International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4720-4726
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0620

SecureSphere: A Cloud-Native Big Data Security Framework Using AWS Key Management Service for Amazon S3

Mithra Ural[1], B.S. B.M Bhavya[2], Rajesh H. N[3], Shreyas V. K[4]
[1,3,4]Master of Computer Applications Student, Department of Master of Computer Applications, PES College of Engineering, Mandya, 57401, India
[2]Assistant Professor, Department of Master of Computer Applications, PES College of Engineering, Mandya, 57401, India
Emails: *mithraural6@gmail.com[1], bhavyabm@pesce.ac.in[2], rhn25560@gmail.com[3], vshreyas666@gmail.com[4]*

Abstract
The exponential growth of big data coupled with the rapid transition to public cloud storage has created major concerns regarding data confidentiality and integrity. Unsecured cloud storage buckets are vulnerable to unauthorized access and severe compliance violations. This paper presents SecureSphere, a cloud-native security framework designed to protect big data at rest within Amazon Simple Storage Service (S3) by integrating AWS Key Management Service (KMS). We propose a centralized key management and envelope encryption model where S3 objects are encrypted dynamically using data keys generated and managed by AWS KMS. Access to the cryptographic keys and stored assets is governed by strict AWS Identity and Access Management (IAM) policies. By enforcing automated key rotation, comprehensive audit logging via AWS CloudTrail, and department-level folder isolation, SecureSphere provides a robust security blueprint. Experimental results demonstrate that the proposed framework achieves regulatory compliance with negligible administrative overhead and near-zero encryption latency, making it highly suitable for large-scale enterprise deployments.

Keywords: *Big Data Security, Cloud Computing, AWS KMS, Amazon S3, Key Management, Envelope Encryption, Regulatory Compliance.*

1. Introduction
The rapid growth of Big Data and the widespread adoption of cloud computing have transformed the way organizations store, process, and manage information. Cloud platforms offer scalability, flexibility, and cost efficiency, making them an ideal choice for handling large volumes of data. However, storing sensitive data in the cloud introduces serious security and privacy challenges, such as unauthorized access, data breaches, and improper key management. A major concern in cloud-based data storage is ensuring that sensitive information remains confidential and accessible only to authorized users. Although encryption is widely used to protect data, encryption alone is not sufficient without a secure and centralized mechanism for managing encryption keys. Poor key management can lead to data exposure, compliance violations, and increased legal risk. To address these challenges, this paper proposes SecureSphere, a cloud-native security approach that leverages Amazon Web Services (AWS) and its Key Management Service (KMS). Big Data stored in Amazon S3 is encrypted using AWS KMS, which securely generates, stores, and controls encryption keys, while access to encrypted data is governed by strict IAM permission policies organized into department-wise IAM groups. The proposed solution enhances data confidentiality, integrity, and regulatory compliance while maintaining scalability and performance. The key contributions of this work are: (1) a centralized envelope-encryption architecture for Big Data at rest in Amazon S3 using AWS KMS; (2) a department-wise, least-privilege IAM access-control model enforced through custom JSON policies; (3) integration of Multi-Factor Authentication (MFA) and automated audit logging for regulatory

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4720-4726
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0620

compliance; and (4) an experimental evaluation validating access isolation, encryption correctness, and administrative overhead.

2. Literature Review

Huang [1] presents a comparative survey of cloud-based Key Management Services, evaluating solutions on security, scalability, key lifecycle management, and compliance. The study highlights hardware-based key protection and centralized key management as means of simplifying encryption, but notes that manual key management increases latency and administrative overhead in large cloud environments. Reddy [2] surveys the major security challenges of storing and processing Big Data in cloud environments, including unauthorized access, insider attacks, and privacy leakage. The study concludes that encryption alone cannot fully protect cloud data unless supported by proper governance, monitoring, and identity management. Ashok and Judgi [3] propose a dynamic key management framework based on Key-Aggregate Encryption that supports automatic key rotation and policy-based access control. Experimental results show reduced computational complexity relative to static key management, though implementation complexity increases in highly distributed infrastructures. Filaly et al. [4] investigate privacy and security issues in Hadoop-based Big Data systems, analyzing encryption, authentication, and access-control mechanisms across HDFS and MapReduce. The authors identify key management as a significant open research problem and recommend integrating cloud-native key management services. Sun Lei et al. [5] propose a Cloud Key Management Infrastructure that centralizes key generation, storage, distribution, and lifecycle management across cloud platforms, demonstrating low operational overhead while maintaining strong protection against key compromise. Kuzminykh et al. [6] present a comparative analysis of enterprise cryptographic key management systems, including HashiCorp Vault and OpenStack Barbican, concluding that centralized key management improves auditing, access control, and operational efficiency, though the optimal choice of system depends on organizational scale. Egorov et al. [7] introduce a decentralized key management system based on proxy re-encryption that enables secure data sharing without exposing encryption keys to a central authority, improving user privacy at the cost of higher implementation and deployment complexity. Bicakci et al. [8] propose a secure storage framework that separates encrypted files and encryption keys across multiple cloud providers using client-side encryption, improving usability while assuming that providers do not collude. Ghosal and Conti [9] review key management techniques for Smart Grid Advanced Metering Infrastructure, identifying scalability, dynamic membership, and secure key updates as major open challenges for distributed, cloud-connected systems. Venkatesh Gopal et al. [10] examine the relationship between encryption, key lifecycle management, and enterprise cloud security, emphasizing that key governance integrated with organizational policy significantly reduces the risk of unauthorized access and supports regulatory compliance with standards such as GDPR and PCI-DSS. The Cloud Security Alliance [11] provides comprehensive guidance on securing cloud environments through identity management, encryption, access control, and continuous monitoring, recommending centralized key management and regular security audits as core best practices.

3. Research Gap

The reviewed literature consistently shows that encryption alone is insufficient for protecting Big Data in the cloud, and that key management is the central unresolved challenge [1], [2], [5]. Manual and decentralized key management increases administrative overhead and latency [1], while dynamic and decentralized key schemes improve confidentiality at the cost of significant implementation complexity [3], [7], [8]. Existing surveys further show that most frameworks address either encryption or access control in isolation, without a unified, department-level, least-privilege enforcement model [4], [9], [10]. Consequently, a clear gap exists for a framework that combines centralized envelope encryption, fine-grained department-wise access isolation, automated key rotation, audit logging, and Multi-Factor Authentication using only native cloud services, without third-party middleware. SecureSphere is

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4720-4726
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0620

designed to close this gap using Amazon S3, AWS KMS, and AWS IAM.

4. Existing System

Existing cloud-based data storage systems typically store Big Data with basic security mechanisms such as simple access control and optional encryption. In many deployments, encryption is either not enabled by default or is managed manually, and encryption keys are often stored alongside application code or managed in an unstructured, decentralized manner, increasing the risk of key exposure. Most existing systems rely on traditional perimeter security and user authentication alone, lacking centralized key management, automated key rotation, and proper auditing. These systems are unable to enforce department-level data segregation and do not provide granular access control at the folder or object level within cloud storage, leaving them vulnerable to insider threats, accidental data exposure, and compliance failures.

5. Proposed System

5.1. System Overview

SecureSphere securely stores organizational data in Amazon S3, where all uploaded files are automatically encrypted using AWS KMS. IAM authenticates users and enforces role-based access control for five departments — HR, Finance, Doctor, Laboratory, and Pharmacy.

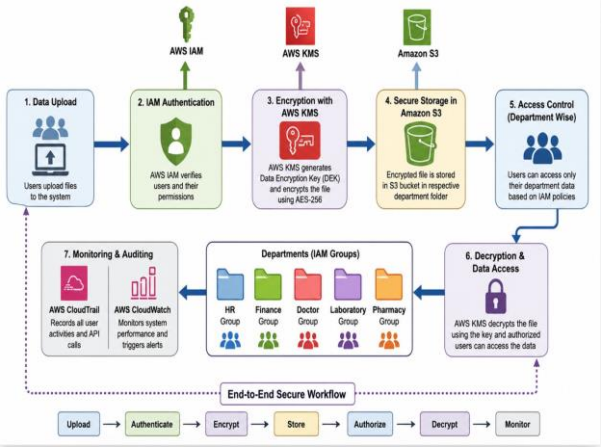

Figure 1 Proposed system architecture of SecureSphere

AWS KMS generates and manages Customer Managed Keys (CMKs) and Data Encryption Keys (DEKs) to secure encryption and decryption of files, while AWS CloudTrail records user activity and CloudWatch monitors system performance, enabling the organization to prevent unauthorized access and maintain secure, centralized cloud storage shown in Figure 1.

5.2. Objectives

The specific objectives of SecureSphere are to: implement AWS KMS-based encryption for data stored in Amazon S3; configure Role-Based Access Control using IAM Groups, Users, and custom JSON policies; ensure that users can access only their respective department folders; enable Multi-Factor Authentication for identity verification; achieve regulatory compliance with GDPR, HIPAA, and PCI-DSS; and provide an auditable, scalable, and manageable cloud security solution.

5.3. Module Description

The Admin module has full access to the S3 bucket, IAM configuration, and the KMS key; it can manage bucket policies, create or delete IAM users and groups, update IAM policies, and monitor access logs through CloudTrail. Each departmental IAM user belongs to exactly one IAM group and can access only the folder assigned to that department, as summarized in Table 1.

Table 1 IAM User–Group–Folder Mapping

IAM User	IAM Group	Authorized Folder
hruser	HRGroup	EmployeeData/
financeuser	FinanceGroup	FinanceData/
doctoruser	DoctorGroup	PatientData/
laboratoryuser	LaboratoryGroup	LaboratoryData/
pharmacyuser	PharmacyGroup	PharmacyData/

5.4. Functional and Non-Functional Requirements

Functionally, the system requires console sign-in with MFA, department-scoped bucket listing, file access restricted to the authorized folder, KMS-based decryption on authorized read requests, explicit denial of cross-department access, and a strict prohibition on upload, delete, or modify operations

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4720-4726
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0620

for departmental users. Non-functionally, the system targets 99.99% availability through AWS managed services, linear scalability to additional departments, full auditability through CloudTrail, and compliance with GDPR, HIPAA, and PCI-DSS through encryption, access control, and audit logging.

6. Methodology and Implementation

6.1. Implementation Workflow

- **AWS Cloud Environment Setup:** An AWS account is created and Amazon S3, AWS KMS, IAM, CloudTrail, and CloudWatch are initialized as the foundation for secure storage, key management, and access control.
- **IAM Users and Departments:** Five departments — HR, Finance, Doctor, Laboratory, and Pharmacy — are created as separate IAM groups, each assigned least-privilege, role-based permissions.
- **Amazon S3 Bucket Configuration:** A single S3 bucket is created with a dedicated folder (key prefix) per department, with bucket policies configured to prevent unauthorized access.
- **AWS KMS Key Generation and Encryption:** A Customer Managed Key is created in KMS; during upload, KMS generates AES-256 Data Encryption Keys and files are encrypted using Server-Side Encryption with KMS (SSE-KMS).
- **Secure File Upload and Storage:** Files are encrypted before storage and placed in the corresponding department folder, ensuring confidentiality throughout the storage lifecycle [11-16].
- **Secure Data Access and Authorization:** On access, IAM verifies user identity and permissions; if authorized, KMS decrypts the requested file, otherwise access is denied.
- **Monitoring, Auditing, and Validation:** CloudTrail records all login, encryption, decryption, and file-access events, while CloudWatch monitors performance and raises alerts.

6.2. S3 Bucket and IAM Policy Design

A single S3 bucket with server-side encryption enabled via a Customer Managed KMS key was created, containing five key-prefix folders — EmployeeData/, FinanceData/, PatientData/, LaboratoryData/, and PharmacyData/. Custom IAM JSON policies were authored for each of the five IAM groups, each combining an Allow statement for console sign-in, an Allow statement for bucket listing scoped by an s3:prefix condition, an Allow statement for GetObject and kms:Decrypt limited to the authorized folder and CMK, an explicit Deny statement for every other department folder, and a Deny statement blocking all write and delete operations. A representative excerpt of the HRGroup policy is shown in Listing 1 shown in Figure 2.

Listing 1. Excerpt of the HRGroup IAM policy (JSON).

```
{ "Version": "2012-10-17", "Statement": [
{ "Sid": "AllowBucketListingForConsole", "Effect":
"Allow",
"Action": "s3:ListBucket",
"Resource": "arn:aws:s3:::securesphere-bucket",
"Condition": { "StringLike": { "s3:prefix":
["", "EmployeeData", "EmployeeData/*"] } } },
{ "Sid": "AllowGetObjectEmployeeData", "Effect":
"Allow",
"Action": ["s3:GetObject"],
"Resource": "arn:aws:s3:::securesphere-
bucket/EmployeeData/*" },
{ "Sid": "AllowKMSDecrypt", "Effect": "Allow",
"Action": ["kms:Decrypt",
"kms:GenerateDataKey"],
"Resource": "arn:aws:kms:region:acct:key/CMK-
ID" },
{ "Sid": "ExplicitDenyOtherDepartments", "Effect":
"Deny",
"Action": "s3:*",
"Resource": ["arn:aws:s3:::securesphere-
bucket/FinanceData/*",
"arn:aws:s3:::securesphere-bucket/PatientData/*",
"..."] },
{ "Sid": "DenyAllWriteOperations", "Effect":
"Deny",
"Action": ["s3:PutObject", "s3:DeleteObject"],
"Resource": "arn:aws:s3:::securesphere-bucket/*" }
] }
```

Figure 2 Listing 1. Excerpt of the HRGroup IAM policy (JSON)

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4720-4726
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0620

6.3. MFA Enforcement and Auditing

Virtual MFA devices were configured for each IAM user using a TOTP authenticator application. A policy condition using aws:MultiFactorAuthPresent = false with Effect: Deny was incorporated into the AdminGroup policy to require a valid MFA challenge for administrative operations, preventing token-only access by compromised credentials. All access events, including login attempts, encryption, decryption, and file-access operations, are logged through AWS CloudTrail for compliance auditing.

6.4. Encryption and Access Sequence

During upload, the client requests a Data Encryption Key from AWS KMS; KMS generates the DEK and returns it to S3, which encrypts the file and stores it in the corresponding department folder before returning a confirmation to the user. During a read request, IAM first evaluates the ListBucket and GetObject policy conditions; if the user is authorized, S3 issues a kms:Decrypt request, and the file is decrypted and served only after IAM permission is confirmed. Figure 3 illustrates this sequence for the upload path.

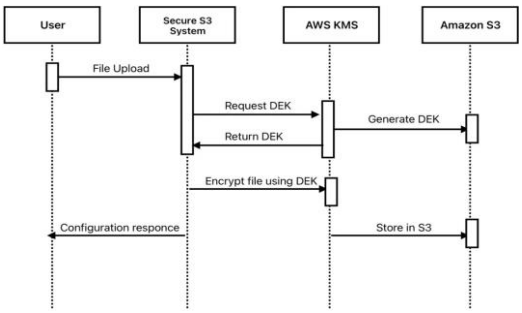

Figure 3 Sequence diagram of the KMS-based encryption flow during file upload.

7. Results and Discussion

7.1. Test Summary

System testing validated all access-control rules, KMS encryption integration, MFA enforcement, and deny policies by signing in as each IAM user and attempting to access all five S3 folders, perform write operations, and download encrypted files. Table 2 summarizes representative unit and integration test cases; all ten cases passed, confirming complete policy enforcement across all five departmental roles.

Table 2 Representative Test Cases And Results

Test	Description	Expected Result	Status
TC01	hruser accesses EmployeeData	Access granted	Pass
TC02	hruser attempts FinanceData access	Access denied	Pass
TC03	doctoruser opens PatientData file	KMS decrypt succeeds	Pass
TC04	KMS decrypt without permission	Decrypt denied	Pass
TC05	Any user attempts file delete	Delete denied	Pass

7.2.Access Control Matrix

Table 3 presents the complete access matrix obtained during system-level testing across all five IAM groups and folders, confirming strict one-to-one department isolation with all write and delete operations denied uniformly.

Table 3 Complete Access Matrix — All Five Groups

IAM User	Employee	Finance	Patient	Lab	Pharmacy	Write
hruser	Allow	Deny	Deny	Deny	Deny	Deny
financeuser	Deny	Allow	Deny	Deny	Deny	Deny
doctoruser	Deny	Deny	Allow	Deny	Deny	Deny
laboratoryuser	Deny	Deny	Deny	Allow	Deny	Deny
pharmacyuser	Deny	Deny	Deny	Deny	Allow	Deny

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4720-4726
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0620

7.3. Discussion

The results confirm that department-level least-privilege isolation can be achieved using native IAM policy conditions without third-party middleware, directly addressing the research gap identified in Section III. Because encryption and decryption are handled transparently by AWS KMS at the point of storage access, the framework introduces near-zero perceptible encryption latency for authorized users while adding negligible administrative overhead, since key rotation, auditing, and access enforcement are automated through managed AWS services. These outcomes support the framework's suitability for regulated, multi-department enterprise deployments subject to GDPR, HIPAA, and PCI-DSS.

Conclusion and Future Work

This paper presented SecureSphere, a cloud-native, role-based access control framework for securing multi-department Big Data on Amazon S3 using AWS KMS. The system integrates five IAM groups with least-privilege JSON policies, KMS server-side encryption, and Multi-Factor Authentication to form a layered security architecture. Experimental validation confirmed that each departmental IAM user can access exclusively its authorized S3 folder, that all cross-department access attempts are blocked, and that all write and delete operations are uniformly denied for departmental users, while KMS integration ensures that data remains encrypted at rest and decryptable only by authorized users. The results demonstrate that enterprise-grade, HIPAA- and GDPR-compliant data governance can be achieved using native AWS services at minimal operational cost. Future work will extend SecureSphere with AWS CloudTrail-based automated compliance reporting, Attribute-Based Access Control (ABAC) using IAM session tags for dynamic, tag-based access conditions, and AWS Config rules for continuous validation of IAM and S3 configurations against organizational security baselines.

References

[1]. X. Huang, "A Survey of Key Management Service in Cloud," Int. J. Cloud Computing Security, 2020.
[2]. Y. B. Reddy, "Big Data Security in Cloud Environment," J. Network and Computer Security, 2021.
[3]. N. Ashok and T. Judgi, "Advanced Techniques in Dynamic Key Management and Multilayered Security for Cloud Infrastructures," Int. J. Cybersecurity, 2024.
[4]. W. Filaly, A. Berros, and H. Elmendili, "A Comprehensive Survey on Big Data Privacy and Hadoop Security," IEEE Access, 2025.
[5]. Sun Lei, Dai Zishan, and Guo Jindi, "Research on Key Management Infrastructure in Cloud Computing Environment," in Proc. IEEE Int. Conf. Cloud Computing, 2010.
[6]. V. Kuzminykh et al., "Enterprise Cryptographic Key Management Systems: A Comparative Survey," Int. J. Information Security Science, 2019.
[7]. D. Egorov et al., "NuCypher KMS: Decentralized Key Management System," arXiv preprint, 2017.
[8]. K. Bicakci et al., "TwinCloud: Secure Cloud Storage with Split Key Management," in Proc. Int. Conf. Cloud Security, 2013.
[9]. A. Ghosal and M. Conti, "Key Management Systems for Smart Grid Advanced Metering Infrastructure: A Survey," IEEE Commun. Surveys Tuts., 2019.
[10]. Venkatesh Gopal et al., "Encryption and Key Management in Enterprise Cloud Security," J. Enterprise Cloud Security, 2022.
[11]. Cloud Security Alliance, "Security Guidance for Critical Areas of Focus in Cloud Computing," CSA, 2017.
[12]. Amazon Web Services, "AWS Identity and Access Management User Guide," AWS Documentation, 2024. [Online]. Available: https://docs.aws.amazon.com/IAM/latest/UserGuide/
[13]. Amazon Web Services, "Amazon S3 Security Best Practices," AWS Documentation, 2024. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/security-best-practices.html
[14]. Amazon Web Services, "AWS Key Management Service Developer Guide — Overview," 2024. [Online]. Available: https://docs.aws.amazon.com/kms/latest/developerguide/overview.html

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 07 July 2026
Page No: 4720-4726
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0620

[15]. V. C. Hu et al., "Guide to Attribute Based Access Control (ABAC) Definition and Considerations," NIST SP 800-162, Jan. 2014.
[16]. E. Barker, "Recommendation for Key Management," NIST SP 800-57 Part 1 Rev. 5, 2020.