

An efficient data compression and storage technique with key management authentication in cloud space

Surekha Pinnapati, Prakasha Shivanna

Department of Computer Science and Engineering, Rama Nagappa Shetty Institute of Technology, affiliated to VTU, Bangalore, India

Article Info

Article history:

Received Jan 13, 2024

Revised Apr 21, 2024

Accepted May 7, 2024

Keywords:

Authentication

Cloud computing

Data compression

De-duplication

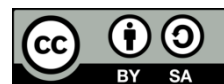
Lossy compression

Regressive probabilistic key encryption

ABSTRACT

Cloud computing is one of the promising technologies that offers cost-effective choices for processing and storing the huge volumes of data. In today's world, data is the most important asset that one can have but it needs to be handled and protected properly. Portability of data can be increased by reducing the size of the data to be stored because of the limited storage space. As a result, data compression has arisen significantly. Data compression is a useful technique for reducing data size and increasing the effectiveness of data transit and storage. Data compression reduces the size of a data file while using lossy or lossless compression. One of the newest techniques for data compression is data duplication, which can reduce the amount of data saved while removing unnecessary data and maintaining an exact copy of the data. This analysis presents an Efficient data compression and storage technique with key management authentication in cloud space. This approach uses Regressive probabilistic key encryption (RPKE) to encrypt the cloud data and Lempel-Ziv-77-Huffman coding (LZ77-HM) is used to compress the huge amounts of cloud data. The Performance of presented approach is evaluated in terms of compression ratio and compression rate.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Surekha Pinnapati

Department of Computer Science and Engineering, Rama Nagappa Shetty Institute of Technology, affiliated to VTU

Banglore, Karnataka, India

Email: sureka.s.p@gmail.com

1. INTRODUCTION

Data compression is one of the fundamental techniques that play a significant role in data transmission as well as storage. The compression means a process which can reduce the number of bits need for data representation. While implementing the data compression technique, one can reduce the storage costs, minimize the usage of network bandwidth and saves huge storage space [1]. The compression process includes two steps process where the encoding technique provides the compressed output while taking message as input and decoding technique reconstructs the original message or appropriate original data approximation from the compressed data [2]. The cloud storage has gain significant importance in information technology (IT) sector. Cloud storage is one of the applications of cloud computing. The cloud storage is one of the most significant components in cloud. The cloud storage depends on distributed file system, cluster application and grid technology and offers online storage to its users [3].

Most of the time, cloud storage offers a very reliable, secure, and affordable storage solution. The most advanced cloud storage application is online file synchronization or backup. Fire, flood, tornadoes, hard drive failure these disasters can occur at any time [4]. All locally stored data may be lost due to such disasters. To secure their data from disasters, users can use remote backup. Keeping imitations of the details

in cloud storage is the alternative [5]. An online backup system is one that uses the Internet and is configured to automatically back up all chosen data. These files are accessible from anywhere since they are kept online. If the local computer or server is lost or destroyed, utilizing online storage services has advantages beyond data security. File sharing across computers and mobile devices is made simple by cloud storage services [6].

Due to its wide acceptance, the cloud faces the security concerns such as authentication, authorization, and integrity. Authentication is the core of security fields whether in network or cloud. Management of authentication and identities are the major challenges in cloud and corporate networks [7]. Encryption and effective key management techniques are employed to protect the cloud data [8].

Data compression means reduce the quantity of data that has to be preserved and transmits the data and sometimes referred to as compaction. Digitally, data compression is accomplished through two methods: Lossless (exact) compression and lossy (inexact) compression are the two types. When data is compressed and then decompressed, the original data is restored because lossless compression maintains data integrity. In the process of compression, redundancy is eliminated, and in the process of decompression, it is added back. Lossless methods are generally suitable only if data loss cannot be compromised [9].

As opposed to lossless compression techniques, lossy compression techniques offer the benefit of a greater compression ratio with some data loss. Since human beings are unable to sense loss or the brain fills in the loss, lossy algorithms are frequently used on static audio, video, and image data. In contrast, lossless techniques are used to compress file data like text files, numeric data, and other types of file data since the programs processing this file data must prevent enduring data problems [10].

Data compression process converts files of types such as audio, video, and text into a database requiring less space. This compression allows retrieval of the original file when needed, thereby preserving the crucial elements of the data. Due to the compressed files smaller size and reduced bandwidth requirements, this procedure makes data storage and transport efficient, simultaneously reducing the utilization of CPU cycles. Zone information protocol (ZIP) files provide a substantial demonstration of file compression. When dealing with a collection of files of various formats that occupy a substantial amount of space, to minimize its size and protect the data from loss during transmission, it is better to convert the database or folder to a zip format. Compression algorithms are essentially responsible for making the compression process possible [11].

The data compression techniques are classified into two types namely: Lossy and Lossless. Both compression techniques use various algorithms for data compression with the aim duplicate the data in Graphic (GIF-Graphics Interchange Format or LZW-Lempel-Ziv Welch) and uses more compact data representation formats. The Lossless compression reduces the number of bits while identifying and eliminating the statistical redundancy [12]. During lossless compression, information can't be lost. In the Lossy compression, number of bits is reduced by removing unnecessary and less important data. The data compression mainly needed because: Uncompressed data occupies huge space that is not good for limited storage space and download speed. As the hardware becomes cheaper and better, the algorithms need for data size reduction helps to evolve the technologies [13].

The Lossless compression techniques include Lempel Ziv Welch (LZW), run length encoding (RLE), zlib and string table compression whereas lossy compression include vector quantization, transform coding and discrete cosine transform (DCT). However, most of the researches were focused on LZW [14]. The LZW technique is one of the most common compression models and is generally used in GIF, AND portable document format (PDF). The LZW is lossless and no data is lost during compression. The implementation of LZW is simple and it has shown high throughput in hardware implementations. It is widely used in Unix file compression and GIF image formats.

In order to facilitate effective storage utilization and shorten transmission times during data transmission across a network, data is encoded into a smaller size than its original size [15]. This process is known as compression. The data compression is made feasible and efficient due to the high level of redundancy present in real-world data. They may acquire a file that is well suited to user needs by applying the right algorithms to the file that has to be compressed [16]. These techniques can be lossless, where all of the original data is returned in its original format, or lossy, where some original data bits may not be recoverable after decompression. Decompression algorithms are the opposite methods that are used to retrieve the original data. Various techniques are available for this purpose. These approaches were created using unique concepts and can be used to handle various data types.

Depending on the type of data, different combinations of redundancy check functions are used to achieve this information [17]. It is challenging to gather, store, analyze, and show data using traditional methods due to the increasing amount and data complexity of cloud services. Different authors have presented approaches using different methods; however, the primary objective is to decrease storage and get eliminate of duplicate data. However, most of the previous data compression techniques were failed to achieve better compression ratio for reducing data duplication [18]. Accurate reconstruction must be performed using lossless compression techniques, which do not lose any information. To secure the data,

save the storage space and improve the performance, an efficient data compression and storage technique with key management authentication in cloud space is presented. This approach not only compress the data but also provides security using RPKE encryption.

The main aim is to minimize the delay in cloud radio network with the help of a virtual control structures. A caching technique based on the preferences and mobility of user is described. A user associated technique is performed with respect to the distances between radio heads and user. Finally, the cloud computing resource allocation issue is formulated as constrained sub-modular function and a heuristic model is presented to determine optimal solutions. The results indicate that this scheme achieved better results in terms of delay [19].

A cloud-based, similarity-aware encrypted deduplication technique with adaptable access control is provided. Techniques for data compression are created according to the kind of data. Text, image, and video data types are the most often utilized types. Every sort of data has a unique feature set and storage format. For each type of data, the deduplication technique utilizes different methods to find and remove duplicate copies. When information is presented in text, image, or video formats, it might be challenging to find and compare it. The compression is carried out through bit-level representation. The minimal amount of data is known as the replication factor [20].

A prediction model inspired by the modes of depth modelling used in high efficiency 3D video coding is described to code depth map. Few prediction residuals are compressed efficiently with various lossy or lossless data compression models. The results indicated that described model eliminated the point cloud data redundant information. This model achieved a 5% of compression ratio i.e. the point cloud is compressed 5% from its original size. This model has shown good performance than other techniques [21].

Data compression and integrity in cloud computing is described. Integrity checking model is employed to store the data remotely and a novel technique is described to compress the data with integrity. Here compression of data deals with the elimination of redundancies for the reduction of storage space and cost over cloud storage. This includes storing and sending of smaller bits. This involves modifying and manipulating the data bits structure in a way to reduce the size. However, still there is huge scope for the optimization of LZW. A new approach i.e. forward-moving approach on most frequently used entries are implemented for avoiding the waiting time in order to determine the codes which make the compression time long [22].

A new approach to scalable data compression that computes the similarity between the divided data chunks. Rather than using simple compressed data units, the compression is applied to the divided data chunks. Certain predictions and restoration functions are designed to restore the original datasets. The results indicated that this compression technique has achieved better improvement in compression efficiency with less accuracy loss [23].

Using high throughput compression (HTC) the data is managed and secured for cloud storage applications. A novel model is described which can reduce the cost and provide better security. Data compression techniques squeeze the data and it needs less disk space to store the data and needs less bandwidth over a data transmission channel. This technique reduced the data size in the cloud and it provides security while providing security using key policy-based encryption and proxy-reencryption [24].

The paper explains the way fully homomorphic encryption and adaptive compression are used to secure and manage data storage on cloud platforms. On the cloud, data is uploaded. Some type of encryption is used to protect data. Fully homomorphic encryption is used to encrypt data. FHE eliminates the need for the owner to either decrypt the data or give the private key to a reliable third party so that it may be computed. After that, the owner will use its private key to decrypt the result and transmit it back to the receiver in encrypted form. One can log in successfully by using the OTP. Another issue that results from implementing fully homomorphic encryption (FHE) is the massive increase in data size, which must be resolved through the use of a lossless compression method [25]. To save the storage space and improve the compression rate, LZW model was designed using systolic model. The results indicate that this system provided good compression rates and space-saving percentages and improved the usage. The compression ratio is improved 23% than earlier models [26].

The organization of the work is as follows: The literature survey is discussed after the introduction. The section 2 presents methodology of an efficient data compression and storage technique with key management authentication in cloud space. The section 3 provides a results analysis. Section 4 presents the conclusion.

2. METHOD

An effective data compression and storage technique with key management authentication in cloud space is described in this analysis. The Figure 1 shows the architecture of presented model. Data that has not

been processed, coded, formatted, or otherwise examined for informational value is known as raw data, commonly known as source data or raw data. Although it serves as a valuable resource, due to its visual confusion and lack of integration, raw data can be difficult to understand or act upon. Raw data, sometimes referred to as source data, atomic data, or original data, is data that has not been processed for application. In certain instances, a distinction is made between data and information, highlighting the fact that information is the product of data processing. In information theory, source coding, data compression, or bit-rate reduction refer to the process of encoding information while utilizing fewer bits than the original representation.

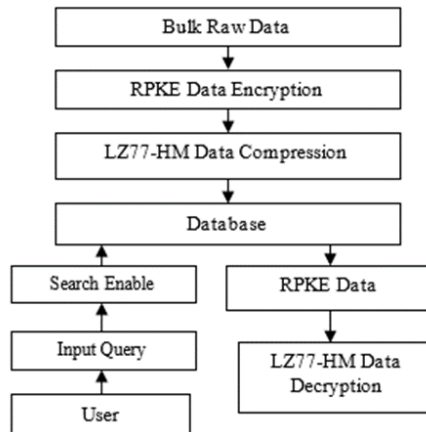


Figure 1. The architecture of presented model

Data compression is the method of reducing a data file's size and is frequently used in place of data compression. This procedure, known as source coding with regard to data transmission, encrypts data at the data source before it is stored or sent. It is important to distinguish source coding from other types of coding, such as line coding and channel coding, which are used to map data onto signals and are also used for error detection and correction.

Compression is useful since it uses fewer resources for data storage and transmission. However, processing resources are needed for the compression and decompression procedures. Space and temporal complexity are traded off during data compression. For instance, a costly piece of hardware could be needed to decompress a video compression technique quickly enough for viewing. In certain circumstances, properly decoding the video before viewing it may be difficult or need more storage. Creating data compression schemes involves achieving a balance between different factors, such as the extent of compression achieved, the level of introduced distortion (in the context of lossy data compression), and the tools and computing power required for compression and decompression.

In the field of public key cryptography, regressive probabilistic key encryption (RPKE) is a type of public-key encryption technique in which a probabilistic Turing machine is run with a different ciphertext for every message that has the same public key. In the context of encryption, randomness is employed in "randomised encryption." Consequently, different ciphertexts are typically produced when the same communication is encrypted more than once. Methods of public key encryption are often associated with the concept of "probabilistic encryption". Furthermore, this property is also present in some symmetric key encryption methods, such as stream ciphers like freestyle that are inherently unpredictable and block ciphers utilized in chaining mode. An encryption method should be probabilistic to achieve semantic security, which entails hiding even a portion of the plaintext's content.

When using public key cryptography, probabilistic encryption is especially important. Assuming that when an adversary detects a ciphertext, they can assume that the plaintext uses a cross, the letters "YES" or "NO," or both. An adversary might easily try encrypting each of his guesses using the recipient's public key in the case of a non-probabilistic encryption technique, and then compare the results to the intended ciphertext. Public key encryption methods should have a random component that enables every plaintext to be mapped to any number of alternative ciphertexts in order to prevent such attacks. A common approach to transforming an encryption scheme from deterministic to probabilistic involves adding a random string to the plaintext before encrypting it using the algorithm. During decryption, the random padding is ignored, and the original plaintext is obtained by applying the deterministic algorithm. The (1) defines the encryption as (1).

$$Enc(x) = f((r), x \otimes b(r)) \quad (1)$$

Here, 'x' denotes single-bit plaintext, 'f' trapdoor permutation (a deterministic encryption method), 'b' the hard-core predicate of f, and 'r' a random string in (1). The amount of text or data used for compression and decompression determines the file size. A compression method, such as lossless comp data, processes the data using Lempel-Ziv-77-Huffman coding (LZ77-HM) once the file size has been determined. LZ77 lossless compression, which makes it easier and more efficient to compress data, is used in this case. Subsequently, the user accesses the compressed data file and verifies the correctness of the compressed data. The compressed file gets deleted when the user closes the programme and the data is accurate. The sliding window will move to the next point and the search will carry on until a match is discovered, albeit, if the data is inaccurate. The compressed file may be decompressed to reveal the original data. Hence, Huffman coding is a technique for data compression using variable-length codes. of this approach, a collection of code words of varying lengths is used, and the code words with the smallest average length are chosen for the data based on the frequently they appear.

Database compression encompasses a collection of techniques used to reorganize database content, which leads to reduced physical storage space usage and enhanced performance speeds. Compression is accomplished through two primary methods: Lossless compression, which uses the compressed data to completely rebuild the original data, and Lossy compression, which reduces data size by actively compromising quality. Reducing the organization's overall database storage footprint is the main objective of database compression. It includes a range of data types, including relational (tables), unstructured (files), indexed, network-transferred, and backup data. Compression has the ability to reduce storage usage from 60% to as low as 20% of the original space, depending on the data cardinality. Tables with sparse population, containing many zeros or spaces in the data, compress much better.

The process of regressive probabilistic key encryption (RPKE) data decryption involves converting the encrypted code or data back to a form that is easily understandable and readable by both humans and machines. This process is commonly referred to as decoding encrypted data and occurs at the receiver's end. The decryption of the message can be performed using either the private key or the secret key. Therefore, decryption is a technique employed in cybersecurity, making it difficult for hackers to intercept and read unauthorized information. Despite encryption being utilized to secure the data, recipients require access to the appropriate decoding or decryption tool to access the original information. The (2) defines the decryption as (2).

$$Dec(y, z) = b(f^{-1}(y)) \otimes z \quad (2)$$

Where the variables x, f, b, and r represent the hard-core predicate of f, deterministic encryption algorithm, and random string, respectively. During the requisition phase, authorized users generate a unique key. Each user has a unique identity assigned for tracing each attribute. The identities and user data are hidden from the users. By using this technique, no information about the qualities matching or mismatching can be obtained from the ciphertexts. The characteristics are divided into two categories: hidden identity attributes (HI) and hidden normal attributes (HN). Enable text compression on the server(s) that supplied these responses in order to pass this audit. Text compression is enabled on the server that provided these responses to pass this audit.

The integration of several services in responsible for data storage and query execution is made possible by cloud technology. Ensuring integrity in the join query results can be achieved through several methods. The user submits a query containing a join operation without possessing any information about the data's storage location or the servers responsible for executing the connected queries. The query execution engine receives the user's request for a join operation and then sends sub-queries to the servers that store the data. The execution server performs the join operation between the two sets of sub-query results after receiving the results of the sub-query, and then returns to the user the final result. The Hybrid LZ77-HM is used to compress the data and eliminate the duplication. This hybrid compression technique LZ77-HM has shown better performance for data compression in cloud environment.

3. RESULT ANALYSIS

In this analysis, an efficient data compression and storage technique with key management authentication in cloud space is presented. The performance of presented approach is compared with LZW in terms of compression ratio and compression rate. In addition, the encryption time of presented RPKE is also evaluated. The Table 1 shows the performance evaluation. Compared to LZW method, presented LZ77-HM has obtained better results. The Figure 2 shows the compression ratio comparison.

Table 1. Performance evaluation

Model/metrics	File size (bytes)	Compression ratio (Mbps)	Compression rate (%)
LZW	1024	26 Mbps	67.8%
LZ77-HM	1024	34 Mbps	74.6%

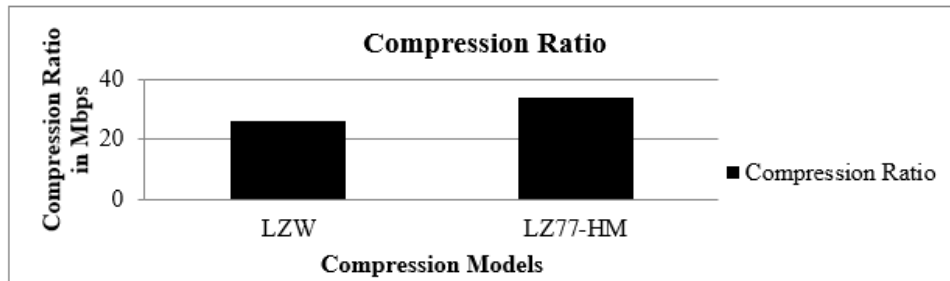


Figure 2. Compression ratio comparison

In Figure 2, the x-axis indicates compression models whereas y-axis indicates compression ratio in terms of megabits per second (Mbps). Compared to LZW model, presented Hybrid LZ77-HM model has better compression ratio. The Figure 3 shows the compression rate comparison.

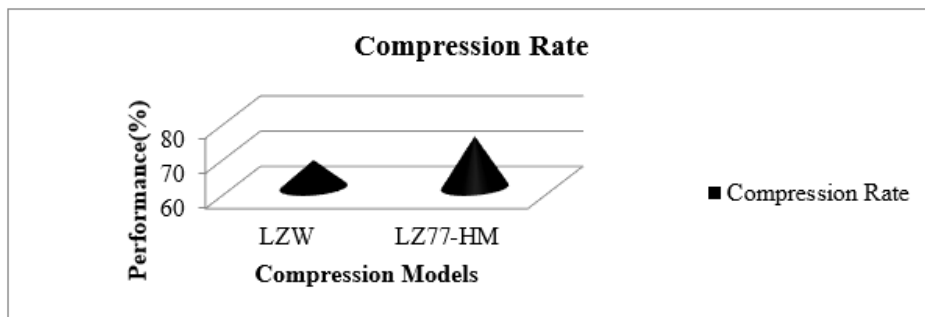


Figure 3. Compression rate comparison

The LZ77-HM model has achieved better compression rate than LZW model. The Figure 4 shows the encryption time comparison. In Figure 4, the x-axis indicates the data file size and y-axis indicates encryption time in milli seconds (ms). Compared to Rivest Shamir Adleman (RSA), presented RPKE encryption technique requires less time for encryption. Hence, presented model has effectively encrypted, compressed and decrypted the huge volumes of cloud storage data. This model has achieved better security and data compression in cloud space.

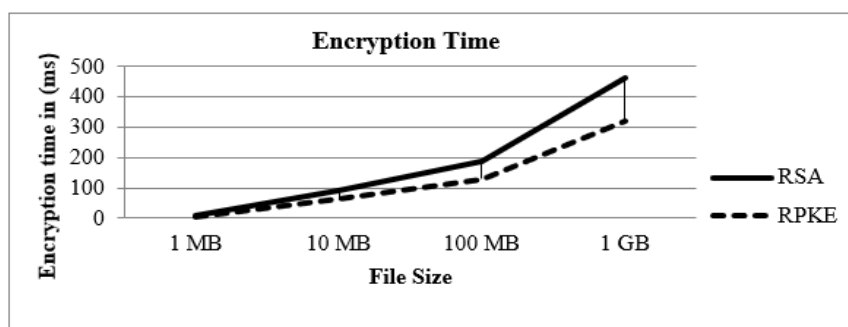


Figure 4. Encryption time comparative graph

4. CONCLUSION

In today's world, the most valuable thing somebody can have is data, yet handling it properly also presents a difficulty. Data compression has become extremely important in order to promote data portability, which is made possible by lowering the amount of data that must be saved due to storage space limitations. To solve these issues, an efficient data compression and storage technique with key management authentication in cloud space is described in this analysis. Regressive probabilistic key encryption is used to encrypt the data. Through utilizing hybrid LZ77-HM compression techniques to reduce the size of the data file, the presented data compression model decreases the amount of data that needs to be maintained and provided. For varying data sizes, the effectiveness of the proposed hybrid compression technique is determined in terms of the compression ratio and compression rate. Compared to other compression-based algorithms, presented LZ77-HM algorithm has better compression performance. In future, data compression technique with hybrid encryption and compression algorithms will be implemented to effectively secure the data, reduce storage space, as well as elimination of data duplication in cloud space.




REFERENCES

- [1] S. G. Rathod, A. K. Bhise, N. S. Shaikh, Y. B. Dongare, and S. R. Kothavle, "A secure storage management and auditing scheme for cloud storage," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 6, pp. 1-8, July, 2023, doi: 10.17762/ijritcc.v11i6.6765.
- [2] B. Venkatesan and S. Chitra, "Data De-duplication process and authentication using erce with poisson filter in cloud data storage," *Intelligent Automation & Soft Computing*, vol. 34, no. 3, pp.1603-1615, May, 2022, doi:10.32604/iasc.2022.026049.
- [3] J. G. Jeslin and P. M. Kumar, "Decentralized and privacy sensitive data de-duplication framework for convenient big data management in cloud backup systems," *Symmetry*, vol. 14, no. 7, pp. 1-20, July, 2022, doi: 10.3390/sym14071392.
- [4] F. S. Ali, H. N. Saad, F. H. Sarhan, and B. Naaem, "Enhance manet usability for encrypted data retrieval from cloud computing," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 18, no. 1, April 2020, pp. 64-74, doi: 10.11591/ijeecs.v18.i1.pp64-74.
- [5] U. Narayanan, V. Paul, and S. Joseph, "A novel system architecture for secure authentication and data sharing in cloud enabled big data environment," *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 6, pp. 3121–3135, 2022, doi:10.1016/j.jksuci.2020.05.005.
- [6] R. Luo, H. Jin, Q. He, S. Wu and X. Xia, "Enabling balanced data deduplication in mobile edge computing," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 5, pp. 1420-1431, May 2023, doi: 10.1109/TPDS.2023.3247061.
- [7] M. Hasson, Ali A. Yassin, A. J. Yassin, A. M. Rashid, A. A. Yaseen, and H. Alasadi, "Password authentication scheme based on smart card and QR Code," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, pp. 140-149, July 2021, doi: 10.11591/ijeecs.v23.i1.pp140-149.
- [8] A. I. Abdulsada, D. G. Honi, S. Al-Darraj, "Efficient multi-keyword similarity search over encrypted cloud documents," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, pp. 510-518, July 2021, doi:10.11591/ijeecs.v23.i1.pp510-518
- [9] P. Singh, N. Agarwal, and B. Raman, "Secure data deduplication using secret sharing schemes over cloud," *Future Generation Computer Systems*, vol. 88, pp. 156–167, 2018, doi: 10.1016/j.future.2018.04.097.
- [10] C. Costa, P. K. Chrysanthis, M. Costa, E. Stavarakis and N. Nicolaou, "Towards a signature-based compression technique for big data storage," *2023 IEEE 39th International Conference on Data Engineering Workshops (ICDEW)*, Anaheim, CA, USA, 2023, pp. 100-104, doi: 10.1109/ICDEW58674.2023.00022.
- [11] W. Xia, C. Wei, Z. Li, X. Wang and X. Zou, "NetSync: A network adaptive and deduplication-inspired delta synchronization approach for cloud storage services," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 10, pp. 2554-2570, Oct. 2022, doi: 10.1109/TPDS.2022.3145025.
- [12] S. Sheik, T. R. Komati, "A way to secure the data in cloud data storage by using cloud data compression mechanism," *Asian Journal of Information Technology*, vol.7, no. 2, pp.236-241, 2019, doi:10.36478/ajit. 2020.236.241.
- [13] K. Rani and R. K. Sagar, "Enhanced data storage security in cloud environment using encryption, compression and splitting technique," *2017 2nd International Conference on Telecommunication and Networks (TEL-NET)*, 2017, pp. 1-5, doi: 10.1109/TEL-NET.2017.8343557.
- [14] B. Koc, Z. Amavut and H. Koçak, "Concurrent encryption and lossless compression using inversion ranks," *2022 Data Compression Conference (DCC)*, Snowbird, UT, USA, 2022, pp. 459-459, doi: 10.1109/DCC52660.2022.00070.
- [15] X. Sheng, L. Li, D. Liu, Z. Xiong, Z. Li and F. Wu, "Deep-PCAC: An end-to-end deep lossy compression framework for point cloud attributes," in *IEEE Transactions on Multimedia*, vol. 24, pp. 2617-2632, 2022, doi: 10.1109/TMM.2021.3086711.
- [16] M. U. Amaizu, M. K. Ali, A. Anjum, L. Liu, A. Liotta and O. Rana, "Edge-enhanced qos aware compression learning for sustainable data stream analytics," in *IEEE Transactions on Sustainable Computing*, vol. 8, no. 3, pp. 448-464, 1 July-Sept. 2023, doi: 10.1109/TSUSC.2023.3252039.
- [17] Z. Miao, W. Li and X. Pan, "Multivariate time series collaborative compression for monitoring systems in securing cloud-based digital twin," *Journal of Cloud Computing*, vol. 13, no. 16, pp. 1-15, 2024, doi:10.1186/s13677-023-00579-4.
- [18] X. Chai, H.Wu, Z. Gan, Y. Zhang, Y. Chen, and K.W. Nixon, "An efficient visually meaningful image compression and encryption scheme based on compressive sensing and dynamic LSB embedding," *Optics and Lasers in Engineering*, vol. 124, Jan. 2020, doi: 10.1016/j.optlaseng.2019.105837.
- [19] J. Zhang, Y. Xu, W. Xia, Y. Xu, S. Cai and H. Zhu, "Edge caching and resource allocation scheme of downlink cloud radio access networks with fronthaul compression," in *IEEE Access*, vol. 7, pp. 118669-118678, 2019, doi: 10.1109/ACCESS.2019.2936666.
- [20] Y. Zhou *et al.*, "A similarity-aware encrypted deduplication scheme with flexible access control in the cloud," *Future Generation Computer Systems*, vol. 84, pp. 177–189, 2018, doi: 10.1016/j.future.2017.10.014.
- [21] X. Sun, H. Ma, Y. Sun and M. Liu, "A novel point cloud compression algorithm based on clustering," in *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2132-2139, April 2019, doi: 10.1109/LRA.2019.2900747.
- [22] A. Choudhury, B. Roy and S. K. Misra, "Data integrity and compression in cloud computing," *International Journal of Computer Applications*, vol. 168, no.13, pp. 14-19, June 2017, doi: 10.5120/ijca2017914553.




- [23] C. Yang and J. Chen, "A scalable data chunk similarity based compression approach for efficient big sensing data processing on cloud," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1144-1157, 1 June 2017, doi: 10.1109/TKDE.2016.2531684.
- [24] Sandeep, B. Deepthi, and M. S. Muneshwara, "Securing and managing data for cloud storage applications using high throughput compression (HTC)," *International Journal of Advanced Scientific and Technical Reserch*, vol. 3, no. 5, 2015.
- [25] B. S. Bumrah and G. Kaur, "Security and storage management of data on cloud using fully homomorphic encryption and adaptive compression approach," *International Journal of Science and Research (IJSR)*, vol. 5, no. 6, June 2016, doi: 10.21275/v5i6.NOV164164.
- [26] G. Mohey, A. Zekry, and H. Zakaria, "FPGA implementation of Lempel-Ziv data compression," *International Journal of Reconfigurable and Embedded Systems (IJRES)*, vol. 10, no. 2, pp. 99-108, July 2021, doi: 10.11591/ijres.v10.i2.pp99-108.

BIOGRAPHIES OF AUTHORS



Mrs. Surekha Pinnapati    is a Research Scholar in the Computer Science and Engineering department at RNS Institute of Technology Bangalore, affiliated with Visvesvaraya Technological University, Belgavi. She completed her bachelor's degree in Information Science and Engineering at PDACE Gulbarga Autonomous Institute Affiliated to Visvesvaraya Technological University, Belgavi, Karnataka, India and her master's degree in Computer Science and Engineering from BIET Davangere Affiliated to Visvesvaraya Technological University, Belgavi. Mrs. Surekha Pinnapati possesses a strong academic and research background, particularly in the areas of bigdata, database management systems, operating systems Unix systems programming, and data mining. She has made significant contributions to her field. She can be contacted at email: sureka.s.p@gmail.com.



Dr. Prakasha Shivanna    holds the esteemed position of Associate Professor in the Department of Information Science and Engineering at RNS Institute of Technology, affiliated with Visvesvaraya Technological University Belgavi. He completed his Ph.D. in the field of data mining and holds a master's degree in Computer Science and Engineering. Dr. Prakasha possesses a strong academic and research background, particularly in the areas of machine learning, big data, database management systems, deep learning, and data mining. He has made significant contributions to his field. He can be contacted at email: prakashasphd@gmail.com.