Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.060836

ARTICLE

Enhanced Triple Layered Approach for Mitigating Security Risks in Cloud

Tajinder Kumar[1], **Purushottam Sharma**[2,*], **Xiaochun Cheng**[3,*], **Sachin Lalar**[4], **Shubham Kumar**[5] and **Sandhya Bansal**[6]

[1]Computer Science & Engineering Department, Jai Parkash Mukand Lal Innovative Engineering & Technology Institute, Radaur, 135133, Haryana, India
[2]School of Computer Science & Engineering, Galgotias University, Greater Noida, 203201, Uttar Pradesh, India
[3]Computer Science Department, Bay Campus Fabian Way, Swansea University, Swansea, SA1 8EN, UK
[4]Department of Engineering and Technology, Gurugram University, Gurugram, 122003, India
[5]Presidency School of Computer Science, Presidency University, Bangalore, 560089, India
[6]Maharishi Markendeshwar Engineering College, Maharishi Markandeshwar (Deemed to be) University, Mullana, Ambala, 133203, India

*Corresponding Authors: Purushottam Sharma. Email: purushottam@galgotiasuniversity.edu.in;
Xiaochun Cheng. Email: xiaochun.cheng@swansea.ac.uk

Received: 11 November 2024; Accepted: 10 February 2025; Published: 26 March 2025

ABSTRACT: With cloud computing, large chunks of data can be handled at a small cost. However, there are some reservations regarding the security and privacy of cloud data stored. For solving these issues and enhancing cloud computing security, this research provides a Three-Layered Security Access model (TLSA) aligned to an intrusion detection mechanism, access control mechanism, and data encryption system. The TLSA underlines the need for the protection of sensitive data. This proposed approach starts with Layer 1 data encryption using the Advanced Encryption Standard (AES). For data transfer and storage, this encryption guarantees the data's authenticity and secrecy. Surprisingly, the solution employs the AES encryption algorithm to secure essential data before storing them in the Cloud to minimize unauthorized access. Role-based access control (RBAC) implements the second strategic level, which ensures specific personnel access certain data and resources. In RBAC, each user is allowed a specific role and Permission. This implies that permitted users can access some data stored in the Cloud. This layer assists in filtering granular access to data, reducing the risk that undesired data will be discovered during the process. Layer 3 deals with intrusion detection systems (IDS), which detect and quickly deal with malicious actions and intrusion attempts. The proposed TLSA security model of e-commerce includes conventional levels of security, such as encryption and access control, and encloses an insight intrusion detection system. This method offers integrated solutions for most typical security issues of cloud computing, including data secrecy, method of access, and threats. An extensive performance test was carried out to confirm the efficiency of the proposed three-tier security method. Comparisons have been made with state-of-art techniques, including DES, RSA, and DUAL-RSA, keeping into account Accuracy, QILV, F-Measure, Sensitivity, MSE, PSNR, SSIM, and computation time, encryption time, and decryption time. The proposed TLSA method provides an accuracy of 89.23%, F-Measure of 0.876, and SSIM of 0.8564 at a computation time of 5.7 s. A comparison with existing methods shows the better performance of the proposed method, thus confirming the enhanced ability to address security issues in cloud computing.

KEYWORDS: Cloud security: data encryption; AES; access control; intrusion detection systems (IDS); role-based access control (RBAC)

1 Introduction

As regards conventional local storage, cloud computing provides greater flexibility, affordability, and access to data. The character of data storage in cloud computing differs is unique since it does not involve the use of major facilities but can be charged based on the volume of use. Moreover, the Cloud information can be accessed anywhere, making the system more flexible and proficient for traversing divisional teams. The improved reliability of Cloud data depends on several methods, including the use of encryption levels and authorized restrictions. It not only helps to protect from various threats such as data breaches, malware attacks, and several unauthorized access attempts, but it also enhances the system's immunity against virus attacks. The biggest threats to cloud-stored data are the threat of access by unauthorized personnel, data leakage, and data loss. The proposed TLSA model mitigates these challenges by providing a layered security wall comprising encryption mechanisms and intrusion detection systems to deal with these threats. The three-tier strategy improves scalability and redundancy by integrating the concept of redundancy and fault tolerance in improving cloud infrastructure [1].

Moreover, integrating RBAC enhances access management's usefulness as it accurately controls user rights. In the age where many industrial systems are getting connected to the Internet, they are prone to many security threats. Security devices such as intrusion detection systems (IDSs) are becoming important in defending industrial infrastructures against identified malicious activities. Several studies on IDS that integrated signature-based and anomaly-based systems showed that the latter was more efficient [2]. When IDS is implemented at the third network layer, it is feasible to identify and prevent dangerous activities. The three-tier strategy also promotes compliance with safety regulations and laws in the workplace. It can show that an organization is committed to protecting an individual's data and following policy regulation standards prescribed by specific industry bodies by having several layers of protection.

1.1 Overview of Cloud Computing and Its Security Challenges

Cloud computing allows businesses to rapidly and smoothly get the right of entry to computing assets that include networks, applications, services, servers, and storage. Nevertheless, the significant advantages of cloud computing are scalability, cost-effectiveness, flexibility, and others. These concepts lead to security issues that need to be resolved to protect the privacy integrity and availability of resources and data [3].

- Data Protection and Privacy: This poses a challenge in cloud computing due to the latent risks associated with unauthorized access, data breaches, or data loss when data is managed on remote servers managed by cloud service providers (CSP).
- Identity and Access Management: Managing security requires managing user identities and restricting access to cloud resources. Reducing the risk of illegitimate access and unauthorized privilege escalation requires sound identity and access management best practices, such as strong user authentication, flexible RBAC, and constant access auditing [4].
- Data Encryption: To improve security and reduce the chances of being accessed and attacked by unauthorized persons, data is encrypted and stored in the Cloud. However, its usage has certain drawbacks—sometimes, it is complicated to regulate encryption keys and guarantee safe representations of keys.
- Cloud Application Security: It is important to avoid threats that attackers can exploit. It is best to use tight means of authentication and authorization in cloud applications to avoid cases of illegitimate access [5].

To address these cloud security concerns, organizations must complete their cloud security initiatives with the risk assessment and management approach. As cloud computing progresses, tackling the problem of cloud security and promoting a safe environment demands integration between enterprises and cloud service providers.

1.2 Importance of Secure Algorithms for Cloud Security

For organizations to ensure their measures work, secure algorithms must first be in place. From the study of literature, the following main arguments underline the importance of current secure algorithms for cloud security:

- Confidentiality and Data Protection: The secure technologies of Elliptic Curve Cryptography, RSA, and Advanced Encryption Standard (AES) can be used to show the importance of data security in cloud computing [6].
- Authentication and Access Control: Other means to verify the identity of the users, and get access to the cloud services via tight measures include secure protocols such as Transport Layer Security (TLS) and digital signatures [7].
- Key Management: Due to secure algorithms, activities that include key creation, distribution, storage, and revocation are easily simplified. According to papers, intense key management is required to ensure encryption keys are processed safely in the Cloud and prevent unauthorized access [8].
- Intrusion Detection and Prevention: IDS/IPS employs strong algorithms to detect and prevent security threats or threats in the cloud environment [9].
- Performance and Efficiency: Thus, aspects such as the performance and efficacy of present-day secure algorithms in cloud systems are more relevant. It also includes performance, computational complexity, encryption/decryption time, and resources [10].

1.3 Data Integrity and Privacy in Cloud Computing

Maintenance of data quality, consistency, and reliability plays a significant role in cloud computing. To maintain data integrity, issues such as unwanted modifications, tampering, and value checking, must be considered. Implement certain security features like authentication, authorization, and auditing to guard data integrity and privacy in cloud resources [9–12]. In case, the sharing of resources is important, data privacy takes a serious hit in the cloud-computing environment. Accessing of sensitive information by unwanted entities is prevented by the execution of technologies such as data anonymization, data masking, and data encryption [13–15]. SSL/TLS and all the other data transfer security protocols are a must if data is to be protected from leakage and interference during transfer. Legal requirements are crucial while dealing with data to ensure the correct procedure is followed. This helps to meet the set legal measures such as GDPR or HIPAA compliance. Cloud providers must retain transparency and accountability to provide insight into data processing, and security measures the customers implement. Measures of data backup and data recovery are important to maintain the data's integrity and security. In case of data loss [16] or system failure, it's crucial to maintain business operations [17,18].

1.4 Purpose and Objectives of the Paper

How companies handle data is significant and has been transformed through cloud computing. Cloud computing technology is an excellent approach to delivering the highest scalable, cost-efficient, and flexible computing with the help of remote servers and diffused computing ingredients. In order to maintain the confidentiality, integrity, and availability of information, many security aspects concerning the use of cloud services [19] have to be considered. One of the main problems of cloud computing is data security. The threat of data vulnerability has become a significant problem in cloud computing. Cloud providers share security certifications, security incident response efficiency, and open security policies, and they have a reasonably significant role in implementing the security of their platforms and services [19,20]. Although cloud computing offers advantages, like being cost-efficient, productive, and reliable, organizations must always watch out for threats and protect their data and systems [21,22]. However, to make the best use of cloud technology's opportunities, businesses must recognize several features specific to the cloud environment and provide reliable protection for sensitive information [23,24].

The subsequent sections encompass the remaining content of this paper: This paper examines the security research in the context of cloud computing in Section 2. Section 3 describes the nature of the dataset utilized in the study and the feature extraction and pre-processing step of the presented method. The following section describes the prediction method of stroke. Section 5 performs the comparative study of the evaluation parameters of the proposed method against the baseline methods. The last section contains the conclusion.

2 Related Work

Personal computing and organizational use, and the collaborative network and hybrid cloud distribution model, as depicted in Fig. 1, are demonstrated. Attention is paid to access control methods, including mandatory access control, role-based access control, and discretionary access control [25]. The proposed approach is used for availability and integrity—to provide users with easy access and efficient performance execution while ensuring confidentiality.

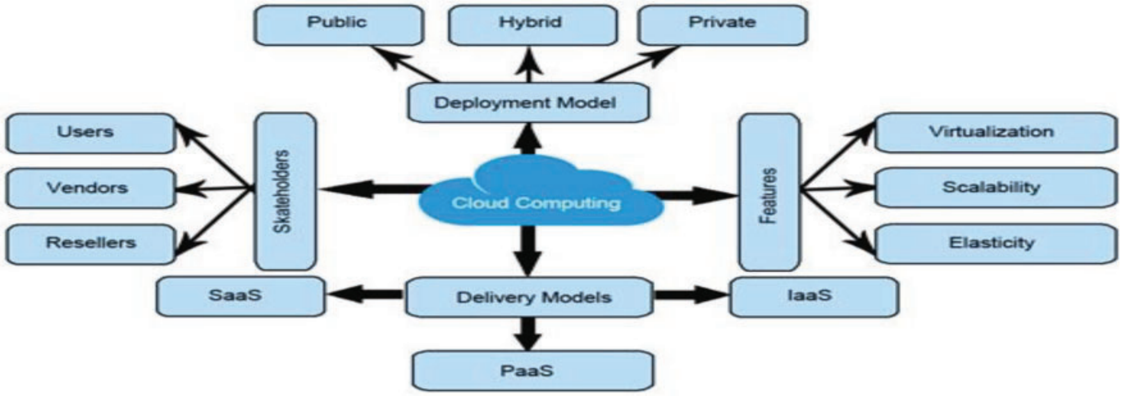


Figure 1: Overview of cloud computing

Fig. 1 explains the Cloud Concept by analyzing the Overview of Cloud Computing. It concentrates on providing computing resources, especially servers, storage, and applications, as services are accessible through the Internet. Three approaches to cloud computing, known as IaaS, PaaS, and SaaS service delivery models, explain cloud computing. Authors in [25] propose a revolutionary Cryptographic Role Engineering (CRE) approach to solve security issues in organizations. In contrast with the previous encryption models that employ El-Gamal, Baillier, and Benaloh techniques in [26], the introduced model utilizes SPHE for both the encryption of data and the computation of the encrypted data, enhancing security. In a cloud context that improves access control and database privacy, SPHE is implemented in conjunction with a role-based user policy. This model is deployed in an Amazon Web Service environment of Elastic Beanstalks (EB). In [27], authors take a closer look at cloud computing opportunities besides Virtual Reality, Augmented Reality, and Metaverse. As the requirement for computation rises, the data owners shift towards the remote server to obtain computation. However, having multiple tenants on the same Cloud brings issues with access to unauthorized personnel and probing of the networks. H-IDS is the host-based intrusion detection system proposed in [28] to protect virtual machines in cloud computing. The NSL-KDD dataset is used to train and test the model, and simulation proves satisfactory for approximately 97.51% of attack detection against normal states.

In this work [29], the authors present the IBET model of Identity-Based Encryption Transformation to address the key difficulty of sharing encrypted data with a more extensive audience than the intended recipients. This IBET model perfectly integrates Identity Based Encryption (IBE) and Broadcast Encryption (IBBE) methods. For addressing data security in cloud-based applications, the paper presents a proposed cryptographic model called Autonomous Path Identity-Based Broadcast Proxy Re-Encryption (APIB-BPRE) but in a different setting. Evaluating and comparing APIB-BPRE reveals that it is effectively applied to practical engineering problems. In [30], authors enhance the security of data stored in cloud storage systems using trust models in conjunction with cryptographic role-based access control (RBAC) schemes. Considering inheritance and role hierarchy, these trust models enable owners and roles to assess each user's and role's trustworthiness inside the RBAC system.

The research gap lacks a holistic security framework that addresses data encryption, access control, and intrusion detection in cloud computing environments. While some papers discuss Role-Based Access Control (RBAC), there is a research gap in the comprehensive coverage of user roles and permissions, especially in dynamically changing cloud environments. The proposed research addresses the challenges of evolving user roles and permissions in the realm of cloud security. This research introduces a comprehensive three-tier security framework that focuses on enhancing cloud computing security through the integration of Advanced Encryption Standard (AES)-based encryption, Role-Based Access Control (RBAC), and Intrusion Detection Systems (IDS). Motivated by the imperative need to fortify cloud data security, our proposed method employs AES as the cornerstone for first-layer data encryption.

3 Working of Proposed Method TLSA

3.1 Components and Layers of the Algorithm

3.1.1 First Layer: Data Encryption

AES strengthens data authenticity and secrecy, meaning the encryption algorithm is computationally secure from brute force attacks. These are encryption features: data at rest and data in transit so that the wrong people cannot intercept or modify the various forms of information. AES also opts for fast data encryption and decryption rate, which is important for large-scale cloud applications. Its features encrypt data at rest and in transit, making it impossible for the wrong force to breach the information. This layer focuses on protecting data using encryption methods such as asymmetric encryption (such as RSA) or symmetric encryption (such as AES). This technology converts the original data into an encrypted version while maintaining confidentiality through mathematical operations and encryption algorithms. Efficient key management is essential for secure encryption. The algorithm includes mechanisms to generate encryption keys and securely transmit them to authorized parties. Key management strategies are employed to safeguard the integrity and confidentiality of cryptographic keys.

3.1.2 Second Layer: Access Control

Depending on their roles and responsibilities, the user can access cloud resources through this layer, which controls and manages this access. RBAC provides a versatile and scalable method for creating and implementing access controls. Depending on user roles and the permissions associated with those roles, the algorithm combines RBAC techniques to grant or restrict access privileges. Trusted authentication techniques such as username/password, biometric authentication, or multi-factor authentication are implemented to confirm the identity of users accessing the cloud environment. Authorization techniques, such as attribute-based access control (ABAC) and access control lists (ACL), define the extent of access granted to authorized users.

3.1.3 Third Layer: Intrusion Detection and Prevention

Before presenting the proposed methods, the paper conducts a comprehensive review of related work in the field of Intrusion Detection. Some concerns for traditional machine learning and deep learning models are the lack of labeled data and the disparity in data distribution that characterizes ICNs. DTL is a potential solution to transfer knowledge from pre-trained models to target tasks with little training data. The paper introduces IDS types (anomaly, signature, and hybrid) and overviews potential issues that arise during their application to ICNs. A schematic block diagram of the proposed method, the Three-Layered Security Access (TLSA) Model, is shown in Fig. 2.

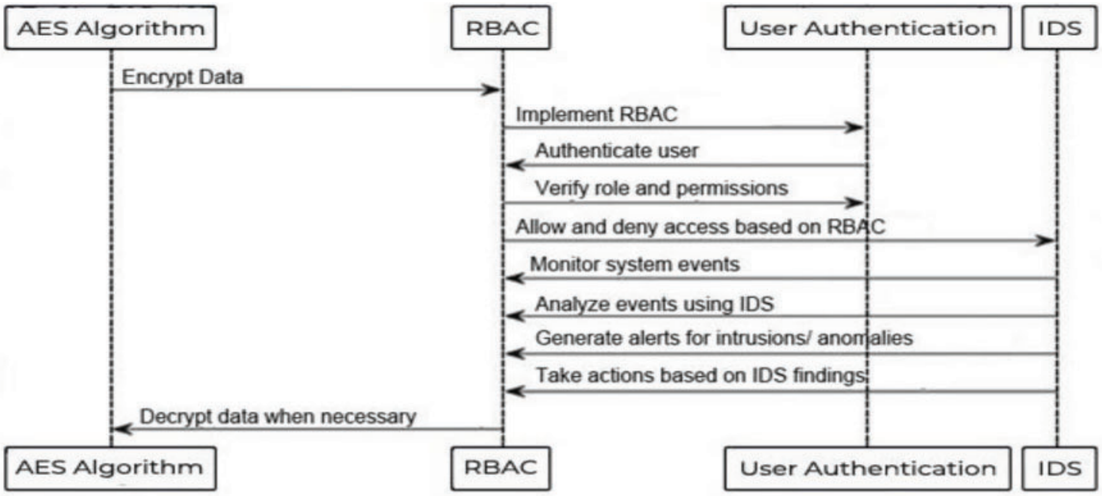


Figure 2: Block diagram of TLSA

3.2 Algorithm of Proposed Method—TLSA

3.2.1 First Layer: Data Encryption

Encryption Techniques and Algorithms Used: This layer employs the AES encryption algorithm, as shown in Fig. 3, a symmetric key cryptographic algorithm widely used for secure data encryption. It starts with adding the first round key, known as the round key (0). Much about a decision point, it looks into a question as to whether the current round, 'i', is equal to the total number of rounds, "Nr." If not, the encryption process will take three steps. However, despite the Data Encryption Standard (DES) comprising 16 rounds of operation, four primary operations bear mentioning: sub bytes, shift rows, mix columns, and ultimately, adding the round key. If yes, only two operations are performed, sub-byte and shift rows, and then the last round key (round key (Nr)) is added. The final step is the output of the ciphertext, often referred to as the encrypted result. The complete Algorithmic details are given in Appendix A as 3.3.1.

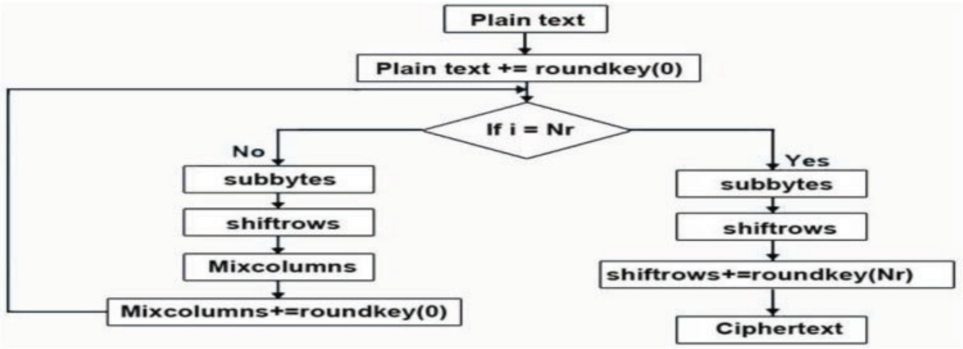


Figure 3: Advanced encryption system

3.2.2 Second Layer: Access Control—RBAC (Role-Based Access Control)

Fig. 4 illustrates Role-Based Access Control (RBAC) in access control. The process ensures that only those employees with the right roles will request the proper Permission to connect to the system.

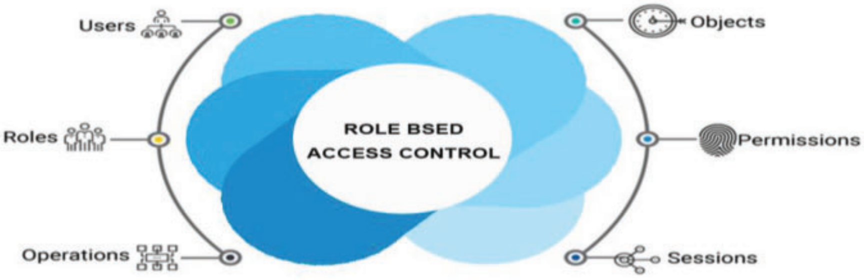


Figure 4: Second layer: access control—role-based access control (RBAC)

In this model, the user authenticates to affirm his identity and authorize his or her access with roles. It then verifies the user's position and his or her authorization level. This enables to allow or deny him or her, the given resource. Such an approach forms the system to ensure that only authorized users can run jobs and are considered appropriate by their status. RBAC is a broadly used access control model that grants users permissions based on their assigned roles, as shown in Fig. 4. The RBAC implementation includes multiple steps, as shown in Appendix A, which are 3.3.2.

3.2.3 Third Layer: Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS) Implementation: IDS are security mechanisms that detect and respond to potential intrusions or malicious activities, as shown in Fig. 5. The IDS implementation includes multiple steps, as shown in Appendix A, such as 3.3.3.

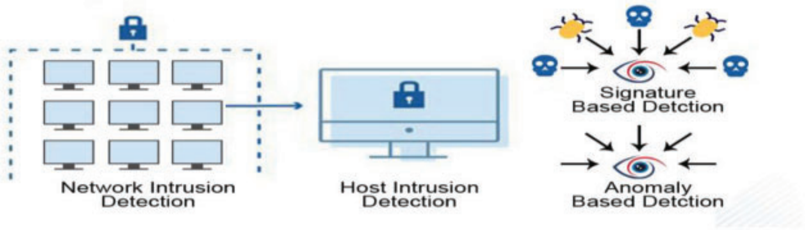


Figure 5: Third layer: intrusion detection systems (IDS)

4 Evaluation and Performance Analysis

4.1 Experimental Setup

The experimental setup specification used for implementation and performance evaluation configured to specific parameters/conditions given in Section 4.1. Experimental setup configuration given in Table 1.

Table 1: Detail of experimental setup

Category	Details
Hardware specifications	**CPU:** Intel Xeon E5-2678 v3 (12 Cores, 24 Threads) **RAM:** 32 GB DDR4 **Storage:** 1 TB SSD (NVMe) for faster I/O operations
Cloud platform setup	**Platform:** AWS EC2 Instances **Region:** US-East (N. Virginia) **Instance type:** m5.xlarge (4 vCPUs, 16 GB RAM) **Virtualization:** Xen/AMI-based virtualization **Security configurations:** VPC, Subnets, Security Groups, and IAM Roles for access control

4.2 Performance Metrics

The performance metrics are often used to evaluate the quality and effectiveness of various algorithms. The following performance metrics are used in this paper:

- SSIM (Structural Similarity Index): It measures the structural similarity between images. It assesses how well the structural information in an image is preserved after processing.

$$SSIM\,(x, y) = \frac{(2 * \sigma xy + C2) * (2 * \mu x * \mu y + C1)}{(\mu x^2 + \mu y^2 + C1) * (\sigma x^2 + \sigma y^2 + C2)} \tag{1}$$

In (1), x, y are the input and processed images, μx, μy are the means of x and y, Standard deviations of x and y is σx, σy, Covariance of x and y is σxy, C1 and C2 are constants to stabilize the division.

- QILV (Quality Index of Luminance and Visibility): QILV is a metric that assesses the quality and visibility of important features in an image.

$$QILV\,(x, y) = \frac{(\mu x * \mu y + k)}{(\mu x^2 + \mu y^2 + k)} \tag{2}$$

In (2), x and y are the input and processed images, means of x and y are μx and μy, k is a constant.

- Precision (%): Precision is a metric used in classification tasks. The percentage of total expected positive cases is measured precisely by the percentage of expected positive cases.

$$\text{Precision} = (\text{True Positives})/(\text{False Positives} + \text{True Positives}) \tag{3}$$

- Sensitivity (%): Sensitivity is the ratio between accurately predicted positive cases and the total number of positive cases that occurred.

$$\text{Sensitivity} = (\text{True Positives})/(\text{True Positives} + \text{False Negatives}) \tag{4}$$

- F-Measure (%): The harmonic mean of Sensitivity and precision is known as F-measure. Classification functions use it to find a trade-off between recall and precision.

$$\text{F} - \text{Measure} = (2 * \text{Precision} * \text{Sensitivity})/(\text{Precision} + \text{Sensitivity}) \tag{5}$$

- PSNR (Peak Signal-to-Noise Ratio): PSNR is a statistic used to evaluate the quality of image noise reduction or compression. Calculates the ratio between the highest possible signal strength and the strength of the signal causing noise.

$$PSNR = 10 * \log 10\left(\frac{(Max^2)}{MSE}\right) \tag{6}$$

In (6), the maximum pixel value, typically 255 for 8-bit images, is denoted by *Max*.

- MSE (Mean Squared Error): MSE measures the mean squared difference between the pixel values of the original and processed image.

$$\text{MSE} = (1/\text{N}) * \Sigma(\text{x} - \text{y})^2 \tag{7}$$

In (7), the total number of pixels is N. The pixel values of the original and processed image at a given location are represented by x and y.

- Accuracy (%): A classification statistic called accuracy calculates the proportion of accurately predicted occurrences to all instances.

$$\text{Accuracy} = (\text{Correct Predictions})/(\text{Total Predictions}) \tag{8}$$

- Computation Time (s): Computation time measures how long a specific operation or algorithm takes to process the data, typically in seconds.

5 Performance Analysis of the Algorithm

Fig. 6 shows the time it takes to perform encryption using different encryption algorithms for various key sizes. As the key size increases to 10 bits, TLSA and DES remain the fastest, but their encryption times have increased significantly. RSA and DUAL-RSA, asymmetric encryption algorithms, show a much more significant increase in encryption time than symmetric algorithms like TLSA and DES. TLSA, using a 50-bit key, completes the encryption process in 2.86 s, while DES takes 2.93 s, DUAL-RSA 3.23 s, and RSA 4.87 s.

The comparison reveals that symmetric encryption methods such as DES and TLSA outperform asymmetric encryption algorithms such as DUAL-RSA and RSA, especially with large key sizes. In addition, the encryption time of RSA and DUAL-RSA is significantly affected by the key size, making them slower than TLSA and DES for larger key sizes.

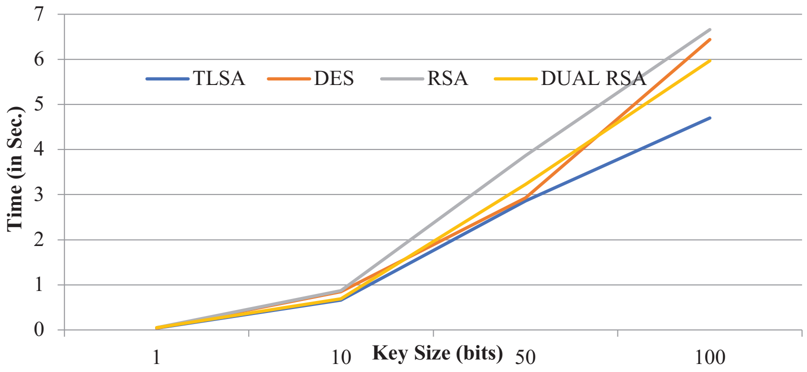


Figure 6: Encryption time comparison

For minimum key sizes, TLSA—Proposed method, DES, RSA, and DUAL-RSA all decrypt the data in Fig. 7 in about 0.05 s, which means they are all fast and have comparable decryption speeds. Although TLSA and DES are relatively fast compared to RSA and DUAL-RSA, decryption times increase as the key size approaches 10 bits. TLSA and DUAL-RSA emerged as the slowest decryption options, with TLSA requiring 2.89 s and DUAL-RSA 5.07 s. RSA follows with 4.99 s, but DES is the fastest with 1.98 s. Even with a key size of 100 bits, TLSA decryption is still somewhat slow, taking 4.7 s. The comparison indicates that symmetric encryption algorithms like DES and TLSA tend to be faster for decryption than asymmetric encryption algorithms like RSA DUAL-RSA [31]. As the key size increases, the decryption times for all algorithms also increase, but the relative performance remains consistent.

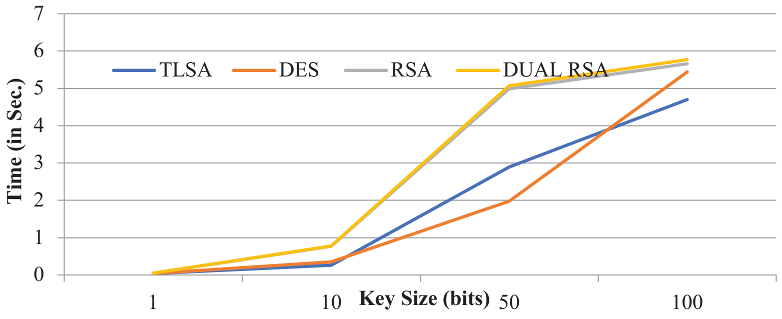


Figure 7: Decryption time comparison

To compare the performance of different encryption methods, several parameters have been employed to measure different aspects of encryption from [32,33]. F-Measure considers both precision and recall to measure encryption reliability where there is an imbalance of the data. Sensitivity evaluates the encryption method and how well it identifies all the important data that needs to be encrypted. MSE measures the level of encrypted and original data. According to the current research, the data quality after decryption using PSNR is studied. Validating SSIM focuses on checking perceived similarity in encryption output. Computation Time gives the time taken to perform the encryption in a given method.

Table 2 compares different encryption methods based on various evaluation metrics. DUAL-RSA performs well across various metrics, balancing quality, accuracy, and speed well. TLSA and DES also perform reasonably well, while RSA lags in quality and accuracy. Table 3 compares access control models based on layer 2 of the proposed method.

Table 2: Performance analysis of encryption methods

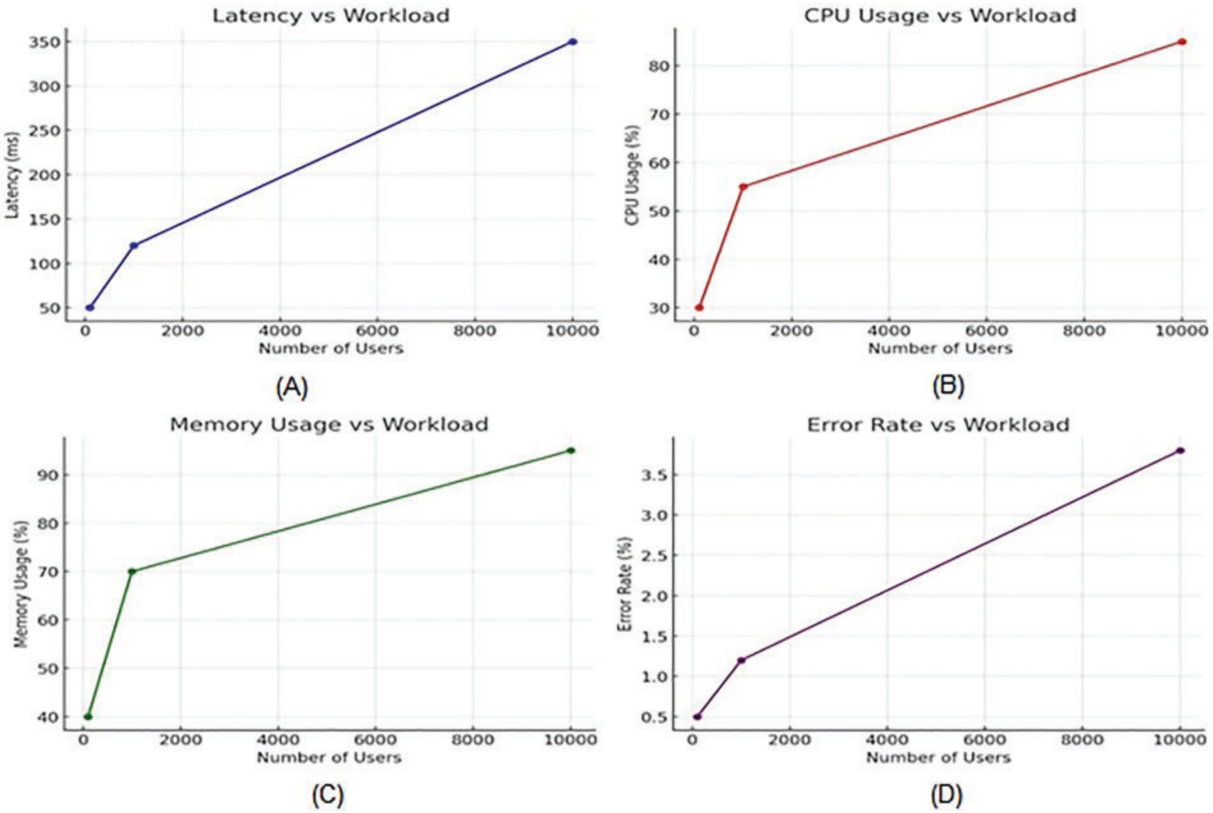
Methods	Accuracy (%)	QILV	F-Measure (%)	Sensitivity (%)	MAP	PSNR (dB)	SSIM	Computation time (s)
DES	92.55	0.8456	0.879	0.865	3.4	66.52	0.8431	5.2
TLSA	89.23	0.9042	0.876	0.853	3.5	65.32	0.8564	5.7
RSA	82.65	0.8992	0.848	0.826	3.9	62.12	0.8241	6.5
DUAL-RSA	93.48	0.9098	0.912	0.894	3.2	68.23	0.8896	3.5

Table 3: Comparison of access control models

Access control model	Access granted	Access permissions	Security implications	Examples
DAC	Based on the user's identification	The access control list defines permissions.	Simple to attack and exploit	Old versions of Windows/UNIX
MAC	System administrator	The administrator has complete control over altering an object's and the user's security clearance.	Vulnerable to exploit	Military applications
RBAC	Depending on the role that a system administrator has allocated to a user	An administrator grants a user a position with predetermined system privileges and rights. Once given a role, a user can only access system resources and carry out the tasks listed in the assigned role. Additionally, the system administrator centrally oversees the tasks assigned to users.	Compared to the MAC and DAC variants, they are more secure and durable	Microsoft Azure, Google Cloud, Most of the enterprise applications

6 Scalability Testing

The scalability test of the Triple Layered Approach for Mitigating Security Risks in the Cloud shows how the system responds to workloads of 100, 1000, and 10,000 users in Fig. 8, which shows trends. From 500 users, latency increases steeply from 50 to 350 ms in Fig. 8a, suggesting system bottlenecks for higher workload levels. It becomes apparent that computational requirements have increased from 100 to 10,000 users, from consuming 30% of the CPU to 85% in Fig. 8b. Likewise, memory usage in Fig. 8c increases from 40% to 95% due to the growing need for memory-bound operations in the system. The error rate also increases from 0.5% at 100 users to 3.8% at 10,000 users in Fig. 8d, which indicates that the system's reliability is affected when the system is under pressure. It also underlines current trends to focus on resource management in general and performance in particular, as well as the determination of system thresholds for large-scale workloads.



(A)

(B)

(C)

(D)

Figure 8: Scalability test of the triple layered approach for mitigating security risks for 100, 1000, and 10,000 users

7 Conclusion

Focusing on data encryption, access management, and intrusion detection systems (IDS), a three-layer secure technique is proposed for cloud security in this paper. This technology uses IDS for intrusion detection, RBAC to control access, and Advanced Encryption Standard (AES) to encrypt data. AES encryption protects Sensitive data during storage and transmission in the Cloud. AES, a well-known and trusted encryption technology, maintains the security and integrity of data while reducing the possibility of illegal access. RBAC, the second layer of the algorithm, provides a systematic method for access control. RBAC secures specific data in the Cloud by allocating roles and specifying pertinent permissions, limiting access to only permitted ones. This layer improves security by imposing stringent access controls based on roles and responsibilities. The intrusion detection system (IDS), which examines network traffic and system records for security breaches and malicious activities, is the attribute of the third layer of the algorithm. This three-layer security algorithm deals with security concerns related to cloud computing, successfully guarding against hostile actions, illegal access, and data breaches.

When the proposed method, TLSA, was compared against the current approaches—DES, RSA, and DUAL-RSA—it produced impressive results on several performance criteria. Notably, the suggested approach performs better in accuracy, Sensitivity, F-Measure, MSE, signal maximum, QILV, SSIM, PSNR, computation time, and accuracy. The proposed method performed the best, DUAL-RSA, and has the greatest SSIM of 0.8896, indicating that the encoded data's structural similarity is optimally preserved. Moreover, it has the greatest QILV (0.9098), demonstrating its exceptional visual coding performance. The DUAL-RSA's dependability is further supported by precision and sensitivity tests, which show that it has the lowest rates of false positives (87.9%) and false negatives (89.4%), with the greatest F-Measure (91.2%). Together with its efficiency, our suggested technique has an accuracy rate of 93.48%, which makes it a complete and practical solution for cloud computing security. A weakness of the proposed method of enhancing data security is that the DUAL-RSA algorithm forms the core part of the three-layer security system. As can be observed, the average response time of DUAL-RSA is higher than that of TLSA and DES. However, its computation complexity makes its overhead relatively high for a resource-constrained environment. Moreover, the choice of one particular cryptographic technique could restrain innovation and portability to modern cryptographical methods or risks. This could become troublesome, especially in environments where the cloud setup is dynamic, scalable, and agile.

Acknowledgement: This work was supported by UKRI EPSRC Grant funded Doctoral Training Centre at Swansea University, through PhD project RS718 on Explainable AI. Authors also have been supported by UKRI EPSRC Grant EP/W020408/1 Project SPRITE+ 2: The Security, Privacy, Identity and Trust Engagement Network plus (phase 2). The authors acknowledge above financial support.

Funding Statement: The authors have been funded by UKRI EPSRC Grant EP/W020408/1 Project SPRITE+ 2: The Security, Privacy, Identity and Trust Engagement Network plus (phase 2) for this study. The authors also have been funded by PhD project RS718 on Explainable AI through UKRI EPSRC Grant funded Doctoral Training Centre at Swansea University.

Author Contributions: Conceptualization, Tajinder Kumar and Purushottam Sharma; methodology, Xiaochun Cheng; software, Sachin Lalar; validation, Xiaochun Cheng, Purushottam Sharma and Shubham Kumar; formal analysis, Tajinder Kumar; investigation, Tajinder Kumar and Purushottam Sharma; resources, Xiaochun Cheng; data curation, Purushottam Sharma; writing—original draft preparation, Tajinder Kumar; writing—review and editing, Sachin Lalar and Purushottam Sharma; visualization, Sandhya Bansal; supervision, Sandhya Bansal; project administration, Purushottam Sharma; funding acquisition, Xiaochun Cheng. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The studies are conducted on already available data and materials for which consent is not required.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

Appendix A

Algorithms details:

Section 3.3.1:

Advanced Encryption Algorithm: The round keys are generated form the initial encryption key using the Key Expansion algorithm of AES. Initial round which take the plaintext and an XOR is done between the plaintext and first round key. In each round it has SubBytes (byte substitution using the S-Box), ShiftRows (cyclic shifts in row wise), MixColumns (column mixing for diffusion) and AddRoundKey (XOR with round key).

Key Expansion Algorithm: KeyExpansion expands the key from its initial size (128, 192, or 256 bits). SubWord, RotWord, and Rcon are helper functions used for transformations and round constant application.

```
$KeyExpansion\ (Key, RoundKey):$
  $Input: Key\ (initial\ encryption\ key)$
```

(Continued)

```
 $Output: RoundKey\ (set\ of\ round\ keys)$
 $RoundKey\,[0] = Key$
 $for\ j = 1\ to\ Nr:$
   $T = RoundKey\,[j-1]$
   $if\ j\ mod\ Nk == 0:$
     $T = SubWord\,(RotWord\,(T))\ \ XOR\ Rcon\,[j/Nk]$
   $else\ if\ Nk > 6\ and\ j\ mod\ Nk == 4:$
     $T = SubWord\,(T)$
   $RoundKey\,[j] = RoundKey\,[j-Nk]\ XOR\ T$
```

Initial Round Algorithm: Initial Round performs an initial XOR operation between the first round key and the input data block.

```
$InitialRound\,(State, RoundKey):$
 $Input: State\ (input\ data\ block), RoundKey\ (first\ round\ key)$
 $Output: State\ (modified\ data\ block)$
 $State = AddRoundKey\,(State, RoundKey)$
```

SubBytes Algorithm: SubBytes performs a byte-level substitution operation using a predefined substitution table (S-Box). SubByte(byte) replaces a byte with a corresponding value from the S-Box.

```
$SubBytes\,(State):$
 $Input: State\ (input\ data\ block)$
 $Output: State\ (modified\ data\ block)$
 $for\ each\ byte\ in\ State:$
     $byte = SubByte\,(byte)$
```

ShiftRows Algorithm: ShiftRows cyclically shifts the bytes in each row of the data block to the left based on their row index.

```
$ShiftRows(State):$
      $Input: State\ (input\ data\ block)$
      $Output: State\ (modified\ data\ block)$
      $for\ each\ row\ in\ State:$
          $row = ShiftRow\,(row)$
```

MixColumns Algorithm: MixColumns Algorithm performs a matrix multiplication operation on each column of the data block using a predefined matrix.

```
$MixColumns\,(State):$
 $Input: State\ (input\ data\ block)$
 $Output: State\ (modified\ data\ block)$
```

(Continued)

```
 for each column in State:
   column = MixColumn (column)
```

AddRoundKey Algorithm: AddRoundKey performs an XOR operation between the round key and the current state.

```
AddRoundKey (State, RoundKey) :
  Input: State (input data block) , RoundKey (round key)
  Output: State (modified data block)
  State = State XOR RoundKey
  Round (State, RoundKey) :
  Input: State (input data block) , RoundKey (round key)
  Output: State (modified data block)
  SubBytes (State)
  ShiftRows (State)
  MixColumns (State)
  AddRoundKey (State, RoundKey)
```

```
AES (Plaintext, Key) :
  Input: Plaintext (input data block) , Key (encryption key)
  Output: Ciphertext (encrypted data block)
  1. KeyExpansion (Key, RoundKey)
  2. State = Plaintext
  3. InitialRound (State, RoundKey [0])
  4. for $i = 1$ to $Nr - 1$:
    Round (State, RoundKey [i])
  5. Round (State, RoundKey [Nr])
  6. Ciphertext = State
```

Section 3.3.2

User and Role Management: AssignRole(user, Roles) is a helper function that assigns a role to a user based on defined rules and policies. It assigns roles to users based on predefined rules and policies.

```
AssignRoles(Users, Roles):
            Input: Users (set of users) , Roles (set of roles)
            Output: UserRoles (mapping of users to roles)
                            for each user in Users:
            UserRoles[user] = AssignRole(user, Roles)
```

1. Permission Assignment: AssignPermission(role, Permissions) is a helper function that assigns specific permissions to a role. It assigns permissions to roles based on predefined rules and policies.

```
$AssignPermissions(Roles, Permissions):$
        $Input: Roles\ (set\ of\ roles), Permissions\ (set\ of\ permissions)$
        $Output: RolePermissions\ (mapping\ of\ roles\ to\ permissions)$
          $for\ each\ role\ in\ Roles:$
        $RolePermissions[role] = AssignPermission(role, Permissions)$
```

2. Authorization: Authorization relies on the mappings from UserRoles and RolePermissions. It verifies whether a user has the necessary permissions to access a specific resource. Authorization relies on the mappings from UserRoles and RolePermissions. Authentication Mechanisms are used to verify the identity of the user prior to granting access.

```
$IsAuthorized\ (User, Resource, Permission):$
      $Input:\ User, Resource, Permission\ (requested\ permission)$
      $Output: Boolean\ (true\ if\ the\ user\ is\ authorized, false\ otherwise)$
      $role = UserRoles[User]$
      $permissions = RolePermissions[role]$
      $if\ Permission\ in\ permissions:$
          $return\ true$
$else:$
  $return\ false$
```

```
$AuthenticateUser\ (User, Credentials):$
    $Input: User\ (user\ requesting\ access), Credentials\ (user's\ credentials)$
    $Output: Boolean\ (true\ if\ authentication\ succeeds, false\ otherwise)$
   $UserRoles[user] = AssignRole\ (user, Roles)$
   $RolePermissions[role] = AssignPermission\ (role, Permissions)$
   $IsAutorized\ (User, Resource, Permission) = (Permission\ \in\ RolePermissions[UserRoles[user]])$
   $AuthenticateUser\ (User, Credentials) = (Authentication\ succeeds)$
```

Section 3.3.3:

System Monitoring: It Monitors system events to detect potential intrusions or anomalies and generates alerts. AnalyzeEvents(Events) is called to analyze events and identify potential issues.

```
$MonitorSystem\ (Events):$
       $Input: Events\ (set\ of\ system\ events)$
       $Output: Alerts\ (set\ of\ generated\ alerts)$
       $//Perform\ system\ monitoring\ and\ analysis\ of\ events$
       $Alerts = AnalyzeEvents\ (Events)$
```

Event Analysis: It analyzes each system event to detect potential intrusions or anomalies and generates appropriate alerts. IsIntrusion(event) determines if a specific event indicates a potential intrusion or anomaly. GenerateAlert(event) creates an alert when an intrusion or anomaly is detected.

```
AnalyzeEvents (Events) :
           Input: Events (set of system events)
           Output: Alerts (set of generated alerts)
           Alerts = {}
           for each event in Events:
              if IsIntrusion (event) :
                 alert = GenerateAlert (event)
                 Alerts.add (alert)
          return Alerts
```

Triple Layer Secure Algorithm:

```
System:
  Input: User credentials, resource requests, system events
      Output: Authentication status, access control response, alerts for intrusions
1. User Database:
  Users = {
    "user1": {"password": "password1", "role": "admin", "permissions": ["read", "write"]} ,
    "user2": {"password": "password2", "role": "user", "permissions": ["read"]}
     }
2. AES Encryption and Decryption:
  AES_Encrypt (data, key) :
    Input: data (plaintext) , key (encryption key)
    Output: nonce, cipher, tag
    cipher_obj = AES.new (AES_MODE_A, key)
    nonce = cipher_obj.nonce
    cipher, tag = cipher_obj.encrypt (data)
    return nonce, cipher, tag
  AES_Decrypt (nonce, cipher, tag, key) :
    Input: nonce, cipher, tag, key
    Output: plaintext
    cipher_obj = AES.new (key, AES_MODE_A, nonce = nonce)
    plaintext = cipher_obj.decrypt_and_verify (cipher, tag)
    return plaintext
3. Intrusion Detection System (IDS) :
  AnalyzeEvents (event) :
    Input: event (system event)
    Output: Boolean (true if intrusion detected, false otherwise)
    if "attack" in event:
     return true
    else:
     return false
```

(Continued)

```
4. User Authentication:
  AuthenticateUser (username, password) :
    Input: username, password
    Output: Boolean, Message
    if username in Users and Users [username] ['password'] == password:
      return true, "Successful Login"
    else:
      return false, "Failed Login"
5. Resource Access Control:
  AccessResource (username, resource) :
    Input: username, resource
    Output: Message, Status Code
    if username in Users:
      user = Users [username]
      if "admin" in user [' role'] :
        return f"Admin has access to {resource} ", 200
      elifresource in user [' permissions'] :
        return f" {username} has access to {resource} ", 200
      else:
        return f"Access denied for {username} to {resource} ", 403
   else:
      return"User not found", 404
6. Flask Application:
  InitializeApp():
    key = GenerateRandomKey(16) #Generate AES encryption key
       StartFlaskApp() #Run the Flask application
```

References

1. Ghosh Ray I, Rahulamathavan Y, Rajarajan M. A new lightweight symmetric searchable encryption scheme for string identification. IEEE Trans Cloud Comput. 2020;8(3):672–84. doi:10.1109/TCC.2018.2820014.
2. Kheddar H, Himeur Y, Awad AI. Deep transfer learning for intrusion detection in industrial control networks: a comprehensive review. J Netw Comput Appl. 2023;220(10):103760. doi:10.1016/j.jnca.2023.103760.
3. Barnes R, Bhargavan K, Lipp B, Wood CA. Hybrid public key encryption. Internet Research Task Force (IRTF); 2022. doi:10.17487/RFC9180.
4. Mihailescu MI, Nita SL. A searchable encryption scheme with biometric authentication and authorization for cloud environments. Cryptography. 2022;6(1):8. doi:10.3390/cryptography6010008.
5. Bauspieß P, Kolberg J, Drozdowski P, Rathgeb C, Busch C. Privacy-preserving preselection for protected biometric identification using public-key encryption with keyword search. IEEE Trans Ind Inform.. 2023;19(5):6972–81. doi:10.1109/TII.2022.3199944.
6. Seth B, Dalal S, Jaglan V, Le DN, Mohan S, Srivastava G. Integrating encryption techniques for secure data storage in the cloud. Trans Emerging Tel Tech. 2022;33(4):e4108. doi:10.1002/ett.4108.
7. Khoda Parast F, Sindhav C, Nikam S, Izadi Yekta H, Kent KB, Hakak S. Cloud computing security: a survey of service-based models. Comput Secur. 2022;114(1):102580. doi:10.1016/j.cose.2021.102580.
8. Zheng T, Luo Y, Zhou T, Cai Z. Towards differential access control and privacy-preserving for secure media data sharing in the cloud. Comput Secur. 2022;113(1):102553. doi:10.1016/j.cose.2021.102553.

9. Khan JA. Role-based access control (RBAC) and attribute-based access control (ABAC). In: Improving security, privacy, and trust in cloud computing. Hershey, PA, USA: IGI Global; 2024. p. 113–26.

10. Rao YS, Prasad S, Bera S, Das AK, Susilo W. Boolean searchable attribute-based signcryption with search results self-verifiability mechanism for data storage and retrieval in clouds. IEEE Trans Serv Comput. 2024;17(4):1382–99. doi:10.1109/TSC.2023.3327816.

11. Chenam VB, Ali ST. A designated cloud server-based multi-user certificateless public key authenticated encryption with conjunctive keyword search against IKGA. Comput Stand Interfaces. 2022;81(4):103603. doi:10.1016/j.csi.2021.103603.

12. Singh N, Kumar J, Singh AK, Mohan A. Privacy-preserving multi-keyword hybrid search over encrypted data in cloud. J Ambient Intell Humaniz Comput. 2024;15(1):261–74. doi:10.1007/s12652-022-03889-8.

13. Kousalya A, Baik NK. Enhance cloud security and effectiveness using improved RSA-based RBAC with XACML technique. Int J Intell Netw. 2023;4(7):62–7. doi:10.1016/j.ijin.2023.03.003.

14. Joshi NS, Sambrekar KP. Privacy-preserving and ranked search using advanced multi-keyword scheme over the encrypted cloud environment. J Electr Syst. 2024;20(1s):353–65. doi:10.52783/jes.776.

15. Varri US, Mallick D, Das AK, Hossain MS, Park Y, Rodrigues JJPC. TL-ABKS: traceable and lightweight attribute-based keyword search in edge-cloud assisted IoT environment. Alex Eng J. 2024;107(101):757–69. doi:10.1016/j.aej.2024.09.030.

16. Rao KR, Ray IG, Asif W, Nayak A, Rajarajan M. R-PEKS: RBAC enabled PEKS for secure access of cloud data. IEEE Access. 2019;7:133274–89. doi:10.1109/ACCESS.2019.2941560.

17. Javadpour A, Ja'fari F, Taleb T, Benzaïd C, Bin Y, Zhao Y. Encryption as a service (EaaS): introducing the full-cloud-fog architecture for enhanced performance and security. IEEE Internet Things J. 2024;11(24):39744–66. doi:10.1109/JIOT.2024.3450192.

18. Verma G. Blockchain-based privacy preservation framework for healthcare data in cloud environment. J Exp Theor Artif Intell. 2024;36(1):147–60. doi:10.1080/0952813X.2022.2135611.

19. Hu X, Chang J, Ahmad T, Zhang F, Zhang Y. Identity-based integrity auditing scheme with sensitive information hiding for proxy-server-assisted cloud storage applications. IEEE Internet Things J. 2024;1. doi:10.1109/JIOT.2024.3491315.

20. Femminella M, Palmucci M, Reali G, Rengo M. Attribute-based management of secure Kubernetes cloud bursting. IEEE Open J Commun Soc. 2024;5:1276–98. doi:10.1109/OJCOMS.2024.3367461.

21. Meng X, Du Y, Wang C. ECMO: an efficient and confidential outsourcing protocol for medical data. IEEE Open J Comput Soc. 2024;6:37–48. doi:10.1109/OJCS.2024.3506114.

22. Arundathi JS, Satyanarayana KV. A secure and efficient framework for multi-user encrypted cloud databases supporting single and multiple keyword searches. Int J Adv Comput Sci Appl. 2024;15(9):537–46. doi:10.14569/issn.2156-5570.

23. Bhansali PK, Hiran D, Kothari H, Gulati K. Cloud-based secure data storage and access control for Internet of medical things using federated learning. Int J Pervasive Comput Commun. 2024;20(2):228–39. doi:10.1108/IJPCC-02-2022-0041.

24. Saha S, Chowdhury C, Neogy S. A novel two phase data sensitivity based access control framework for healthcare data. Multimed Tools Appl. 2024;83(3):8867–92. doi:10.1007/s11042-023-15427-5.

25. Samala AD, Rawas S. Transforming healthcare data management: a blockchain-based cloud EHR system for enhanced security and interoperability. Int J Onl Eng. 2024;20(2):46–60. doi:10.3991/ijoe.v20i02.45693.

26. Besharati E, Naderan M, Namjoo E. LR-HIDS: logistic regression host-based intrusion detection system for cloud environments. J Ambient Intell Humaniz Comput. 2019;10(9):3669–92. doi:10.1007/s12652-018-1093-8.

27. Hu H, Cao Z, Dong X. Autonomous path identity-based broadcast proxy re-encryption for data sharing in clouds. IEEE Access. 2022;10(1):87322–32. doi:10.1109/ACCESS.2022.3200084.

28. Kheddar H, Dawoud DW, Awad AI, Himeur Y, Khan MK. Reinforcement-learning-based intrusion detection in communication networks: a review. IEEE Commun Surv Tutor. 2024. doi:10.1109/COMST.2024.3484491.

29. Xie M, Yang X, Hong H, Wei G, Zhang Z. A novel verifiable Chinese multi-keyword fuzzy rank searchable encryption scheme in cloud environments. Future Gener Comput Syst. 2024;153(3):287–300. doi:10.1016/j.future.2023.11.017.
30. Agarwal A, Sharma P, Alshehri M, Mohamed AA, Alfarraj O. Classification model for accuracy and intrusion detection using machine learning approach. PeerJ Comput Sci. 2021;7(3):e437. doi:10.7717/peerj-cs.437.
31. Deng H, Qin Z, Wu Q, Guan Z, Deng RH, Wang Y, et al. Identity-based encryption transformation for flexible sharing of encrypted data in public cloud. IEEE Trans Inf Forensics Secur. 2020;15:3168–80. doi:10.1109/TIFS.2020.2985532.
32. Saxena UR, Alam T. Role-based access using partial homomorphic encryption for securing cloud data. Int J Syst Assur Eng Manag. 2023;14(3):950–66. doi:10.1007/s13198-023-01896-2.
33. Zhou Y, Tang B, Yang Y. A lattice-based searchable encryption scheme with multi-user authorization for the certificateless cloud computing environment. Trans Emerging Tel Tech. 2024;35(4):e4960. doi:10.1002/ett.4960.