iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** V **Month of publication:** May 2025

DOI: https://doi.org/10.22214/ijraset.2025.70183

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 13 Issue V May 2025- Available at www.ijraset.com

Enhancing Secure Cloud Storage: A Four-Tier Architecture with ChaCha20 Encryption, Blake3 Hashing, and Dynamic Chunk Allocation in Multi-Cloud Environment

Gaurav Thakur[1], Anita Ganpati[2]
Department of Computer Science, Himachal Pradesh University

Abstract: ***With the growing reliance on cloud computing for storing and managing data, providing security and performance for cloud storage systems has become very important and multicloud solutions offers both. Current cloud architectures suffer from scalability, security, and efficiency issues in distributed cloud systems. In this paper, we introduce a ChaCha20-encryption, a fast stream cipher and Blake3, an adaptive cryptographic hash function, and dynamic chunk allocation to an existing architecture that progressively improves data confidentiality, integrity, and redundancy. The experiment focuses on the system's performance relative to Four Tier Secure Cloud Storage Architecture, which utilizes AES, MD5 hashing, and static chunk allocation strategies for data storage. To ensure a balance between performance while offering the security this architecture implements ChaCha20 which is fast and resistant to side-channel attack – making it ideal for modern multi cloud deployments. Experiments show that this new architecture is faster, more lightweight and adaptable than existing model.***
Keywords: Cloud Computing, Multi Cloud, FTSCSA, Chacha20, Blake3, Dynamic Chunk Allocation.

I. INTRODUCTION

A new form of consuming and delivering computing services such as servers, storage, and databases over the Internet is known as cloud computing [1].The Shift in mindset obviates the need for physical hardware ownership and operation, allowing both individual users and businesses to consume computing resources on a flexible pay-per-use basis [2]. Cloud computing is the technology that transformed IT infrastructure simply by offering access to resources on demand [3]. This leaves enterprises free to focus on innovation rather than maintaining complex systems or data centers. The multicloud model represents a strategic approach where organizations utilize services from multiple cloud providers to meet their diverse and dynamic needs [4].Based on All cloud infrastructure report 2020, it was been expected that 85% of the enterprises and organizations wish to move their traditional infrastructure and workloads on Cloud Computing by 2020.Rather than relying on a single cloud provider for all their infrastructure and application requirements, businesses adopt a multi-cloud strategy to leverage the unique strengths and features offered by different providers. AES encryption is utilized in four-tier secure cloud storage architecture, a multicloud approach [5]. Cryptographer Daniel J. Bernstein created the more recent CHACHA20 stream cipher in 2008 to replace the antiquated Salsa20 cipher [6]. Because of its well-known speed, ease of use, and security, CHACHA20 is a great option for applications that need high-performance encryption. Stream ciphers, like CHACHA20, work by generating a pseudorandom stream of bits (the key stream), which is XORed with the plain text to produce the cipher-text. This is in contrast to block ciphers, which encrypt fixed-size blocks of data [7].

Because of CHACHA20's effectiveness across a range of hardware plat-forms, including low-power devices, it has gained popularity in current cryptography. Because of its ease of use and defense against side-channel assaults, it is a popular option for secure communication protocols including SSH, VPNs, and TLS (Transport Layer Security) [8].

Blake3 a fast and secure cryptographic hashing function which outperform traditional algorithms like SHA-256, MD5 and SHA-512, especially in multi-core environment provides strong collision resistance (1/2^256 probability) and improved processing speed while using less memory, making it suitable for cloud storage systems.

II. LITERATURE REVIEW

H.S. Baqtian et al. compared MD5, SHA-1, and SHA-256 for verifying the integrity of digital Holy Quran texts. Results indicate SHA-256 provides superior resistance to tampering, addressing vulnerabilities in older algorithms. The research underscores the necessity of robust cryptographic hash functions to ensure authenticity in digitally preserved religious scriptures [19].

B. Kezia Rani et al. examined inter-cloud computing, discussing cloud types, connection topologies, and emerging challenges like interoperability, security, SLA management, and standardization. Emphasizing scalable, collaborative cloud infrastructures, the paper highlights the need for efficient frameworks to address complexities in multi-cloud ecosystems and support demands in big data and global computing [20].

Jarosław Sugier analysed power consumption in pipelined FPGA implementations of the BLAKE3 hash function. The study compares pipelining strategies, revealing trade-offs between energy efficiency and throughput. Targeting low-power embedded systems, the work advances understanding of optimizing BLAKE3 for secure, high-performance cryptographic processing in real-world hardware environments [21].

Noura Aleisa compared 3DES and AES encryption standards, focusing on structural differences, performance, and security. The study highlights AES's superior speed, scalability, and resistance to attacks, explaining its dominance over the outdated 3DES. This work reinforces AES's role as the preferred choice for modern, secure data transmission in digital systems [22].

Sunil Kumar et al. investigated Third-Party Auditor (TPA) models to enhance privacy and security in cloud systems. Utilizing homomorphic encryption and privacy-preserving protocols, TPAs verify data integrity without accessing content. The study emphasizes efficient, trustworthy auditing frameworks crucial for maintaining confidentiality, accountability, and user trust in modern cloud infrastructures [23].

Deepak Puthal et al. presented a comprehensive overview of cloud computing, highlighting features like scalability and on-demand service, alongside challenges such as security, privacy, and vendor lock-in. The paper underscores the need for innovative solutions to address issues in multi-tenant environments, offering valuable insights into cloud infrastructure complexities and evolution [24].

Nicky Mouha presented a detailed review of the Advanced Encryption Standard (AES), analysing its structure, performance, and resistance to cryptographic attacks. The paper emphasizes AES's efficiency and robustness compared to older ciphers like DES and 3DES, reinforcing its status as the dominant encryption standard in modern secure communication systems [5].

Z. Najm et al. examined the side-channel vulnerabilities of AES and ChaCha20 on microcontrollers, highlighting AES's susceptibility to cache-based timing attacks due to its S-boxes. In contrast, ChaCha20's ARX structure makes it more resistant to such attacks. The paper emphasizes ChaCha20's suitability for IoT, multicloud, and edge computing environments where side-channel resistance is critical [18].

J. P. Degabriele et al. explored the ChaCha20-Poly1305 AEAD scheme, noting its advantages over AES-GCM in cloud environments. The paper evaluates its resistance to differential cryptanalysis, timing attacks, and key-recovery vulnerabilities, highlighting its strong security and performance. It stresses the importance of proper nonce management to avoid catastrophic security failures in multi-cloud architectures [25].

R. K. Muhammed et al. provided a comparative analysis of five image encryption algorithms AES, Blowfish, Twofish, Salsa20, and ChaCha20 evaluating their efficiency, security, and performance. The study assesses key management, encryption speed, and attack resistance, highlighting ChaCha20's suitability for modern applications and its superior performance in software environments over traditional block ciphers [8].

III. RESEARCH REVIEW

This study adopts a comparative experimental research methodology to evaluate an enhanced Four-Tier Secure Cloud Storage Architecture (FTSCSA). The proposed architecture incorporates ChaCha20 encryption, Blake3 hashing, and Dynamic Chunk Allocation (DCA), aiming to improve the overall security, efficiency, and scalability of multi-cloud storage systems. The effectiveness of the proposed model is assessed against an existing FTSCSA configuration that utilizes AES encryption, MD5 hashing and Circular Shift Chunk Allocation (CSCA).

The research design involves a structured experimental framework, where both models are tested under controlled conditions using a diverse set of datasets. The implementation is carried out using Python 3.10 on Google Colab, leveraging appropriate cryptographic and data storage libraries. The architecture is built on four logical layers: (1) the Encryption Layer, which secures data using ChaCha20 or AES; (2) the Encoding Layer, which dynamically splits data based on type and size; (3) the Hashing Layer, employing Blake3 or MD5 for data integrity; and (4) the Allocation Layer, which distributes data chunks dynamically or via circular shift methods across multiple cloud platforms.

The experimental setup includes a client machine with an Intel processor (4 cores, 8 logical threads), 8 GB RAM, and a 2.40 GHz CPU, with all executions taking place in a cloud-based Python environment. Three datasets of varying sizes were used: a small dataset (~10 MB) for quick encryption and decryption testing, a medium dataset (~100 MB) for evaluating hashing and allocation efficiency, and a large dataset (~500 MB) for benchmarking performance at scale.

Several evaluation metrics were defined for comparative analysis: encryption time, decryption time, hashing time (during both storage and retrieval), chunk allocation time, total processing time, and a security analysis focusing on collision resistance and encryption strength. The experimental procedure involved pre-processing datasets into chunks, applying encryption using both ChaCha20 and AES, hashing encrypted chunks with Blake3 and MD5, and finally allocating them using either Dynamic or Circular Shift techniques. Statistical tools were used to analyse and interpret the results.

IV. CLOUD COMPUTING

Cloud computing is on-demand access to a shared pool of configurable computing resources (e.g., servers, storage, applications) over the internet This means the user can quickly scale the resources he needs to rent to meet demand with little management effort or interaction with a service provider, thus providing a more flexible and cost-effective approach [9]. Cloud services are generally offered in three models such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), and each model has its own advantages in terms of data storage, application development and software access [10].

A. *Characteristics*

1) On-Demand Self-Service: Without having to contact your cloud provider, users are able to provision re-sources over cloud computing—including networking, processing power, and storage. It is fast and flexible, as you enable users to quickly respond to business needs without needing the IT team to manually set up the resources beforehand [3].
2) Broad Network Access: Based on internet delivery, or in some cases over private networks, it can be accessed from everywhere through any connected device e.g., any laptop, tablet, smartphone etc. this feature promotes mobility and flexibility [11].
3) *Resource Pooling:* Resource pooling allows cloud service providers to provide a different set of services to multiple clients as per the needs by sharing resources [12].
4) Service Measured (Pay-Per-Use): In cloud platforms, resource usage is continuously recorded and billed to the customer either by the hour (e.g., compute instances) or by the gigabyte (e.g., storage). Usage metrics are visible to both the provider and the customer [13].

V. MULTI CLOUD MODELS

It is extremely risky to store all of your data on a single cloud system because there is a chance that the cloud will collapse or become insecure. Thus, storing the file on several clouds are the best and most practical course of action [5].Some of the workable multi-cloud models are listedhere.1.2X Replication Model 2.Data Partitioning Model 3. Cloud-RAID Model

A. *2X Replication Model*

One of the most widely used models to guarantee data availability in storage is replication. This model is used by many systems to guard against data loss. The same data is written or copied using the 2X replication paradigm and is duplicated in several places. To put it another way, 2X replication keeps a backup copy of the same data so that data retrieval is unaffected by the failure of a single machine.

The 2X replication model's general architecture is depicted in Figure 2.1.The user data that needs to be stored in this approach is first encrypted and kept in two distinct clouds. The data can be recovered from the other cloud in the event that one of them fails. This model's primary drawback is its increased storage, expense, time, and bandwidth requirements. The CIA Triad Model's concepts of Confidentiality and Availability are satisfied by the 2X replication model, although requiring twice as much as the storage [14].

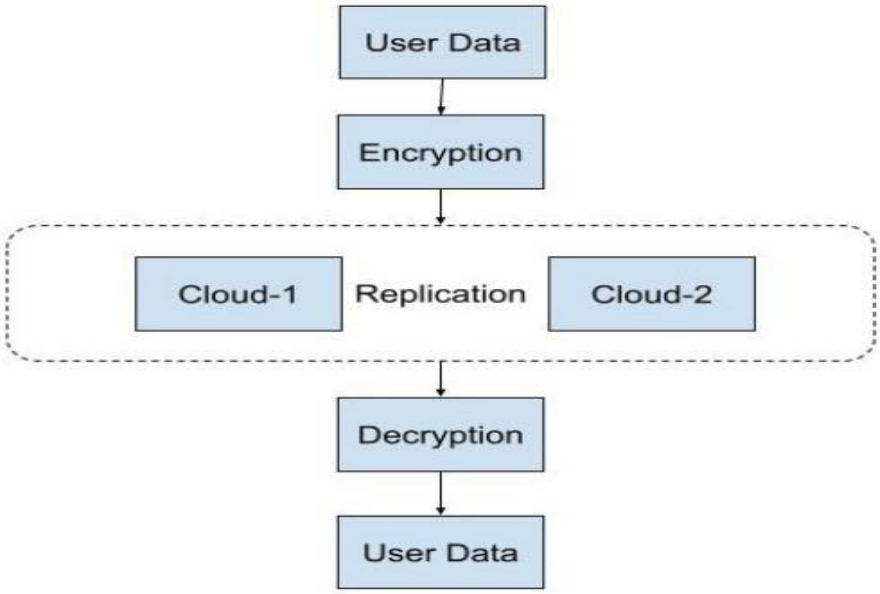


Figure2.1: 2XReplicationModel [15]

B. Data Partitioning Model

Data partitioning refers to the process of separating data over multiple clouds to improve data security. Storing data in a single cloud or using a 2X replication technique does not ensure its security and confidentiality. The Data Partitioning Model protects data by isolating and storing it across many clouds [15]. Figure 2.2 displays the architecture of the Data Partitioning Model. This concept involves encrypting user data before dividing it into two partitions using the Maximum Distance Separable (MDS) technique for cloud storage. Each encrypted partition is copied and stored across many clouds. If any partitions are missing during retrieval, they can be rebuilt from another duplicated partition. The combining process combines both partitions into a single file, which is then encrypted to get the original file. This model protects customer data in the cloud, preventing cloud providers or attackers from viewing it.

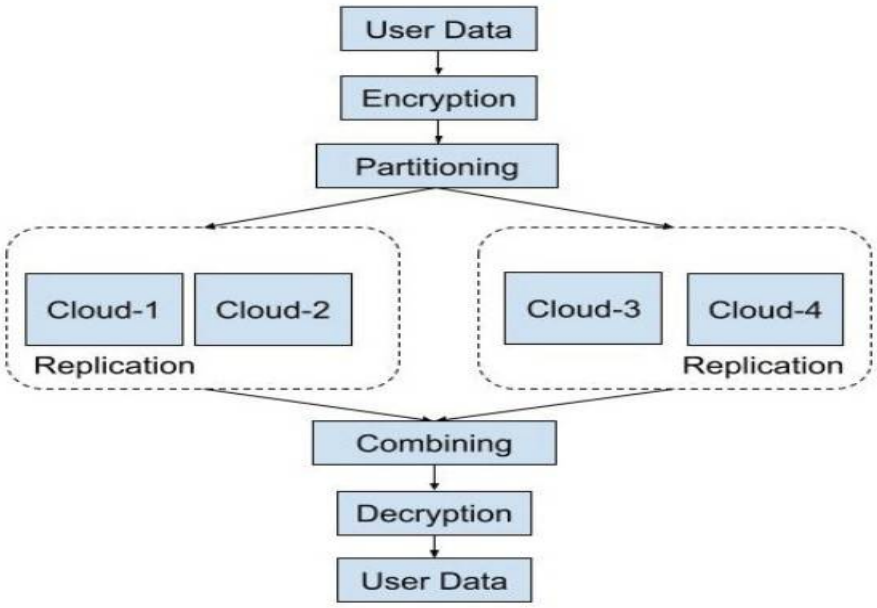


Figure2.2: Data Partitioning Model [15]

C. Cloud Raid Model

RAID, or redundant array of independent discs, protects against drive failures by storing identical data in many locations. RAID levels can provide data redundancy for a variety of applications, be it not all are intended to do so. RAID level 6 relies on parity block-level striping. Using additional parity allows the array to function even if two disks fail simultaneously [15]. Figure2.3 illustrates the architecture of the cloud-RAID paradigm. To ensure secrecy, user data is initially encrypted before being stored in the cloud. RAID 6 ensures data availability even in event of loss. In the event of a cloud outage, data is recreated using parity strips. The cloud-RAID approach ensures confidentiality and availability. The biggest downside of this design is the sluggish writing procedure. Additionally, If a cloud breaks, rebuilding a RAID array may take longer.

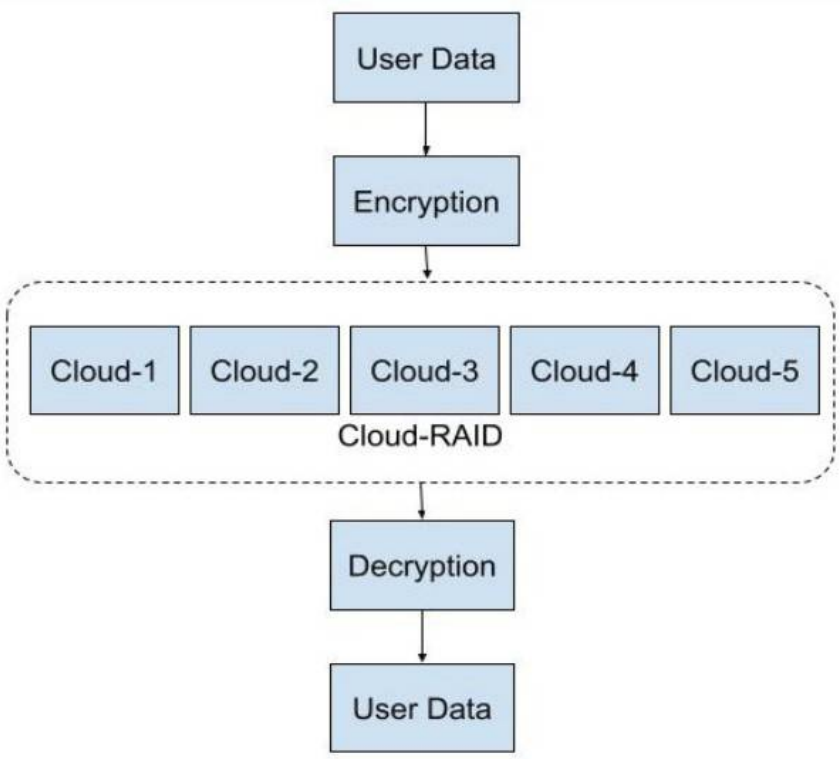


Figure2.3: Cloud RAID Model [15]

VI. FOUR TIER SECURE CLOUD STORAGE ARCHITECTURE

Four-tier secure cloud storage architecture (FTSCSA) model is a new approach for introducing the data storing and retrieval of both public as well private clouds. There are four layers to it — encryption, encoding, hashing, and allocation. The file is encrypted with an auto-generated key while it gets stored. Based on a simple guide for regeneration codes [16], the participants were then grouped by sections. Each encoded chunk is hashed separately using the MD5 algorithm [17]. Finally, the chunks are allocated to different cloud storages using the Circular Shift Chunk Allocation (CSCA) algorithm. The retrieval process was reversed. This CSCA algorithm locates different cloud chunks, encrypts and hashes them upon download in order to check their integrity. Regenerated codes were used to recover missing or tampered chunks. These chunks were combined, decoded and decrypted to restore the file. The solution improves security, integrity and availability of data across multiple clouds.

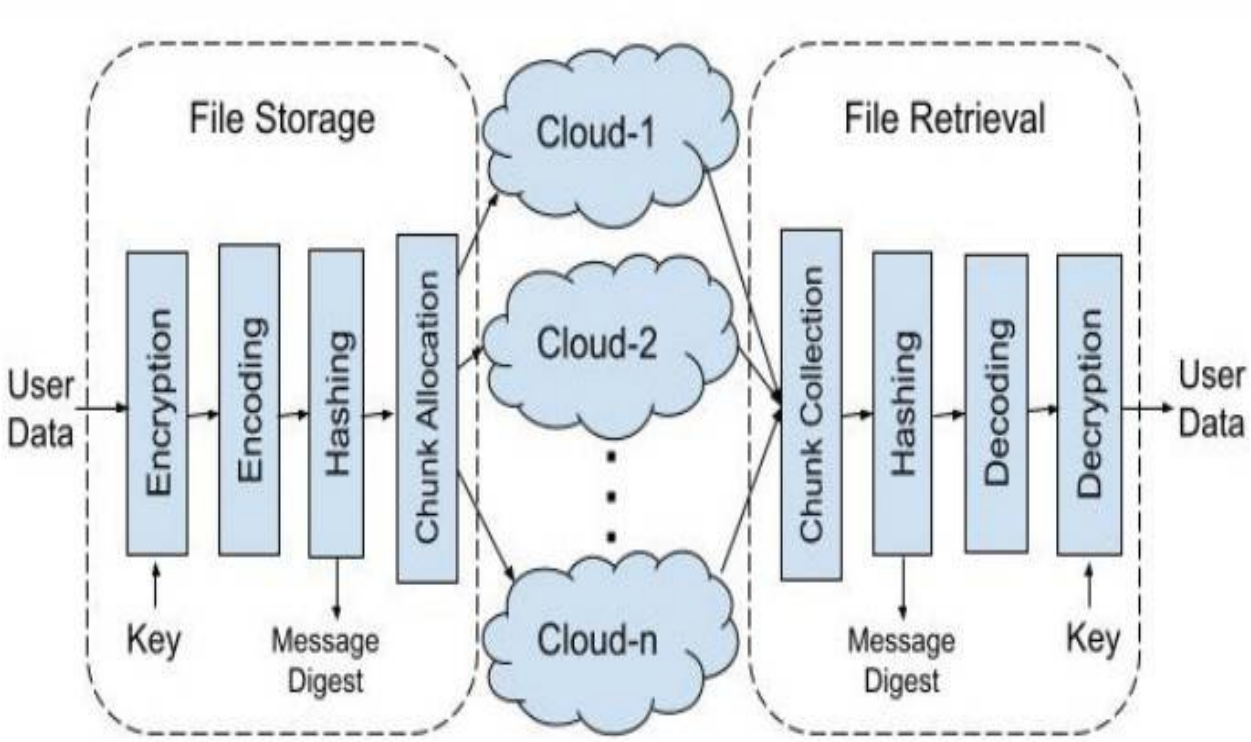


Figure 4.1: Four Tier Secure Cloud Storage architecture [15]

A. ChaCha20 Encryption, Blake3 Hashing, and Dynamic Chunk Allocation in Existing Architectures

The Advanced Encryption Standard (AES) is a popular encryption due to using a highly-robust cipher and is widely used in cloud environments. But it could be susceptible to timing and side-channel attacks, which are common in shared environments like multicloud [18]. A highly performing and secure stream cipher called ChaCha20 raises as a good alternative especially in multicloud cases where speed of encryption really matters, and low computational overhead is important. ChaCha20 encryption is applied to each data segment before it is distributed. This decentralized encryption process ensures that data is protected individually across all clouds, minimizing risks from provider-based vulnerabilities. MD5 is a cryptographic hash function commonly used with cloud storage systems for data integrity validation but MD5 is not collision resistant, means where two different inputs can produce the same hash output. As computing power increases, the possibility of finding collisions increases as well, making MD5 unsuitable for high-security applications. Blake3 is designed to avoid vulnerabilities that have affected MD5, such as collision resistance. Blake3 is optimized for parallelism and can hash data more quickly than older algorithms and handle large volumes of data efficiently. CSCA is an approach where the data is simply split into uniform, preset block sizes. These chunks are then scattered throughout the cloud infrastructure. Using fixed-size chunks might leads to internal fragmentation, where files waste storage by not completely filling out a chunk. The difference with the CSCA chunking method which is static in nature is that the dynamic chunk allocation will reduce the problem of fragmentation where chunks are allocated based on the type of the data being stored thereby ensuring that storage is effectively well utilized.

VII. RESULT AND ANALYSIS

This research's sole purpose is to enhance the secure multi cloud storage via integrating Chacha20, Blake3 and Dynamic chunk allocation into FTSCSA. The performance is evaluated against existing FTSCSA which uses AES, MD5 and CSCA. For this we have taken three different datasets D1, D2 and D3 varying in size. The hardware specifications for experiment

include client machine with an Intel Processor (8 logical processors, 4 cores, 8GB of RAM and 2.40GHz. Additionally, Google Colab is used with Python 3.10 version.

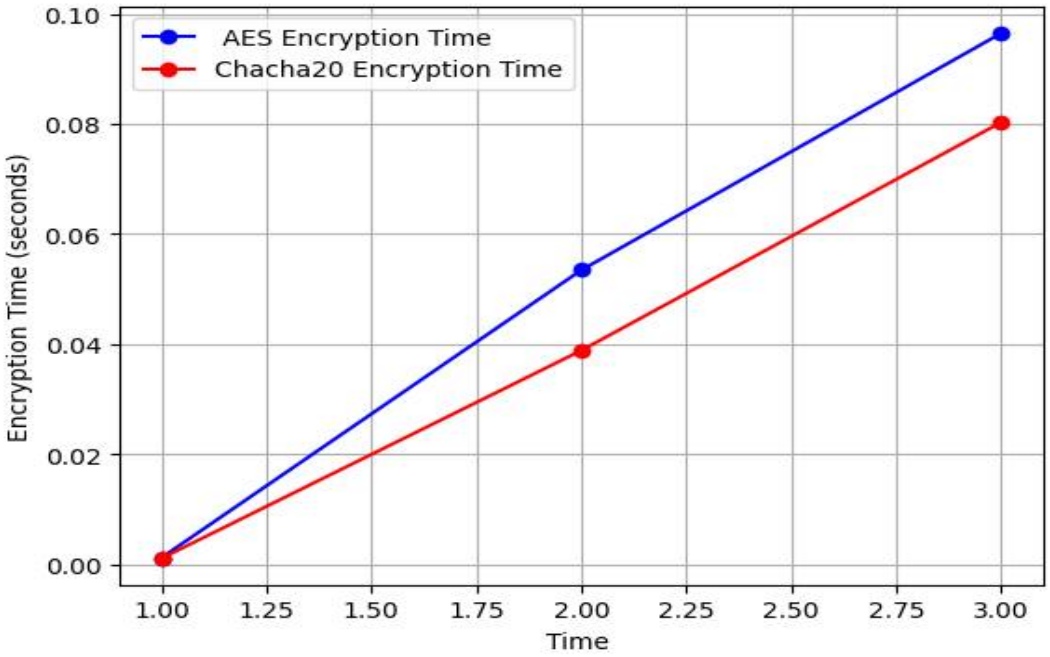


Figure 7.1: Encryption Time Taken By AES and Chacha20 for Dataset (1, 2 and 3)

Dataset	AES Encryption Time (s)	ChaCha20 Encryption Time (s)	Improvement (%)
Dataset 1	0.001166	0.001051	9.86% Faster
Dataset 2	0.053446	0.038845	27.31% Faster
Dataset 3	0.096493	0.080312	16.76% Faster

Figure 7.1 shows Chacha20 encryption performed faster than AES across datasets that we used. The table shows that Chacha20 performed 17.9766% better than the existing AES.

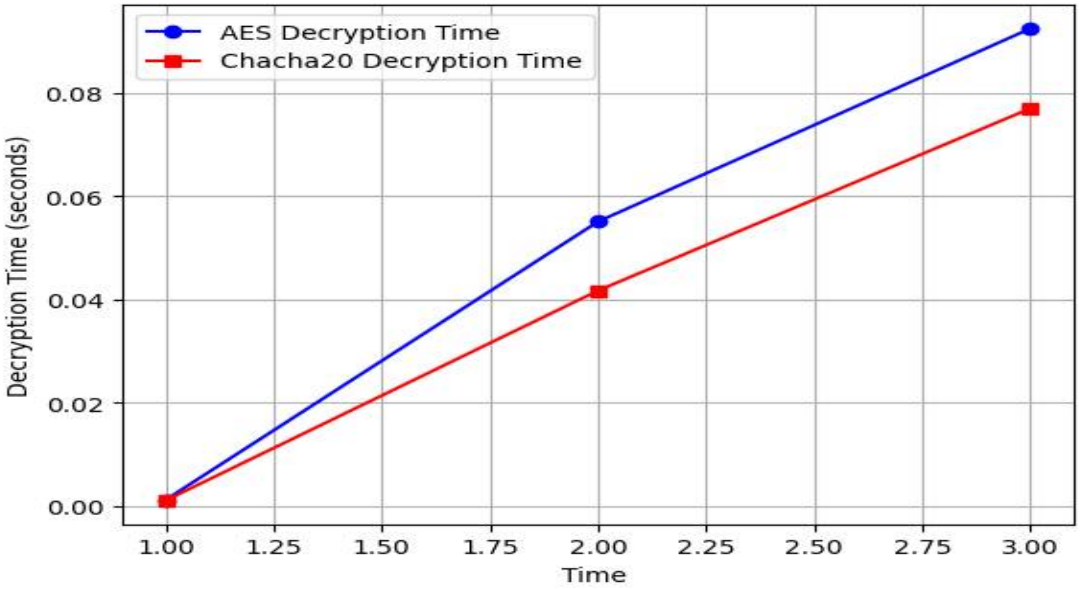


Figure 7.2: Decryption Time Taken By AES and Chacha20 for Dataset (1, 2 and 3)

Dataset	AES Decryption Time (s)	ChaCha20 Decryption Time (s)	Improvement (%)
Dataset 1	0.001125	0.001011	10.13% Faster
Dataset 2	0.055120	0.041733	24.28% Faster
Dataset 3	0.092591	0.077087	16.78% Faster

Figure7.2 shows the decryption time taken by both the techniques. Chacha20 exhibited faster decryption than AES. The above table shows that Chacha20 performed 17.0633% better than the existing AES in decryption part.

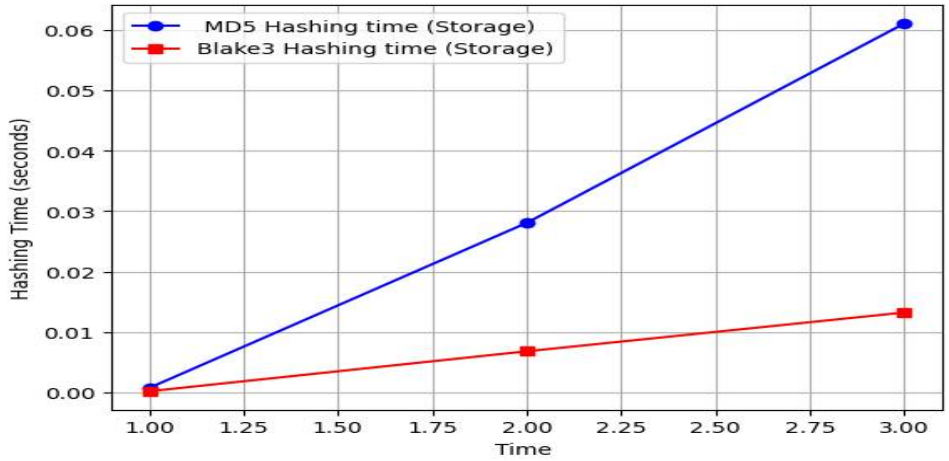


Figure 7.3: Hashing Time for Storage Taken By MD5 and Blake3 for Dataset (1, 2 and 3)

Dataset	MDS Hashing Time (s)	Blake3 Hashing Time (s)	Improvement (%)
Dataset 1	0.005636	0.000213	96.22% Faster
Dataset 2	0.028067	0.006826	75.67% Faster
Dataset 3	0.061053	0.013243	78.30% Faster

Figure7.3 shows that Blake3 hashing technique while storage process took less time than MD5 hashing outperforming MD5 in performance which is being used in the existing system. Blake3 is much more secure than MD5. The experiment shows that for Dataset1, Dataset2 and Dataset3 Blake3 maintained its advantage with 83.3966% times as compared to MD5's.

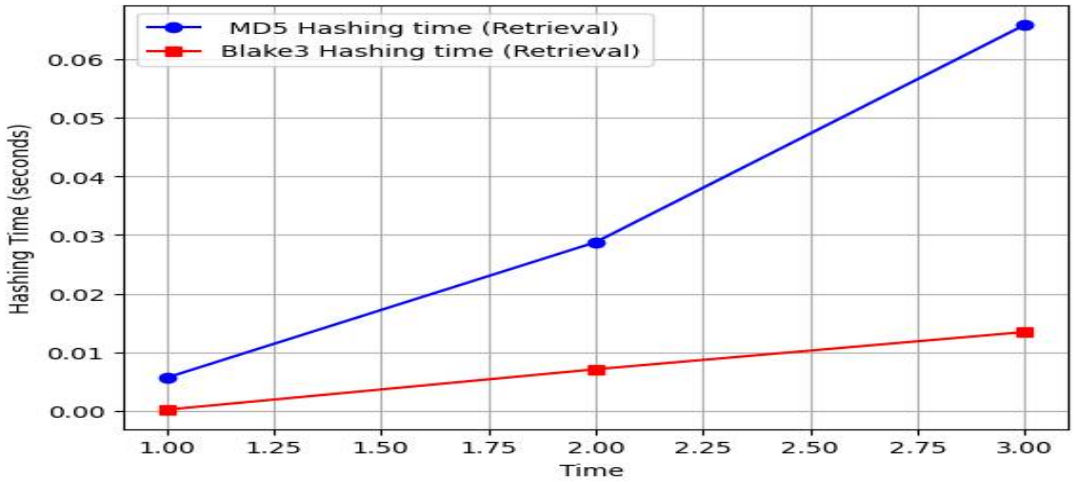


Figure 7.4: Hashing Time for Retrieval Taken By MD5 and Blake3 for Dataset (1, 2 and 3)

Dataset	MDS Hashing Time (s)	Blake3 Hashing Time (s)	Improvement (%)
Dataset 1	0.005412	0.000198	96.34% Faster
Dataset 2	0.027543	0.006415	76.71% Faster
Dataset 3	0.059827	0.012689	78.78% Faster

Figure 7.4 also shows that Blake3 is performing better than the classic technique used in FTSCSA.

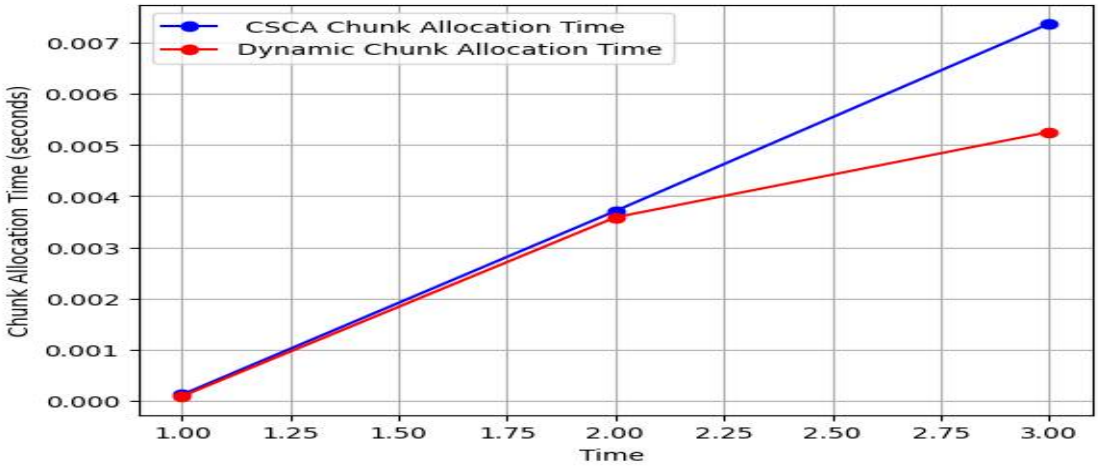


Figure 7.5: Chunk Allocation Time Taken By CSCA and Dynamic Chunk Allocation for Dataset (1, 2 and 3)

Dataset	CSCA Time (s)	Dynamic Chunk Allocation Time (s)	Improvement (%)
Dataset 1	0.007143	0.004212	41.03% Faster
Dataset 2	0.035478	0.020631	41.86% Faster
Dataset 3	0.071923	0.042784	40.50% Faster

Figure7.5 shows the time taken by Dynamic chunk allocation is less and optimal when it is being compared to CSCA which is static allocation method.

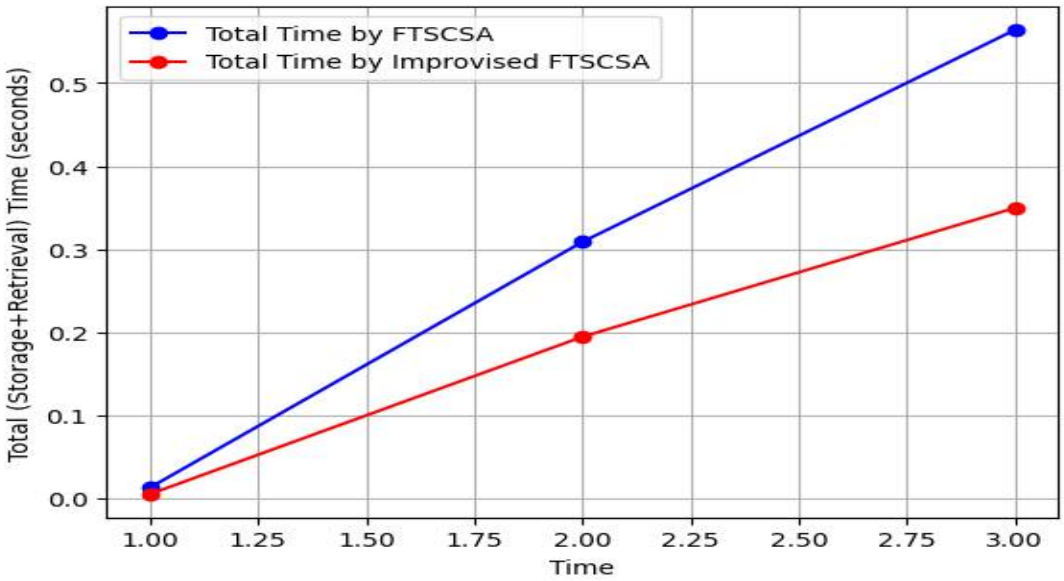


Figure 7.6: Total (Storage and Retrieval) Time Taken By FTSCSA and Improvised FTSCSA for Dataset (1, 2 and 3)

Dataset	Existing FTSCSA Time (s)	Improvised FTSCSA Time (s)	Improvement (%)
Dataset 1	0.089645	0.032965	63.2% Faster
Dataset 2	0.176932	0.111455	37.01% Faster
Dataset 3	0.253879	0.157406	38% Faster

Figure7.6 tells the total time taken by the datasets to process under the existing model and the improvised version of it. The graph shows that the improvised version used with Chacha20, Blake3 and Dynamic Chunk Allocation takes lesser time than FTSCSA and is much more secure than the existing FTSCSA. Experiment shows that improvised version was 63.2% faster than existing one for Dataset1, 37.01% faster for Dataset2 and 38% faster for Dataset3.

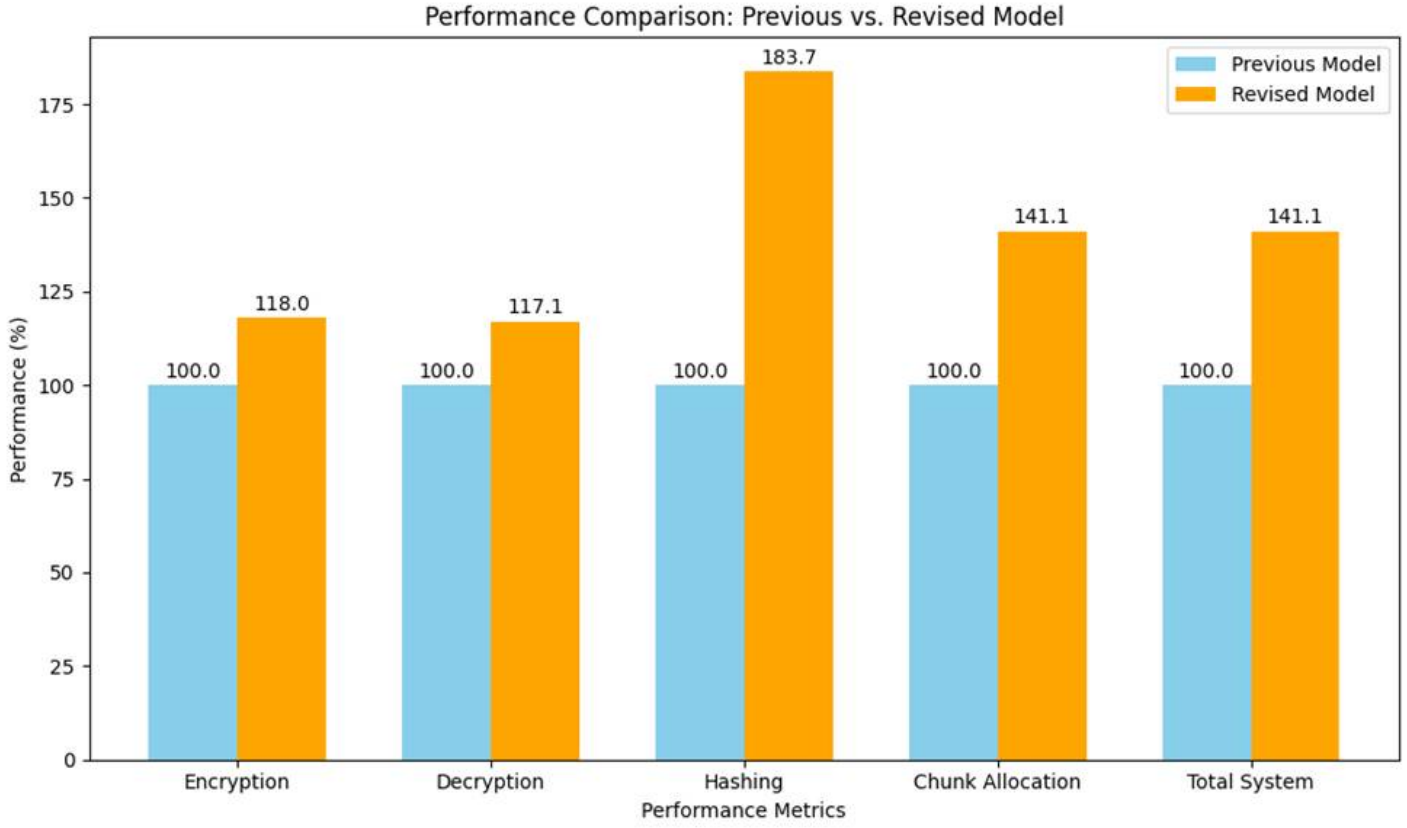


FIGURE 7.7: Comparative Analysis of performance metrics on the models

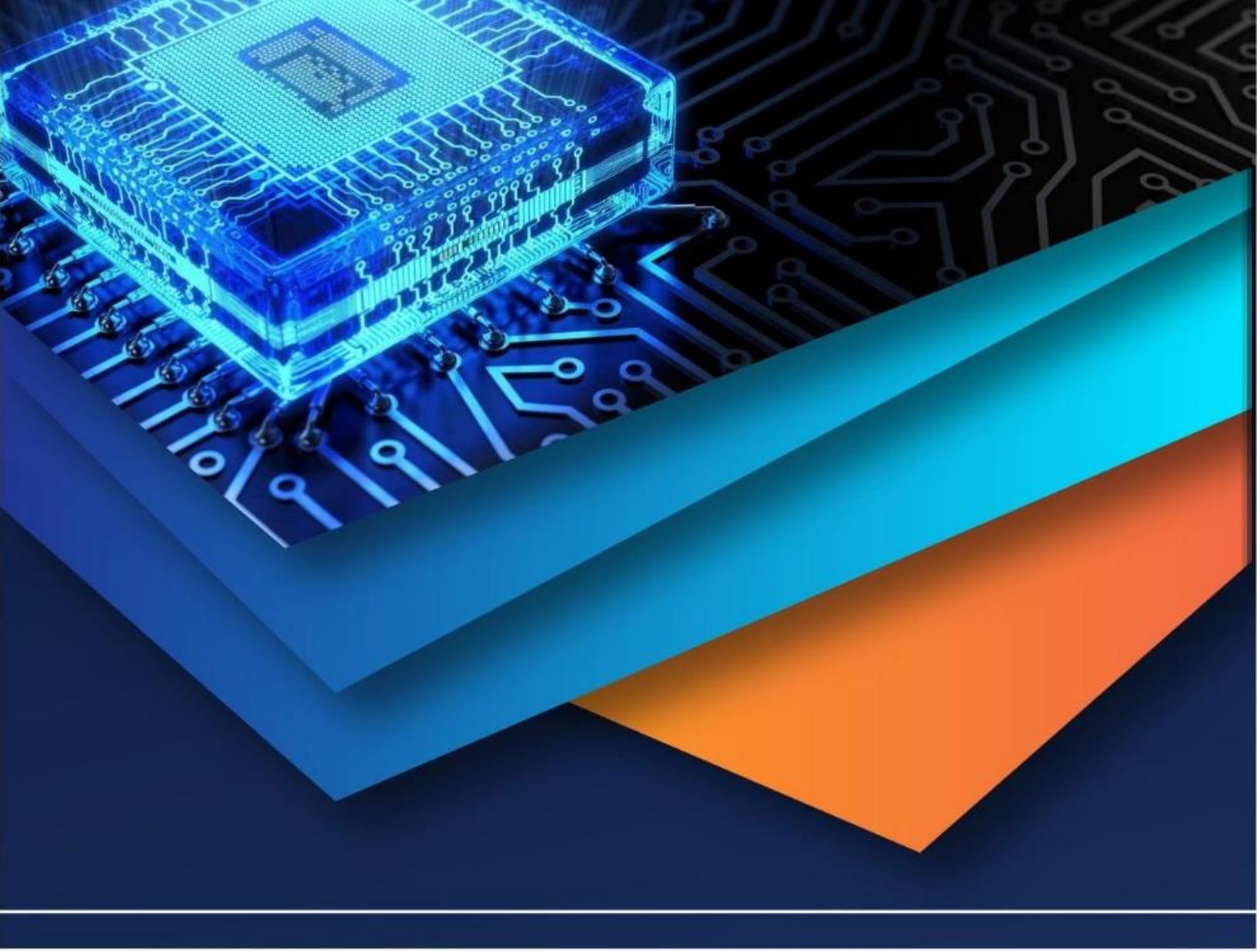
VIII. CONCLUSION

The paper proposed a security, efficiency and scalability architecture for multi-cloud storage system. The new architecture solved the prompts of the original FTSCSA, used ChaCha20 encryption, Blake3 hashing, and dynamic chunk allocation. With it came superior encryption speed, robust data integrity, and optimized resource utilization. These results further confirm that the proposed system is a feasible solution for enterprises who new desire of improved data privacy, integrity, and performance in up to-date multi-cloud environments. Future work may investigate further optimization for various workload patterns, and other cryptographic constructs to broaden its applicability.

REFERENCES

[1] G. Garrison, S. Kim, and R. L. Wakefield, "Success factors for deploying cloud computing," Commun. ACM, vol. 55, no. 9, pp. 62–68, Sep. 2012, doi: 10.1145/2330667.2330685.
[2] L. Shao Xiong and T. Badarch, "Research on Designs of Modern Payment Systems in China," Am. J. Comput. Sci. Technol., Mar. 2023, doi: 10.11648/j.ajcst.20230601.12.
[3] C. Olive, "Cloud Computing Characteristics Are Key".

[4] J. Hong, T. Dreibholz, J. A. Schenkel, and J. A. Hu, "An Overview of Multi-cloud Computing," in Web, Artificial Intelligence and Network Applications, vol. 927, L. Barolli, M. Takizawa, F. Xhafa, and T. Enokido, in Advances in Intelligent Systems and Computing, vol. 927. , Cham: Springer International Publishing, 2019, pp. 1055–1068. doi: 10.1007/978-3-030-15035-8_103.
[5] N. Mouha, "Review of the advanced encryption standard," National Institute of Standards and Technology (U.S.), Gaithersburg, MD, NIST IR 8319, Jul. 2021.doi: 10.6028/NIST.IR.8319.
[6] D. J. Bernstein, "ChaCha, a variant of Salsa20".
[7] G. Procter, "A Security Analysis of the Composition of ChaCha20 and Poly1305".
[8] R. K. Muhammed et al., "Comparative Analysis of AES, Blowfish, Twofish, Salsa20, and ChaCha20 for Image Encryption," Kurd. J. Appl. Res., vol. 9, no. 1, pp. 52–65, May 2024, doi: 10.24017/science.2024.1.5.
[9] P. Mell, "The NIST Definition of Cloud Computing," Recomm. Natl. Inst. Stand. Technol., 2011, Accessed: Nov. 25, 2024. [Online]. Available: https://cloudinfosec.wordpress.com/wp-content/uploads/2013/05/the-nist-definition-of-cloud-computing.pdf
[10] A. E. Youssef, "Exploring Cloud Computing Services and Applications," vol. 3, no. 6, 2012.
[11] J. Broberg and A. Goscinski, "The University of Melbourne and Manjrasoft Pty Ltd., Australia".
[12] A. Rashid and A. Chaturvedi, "Cloud Computing Characteristics and Services A Brief Review," Int. J. Comput. Sci. Eng., vol. 7, no. 2, pp. 421–426, Feb. 2019, doi: 10.26438/ijcse/v7i2.421426.
[13] Z. Mahmood, "Cloud Computing: Characteristics and Deployment Approaches," in 2011 IEEE 11th International Conference on Computer and Information Technology, Paphos, Cyprus: IEEE, Aug. 2011, pp. 121–126. doi: 10.1109/CIT.2011.75.
[14] P. R. Kumar, P. H. Raj, and P. Jelciana, "Exploring Data Security Issues and Solutions in Cloud Computing," ProcediaComput. Sci., vol. 125, pp. 691–697, 2018, doi: 10.1016/j.procs.2017.12.089.
[15] A. J. Kumar, A Novel Four Tier Secure Cloud Storage Architecture for Enhanced Data Security, Ph.D. dissertation, guided by C. Columbus, 2023.
[16] O. Khan, R. Burns, J. Plank, W. Pierce, and C. Huang, "Rethinking Erasure Codes for Cloud File Systems: Minimizing I/O for Recovery and Degraded Reads".
[17] P. Gupta and S. Kumar, "A Comparative Analysis of SHA and MD5 Algorithm," vol. 5, 2014.
[18] Z. Najm, D. Jap, B. Jungk, S. Picek, and S. Bhasin, "On Comparing Side-channel Properties of AES and ChaCha20 on Microcontrollers," in 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Chengdu: IEEE, Oct. 2018, pp. 552–555. doi: 10.1109/APCCAS.2018.8605653.
[19] H. S. Baqtian and N. M. Al-Aidroos, "Three Hash Functions Comparison on Digital Holy Quran Integrity Verification," International Journal of Scientific Research in Network Security and Communication, vol. 11, no. 1, pp. 1–7, Feb. 2023.
[20] B. K. Rani, B. P. Rani, and A. V. Babu, "Cloud Computing and Inter-Clouds – Types, Topologies and Research Issues," ProcediaComput. Sci., vol. 50, pp. 24–29, 2015, doi: 10.1016/j.procs.2015.04.006.
[21] J. Sugier, "Comparison of power consumption in pipelined implementations of the BLAKE3 cipher in FPGA devices," Int. J. Electron.Telecommun., pp. 23–30, Mar. 2024, doi: 10.24425/ijet.2023.147710.
[22] N. Aleisa, "A Comparison of the 3DES and AES Encryption Standards," Int. J. Secur. Its Appl., vol. 9, no. 7, pp. 241–246, Jul. 2015, doi: 10.14257/ijsia.2015.9.7.21.
[23] S. Kumar, D. Kumar, and H. S. Lamkuche, "TPA Auditing to Enhance the Privacy and Security in Cloud Systems," Journal of Cyber Security and Mobility, vol. 10, no. 3, pp. 537–568, 2021.
[24] D. Puthal, B. P. S. Sahoo, S. K. Mishra, and S. Swain, "Cloud Computing Features, Issues and Challenges: A Big Picture," in Proceedings of the 2015 International Conference on Computational Intelligence & Networks (CINE), Bhubaneswar, India, Jan. 2015, pp. 116–123, doi: 10.1109/CINE.2015.31.
[25] D. Puthal, B. P. S. Sahoo, S. K. Mishra, and S. Swain, "Cloud Computing Features, Issues and Challenges: A Big Picture," in Proceedings of the 2015 International Conference on Computational Intelligence & Networks (CINE), Bhubaneswar, India, Jan. 2015, pp. 116–123, doi: 10.1109/CINE.2015.31



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY