International Journal of Computer Science and Mathematical Theory (IJCSMT) E-ISSN 2545-5699
P-ISSN 2695-1924 Vol 10. No.3 2024 www.iiardjournals.org Online Version

# Secure Data Replication in Edge and Fog Computing Environments

**Alalibo, H., Bennett, E.O., Nwiabu, N.D. & Matthias, D.**
Department of Computer Science,
Rivers State University,
Port Harcourt, Nigeria
Email: henrietta.alalibo@ust.edu.ng, bennett.okoni@ust.edu.ng,
nwiabu.nuka@ust.edu.ng, Matthias.daniel@ust.edu.ng
DOI: 10.56201/ijcsmt.v10.no3.2024.pg187.201

***Abstract***
*This paper presents a comprehensive framework for secure data replication in edge and fog computing environments, addressing the challenges and solutions associated with these environments. The research proposes a multi-layered security model that integrates encryption, access control, and anomaly detection to safeguard replicated data across distributed nodes. The model leverages lightweight cryptographic techniques to ensure data confidentiality without imposing significant computational overhead, making it suitable for resource-constrained edge devices. The paper also explores advanced replication strategies that balance the trade-offs between consistency, availability, and partition tolerance, known as the CAP theorem. The approach adopts adaptive replication protocols to adjust to network conditions and workload variations dynamically, ensuring optimal data availability and fault tolerance. Extensive simulations and real-world experiments validate the efficacy of the proposed framework, showcasing significant improvements in data security and system resilience over traditional methods. The research contributes to the body of knowledge by providing practical insights and technical solutions for secure data replication, paving the way for more secure and reliable edge and fog computing systems.*

***Key Words:*** *Secure data replication, edge and fog computing environments, role-based access control, attribute-based access control, data confidentiality and integrity.*

## 1. INTRODUCTION

The proliferation of IoT devices and the demand for real-time data processing have driven the evolution from centralized cloud computing to distributed models like edge and fog computing. These models are designed to reduce latency and bandwidth usage, enabling quicker decision-making processes. However, they also introduce significant data security and privacy challenges due to the replication of sensitive data across multiple nodes in a distributed network.

Edge and fog computing provide several advantages over traditional cloud computing. They process data closer to the source, reducing latency and making them ideal for applications requiring real-time processing, such as autonomous vehicles and industrial IoT. Fog computing, in particular, extends the capabilities of edge computing by adding a layer of computing infrastructure between edge devices and the cloud, which helps manage the load and provides additional security measures [1] & [2].

However, this shift towards distributed computing models increases the risk of unauthorized access, data tampering, and disclosure. Traditional access control mechanisms like role-based access control (RBAC) and attribute-based access control (ABAC) face limitations in these dynamic and resource-constrained environments. To address these challenges, integrating RBAC and ABAC can provide a more robust security framework. This integration leverages the strengths of both approaches, allowing data owners to configure access controls that consider specific user responsibilities and characteristics. This method enhances data security by ensuring that only authorized entities can access replicated data, thus reducing the likelihood of data tampering and unauthorized disclosure. [3]

Overall, while edge and fog computing offer substantial benefits in terms of latency reduction and real-time processing capabilities, they also necessitate advanced security measures to protect sensitive data in distributed environments. The proposed integration of RBAC and ABAC represents a promising approach to enhancing the security and efficiency of data replication processes in these computing paradigms.

## 2. RELATED LITERATURE

[4] This work proposes a comprehensive solution to address these challenges by integrating Attribute-Based Access Control (ABAC) and Role-Based Access Control (RBAC) mechanisms into data replication. Our approach provides precise access control by combining Role-Based Access Control (RBAC) with Attribute-Based Access Control (ABAC). Data owners have the ability to define access restrictions based on users' responsibilities and attributes. By employing edge and fog computing architecture, our technology ensures that only authorized entities with the appropriate responsibilities and attributes may access and replicate data. This measure serves to lessen the probability of unauthorized disclosure or manipulation.

[5] Cloud computing has increased security and privacy concerns due to data outsourcing, making data access control a challenge. Attribute-based Encryption (ABE) is a popular technology for controlling data access in cloud storage systems. This paper surveys attribute-based access control schemes, categorizes them into centralized, decentralized, and hierarchal classes, and analyzes their advantages, disadvantages, significance, requirements, and research gaps. It also presents open issues and challenges for further investigation.

[6] Fog computing as a decentralized cloud-like platform, offers computing resources at the network's edge, allowing for localized processing of large amounts of data. It is suitable for applications requiring real-time responsiveness and location awareness, such as IoT devices. However, concerns about data security,

virtualization, segregation, network vulnerabilities, malware, and monitoring remain. This paper provides a comprehensive overview of Fog computing applications, focusing on identifying security gaps and incorporating technologies like Edge computing, Cloudlets, and Micro-data centres. The paper explores the impact of security issues and offers potential solutions, providing guidance for those designing, developing, and maintaining Fog systems.

[7] Fog computing faces challenges in data replication, distribution, and mobility due to its geographically distributed infrastructure. Its applications require re-implementation of data management for new software, and proposed solutions which are limited to specific domains like IoT. This paper presents FReD, a data replication middleware for fog that simplifies data distribution, enables low-latency, high-bandwidth, and privacy-sensitive applications, and provides transparent and controllable data distribution.
FReD was used as a common data access interface across heterogeneous infrastructure and network topologies, providing transparent and controllable data distribution. It was integrated with applications from different domains and evaluated using three case studies of fog computing applications. It managed data replication across any fog topology and provides mechanisms for data consistency with a client library and building data processing applications with trigger nodes. It filled a research gap in fog data management systems that abstract the complexity of fog computing while being applicable to any fog application.

Role Based Access Control (RBAC) and Attribute Based Access Control (ABAC) are two access control models that simplify data replication by assigning roles based on organizational responsibilities. RBAC assigns specific permissions to users, reducing unauthorized access or tampering. ABAC evaluates access requests based on user, resource, and environmental attributes, allowing for more granular control in edge and fog computing environments. Combining RBAC and ABAC allows organizations to implement comprehensive policies considering organizational structure and dynamic context. Both models can be enforced across edge and fog nodes, ensuring consistent access control across distributed infrastructure. RBAC and ABAC also facilitate auditing and compliance monitoring, enabling accountability and regulatory compliance. Auditing capabilities can detect unauthorized access attempts, anomalous behavior, or policy deviations in data replication processes, supporting incident response and forensic analysis.

[8] The authors explore the possibility of information leakage and the computational burden faced by EHR owners when dealing with users who have multiple occupations. They suggest Attribute-Based Encryption (ABE) as a potential solution, enabling multiple users to access a shared document while limiting decryption to specific sections. The system's hierarchical structure incorporates ABE and Identity Based Encryption (IBE), which is a method of public-key encryption. IBE is commonly employed for the secure transmission of ABE keys, as ABE allows for precise and fine-grained access control. The Trusted Server ensures the secure encryption of electronic health records prior to their upload to the cloud, while domain servers responsibly distribute the necessary keys to authorized entities. This approach is highly effective in environments with a large number of users and does not necessitate predefined values for each user.

[9] Proposed a scheme for fog systems, used a dynamic key-dependent cryptographic scheme to enhance security. It combined a shared secret key with a generated random nonce to create a dynamic key for cryptographic primitives like encoding matrices and selection/update tables. This creates dynamic and secure encoding matrices, potentially generating distinct fragments for the same input message. AES was used in conjunction with the GCM operation mode to ensure message integrity, confidentiality, and source authentication. The paper concludes with security tests and performance evaluations, confirming the effectiveness and security of the proposed fog system deployment strategy.

[10] The article reviews security and privacy challenges in cloud, edge, and fog computing environments. It identifies common threats, compares different paradigms, and discusses countermeasures. Cloud computing offers scalable storage but faces data privacy issues. Edge computing brings data processing closer together, reducing latency but increasing localized attacks. Fog computing extends cloud services to the network's edge, balancing processing load but introducing complexity. Countermeasures include encryption, access control, authentication, intrusion detection systems, and blockchain. Deployment challenges include seamless integration, managing trade-offs, and ensuring standardized protocols. Future directions include research on adaptive security mechanisms and user-centric privacy-preserving technologies.

[11] The article emphasizes the importance of secure data storage and retrieval in cloud computing due to its scalability and cost-effectiveness. Key concerns include data breaches, unauthorized access, and data loss. The authors suggest encryption, data integrity, and robust access control to enhance security. Real-world applications demonstrate the effectiveness of searchable encryption in preventing unauthorized access. However, the article also highlights limitations such as computational overhead, key management challenges, and inconsistent security practices across providers. It calls for further research to develop efficient encryption algorithms, improve key management, and establish standardized security protocols to address evolving cyber threats.

[12] The article discusses the trade-offs between consistency and availability in distributed database systems, focusing on the CAP theorem. It explains that a system can only achieve two of three properties: consistency, availability, and partition tolerance. Protocols like two-phase commit and Paxos ensure consistency, but they can introduce latency. To enhance availability, systems often use eventual consistency, allowing asynchronous data propagation but causing temporary inconsistencies. However, the article also highlights the limitations of prioritizing consistency over availability, as strong consistency protocols can introduce latency and be complex to implement at scale. The article concludes that balancing consistency and availability is a core challenge in distributed database systems, requiring further research to develop adaptive systems and efficient protocols.

[13] A Survey and Analysis of Security Threats and Challenges provides an overview of security threats and challenges in the contexts of mobile edge computing (MEC) and fog computing. It discusses unique security concerns such as data privacy, authentication, and network vulnerabilities. However, the paper may not delve deeply into specific threats or mitigation strategies due to the broad nature of these technologies. The depth of analysis may not be extensive, and the rapid evolution of technology may render some information outdated. The paper might not go into detail about how to reduce risks, all the rules that need to be followed, or upcoming trends such as how AI and machine learning will affect finding threats or how blockchain will be used to make secure data transfers in MEC and fog environments.

RBAC and ABAC play complementary roles in securing data replication in edge and fog computing environments by

i. Providing structured access control mechanisms, fine-grained access policies, and
ii. Policy enforcement capabilities.

The integration of RBAC and ABAC enables organizations to establish comprehensive access control frameworks tailored to the unique requirements and challenges of distributed computing at the network edge.

## 3. Methodology

The research method adopted is constructive research and Object Oriented Design Analysis (OODA).

**Data Flow Diagram (DFD)**
A DFD illustrates how data moves through a system and the processes that transform or manipulate the data. DFD provides an overview of the major processes and data flows within the data confidentiality and integrity model. It demonstrates how data is replicated, encrypted, verified, and stored, as well as how the system updates status, sends alerts, and maintains historical logs. Below is DFD for a model focused on data confidentiality and integrity during replication in a fog and edge computing environment:

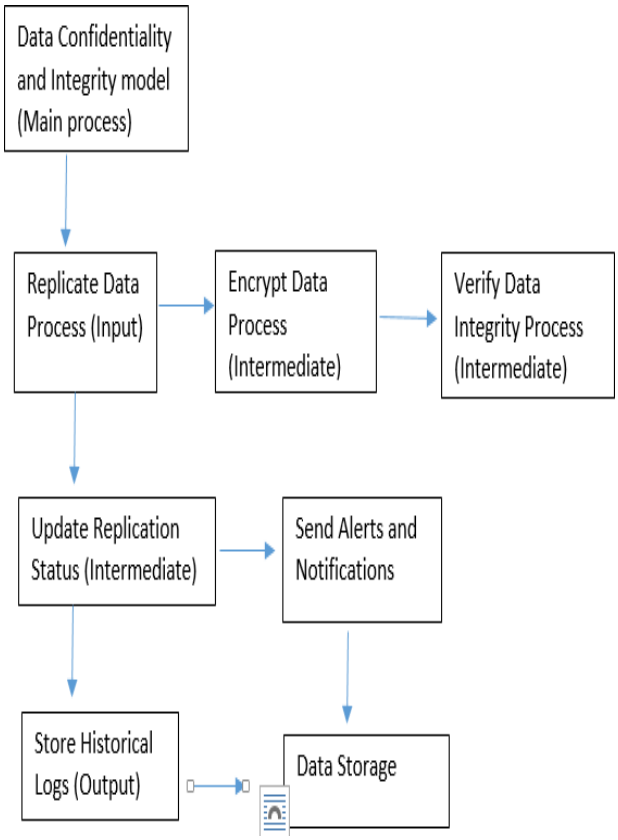


**Figure 1: Data flow diagram**

**Replicate Data Process (Input)**: Initiates the data replication process by obtaining data from the fog and edge computing devices.

**Encrypt Data Process (Intermediate):** Receives the replicated data and performs encryption using specified algorithms to ensure data confidentiality.

**Verify Data Integrity Process (Intermediate):** Receives the encrypted data and verifies its integrity using checksums or hash functions.

**Update Replication Status Process (Intermediate**): Updates the replication status based on the success or failure of the replication process.

**Send Alerts and Notifications Process (Intermediate):** Generates and sends alerts or notifications in case of critical events or issues during the replication process.

**Store Historical Logs Process (Output**): Stores historical logs of the replication process, including details on successful replications and encountered issues.

**Data Storage:** Represents the storage where the encrypted and verified data is stored securely.

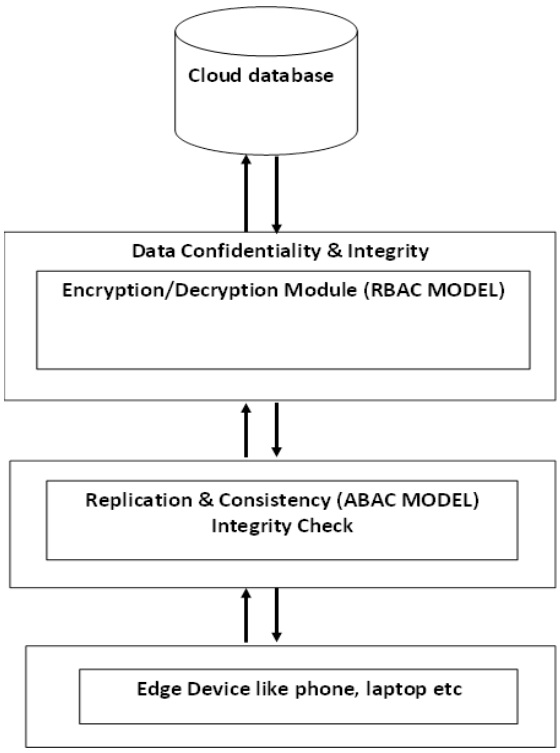


**Figure 2 – System Architecture**

**Central Cloud**: Represents a cloud infrastructure that hosts a centralized Cloud Database.

**Fog node** refers to a compute and storage resource positioned at the edge of the network. Fog nodes play a crucial role in handling and storing data generated by IoT devices. It serves as a crucial processing node that bridges the gap between the Edge and the Central Cloud. This node incorporates Replication & Consistency modules to effectively handle data replication and ensure consistency.

**Edge devices**: This is where data is first generated. Include modules for replication and consistency. An edge node is a compute and storage resource that is situated in close proximity to the data source. Edge nodes have the important task of gathering and analyzing data from IoT devices.

**Data Flow diagram**: Illustrates the movement of data between Edge, Fog, and the Central Cloud in both directions.
Data Confidentiality & Integrity Module: Incorporates a cutting-edge Encryption Module to guarantee the utmost confidentiality of data during transmission. Integrates integrity checks to ensure data integrity throughout the replication process.

**Data Confidentiality and Integrity Module:** Includes Encryption module for ensuring data

confidentiality during transmission, it includes integrity checks to verify data integrity during replication. It involves defining the various components and their interactions. Below is the representation of the model with notations for key components.

**Mathematical Model**

Let D be the original data, and D′ be the replicated data

**Data Confidentiality Components**:

i.**Encryption**

Let $E_k(D)$ represent the encryption of data D with a key k.

Let $(E^{-1}(D'))$ represent the decryption of replicated data D′ with the same key k. $D' = E_k(D)$

**Data Integrity Component**:

i.**Checksum Calculation**

Define a checksum function Checksum (D) that calculates a checksum for the original data. Let Checksum (D′) calculate the checksum for replicated data. Checksum (D′) = Checksum (D)

ii.**Integrity Validation**

Introduce an integrity validation function

Integrity Validation (D, D′) that compares the checksums of the original and replicated data. Integrity Validation (D, D′) = True

**Performance Metrics**

i.**Encryption and Decryption Time:**

Measure the time taken for encryption and decryption processes.

Let Encryption Time and Decryption Time represent these times.

Encryption Time = Time Taken ($E_k(D)$) Decryption Time = Time Taken($E^{-1}(D')$)

ii.**Time Complexity**

Define the time complexity T(n) of the replication process, where n represents the size of the data.

T(n) = Time Taken (Replication(D))

iii.**Space Complexity**

Define the space complexity S(n) of the replication process. S(n) = Space Used(Replication(D))

**Summary**

The overall replication process can be represented as:

$D' = E_k(D)$

With checks for confidentiality and integrity: Confidentiality Check(D, D′) = True Checksum(D′) = Checksum(D) Integrity Validation(D, D′) = True

And performance metrics:

Encryption Time = Time Taken ($E_k(D)$) Decryption Time = Time Taken($E^{-1}(D')$) T(n) = Time Taken(Replication(D))

S(n) = Space Used(Replication(D))

**Replication Protocol for Data Integrity**

Data integrity is maintained by distributed replication. It checks replicated data consistency with checksums

and consensus.

i. Checksum Calculation: Each replica stores a checksum (e.g., CRC32, SHA-256) for its data.
ii. Consensus Mechanism). Before changing data, consensus is reached.
iii. Replication Process: Data changes at one replica trigger the replication process.
iv. The other replicas receive the proposed change and its checksum.
v. Checksum Verification: Recalculating checksums ensures data integrity when receiving replicas.

**Mathematical Model for Data Replication Protocol**:

Variables

- D : Set of all data items.
- R : Set of replicas ($r_i$ represents an individual replica).
- C : Set of all checksums.
- P : Set of all proposed changes.

Parameters:
***data$_i$*** *: Data at replica $r_i$.*

***checksum$_i$*** *: Checksum of data at replica $r_i$.*

***p$_{ij}$*** *: Proposed change from replica $r_i$ to replica $r_j$.*

**Consensus**(p) : Function that returns true if consensus is reached on change p, false otherwise.

**Replication Process**

$$R_i' = P(D_i)$$

$R_i'$ represents the newly replicated data at the target node

**Integrity Check:** After replication, an integrity check is performed to ensure the integrity of the replicated data.

$$I(R_i') = I(D_i)$$

This equation ensures that the integrity of the replicated data matches the integrity of the original data.

**Update Replicated Data:** If the integrity check is successful, update the replicated data at the target node.

$$R_i = R_i'$$

**Consistency Check:**
Consistency is performed to ensure that all replicas across different nodes are consistent.
**Ri = Rj**
This equation ensures that the replicated data at different nodes is consistent

Framework for the mathematical model:

i. $D_i$: Original data at the source node (data to be replicated).
ii. $R_i$: Replicated data at the target node.

iii. *I*: Integrity check or hash function.
iv. *P*: Replication protocol.
v. *V*: Versioning mechanism.
vi. *E*: Error handling mechanism.

*C*: Conflict resolution mechanism

The mathematical model can be expressed as follows:

**Initialization:** Ri=∅ (initially, the replicated data is empty).

**Replication Process:** At each replication interval, the source node initiates the replication process.
$(R_i', V_i') = P(D_i, R_i, V_i)$

Ri′ represents the newly replicated data, and Vi′ represents the updated version.
**Integrity Check:** After replication, an integrity check is performed to ensure the integrity of the replicated data.
I(Ri′)=I(Di)
This equation ensures that the integrity of the replicated data matches the integrity of the original data.

**Versioning:** The versioning mechanism ensures that each replica has a unique version identifier.
$V_i' = V_i + 1$

This equation increments the version number.
**Error Handling:** The error handling mechanism detects and resolves errors during replication.
$(R_i'', V_i'') = E(R_i', V_i')$

This equation represents the corrected replicated data and version after error handling.

**Access Control**
An access control mechanism, that is Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) was designed to restrict access to replicated data based on predefined policies.
Mathematical Equation: User Role Assignment:

$assign$

$U \rightarrow R$
*U*: User

*R*: Role

Permission Assignment:
Mathematical Equation:
$assign$

$R \rightarrow P$
Notations:
P: Permission

Pseudocode:

*class RBAC: roles = {}*
*permissions = {}*

*def assign_permission(self, role, permission): # Assign permission to a role*
*roles[role].add_permission(permission)*

*class ABAC: attributes = {} policies = [] def assign_attribute(self, user, attribute): # Assign attribute to a user attributes[user].add(attribute)*

*def add_policy(self, conditions, actions):*
*# Add a policy with specified conditions and actions policies.append({"conditions": conditions, "actions": actions})*

*class PEP:*
*def enforce_policy(self, user_attributes, requested_action):*
*# Enforce policies based on user attributes and requested action decision = PDP.evaluate_policies(user_attributes, requested_action) if decision == "Allow":*
*# Grant access*
*return "Access Granted" else:*
*# Deny access*
*return "Access Denied"*

*class PDP: policies = []*

*def add_policy(self, conditions, actions):*
*# Add a policy with specified conditions and actions policies.append({"conditions": conditions, "actions": actions})*

*def evaluate_policies(self, user_attributes, requested_action):*
*# Evaluate policies based on user attributes and requested action for policy in policies:*
*if satisfies_conditions(user_attributes, policy["conditions"]) and \ requested_action in policy["actions"]:*
*return "Allow" return "Deny"*
*def satisfies_conditions(user_attributes, conditions): # Check if user attributes satisfy the conditions for condition in conditions:*
*if condition not in user_attributes: return False*
*return True*

*class DataReplication:*
*def replicate_data(self, data, destination):*
*# Replicate data to the specified destination*

*# Include mechanisms for ensuring data confidentiality and integrity encrypted_data = encrypt(data)*
*hash_value = calculate_hash(data)*

*# Send encrypted_data and hash_value to the destination # Verify integrity at the destination*

*def encrypt(data):*
*# Implement encryption algorithm pass*

*def calculate_hash(data):*
*# Implement hash calculation algorithm pass*

*# Static Analysis*
*# - Perform consistency checks on RBAC and ABAC configurations # - Analyse policies for completeness and conflicts*

*# Dynamic Analysis*
*# - Simulate access requests with varying user attributes and actions # - Evaluate the dynamic behaviour of policies in PDP*

*# Performance Testing*
*# - Measure response time, throughput, and resource utilization during data replication # -*

## RESULTS

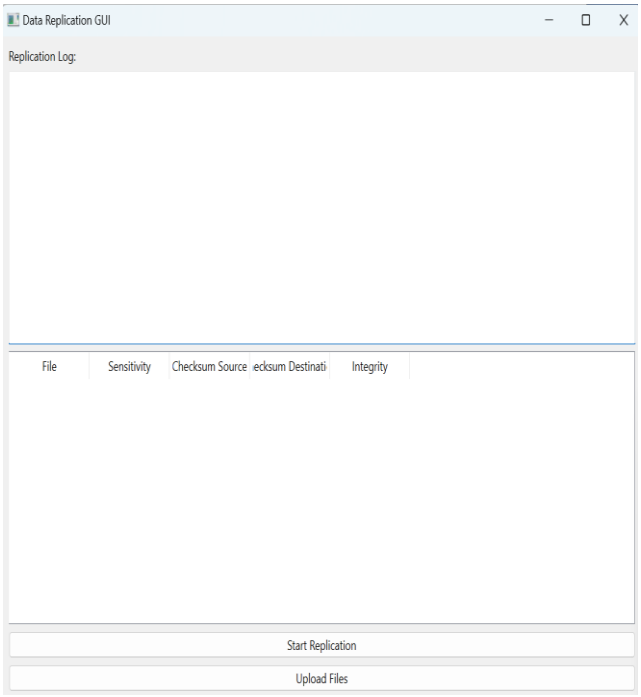


**Figure 3 - Data replication Screen**

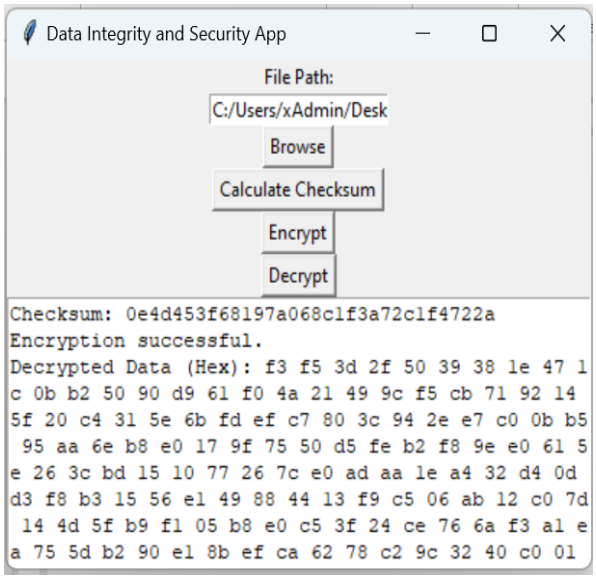


**Figure 4: Encryption/Decryption**

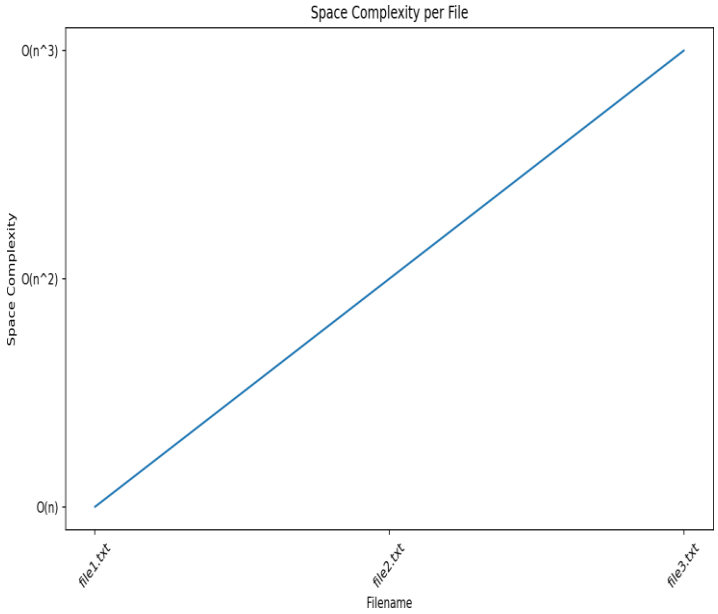


**Figure 5: Space Complexity**

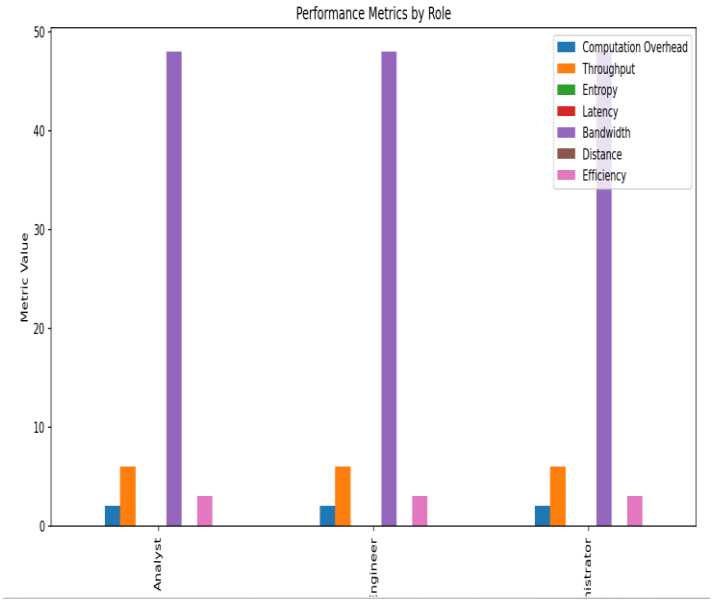


**Figure 6: Performance Metrics**

Each user with specific attributes is given a role /assignment to carry out assigned tasks.

Results evaluation was based on:

**Performance Metrics**
Performance metrics for access control mechanisms are often measured in terms of efficiency, scalability, and response time

**i. Efficiency (E)**
Mathematical Equation

$$E = \frac{\text{Number of Authorized}}{\text{Access Total Number of Attempt}} \times 100$$

**ii. Scalability(S)**

$$S = \frac{\text{Number of Users}}{\text{System Resources Utilization}}$$

Number of Users = Count of user entities.

System Resources Utilization = Measure of system resources consumed

**iii. Response Time (RT) is given as:**

$$\text{RT} = \frac{\text{Total Time Taken for Number of Access Requests}}{\text{Number of Access Requests}}$$

**Conclusion**
Secure data replication in edge and fog computing environments presents unique challenges and opportunities for ensuring confidentiality, integrity, and availability of replicated data. The decentralized nature of these environments, coupled with resource constraints and dynamic network conditions, introduces new security concerns. This paper explores the role of access control mechanisms such as Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) in enhancing security. RBAC

simplifies access control management by assigning roles and granting permissions based on users' responsibilities, while ABAC offers fine-grained access control based on attributes associated with users, resources, and environmental conditions. This integrated approach enables organizations to enforce access policies considering both role-based privileges and attribute-based criteria, enhancing security and mitigating the risk of unauthorized access or data breaches. Various data replication techniques and algorithms were discussed, optimizing data availability, reliability, and performance.

Secure data replication in edge and fog computing environments requires a holistic approach involving access control, data replication, and security. RBAC and ABAC enhance access control, while innovative techniques ensure efficient, reliable data replication in decentralized architectures.

**REFERENCES**

[1] Al Yami, M., & Schaefer, D. (2019). FC as a complementary approach to cloud computing. In *International Conference on Computer and Information Science (ICCIS)*, Jouf University, Al Jouf Region, Kingdom of Saudi Arabia. https://doi.org/10.1109/ICCISci.2019.8716402

[2] Dustdar, S., Avasalcai, C., & Murturi, I. (2019). Edge and FC: vision and research challenges. In *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE. https://doi.org/10.1109/SOSE.2019.00023

[3] Ni, J., Zhang, K., Lin, X., & Shen, X. (2018). Securing fog computing for internet of things applications: challenges and solutions. *IEEE Communications Surveys & Tutorials, 20*(1), 601-628. https://doi.org/10.1109/COMST.2017.2762345

[4] Sookhak, M., Yu, F.R., Khan, .K. & Xiang. (2016). Attribute-Based Data Access Control in Cloud Computing: Taxonomy and Open Issues. Future Generation Computer Systems. Volume 72, July 2017, Pages 273-287

[5] Khan .S.,Parkinson .S. and Qin .Y. (2017). Fog computing security: a review of current applications and security solutions. Journal of Cloud Computing: Advances, Systems and Applications. (2017) 6:19 DOI 10.1186/s13677-017-0090-3

[6] Pfandzelter, T., Japke, N., Schirmer, T., Hasenburg, J., & Bermbach, D. (2023). Managing data replication and distribution in the fog with FReD. *Software: Practice and Experience, 53*, 1958 - 1981.

[7] Alshiky .A.M., Buhari .S., and Barnawi A. (2017). Attribute Based Access Control (ABAC) for HER in Fog Computing Environment. International Journal on Cloud Computing: Services and Architecture (IJCCSA) Vol. 7, No. 1, February 2017. DOI:10.5121/ijccsa.2017.7102

[8] Noura .H., Salman .O, Chehab .A., Couturier .R. (2019). Preserving data security in distributed fog computing. Ad Hoc Networks, 2019, 94, pp.101937 (16). ffhal-02366770f.

[9] Ometov, A., Molua, O. L., Komarov, M., & Nurmi, J. (2022). A Survey of Security in Cloud, Edge, and Fog Computing. *Sensors*, *22*(3), 927. https://doi.org/10.3390/s22030927

[10] Saleem, Ahsan & Khan, Abid & Malik, Saif & Pervaiz, Haris & Malik, Hassan & Alam, Muhammad & Jindal, Anish. (2019). FESDA: Fog-Enabled Secure Data Aggregation in Smart Grid IoT Network. IEEE Internet of Things Journal. PP. 1-1. 10.1109/JIOT.2019.2957314.

[11] Kongara, Ravindranath & Reddy, M.S. & Reddy, M. & Chaitanya, D.. (2019). Secure data storage and retrieval in the cloud. International Journal of Innovative Technology and Exploring Engineering. 8. 34-38.

12] Zhu, T., Guo, J., Zhou, H., Zhou, X., & Zhou, A.-Y. (2018). Consistency and availability in distributed database systems. *Ruan Jian Xue Bao/Journal of Software*, *29*(1), 131-149. https://doi.org/10.13328/j.cnki.jos.005433

[13] Roman, R., Lopez, J., & Mambo, M. (2018). Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges. *Future Generation Computer Systems*, *78*, 680-698. https://doi.org/10.1016/j.future.2016.11.009