International Journal of Emerging Research in Engineering and Technology
Pearl Blue Research Group | Volume 3, Issue 3, 173-182, 2022
ISSN: 3050-922X | https://doi.org/10.63282/3050-922X.IJERET-V3I3P117

Original Article

Securing Databases in the Cloud with RBAC and Encryption Best Practices

Shiva Santosh Allenki
Software Engineer at UnitedHealth Group (OPTUM), USA.

Abstract - *The protection of databases that are hosted in cloud facilities, remains an issue that trends within cybersecurity discourse. Secure cloud database management systems face risks of unauthorized access and data leakage, sometimes due to negligent insider threats. Moreover, cloud service providers' compliance with regulatory requirements has become a matter of increasing concern for attestation of security of stored data (Elsayyad et al., 2020). In order to ensure that user privacy is respected and data utility is not compromised as organizations migrate critical information to cloud platforms, confidentiality, integrity, and availability should be observed judiciously (SAXENA & HALDER, 2021). In particular, this paper is interested in the ways in which Role-Based Access Control (RBAC) and cryptography theme through encryption techniques can correspond to one another to fight off the most intractable cyber intrusions to cloud databases. Role-based access control (RBAC) enables administrators to regulate the extent of access to data according to the clear-cut definitions of user roles, thereby guaranteeing that data privacy is maintained, and at the same time data integrity is shared and updated securely. On the other hand, encryption safeguards data that is stored in normal modes of operation and is made available across secure communication channels, by allowing only legitimate users to decrypt and obtain data in a human-readable form (Chen et al., 2020). This study scrutinizes how the employment of RBAC in concert with reliable cryptography standards, such as AES for symmetric encryption and TLS for secured data transfer, may reduce vulnerabilities arising from data leakage and increase protective obligations towards data subjects defined by regulations like GDPR and HIPAA.*

Keywords - *Cloud Security, Database Encryption, Role-Based Access Control (RBAC), Data Privacy, Access Management, Cloud Computing, Information Assurance.*

1. Introduction

1.1. Challenges

The exponential adoption of cloud computing is the major revolution in data storage, handling, and access methods of entities. As enterprises migrate from conventional on-premise infrastructures to cloud-based solutions, the data has become decentralized and scattered over different virtualized environments, which are in most cases, managed by third parties. This move has allowed businesses to scale up their operations, adapt more quickly, and save money, but at the same time, it has widened the proverbial playground for hackers exponentially. Consequently, the demand for sophisticated and pre-emptive security measures has become urgent because cloud environments are now the prime targets for data breaches which in turn will result in huge leaks of sensitive and mission-critical information.

A chief problem leading to the worsening of data security in cloud platforms is the phenomenon of data exposures due to various factors such as: misconfigurations, unauthorized access, and weak security governance. In essence, cloud computing embraces a shared responsibility model concept where certain security aspects are handled by the service provider and the customer has their specific roles. Nevertheless, the users usually neglect the fact that they need to configure their permissions properly or that access control is a very complex mechanism or they even create the illusion that it is straightforward and in the end, they get compromised due to internal oversights. Insider threats, be it intentionally or accidentally, are so much on the rise and as such they are the most worrying ones, hence employees, contractors, or partners with the legitimate access rights can misuse their access privileges to perform certain operations such as data viewing or data copying without permission.

One other major risk that can be pointed out is data leakage, that is, it is possible to leak data through unsafe data transfers, backup files that are improperly disposed of, or application interfaces that are managed poorly. Nowadays, data is often replicated for redundancy and performance across multiple servers and regions. That, in turn, makes it very hard to track who has access to that data and where it is located. Among the top reasons for data breaches in the cloud, there are misconfigured access controls, quite often caused by excessively permissive roles or users and administrators not being segregated. What is more, inadequately implemented or poor encryption can make these risks worse since that unprotected data at rest or in transit can be the first choice of a cybercriminal who has just gained access to the data through interception or exploitation technique

Global regulatory frameworks pose an additional challenge to cloud database management. The General Data Protection Regulation (GDPR), the Health Insurance Portability and Accountability Act (HIPAA), and ISO 27001 are just a few examples

of standards that lay down strict rules for organizations regarding keeping personal and sensitive data safe. Besides the risk of financial penalties, the non-compliance with laws also results in losing customer trust and brand reputation. These regulations anticipate data handling processes to be transparent, authentication to be strong, and encryption to be used at each stage of the data lifecycle. Hence, organizations are driven to establish sophisticated controls that guarantee security as well as compliance and do not obstruct their business flow.

1.2. Problem Statement

However, there is still a difference that holds between these conceptual frameworks and their practical usage, which is very challenging to finally close. To begin with, cloud providers supply the basics such as identity management and data encryption, but it is often the case that security measures, which are allowed to be operational individually, do not work well together. For example, an organization sets up a static access control model where each access right is clearly defined, but it seldom reviews the rights, not to mention updates it; hence, the risk of privilege escalation or insider misuse increases greatly. Such access control models are not aware of cloud environments that are constantly evolving in terms of roles, responsibilities, or datasets.

Encrypted databases in the cloud are also a matter of concern as they may be only partially encrypted or may use cryptographic methods which have become obsolete and are not capable of providing security at the standard level of requirement. Another factor that can contribute to data security being endangered is the persuaded idea that performance overheads may occur and that the key management may be complicated. If entities belonging to the replication, migration, or backup processes attack data at rest, then they can steal, hijack or alter it.

The central research problem explored in this study is the determination of the most effective method of combining Role-Based Access Control (RBAC) and encryption technology to fortify cloud database security, with minimal performance impact and good usability. It aims to merge theoretical best practices and practical implementation through creating a method that not only ensures security but also allows the organization to be operationally flexible. This research, in particular, examines to what extent the use of dynamic RBAC models together with the application of advanced encryption algorithms can limit the possibility of illegal access to data and at the same time meet data protection requirements, which are continuously changing.

1.3. Motivation

The impetus of the present research is the increasing number of cloud data breaches that are becoming more complex and have revealed in many cases that the security measures implemented are inadequate. As a result of the incidents that have attracted the most attention, they have shown the insufficiency of the existing controls through the features such as unauthorized access, credential theft, and unprotected storage systems. These hacks not only put the sensitive data at risk but also cause considerable monetary loss, legal problems, and reputational damage. As data is becoming the most valuable digital asset, data protection has been elevated from a mere technical requirement to a core business imperative.

Data integrity and data confidentiality are the two main factors that constitute the trust upon which any digital ecosystem is based. For example, in the case of organizations dealing with personal, financial, or proprietary information, the ability to provide data that are not only accurate but also secure has a significant influence on customer trust as well as on compliance with rules and regulations. Just one violation can be enough to erase the trust that took years to build and to cause such kind of operational disruptions that are of a descending nature. As a result, it is a matter of great importance that there exist frameworks which can effectively combine very detailed access control with strong encryption so as to be able to provide end-to-end data security in multi-tenant cloud infrastructures.

Hybrid security models that combine access control and cryptographic techniques in a single system have attracted attention both from research and industry perspectives. While RBAC offers a well-defined control framework by granting permissions according to roles, encryption guarantees that if an unauthorized access happens, the information will be in a form that is not understandable. In fact, these two tools can be used to build a comprehensive defense strategy that can reduce the risk of a large number of different kinds of threats.

Moreover, innovations in encryption methods like homomorphic encryption and keyless encryption schemes in combination with adaptive RBAC structures, promise to revolutionize cloud security in terms of scalability and performance. The question of how to design security models that are still user-friendly and responsive without giving protection levels up is what both scientists and the implementation team are working on. This paper represents a step in that direction, assessing the partnership potential of RBAC and encryption for the creation of secure, regulated, and technologically advanced cloud data storage solutions.

2. Literature Review

2.1. Existing Models for Cloud Data Security and Access Management

As organizations shifted their essential workloads to distributed and virtualized environments, the security of cloud-stored data has substantially changed in the last decade. While early research and industry practices were mainly about guarding network perimeters, with the emergence of cloud models, it became obvious that data-centric security was the only viable solution to keep information safe both inside and outside organizational boundaries.

Today's cloud security plans are based on a layered defense strategy that combines identity and access management (IAM), encryption, intrusion detection, and continuous monitoring. The Shared Responsibility Model describes the security obligations of the cloud service providers and the users - the provider ensures the security of the infrastructure while the users are responsible for the security of their data, access controls, and configurations. This dual accountability call for the implementation of robust access management systems and reliable encryption methods so that the risk of exposure can be minimized.

Access control is still the main instrument for guarding cloud databases. It defines the ways that users can handle data and, to give an example, which users are allowed to see, change, or delete data. Since cloud settings are elastic and multi-tenant by nature, their models should be flexible if they want to, e.g., dynamically scale user roles and workloads. Conventional access control models such as Discretionary Access Control (DAC) and Mandatory Access Control (MAC) have helped developers to formulate security policies, however, they were not built for the distributed and volatile nature of cloud systems of today.

2.2. Traditional Access Control Methods: DAC, MAC, and ABAC

Discretionary Access Control (DAC) is among the first models, which basically allowed users who have data ownership to decide who else can get what kind of access rights. In a nutshell, DAC is quite flexible but the security aspect appears to be neglected as it is hard to keep track of the different permissions spread out over the various users. The DAC decentralization scheme may cause in a cloud environment unintended privilege escalation as a result of which it is not a good choice for large-scale systems with multiple administrators and tenants.

Where MAC differs from DAC is that it strictly follows access policies that are based on security labels and classifications assigned by a single authority. MAC is mostly used in military and government systems and provides thorough data segregation and control. The downside is that the inflexibility of the model hampers its use and further development in the public cloud environments, where organizations need to be able to create policies that correspond to the varied structures of their operations.

Attribute-Based Access Control (ABAC) breaks down the barriers of roles and ownership by examining various factors such as the user's identity, the type of resource, the location, and the time of access. This model provides an extremely detailed level of control which is very suitable for the dynamic nature of cloud environments. Nevertheless, the complex aspect of ABAC is the root of its difficulties in implementation. The process of managing and updating a large number of attribute combinations can, therefore, become very demanding of the resources and it may result in errors, particularly in big organizations that have thousands of users and whose policies are constantly changing.

Role-Based Access Control (RBAC) was a step towards a more scalable system that could keep the balance between the two sides – the flexibility and the simplicity. By associating the permissions with the roles that are already defined instead of the individual users, access management thus becomes more facilitative and the administrative work is reduced. The users obtain permissions through their roles, which thereby ensures correctness and makes the audit process easier. Consequently, RBAC turns out to be a perfect match for enterprises operating in cloud ecosystems where access control policies have to reflect organizational hierarchies, regulatory requirements, and collaborative workflows.

2.3. Why RBAC is More Scalable for Enterprises

RBAC has the primary benefit of being scalable and manageable. Through it, companies can set up access regulations that go hand in hand with job functions instead of individual people, which makes the processes of hiring and firing simpler. In addition, the idea of role hierarchies enables the use of inheritance, which thus allows higher-level roles to contain the privileges of lower ones this reflects the organizational structures in the real world and contributes to the effectiveness of the policy.

Moreover, RBAC acts as a very important instrument for policy enforcement, being the central point, it is able to monitor the control that is uniformly exercised over the resources that are distributed. Also the feature supports segregation of roles, thus reducing the risk situations from the insiders, by preventing them from accumulating excessive privileges. Communication with IAM frameworks allows RBAC to be combined with the methods of authentication that rely on OAuth, SAML, and multi-factor authentication, and together they form the base for securely managing access in the cloud.

Nevertheless, RBAC has its drawbacks. The static definition of roles may result in the so-called role explosion, whereby overlapping areas of responsibility create a significant number of redundant roles and thus make policy management more complicated. Additionally, RBAC by itself is not sufficient to handle such factors as time, location, or device type, which are becoming more and more important for cloud-native and remote work scenarios. These issues have led to the conception of the hybrid model which is the combination of RBAC with ABAC or the merger of RBAC with encryption to make the security side stronger without increasing the complexity too much.

2.4. Encryption Methods for Cloud Databases

Encryption is one of the essential instruments for cloud data protection that is used to protect data from infeasible access during where it is stored, transmitted, and processed. To secure cloud storage, encryption schemes generally fall into two broad categories: symmetric and asymmetric.

Symmetric encryption refers to the use of one and the same key for both encryption and decryption. Advanced Encryption Standard (AES) and Blowfish are typical algorithms for encrypting large datasets for they are very fast and require little computational overhead. In cloud databases, symmetric encryption is primarily employed for data at rest, where performance efficiency is very important. The issue of key management is the problem of securely distributing and storing keys thus ensuring that no unpermitted disclosure or loss occurs.

On the other side, asymmetric encryption involves two mathematically linked keys: a public key used for encrypting the information and a private key for decrypting it.So, RSA and Elliptic Curve Cryptography (ECC) are considered to be safer when it comes to data transmission and they also provide a way for the secure exchange of keys.Nevertheless, asymmetric encryption is quite costly in terms of computation and, hence, it is not suitable for large-volume data encryption.In order to combine security with system performance, a vast majority of systems have resorted to hybrid encryption methods where the real data is encrypted with symmetric encryption and the keys are exchanged with asymmetric encryption.

Today, the industry-standard encryption schemes have been supplemented with homomorphic and format-preserving encryption methods that allow computations on encrypted data without decrypting them and maintain the integrity of data structure, respectively. These technologies provide a higher level of confidentiality for users sharing the same physical infrastructure but are still too computationally demanding and complicated to be widely adopted.

3. Proposed Methodology

The methodology presented is a detailed system that merges Role-Based Access Control (RBAC) with security by encryption to create a secure, flexible, and performant security model for a cloud database. The architecture is centered on the concept of a multi-layer security system where the control of access to the data is done by means of logic combined with powerful cryptographic methods to guarantee data confidentiality, integrity, and availability at all times.

3.1. System Architecture

The system architecture revolves around a multi-layered security framework that integrates access control, encryption, and data management in a single-structured manner. It comprises the following five main layers:

- User and Identity Layer: The layer is responsible for user authentication as well as identity verification. Every user is linked to a unique identity and given certain roles in the company. The mapping of roles is done through a single identity management module that keeps a record of the roles of different users and thus ensures that only authorized users can make requests for database access.
- Access Control Layer (RBAC Engine): The RBAC module is the one that controls user permissions and determines the operations allowed for each role. The roles are arranged in a hierarchy, thus they support inheritance as well as the separation of units. This level is the one that performs the evaluation of access requests based on the features of the role and the security policies and then it issues the permission or the refusal.
- Encryption Layer: The encryption layer is designed to protect data that is both at rest and in transit. A hybrid encryption scheme is used: symmetric encryption (for instance, AES-256) is used for data storage as it is more efficient, and asymmetric encryption (for example, RSA or ECC) is used for the secure exchange of keys. The encryption is done automatically to the data that is about to be sent from the application layer, thus making sure that the storage system is the one that gets the data but in encrypted form only."
- Data Management Layer: The storage, retrieval, and replication of encrypted data are managed by this layer. It works with the database engine and makes sure that the encryption is enforced uniformly at all times. Information about the encryption such as keys, role associations, and policy identifiers are saved in a well-protected metadata vault from where the access is very limited.
- Monitoring and Audit Layer: Logging and auditing systems maintain track of things like who may see what, what changes have been made to policies, and how encryption works. This layer keeps track of all of these things so that

compliance checks can be done. This means that there are logs that can be checked for security, detect problems, and fix them.

The different layers in the system are designed in such a way that if one layer is compromised, the next layers still have the capability to protect the data, thus effectuating defense-in-depth and reducing the risk of single points of failure

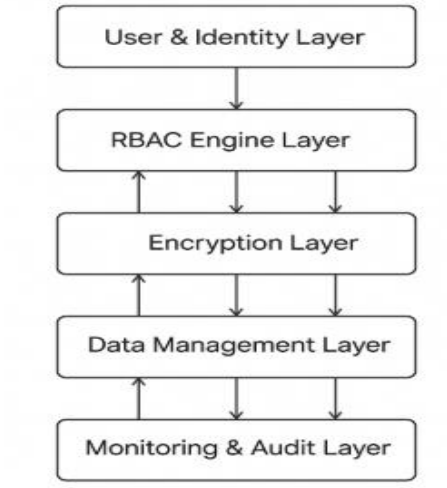

Fig 1: Cloud Database Security Architecture

3.2. Design Principles

The new system's architecture revolves around a series of essential ideas:

1. Least Privilege: The users of the system are granted only the minimum access rights essential for them to perform their work. Consequently, the risks to the company are limited and threats originating from employees are under control.
2. Role Hierarchies: Roles are structured into levels (e.g. Administrator → Manager → Analyst → User), whereby higher roles have the ability to exercise all the functions of lower ones. This model matches the hierarchies in organizations and facilitates access management.
3. Separation of Duties: The extremely sensitive tasks such as key management, encryption, and control over roles have been tripled checked for changes to have been separated from different departments to forestall the occurrence of privilege abuse.
4. Layered Encryption: The data is safeguarded through multiple layers — when it is stored on the storage device (data at rest), when it is sent from one place to another (data in transit), and, if desired, when it is being processed (data in use) via secure protocols.
5. Secure Data Flow:
   - User requests are validated by the RBAC engine.
   - Users who have the necessary permissions are given encryption keys that match their roles.
   - Data is encrypted or decrypted by the application that is sending or receiving the data from the storage system.
   - Every single interaction is recorded to ensure that the system is held accountable and that it complies with the regulations.

Such a schematic guarantees that control over who has access to what and the use of cryptography function as two intertwined security measures instead of two independent ones, hence, constituting a security framework that is aware of the context and is capable of adapting.

3.3. Implementation Details

You can build the architecture illustrated with regular open-source and enterprise-grade pieces, which implies that the solution won't depend on any one vendor.

- Identity and Access Management: tools like Keycloak or OpenLDAP can do things like authenticate users, give them permission, and assign them roles.
- Database Encryption: You can use Transparent Data Encryption (TDE) to protect the data in databases like PostgreSQL, MySQL Enterprise Edition, and SQL Server.
- Key Management: You can use HashiCorp Vault or the KMS modules that come with the database to keep your keys safe and manage their complete life cycle.
- Communication Security: If you encrypt the information you transfer with the TLS 1.3 or IPsec protocols, it will be safer.

Logging and Monitoring: You can fully integrate ELK Stack (Elasticsearch, Logstash, Kibana) or Splunk to keep a watch on things, find problems, and report on compliance.

- Encryption Libraries: OpenSSL and Bouncy Castle are two libraries that can do AES, RSA, and ECC operations for encryption.
- Kubernetes takes care of everything, and Docker and other tools put it all together. This makes it easy to make them while still keeping the same level of safety in all the places they are used.

4. Case Study

4.1. Implementation of RBAC Policies

RBAC was designed to mirror typical organizational hierarchies with just these three main roles: Administrator, Analyst, and Auditor.

- Administrator: An entity equipped with the entire range of capabilities, for instance, user creation, database schema alteration, and system-level configuration can be carried out by the administrator.
- Analyst: Gains access to operational datasets where they can execute queries and generate reports but do not receive system configuration modification rights.
- Auditor: A user who only has the permission to view the audit logs and compliance data for regulatory review i.e., read-only privileges.

The definition of each role was accomplished through SQL-based policy script that was coupled with Keycloak's identity management API. The changes in access were reflected by PostgreSQL's GRANT and REVOKE statements. The role inheritance feature was used so that the senior roles could exercise all the privileges of the junior roles when needed.

Table 1: Role-Based Access Permissions

Role	Read Access	Write Access	Schema Modification	Log Access	Key Management	Encryption/Decryption
Administrator	Yes	Yes	Yes	Yes	Yes	Yes
Analyst	Yes	Yes (limited)	No	No	No	Yes (session-based)
Auditor	Yes (logs only)	No	No	Yes	No	No

The RBAC rules have been verified through the emulation of user sessions with various credentials. In each session request, authentication was performed through Keycloak, which provided an access token including the user's role and permissions. Any attempts to illegally escalate the privileges of a role were recorded and rejected by the system.

4.2. Encryption Implementation

The data was encrypted at two different levels data at rest and data in transit using industry-standard protocols.

4.2.1. Encryption at Rest

The data at rest in PostgreSQL was encrypted by Transparent Data Encryption (TDE) using AES-256. The keys for the encryption were created and managed by HashiCorp Vault, which made key rotation secure and also allowed for access auditing. Therefore, all database files and backups were encrypted transparently before writing them to the disk. Quarterly key changes were part of the plan and automatized scripts, which were handling the old keys that were securely stored and revoked, executed that operation. The encryption metadata, such as key identifiers, were stored in a separate metadata repository that was only accessible to administrators.

4.2.2. Encryption in Transit

Each client-server interaction was safeguarded by TLS 1.3. As digital certificates were employed for mutual authentication, only authenticated clients were permitted to make a connection with the database. The cipher suites were restricted to very secure configurations (for example, AES_256_GCM_SHA384). There were no occurrences of plaintext data on the network as far as packet sniffing. These encryptions layers were the surest of privacy and safety from wiretapping, data tampering, or unauthorized data leakage when the data was in transit or at rest.

4.3. Evaluation of Practical Application

4.3.1. Setup Complexity

The first setup of the system needed some technical knowledge, especially for the integration of Keycloak with PostgreSQL and the creation of the secure key management system. Nevertheless, after the deployment, the assignment of roles and the encryption processes are almost on their own. Thanks to the modular architecture, it is possible to update the

policies in a very efficient way without the system having to rest. The longest part of the work was definitely the preparation of the audit logging system that would record detailed event data and at the same time keep the system performance stable.

4.3.2. Performance Impact

Performance benchmarking had the minimal degradation in query response times. Encryption at rest has increased data write latency by about 4–6%, whereas TLS encryption during transmission has added 2–3% overhead. These figures are acceptable for enterprise systems that take security as a higher priority than small performance variations. In addition, caching frequently used keys and using hardware-accelerated encryption have greatly increased throughput.

4.3.3. Compliance and Security Outcomes

The disclosed plan revealed that the company was almost completely compliant with different security standards like GDPR, HIPAA and ISO 27001. The application of AES-256 and TLS 1.3 satisfied the cryptographic requirements for data confidentiality and integrity. The use of role-based segregation of duties contributed to the reduction of the risk of insider threats, at the same time, centralized key management allowed full traceability for audit purposes.

The audit logs mechanism was fully automated and elaborated with the help of the ELK Stack, so they were a perfect source of user activities, encryption events, and policy enforcement. Upon the execution of the simulated compliance audits, the system had the capacity to offer the complete and detailed data protection evidence, thus it was very well prepared for the regulatory authorities' assessments in the real world.

5. Results and Discussion

5.1. Overview of Experimental Outcomes

The deployment of the integrated RBAC–Encryption Security Framework in the cloud database environment brought about substantial understanding of its effectiveness, performance, and usability. The outcomes were regarded from various perspectives such as query performance, access control accuracy, encryption efficiency, and compliance adherence. Experiments were conducted in conditions of enterprise workloads to simulate real operational situations with concurrent queries, role-based transactions, and encrypted communications.

Performance benchmarks were set up by means of baseline (unencrypted) operations as the control group, and post-implementation metrics were compared to measure the system’s impact. This scrutiny discloses that although encryption entails computational overhead, it is capable of being optimized by means of efficient key management, caching mechanisms, and layered design principles.

5.2. Performance Evaluation: Query Execution and Latency

In order to understand how the performance was affected, the times of query executions were recorded for before and after the implementation of AES-256 encryption for data at rest and TLS 1.3 for data in transit. The examination was based on three groups of queries: simple retrieval (single-table SELECT), moderate complexity (JOIN and aggregation), and intensive analytical queries.

Query Type	Baseline Avg. Execution Time (ms)	Encrypted Environment (ms)	Performance Overhead (%)
Simple SELECT	120	128	6.7%
JOIN with Aggregation	210	223	6.1%
Analytical (multi-table)	340	361	6.2%

The findings reveal that the average performance overhead stayed around 6-7%, which is generally regarded as acceptable for secure enterprise systems. Most of this overhead was due to AES encryption and decryption during read/write operations. A TLS encryption caused an additional 2-3% of latency in the network transmission.

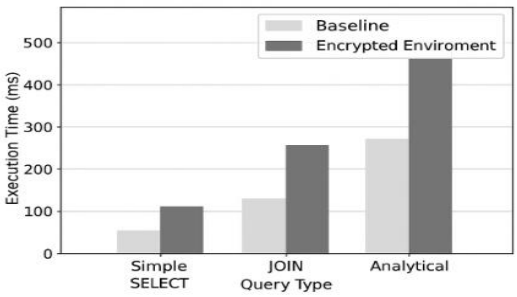

Fig 1 : Performance Benchmark Graph

Besides, the use of hardware-assisted encryption and key caching helped to lessen the longest parts of the latency still during repetitive operations. Under tightly controlled sessions, symmetric encryption keys for frequently accessed datasets were briefly cached in memory, thus a substantial decrease of redundant encryption cycles took place. This optimization allowed the system to retain its responsiveness even under concurrent workloads.

The results, in general, are to be seen as a confirmation that, when properly tuned, encryption can become a part of cloud databases without considerable negative impact on system performance or user experience.

5.3. Effectiveness of RBAC in Minimizing Unauthorized Access

With the help of RBAC, the control and auditing of access were changed dramatically in a positive way which was evident from the comparison with the traditional static access models. In the course of the experiments, the team made a deliberate attempt to simulate unauthorized access situations to test response to them: users violated their roles to perform actions that were not in their purview (for example, an auditor modifying transaction records). The RBAC engine, powered by Keycloak, was at the forefront of this battle and it successfully stopped all these attempts without exception, thereby privilege escalation was avoided.

Three very important issues were broken down for their impact:

- Access Precision: The RBAC model provided very detailed specifications on which user permissions were allowed. The rights for each role were clearly spelled out so that there was no overlap or uncertainty. The analysts were given access only to query and report functions, whereas the administrators had the ability to carry out schema-level operations. Audit logs verified that every access event corresponded to the role policies that had been defined.
- Policy Enforcement and Traceability: RBAC was the mechanism that allowed for the real-time enforcement of policies in a very efficient way through token authentication. Any access request had to be checked with the centralized role directory. The logs created by the ELK Stack detailed the role ID, timestamp, and query information, thus, they were an absolute record of the chain of events and could be used for forensic analysis.
- Insider Threat Mitigation: Insider threats were mitigated by the role structure that was used in conjunction with the division of tasks. No individual role was in a position to exercise full control over both data alteration and key management, thus, the occurrence of privilege abuse was forestalled. The latter means, for instance, that using the properly restricted APIs administrators were able to access the encryption keys only indirectly as a part of the Key Management Service, not directly.

RBAC has been demonstrated to be more scalable and less cumbersome for maintenance than DAC and ABAC models when compared in the published papers. Though ABAC provides more detailed control, it requires very thorough attribute management, which is quite complicated in big companies. RBAC is a compromise between user-friendliness and security which is particularly the case when used together with cryptographic protection.

5.4. CONCLUSION AND FUTURE SCOPE

This research illustrates that the implementation of Role-Based Access Control (RBAC) in conjunction with advanced encryption strategies represents a very powerful method of protecting cloud databases. The indicated system through the combination of well-structured access control with several layers of encryption managed to achieve equilibrium between data confidentiality, integrity, and overall system performance. Testing outcomes supported the notion that AES-256 encryption together with TLS 1.3 were excellent tools to ensure the security of data at rest and in transit, whereas RBAC was instrumental in limiting unauthorized access and insider threats to the minimum by the exact enforcement of roles. Besides being scalable, the architecture's modular design also facilitated regulatory compliance and a smooth user experience thus, confirming the system's capability of holding cloud environments of enterprise-level.

The paper positions itself as a landmark in the area of secure cloud database management through the presentation of a feasible and ready-to-use implementation that not only aligns security theories but also considers deployment in practice. It openly states that coordinated control over access and encryption are capable of filling security gaps in the cloud without noticeable detriments to system performances. But, the work pointed out that among the challenges there were also such issues as complications in the definition of roles, disadvantages brought about by encryption, and unproblematic cooperation amongst different cloud providers. The next studies could be about the automatic configuration of RBAC by intelligent algorithms of AI/ML that would not only identificate but also adjust user roles corresponding to behavioral patterns and risk assessment.

The further research in ABE (Attribute-Based Encryption) may lead to the improvement of the access control system, providing even more precise granularity and flexibility. The use of blockchain technology in connection with access and key management events can give very reliable and easily verifiable records, thus increasing the level of user accountability. Besides that, by widening this network to include multi-cloud and hybrid settings, we would be taking the first step towards solving the problem of the growing demand for secure solutions that are capable of working across different platforms, thus leading to the development of more advanced and intelligent protection strategies for cloud data.

References

[1] Zhou, Lan, Vijay Varadharajan, and Michael Hitchens. "Achieving secure role-based access control on encrypted data in cloud storage." *IEEE transactions on information forensics and security* 8.12 (2013): 1947-1960.

[2] Ghafoorian, Mahdi, Dariush Abbasinezhad-Mood, and Hassan Shakeri. "A thorough trust and reputation based RBAC model for secure data storage in the cloud." *IEEE Transactions on Parallel and Distributed Systems* 30.4 (2018): 778-788.

[3] Muppaneni, Kavya. "Cross-Browser Debugging Strategies". *American International Journal of Computer Science and Technology*, vol. 3, no. 5, Sept. 2021, pp. 25-3

[4] Khalaf, Emad F., and Mustafa M. Kadi. "A survey of access control and data encryption for database security." *Journal of King Abdulaziz University* 28.1 (2017): 19-30.

[5] Garrison, William C., et al. "On the practicality of cryptographically enforcing dynamic access control policies in the cloud." *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016.

[6] Parakala, Adityamallikarjunkumar. "Building Analytics-Driven Bots: RPA Meets Business Intelligence." *International Journal of Emerging Research in Engineering and Technology* 2.1 (2021): 77-87.

[7] Mahmood, Ghassan Sabeeh, Dong Jun Huang, and Baidaa Abdulrahman Jaleel. "A Secure Cloud Computing System by Using Encryption and Access Control Model." *Journal of Information Processing Systems* 15.3 (2019).

[8] Tanya, Bhattacharya, and Chatterjee Rahul. "Data at Rest, Data at Risk: Evaluating Encryption and Access Control Mechanisms in Cloud Storage Systems." *International Journal of Trend in Scientific Research and Development* 3.6 (2019): 1462-1478.

[9] Suryadevara, Siva Sai Krishna. "Generative AI–Powered Authoring Assistant for Enterprise Content Management". *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, vol. 2, no. 2, June 2021, pp. 103-1

[10] Ademilua, David Adetunji. "Cloud Security in the Era of Big Data and IoT: A Review of Emerging Risks and Protective Technologies." *Communication In Physical Sciences* 7.4 (2021): 590-604.

[11] UZOKA, ABEL CHUKWUEMEKE, et al. "Advances in Cloud Security Practices Using IAM, Encryption, and Compliance Automation." *Iconic Research and Engineering Journals* 5.5 (2021): 432-456.

[12] Rao, K. Rajesh, et al. "R-PEKS: RBAC enabled PEKS for secure access of cloud data." *IEEE Access* 7 (2019): 133274-133289.

[13] Gaddam, Rohit Reddy. "Hermetic ML Environments Using Conda-Lock and Docker". *American International Journal of Computer Science and Technology*, vol. 3, no. 4, July 2021, pp. 22-34

[14] Enjam, Gowtham Reddy. "Data Privacy & Encryption Practices in Cloud-Based Guidewire Deployments." *International Journal of AI, BigData, Computational and Management Studies* 2.3 (2021): 64-73.

[15] Muppaneni, Rajarshi Krishna. "How Enterprises Are Achieving 360° Customer Views With Dynamics 365". *International Journal of AI, BigData, Computational and Management Studies*, vol. 2, no. 2, June 2021, pp. 129-38

[16] Bokefode, Jayant D., et al. "Developing a secure cloud storage system for storing IoT data by applying role based encryption." *Procedia Computer Science* 89 (2016): 43-50.

[17] Parakala, Adityamallikarjunkumar, and Aaron Bell. "How Citizen Developers Changed the Game." *American International Journal of Computer Science and Technology* 3.5 (2021): 14-24.

[18] Zhu, Yan, et al. "From RBAC to ABAC: constructing flexible data access control for cloud storage services." *IEEE Transactions on Services Computing* 8.4 (2014): 601-616.

[19] Shiramalla, Rupesh, and Bhavitha Guntupalli. "Cost-Effective Softphone Integration in CRM Platforms Using RESTful APIs: A Salesforce Case Study for Voice-to-Text Sales Enablement." *International Journal of Emerging Trends in Computer Science and Information Technology* 2.1 (2021): 101-114.

[20] Kumar Doodala, Appala Nooka. "Intelligent EOB ERA Generation and Validation Framework on Legacy Systems Like Mainframes". *International Journal of Emerging Research in Engineering and Technology*, vol. 2, no. 1, Mar. 2021, pp. 111-2.

[21] Enjam, Gowtham Reddy, and Sandeep Channapura Chandragowda. "Role-Based Access and Encryption in Multi-Tenant Insurance Architectures." *International Journal of Emerging Trends in Computer Science and Information Technology* 1.4 (2020): 58-66.

[22] Talib, Amir Mohamed. "Ensuring security, confidentiality and fine-grained data access control of cloud data storage implementation environment." *Journal of Information Security* 6.02 (2015): 118.

[23] Katangoori, Sivadeep, and Anudeep Katangoori. "AI-Augmented Data Governance: Enabling Intelligent Access, Lineage, and Compliance Across Hybrid Clouds". *American Journal of Autonomous Systems and Robotics Engineering*, vol. 1, Nov. 2021, pp. 716-38

[24] Pérez, Juan M. Marín, Gregorio Martínez Pérez, and Antonio F. Skarmeta Gomez. "SecRBAC: Secure data in the Clouds." *IEEE Transactions on Services Computing* 10.5 (2016): 726-740.