Asian Journal of Engineering and Applied Technology
ISSN: 2249-068X (P) Vol. 8 No. 2, 2019, pp. 37-40
© The Research Publication, www.trp.org.in
DOI: https://doi.org/10.51983/ajeat-2019.8.2.1141

An Intelligent Multimedia Data Encryption and Compression and Secure Data Transmission of Public Cloud

Sheik Saidhbi
Assistant Professor, Department of Information Systems, Faculty of Informatics, University of Gondar, Ethiopia
E-Mail: sfajju.syed@gmail.com
(Received 7 March 2019; Revised 31 March 2019; Accepted 2 May 2019; Available online 10 May 2019)

Abstract **- Data compression is a method of reducing the size of the data file so that the file should take less disk space for storage. Compression of a file depends upon encoding of file. In lossless data compression algorithm there is no data loss while compressing a file, therefore confidential data can be reproduce if it is compressed using lossless data compression. Compression reduces the redundancy and if a compressed file is encrypted it is having a better security and faster transfer rate across the network than encrypting and transferring uncompressed file. Most of the computer applications related to health are not secure and these applications exchange lot of confidential health data having different file formats like HL7, DICOM images and other audio, image, textual and video data formats etc. These types of confidential data need to be transmitted securely and stored efficiently. Therefore this paper proposes a learning compression- encryption model for identifying the files that should be compressed before encrypting and the files that should be encrypted without compressing them.**
Keywords: **Encryption, Compression, Data Security, ECB**

I. INTRODUCTION

Data compression is a method of reducing the size of the data file so that the file should take less disk space for storage [11]. The file that contains redundancy gets reduced by compression. In lossy data compression algorithms there is loss of original data while performing compression [20]. In computer science and Information technology, a data encoding method in which the data is compressed by losing some amount of data is lossy compression.

A. Lossless Data Compression Algorithms

In case of lossless data compression algorithm there is no data loss while compressing a file, it guarantees to reproduce the exactly same data as input. If data loss is not desirable the lossless data compression algorithms should be used. Some of the peculiar examples include executable text documents, programs and source codes etc. Some of the image file formats also uses lossless compression.

Huffman Code-It assign more bits to symbols appear less and fewer bits to the symbols that occur more frequently. Every Huffman code having the same average code length. It optimizes the single byte at time.

1. Deflate- It combines the LZ77 and Huffman code for compression in which LZ77 optimizes sequence of bytes whereas Huffman works on single byte.
2. LZ4-It lossless data compression algorithm that is primarily focused on compression and decompression speed. The algorithm gives a slightly worse compression ratio than others.
3. LZF- It is a fast compression algorithm that takes very little working memory and code space.
4. Zip- Zip is a lossless compression Algorithm. Zip compress as well as archive the file. Content of the .Zip file can be a single file of group of files enclosed in a folder.

B. Cryptographic Algorithms

Symmetric key cryptographic ciphers come in two types, stream and block ciphers. Stream ciphers works on bits stream or bytes stream. Stream ciphers are used for securing data of terminal and wireless applications. Block ciphers performs encryption or decryption on fixed size block of data. In network applications block ciphers are used for transmission of files of huge sizes which require high security. Deciphering cipher text without knowing the key is called cryptanalysis. Cryptanalysis of block ciphers is difficult compared to stream ciphers [9]. Hence in most of the applications, block ciphers are used for providing better security than stream ciphers.

Block ciphers come in various block modes. Block mode for cipher algorithm determines how cipher text blocks are created by encryption from plaintext blocks and vice versa. ECB, CBC, CFB, OFB, PCBC and CTR etc are commonly used block modes [9]. ECB has poor security properties since encryption of a block with a fixed size always yields the same result; hence susceptible to dictionary attacks, replay attacks etc. In case of CBC first plaintext block is XORed with IV and remaining all plaintext blocks are XORed with previous cipher text blocks; while in case of PCBC operation on first plaintext block is similar to CBC but remaining all plaintext blocks are XORed with previous plaintext as well as previous cipher text block [9] [12].

In paper [15], the author has simulated different symmetric key cryptographic algorithms like AES, DES, 3-DES and Blowfish. The simulation was done on 0.5 to

20MB data blocks. The simulation results show that the Blowfish yields better results than other symmetric key

cryptographic algorithms when it comes to processing power. AES yield poor results as it requires high processing power. Initially all the simulations were taken in ECB mode and it was observed that Blowfish takes comparatively less processing time than others. AES takes relatively higher time when the block size is high. It was also concluded that 3-DES will always take more time as compared to 3-DES as it involves 3 phases of encryption. Another simulation was done on all the symmetric key algorithms in CBC mode. It was concluded that CBC took more time for performing encryption than ECB mode.

II. DESIGN OF MODEL OF COMPRESSION ALGORITHM FOR DATA TYPE ANALYSIS

Every compression algorithm has different level of compression for different files. For example, there are two compression algorithms A and B. Suppose compression algorithm A has compression ratio of 30% for X file. It is not necessary that compression algorithm B will have same compression Ratio for X file. Compression Algorithm B can give more, less or no compression at all. This sub section provides information on calculating the compression ratio for different compression algorithms. The following algorithm describes the process for calculating compression ratio for a file.

Procedure Name: Compress

Input Parameters: input File: Name of input file for compression

Start Procedure

Array = Read all the compression algorithm in Array

Read input file for compression

Determine the data type from file extension

Find uncompressed size of file for i = 1 to Array Length

Compress file by Compression Algo[i] Calculate compressed file size;

Calculate compression Ratio by eq. 1

End for

Write the compression ratio to database

End Procedure

All cipher algorithms are implemented using sun provider except skipjack, which is implemented using Bouncy Castel provider. Fo llowing is the piece of code used for analysis:

```
// BC = Bouncy Castel Provider
cipher = Cipher.getInstance(algorithmName, "BC");

// for encryption

operation = Cipher.ENCRYPT_MODE;

//Initializing cipher cipher.init(operation, secretKey);

//Performing Encryption

encryptOutLength = cipher.update(inputBytes, 0,
bufferSize, outputBytes);

encryptOutLength = cipher.doFinal(inputBytes, 0,
inLength);

// for decryption

operation = Cipher.DECRYPT_MODE;

//Initializing cipher cipher.init(operation, secretKey);
//Performing Decryption

decryptOutLength = cipher.update(inputBytes, 0,
bufferSize, outputBytes);

decryptOutLength = cipher.doFinal(inputBytes, 0,
inLength);
```

From the related works, it is realized that none of the work did a very detailed analysis of the performance of various symmetric algorithms, on various parameters of different type of files. In order to select the most suitable cryptographic algorithm for encryption, following test cases are considered to analyze the time taken for encryption by various cryptographic algorithms.

A data file format represents the standard for encoding the information to be stored in computer file. This case study is taken to check whether the encryption has dependency on type of data or not. Different data type files like audio, image, textual, video and health data file format like DICOM of nearly 50MB, and 100MB in size are chosen and encryption time of different cipher algorithms is calculated for these data.

A. Data Files of Same Type with Different Sizes

This case study is taken to ensure once again the observations obtained from case study 1, that encryption time depends on number of bytes in the file. In this study is, different files of same types but different sizes are given for encryption and estimated the encryption time. For all executions, key size and block mode are kept at bare minimal parameters.

Table I gives the details about the files used for all executions and Figure 3, 4 and 5 show the execution results.

TABLE I EXECUTION PARAMETERS FOR FILES OF DIFFERENT SIZES

File Type	Varying Parameters (Data Size)	Constant Parameters
AIFF	10.7MB, 50MB, 100MB	Data Type, Key size, Block Mode
AVI	50MB, 100MB, 482MB	
DICOM	14.9MB, 50.3MB, 115MB, 151MB	

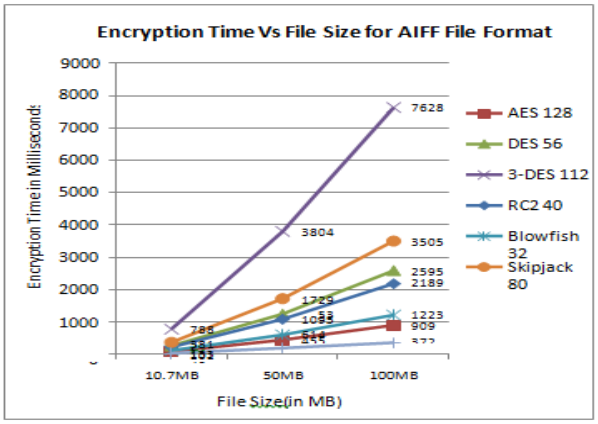

Fig. 1 File size Vs Encryption time for AIFF file of different sizes

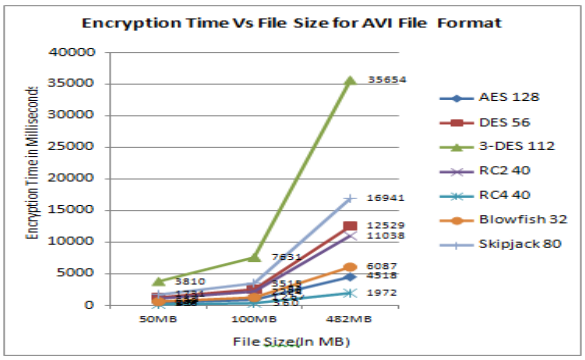

Fig. 2 File size Vs Encryption time for AVI file of different sizes

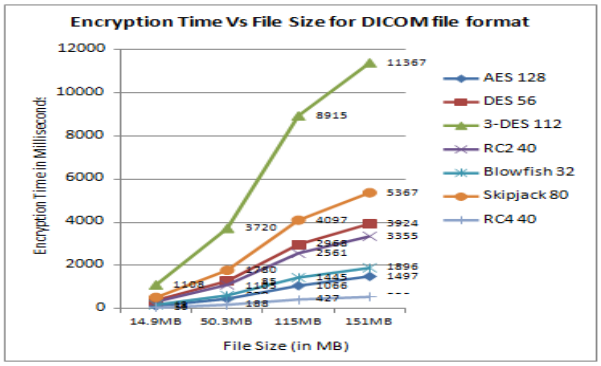

Fig. 3 File size Vs Encryption time for DICOM file of different sizes

TABLE II ENCRYPTION RATE FOR FILES OF DIFFERENT SIZES

File Type	Size (In MB)	Encryption Rate In (MB/sec)						
		AES 128	DES 56	3-DES 112	RC2 40	Blowfish 32	Skipjack 80	RC4 40
AIFF	10.7	109.18	39.43	13.61	45.06	80.63	28.14	268.12
	50	109.95	39.92	13.15	45.68	81.47	28.93	270.66
	100	110.07	38.55	13.11	45.71	81.81	28.54	268.96
AVI	50	107.83	39.49	13.14	45.04	79.62	28.93	265.52
	100	109.3	38.8	13.14	45.11	79.19	28.54	270.72
	482	108.74	38.49	13.52	45.09	79.22	28.46	265.55
DICOM	14.9	107.65	38.93	13.45	45.46	80.61	28.19	266.31
	50.3	108.48	39.17	13.53	45.63	81.19	28.28	267.76
	115	108.69	39.03	12.99	45.24	80.18	28.28	271.34
	151	108.21	38.61	13.32	45.15	79.91	28.23	270.47
Average		108.8	39.04	13.3	45.32	80.383	28.452	268.5

B. Files with Different Densities of Data

Encryption rate is evaluated for different files, a sparse AIFF file of 69MB and a dense AIFF file of 58.5MB. For a cipher algorithm, key size and block mode are kept at bare minimal parameters. The results of execution are shown in Table III.

C. Cipher Algorithms with Different Block Modes

Security of cipher algorithm also varies according to block cipher modes. Different block cipher modes are used for different applications. For example PCBC is used in WASTE and Kerberos v4. Security levels may differ according to type of application and can be classified as:

This study is to check the encryption time variation with respect to block cipher modes. All block cipher algorithms have been executed for different block modes with PKCS#5 padding scheme on 50.5MB DICOM file. The key size for particular block cipher algorithm is kept at the bare minimal value. The various block modes mentioned in Table II are used for evaluation. The Figure 4 shows the block cipher variation for AES 128 and Figure 5 shows the result of execution for all encryption algorithms.

TABLE III ENCRYPTION RATE FOR SPARSE AND DENSE DATA FILE

Algorithm Name	Sparse (72000118 Bytes) AIFF file		Dense (61392454 Bytes) AIFF file	
	Encrypt Time(ms)	Encryption Rate(MB/s)	Encrypt Time(ms)	Encryption Rate(MB/s)
AES 128	634	108.28	540	108.40
DES 56	1801	38.11	1537	38.08
3-DES 112	5076	13.52	4365	13.41
RC2 128	1520	45.16	1285	45.55
Blowfish 128	854	80.38	723	80.96
Skipjack 128	2386	28.77	2042	28.66
RC4 128	253	271.35	216	271.01

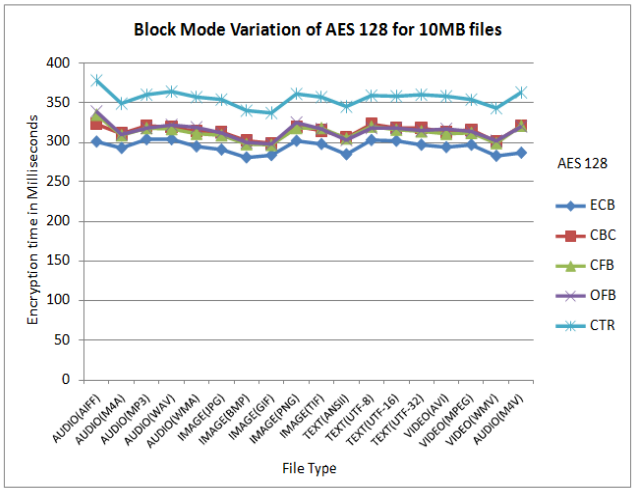

Fig. 4 Block Mode Variation of AES 128 for 10MB files

III. CONCLUSION

After analysis of all parameters, AES was found to be most suitable encryption algorithm having encryption rate of 108MB/sec in ECB mode. AES was used in the proposed compression-encryption model. A compression-encryption model was proposed for identifying the files that should be compressed before encrypting and the files that should be encrypted without compressing them. A formula was derived imperially to determine best suitable compression algorithm that should be used for compressing the file according to data type and data size to reduce the overhead of time for compression and to increase the efficiency and security to data that is being transferred.

REFERENCES

[1] A. L. Jeeva, Dr. V. Palanisamy and K. Kanagaram, "Comparative Analysis of Performance Efficiency and Security Measures of Some Encryption Algorithms", *International Journal of Engineering Research and Applications (IJERA)*, Vol. 2, No. 3, pp.3033-3037, May-Jun 2012.

[2] S. Soni, H. Agrawal and M. Sharma, "Analysis and comparison between AES and DES Cryptographic Algorithm", *International Journal of Engineering and Innovative Technology*, Vol. 2, No. 6, pp. 362-365, Dec. 2012.

[3] Nidhi Singhal and J. P. S. Raina, "Comparative Analysis of AES and RC4 Algorithms for Better Utilization", *International Journal of Computer Trends and Technology*, Vol. 2, No. 6, pp. 177-181, July-Aug 2011.

[4] Jawahar Thakur and Nagesh Kumar, "DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis", *International Journal of Emerging Technology and Advanced Engineering*, Vol. 1, No. 2, pp. 6-12, Dec. 2011.

[5] Allam Mousa and Ahmad Hamad, "Evaluation of the RC4 Algorithm for Data Encryption", *International Journal of Computer Science & Applications*, Vol. 3, No. 2, pp. 44-56, June 2006.

[6] Aamer Nadeem, M. Younus Javed, "A Performance Comparison of Data Encryption Algorithms", *First International Conference on IEEE Information and Communication Technologies (ICICT)*, pp. 84-89, 27-28 Aug. 2005.

[7] Kofahi, N.A, Turki Al-Somani, Khalid Al-Zamil, "Performance evaluation of three Encryption/Decryption Algorithms", *IEEE 46th Midwest Symposium on Circuits and Systems*, pp. 790-793, 30-30 Dec. 2003.

[8] Jonathan Knudsen, *Java Cryptography*, 2nd Edition, O'Reilly, 2011.

[9] William Stallings, *Cryptography and Network Security*, 5th Edition, Pearson, 2012.

[10] O.S. Pianykh, *Digital Imaging and Communications in Medicine (DICOM)*, 1st Edition, Springer, 2008.

[11] John Miano, *Compressed Image File Formats*, 1st Edition, Addison Wesley Longman, Inc, 1999.

[12] Atul Kahate, *Cryptography and Network Security*, 2nd Edition, Tata McGraw Hill, 2012.

[13] V.K. Pachgare, *Cryptography and Information Security*, 1st Edition, PHI Learning PVT LTD, 2008.

[14] Dworkin, M. NIST Special Publication 800-38A. Recommendation for block cipher Modes of operation: Modes and techniques, Dec. 2001.

[15] Simar Preet Singh and Raman Maini, "Comparison of Data Encryption Algorithms", *International Journal of Computer Science and Communication (IJCSC)*, Vol. 2, No. 1, pp. 125-127, January-June 2011.

[16] Lalit Singh and R.K. Bharti, "Comparative Performance Analysis of Cryptographic Algorithms", *International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE)*, Vol. 3, No. 11, pp. 563-568, November 2013.