May-June 2020
ISSN: 0193-4120 Page No. 3721-3728

Enhancing Security in Elgamal Cryptosystem Using Burrows-Wheeler Transformation and Run Length Encoding

Devi. A

Assistant Professor, School of Computer Science and Applications, REVA University, Bangalore, Karnataka, India

Article Info
Volume 83
Page Number: 3721-3728
Publication Issue:
May-June 2020

Abstract
In day to day life, a secure communication is an important criteria over non-secure network channel. While transmitting the plain text, it is necessary to compress the text before encrypting the plain text, so that the speed of transmission of data, data storage space can be increased and also the redundancy of data in the plain text can be reduced. The process of encoding characters forms the different format so that fewer bits will be representing the original data whereby the size of the original data is reduced. Compression technique plays a vital role to compress the plain text. Though different compression techniques like lossy and lossless are available, the lossless compression technique can recover the original text from the reconstructed text. While compressing the larger amount of text, the reconstructed text must be identical to the original text. The Burrows-Wheeler Transform (BWT) technique of lossless compression is used in this paper to transform the plain text and the transformation permutes the order of characters. To reduce the redundancy and also to increase the efficiency of algorithm, move-to-front transformation is done by BWT. Further, the transformation code is compressed by using run length encoding so that the security will be increased after applying the cipher text in Elgamal public-key algorithm. The transmission speed, the security of data can be increased. Due to double security of the plain text, the hackers may not hack the code easily.

Keywords: *BWT, Run-Length Encoding, Elgamal*

Article History
Article Received:*19 August2019*
Revised: *27 November 2019*
Accepted: *29 January 2019*
Publication: *12 May 2020*

1. Introduction

Encryption ensures security to access confidential data by an authorized recipient and avoid accessing the data from third party. Encryption is used to protect data when transmitting the data across networks against eavesdropping of network traffic by unauthorized users [12]. It is also used to protect sensitive information by encoding and transforming the information into an unreadable cipher text and the cipher text may be decrypted into a readable form using key. Secret key and public key cryptography [13] are the two types of cryptosystems in which secret key cryptography uses the same key for both encryption and decryption. Whereas in public- key cryptography, secret key and public key is used by each user. The public key is used for Encryption and the secret key is used for decryption. The most popular public- key algorithm is Elgamal public-key cryptosystem. The design of a robust encryption algorithm in cryptanalysis is used to find and correct any weaknesses [11]. Encryption provides only security but not increasing the transmission speed [9]. To increase the data transmission speed, the data compression technique is efficient to remove the redundant character strings in a file in which the compressed file has uniform distribution of characters and it provides shorter cipher text. It also reduces the time to encrypt and decrypt the message. There are

May-June 2020
ISSN: 0193-4120 Page No. 3721-3728

three types of compression modals available namely static, semi-static and adaptive. Static modal does not depend on the data which is being compressed and it is a fixed modal known by compressor and decompressor. Semi static model is constructed from the data to be compressed and it is also a fixed modal. An adaptive modal is a function of previously compressed part of the data. The Burrows-Wheeler transformation [8], also called as block sorting and it is based on a permutation of the input sequence. Compression and Indexing are the two important applications in BWT. It was developed by Michael Burrows and David Wheeler in 1983. BWT is useful for compression and searching. A limited memory with probabilistic process is generated by the Run-Length encoding for messages. That means the probability of a given letter may depend on previous letters. It depends on lossless compression algorithm. That is the reconstructed message is exactly same as the original message. In [10] BWT, the entire sequence to be coded. For that the original sequence must be cyclically shifted from right to left and arranged in the form of lexicographic order. The sequence of last letters and the original form of sequence in the sorted list are to be coded then code is sent to the decoder. In [14], the original sequence is cyclically shifted from left to right and arranged in the form of lexicographic order. Then, finding the first and last sequence from the sorted list. The last sequence could be compressed efficiently than the original form of sequence. Here, the first sequence is considered for compression using mtf approach (Move to Front). If the last sequence is chosen for mtf, the sequence of Intermediate plaintext IM1 is larger, transmission speed of IM1 may be decreased due to decrease of compression ratio and also the security level may be reduced after applying into Elgamal algorithm. When the first sequence is chosen for mtf approach, the compression will be higher and the sequence of IM1 will be less. In this approach, considering the source alphabet and symbol at the top of the list is assigned the number 0 and the next one is assigned as 1 and so on. Then, Moving the symbols to zeroth place of every letter of the first sequence. If same letter is occurred second time, no need to transfer the letter to the top or zeroth place rather zero can be added.

Finally, the sequence of moving position of every symbol to the top. Then, the sequence is converted into binary and applied to Run-Length Encoding to increase the security level in Elgamal algorithm. The compressed code is applied for Elgamal public-key algorithm to encrypt the code. The reverse process is decryption, decompressing the sequence in Run-Length Encoding and decoding the sequence into original message in BWT.

2. Background and Related Work

The Methodologies regarding Lossless compression and comparison of different Algorithms has been proposed by S. Porwal, Y. Chaudhary, J. Joshi, M. Jain [1]. The performance of Huffman and Arithmetic encoding is compared in this paper and proved that ratio of compression for arithmetic encoding is better than the Huffman encoding. The bandwidth of channel and time is reduced in arithmetic encoding. It is proved that the speed of compression in arithmetic encoding is less than the Huffman coding. U. Khurana and A. Koul [2] have been proposed, "Text Compression and Superfast searching" and proved that an efficient technique of high compression ratios and fast search through the text. S. Kaur and V. S. Verma [3] experimented a "Design and Implementation of LZW Data Compression Algorithm". LZW compression algorithm has been implemented to prove the plain text is effectively compressed. Huffman - LZW Encoding Technique is implemented by Md. Rubaiyat Hasan [4] transmitting of a digital image from data source to a data receiver. It has been proved that transmission speed and time are better. Rajan.S. Jamgekar et.al [5] presented a "MREA" algorithm for "File encryption and decryption using secure RSA". MREA algorithm is used for encrypting files and the encrypted file is being transmitted to another end after the decryption is over. But it is good for smaller file size because it takes more time for larger file size. In [6], Monisha Sharma et.al proposed about a novel, "Approach of Image Encryption and Decryption by using partition and Scanning Pattern". Lossless encryption of image has been proposed by the author and also given access to different lengths of the encryption keys.

The rest of the paper is organized as follows. Section 3 presents the Proposed methodology. Proposed methodology with example is discussed in Section 4. Elgamal Algorithm is discussed in section 5. Result and Discussion part is explained in section 6. Finally, section 7 ends with conclusion.

3. Proposed Methodology

This proposed method consists of five phases viz., shifting of M characters towards right circularly, arrange them in lexicographic order and formation of mtff, RLE and Elgamal Scheme to check the security level. Fig.1 shows the entire process of RLE-MTFF-BWT- Elgamal scheme.

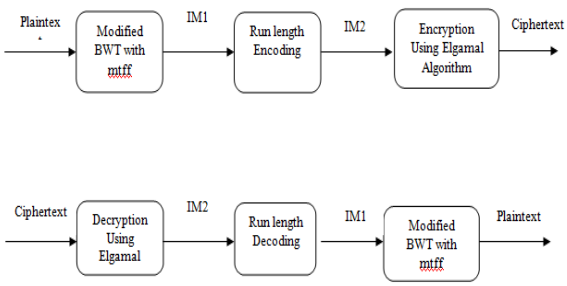


Figure 1: RLE- MTFF-BWT–Elgamal for Encryption and Decryption process

TEST Engineering & Management

May-June 2020
ISSN: 0193-4120 Page No. 3721-3728

Shifting of plaintext characters towards right circularly

Let M be a plaintext and the length of characters in M is L. L-1 sequences must be created after shifting of characters towards right circularly from the original M. Suppose d distinct characters are occurred $n_1, n_2, \dots, n_d$ times respectively. In the modified form of BWT, shifting of one position $p_i, i = 1,2, \dots, k$ towards right circularly. In row 0 it consists of original M and shifting all the elements in 0^{th} row one position towards right circularly so that row 1 is obtained. In general, $(n-1)^{th}$ row is obtained by shifting each element of $(n-2)^{th}$ row one position towards circular right. The order of sequence in modified BWT is different from original BWT.

Arranging the Alphabets in Lexicographic Order

After forming the cyclically shifted sequences, sorting the sequence in the form of lexicographical order. For that, first identify the number of blank spaces b_k in M. If number of b_k occurs in M is k, then number of words in M is $k+1$. Let the blank spaces are denoted as $b_1, b_2, \dots, b_k$ and the blank spaces $b_i \in M$ occurs at position $p_i, i = 1,2, \dots, k$. After that identify the distinct alphabets a_i, where, i =1, 2, 3, ..., d with d≤n from M and sort them in ascending order.

Formation of mtff

From the lexicographical order, the first sequence is considered for move-to-front(mtf) coding scheme. The source alphabet is $\{n_b, m_1, m_2, \dots, m_d\}$. To fill the value in row 1, the top position of 0 must be filled by b_1 (the first blank space) and the succeeding columns in row r_1 are filled with other alphabets from source alphabet. Search the position of first alphabet in the previous row and move the elements from that position to the top position and fill the rest of the alphabet in the succeeding position. The process is repeated until all d distinct elements are filled from the source sequence.

After finding the first sequence, it is converted into binary called Intermediate Message (IM1) which is then used in RLE for further compression. The result of compressed IM1 is called Intermediate Message2 (IM2).

Run Length Encoding

RLE is a technique used to reduce the size of repeating string of characters. The repeating string is called run length. It is used to replace sequences of the same data values within a file by a count number and a single value. For example, the bit stream, 11111111111111100000000000000000001111 is compressed using RLE as 15119041. In the proposed method, mtf based BWT and RLE are used before encryption and it produces the first level of security. For performing encryption and decryption, the Elgamal algorithm is used and the proposed methodology is now termed as RLE-BWT-MTFF-Elgamal.

RLE-BWT-MTFF-Elgamal

In original Elgamal, the key size taken is either 2048 or 4096-bit and it is always fixed. In the Elgamal encryption, three phases: the key generator, the encryption algorithm, and the decryption algorithm occurred. First choosing the prime number p and choosing the generator Zp^*: α and it can be shared among a group of users. Choosing the key randomly based on kϵ{1…p-1}, k and α must be less than p. So, computing the encryption, a(signature) or $y= \alpha^k$ mod p and b(signature)= y^k M mod p. Where M is the plaintext. For the decryption, $M=b/a^x$ mod p. Where x is the private key. so that the original plaintext M will be obtained from ciphertext C which increases the second level of security.

4. Proposed Methodology – An Example

In order to understand the relevance of the work, let M is "KANNAN BABA". Now d=5 with m_1='K', m_2='A', m_3='N', m_4=" ", m_5=" B" and n_1=1, n_2=4, n_3=3, n_4=1, n_5=2. Then sorted order of m_i, i=1, 2, ...,5 is {blank, A, B, K, N}. Since n==11, form the original BWT matrix of order 11×10. First, fill M in 0^{th} row then shift one position towards left circularly. The process is repeated for the last character "A" in M. It is shown in Table 4.

Table 4: Original BWT with left shift and Modified BWT with right shift

Original BWT with Left Shift													Modified BWT with Right Shift											
0	K	A	N	N	A	N	ϕ	B	A	B	A	K	0	K	A	N	N	A	N	ϕ	B	A	B	A
1	A	N	N	A	N	ϕ	B	A	B	A	K	A	1	A	K	A	N	N	A	N	ϕ	B	A	B
2	N	N	A	N	ϕ	B	A	B	A	K	A	N	2	B	A	K	A	N	N	A	N	ϕ	B	A
3	N	A	N	ϕ	B	A	B	A	K	A	N	N	3	A	B	A	K	A	N	N	A	N	ϕ	B
4	A	N	ϕ	B	A	B	A	K	A	N	N	N	4	B	A	B	A	K	A	N	N	A	N	ϕ
5	N	ϕ	B	A	B	A	K	A	N	N	A	N	5	ϕ	B	A	B	A	K	A	N	N	A	N
6	ϕ	B	A	B	A	K	A	N	N	A	N	ϕ	6	N	ϕ	B	A	B	A	K	A	N	N	A
7	B	A	B	A	K	A	N	N	A	N	ϕ	B	7	A	N	ϕ	B	A	B	A	K	A	N	N
8	A	B	A	K	A	N	N	A	N	ϕ	B	A	8	N	A	N	ϕ	B	A	B	A	K	A	N
9	B	A	K	A	N	N	A	N	ϕ	B	A	B	9	N	N	A	N	ϕ	B	A	B	A	K	A
10	A	K	A	N	N	A	N	ϕ	B	A	B		10	A	N	N	A	N	ϕ	B	A	B	A	K

May-June 2020
ISSN: 0193-4120 Page No. 3721-3728

After forming Table 4, again fill M in 0th row of Table 4. Start with first alphabet in sorted list i.e., A in row 0, fill the next row of table 4 shifting towards circularly right row 0 so that row 1 is obtained. In row 1, now the next alphabet in sorted order is B. Start with B fills all the characters which occur in row 1 by shifting towards right circularly. The process is repeated until the last character "N" is processed. The resultant is shown in Table 4. The second step is to sort the sequence in the lexicographical order shown in table 5. Table 5 consists of the cyclically shifted sequences which are in the lexicographical order.

Table 5: Lexicographic order for modified BWT

0	ϕ	B	A	B	A	K	A	N	N	A	N
1	A	B	A	K	A	N	N	A	N	ϕ	B
2	A	K	A	N	N	A	N	ϕ	B	A	B
3	A	N	ϕ	B	A	B	A	K	A	N	N
4	A	N	N	A	N	ϕ	B	A	B	A	K
5	B	A	B	A	K	A	N	N	A	N	ϕ
6	B	A	K	A	N	N	A	N	ϕ	B	A
7	K	A	N	N	A	N	ϕ	B	A	B	A
8	N	ϕ	B	A	B	A	K	A	N	N	A
9	N	A	N	ϕ	B	A	B	A	K	A	N
10	N	N	A	N	ϕ	B	A	B	A	K	A

Table 6 shows the first and last sequence of the sorted elements.

Table 6: First and last sequence of elements

F_i	ϕ	A	A	A	A	B	B	K	N	N	N
L_i	N	B	B	N	K	ϕ	A	A	A	N	A

i^{th} Element of First and Last sequence

Further, the sequence is applied to mtf coding scheme to encode the sequence. For that consider the first sequence F= "ϕAAAABBKNNN". Normally, the last sequence L is considered for mtf. But, in the proposed methodology, the first sequence F is considered so that the number of moving positions of each element to the 0th position is very less. This is because F gets the repeated element several times. The source alphabet A based on F is {ϕ, A, B, K, N}. In this coding scheme, the first element is blank space which is already in the top position and there is no change in the position. Thus, 0 is placed in the sequence. The second element is A from the sequence moving from first position into the top of the list. This is encoded as 1. The next subsequent three elements from sequence are A and hence it is not necessary to encode those elements instead and three zeros in the sequence. The next element is B moving towards the top of the list from second position and it is encoded as 2. Again, B is repeated and it is considered as zero. After B, the element K is moving towards the top of the list and it is encoded as 3. The last three elements from the sequence is N which is moving towards the top of the list from the position four and it is encoded as 4. The two elements out of three are zeros. The final sequence from this coding scheme is 01000203400. The mtf for first sequence and last sequence are shown in Table 7.

Table **7**: Move to Front Coding Scheme for first sequence and last Sequence

Move to Front Coding Scheme for First sequence						Move to Front Coding Scheme for Last Sequence								
NO.	I	II	III	IV		NO.	I	II	III	IV	V	VI	VII	VIII	IX
0	ϕ	A	B	K		0	ϕ	N	B	N	K	ϕ	A	N	A
1	A	ϕ	A	B		1	A	ϕ	N	B	N	K	ϕ	A	N
2	B	B	ϕ	A		2	B	A	ϕ	ϕ	B	N	K	ϕ	ϕ
3	K	K	K	ϕ		3	K	B	A	A	ϕ	B	N	K	K
4	N	N	N	N		4	N	K	K	K	A	A	B	B	B

The first sequence is ϕAAAABBKNNN and the mtf first sequence is 01000203400. The compressed form of mtf first sequence after applying RLE is 0130203420. The last sequence is NBBNKϕAAANA and the mtf final sequence is 43014340031. The compressed form of mtf last sequence after applying RLE is 43014342031. So, the number digits are more in mtf last sequence than in mtf first sequence. Using Elgamal algorithm performing encryption and decryption scheme.

Elgamal Algorithm

Elgamal [7], is a public key cryptosystem which is designed by Dr. Taher Elgamal and the encryption, decryption technique is done as separate functions and involves four steps namely generating the key, distribution of key, encryption and decryption. The encoded sequence from RLE-BWT algorithm is taken into Elgamal for checking the security level. The reverse process is decompression, decoding the cipher text so that the original plaintext will be available at the receiver side. Let p be a prime number and it can be shared among a group of users (g<p). So that $y=g^x$ mod p and private key k<p. Choosing k at random, relatively prime to p-1. For encryption, a(signature)=g^k mod p and b(signature)=y^k M mod p. The reverse process is decryption, M=b/a^x. For example, Let p=23, g=5, M=3. Let k=5 (random key) and x=7 (private key). The public key, $y=5^7$ mod 23=17 and a=g^k mod p=5^5 mod 23→3125 mod 23=20. Similarly, b=y^x M mod p=17^5 * 3 mod 23=1419857*3 mod 23=4259571. So, b=17. The decryption process is M=17/20^7 mod 23=21. M=17*21^{-1} mod 23=187 mod 23. M=3. The original message M is 3.

5. Results and Discussion

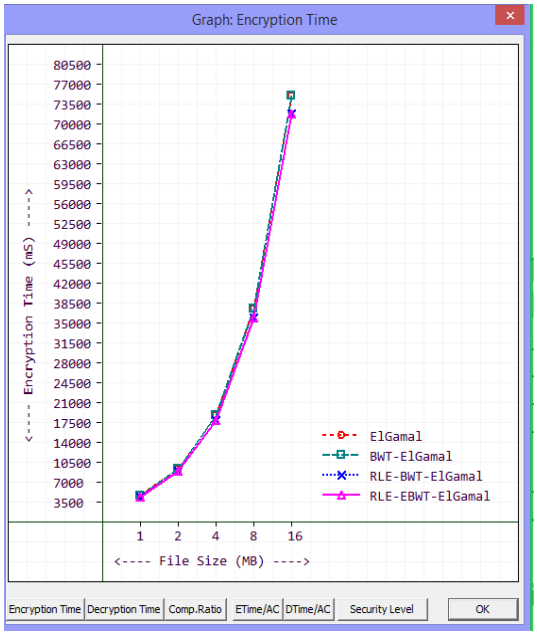
The Hackman tool is used to implement the concept BWT-RLE-Elgamal in VC++. The Encryption time

and Decryption time for Elgamal algorithm before compression is analyzed for various sizes(1MB,2MB,4MB,8MB&16MB) of plain text. The Encryption time and decryption time of 1MB file size is 4775 and 4757. The encryption and decryption time for 16 MB is 75030 and 75004.After applying BWT encoding with Elgamal algorithm, the Encryption and the Decryption time for 1MB And 16 MB are 4752,4709 and 75007,75046.After using RLE-BWT-ElGAMAL, the encryption and decryption time varies from 71754 to 71781 for 16MB.For RLE-EBWT-ElGamal, the encryption time and the decryption time lies between 71732 and 71746 of 16MB file size of plain text as shown in table 6.

Table 6: Encryption Time and Decryption Time Before Compression

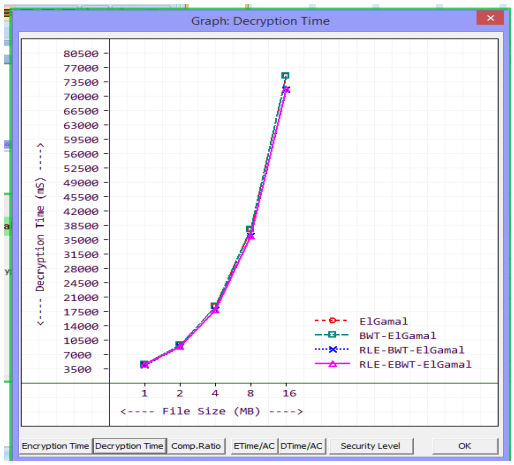
File size	Encryption Time(ms) 1MB	2MB	4MB	8MB	16MB	Decryption Time(ms) 1MB	2MB	4MB	8MB	16MB
ElGamal	4775	9394	18761	37519	75030	4757	9450	18790	37491	75004
BWT-ElGamal	4752	9405	18773	37514	75007	4709	9393	18762	37508	75046
RLE-BWT-ElGamal	4492	9029	17959	35927	71754	4510	9043	17928	35884	71781
RLE-EBWT-ElGamal	4551	9040	17965	35910	71732	4547	9047	18009	35853	71746

The fig. 2 shows the encryption time for various file sizes of plain text using RLE-EBWT-ElGAMAL before compression.



Figurre 2: Encryption time before compression

The Decryption time before compression of various file sizes of plain text using RLE-EBWT-ElGAMAL as shown in fig 3.



'Figure 3: Decryption time before compression

The Encryption time and Decryption time for Elgamal algorithm after compression is analyzed for various sizes(1MB,2MB,4MB,8MB&16MB) of plain text. The Encryption time and Decryption time of 1MB file size is 4786 and 4758 .The encryption and decryption time for 16 MB is 75034 and 75006.After applying BWT encoding with Elgamal algorithm, the Encryption and the Decryption time for 1MB And 16 MB are 4763,4710 and 75011,75048.After using RLE-BWT-ElGAMAL, the encryption and decryption time varies from 58088 to 58094 for 16MB.For RLE-EBWT-ElGamal, the encryption time and the decryption time lies between 60046 and 60053 of 16MB file size of plain text as shown in table 7.

May-June 2020
ISSN: 0193-4120 Page No. 3721-3728

Table 7: Encryption time and decryption time after compression

File size	Encryption Time(ms) 1MB	2MB	4MB	8MB	16MB	Decryption Time(ms) 1MB	2MB	4MB	8MB	16MB
ElGamal	4786	9392	18754	37517	75034	4758	9445	18788	37491	75006
BWT-ElGamal	4763	9403	18766	37512	75011	4710	9388	18760	37508	75048
RLE-BWT-ElGamal	3838	7554	15219	29746	58088	3845	7545	15171	29672	58094
RLE-EBWT-ElGamal	3961	7800	15546	30726	60046	3968	7791	15497	30650	60053

The Encryption time after compression RLE-EBWT-ELGAMAL is shown in fig.4.

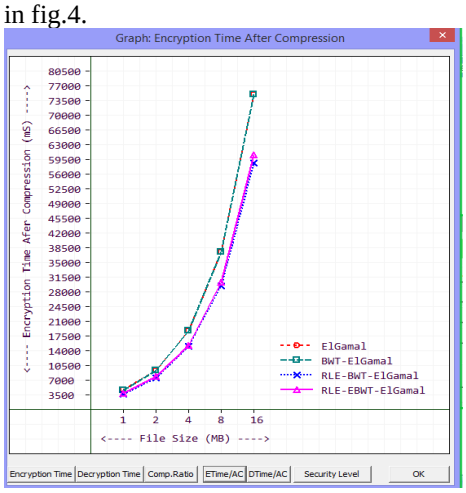


Figure 4: Encryption time after compression

The Decryption time after compression RLE-EBWT-ELGAMAL is shown in fig.5.

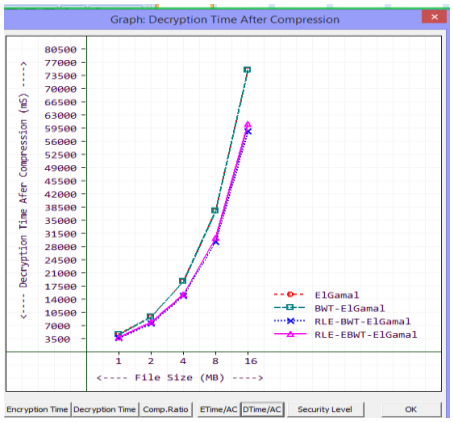


Figure 5: Decryption time after Compression

The compression ratio is same for Elgamal and BWT-ELGAMAL as 100% and also the compression ratio for RLE-BWT-ELGAMAL and RLE-EBWT-Elgamal is same for 1MB,2MB,4MB,8MB and 16MB as 90%,91%,89%,87% and 87% as shown in table 8.

Table 8: Compression Ratio

File size	Compression Ratio(%) 1MB	2MB	4MB	8MB	16MB
ElGamal	100	100	100	100	100
BWT-ElGamal	100	100	100	100	100
RLE-BWT-ElGamal	90	91	89	87	87
RLE-EBWT-ElGamal	90	91	89	87	87

The fig 6. Shows the compression ratio for various file sizes of encoded plain text.

May-June 2020
ISSN: 0193-4120 Page No. 3721-3728

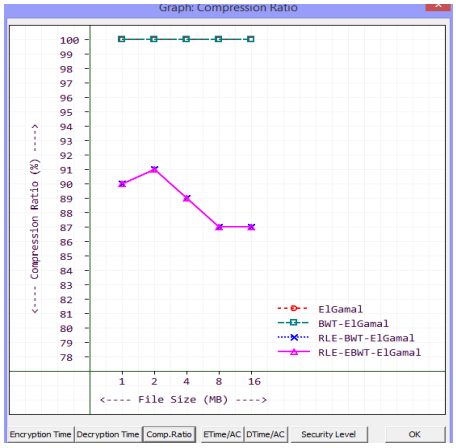


Figure 6: Compression Ratio

The table 9 shows the security level for various file sizes of cipher text.Because of run length encoding is applied after encoding using BWT and the first sequence is considered for mtf in BWT rather than last sequence ,the security level of RLE-EBWT-Elgamal is increased upto 92% for 16MB compared to RLE-BWT-Elgamal with 89%.

Table 9: Security Level

File size	Security Level (%) 1MB	2MB	4MB 8MB	16MB	
ElGamal	89	87	88	86	86
BWT-ElGamal	93	92	89	90	88
RLE-BWT-ElGamal	94	92	93	91	89
RLE-EBWT-ElGamal	97	95	95	94	92

The various levels of security are shown in fig 7.

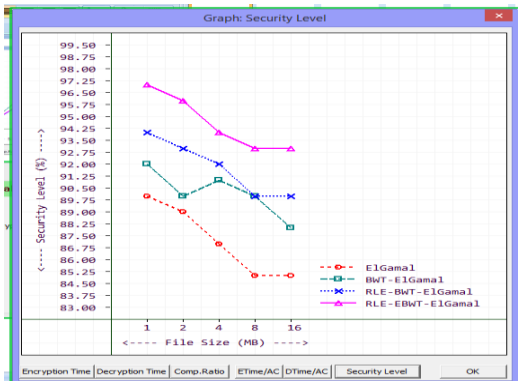


Figure 7: Security Level

6. Conclusion

A modified mtf based BWT methodology is used in this paper to encode the plaintext M and the result of IM1 is further compressed using RLE for increasing the transmission speed of M. The result of IM2 is used in Elgamal public-key cryptosystem for encrypting the ciphertext to enhance the security level and the reverse process is decryption of ciphertext with decompression of IM2 and decoding the IM1.So that the original plaintext M is obtained. The experimental result is proved that it enhances the security level using the modified mtf based BWT methodology with Elgamal public-key cryptosystem. In future, the security level of modified mtf based BWT methodology with Elgamal cryptosystem and RSA cryptosystem may be compared.

References

[1] S. Porwal, Y.Chaudhary, J.Joshi, M. Jain, “Data Compression Methodologies for Lossless Data and Comparison between Algorithms”, International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 2, Issue 2, March 2013.

[2] U. Khurana and A. Koul, “Text Compression and Superfast Searching”, Thapar Institute of Engineering and Technology, Patiala, Punjab, India-147004.

[3] S. Kaur and V.S. Verma,” Design and Implementation of LZW Data Compression Algorithm” International Journal of Information Sciences and Techniques (IJIST), Vol.2, No.4, July 2012.

[4] Md. RubaiyatHasan, “Data Compression using Huffman based LZW Encoding Technique”, International Journal of Scientific & Engineering Research Volume 2, Issue 11, November-2011.

[5] Rajan.S.Jamgekar, Geeta Shantanu Joshi, "File Encryption and Decryption Using Secure RSA",International Journal of Emerging Science and Engineering (IJESE), Vol-1, Issue-4, February 2013.

[6] Monisha Sharma, Chandrashekhar Kamargaonkar, Amit Gupta, "A Novel Approach of Image Encryption and Decryption by using partition and Scanning Pattern", International Journal of Engineering Research & Technology (IJERT), Vol. 1, Issue 7, September- 2012.

[7] https://en.wikipedia.org/wiki/Elgamal_(cryptosystem)#Key_distribution

[8] D. Adjeroh, T. Bell, and A. Mukherjee, “The Burrows-Wheeler Transform :: Data Compression, Suffix Arrays, and Pattern Matching”, Springer, 1 edition, July 2008.

[9] K.Mani and A.Devi ,”Enhancing Security in Cryptographic Algorithms Based on IENCCRS Scheme”, IJAER, Vol.10, No.82, 2015.

[10] J. Seward,”On the performance of bwt sorting algorithms”, In Data Compression Conference, pages 173–182, IEEE Computer Society, 2000.

[11] https://en.wikipedia.org/wiki/Elgamal.

May-June 2020
ISSN: 0193-4120 Page No. 3721-3728

[12] William Stallings," Cryptography and Network Security, Principles and Practices", Fourth Edition , November, 2005.

[13] Hans Delfs and Helmut Knebl., "Introduction to Cryptography Principles and Applications", Springer- Verlag, Berlin, Heidelberg, 2001.

[14] K.Mani and Devi.A, ,"Enhancing Security in RSA Cryptosystem Using Burrows-Wheeler Transformation and Run Length Encoding", International Journal of Scientific Research in Computer Science Engineering and Information Technology(ijsrcseit), Volume 3, Issue 1, ISSN:2456-3307.