data security via both cryptography and compression algorithms begin to take form. Based on the processual sequences the combination of these two algorithms can be divided into three categories which are a cryptography algorithm followed a compression algorithm, vies versa, and both algorithms used in a single process [22]. The first category has been done in [21] which focused on mobile communication security. At first step, Elliptic Curve technique is utilised for encryption whereas Shannon-Fano is utilized for compression at the second step. Low efficiency on different points has been showed which leads resultant into negligence. Several algorithms have been proposed based on second category [23] introduced an optimized technique for text data. Huffman compression is used at first stage and combined with recently improved symmetric cryptographic techniques.

In [24] data compressed by half at the first step and at the second step Shalloon's notion has been justified of diffusion by producing various cipher text character for single plaintext character for its different occurrences in the plaintext. in [25] Arnold transform, key matrix and chaos system are used to introduce a hybrid technique for image. In order to comprise and encrypt the image is decomposed into four bands. First the duplicated data has been eliminated to achieve the compression while chaos system with Arnold transform are applied for encryption. In normal cases, the first two categories are done which are called sequential methods, but this extremely causes increasing the time complexity. Thus, several simultaneous methods have been proposed in which both techniques compression and encryption can be used at the same stage which is far better than sequential methods [1]. A simultaneous technique is introduced that is based on Cosine Number Transform [26]. This technique is achieved by using a key in the measurement matrix generation based on chaotic map. This technique can be resistant to statistical analysis also an effective compression is provided. In [1] a hybrid compression – encryption algorithm is introduced for ratio compression and time. Based on Arithmetic coding and XOR both compression and encryption is done respectively. A simultaneous technique for both data compression and encryption based on DNA is proposed here.

III. PROPOSED SCHEME

The main purpose of the proposed scheme is to ensure security and capacity for secret data. This scheme has been applied on different secret data which are text and image. Our proposed scheme is based on achieving compression and encryption based on DNA. The proposed simultaneous scheme consists of two phases. Each character of secret text and image pixel represents DNA nucleotides at first phase. Each DNA nucleotide represents a binary bit with generating a secure symmetric key at the second phase. The simultaneous compression of the proposed scheme has been explained in more detail in the following sections.

A. First Phase

The compression consists of two phases as we mentioned above. The first phase is encoding the secret data to DNA by mapping each character of secret text and each pixel of image to DNA nucleotides. Each DNA nucleotide represents two binary bits as it is shown in Table I. Because of representing each text character and image pixel by eight binary bits, each of them represents four DNA nucleotides. Some previous works [27] encoded each text character by three DNA nucleotides. The drawback of their work is that secret text should contain only (26 capital letters, 26 small letters, numbers from 0 to 9, space, and dot) which cannot contain any other punctuation marks such as (+,?, $, ... , etc). Therefore, their algorithm cannot encode a secret text which contain punctuation marks. Due to the importance of having punctuation marks in secret text, in this phase we improved their work by extending of using other punctuation marks where the secret text can contain (26 capital and small latter's, numbers from 0 to 9, space, dot, and 26 different punctuation marks). Based on 64 different codons of DNA we generate a permutation which works on mapping from each text character to three DNA nucleotides.

Table I: DNA Encoding Rule

Decimal	DNA	Binary
0	A	00
1	C	01
2	G	10
3	T	11

This Algorithm works to generate 64 numbers, each of this number consists of combination of three different decimal numbers of the range between 0 to 3, total combination numbers (4^3 equal to 64) is same as DNA codon numbers. Consequently, each DNA codon can represent a generated combination of decimal numbers. Each number of the generated serial of decimal numbers can represent a DNA nucleotide or two binary bits depending on Table I. Then, in this phase we can describe a 1-1 map from all possible combinations between these 64 numbers and (26 capital and small latter's, numbers fro m 0 to 9, space, dot, and 26 different punctuation marks). This map is known for both sender and receiver and it should keep secret. For example, at first generation, this Algorithm will generate combinations of decimal numbers (000, 213, …, 333), these combinations can be encoded to DNA nucleotides and binary bits based on Table I, where (0 = A = 00, 1 = C = 01, 2 = G = 10, 3 = T = 11). In our improved method each of the same capital and small letters can represent same combination of decimal numbers such as (*A* or *a* represents 000, *H* or *h* represents 013). This can be encoded without any problem, but during decoding the algorithm decoded to small letter where capital letters disregarded always. Thus, to overcome this problem we considered each name should be written in double brackets and after each dot the sentences must be start with capital letter. In this case, our algorithm can encode each letter of secret text by 6 -bits instead of 8-bits. Furthermore, this algorithm can encode a secret text which contains 26 different punctuation marks instead of encoding a secret text without any punctuation mark. For black-white images there are 256 pixels where consists from 0 to 255, each pixel can represent by 8 binary bits. Every two bits represent a DNA nucleotide based on Table I. Based on this encoding method each 4 pair of image pixels represent a four DNA nucleotide. For example, pixel 0 is equal to AAAA, pixel 182 is equal to GTCG, pixel 255 is equal to TTTT, and so on. Because (52 capital and small letters, numbers from 0 to 9, space, dot, and 26 punctuation marks are equal to 90), proposed scheme can denote capital and small letters by same DNA codon. Thus, each character of plaintext can be converted into 3 DNA bases, but each image pixel cannot convert into 3 DNA bases because image contain 256 different pixels. As a result, in this phase we compressed secret text with %25 by representing each letter by 6- bits whereas image is converted into DNA form as it is shown in Figure 2.

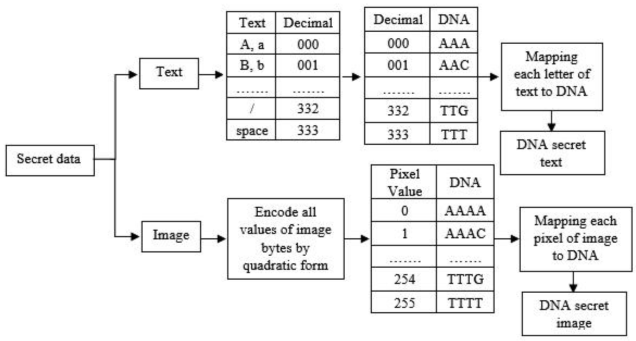

Figure 2: Block Diagram of First Phase

B. Second Phase

In this phase, the researcher proposed a method for further text compression and to compress image by it is half. We consider this phase is a data compression and encryption at the same time. In this phase, the obtained form of secret data from first phase will be used as input which represented by a DNA sequence such as, ACAGTAC. In normal cases, each DNA nucleotide represents by two binary bits. Our proposed method can represent each DNA nucleotide by only one binary bit with generating a symmetric key. In the first step, the DNA sequence will be divided into two segments, in case of having a unique nucleotide at the end after segmentation an (A) will be added which is value is equal to 0. With four possible nucleotides, the two nucleotides can give 4^2 = 16 different possibilities (segments), and the frequency of each segment will be calculated. In the second step, a 4 * 4 matrix will be generated as it is shown in Figure 3. Then, the four highest frequencies will set into (00, 01, 10, 11) fields, the second four highest frequencies will set into (20, 21, 30, 31) fields, the third four highest frequencies will set into (02, 03, 12, 13) fields in order to increase the number of (0's and 1's), and the four lowest frequencies will set into (22, 23, 33, 32) fields during conversion. Security also is obtained in this step because it is hard to know by attackers which segment is settled on which field. After that, the DNA sequence will be converted into a serial number of ranges between 0 to 3 based on generated matrix. This matrix is known for both sender and receiver and kept secret. Each number of the obtained sequence number represents two binary bits. In the third step, we represent each number of new forms by only one bit with keeping all (0's and 1's) as same whereas each 2 will be converted by 0 and each 3 will be converted by 1 with saving the positions within a file as a symmetric key. Therefore, in this phase the size of secret data is reduced by it is half size because each DNA nucleotide represented by only one binary bit. As a result, the size of secret text is compressed by %62.5 and image by %50 after applying these two phases.

r/c	0	1	2	3
0	$high_1$	$high_2$	$high_5$	$high_6$
1	$high_3$	$high_4$	$high_7$	$high_8$
2	$high_9$	$high_{10}$	$high_{13}$	$high_{14}$
3	$high_{11}$	$high_{12}$	$high_{15}$	$high_{16}$

Figure 3: Generated 4 * 4 Matrix

To provide further security, the symmetric key has been encrypted. Generated symmetric key is a file for example n which holds the position of 2's and 3's as we mentioned above. The basic concept of our proposed method is that the first step is to select the first value of n, and for the second value we used subtraction operation between first and second and so on. If n $= R_1, R_2, R_3, \ldots, R_n$, to produce new value of n in this step R_1 will be selected and unchanged; new value of $R_2 = R_2\text{-}R_1$, $R_3 = R_3\text{-}R_2$, $R_n = R_n\text{-}R_{n-1}$. Here, each number will be considered as four binary bits from 1 to 9, but in case of producing a number out of 1 to 9 we put between two comma which considered it is value as 0 because there is no 0 value in *n*. For example, if *n* = {1,2,5,7,17,30,31} after applying this step the new produced file of n is equal to {1132,10,131}, the general steps of this phase has been illustrated in Figure 4.

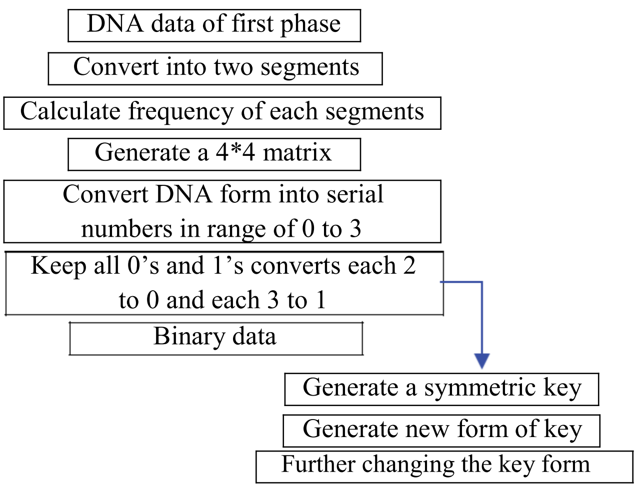

Figure 4: General Diagram of Second Phase

IV. DATASETS AND PERFORMANCE EVALUATION

Our attention will be focused in this section to evaluate the performance of the proposed scheme based on text and image datasets. Different corpora's are composed of a collection of text files designed specifically as datasets for test applications lossless compression methods. Seven different files from different corpora have been selected and tested. Calgary corpus contains of 18 different files which are totalling about 3.2 million bytes. Four files have been selected from this corpus which are *paper1, paper2, book1, and book2*, category of these files are technical papers, fiction and non-fiction books. The Canterbury corpus consists of 11 different type of files totalling about 2.8 million bytes. Three files have been selected from this corpus which are *alice29, asyoulik, and plrabn12*, the category of them are English text, Shakespeare, and a poetry. *lena, baboon, and peppers* in quality 128x128 and JPG format has been used as image datasets to evaluate the performance of proposed method.

Based upon mentioned datasets some commonly used data compression performance indices, including compression ratio, bit per character, bit per pixel, and compressed file size, will be calculated and discussed. The performance of plaintext for the proposed data compression scheme were compared through their compression ratio, bit per character, and compressed file size with other basic data compression techniques such as LZFG, Run Length Encoding (RLE), Huffman Encoding (HUFF), Shannon Fano (SF), and Arithmetic Coding (AC). However, the performance of the proposed scheme for image were compared through their bit per pixel, and compared with HUFF and AC. Following measurements to evaluate the performances of proposed scheme have been reviewed.

Compression Ratio (CR) is utilized to find the ratio between the original data size and the corresponding compressed data size. This is to find out how much data gets compression by using existing scheme, mathematically it is denoted as CR and it is calculated by following formula 1 [28]. The file size after processing will be depended by CR. The output files which compressed more have less CR. The compressed file size is greater than the initial file size when the CR exceeds one. [17].

CR = compressed size / uncompressed size (1)

Bit Per Character (BPC) means how many bits need to represent in each character. So, less BPC is better than the higher BPC which makes the file size smaller because Smaller files use less space to store.

BPC = number of bits in compressed text / number of characters in uncompressed text (2)

Also, the compression factor (CF) and saving data (SD) for text data have been calculated by using equation 3 and 4 respectively.

CF = uncompressed size / compressed size (3)

SD = (1-compression ratio) *100 (4)

Bit Per Pixel (BPP) is the bits are stored per each pixel of an image. This is considered as the ratio bit numbers in an image after compression to all pixel numbers in initial image [34]. large memory space is required to store an Image when BPP is large, because of many colours in image so BPP should be less.

$$BPP = \text{number of bits in compressed image / total number of pixel in the image} \quad (5)$$

File Size (FS): is the size of compressed data file.

V. RESULTS AND DISCUSSION

After the researcher simulated our simultaneous data compression encryption scheme, the following results were obtained from the simulation.

A. Plaintext Datasets

The results of applying the proposed data compression scheme on standard seven text data files are presented in the Table II. The table shows the size of original text files in KB also the number of bits before applying the proposed scheme have been showed. In the field of the compressed data, the size text files in KB also the number of bits is reviewed after applying the proposed scheme. As observed from the Table II, the size of all files and the number of bits continuously decreased after compression. This means the number of characters in each file has been decreased. As we see the number of bits in paper1 was 415,288 bits means 51,911 characters whereas after compression we obtained 154,214 bits which about 19,276 characters. Hence, 32,635 characters have been omitted from the original file.

Table II: Original and Compressed Plaintext Data

Dataset	Original Data		Compressed Data			
	File size	No. of bits	No. of bits	CR	SP	CF
bib	109	839848	317558	0.3781	0.6219	2.6435
book1	751	3317427	1248734	0.3764	0.6236	2.6566
book2	597	4761776	1780264	0.3738	0.6262	2.6747
news	369	2936400	1095624	0.3731	0.6269	2.6801
paper1	52	415288	154214	0.3713	0.6287	2.6929
paper2	81	643744	242090	0.376	0.624	2.6591
progc	39	304992	110572	0.3625	0.6375	2.7583

Previous works accomplished to evaluate the efficiency of any compression technique are performed having important parameters. In Table III several experiments conducted and evaluated. Table shows the results of the compression based on CR, BPC, and FS for the proposed scheme, and the standards compression algorithms such as RLE, HUFF, SF, and AC. The results of the standards compression algorithms obtained from [17,19,33], then compared with the proposed scheme. According to the results of [17,19,33], the HUUF, SF, and AC techniques obtained very similar against standard text files whereas RLE obtained far results. Here, we can see that, for some of the text files, the RLE has CR greater than 1, which means, this technique expands the original size file instead of the compressed file size. At most the CR of the standard algorithms is in the range of 0.57 to 1.95 whereas the CR of the proposed scheme is located in the range of 0.20 to 0.36. The results of CR on tested text files based on the proposed scheme is the best ratio achieved, this mean text size is reduced more compared to the standard compression algorithms. The BPC of the standard algorithms is in the range of 4.55 to 5.09 except the RLE range located in 8.17 to 8.12, this means more space required to store the text file. The BPC of the proposed scheme is located in the range of 1.62 to 2.94. Thus, table results explain the proposed scheme obtained less results of BPC, so it needs fewer numbers of BPC than the standard compression algorithms. Also, the table shows that the compressed file size in byte or KB is smaller than the size of the standard compression algorithms. As illustrated in Table III, the results show that the proposed scheme has better results than the standard compression algorithm mentioned above.

Table III: Comparison between Proposed and Standard Algorithms for Plaintext Data

Dataset	LZFG	RLE	HUFF	SF	AC	Proposed method
bib	2.90	8.16	5.26	5.56	5.23	3.02
book1	3.62	8.17	4.57	4.83	4.55	3.01
book2	3.05	8.17	4.83	5.08	4.78	2.99
news	3.44	7.98	5.24	5.41	5.19	2.98
paper1	3.03	8.12	5.09	5.34	4.98	2.97
paper2	3.16	8.14	4.68	4.94	4.63	3.00
progc	2.89	8.10	5.33	5.47	5.23	2.90

B. Image Datasets

In the proposed method the BPP of lena, baboon, and pappers are calculated and compared with AC and HUFF. As it is illustrated in Table IV the BPP of AC is in the range of 5.58 to 6.51, and HUFF in the range 5.84 to 6.49, this means more space required to store image. The BPP of the proposed method is located in the range of 3.71 to 3.75, this means our method needs a smaller number of BPP, so it needs fewer numbers of BPC than the standard compression algorithms which has better results than the standard compression algorithm.

Table IV: Comparison between Proposed and Standard Algorithms for Image Data

Datasets	AC	HUFF	Proposed method
lena	5.85	5.84	3.71
baboon	6.51	6.49	3.75
peppers	6.2	6.18	3.73

VI. CONCLUSION

Previously the researchers have demonstrated that the simultaneous techniques are far better than sequential techniques. An efficient simultaneous technique for image and plaintext datasets based on DNA is proposed here. For performance evaluation various files in different size are inputted to the proposed technique. Compression ratio, bit per character, bit per pixel, and compressed file size are used to evaluate the technique performance, then compared with different standard compression techniques. Results showed that the proposed scheme can compress about 62.5% of plaintext whereas 50% of image. Further the security is increased by encrypting the generated symmetric key, when the compressed data is transmitted, attackers must decrypt the key using the same procedures. In the future we can extend this work for audio as well as videos.

REFERENCES

[1] Raju, Jini. "A study of joint lossless compression and encryption scheme." In Circuit, Power and Computing Technologies (ICCPCT), 2017 International Conference, Kollam, India, pp. 1-6. IEEE, 2017.

[2] Shoba, D. Jasmine, and S. Sivakumar. "A Study on Data Compression Using Huffman Coding Algorithms." International Journal of Computer Science Trends and Technology (IJCST), ISSN: 2347-8578, Vol. 5, Issue No. 1, pp. 58-63, 2017.

[3] Sethi, G., Sweta S., Vinutha, K., and Chakravorty, C. "Data Compression Techniques." International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5, Issue No. 4, pp. 5584-5586, 2014.

[4] Sharma, N., Kaur, J., and Kaur, N. "A review on various Lossless text data compression techniques." Research Cell: An International Journal of Engineering Sciences, ISSN: 2229-6913, Vol. 12, Issue No. 2, pp. 58-63, 2014.

[5] Gautam, R., Murali, S. "An Optimized Huffmans Coding by the method of Grouping." arXiv preprint arXiv: pp. 1607.08433, Jul 2016.

[6] Mahmood, A., Latif, T., and Hasan, K. M. A. "An efficient 6-bit encoding scheme for printable characters by table look up." In Electrical, Computer and Communication Engineering (ECCE), International Conference, Cox's Bazar, Bangladesh, IEEE, pp. 468-472, Feb 2017.

[7] Sandoval, M. M., & Feregrino-Uribe, C. "A hardware architecture for elliptic curve cryptography and lossless data compression". In Electronics, Communications and Computers. CONIELECOMP 2005. Proceedings. 15th International Conference, IEEE, pp. 113-118, 2005.

[8] Ramya, K. A., Pushpa, M. "A Survey on Lossless and Lossy Data Compression Methods". International Journal of Computer Science and Engineering Communications, Vol. 4, Issue No. 1, pp. 1277-1280, 2016.

[9] Fitriya, L. A., Purboyo, T. W., & Prasasti, A. L. (2017). "A Review of Data Compression Tec niques". International Journal of Applied Engineering Research, ISSN 0973-4562, Vol. 12, Issue No, 19, pp. 8956-8963, 2017.

[10] Hosseini, M. "A Survey of Data Compression Algorithms and their Applications." Conference: Applications of Advanced Algorithms, At Simon Fraser University, Jan 2012.

[11] Al-Wattar, A. H., Mahmod, R., Zukarnain, Z. A., & Udzir, N. I. "A New DNA-Based Approach of Generating Key-dependent ShiftRows Transformation." arXiv preprint arXiv:1502.03544, 2015.

[12] Zebari, D. A., Haron, H., Zeebaree, S. R., & Zeebaree, D. Q. (2018, October). Multi-Level of DNA Encryption Technique Based on DNA Arithmetic and Biological Operations. In *2018 International Conference on Advanced Science and Engineering (ICOASE)* (pp. 312-317). IEEE.

[13] Patel, h., Itwala, U., Rana, R., and Dangarwala, K. "Survey of Lossless Data Compression Algorithms." In International Journal of Engineering Research and Technology, ESRSA Publications, Vol. 4, Issue No. 4, pp. 926-929, April 2015.

[14] Lamorahan, C., Pinontoan, B., & Nainggolan, N. "Data Compression Using Shannon-Fano Algorithm." de CARTESIAN, Vol. 2, Issue No. 2, pp. 10-17, September 2013.

[15] Kaur, K., Saxena, J., and Singh, S. "Image Compression Using Run Length Encoding (RLE)." International Journal on Recent and Innovation Trends in Computing and Communication (IJRITCC), ISSN: 2321-8169, Vol. 5, Issue No. 5, pp. 1280 – 1285, May 2017.

[16] Sarika, S., Srilali, S. "Improved Run Length Encoding Scheme for Efficient Compression Data Rate." Int. Journal of Engineering Research and Applications, ISSN: 2248-9622, Vol. 3, Issue No. 6, pp.2017-2020, Nov-Dec 2013.

[17] Sailunaz, K., Alam, M. R., Huda, M. N. "Data Compression Considering Text Files." International Journal of Computer Applications, Vol. 90, Issue No. 11, pp. 27-32, March 2014.

[18] Gomathymeenakshi, M., Sruti, S., Karthikeyan, B., Nayana, M. "An Efficient Arithmetic Coding Data Compression with Steganography." International Conference on Emerging Trends in Computing, Communication and Nanotechnology (ICECCN), IEEE, pp. 342-345, 2013.

[19] Senthil, S., and Robert, L. "Text Compression Algorithms: A Comparative Study." Journal on Communication Technology, Vol: 2, Issue No. 4, pp. 444-451, December 2011.

[20] Li, T., Zhao, T., Nho, M., & Zhou, X. "A novel RLE & LZW for bit-stream compression." In Solid-State and Integrated Circuit Technology (ICSICT), 2016 13th IEEE International Conference, IEEE, pp. 1600-1602, October 2016.

[21] Chavan, R. R., and Sabnees, M. "Secured Mobile Messaging". Int. Conf. Comput. Electron. Elect. Technol., ICCEET, IEEE, pp. 1036-1043, 2012.

[22] Setyaningsih, E., & Wardoyo, R. "Review of Image Compression and Encryption Techniques." International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 8, Issue No. 2, pp. 83-94, 2017.

[23] Sangwan, N. "Combining Huffman text compression with new double encryption algorithm." In Emerging Trends in Communication, Control, Signal Processing & Computing Applications (C2SPCA), 2013 International Conference, IEEE, pp. 1-6, October 2013.

[24] Singh, R., Panchbhaiya, I., Pandey, A., & Goudar, R. H. "Hybrid Encryption Scheme (HES): An Approach for Transmitting Secure Data over Internet." International Conference on Intelligent Computing, Communication & Convergence, Procedia Computer Science, 48, pp. 51-57, 2015.

[25] Ilakkiya, A. Pushparani, M. "An Image Compression – Encryption Hybrid Algorithm Using DWT and Chaos System." International Journal on Recent and Innovation Trends in Computi ng and Communication (IJRITCC), ISSN: 2321-8169, Vol. 4, Issue No. 4, pp. 638-643, April 2016.

[26] Ponuma, R., Aarthi, V., & Amutha, R. "Cosine Number Transform based hybrid image compression-encryption." In Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference, IEEE, pp. 172-176, March 2016.

[27] Ibrahim, F. E., Abdalkader, H. M., & Moussa, M. I. "Enhancing the Security of Data Hiding Using Double DNA Sequences." In Industry Academia Collaboration Conference (IAC), pp. 6-8, 2015.

[28] Avudaiappan, T., Ilam Parithi, T., Balasubramanian, R., Sujatha, K. "Performance Analysis on Lossless Image Compression Techniques for General Images". International Journal of Pure and Applied Mathematics, ISSN: 1314-3395, Vol. 117, Issue No. 10, pp. 1-5, 2017.

[29] Zebari, D. A., Haron, H., & Zeebaree, S. R. "Security Issues in DNA Based on Data Hiding: A Review." International Journal of Applied Engineering Research, ISSN: 0973-4562, Vol. 12, Issue No. 24, pp. 15363-15377, December 2017.