

Optimized data security and storage using improved blowfish and modular encryption in cloud-based internet of things

Saritha Ibakkanavar Guddappa¹, Sowmyashree Malligehalli Shivakumaraswamy¹,
Naveen Ibakkanavar Guddappa²

¹Department of Electronics and Communication Engineering, BMS Institute of Technology and Management, Bengaluru, India

²Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bengaluru, India

Article Info

Article history:

Received Oct 17, 2024

Revised Jun 17, 2025

Accepted Jul 10, 2025

Keywords:

Cloud-based storage system

Efficient data storage

Improved blowfish algorithm

Internet of things

Modular encryption standard

ABSTRACT

The increasing development of the internet of things (IoT) has made cloud-based storage systems essential for storing, processing, and sharing IoT data. Ensuring cloud security is crucial as it manages a large volume of sensitive and outsourced data vulnerable to unauthorized access. This research proposes an improved blowfish algorithm and modular encryption standard (IBA-MES) for secure and efficient data storage in cloud-based IoT systems. The block cipher structure in improved blowfish algorithm (IBA) enables scaling for different data sizes, ensuring secure data handling across a wide range of IoT devices. Additionally, IBA-MES adaptability helps maintain data integrity, enhancing both the security and efficiency of data storage in cloud-based IoT environments. Modular encryption standard (MES) reduces latency during encryption operations, ensuring quick data transactions between the cloud server and IoT devices. By combining blowfish's speed and strength with modular encryption's adaptability, IBA-MES provides robust data protection. Metrics such as execution time, central processing unit (CPU) usage, encryption time, decryption time, runtime, and latency are calculated for the proposed IBA-MES. For 700 blocks, the IBA-MES achieves encryption and decryption times of 270 and 415 ms, respectively, outperforming the triple data encryption standard (TDES).

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Saritha Ibakkanavar Guddappa

Department of Electronics and Communication Engineering

BMS Institute of Technology and Management

Doddaballapur Main Road, Avalahalli, Yelahanka, Bengaluru-560119, India

Email: saritha.i.g@bmsit.in

1. INTRODUCTION

Cloud computing (CC) is widely implemented in many organizations to store and process data effectively. Organizations prefer CC due to its advantages of flexibility, scalability, and reliability [1]. Cloud servers are optimal options for organizations seeking faster response times and greater flexibility [2]. Users delegate data storage and processing to the cloud because of its flexibility and scalability, while cloud service providers (CSPs) ensure security of sensitive data. Additionally, users have an option to encrypt their data before uploading it to the cloud [3], [4]. CC offers virtual computing services to small, medium, and large industries, such as platform as a service (PaaS), infrastructure as a service (IaaS), and software as a service (SaaS) [5]. Quality of service (QoS), cost-effectiveness, and stability have made CC a better choice for handling computationally intensive tasks [6], [7]. However, security remains a key issue in CC, and a revision of existing approaches is essential for advancing current techniques [8]. Cloud data security and user access controls are current security concerns [9]. A trusted third-party auditor (TPA) can verify cloud data to

minimize the user's burden, a process known as public auditing [10]. In healthcare, information is typically transferred to doctors through personal communication in recent methodologies, which enhances health outcomes [11]. E-healthcare allows for the retention of medical records, which can be accessed by doctors with the patient's permission during visits [12]. This service facilitates efficient health record management, and internet of things (IoT) systems are used to generate real-time data [13]. Electronic health records (EHR), leverage cloud servers to ensure higher-quality infrastructure [14]–[17]. This electronic storage minimizes the need for physical records and enables unified sharing with companies, third-party administrators, and medical professionals [18]. Ensuring patient privacy is crucial in addressing the security concerns of EHR systems [19]. CSPs manage and maintain cloud servers, creating a reliable environment where security is a top priority [20].

The existing research based on secure and efficient data storage in CC is analyzed with its drawbacks in this section. Rahman *et al.* [21] suggested a blockchain-based secure architecture for CC in IoT. The suggested model is named as distributed blockchain-software-defined cloud (DistB-SDCloud) which enhanced cloud security for IoT applications. It utilized distributed blockchain to provide security, privacy, and integrity when enduring scalable and flexible security. The industrial sector customers benefit from a decentralized, distributed and effective blockchain environment. However, it required huge overhead and resources due to additional computational tasks that enhance execution time. Ramachandra *et al.* [22] developed an effective secure data storage using triple data encryption standard (TDES) in CC. It provided a simple technique through enhancing key size in data encryption standard (DES) to secure from attacks and protect data privacy. The managing of symmetric key and partition assist in data structuring which enhanced network effectiveness. The three key padding and subkey helped to manage the data structure. It suffered from less scalability and adaptability because of its complex structure that affects the overall performance. Ullah *et al.* [23] presented an IoTChain model for secure storage and trusted data sharing in CC. The IoTChain model fine-grained permission mechanism was integrated with attribute-based access control (A-BAC) using ethereum blockchain. The advanced encryption standard (AES) was applied for encryption and elliptic curve diffie-hellman key exchange (ECDHKE) protocol was applied for secret key sharing among users and data owners. However, it has lacked from time complexity which leads to higher execution time.

Kashif and Kalkan [24] implemented a differential privacy preserving based framework using blockchain for IoT. Initially, the developed model classified transaction as private and public stream level through IoT. This provides two separate database in blockchain node and validation logic was modified for processing private and public transactions. The developed model utilized to identify privacy level such as low, medium and high through data owner thereby defining trade-off among privacy and utility. This provides privacy monitority thereby ensuring its level are maintained over time. However, the model was affected through encryption time and latency because of insufficient computational optimization and lack of scalability to process high volume of IoT transactions. Ahamad *et al.* [25] introduced a hybrid jaya-based shark smell optimization (J-SSO) for multi-objective privacy preservation in cloud security. The optimal key generation was accomplished through deriving multi-objective function which includes parameters like hiding ratio, information ratio preservation and degree of modification. The J-SSO involve better effectiveness in resolving real-world issues and provides an effective choice of parameters thereby generating optimal exploration capability at initial search progress. However, when dealing with high data in the cloud, the model enhanced the memory usage during the encryption and decryption process. From the overall analysis, the existing methods have drawbacks such as requiring huge overhead and resource consumption due to additional computational tasks that enhance execution time. Lack of coordination between multiple points impacts the encryption process and overhead. It lacks time complexity which leads to higher execution time. When dealing with high data in the cloud, the model enhanced the memory usage during the encryption and decryption process. To overcome these drawbacks, this research proposes an improved blowfish algorithm and modular encryption standard (IBA-MES) for encryption and decryption process to secure data storage in the cloud. The contribution of the research is as follows:

- The improved blowfish algorithm (IBA) rapidly performs encryption and decryption, thereby reducing latency. Due to its strong encryption, it offers improved protection against attacks, ensuring the safety of sensitive IoT data stored in the cloud. It is optimized for minimal resource consumption, making it ideal for IoT devices, and has a shorter execution time.
- The modular encryption standard (MES) scalability and flexibility enable the adoption of various IoT devices with differing computational capabilities in cloud systems. Its effective encryption and decryption reduce the computational overhead required to handle large volumes of data.

The research paper is arranged in the following manner. Section 2 explains the proposed methodology and section 3 provides result analysis with discussion. The conclusion of this research is given in section 4.

2. PROPOSED METHOD

Figure 1 illustrates the encryption and decryption process using the proposed IBA-MES for secure cloud storage. Initially, the input data is encrypted using IBA-MES to secure data during transmission. The encrypted data is then stored in the cloud to ensure its security. When the data is retrieved, it is decrypted using IBA-MES to recover the original information. This process ensures that data remains secure throughout its storage in the cloud. Only authorized users can access the actual data via decryption. IBA-MES provides both encryption and decryption efficiently, thereby enhancing the security of cloud storage.

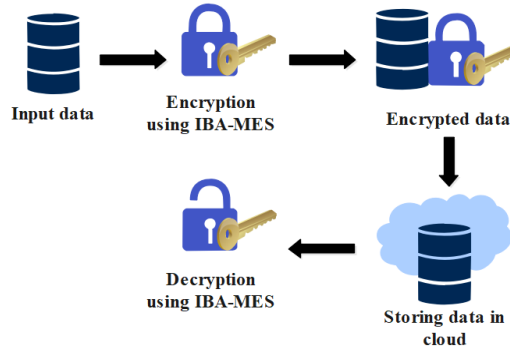


Figure 1. Block diagram for secure data storage

2.1. Input data

In this research, the healthcare dataset [22] with 3024 instances is used as input which contains 17 attributes such as patient name, month, age, gender, disease, history, symptoms, and medical measurements. The key attributes includes blood pressure, electrocardiographic result, body mass index, serum cholesterol, consultant name, maximum heart rate, body weight, and height. From the patient attribute, the essential details are selected and taken for the encryption process.

2.2. Improved blowfish algorithm for encryption and decryption

The difference from actual algorithm is a size of input block. From previous 64-bit block, the input size is enhanced to 128-bit which is separated to dual identical 64-bit blocks on the left (LE0) and right (RE0). LE0 was first XORed with P1 and P11 from a P-array, where each contains 32 bits. Then 64-bit XOR result by P1 and P11 which is given to F-function. Then, F-function result is XORed by the RE0 input block. After eighth round, LE8 and RE8 are swapped with RE8 which is XORed with P9 and P19. Subsequently, P10 is XORed with P20 which results in a final 128-bit ciphertext created by combining LE9 and RE9.

There are two modifications where F-functions now accept a 64-bit data stream as input, which is divided into eight 8-bit blocks (a, b, c, d, e, f, g , and h) with the first block containing an initial 8 bits. The b is the next 8 bits, repeatedly until h . Each 8-bit block is processed through an S-box and converted to a second 32-bit S-box value during the previous acquisition. As certain variables are inserted into the S-box, they are moved to the left or right while other variables will be moved after S-box. It is then XORed with the output of b, c and d after their values have been processed by S-box 1. This returns final 32-bit value for the S-box. Following this steps for S-box 2 with e, f, g and h . The final 64-bit output is produced which combines the results of S-box 1 and S-box 2. The construction of this modified F-function is detailed in (1) and (2).

$$IF(LE0) = ((s1(a) + S1(b) \ll 1 \bmod 2^{32}) XOR S1(c) \gg 1) + S1(d \ll 1) \bmod 2^{32} | ((S2(e) + S2(f) \ll 1 \bmod 2^{32}) XOR S2(g) \gg 1) + S2(h \ll 1) \bmod 2^{32} \quad (1)$$

$$F(LE0) = ((s1(a) + S1(b) \ll 1 \bmod 2^{32}) XOR S1(c) \ll 1) + S1(d \gg 1) \bmod 2^{32} | ((S2(e) + S2(f) \ll 1 \bmod 2^{32}) XOR S2(g) \ll 1) + S2(h \gg 1) \bmod 2^{32} \quad (2)$$

With its effective design, it accomplishes encryption and decryption rapidly thereby reducing latency. Because of its strong encryption, it provides improved protection against attacks thereby allowing the safety of sensitive IoT data stored in cloud. It is optimized with less resource consumption making it ideal for IoT devices which has less execution time.

2.2.1. Key expansion

In the IBA, the key expansion procedure converts a 128-bit key into an array of multiple subkeys. The transformation reduces total memory usage from 4168 to 2128 bytes. This P-array (P_1, P_2, \dots, P_{20}) where each element is a 32-bit subkey, also uses two S-boxes. Every entry has 256-entries ($S_1[0..255], S_2[0..255]$) of 32-bit values. In IBA, a number of iterations is required to produce each essential subkeys which is reduced from 521 to 266. It denotes fewer storage constraints for S-boxes and P-array. The estimation of subkeys is done using dual S-boxes and changes in the receiving process.

2.3. Modular encryption standard

The MES has three crucial measures Identification, classification and securing in which the identification and classification are performed at the CC user end. While securing is handled by the crypto-cloud which serves as an intermediate for cryptography measures. The MES scalability and flexibility enable to adoption of various IoT devices with differing computational capabilities in cloud systems. Its effectiveness in encryption and decryption reduces the computational overhead that is required to handle high data volumes. It reduces the latency through encryption operations which ensures quick data transactions among cloud servers and IoT devices.

2.3.1. Identification and classification

The constraint for securing cloud data is focused on the identification and classification of data based on its confidentiality level. The identification aims to differentiate the importance and sensitivity of cloud data. Generally, it has dual common classifications by following sub-classifications such as confidential (with higher security) and public (which does not necessitate security) healthcare information. In healthcare information, classification selects the degree of confidentiality according to the record nature. It is helpful to prioritize healthcare information that needs to be secured and it subsequently reduces security expenses. These dual classifications are classified into five individual sub-classifications according to a degree of confidentiality. The securing measure contains five exclusive unique key types for the five individual sub-classifications.

2.3.2. Securing and modular interaction

The securing includes remaining cryptographic steps which are performed in crypto-cloud. This step includes 9 rounds through 10 keys in which the key-0 is used for key whitening and the remaining 9 keys are used for 9 rounds. Three modules from the user end generate the construction of patients to smart devices. Then, smart device connection into the crypto cloud is performed by applying secure measures in the second layer that has 8 sub-measures. Finally, crypto-cloud into multi-cloud connection is performed in which encrypted cipher text is transformed into multi-cloud.

2.3.3. Mathematical model

The mathematical model defines the encryption and decryption process of proposed IBA-MES which ensures secure data storage in cloud-based IoT environments. It uses a block cipher approach to change plaintext to ciphertext when maintaining data integrity. The block cipher approach in an instance of x bit block and y bit key is specified in (3) and (4) which involved the encryption and decryption processes. Where, K is a key, PT and CT are plain and cipher text which shows encryption standards on PT using the private key.

$$(PT, K) \rightarrow CT \quad (3)$$

$$CT = \varepsilon (PT, K) \quad (4)$$

- i) Encryption at the patient side: the PT extension from 56 to 64 bits is defined in (5) and (6). The output data is considered as lightly encrypted T ($LFPT$). Where, Ext and Exp are extension and expansion, K_0 is a key utilized for key whitening. To accomplish key whitening, $LEPT$ is prolonged provisionally from 64 to 128 bits. The K_0 is an individual key which transforms $LEPT$ in single time.

$$LEPT = Ext(PT) \quad (5)$$

$$LEPT = Exp \oplus K_0 \quad (6)$$

Therefore, the remaining keys such as key-1 to key-9 transforms twice the $LEPT$. To remove the provisional contraction, $DExp$ is used to minimize the $LEPT$ to 64 bit that is specified in (7). The permutation of every r th round P^r is accomplished as specified in (8). After accomplishing the substitution

for r th round, the left half key is included for key addition while key subtraction is performed through the right half key as (9). Where, $DExp$ is a discard expansion. S^r is a substitution of r th round, K_L^r and K_R^r are left and right half key of r th round.

$$DExp((LEPT)Exp) \oplus K_0 \quad (7)$$

$$(LEPT \oplus K_0)P^r \quad (8)$$

$$\left(\left((LEPT \oplus K_0)P^r \right) S^r \oplus K_L^r \right) \oplus K_R^r \quad (9)$$

- ii) Decryption on physician side: decryption is performed when the CC user, doctor and patient attempt to access healthcare information from the cloud. During this side, the K_R^r is subtracted to stop the key subtraction effect that is done at encryption side as (10). Then, K_L^r is applied to perform key addition to remove key addition effect at encryption end as (11). The inverse subtraction is performed in every r th round to eliminate the substitution effect on encryption side as defined in (12). Similarly, the inverse permutation is applied in r th round to remove premature effect at encryption side as (13). The key whitening on the decryption end which remove key whitening effect on encryption end as (14). After accomplishing reduction at decryption end, LEPT is transformed into normal PT as (15).

$$\left(\left((LEPT \oplus K_0)P^r \right) S^r \oplus K_L^r \right) \oplus K_R^r \oplus K_R^r \quad (10)$$

$$\oplus K_R^r \oplus K_R^r \left((LEPT \oplus K_0)P^r \right) S^r \oplus K_L^r \oplus K_L^r \quad (11)$$

$$S_r' \left(\left((LEPT \oplus K_0)P^r \right) S^r \right) \quad (12)$$

$$P_r' \left((LEPT \oplus K_0)P^r \right) \quad (13)$$

$$LEPT K_0 \oplus K_0 \quad (14)$$

$$PT = Cn(LEPT) \quad (15)$$

Where Cn is a contraction. Therefore, the result of plaintext is attained at the decryption side. The IBA-MES ensures robust data protection by combining blowfish's strength speed and modular encryption adaptability. This dual approach improves the security and overall performance of both data encryption and decryption processes in the cloud environment.

3. RESULTS AND DISCUSSION

The IBA-MES is simulated by Python with system design of 8 GB RAM, windows 10 OS and i5 processor. The metrics such as execution time, CPU usage, encryption time, decryption time, running time, and latency are calculated for the proposed IBA-MES as discussed in this section. Additionally, the comparison of proposed IBA-MES for all performance metrics is described in this section.

In Figure 2, the execution time of the proposed IBA-MES is given for various data sizes of 100-500 with existing methods. The AES, IBA, and MES are considered existing methods to compare the efficiency of the proposed IBA-MES. The IBA-MES achieves 15, 20, 25, 35, and 45 minutes for data size of 100-500 respectively. In Figure 3, the CPU usage of the proposed IBA-MES is presented for various data sizes ranging from 100-500 with existing methods. The AES, IBA and MES are considered as existing methods to evaluate the efficiency of the proposed IBA-MES. The IBA-MES achieves 20, 20, 23, 25, and 29% for data size of 100-500 respectively.

In Figure 4, the encryption time of the proposed IBA-MES is given for various no. of blocks of 100, 300, 500, 700, and 900 with existing methods. The AES, IBA, and MES are considered as existing method to evaluate the efficiency of proposed IBA-MES. The IBA-MES achieves 185, 210, 245, 270, and 285 ms for 100, 300, 500, 700, and 900 blocks respectively. In Figure 5, the decryption time of the proposed IBA-MES is given for 100, 300, 500, 700, and 900 blocks with existing methods. The AES, IBA, and MES are considered as existing methods to evaluate the efficiency of the proposed IBA-MES. The IBA-MES achieves 395, 380, 390, 415, and 430 ms for 100, 300, 500, 700, and 900 blocks respectively.

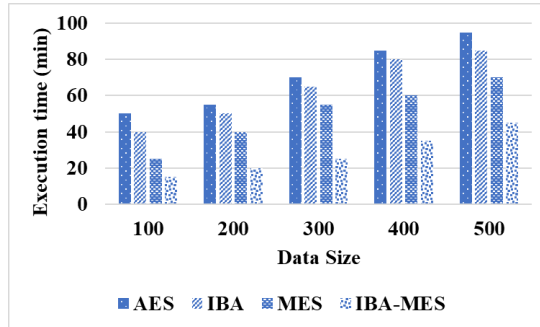


Figure 2. Execution time (min) of IBA-MES

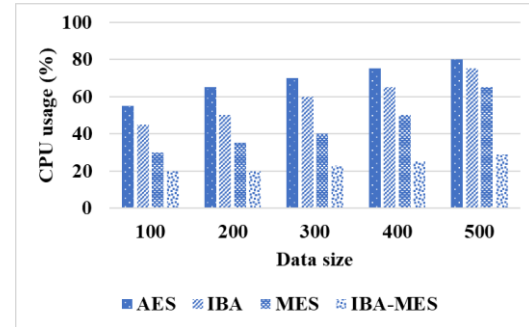


Figure 3. CPU usage (%) of IBA-MES

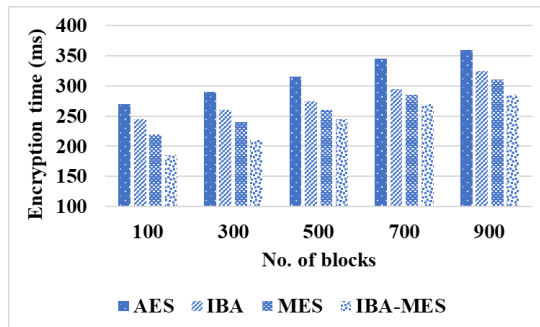


Figure 4. Encryption time (ms) of IBA-MES

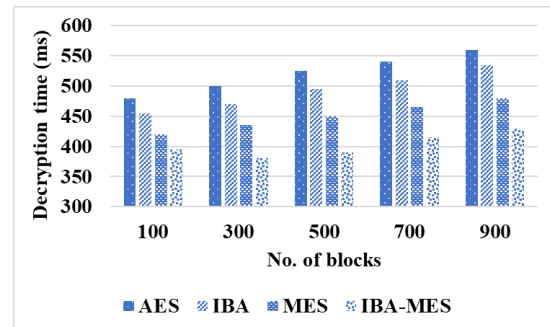


Figure 5. Decryption time (ms) of IBA-MES

In Figure 6, the running time of the proposed IBA-MES is given for no. of blocks of 100, 300, 500, 700, and 900 with existing methods. The AES, IBA, and MES are considered as existing methods to evaluate the efficiency of the proposed IBA-MES. The IBA-MES achieves 595, 570, 615, 640, and 685 ms for 100, 300, 500, 700, and 900 blocks respectively. In Figure 7, the latency of the proposed IBA-MES is given for various number of blocks of 100, 300, 500, 700, and 900 with existing methods. The AES, IBA, and MES are considered as existing methods to compare the efficiency of the proposed IBA-MES. The IBA-MES achieves 10, 35, 50, 65, and 80 ms for 100, 300, 500, 700, and 900 blocks respectively. It reduces the latency through encryption operations which ensures quick data transactions among cloud servers and IoT devices.

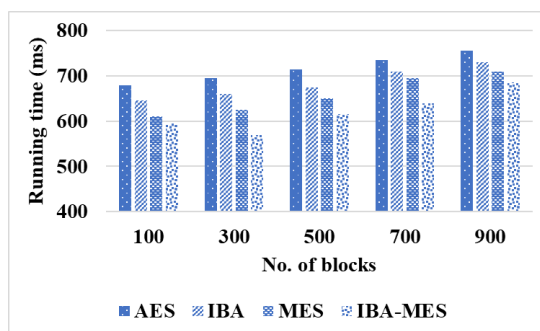


Figure 6. Running time (ms) of IBA-MES

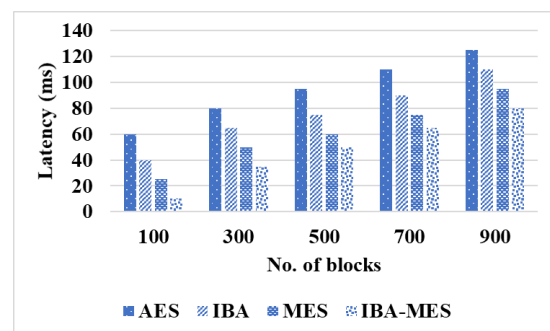


Figure 7. Latency (ms) of IBA-MES

3.1. Comparative analysis

The comparison of proposed IBA-MES for execution time and CPU usage is described in Table 1. The encryption, decryption, running, and latency are described in Table 2. The existing method such as TDES [22] is compared to show the proposed IBA-MES's efficiency. The IBA-MES achieves less execution time of 15, 20, 25, 35, and 45 minutes for data sizes of 100-500 respectively. The IBA-MES achieves less CPU usage by 20, 20, 23, 25, and 29% for data size of 100-500 respectively. The IBA-MES achieves less

encryption time of 185, 210, 245, and 270 ms for number of blocks of 100, 300, 500, and 700 respectively. The IBA-MES achieves less decryption time of 395, 380, 390, and 415 ms for 100, 300, 500, and 700 blocks respectively. The IBA-MES achieves less running time of 595, 570, 615, and 640 ms for 100, 300, 500, and 700 blocks respectively. The IBA-MES achieves a lower latency of 10, 35, 50, and 65 ms for 100, 300, 500, and 700 blocks, respectively.

Table 1. Execution time and CPU usage of proposed IBA-MES

Metrics	Methods	Data size				
		100	200	300	400	500
Execution time (min)	TDES [22]	20	25	30	40	55
	IBA-MES	15	20	25	35	45
CPU usage (%)	TDES [22]	24	24	28	30	33
	IBA-MES	20	20	23	25	29

Table 2. Encryption, decryption, running time, and latency of proposed IBA-MES

Metrics	Method	Number of blocks			
		100	300	500	700
Encryption time (ms)	TDES [22]	204	266	327	334
	IBA-MES	185	210	245	270
Decryption time (ms)	TDES [22]	428	415	425	522
	IBA-MES	395	380	390	415
Running time (ms)	TDES [22]	833	816	887	946
	IBA-MES	595	570	615	640
Latency (ms)	TDES [22]	14	56	61	72
	IBA-MES	10	35	50	65

3.2. Discussion

The results are compared with existing research and state-of-the-art methods, where the proposed IBA-MES demonstrates better performance. Metrics such as execution time, CPU usage, encryption time, decryption time, running time, and latency are calculated in the results section for the proposed IBA-MES. Existing methods, such as TDES [22], suffer from lower scalability and adaptability due to their complex structure, which negatively impacts overall performance. In contrast, the IBA's block cipher structure efficiently scales for various data sizes, ensuring secure data handling across a wide range of IoT devices. Additionally, its adaptability helps maintain data integrity, thereby improving both the security and efficiency of data storage in cloud-based IoT systems. MES reduces latency through efficient encryption operations, ensuring quick data transactions between cloud servers and IoT devices. IBA-MES ensures robust data protection by integrating the speed and strength of Blowfish with modular encryption adaptability. The major goal of this research is to develop a secure and effective data storage mechanism for cloud based IoT system. The performance metrics includes execution time, CPU usage, encryption time, decryption time, running time, and latency are validate the effectiveness of IBA-MES in securing cloud-based IoT data to ensure less processing delays. The results attained from IBA-MES denotes a better performance in data security and effectiveness in cloud-based IoT system. The encryption and decryption times (270 and 415 ms for 700 blocks) outperforms existing techniques such as TDES. Furthermore, IBA-MES optimizes resource consumption which makes it better for IoT device with less processing abilities. The finding of this research is valuable for different stakeholders and the policymakers utilize this to provide improved security for IoT based cloud storage. Additionally, the business executives in finance, healthcare and industrial sectors utilize this to ensure data security and integrity against cyber threats.

4. CONCLUSION

In this research, IBA-MES is proposed for secure and efficient data storage in cloud-based IoT environments. This dual approach enhances both the security and performance of the encryption and decryption processes in the cloud. IBA rapidly completes encryption and decryption, thus reducing latency. With its strong encryption capabilities, it offers improved protection against attacks, ensuring the safety of sensitive IoT data stored in the cloud. It is optimized for lower resource consumption, making it suitable for IoT devices that require minimal execution time. The scalability and flexibility of MES enable it to accommodate various IoT devices with limited processing power and low execution times. Its effectiveness in encryption and decryption reduces the computational overhead required to handle large data volumes. IBA-MES achieves encryption and decryption times of 270 and 415 ms, respectively, for 700 blocks, outperforming existing techniques. In the future, different data sizes will be considered to further enhance the model's performance.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Saritha Ibakkanavar	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Guddappa														
Malligehalli		✓				✓		✓	✓	✓	✓	✓		
Shivakumaraswamy														
Sowmyashree														
Ibakkanavar Guddappa	✓		✓	✓			✓			✓	✓		✓	✓
Naveen														

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data availability is not applicable to this paper as no new data were created or analyzed in this study.




REFERENCES

- [1] S. Dhar, A. Khare, and R. Singh, "Advanced security model for multimedia data sharing in internet of things," *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 11, Nov. 2023, doi: 10.1002/ett.4621.
- [2] J. Wang, J. Chen, Y. Ren, P. K. Sharma, O. Alfarrag, and A. Tolba, "Data security storage mechanism based on blockchain industrial internet of things," *Computers & Industrial Engineering*, vol. 164, Feb. 2022, doi: 10.1016/j.cie.2021.107903.
- [3] A. Ahmed, S. Abdullah, S. Iftikhar, I. Ahmad, S. Ajmal, and Q. Hussain, "A novel blockchain based secured and QoS aware IoT vehicular network in edge cloud computing," *IEEE Access*, vol. 10, pp. 77707–77722, 2022, doi: 10.1109/ACCESS.2022.3192111.
- [4] G. Ramesh, J. Logeshwaran, and V. Aravindarajan, "A secured database monitoring method to improve databackup and recovery operations in cloud computing," *BOHR International Journal of Smart Computing and Information Technology*, vol. 4, no. 1, pp. 17–23, 2023, doi: 10.54646/bijscit.2023.33.
- [5] U. Narayanan, V. Paul, and S. Joseph, "A novel system architecture for secure authentication and data sharing in cloud enabled big data environment," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 3121–3135, Jun. 2022, doi: 10.1016/j.jksuci.2020.05.005.
- [6] Y. Zhao, Q. Li, W. Yi, and H. Xiong, "Agricultural IoT data storage optimization and information security method based on blockchain," *Agriculture*, vol. 13, no. 2, Jan. 2023, doi: 10.3390/agriculture13020274.
- [7] U. Panahi and C. Bayilmiş, "Enabling secure data transmission for wireless sensor networks based IoT applications," *Ain Shams Engineering Journal*, vol. 14, no. 2, Mar. 2023, doi: 10.1016/j.asej.2022.101866.
- [8] S. Gousteris, Y. C. Stamatiou, C. Halkiopoulos, H. Antonopoulou, and N. Kostopoulos, "Secure distributed cloud storage based on the blockchain technology and smart contracts," *Emerging Science Journal*, vol. 7, no. 2, pp. 469–479, Feb. 2023, doi: 10.28991/ESJ-2023-07-02-012.
- [9] Z. A. Hussien *et al.*, "Lightweight integrity preserving scheme for secure data exchange in cloud-based IoT systems," *Applied Sciences*, vol. 13, no. 2, Jan. 2023, doi: 10.3390/app13020691.
- [10] J. S. Jayaprakash, K. Balasubramanian, R. Sulaiman, M. Kamrul Hasan, B. D. Parameshachari, and C. Iwendu, "Cloud data encryption and authentication based on enhanced merkle hash tree method," *Computers, Materials & Continua*, vol. 72, no. 1, pp. 519–534, 2022, doi: 10.32604/cmc.2022.021269.
- [11] M. Akbar, I. Ahmad, M. Mirza, M. Ali, and P. Barmavatu, "Enhanced authentication for de-duplication of big data on cloud storage system using machine learning approach," *Cluster Computing*, vol. 27, no. 3, pp. 3683–3702, Jun. 2024, doi: 10.1007/s10586-023-04171-y.
- [12] S. Mittal *et al.*, "Using identity-based cryptography as a foundation for an effective and secure cloud model for e-health," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–8, Apr. 2022, doi: 10.1155/2022/7016554.
- [13] A. Kumar, S. A. Kumar, V. Dutt, A. K. Dubey, and S. Narang, "A hybrid secure cloud platform maintenance based on improved attribute-based encryption strategies," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 2, pp. 150–157, 2023, doi: 10.9781/ijimai.2021.11.004.




- [14] K. Dubey, S. C. Sharma, and M. Kumar, "A secure IoT applications allocation framework for integrated fog-cloud environment," *Journal of Grid Computing*, vol. 20, no. 1, Mar. 2022, doi: 10.1007/s10723-021-09591-x.
- [15] A. Razaque, Y. Jararweh, B. Alotaibi, M. Alotaibi, S. Hariri, and M. Almiari, "Energy-efficient and secure mobile fog-based cloud for the Internet of Things," *Future Generation Computer Systems*, vol. 127, pp. 1–13, Feb. 2022, doi: 10.1016/j.future.2021.08.024.
- [16] R. Chaganti, V. Varadarajan, V. S. Gorantla, T. R. Gadekallu, and V. Ravi, "Blockchain-based cloud-enabled security monitoring using internet of things in smart agriculture," *Future Internet*, vol. 14, no. 9, Aug. 2022, doi: 10.3390/fi14090250.
- [17] S. Gadde, J. Amutharaj, and S. Usha, "A security model to protect the isolation of medical data in the cloud using hybrid cryptography," *Journal of Information Security and Applications*, vol. 73, 2023, doi: 10.1016/j.jisa.2022.103412.
- [18] A. Kumar *et al.*, "Development of a cloud-assisted classification technique for the preservation of secure data storage in smart cities," *Journal of Cloud Computing*, vol. 12, no. 1, Jun. 2023, doi: 10.1186/s13677-023-00469-9.
- [19] F. Sajid *et al.*, "Secure and efficient data storage operations by using intelligent classification technique and RSA algorithm in IoT-based cloud computing," *Scientific Programming*, vol. 2022, pp. 1–10, Apr. 2022, doi: 10.1155/2022/2195646.
- [20] B. Seth, S. Dalal, V. Jaglan, D. Le, S. Mohan, and G. Srivastava, "Integrating encryption techniques for secure data storage in the cloud," *Transactions on Emerging Telecommunications Technologies*, vol. 33, no. 4, Apr. 2022, doi: 10.1002/ett.4108.
- [21] A. Rahman, M. J. Islam, S. S. Band, G. Muhammad, K. Hasan, and P. Tiwari, "Towards a blockchain-SDN-based secure architecture for cloud computing in smart industrial IoT," *Digital Communications and Networks*, vol. 9, no. 2, pp. 411–421, Apr. 2023, doi: 10.1016/j.dcan.2022.11.003.
- [22] M. N. Ramachandra, M. S. Rao, W. C. Lai, B. D. Parameshachari, J. A. Babu, and K. L. Hemalatha, "An efficient and secure big data storage in cloud environment by using triple data encryption standard," *Big Data and Cognitive Computing*, vol. 6, no. 4, Sep. 2022, doi: 10.3390/bdcc6040101.
- [23] Z. Ullah, B. Raza, H. Shah, S. Khan, and A. Waheed, "Towards blockchain-based secure storage and trusted data sharing scheme for IoT environment," *IEEE Access*, vol. 10, pp. 36978–36994, 2022, doi: 10.1109/ACCESS.2022.3164081.
- [24] M. Kashif and K. Kalkan, "Differential privacy preserving based framework using blockchain for internet-of-things," *Peer-to-Peer Networking and Applications*, vol. 18, no. 1, pp. 1–23, 2025, doi: 10.1007/s12083-024-01858-w.
- [25] D. Ahamad, S. A. Hameed, and M. Akhtar, "A multi-objective privacy preservation model for cloud security using hybrid Jaya-based shark smell optimization," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2343–2358, Jun. 2022, doi: 10.1016/j.jksuci.2020.10.015.

BIOGRAPHIES OF AUTHORS






Saritha Ibakkanavar Guddappa    completed B.E. (TCE) from MVIT, Bangalore, M.Tech. (Digital Electronics and Communication) from MSRIT, Bengaluru, and Ph.D. (Cyber Security) from Visvesvaraya Technological University, Belagavi. She has 17 years of teaching and 6 years of research experience. At present, she is working as an Assistant Professor in the Department of Electronics and Communication Engineering, BMS Institute of Technology and Management, Bengaluru. She is a member of professional bodies such as ISTE and IAPURAI. She has organized and attended many workshops, FDPs, and STTPs. She has also guided several UG projects. She has published many technical papers in national and international journals and conferences. She can be contacted at email: saritha.i.g@bmsit.in.



Sowmyashree Malligehalli Shivakumaraswamy    completed B.E. (ECE) from Oxford College of Engineering, Bangalore, M.Tech. (Electronics and Communication) from SMVIT, Bengaluru, and Ph.D. (WSN) from Visvesvaraya Technological University, Belagavi. She has 16 years of teaching and 6 years of research experience. At present, she is working as an Assistant Professor in the Department of Electronics and Communication Engineering, BMS Institute of Technology and Management, Bengaluru. She is a member of professional bodies such as ISTE and IETE. She has organized and attended many workshops, FDPs, and STTPs. She has guided several UG projects and published numerous technical papers in national and international journals and conferences. She can be contacted at email: sowmyashree.m.s@bmsit.in.



Naveen Ibakkanavar Guddappa    completed B.E. (ECE) from SJGIT, Chickaballapur, M.Tech. (VLSI and Embedded Systems) from Dr. AIT, Bengaluru, and Ph.D. (Mixed Mode VLSI Design) from Visvesvaraya Technological University, Belagavi. He has 17 years of teaching, 7 years of research, and 2 years of industry experience. At present, he is working as an Associate Professor in the Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bengaluru. He is a member of professional bodies such as IEEE, ISTE, and IAENG. He has organized and attended many workshops, FDPs, and STTPs. He has guided 8 M.Tech. students and published several technical papers in national and international journals and conferences with good impact factors, indexed in Scopus and Google Scholar. He can be contacted at email: naveen.ig@nmit.ac.in.