Role Based Dynamic Encryption for Secure Cloud Data Access

[1]Pushpjeet Cholkar, [2]Dr. Margi Patel

[1]Department of Computer Science Engineering, Indore Institute of Science and Technology, Indore (M.P.)

[2]Department of Computer Science Engineering, Indore Institute of Science and Technology, Indore (M.P.)

Email-: [1]pushpjeet@gmail.com, [2]margi.patel@indoreinstitute.com

* Corresponding Author: Pushpjeet Cholkar

Abstract: *Cloud computing represents a major advancement in Internet-based computing, providing on-demand resource sharing, such as processing and data storage, across globally distributed machines and smart devices. This model enables ubiquitous access to a shared pool of configurable computing resources. A key application area of cloud computing is cloud data storage solutions. Motivated by this, the work focuses on cloud storage security, aiming to protect data, and establish authentication protocols for users and devices. This work outlines a role-based dynamic encryption algorithm for cloud storage security model that categorizes users into three distinct roles: AuthU, LAU, and NAU. Distinct encryption rule is applied for each role. The result analysis presented its efficacy in terms of encryption, decryption and memory utilization.*

Keywords: *Cloud Storage, Cloud Security, Data Confidentiality, Encryption, Multi-User Architecture, Role-based Encryption.*

I. INTRODUCTION

Cloud storage, a pivotal element in modern computing, offers scalable and on-demand access to data, revolutionizing how individuals and organizations manage and access information. Despite its numerous benefits, such as cost-efficiency and the elimination of physical infrastructure, the security of data in the cloud is a pressing concern. The contemporary reliance on cloud storage brings forth challenges in maintaining the confidentiality, integrity, and availability of sensitive information within its dynamic and shared environment [1]. Addressing these issues, current research focuses on exploring the utilization of multi-tenant architecture and the integration of dynamic encryption. This approach aims to enhance security in cloud storage while accommodating its unique architecture. The anticipated outcomes of this research are significant, expected to benefit both individual and business users of cloud storage, and potentially influencing future advancements in secure cloud computing. The objective is to pave the way for a more resilient and privacy-centric cloud storage paradigm, navigating the complexities of security in a shared, digital ecosystem [2].

A. Cloud Computing

Cloud computing is a method of delivering various computing resources over the internet, hosted remotely in data centers managed by cloud service providers (CSPs). These resources include applications, servers (both physical and virtual), data storage, development tools, and networking capabilities. Users typically access these services via a subscription or a pay-as-you-go model. Compared to traditional on-premises IT infrastructure, cloud computing offers several benefits:

- Reduced IT Expenditures: It eliminates or reduces the costs and efforts associated with purchasing, setting up, configuring, and managing on-site infrastructure.
- Increased Agility and Time-to-Value: Enterprises can quickly deploy applications, avoiding the delays in acquiring and setting up hardware and software. It also allows fast access to infrastructure for specific users like developers and data scientists .
- Scalability and Cost-Efficiency: Users can scale services up or down based on demand, avoiding the expense of unused capacity. The global network presence of cloud providers also enhances application performance worldwide.

Additionally, cloud computing relies on virtualized IT infrastructure, including servers, operating systems, networking, etc., which are abstracted and shared across different hardware, using specialized software [4].

As cloud storage becomes increasingly essential in modern IT infrastructures, it faces several security challenges:

- Data Breaches and Unauthorized Access: Unauthorized access, often due to weak authentication, compromised credentials, or vulnerabilities in the cloud service provider's infrastructure, remains a major concern.
- Inadequate Identity and Access Management (IAM): Poorly configured or incomplete IAM policies can lead to improper access permissions, elevating the risk of unauthorized data access and actions within cloud environments.
- Data Leakage and Loss: The risk of unintentional exposure or loss of sensitive data is high, stemming from misconfigurations, accidental sharing, or insider threats.
- Insufficient Data Encryption Practices: Weak encryption of data, both at rest and in transit, heightens the risk of data exposure, particularly if encryption keys are poorly managed.

- Lack of Visibility and Control: Limited insight into the security controls of cloud service providers challenges organizations in assessing and managing security risks effectively.

Addressing these challenges is vital for organizations to harness the benefits of cloud storage while protecting sensitive information.

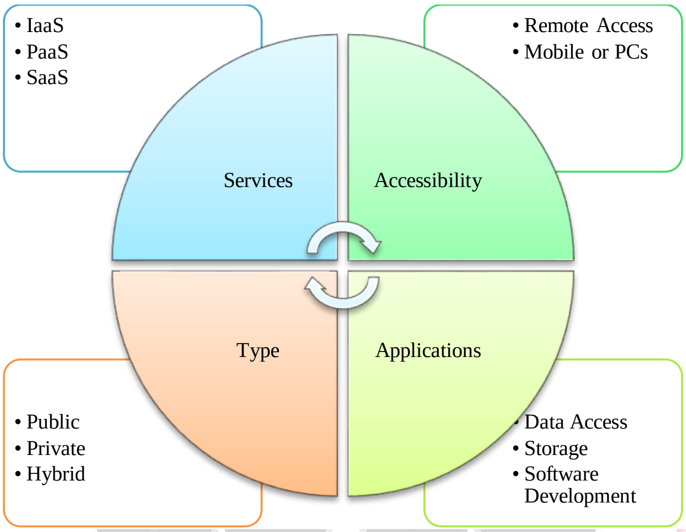

Figure 1: Cloud Computing

II. LITERATURE REVIEW

Masood et al. [1] presents a state-of-the-art study of VCC that focuses on VCC architecture, its features analysis, and extensive VCC applications. Second, the proposed threats identification taxonomy and an exhaustive survey on security and privacy issues in VCC are presented under a layered approach: physical resource layer, vehicle-to-anything (V2X) network layer, and vehicular cloud layer, as well as at a complete system level. Finally, we highlight and discuss challenges and open research issues that can be considered as future research directions. Yassin et al. [2] proposed a multi-tenant intrusion detection framework as a service for SaaS (MTIDaaS) to allow the provider to undertake such integration. Our MTIDaaS has been integrated and tested in a real public cloud environment. It provides security-as-a-service (SecaaS) for both provider and tenant with high levels of portability, flexibility and cost-effectiveness. The experimental results demonstrate that our MTIDaaS offers easy integration of IDS with little virtualization overhead and insignificant impact on HTTP response time. Shin et al. [3] extend server-aided encryption to a decentralized setting that consists of multiple KSs.This way, our scheme simultaneously offers flexibility of KS management and cross-tenant deduplication over encrypted data. Therefore, it allows cloud storage services to offer high deduplication efficiency and scalability while preserving strong data confidentiality. We show the result of performance analysis on the proposed scheme by conducting extensive experiments. In addition, our security analysis demonstrate that the proposed scheme satisfies all desired security properties. Zhang et al. [4] introduces a carefully-designed key wrapping layer to overcome these challenges. A small symmetric wrapping key (SWK) is generated for each tenant as the master key to resolve the former two challenges, while a special private key wrapping scheme is adopted to resolve the transparency limitation. Additionally, QKPT incorporates certificate trust to enhance the security of the SWK lifecycle and provides a hardened key server solution without expensive HSM. The evaluation shows that QKPT has a low runtime overhead (≤ 1.2% for SSL/TLS handshakes) and still greatly outperforms the software baseline (3.5x-17x) owing to the crypto offloading. Cogranne et al. [5] presents a robust and cost-effective solution to detect malicious activities in a public virtualized environment. Its contribution is twofold: 1) a scalable and robust workload estimation of the virtual host activities in a cloud and 2) a detection algorithm able to discriminate infected hosts with low malicious activities hidden within their legitimate workload and potentially scattered across several tenants. For both of these contributions, we establish their theoretical performance, which demonstrates their optimality, and we evaluate their efficiency on a dataset made of real data collected on PlanetLab. Wang et al. [6] propose a new architecture and use case driven designs to enable confidential, flexible and collaborative data sharing among such organizations using the same MSS platform. MSS platform is a complex environment where different stakeholders, including authorized MSSP personnel and customers' own users, have access to the same platform but with different types of rights and tasks. As an innovative and pioneering attempt to address the challenge of data sharing in the MSS platform, we hope to encourage further work to follow so that confidential and collaborative sharing eventually happens among MSS platform customers. Syafalni et al. [11] implemented cloud security using homomorphic encryption for data analytics, demonstrating that data can be processed without decryption. Their experiments showed execution times of 2.2 ms, 4.4 ms, and 25 ms for polynomial degrees of 2^6, 2^8, and 2^10, respectively. This approach is particularly beneficial for securing big data in fields like environmental, financial, and hospital analytics. Fursan et al. [12] introduced a novel, lightweight homomorphic cryptographic algorithm with dual encryption layers. The first layer employs a new, effective, lightweight cryptographic method, while the second layer uses multiplicative homomorphic schemes. This technique combines features of both symmetric and asymmetric cryptography, enhancing data security in cloud computing. Boomija and Raja [13] proposed a Secure Partially Homomorphic Encryption (SPHE) algorithm to protect outsourced data and enable multiplication and division operations on ciphertext. They address challenges in cloud environments, such as flexible access control policies that can be abused by attackers, compromising database privacy. Their model integrates the SPHE scheme with a role-based user policy to manage these issues and create a more secure and efficient cloud environment.

III. OBJECTIVE

- To study role of encryption-based cloud security aspects.
- To design a cloud storage security system using role-based dynamic encryption algorithm.
- To decrease computational complexity and reduce computational space required.
- To enhance performance as compared to existing state-of-art models.

IV. PROPOSED METHODOLOGY

This section presents a methodology in which the cloud storage access varies based on the user's role. Users are categorized as either insiders or outsiders. Insiders are members of the organization and are granted full access permissions due to their trusted status. Outsiders, on the other hand, are not part of the organization and are considered potentially unsafe. As a result, an additional layer of security is recommended to provide secure cloud storage from potential threats posed by outsider access. In this proposed model, the role based dynamic encryption will categorize the accessing users in three categories. The flowchart is presented in fig 2.

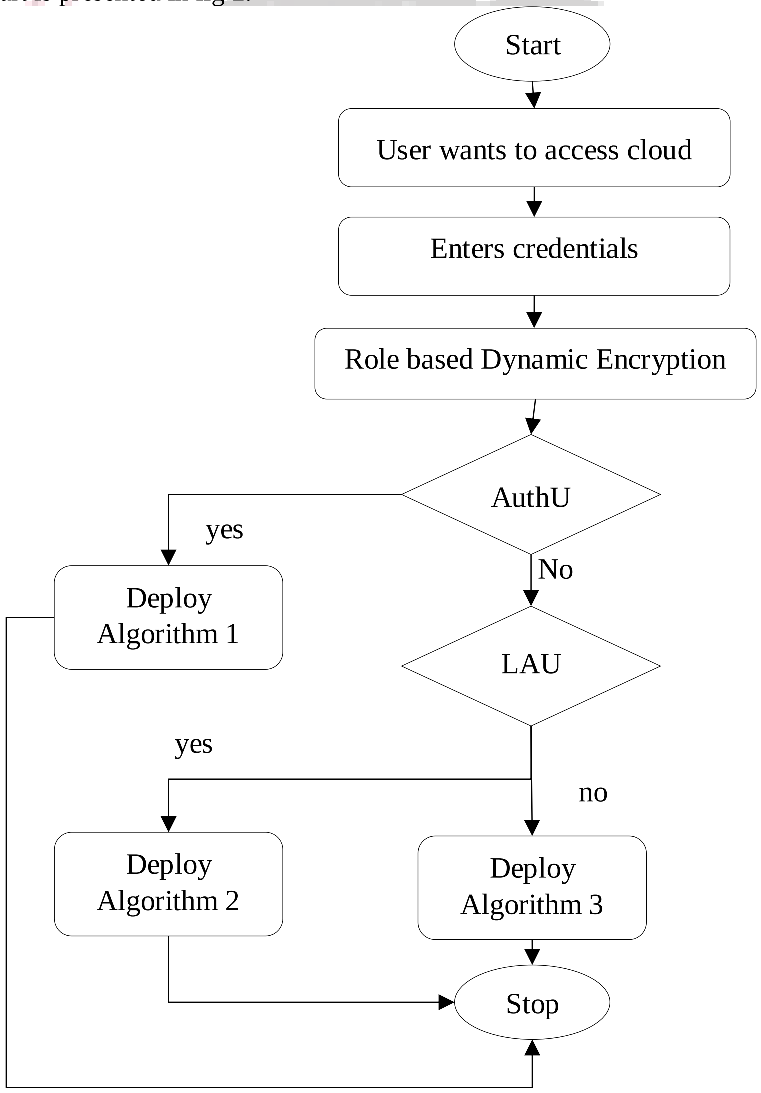

Figure 2: Flowchart of Work

Authorized User Role (AuthU Role): These are those users whose access is fully authorized by server. According to rules of storage server, if provided credentials and informations are matched then, a session secure key is used using asymmetric algorithm.

Limited Access User Role (LAU Role): These are those users whose access is not fully authorized by server. According to rules of storage server, if provided credentials and informations are matched then, user will be provided a session license to access file. The file at cloud storage end will be accessed using hybrid key encryption.
Non-Authenticated User Role (NAU Role): These users may or may not be attackers who pretent to be AuthU or LAU and wants to access file. If they are detected to be NAU then server will apply fuzzy rule based reencryption will be performed.

Authorized User Role (AuthU Role) Algorithm

In this step, the the authorized user access the cloud services using symmetric algorithm as they are fully authorized by the server. The encryption and decryption algorithm is presented as below:

Encryption

Plaintext: A block of 128 bits.
Key: Can be 128, 192, or 256 bits.
Ciphertext: Encrypted block of 128 bits.
KeyExpansion: Expand the input key into a key schedule (44, 52, or 60 words for 128, 192, or 256-bit keys respectively).
AddRoundKey: XOR the plaintext block with the first four words of the expanded key.
For 9, 11, or 13 rounds (depending on key size):
SubBytes: Apply the S-box to each byte of the block for non-linear substitution.
ShiftRows: Cyclically shift the rows of the block over different offsets.
MixColumns: Mix each column of the block.
AddRoundKey: XOR the block with a block of the key schedule.
Final Round: SubBytes and ShiftRows
AddRoundKey
Return
Return the resulting 128-bit block as ciphertext.

Decryption

Ciphertext: A block of 128 bits.
Key: Can be 128, 192, or 256 bits.
Plaintext: Decrypted block of 128 bits.
KeyExpansion: Same as in encryption.
AddRoundKey: XOR the ciphertext block with the last four words of the expanded key.
For 9, 11, or 13 rounds (depending on key size):
InvShiftRows: Inverse of the ShiftRows operation.
InvSubBytes: Apply the inverse S-box to each byte of the block for substitution.
AddRoundKey: XOR the block with a block of the key schedule.
InvMixColumns: Inverse of the MixColumns operation.
InvShiftRows
InvSubBytes
AddRoundKey
Return
Return the resulting 128-bit block as plaintext.

Limited Access User (LAU) Role Algorithm

Setup

Input: Security parameter λ and level L.

Process:

Generate L large prime numbers $0, \ldots, q_0, \ldots, q_{L-1}$ where $0 < 1 < \cdots < q_0 < q_1 < \cdots < q_{L-1}$.
Select a discrete Gaussian distribution χ for the error distribution.
Plaintext m.
Public parameters are $0, \ldots, q_0, \ldots, q_{L-1} and\ \chi$.

KeyGen

Input: Public parameters.
Process:
Select a random vector s as the secret key.
Compute random switch keys $b = -(a \cdot s + p \cdot e) mod\ q_{L-1}$ where a is random in $R_{q_{L-1}}$, e is sampled from χ, and p is the plaintext modulus.
Public key is (a, b)

Encrypt

Input: Plaintext m.
Process:
Randomly select a vector v with each $\in \{0, \pm 1\} v_i \in \{0, \pm 1\}$.
Sample e_o and e_1 from χ.
Compute $c_o = (b.v + p.e_0 + m) mod q_{L-1}$ and $c_1 = (a.v + p.e_1) mod q_L$

Initial ciphertext is $c_t = (c_o, c_1, L-1)$.

Decrypt

Input: Ciphertext $c_t = (c_o, c_1, i)$ and secret key s.

Process:

Compute $m \leftarrow c_o + c_1 \cdot s \ mod \ q_i \ mod \ p$.

Non-Authenticated User (NAU) Role Algorithm

Users identified as attackers are subjected as unauthorized user role. If they attempt to access the cloud, using any other's session license then reencryption is performed using some randomly generated set of fuzzy rules.

Algorithm

Step 1: Identify the input variables that will determine rule selection.

Step 2: Define Fuzzy Sets and Membership Functions. For each input variable, define fuzzy sets (e.g., Low risk, Medium risk, High risk).

Step 3: Define membership functions for each fuzzy set. A membership function maps the set of rules.

Now of the risk is low then random key is generated for reencryption. If the risk is low then re-encryption will be performed using algorithm 1 and if the risk is medium or high then it is re-encrypted using algorithm 2 but with distorted keys. For distorted keys, the key is represented as the matrix as M with n rows and n columns, where each entry M_{ij} represents a specific bit structure that use the function swap $swap(M_{ij})$ to denote swapping the i-th and j-th bits of each character in the matrix with n number of times. The swapping operation can be defined as: M_{ij}'=M_{ib} and M_{ib}'=M_{ij} for all j in each character.

V. Results and Discussion

This section provides an analytical and numerical analysis of the proposed cloud security framework. This section details the performance assessment of the suggested methodology, which was conducted using simulation analysis on an Intel Core i5 processor with an 8 GB hard drive, utilizing the cloudsim platform. The performance of the proposed algorithm is assessed based on four critical indicators: key generation time, encryption time, decryption time, and computational storage. The focus of the analysis was to evaluate the time taken for encryption, decryption, and key generation processes in the implemented system.

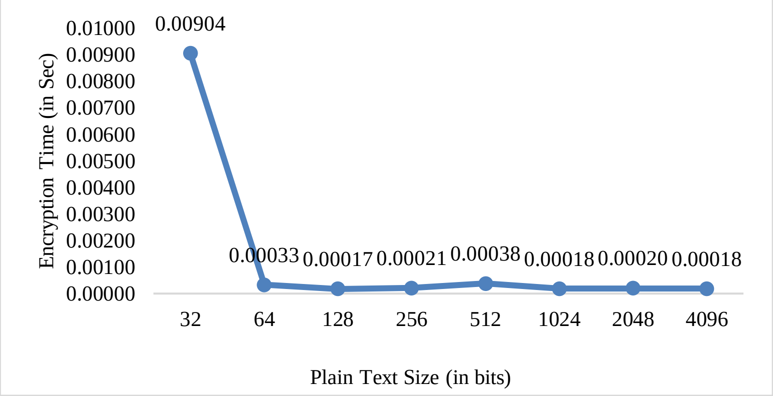

Figure 3: Encryption Time Evaluation for AuthU Role

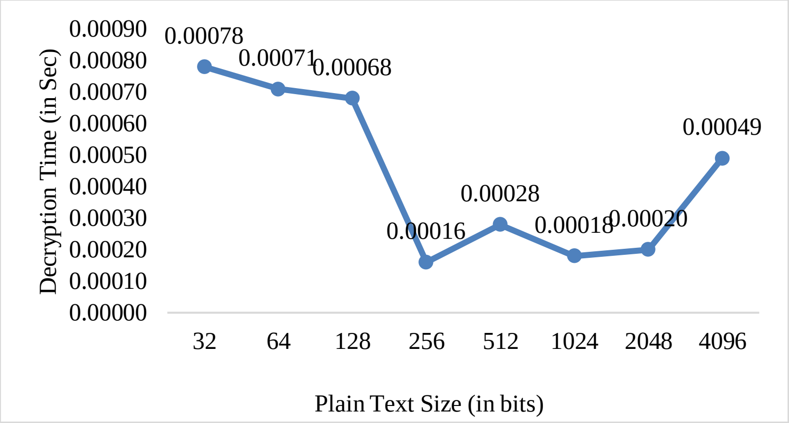

Figure 4: Decryption Time Evaluation for AuthU Role

Figure 3 presents the encryption time analysis for AuthU role. The initial encryption time is relatively high at 0.00904 seconds but then significantly decreases, fluctuating between 0.00017 to 0.00038 seconds in the subsequent measurements. Figure 4 presents the decryption time analysis. The initial decryption time is relatively high at 0.00078 seconds but then significantly decreases, fluctuating between 0.00016 to 0.00049 seconds in the subsequent measurements.

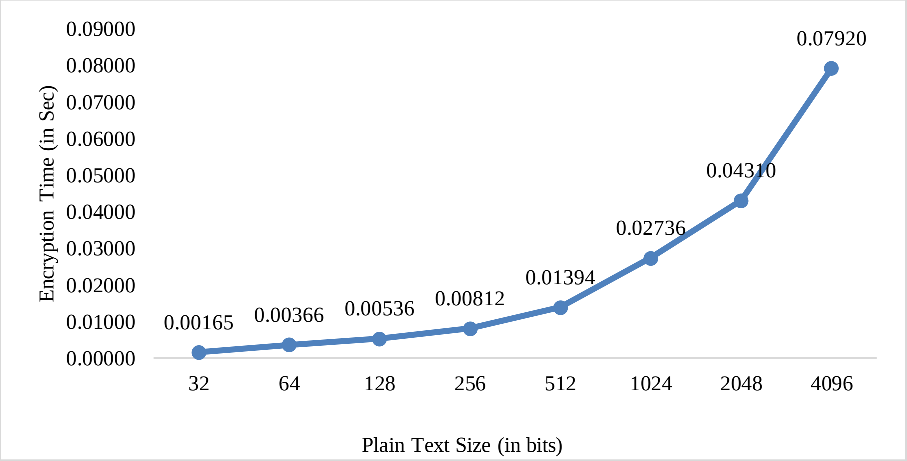

Figure 5: Encryption Time Evaluation for LAU Role

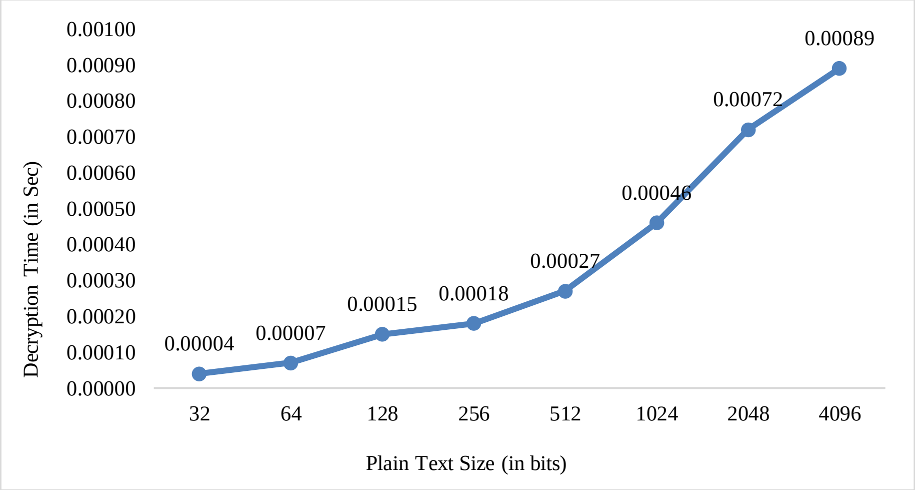

Figure 6: Decryption Time Evaluation for LAU Role

Figure 5 shows encryption time that increases with the plaintext size, starting from 0.00165 seconds for 32 bits and reaching up to 0.07920 seconds for 4098 bits. This suggests that larger plaintext sizes require more time for encryption. Figure 6 shows decryption time that increases as the plaintext size grows, although the times remain quite low, varying from 0.00004 to 0.00089 seconds. The trend is consistent with what is generally expected in cryptographic systems – larger data sizes take longer to decrypt.

Table 1: Execution Time Analysis

User Role	Execution Time (in Sec)
AuthU Role	~0.0006
LAU Role	~0.08
NAU Role	~0.004
Total	~0.08

The execution time analysis of different user roles in a cloud storage security system is presented in table 1. AuthU role based user experience the fastest execution at approximately 0.0006 seconds. In contrast, LAU role based user encounter a longer execution time of around 0.08 seconds. NAU role based user are handled in about 0.004 seconds. The comparative feature analysis is presented below in table 2.

Table 2: Comparative Feature Analysis

Features	Syafalni et al. [11]	Fursan et al. [12]	Boomija and Raja [13]	Proposed
Algorithm	Asymmetric	Asymmetric	Asymmetric	Role based dynamic encryption
Key Gen Time	Moderate	Moderate	Moderate	Faster
Encryption Time	Moderate	Moderate	Moderate	Faster
Decryption Time	Moderate	Moderate	Moderate	Faster
Power Consumption	-	High	Moderate	Low
Memory Consumption	Moderate	Moderate	Moderate	Low

V. CONCLUSION

This study tackles the critical challenge of ensuring data security in cloud storage by introducing a cutting-edge method that combines the principles of multi-user architecture with the robustness of dynamic encryption. The findings reveal that our approach not only strengthens security measures but also maintains low computational and storage overhead, thereby enhancing the overall performance in contrast to current models. The implementation of a role-based dynamic encryption scheme within this framework marks a substantial step forward in crafting secure cloud storage solutions. Looking ahead, further research may refine these techniques and test their adaptability across various computing scenarios to ensure enduring security in the pervasive landscape of cloud services.

REFERENCES

[1] Masood, D. S. Lakew, and S. Cho, “Security and Privacy Challenges in Connected Vehicular Cloud Computing,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 4, pp. 2725–2764, 2020, doi: 10.1109/COMST.2020.3012961.

[2] M. Yassin, H. Ould-Slimane, C. Talhi, and H. Boucheneb, “Multi-Tenant Intrusion Detection Framework as a Service for SaaS,” IEEE Trans. Serv. Comput., vol. 15, no. 5, pp. 2925–2938, 2022, doi: 10.1109/TSC.2021.3077852.

[3] Y. Shin, D. Koo, J. Yun, and J. Hur, “Decentralized Server-Aided Encryption for Secure Deduplication in Cloud Storage,” IEEE Trans. Serv. Comput., vol. 13, no. 6, pp. 1021–1033, 2020, doi: 10.1109/TSC.2017.2748594.

[4] Z. Zhang et al., “QKPT: Securing Your Private Keys in Cloud With Performance, Scalability and Transparency,” IEEE Trans. Dependable Secur. Comput., vol. 20, no. 1, pp. 478–491, 2023, doi: 10.1109/TDSC.2021.3137403.

[5] R. Cogranne, G. Doyen, N. Ghadban, and B. Hammi, “Detecting Botclouds at Large Scale: A Decentralized and Robust Detection Method for Multi-Tenant Virtualized Environments,” IEEE Trans. Netw. Serv. Manag., vol. 15, no. 1, pp. 68–82, 2018, doi: 10.1109/TNSM.2017.2785628.

[6] X.-S. Wang, I. Herwono, F. Di Cerbo, P. Kearney, and M. Shackleton, “Enabling Cyber Security Data Sharing for Large-scale Enterprises Using Managed Security Services,” in 2018 IEEE Conference on Communications and Network Security (CNS), 2018, pp. 1–7. doi: 10.1109/CNS.2018.8433212.

[7] J. Li, Y. Zhang, X. Li, X. Gao, P. Wang, and R. Wang, “Architecture Design and Key Technologies of Electric Vehicle Charging Network Operation Service System Based on Cloud Computing,” in 2018 2nd International Conference on Robotics and Automation Sciences (ICRAS), 2018, pp. 1–5. doi: 10.1109/ICRAS.2018.8443257.

[8] P. Dhiman et al., “Secure Token&ndash;Key Implications in an Enterprise Multi-Tenancy Environment Using BGV&ndash;EHC Hybrid Homomorphic Encryption,” Electronics, vol. 11, no. 13, 2022, doi: 10.3390/electronics11131942.

[9] M. Marwan, A. Kartit, and H. Ouahmane, “Applying homomorphic encryption for securing cloud database,” in 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), 2016, pp. 658–664. doi: 10.1109/CIST.2016.7804968.

[10] K. K. Hingwe and S. Mary Saira Bhanu, “Hierarchical Role-Based Access Control with Homomorphic Encryption for Database as a Service,” in Proceedings of International Conference on ICT for Sustainable Development, 2016, pp. 437–448

[11] I. Syafalni et al., “Cloud Security Implementation using Homomorphic Encryption,” in 2020 IEEE International Conference on Communication, Networks and Satellite (Comnetsat), 2020, pp. 341–345. doi: 10.1109/Comnetsat50391.2020.9328979.

[12] Thabit, Fursan, et al. "A Novel Effective Lightweight Homomorphic Cryptographic Algorithm for data security in cloud computing." International Journal of intelligent networks 3 (2022): 16-30.

[13] Boomija, M. D., and SV Kasmir Raja. "Securing medical data by role-based user policy with partially homomorphic encryption in AWS cloud." Soft Computing 27.1 (2023): 559-568.