Data Security using Compression and Cryptography Techniques

Ruchita Sharma
M.Tech Student (CSE)
Mukesh Patel School of Technology,
Management & Science, Vile Parle (West),
Mumbai (Maharashtra)

Swarnalata Bollavarapu
Assistant Professor
Mukesh Patel School of Technology,
Management & Science, Vile Parle (West),
Mumbai (Maharashtra)

ABSTRACT
Data is any type of stored digital information. Security is about the protection of assets. Data security refers to protective digital privacy measures that are applied to prevent unauthorized access to computers, databases and websites. Cryptography is evergreen and developments. Cryptography protects users by providing functionality for the encryption of data and authentication of other users. Compression is the process of reducing the number of bits or bytes needed to represent a given set of data. It allows saving more data. The project aims to implement various cryptography algorithm for data security. The data will be first compressed using compression techniques and then encryption techniques will applied and then comparative analysis will be carried out for different combinations of compression and encryption techniques. If encryption and compression are done at the same time then it takes less processing time and more speed.

Keywords
Cryptography, Compression, Run length, Huffman, LZW, Arithmetic coding, RC4, Caesar Cipher, DES.

1. INTRODUCTION
Need of security is to ensuring that your information remains confidential and only access to authorized user and ensure that no one has been able to change your information, so it provide full accuracy. To secure the data, compression is used because it use less disk space (saves money), more data can be transfer via internet. It increase speed of data transfer from disk to memory. Security goals for data security are Confidential, Authentication, Integrity, and Non-repudiation. Data security delivers data protection across enterprise. Data compression is known for reducing storage and communication costs. It involves transforming data of a given format, called source message to data of a smaller sized format called code word [1]. Data encryption is known for protecting information from eavesdropping [1]. It transforms data of a given format, called plaintext, to another format, called cipher text, using an encryption key [1]. Currently compression and encryption methods are done separately [1]. The major problem existing with the current compression and encryption methods is the speed, the processing time required by a computer, more cost [1]. To overcome this disadvantage, combine the two processes into one.

2. CRYPTOGRAPHY
To hide any data two techniques are mainly used one is Cryptography other is Steganography. In this paper we use Cryptography. Cryptography is the science of protecting data, which provides methods of converting data into unreadable form, so that Valid User can access Information at the Destination [4]. Cryptography is the science of using mathematics to encrypt and decrypt data [4]. Fig. 1 shows cryptographic process in which P represent Plaintext, C represent Ciphertext, E stands for Encryption & D stands for Decryption.

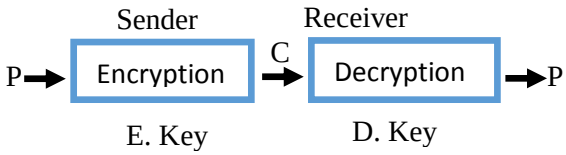

Fig 1: Cryptography Process

3. COMPRESSION
Data compression implies sending or storing a smaller number of bits. Compression is the reduction in size of data in order to save space or transmission time [5]. Many methods are used for this purpose, in general these methods can be divided into two broad categories: **Lossy** and **Lossless** methods. Lossy Compression generally used for compress an Images. In this original data is not identical to compressed data that means there is some loss e.g. Block Truncation Coding, Transform Coding, etc... Lossless Compression used for compress any textual data. In this original data and compressed data are equal that means there is no loss e.g. Run Length Coding, Huffman Coding, LZW, Arithmetic Coding.

4. METHODOLOGIES

4.1 Compression Techniques

4.1.1 Run Length Encoding
Run-length encoding is the simplest method of compression. It can be used to compress data made of any combination of symbols [7]. The idea behind this method is to reducing the size of a repeating string of characters. If data have more than two consecutive characters then this method would give better result.

4.1.2 Huffman Coding
A Commonly used method for data compression is Huffman coding. The Huffman algorithm is based on statistical coding, which means that the more probable the occurrence of a symbol is, the shorter will be its bit-size representation [8]. In any file, certain characters are used more than others. Using binary representation, the number of bits required to represent each character depends upon the number of characters that have to be represented [8].

4.1.3 LZW

LZW compression is a lossless compression. It compress a file into a smaller file using a table-based lookup algorithm invented by Abraham Lempel, Jacob Ziv, and Terry Welch. It is a 'dictionary based' compression algorithm that scan a file for sequences of data that occur more than once [6]. These sequences are then stored in a dictionary and references are put where-ever repetitive data occurred [6].

4.1.4 Arithmetic Coding

Arithmetic coding is a form of entropy encoding used in lossless data compression. Arithmetic coding, which is a method of generating variable-length codes, is useful when dealing with sources with small alphabets such as binary sources [9]. It encodes data (the data string) by creating a code string which represents a fractional value on the number line between 0 and 1. Replace the entire input with a single floating-point number.

4.2 Cryptographic Techniques

4.2.1 RC4 (Rivest Cipher)

RC4 design by Ron Rivest of RSA Security in 1987. It is stream cipher, Symmetric key encryption. RC4 kept as a trade secret by RSA Security earlier. But someone anonymously posted RC4 code on internet it was a big loss. After that RSA security tell that it is still a trade secret but it was too late. The algorithm is used for both encryption and decryption as the data stream is simply XORed with the generated key sequence [11]. It uses a variable length key from 1 to 256 bit to initialize a 256-bit state table [11].

4.2.2 Caesar Cipher

Caesar Cipher is named after Julius Caesar. It is one of the simplest and most widely known encryption techniques. It shifts the alphabet. The key is the number of letters you shift. It is a Substitution Cipher that involves replacing each letter of the secret message with a different letter of the alphabet which is a fixed number of positions further in the alphabet [12].

4.2.3 DES

The Data Encryption Standard (DES) is based on a symmetric-key algorithm that uses a 56-bit key. DES is a block cipher, which means that during the encryption process, the plaintext is broken into fixed length blocks and each block is encrypted at the same time [13]. DES consists of 16 steps, each of which called as a Round.

5. IMPLEMENTATION

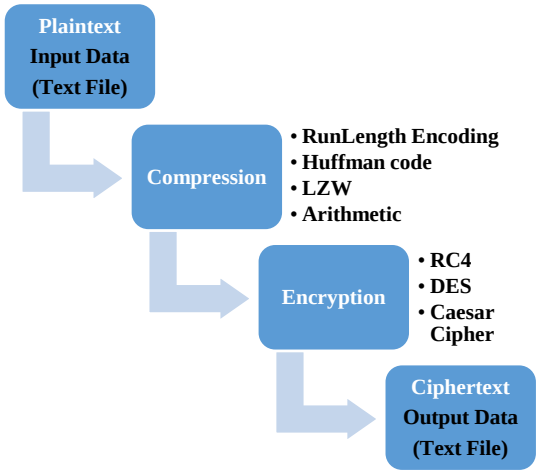

Fig 2: Flow of Work

Fig. 2 shows Flow of work of the combination of compression and encryption. In this, four compression & three cryptographic techniques are applied on text data then calculate performance analysis with respect to file size, their compression ratio and their execution time. Flow of work is to take any text file first we compress a text file after that whatever output will get that will be go to encryption part to encrypt that file. Combination of compression and encryption algorithms are applied on text file are as follows:

Four compression techniques which are RLE (Run Length Encoding), Huffman coding, LZW, Arithmetic and three cryptographic techniques which are RC4, Caesar Cipher and DES are used. To secure the data more we use combination of compression and cryptographic techniques on text file.

So here the combination which are used to secure data are: RLE + (RC4 & Caesar Cipher), Huffman + (RC4 & DES), LZW + (RC4 & DES), Arithmetic + (RC4 & DES) with their corresponding file size, compression ratio and execution time.

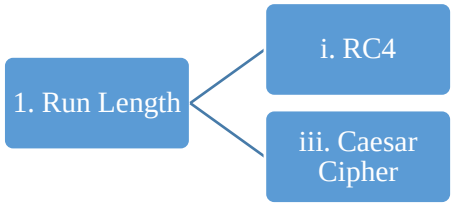

Fig 3: RLE + (RC4/Caesar Cipher)

Fig 3 shows Run length encoding with RC4 and Caesar Cipher will apply on five different size of text file and result will be carried out. This combination is good when two or more consecutive character will occur in any text file. If there are less number of consecutive character in any text file then result will be not good as comparative to other techniques.

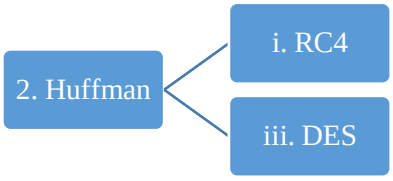

Fig 4: Huffman + (RC4/DES)

Fig 4 shows Huffman with RC4 & DES will apply on those five different size of text file. This combination is best over any compression techniques. Their compression ratio is very good. If Huffman coding is used to compress a file then the file size is just half and it is good to send the more data on internet.

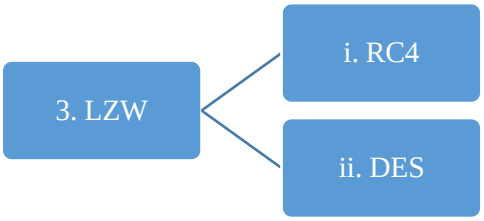

Fig 5: LZW + (RC4/DES)

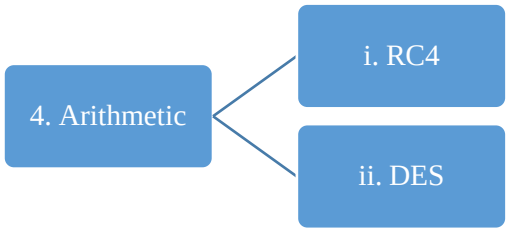

Fig 6: Arithmetic + (RC4/DES)

Fig 5 & 6 shows LZW & Arithmetic with RC4 & DES will apply on those five different size of text file. It compress a file and give result that is good but not as a good as Huffman will give. There are minor difference in values after compression of both the techniques (Arithmetic & LZW). After compression, encryption will perform and there is not as much difference in values.

6. RESULT AND DISCUSSION

After applying the combination following result will be carried out. Here we take five different text file which are different in size. After the result we analysis that higher the file size better the compression ratio. After the compression we perform encryption to give better security. Various combination which are used on text file that shows which combination we will use for better security and better compression ratio.

Table 1 Performance Evaluation With Respect To File Size

Original (O) File Size (In Bytes)	O + 1	1 + i	1 + iii	O + 2	2 + i	2 + ii	O + 3	3 + i	3 + ii	O + 4	4 + i	4 + ii
9571	9324	9324	9324	**289**	**289**	**296**	5272	5272	5280	6057	6057	6064
30320	30314	30314	30314	**645**	**645**	**648**	17574	17574	17576	17767	17767	17768
50587	50585	50585	50585	**707**	**707**	**712**	27102	27102	27104	29732	29732	29736
101433	101418	101418	101418	**696**	**696**	**704**	43268	43268	43268	57444	57444	57448
243945	243906	243906	243906	**697**	**697**	**704**	121643	121643	121649	136546	136546	136552

Table 1 shows values of five different file size of text file. All the techniques which are used and it shows that which combination gave better result. And we can see that Combination of Huffman coding will give better result as compared to other combination of compression and cryptographic techniques.

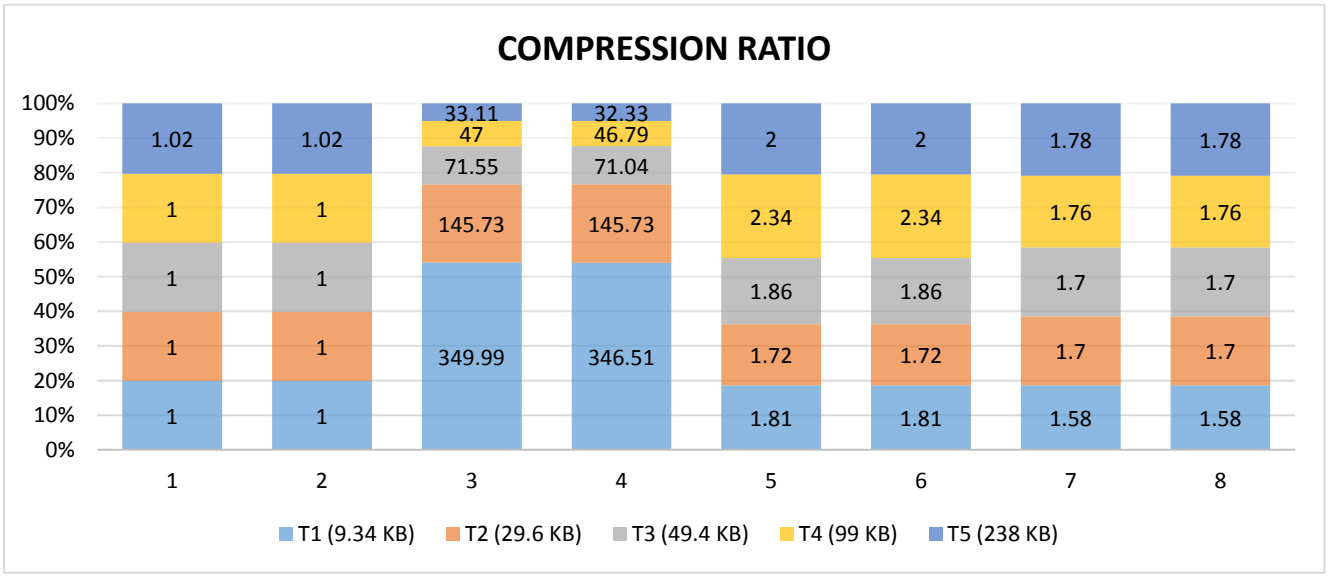

Fig 7: Performance Evaluation With Respect To Compression Ratio

Fig. 7 shows Compression Ratio of five different text file. Here we can see that Huffman gave better Compression Ratio among all other techniques almost 50-80%. So overall best compression algorithm is Huffman then LZW then Arithmetic then run length will give better result for text compression.

7. CONCLUSION

In this paper we evaluate the performance with respect to different parameters. It shows basic information about cryptography and compression, & their techniques are applied on text file. For data security, combination of compression and cryptographic techniques are used. To secure our data more that's why we compressed the data first and then encrypt that compressed data. It has many advantage of doing this we can transfer more and more data via internet. If combination is used it may be less costly, it save time, more secure.

In future, different cryptography techniques can be applied with the combination of compression techniques. Other compression technique will also be implemented. Performance analysis will be carried out for implementing techniques. After getting the result we will able to design a new algorithm that will be used for data security.

8. REFERENCES

[1] T.SubhamastanRao, M.Soujanya, T.Hemalatha, T.Revathi, "Simultaneous data compression and encryption" (IJCSIT) International Journal of Computer Science and Information Technologies, ISSN 0975-9646, Volume-2(5), 2011.

[2] Senthil Shanmugasundaram, Robert Lourdusamy "A Comparative Study Of Text Compression Algorithms" International Journal of Wisdom Based Computing, Vol. 1 (3), December 2011.

[3] Harshraj N. Shinde, Aniruddha S. Raut, Shubham. Vidhale, Rohit V. Sawant, Vijay A. Kotkar "A Review of Various Encryption Techniques" International Journal of Engineering And Computer Science ISSN: 2319-7242, Volume 3, Issue 9, September 2014.

[4] Ms. Ayushi Aggarwal, Anju "Enciphering Data for Larger Files" International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 3, Issue 5, May 2013.

[5] Haroon Altarawneh, Mohammad Altarawneh "Data Compression Techniques on Text Files: A Comparison Study" International Journal of Computer Applications, (0975 – 8887), Volume 26– No.5, July 2011.

[6] MohiniChaudhari, Dr. KanakSaxena "Fast and Secure Data Transmission using Symmetric Encryption and Lossless Compression" International Journal of Computer Science and Mobile Computing, ISSN 2320–088X, Vol. 2, Issue. 2, February 2013.

[7] "Data Compression" by Behrouz Forouzan.

[8] "Huffman Compression" by webopedia.

[9] Yu-Yun Chang "Tutorial: Arithmetic Coding".

[10] T.D.B Weerasinghe "Analysis of a Modified RC4 Algorithm" International Journal of Computer Applications, ISSN0975 – 8887, Volume 51– No.22, August 2012.

[11] AL.Jeeva, Dr.V.Palanisamy, K.Kanagaram "Comparative Analysis of Performance Efficiency and Security Measures of Some Encryption Algorithms" International Journal of Engineering Research and Applications (IJERA), ISSN: 2277 128X, Volume 3, Issue 6, June 2013.

[12] Mr. Vinod Saroha, Suman Mor, Anurag Dagar "Enhancing Security of Caesar Cipher by Double Columnar Transposition Method" International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X, Volume 2, Issue 10, 2012.

[13] Sombir Singh, Sunil K. Maakar, Dr.Sudesh Kumar "Enhancing the Security of DES Algorithm Using Transposition Cryptography Techniques" International Journal of Advanced Research in Computer Science and Software Engineering, ISSN 2249-6343, Volume 2, Issue 1, Jan 19 2012.