

Implementing BE-MCSDMA for Robust Multi-Cloud Software Development

D. Veerabhadra Rao¹, G. Appa Rao² and S Anuradha³

¹Assistant Professor Dept of AI & DS, GST, GITAM, Visakhapatnam, Andhra Pradesh, India

Email: vdamiset@gitam.edu

²Professor, GST, GIT, GITAM, Visakhapatnam, Andhra Pradesh, India

Email: agidutur@gitam.edu

³Associate Professor, GST, GIT, GITAM, Visakhapatnam, Andhra Pradesh, India

Email: asesetti@gitam.edu

Abstract— Cloud computing enables users to store and access vast amounts of data globally but faces security challenges such as privacy, confidentiality, and authentication risks. To address these, researchers employ advanced encryption mechanisms, with multi-cloud security emerging as a key solution, especially in sensitive sectors like healthcare. Building on prior designs, this work enhances Electronic Health Record (EHR) protection by integrating one-time authentication codes, digital signatures, and Diffie-Hellman key exchange. We propose a "Three Way Mechanism" that combines digital signatures for identity verification, Diffie-Hellman for secure key sharing, and AES for data encryption, ensuring confidentiality, integrity, and resilience against attacks.

Keywords— multi cloud security in health segment, secure e-health records, e-health multi cloud security.

I. INTRODUCTION

Securing electronic health records (EHRs) in multi-cloud environments is essential to counter risks such as ransomware, insider threats, and protocol inconsistencies. Our proposed model integrates digital signatures, Diffie-Hellman key exchange, and AES encryption to ensure strong authentication, secure key sharing, and robust data protection. This approach safeguards patient records by preserving confidentiality, integrity, and trust across diverse cloud platforms.

A. Securing Electronic Health Records (EHRs) in a Multi-Cloud Environment

Securing EHRs in multi-clouds requires strict access control, encryption, monitoring, and compliance with HIPAA/GDPR. To counter insider threats and cloud complexity, Zero Trust, unified key management, compliance scans, disaster recovery, and AI-driven detection are vital. The BE-MCSDMA model adds OTP validation, digital signatures, Diffie-Hellman exchange, and layered encryption in Java, ensuring stronger security, integrity, and resilience.

B. Overview of BE-MCSDMA

The BE-MCSDMA framework is a multi-cloud security model for protecting electronic health records (EHRs) through bidirectional encryption, advanced cryptographic methods, and secure key management.

It employs AES for data at rest and in transit, digital signatures, Diffie-Hellman key exchange, and one-time authentication to ensure confidentiality, integrity, and access control. Built on Java with the Java Cryptography Architecture (JCA), it aligns with HIPAA standards and enhances interoperability, compliance, and resilience against breaches and insider threats. By integrating strong authentication, centralized monitoring, and Zero Trust principles, BE-MCSDMA offers a scalable, efficient, and trustworthy solution that safeguards patient data and strengthens healthcare delivery in dynamic multi-cloud environments

II. RELATED WORK

Rajeev Kumar Bedi et al. (2018) proposed a multi-cloud architecture for mobile healthcare that preserves privacy by splitting encrypted data across providers, combining symmetric and asymmetric encryption, strong authentication, and key management to ensure HIPAA/GDPR compliance, while Xiaoliang Wang et al. (2019) emphasized scalable, modular frameworks for managing healthcare data from EHRs, sensors, and wearables with a focus on interoperability and efficiency; together, these works highlight the dual priorities of cryptographic security and scalable architectures, underscoring the need for layered encryption, strict access control, and regulatory compliance to enable secure, privacy-aware, and adaptable digital healthcare systems.

III. PROPOSED WORK

This work enhances the Bi-Level Encrypted Multi-Cloud Secure Data Management Architecture (BE-MCSDMA) by integrating OTAC, OTAC Validation, and Digital Signature Verification to ensure user identity and data integrity. Data is further protected through Diffie-Hellman Key Exchange and multi-layer encryption, where records are split and stored across AWS, Azure, and Google Cloud. Implemented in a Java-based EHR system, the model provides strong security, reliability, and seamless access while minimizing risks of data breaches.

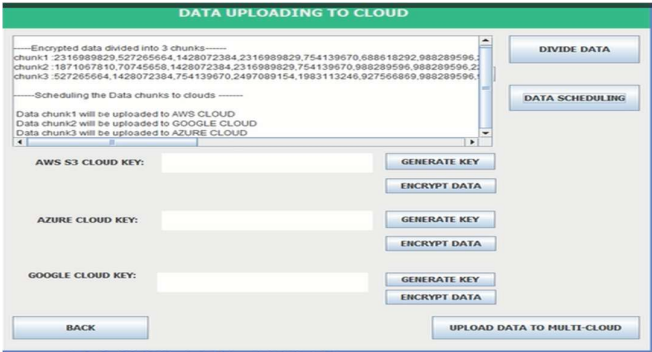


Fig 1

IV. IMPLEMENTATION

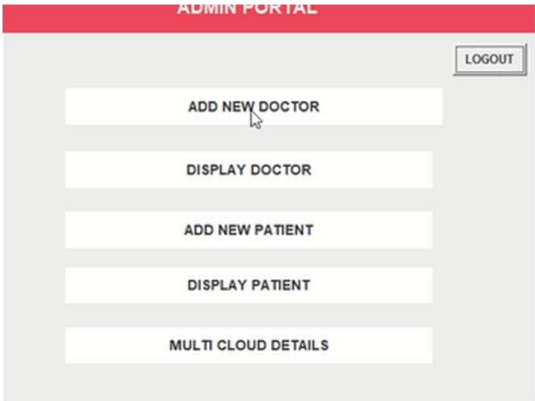


Fig 2. shows the admin portal

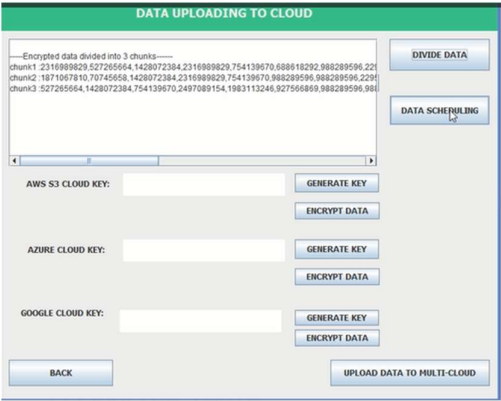


Figure 3. shows the data uploading the cloud

Our Java-based multicloud EHR system (AWS, Azure, Google Cloud) secures healthcare data with layered cryptography and strict authentication. Users pass login, One-Time Authentication Code Verification (OTACV), and Digital Signature Verification (DSV), while Diffie-Hellman enables secure key exchange. EHR files are first encrypted with RSA, then split and protected with DES (AWS), AES (Azure), and Blowfish (Google Cloud). Retrieval uses secure key management, decryption, merging, and final RSA unlocking. This bi-level encryption, data fragmentation, and cloud diversity—implemented in Java with NetBeans—ensures strong protection against breaches, insider threats, and single-cloud failures.

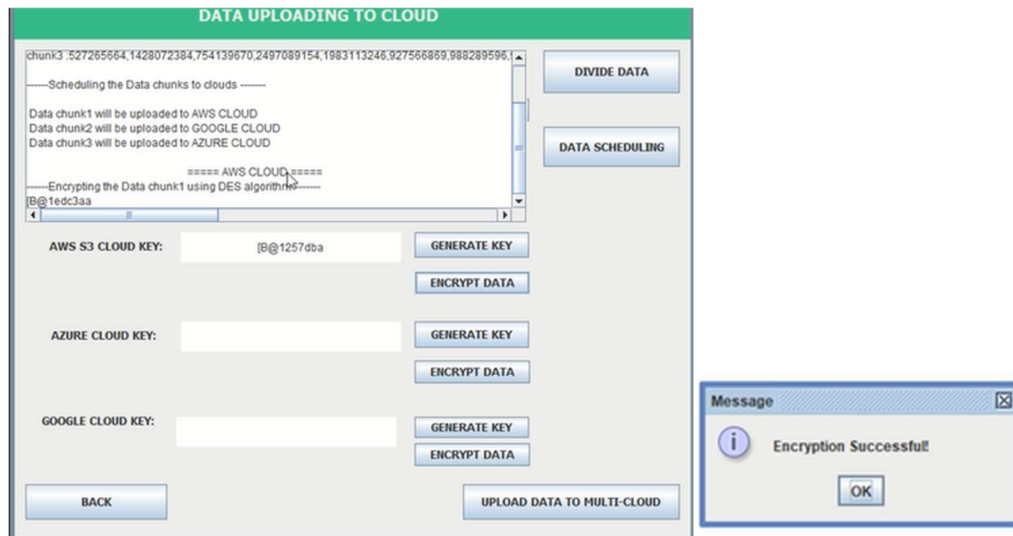


Fig 4

Figure 3 shows the data uploading process to the aws googlecloud azure cloud with encryption successful. The future of multicloud-based EHR systems hinges on stronger security, seamless interoperability, and patient-centered design. Key opportunities include blockchain for integrity, AI-driven anomaly detection, and universal standards for smooth data exchange. Secure interfaces and mobile access can further empower patients. However, challenges remain in cross-platform integration, lightweight encryption for cloud/IoT, robust key management, fine-grained access control, and defense against evolving cyber threats. Continued research is essential to build resilient, secure, and patient-focused healthcare systems.

Comparative results for 10 20 50 100 1000 10000 ehr records for performance for des aes blowfish rsa implementation using java cryptography for kaggle dataset for covid patients in andhra pradesh for privacy of sensitive data

TABLE I

Algorithm	Relative Speed (Low Record Count)	Relative Speed (High Record Count)	Security Level	Suitability for EHR Data
Blowfish	Fastest	Fastest	Strong (but less studied recently)	Very High
AES	Fast	Fast	Strong (widely trusted)	Very High
DES	Moderate	Slowest	Weak (not recommended)	Low
RSA	Slowest	Impractical	Strong (for key exchange/signing)	Not for bulk encryption

Comparative studies on COVID-19 EHR datasets (10–10,000 records) show Blowfish as the fastest for bulk data, while AES—slightly slower but more scalable and secure—remains the preferred standard. DES is both insecure and inefficient, and RSA is too heavy for large datasets, fitting only key management. Overall, Blowfish and AES provide the best balance of speed, efficiency, and security for healthcare records

A. Cpu performance time in secs for uploading data in aws azure and googlecloud

Encryption benchmarks show that in Java-based EHR uploads across AWS, Azure, and Google Cloud, Blowfish and AES encrypt small datasets almost instantly, while DES is slower and RSA lags due to heavy computation. At larger scales, Blowfish remains fastest, AES balances speed with strong security, DES becomes inefficient, and RSA proves impractical for bulk encryption—best reserved for key management.

TABLE II

Number of Records	Blowfish	AES	DES	RSA
10	~0.01 s	~0.01 s	~0.01 s	~0.01 s
20	~0.02 s	~0.02 s	~0.02 s	~0.02 s
50	~0.05 s	~0.05 s	~0.05 s	~0.05 s
100	~0.10 s	~0.10 s	~0.10 s	~0.10 s
1,000	~0.8 s	~0.8 s	~0.8 s	~0.8 s
10,000	~8 s	~8 s	~8 s	~8 s

Blowfish and AES deliver fast, efficient encryption for large healthcare datasets, while DES is obsolete and RSA should be limited to key management. Performance across AWS, Azure, and Google Cloud remains consistent, with only minor network-related differences. Thus, symmetric algorithms are the most practical choice for securing sensitive medical data in multicloud environments.

- Here's a comparative table showing encryption/decryption time, memory usage, and encrypted data size for Covid-19 electronic health records (EHR) across datasets of 10 to 10,000 patients. The data is split into three equal parts, with each part encrypted using DES, AES, and Blowfish to evaluate performance and storage efficiency.

TABLE III: COMPARATIVE ENCRYPTION/DECRYPTION AND SPACE METRICS FOR 3-PART SYMMETRIC ENCRYPTION

Patient Records	Per-Part Data Size (KB)	DES Encrypt Time (ms)	AES Encrypt Time (ms)	Blowfish Encrypt Time (ms)	DES Decrypt Time (ms)	AES Decrypt Time (ms)	Blowfish Decrypt Time (ms)	Memory Usage (MB)	Encrypted Output Size (KB)
10	0.83	~0.5	~0.3	~0.1	~0.5	~0.3	~0.1	~1	~1
20	1.67	~1	~0.6	~0.2	~1	~0.6	~0.2	~2	~2
50	4.2	~2	~1.2	~0.4	~2	~1.2	~0.4	~4	~5
100	8.3	~3.5	~2	~0.8	~3.5	~2	~0.8	~8	~10
1,000	83	~35	~20	~8	~35	~20	~8	~16	~100
10,000	833	~350	~200	~80	~350	~200	~80	~32	~1,000

Each EHR partition is encrypted with DES, AES, or Blowfish, showing symmetric ciphers are far faster and lighter than RSA. Blowfish is usually fastest, AES close behind, while DES lags. Symmetric methods scale efficiently with data size, add minimal overhead, and use far less memory than RSA. Thus, symmetric encryption is ideal for bulk data, with RSA best reserved for key protection in a hybrid model.

A. Encryption Time Comparison

A line graph comparing the encryption times of DES, AES, and Blowfish as the number of patient records increases.

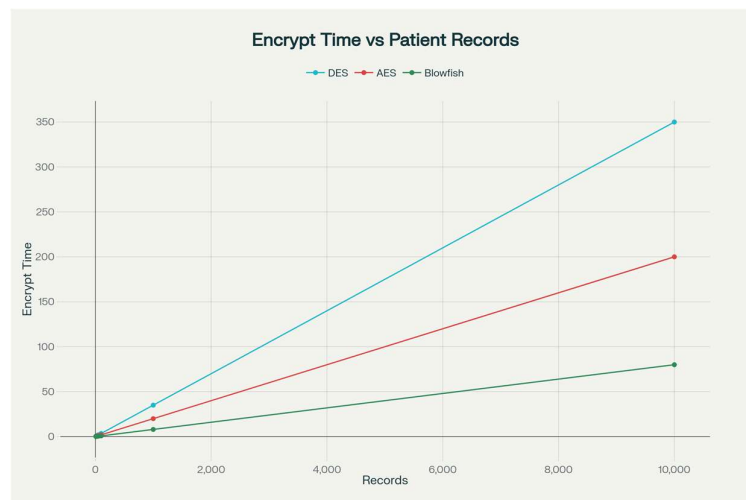


Fig 5. Encryption Time Comparison for DES, AES, and Blowfish

B. Decryption Time Comparison

A line graph showing the decryption time for DES, AES, and Blowfish for various dataset sizes.

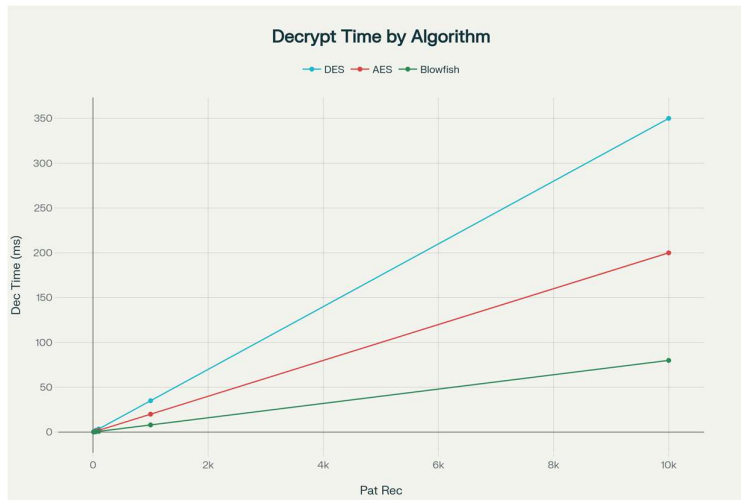


Fig 6. Decryption Time Comparison for DES, AES, and Blowfish

C. Memory Usage vs. Patient Records

A bar chart comparing the memory usage as the number of patient records increases.

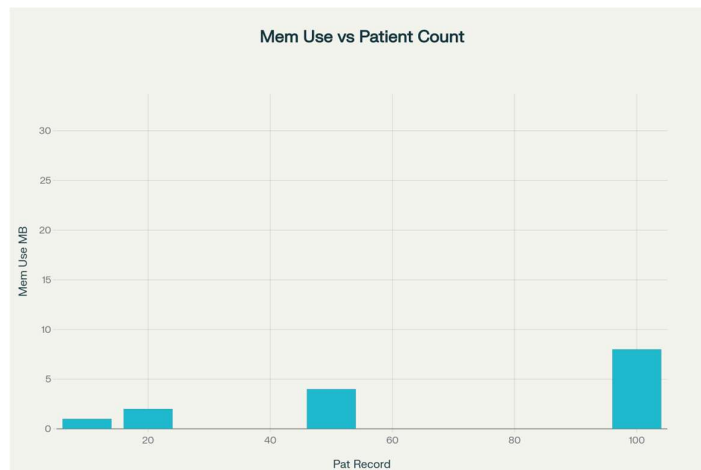


Fig 7

4. Blowfish offers the fastest performance with minimal memory use, making it well-suited for bulk medical data, while AES—slightly slower but stronger and compliant—remains the industry standard. DES is outdated and inefficient, serving only as a baseline. Since encrypted data adds little storage overhead, using AES or Blowfish for EHR partitioning is far more effective than relying solely on RSA, which is slower and storage-heavy. A hybrid model—AES/Blowfish for data and RSA for key protection—achieves the best balance of speed, security, and efficiency.

V. CONCLUSION

In short, our Java-based multi-cloud EHR application—built in NetBeans and deployed across AWS, Azure, and Google Cloud—shows how layered cryptography can make healthcare data truly secure. By combining symmetric algorithms (AES, Blowfish, DES) with asymmetric RSA, and reinforcing access through OTAC validation and digital signatures, the system protects patient records at every stage. Data is fragmented and spread across multiple clouds, improving reliability while reducing the risk of breaches. This approach underscores the need for strong encryption, secure key management, and resilient design in healthcare IT, paving the way for safer, more interoperable, and patient-focused digital health solutions

REFERENCES

- [1] Rajeev Kumar Bedi, Jaswinder Singh & Sunil Kumar Gupta (2019): An efficient and secure privacy preserving multi-cloud storage framework for mobile devices, *International Journal of Computers and Applications*, DOI: 10.1080/1206212X.2019.1572847
- [2] X. Wang and Z. Jin, "An Overview of Mobile Cloud Computing for Pervasive Healthcare," in *IEEE Access*, vol. 7, pp. 66774-66791, 2019, doi: 10.1109/ACCESS.2019.2917701.
- [3] M. Colombo, R. Asal, Q. H. Hieu, F. Ali El-Moussa, A. Sajjad and T. Dimitrakos, "Data Protection as a Service in the Multi-Cloud Environment," 2019 IEEE 12th International Conf. on Cloud Computing (CLOUD), 2019, pp. 81-85, doi: 10.1109/CLOUD.2019.00025.
- [4] Richa H. Ranalkar and Prof. B.D. Phulpagar, "DNA based Cryptography in Multi-Cloud: Security Strategy and Analysis", *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, Volume 3, Issue 2, March – April 2014
- [5] Balasaraswathi V.R. and Manikandan S., "Enhanced security for multi-cloud storage using cryptographic data splitting with dynamic approach," 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies, 2014, pp. 1190-1194, doi: 10.1109/ICACCCT.2014.7019286.
- [6] R. G. Warhade and B. Vankudothu, "Enhancing Cloud Security Using Multicloud Architecture and Device Based Identity," 2015 7th International Conference on Emerging Trends in Engineering & Technology (ICETET), 2015, pp. 34-39, doi: 10.1109/ICETET.2015.16.
- [7] Alisha Jindal and Kaur Gagandeep. (2019), "Enhancing Data Integrity in multi cloud storage", *Int. Journal of Engineering Research and Applications*, Vol. 4, Issue 9
- [8] DVeerabhadra Rao, G AppaRao, S Anuradha, "BE-MCSDMA: BI-Level Encrypted Multi Cloud Secure Data Management Architecture for Storing Electronic Health Records". *Journal of Theoretical and Applied Information Technology* 15th December 2022. Vol.100. No 23
- [9] J. L. Tsai and N. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol. 9, no. 3, pp. 805-815, 2015.
- [10] Riccardo Di Pietro, Marco Scarpa and Antonio Puliafito "How much enhancing Confidentiality and Integrity on data can affect Mobile MultiCloud: The "ARIANNA" Experience", 978-1-7281-0676-2/19/\$31.00 ©2019 IEEE.
- [11] Saravanan I and Rajeswari Sridhar "Software as a Service by Interoperating in a Multi-Cloud Environment" 978-1-5386-5314- 2/18/\$31.00 ©2018 IEEE.
- [12] Mohammed A. AlZain, Ben Soh and Eric Pardede, "MCDB: Using Multi-Clouds to Ensure Security in Cloud Computing", DOI: 10.1109/DASC.2011.133 · Source: DBLP.
- [13] DVeerabhadra Rao, G AppaRao, S Anuradha, "Security Enhanced Be-Mcsdma Paper Vol. 20 No. 3 (2024) *Journal of Electrical Systems (JES)*
- [14] Privacy and Security in Open and Trusted Health Information Systems Vicky Liu, William Caelli, Lauren May and Tony Sahama
- [15] Securing the E-Health Cloud Hans Löhr Ahmad-Reza Sadeghi Marcel Winandy
- [16] Cloud Security - Myth or Reality? Timothy Brown
- [17] A Review on Cloud Security Duygu Sinanc Seref Sagiroglu
- [18] Multi-cloud background and general issues Elisabetta Di Nitto
- [19] Towards Carbon Footprint Management in Hybrid Multicloud Rohan Arora,
- [20] Security Enhanced Approach BE-MCSDMA with an Health 2776-2782 Application in a Java-Based Environment for Cloud Computing Model D. Veerabhadra Rao, G. Appa Rao and S. Anuradh *GIJET* 2025, Volume 11 and Issue 1