Double Encrypted Key Based AES Combined Coding For Improved Cloud Security

K. Srilakshmi[1*], P.Bhargavi[2]
[1]Research Scholar, [2]Assistant Professor
Department of Computer Science
Sri Padmavati Mahila Visvavidyalayam
Tirupati, Andhra Pradesh, India.
[1]srilakshmi21kakarla@gmail.com, [2]pbhargavi18@yahoo.co.in

Abstract

With the advent of the technology, documents that are being accessed on the fly are increasing, and surprisingly data that is given as input for specific applications designed for specific purpose via online are increasing exponentially. Hence, providing security to such data has become an imperative task. Cloud computing in this juncture provides access to these applications that can be remotely accessed such that necessary tasks can be completed from a remote place. Hence, it is the need of the hour to develop techniques that reduce the storage complexity and complete the task in less time. In this paper, it is proposed to use Huffman and LZMA for compression of the input media and then encrypt the data, resulting in compressed data and securing the same by AES based encryption. The keys used for encryption are further encrypted using RSA provides additional security to stored data on cloud. The proposed system automatically selects the type of compression mechanism to be applied to the media. Results show that the data is intact and the compression achieved by LZMA is about 1.98% more than that of the Huffman.

Keywords: *AES, Huffman coding, LZMA, RSA, Cloud computing, security, compression.*

1. Introduction

Securing data is an evergreen problem, which is the most common area of research for hackers as well as research to give problems and solutions, respectively. As the data size is increasing new way of accessing apps and data from the cloud has erupted. Still, there are several threats and attacks which need to be considered while processing data on the cloud. These are mainly from malicious insiders, breach of data, hijacking, malware injection, attacks due to unauthenticated logins, etc., listed to be few, but not limited [1]. Figure 1 shows a cloud model wherein different types of files are processed on the cloud and need to be secured. Cloud computing [7] has paved the way to several problems viz., cloud storage wherein the files need to be stored in an effective and secure environment that provides a safe shell from hackers and intruders [2]. Further, storage complexity of on cloud increases as multiple applications accessing increase. Hence, in this work, a novel mechanism has been created for this purpose.

Jahoon Koo et al. [3] has discussed safety prerequisities for cloud computing in Korea and finalized some standards that must be followed. Carlo Di Giulio et al. [4] discussed cloud standards and are compared. FedRAMP and C5 are compared, based on which further analysis is suggested. In general, it is a known fact that no solution can provide complete security in the cloud [5]. Anmol Rastogi and Amit Agarwal proposed a combination of symmetric and asymmetric encryption mechanism which is termed as Hybrid Cryptographic System having a better security on the cloud. Salting and hashing are used for providing strength to the algorithm [6]. But the efficiency tends to be low.

Figure 1. Cloud Security required for prevention of threats and attacks

The three significant dimensions of cloud security is having secured transmission in the form of viz., information, computer and network security. Computer security mainly deals with attacks and types of attacks. Spoofing, tampering, Denial of service (DoS) and backdoor attacks are some of the attack types which have more prominence. These attacks are applied to the layers which pave the way for other dimension problem in cloud security [14], which is network security. Cryptography, on the other hand, provides better security in contrast to the existing techniques. In this paper, AES [12] is used to provide such security. Network security can be enhanced using Secured Socket Layer (SSL). Information security can be handled using authentication services, maintaining proper authentication from the users at the login side [8]. An argument arises to learn the state of the cloud for security enhancement which in general is dynamic and uncertain aswell. Hence, probabilistic models are used to learn and predict the stage of security [9]. The apps that are deployed on the cloud must consider all types of possible security options that are available on a server or on a standalone system [10]. Xiang Li et al. [11] has proposed a reputation model that checks for certainty and credibility. STRAF is proposed for such purpose. This framework is observed to enhance the security of the cloud-based IoT, providing trustworthy service. Still, there is a requiremtn for a policy that details cloud storage mechanism.[12-13]. In this paper, it is proposed to exploit storage complexity in the cloud so that computational complexity can also be improved based on the proposed technique. Section II details the encryption mechanism used in the proposed technique. Section III discusses in detail the coding techniques used in this work. Section IV clearly outlines the proposed system and the methodology proposed. Section V presents the results of the proposed system in contrast to the existing benchmarks. Detailed conclusion and future scope are discussed in Section VI.

2. AES

The proposed technique uses a symmetric block cipher, Advanced Encryption Standard (AES), that is used widely and has replaced DES. 18-bit cipher is used, and the length key varies from 128 to 256 bits. Depending on the key length, the name will be coined as AES-128, AES - 256 etc. N rounds will be formed based on the key length, wherein the first N-1 calculations will transform using shifting the rows, substitution of bytes, mixing the columns, and add round key. Finally, encryption and decryption will be performed [16]. AES is observed to be highly secure in contrast to the existing encryption algorithms.

From figure 2, it can be observed that the encryption and decryption times are very less, which has made way for its use in the cloud. The time complexity must be less enough to have the encryption technique to be used in the cloud. The encrypted size and decrypted size are similar, hence, proving no loss of data. This shows that AES is secured.

Tests were conducted with varying sizes of data viz.; images of different sizes are given as inputs, and statistics are plotted from them.

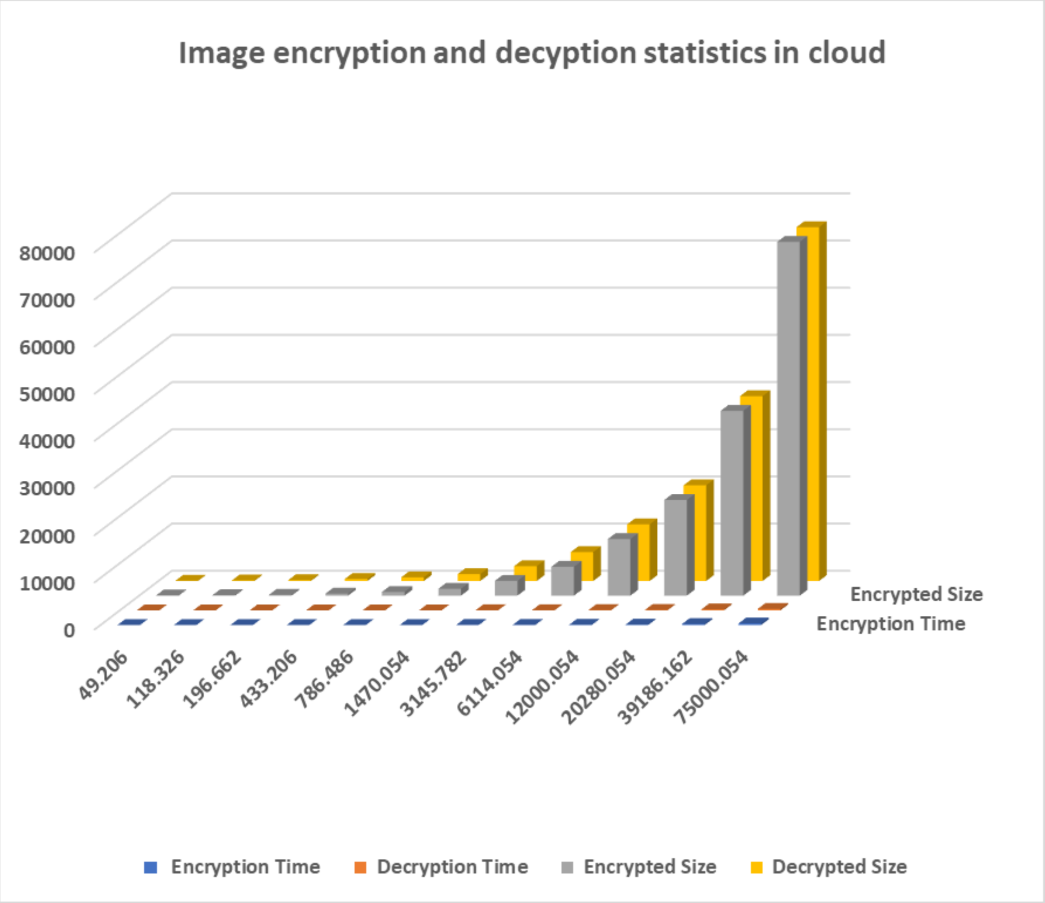

Figure 2. Image encryption and decryption statistics in cloud

3. Coding schemes

Huffman [15] coding delivers a minimum code that is a variable-length character code, generated based on the occurence of the character that is being calculated using probability. This coding technique will provide a tree structure based on the principle discussed. The tree-like structure is formed, which is a heuristic process. One code is assigned to each of the level. The highest frequency character has the minimal length codeword, which results in the least value of the code given to a character. Hence, it can be observed data compression is possible using Huffman coding [17].

Lempel-Ziv-Markov chain algorithm (LZMA) [18] provides a high compression ratio in contrast to other existing techniques using a dictionary-based compression. Figure 3 shows a comparison of LZMA with Huffman for varying sizes of images.

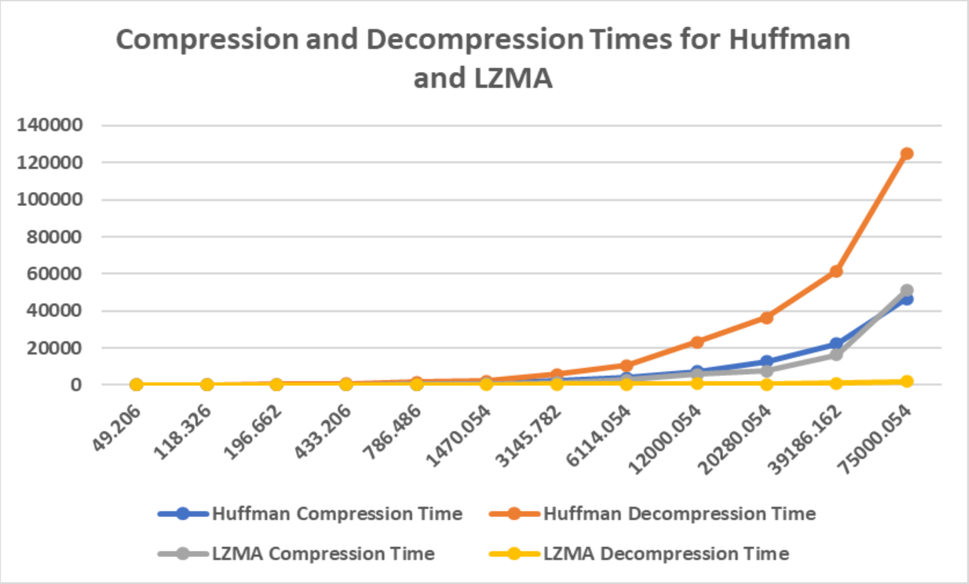

Figure 3. Comparison of compression and decompression times for Huffman and LZMA

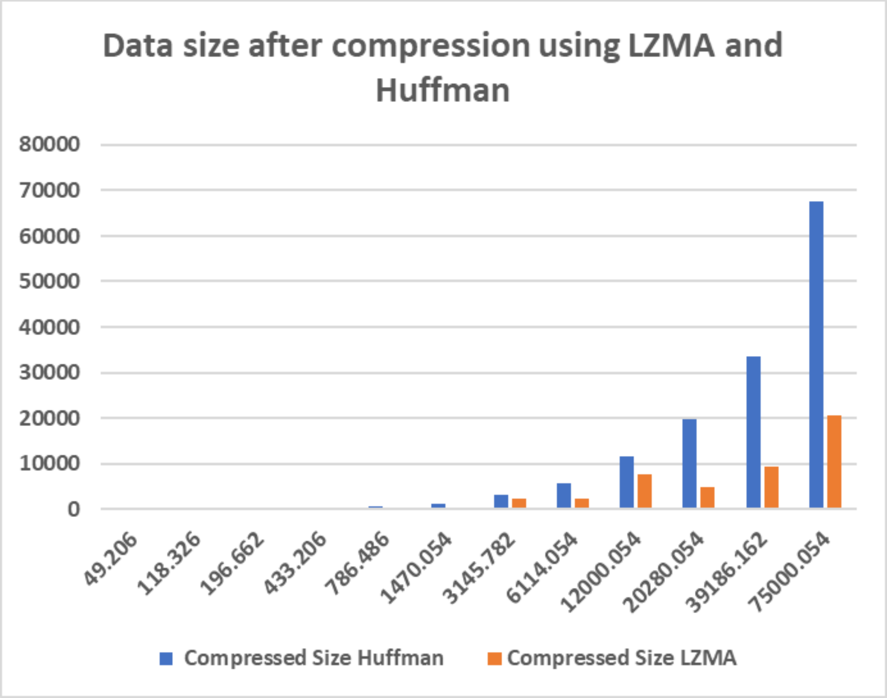

Figure 4. comparison of data size after compression using Huffman and LZMA.

It can be observed that the compression and decompression times are very less in terms of milliseconds in LZMA in contrast to seconds using Huffman coding, which is evident from figure 3. From figure 4, it is evident that data size has got nearly 50% compressed when compared to Huffman. Hence, in this work, LZMA is used for compression rather than Huffman.

4. Proposed System

Figure 5 shows the block diagram of the proposed system. Input is a file rather than an image, which is as shown in the block diagram. In this proposed scheme, the input files are image, text, pdf, and ppt of varied size. The input file is processed using Huffman coding prior to AES, which will give a correct coded version of the file that is given as input. The entire process can be divided into two parts. One on the cloud side and the other on the client-side. The input file is provided as input to the cloud wherein the entire process is performed in the cloud. The compression of the file is performed using Huffman and LZMA. Figure 4 clearly shows that LZMA has a better compression ratio with less time complexity. Hence, in this work, it is proposed to use LZMA and compare the results with the Huffman coding scheme in line with AES.

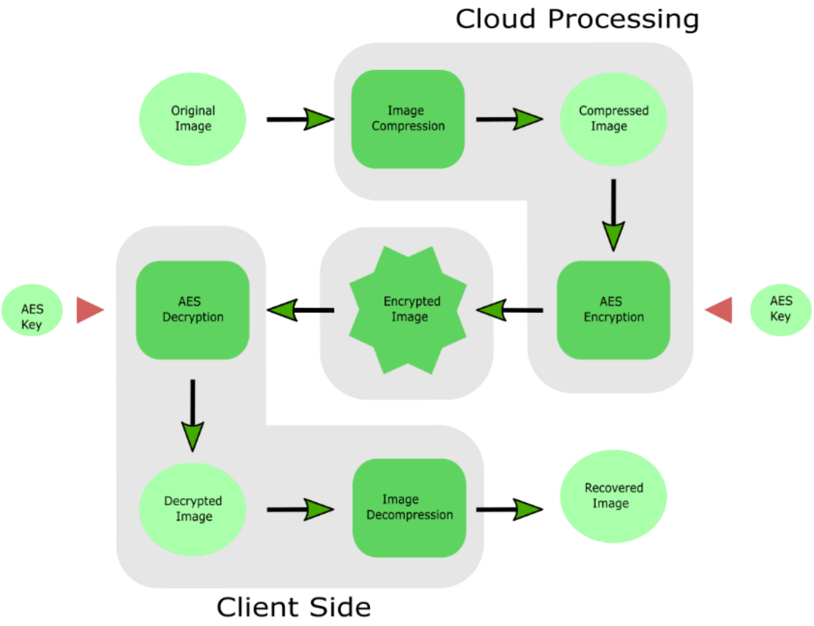

Figure 5. Block diagram of the proposed system

The encrypted file is saved in the cloud, which can be accessed only after decryption at the client-side. It is a well-known fact that AES provides better security in contrast to the existing techniques hence, making the system shielded over the attacks.

Figure 6 represents the utilization of LZMA and Huffman Compression by the proposed framework. Retrieve the input media. All are made into a package and given as input. The input media is queued. If Queue is not empty, then the file the determined for the type of media. It is proposed to have two different procedures for the media. Determining the type of media and processing it. The other way is to process any media with all the techniques. Automated selection based on the type of media is performed in the proposed case. Based on the input type of media, the compression technique is selected and applied. The output is queued and sent to the decryption phase.

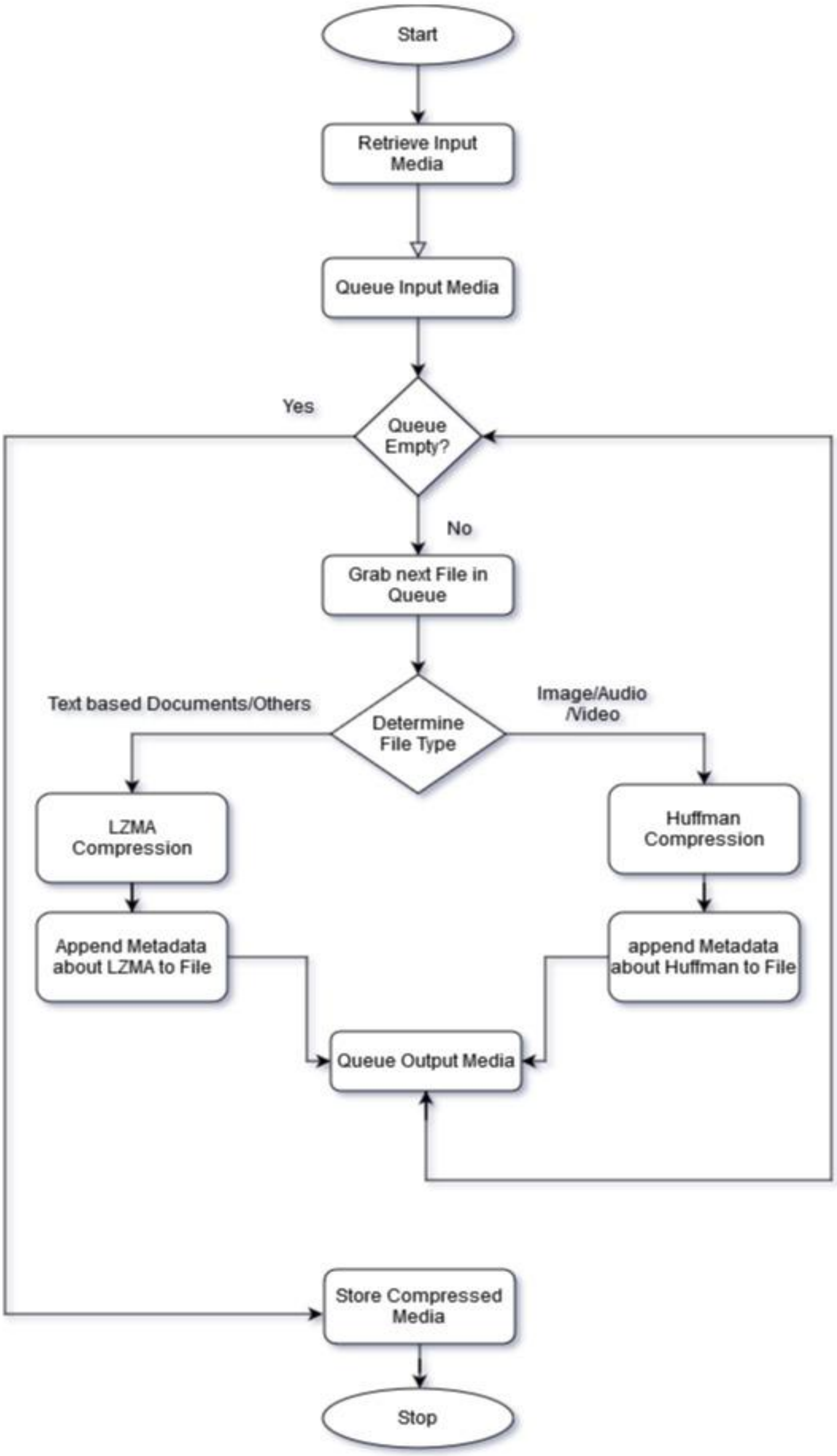

Figure 6. Flowchart of the proposed technique.

Data Encryption Phase

The data obtained from the compression phase is then passed to the Encryption Phase, which uses the AES Algorithm to encrypt the data using randomly generated encryption keys. AES is well known for being a highly secure symmetric algorithm and is employed in even military-grade projects. AES – 256 uses 32-byte cipher blocks to encrypt the data, while there are alternate variants like AES – 128, AES – 512, etc., AES – 256 tends to be faster than the 512-bit variant while still being a very highly secure way to secure files. In the proposed technique, AES – 256 is employed to have a balance between speed and security, considering the complexity. The Encryption Keys are randomly generated for every Individual file to add more diversity among the encryption keys used so that, even when a file gets somehow decrypted, other files remain safe since each of them are encrypted by using unique and different Keys. Figure 7 shows the flowchart of the AES encryption framework implemented.

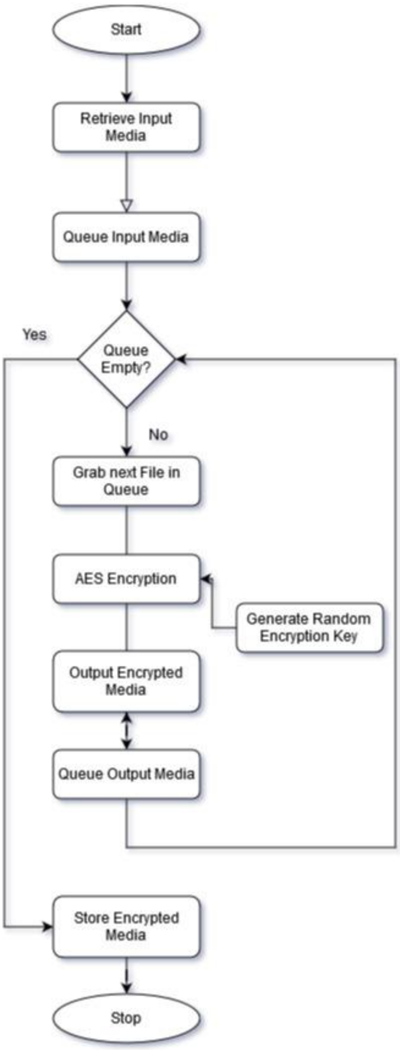

Figure 7. Individual file encryption with random key

Key Encryption Phase

Infrastructure Security concerns arise when the Individual Files' Encryption Keys are stored along with their corresponding files, since the maintainers can still have access to these Encryption Keys and can Decrypt the Data if desired, making the whole Encryption Phase useless. To prevent such cases from happening, all the set of Encryption Keys are again encrypted through RSA Encryption with the Data Owner's Public RSA key. This framework chose specifically the RSA Encryption method because the Public Keys can be easily shared to everyone including the Cloud server to facilitate the Encryption of the data. The Public Keys cannot be used to decrypt the data and only the Owner who has access to the Private Keys can decrypt the data. Figure 8 shows the flowchart of the RSA Encryption Framework.

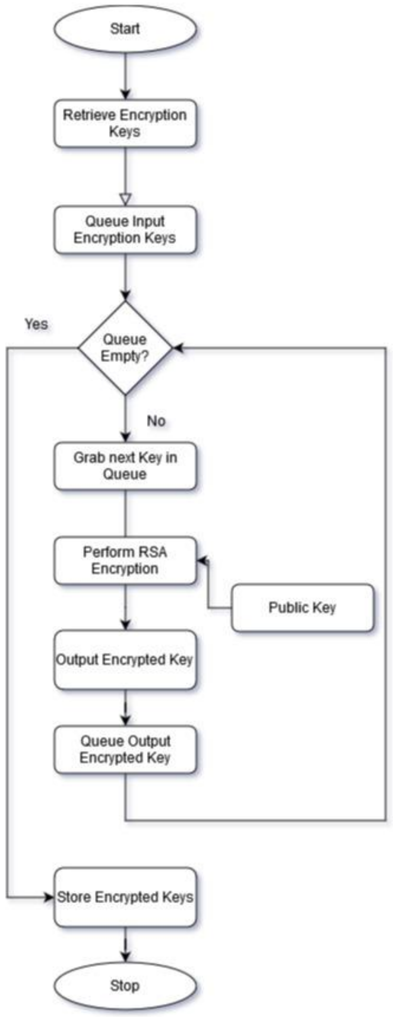

Figure 8. Key encryption using RSA

5. Results and Discussions

Experiments were conducted on different types of media viz., text documents, pdf documents, images, and PowerPoint presentations. The type of media is selected randomly, and the performance metrics are tabulated by varying the size of the media. All the media files are packed and provided for analysis to the system. Figure 9 shows the time taken for encryption after compression for images as media using Huffman and LZMA. It is observed that LZMA outperforms the existing Huffman based system. In contrast to the performance, the computational complexity of the LZMA based system is very less when compared to Huffman.

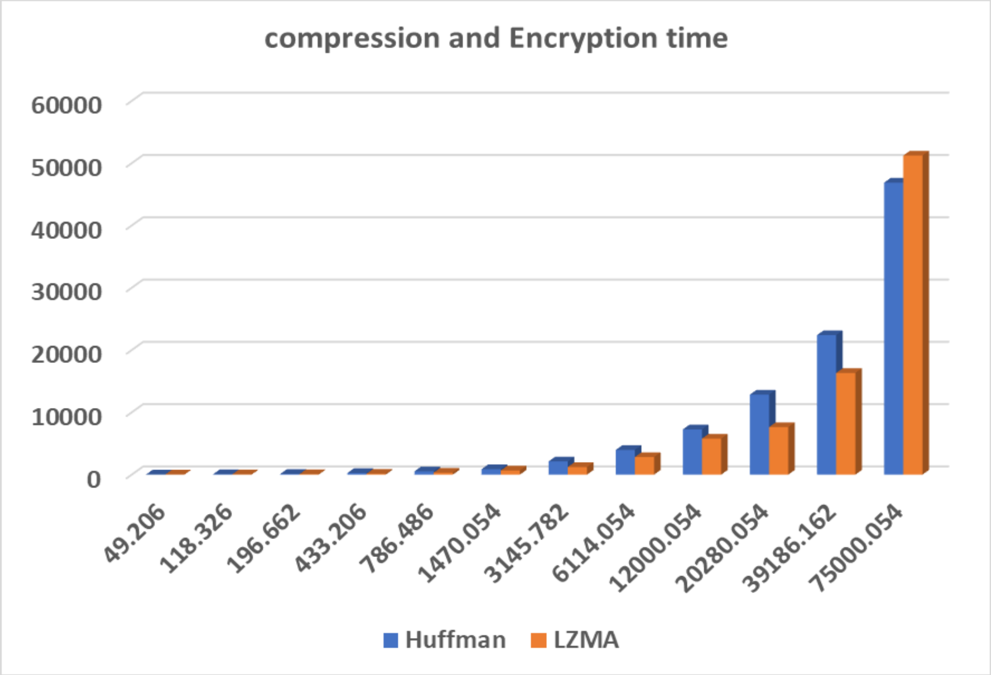

Figure 9. Comparison of Encryption with compression time for Huffman and LZMA for images.

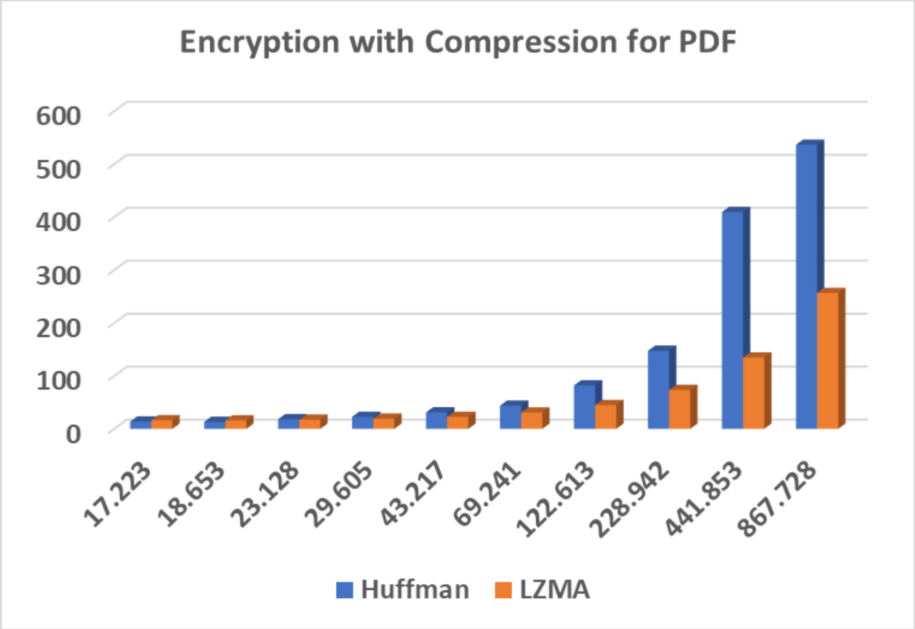

Figure 10. Comparison of Encryption with compression time for Huffman and LZMA for PDF files.

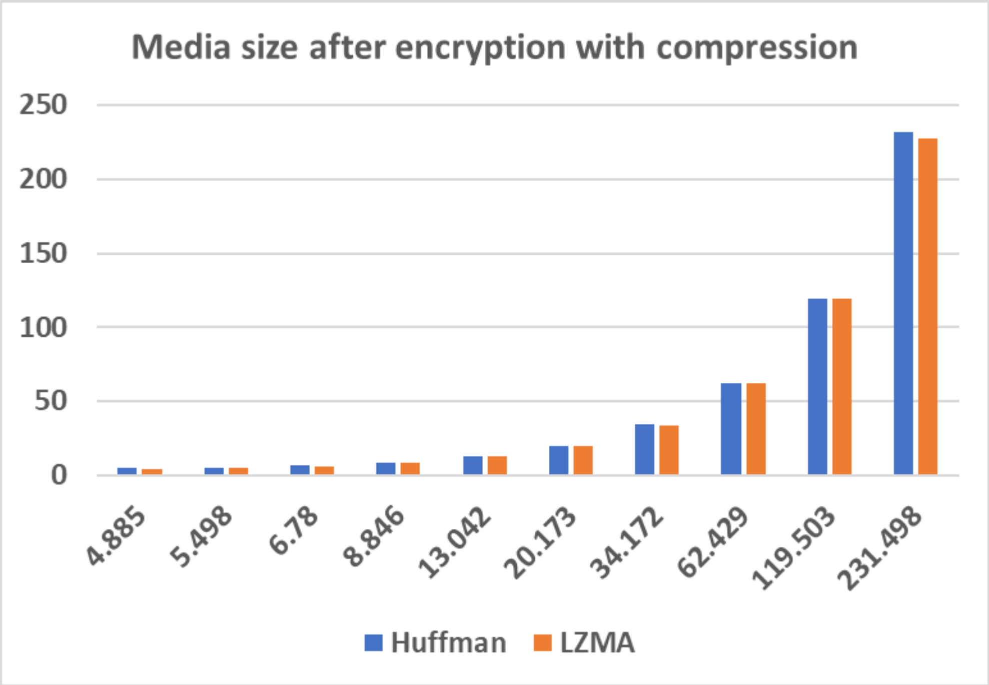

Figure 11. Comparison of media size after encryption and compression.

Further experiments were conducted by providing different types of media files. Figure 10 shows the time complexity analysis of Huffman and LZMA for encryption after compression for PDF files. The time in compression has greatly reduced to milliseconds in the case of the LZMA in contrast to Huffman. It can be thought of to have a storage complexity analysis as the computational complexity which is carried out here is far better in the case of LZMA based on the time comparison graph in figure 10. Hence, the size of the compressed data before passing it to the cloud is measured and tabulated. Figure 11 shows the size of the media after compression and encryption. It can be observed that LZMA has better performance in comparison to Huffman.

Table 1. Comparison of encryption with compression time for LZMA and HUFFMAN

Original Size of media (in KB)	Huffman (in sec)	LZMA (in ms)
1.292	2.058982849	9.666204
2.537	2.477645874	10.02407
5.044	3.376960754	12.2602
10.005	5.352258682	14.22453
20.021	10.6523037	17.27581
39.963	18.36204529	23.88287
79.977	36.95631027	35.66766
159.985	73.09484482	59.7384
320.039	145.7462311	101.8503
639.99	298.6786366	157.3129

For a 231.498 KB file of .doc type is observed to have the same file size after compression and encryption with Huffman, but it is observed to have a 1.98% reduction in size with LZMA, which is shown in figure 11. Table 1 shows that the time for encryption with compression is almost half of that of the Huffman.

6. Conclusion

Security on the cloud is the major research area where most of the researchers tend to concentrate and develop algorithms based on several attacks. AES is a well-known encryption mechanism that has proved to have better security. Hence, in this paper, it is proposed to use AES – 256 for better security. But, the storage complexity in the cloud is still an unsolved problem. Hence, it is exploited, and different compression schemes are applied before encryption. Applying compression to the data or media reduces the load on the network, further, making the network to have less congestion at massive traffic time along with the load. As the cloud may be accessed from a remote place, it is imperative to consider the network bandwidth requirements while processing the data on the cloud. Hence, compression before encryption is proposed. Results show that the compression achieved using LZMA is far better when compared to Huffman based coding. The compression with encryption time of LZMA is observed to be in milliseconds in contrast to the time in seconds for Huffman based coding scheme. Further, to provide additional security to the keys that are used for encryption, RSA is used to encrypt the keys that are used for AES.

References

[1] Akshat Kumar Dixit, Charu Gandhi, "Multilevel security framework for cloud data", Intl. Conf. on Computing and Communication Technologies for Smart Nation (IC3TSN), Gurgaon, India, February 2018, pp. 209-214.

[2] D. Zhe, W. Qinghong, SU Naizheng, Z. Yuhan, "Study on Data Security Policy Based On Cloud Storage", Intl. Conf. on Big Data Security on Cloud, Beijing, China, July 2017, pp. 145-149.

[3] Jahoon Koo, Young-Gab Kim, Sang-Hoon Lee, "Security Requirements for Cloud-based C4I Security Architecture", 2019 Intl. Conf. on Platform Technology and Service, Jeju, Korea (South), March 2019.

[4] C.D. Giulio, C. Kamhoua, Roy H. Campbell, R.Sprabery, K. Kwiat, Masooda N. Bashir, "Cloud Standards in Comparison are New Security Frameworks Improving Cloud Security?", International Conference on Cloud Computing, Honolulu, CA, USA, September 2017, pp. 50-57.

[5] A. Gordon, "The hybrid cloud security professional", IEEE Cloud Computing, vol. 3, no. 1, pp. 82-86, 2016.

[6] Arora A, Khanna A, Rastogi A, Agarwal A, “Cloud Security Ecosystem for Data Security and Privacy”, IEEE, 2017.

[7] AnaghaMarkandey, Prajakta Dhamdhere, Yogesh Gajmal, "Data Access Security in Cloud Computing: A Review", Intl. Conf. on Computing, Power and Communication Technologies (GUCON), India, March 2019, pp. 633-636.

[8] Xiaotong S, "Critical Security Issues in Cloud Computing: A Survey", Intl. Conf. on Big Data Security on Cloud, USA, November 2018, pp. 216-221.

[9] Z. Li, L. Liu, Y. Zhang and B. Liu, "Learning and Predicting Method of Security State of Cloud Platform Based on Improved Hidden Markov Model," 2018 3rd International Conference on Smart City and Systems Engineering (ICSCSE), Xiamen, China, 2018, pp. 600-605.

[10] G. J. Nieves Arreaza, "Methodology for Developing Secure Apps in the Clouds. (MDSAC) for IEEECS Confererences”, Intl. Conf. on Cyber Security and Cloud Computing (CSCloud), France, 2019, pp. 102-106.

[11] X. Li, Q. Wang, X. Lan, X. Chen, N. Zhang and D. Chen, "Enhancing Cloud-Based IoT Security Through Trustworthy Cloud Service: An Integration of Security and Reputation Approach," IEEE Access, Vol. 7, 2019, pp. 9368-9383.

[12] B. Lee, E. K. Dewi and M. F. Wajdi, "Data security in cloud computing using AES under HEROKU cloud," Conf. on Wireless and Optical Communication, Hualien, 2018, pp. 1-5.

[13] L. Li and X. An, "Research on Storage Mechanism of Cloud Security Policy," Intl. Conf. on Virtual Reality and Intelligent Systems (ICVRIS), Changsha, 2018, pp. 130-133.

[14] M. Kang and H. Kwon, "A Study on the Needs for Enhancement of Personal Information Protection in Cloud Computing Security Certification System," Intl. Conf. on Platform Technology and Service, South Korea, 2019, pp. 1-5.

[15] R. Arshad, A. Saleem and D. Khan, "Performance comparison of Huffman Coding and Double Huffman Coding," Intl. Conf. on Innovative Computing Technology, Dublin, 2016, pp. 361-364.

[16] Babitha M.P. and K. R. R. Babu, "Secure cloud storage using AES encryption," Intl. Conf. on Automatic Control and Dynamic Optimization Techniques, Pune, 2016, pp. 859-864.

[17] K. A. Babu and V. S. Kumar, "Implementation of data compression using Huffman coding," 2010 Intl. Conf. on Methods and Models in Computer Science, New Delhi, 2010, pp. 70-75.

[18] X. Zhao and B. Li, "Implementation of the LZMA compression algorithm on FPGA," Intl. Conf. on Electron Devices and Solid-State Circuits, Hsinchu, 2017, pp. 1-2.

Authors

K. Srilakshmi received her Masters Degree from Sri Venkateswara University, Tirupati. She is currently working as Lecturer in the Department of Computer Science, Sri Padmavati Degree College, Tirupati and working towards Ph.D. in the area of Cloud Security, Department of Computer Science, Sri Padmavati Mahila Visvavidyalayam, Tirupati.

Dr. Peyakunta Bhargavi is working as Assistant Professor in the Department of Computer Science, Sri Padmavati Mahila Visvavidyalayam, Tirupati. She Received her Ph.D. from Sri Padmavati Mahila Visvavidyalayam, Tirupati in Data Mining. She has 21 years of teaching and research experience. Currently she is guiding 9 Ph. D. scholars. She is member in IEEE, CSI, ISTE, ACM. More than 30 research papers were published in International and National journals in the areas of Data mining, Soft Computing, Big Data Analytics, Cloud Computing, Bioinformatics and GIS.