Bulletin of Electrical Engineering and Informatics
Vol. 15, No. 6, December 2026, pp. 5123~5135
ISSN: 2302-9285, DOI: 10.11591/eei.v15i6.13215

A hybrid AES-blockchain model for decentralized image security in cloud environments

Vishnu A. Suryawanshi[1], Nakul Sharma[2], Ramesh Yashwant Mali[1], Anil Tukaram Lohar[3], Nandkishor Padmakarrao Karlekar[4], Vishal Ambaji Bogam[4]
[1]Department of Electrical and Electronic Engineering, School of Computing, MIT Art, Design and Technology University, Pune, India
[2]Department of Computer Science and Engineering (IOT CS inc. Blockchain), Vishwakarma Institute of Technology, Pune, India
[3]Department of Computer Engineering, Anantrao Pawar College of Engineering and Research, Pune, India
[4]Department of Computer Science and Engineering, School of Computing, MIT Art, Design and Technology University, Pune, India

Article Info

Article history:

Received Apr 27, 2026
Revised Aug 23, 2026
Accepted Sep 12, 2026

Keywords:

Advanced encryption standard enhancement
Blockchain
Cloud security
Decentralized authentication
Digital signature
Image encryption

ABSTRACT

The growing tendency to use cloud storage for image data is a major concern for confidentiality, integrity and authenticity. Although the advanced encryption standard (AES) or the digital signatures offer protection, they cannot be guaranteed to be secure. Single-point failures and centralized cloud architecture are not resistant to single-point failures. This paper suggests a hybrid model that incorporates an improved AES algorithm. Authentication is based on a blockchain encryption algorithm to attain decentralized and tamper-resistant image security. This is needed in a private environment setting. The suggested model encrypts and divides the pictures with the help of an optimized version of AES algorithm and stores them. It scatters encrypted fragments on cloud nodes spread across a distributed cloud and keeps hash records of authentication on a blockchain ledger. Experimental analysis shows a faster encryption speed and smaller key management. The complexity is improved as compared to the standard AES algorithm. The integration of blockchain ensures integrity and non-repudiation, addressing the limitations of cryptographic systems.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Vishnu A. Suryawanshi
Department of Electrical and Electronic Engineering, School of Computing, MIT Art Design and Technology University
Pune, India
Email: vishnusam2007@gmail.com

1. INTRODUCTION

The rapid expansion of cloud computing has revolutionized how digital images are stored, processed and transmitted across distributed infrastructures. With the exponential growth of multimedia data in healthcare, surveillance, and social media, ensuring the confidentiality and integrity of image data has become a critical research priority [1]. Cloud platforms, while offering scalability and accessibility, remain vulnerable to unauthorized access, insider threats, and data breaches due to their centralized architecture [2].

Conventional cryptographic algorithms such as the advanced encryption standard (AES) are widely used to secure cloud-stored images because of their efficiency and robustness against brute-force attacks [3]. However, the increasing sophistication of cyberattacks and the emergence of distributed cloud ecosystems demand more adaptive and resilient encryption mechanisms [4]. Although AES provides strong confidentiality, it lacks decentralized verification and traceability features, which are essential for modern cloud environments [5].

New research has been conducted on hybrid cryptographic models, that are a combination of AES with other algorithms, including Rivest–Shamir–Adleman (RSA), Blowfish, and homomorphic encryption in order to increase the protection of data [6]. Although these methods enhance security, they also tend to bring about computational space and complicated key management challenges [7]. To overcome these limitations, emerging frameworks are integrating blockchain technology with AES to achieve decentralized authentication, tamper-proof verification, and transparent auditability [8].

Figure 1 shows the traditional security of image uploading in cloud. The user uploads image that gets encrypted by AES algorithm. It is then sent to centralized storage facility for further processing. In this study, the developments are further enhanced by proposing an improved model of encryption based on AES that is combined with blockchain for decentralized image security in cloud environments. The proposed framework aims to strengthen confidentiality, integrity, and authenticity while maintaining computational efficiency and scalability.

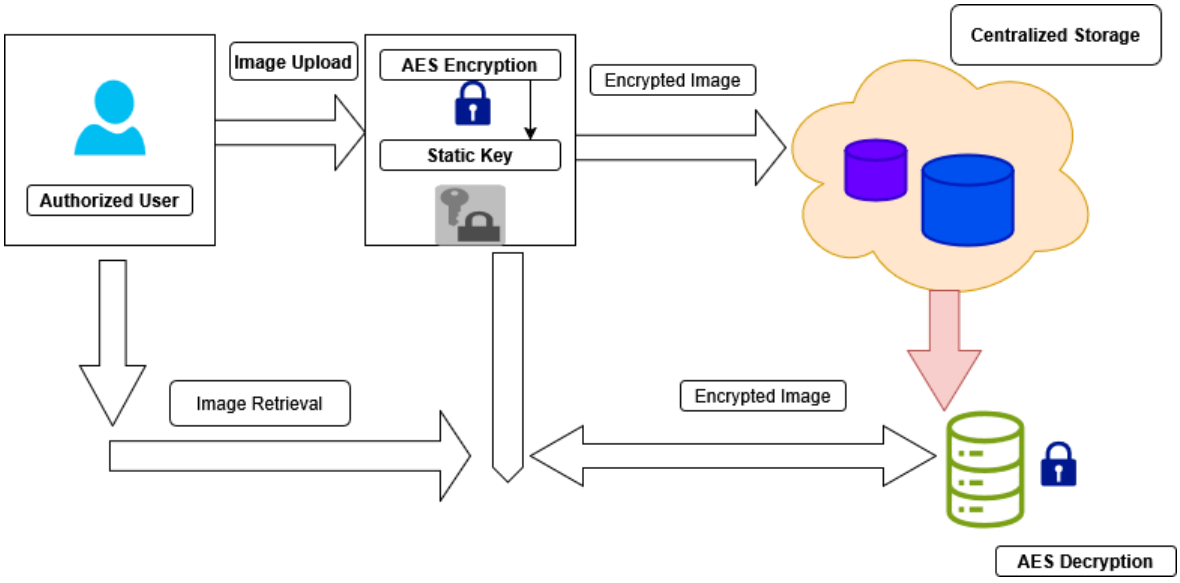


Figure 1. AES security algorithm work flow

Cloud image security has become a critical research area due to the exponential growth of multimedia data stored and transmitted through cloud infrastructures. The increasing sophistication of cyber threats has driven the need for advanced cryptographic techniques, such as the AES, digital signatures, and blockchain-based authentication systems, to ensure confidentiality, integrity, and authenticity of image data. Cloud environments present unique challenges for image security, including unauthorized access, data leakage, and integrity. Cryptographic methods have been widely adopted to mitigate these risks by encrypting image data before cloud storage or transmission. Authors proposed an AES-based image encryption framework that integrates integrity in cloud storage [8]. A novel encryption algorithm is proposed by authors which is based on logistic maps and uses the principle of chaos encryption algorithm [9].

Thabit *et al.* [1] has highlighted how lightweight cryptographic algorithms can be used to protect cloud fragmentation and digital signatures when protecting confidentiality in IoT data, with trade-offs between computational and encryption strength. Other more recent contributions have discussed hybrid encryption schemes that make use of both symmetric and asymmetric cryptography to balance between performance and security. Researchers have attempted to improve the AES work to overcome the arising security and computational issues and proposed a hybrid AES-RSA-HMAC model, which was faster and more resistant to brute-force attacks [10].

Al-Omari [11] had come up with a slim and dynamic AES version that is most suited to the next-generation internet applications, characterized by low power consumption and better throughput. Furthermore, combined AES with elliptic curve cryptography (ECC) to strengthen data confidentiality and integrity in cloud environments [5]. These enhancements demonstrate that AES can be adapted to meet the evolving demands of secure multimedia communication and storage. Digital signatures play a vital role in verifying the authenticity and integrity of digital images. In cloud-based systems, digital signatures are often

implemented using public-key cryptography, such as RSA or ECC. Makkaoui *et al.* [12] proposed a cloud-RSA scheme that enables third-party operations on encrypted data while maintaining ownership verification through digital signatures. Similarly, Seth *et al.* [6] integrated digital signatures into a hybrid Paillier-Blowfish encryption model, ensuring both confidentiality and authenticity. The use of OpenSSL-based digital signature generation and verification, as demonstrated by [8], further enhances trust in cloud image exchanges. Blockchain technology proposes a decentralized and non-tampering system in the process of authentication and verifying data integrity. Another set of authors proposed a verifiable fully homomorphic encryption (FHE) scheme integrated with blockchain to secure big data in cloud computing [13]. Although AES and the digital signatures offer solid cryptographic tools, there is a potential of incorporating blockchain to enable decentralized authentication as a future research subject.

Although previous research has explored a set of AES encryption and blockchain-based integrity verification, the proposed framework differs from current AES–blockchain, AES–ECC and blockchain-enabled cloud security models. A master key and extended round keys are derived from SHA-256 hashing of user credentials, photo metadata. This makes the encryption technique dynamic and image-specific in preference to aggregate or static AES key calculations. The framework uses phase blockchain anchoring, where hash metadata of encrypted image blocks are recorded in a blockchain ledger to provide decentralized integrity verification, traceability and tamper detection. The proposed methodology uses preprocessing and dynamic encryption of AES-256. A session-dependent AES-256 encryption mechanism using SHA-256-based seed generation from user credentials, image metadata. A modified key scheduling process that generates dynamic round keys for each encryption session. A blockchain-based fragment verification layer that stores hash records and metadata for tamper detection and traceability. An end-to-end cloud image security architecture for encryption, distributed storage, authentication, and reconstruction. Table 1 shows the gap analysis for technique used and blockchain integration with its limitations.

Table 1. Gap analysis

Ref.	Core focus	Technique used	Blockchain integration	Limitation
[1]	Cloud data protection using hybrid cryptography	AES+RSA	Not included	Centralized architecture and decentralized verification
[5]	Secure IoT data transmission and storage	EEC+AES hybrid encryption	Partial	Limited scalability, smart contracts cannot be automated
[6]	Image encryption and secure storage in cloud	AES with optimized key expansion	Not included	No decentralized integrity verification
[12]	Decentralized data integrity verification	SHA-256 and smart contracts	Fully integrated	High computational cost and limited AES optimization
[13]	Blockchain with FHE	FHE+blockchain	Fully integrated	High latency; not useful for real-time applications

Existing AES–blockchain along with AES encryption uses the blockchain framework to record hashes or access logs. The proposed framework modifies the AES layer itself through consultation-based key assignment derived from SHA-256 and applies blockchain at the block verification layer for distributed cloud storage. The proposed method yields a complete image security pipeline that includes pre-processing, more appropriate encryption, fragmentation, blockchain anchoring, verification, and reconstruction.

2. IMPLEMENTATION

The AES algorithm is also proposed to be incorporated into blockchain technology to provide decentralized and tamper-resistant image security and authentication in the cloud-based implementation. System architecture is made in such a way that it guarantees confidentiality, integrity, and authenticity of image data without compromising on computational efficiency and scalability. The process of implementation is segmented into the five main modules, namely, image preprocessing, enhanced AES encryption, image fragmentation and distribution, blockchain-based authentication, and image reconstruction and verification. Figure 2 shows AES implementation with blockchain in cloud image security.

2.1. System architecture

The architecture has three main layers: the first layer is the user layer that handles the operations of image upload, encryption as well as retrieval. Second is the cloud storage layer, which manages distributed storage of encrypted image fragments in several cloud nodes. The final is the layer of blockchain that keeps the decentralized registry of authentication hashes, transaction records, and verification metadata. All layers have secure API communication, which guarantees privacy and integrity in communication. The blockchain layer is used as a trust anchor, and there is no central authority that can verify anything.

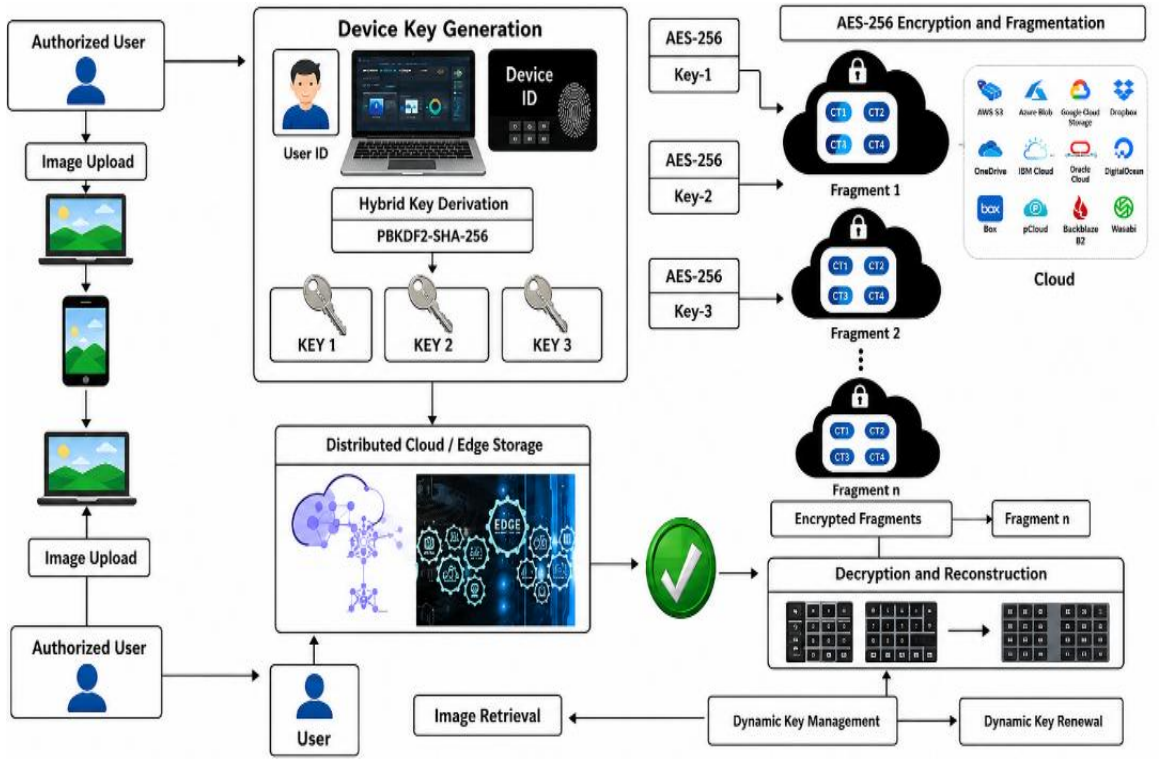


Figure 2. Proposed implementation of blockchain with AES

2.2. Enhanced advanced encryption standard encryption module

The framework proposed here is based upon AES-256 as the main encryption algorithm, but it improves it with additional algorithms such as session-dependent key derivation and implementation-level S-box optimization. The structure of AES is retained, consisting of SubBytes, ShiftRows, MixColumns, and AddRoundKey, but the master key and expanded round keys are dynamically generated for each encryption session. This method enhances the resistance against the key reuse attack and introduces user credentials and image metadata and session-specific information into the encryption process.

Assume that I is the image that is being sent, U are the user's credentials, M is the metadata of the image, and N is the session nonce. The input image is first denoised and then converted to a byte stream of binary ones. Next, a 256-bit seed is created by applying SHA-256, as (1):

$$S = SHA - 256(U||M||N) \quad (1)$$

where || denotes concatenation. The seed (S) serves as the basis for generating the dynamic encryption key for the current session. Since SHA-256 produces a 256-bit output, the seed is directly suitable for AES-256 keying. The dynamic master key is defined as (2):

$$K_0 = S \quad (2)$$

where K_0 is the 256-bit session key used as the input to the AES-256 key expansion function. Because the key is generated from user-specific and image-specific parameters, the same image encrypted in different sessions will produce different ciphertexts.

For AES-256, the key expansion procedure generates 60 32-bit words $(\omega_0, \omega_1, \ldots, \omega_{59})$, which are grouped into 15 round keys $(RK_1, RK_2, \ldots, RK_{14})$. In the proposed method, the standard AES-256 key expansion is modified by introducing a session-dependent perturbation term derived from the seed (S).

2.2.1. Dynamic key scheduling

For AES-256, the key expansion procedure generates 32-bit words $(\omega_0, \omega_1, \ldots, \omega_{59})$, which are grouped into 15 round keys $(RK_1, RK_2, \ldots, RK_{14})$. In the proposed method, the standard AES-256 key expansion is modified by introducing a session-dependent perturbation term derived from the seed (S).
The initial 8 words are extracted from (K_0):

$$\omega_0,\ \omega_1, \dots, \omega_7 \leftarrow K_0 \tag{3}$$

For $i \geq 8$, the modified key schedule is defined as (4):

$$\omega_i = \begin{cases} \omega_{i-8} \oplus SubWord(RotWord(\omega_{i-1})) \oplus Rcon_{i/8} \oplus \Delta_i, & for\ i\ mod\ 8 = 0 \\ \omega_{i-8} \oplus SubWordRotWord(\omega_{i-1}) \oplus \Delta_i, & for\ i\ mod\ 8 = 4 \\ \omega_{i-8} \oplus \omega_{i-1,} & otherwise \end{cases} \tag{4}$$

where:
- $RotWord(.)$ performs a one-byte cyclic left shift,
- $SubWord(.)$ applies byte substitution using the AES S-box,
- $(\mathrm{Rcon_j})$ is the AES round constant,
- (Δ_i) is a session-dependent perturbation term.

The perturbation term is computed as:

$$\Delta_\mathrm{i} = \mathrm{Trunc}_{32}(\text{SHA-256}(S \parallel \mathrm{i})) \tag{5}$$

where $(\mathrm{Trunc}_{32}(\cdot))$ denotes the leftmost 32 bits of the SHA-256 output. This makes the expanded round keys dependent on both the session seed and the word index, thereby increasing variability across sessions and reducing key reuse.

The round keys are then formed by grouping every four consecutive words:

$$\mathrm{RK_r} = [\mathrm{w_{4r}}, \mathrm{w_{4r+1}}, \mathrm{w_{4r+2}}, \mathrm{w_{4r+3}}], \quad \mathrm{r} = 0,1, \dots, 14 \tag{6}$$

2.2.2. S-box optimization

The proposed implementation retains the standard AES S-box mapping to preserve invertibility and compatibility with AES-256. The optimization is performed at the implementation level by precomputing the S-box lookup table once per session and reusing it for all blocks of the same image. This avoids repeated computation during byte substitution and reduces memory access overhead during encryption.

Accordingly, the substitution step is written as (7):

$$\mathrm{SubBytes(x)} = \mathrm{T[x]} \tag{7}$$

where *T* is the precomputed AES S-box table and (x) is an input byte. For decryption, the corresponding inverse table $\mathrm{T^{-1}}$ is used:

$$\mathrm{InvSubBytes(y)} = \mathrm{T^{-1}[y]} \tag{8}$$

This optimization does not alter the mathematical definition of the AES S-box, but it improves execution efficiency by avoiding redundant table generation.

2.2.3. Encryption procedure

After preprocessing, the image is divided into 128-bit blocks $\mathrm{P_1}, \mathrm{P_2}, \dots, \mathrm{P_m}$. Each block is encrypted using the dynamically generated round keys. For the (j) − th plaintext block $(\mathrm{P_j})$, the initial state is (9):

$$\mathrm{X_j^{(0)}} = \mathrm{P_j} \oplus \mathrm{RK_0} \tag{9}$$

For rounds $\mathrm{r} = 1,2, \dots, 13$, the state update is:

$$\mathrm{X_j^{(r)}} = \mathrm{AddRoundKey}\left(\mathrm{MixColumns}\left(\mathrm{ShiftRows}\left(\mathrm{SubBytes}\left(\mathrm{X_j^{(r-1)}}\right)\right)\right), \mathrm{RK_r}\right) \tag{10}$$

The final round omits the MixColumns operation:

$$\mathrm{C_j} = \mathrm{AddRoundKey}\left(\mathrm{ShiftRows}\left(\mathrm{SubBytes}\left(\mathrm{X_j^{(13)}}\right)\right), \mathrm{RK_{14}}\right) \tag{11}$$

where $\mathrm{C_j}$ is the ciphertext block. The encrypted image is obtained by concatenating all ciphertext blocks:

$$\mathrm{C} = \mathrm{C_1} \parallel \mathrm{C_2} \parallel \cdots \parallel \mathrm{C_m} \tag{12}$$

The same seed (*S*), nonce (*N*), and dynamic key derivation process are used during decryption to regenerate the identical round keys and inverse substitution table, ensuring lossless reconstruction of the original image.

2.3. Image fragmentation and distributed storage

Once the image is encrypted, it is broken into several fragments which improve security and allow distributed storage. Each fragment is allotted a special identifier and is stored in various places. To improve cloud image security by decentralization, the encrypted image is first decomposed into unequal-sized fragments. All fragments are subjected to cryptography. SHA-3 produces a unique hash. These fragments, together with their attendant metadata--including fragment identifiers and parameters of encryption are sent to different cloud nodes in order to reduce the risk of centralization. To ensure tamper-proof verification and traceability, the hash values and their respective, block IDs of storage nodes are stored in a blockchain. This is a decentralized registry that offers non-repudiable authentication. This enables image data non-repudiation and audit trails. This division provides that the cloud infrastructure is divided and compromised. Even though a single node is compromised, the entire image cannot be compromised. The image can be rebuilt with fragments and counterparts.

2.4. Blockchain-based authentication and integrity verification

There is blockchain technology, which is integrated to give decentralized authentication and tamper-proof verification. Every transaction, such as uploading of images, encryption, and retrieval, is stored as a block in the ledger. Smart contracts are prepared to streamline and control the key processes such as registration of images, validation of their hash, and access control. The hash of each fragment of the encrypted image, together with the time, is recorded as a transaction on the blockchain ledger, ensuring immutability and traceability. The system also conducts integrity checks during image retrieval by recalculating the hash of every fragment and comparing it to the blockchain record of that fragment. This mechanism will ensure that the unauthorized alteration and infiltration attempts are limited. this will strengthen the credibility and validity of image information stored on the clouds. It provides non-repudiation, traceability, and immutability of the image data.

2.5. Image reconstruction and verification

In the proposed framework, the system completes the secure image retrieval when a user requests the image retrieval, where the blockchain-based identity verification is done to authenticate the user first. After authentication, the system recovers coded fragments stored in distributed cloud nodes. The integrity of fragments is then checked by comparing the recalculated hash of fragment with stored hash on the blockchain list to ensure that the integrity has not been compromised. Once it has successfully been verified, the pieces are pieced back up into the whole encrypted picture. Lastly, the improved AES decryption algorithm is combined with the same dynamic key that was used during the encryption, thus rebuilding the initial image without breaching confidentiality, integrity, and authenticity of the retrieval process. The resulting image reconstructed is the same as the original one, indicating integrity and authenticity of data.

2.6. Eexperimental setup and evaluation

Python and Solidity were used to execute implementation of the proposed framework in order to support a smooth integration of blockchain. To increase the level of security and flexibility, AES encryption module was coded using the PyCryptodome library with some modifications on dynamic key scheduling. In blockchain simulation, Ganache and Truffle tools were used to develop and test smart contracts, which guaranteed the efficiency of smart contracts in terms of transaction processing and verification. The process of performance evaluation was performed in several dimensions: the time of the encryption process was timed with regard to the size of the image (256×256, 512×512, and 1024×1024), and the quality of the decryption process was checked with the help of the peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). The latency of the blockchain was measured in terms of the confirmation and speed of hashes, and the security analysis was conducted thoroughly to determine its withstanding against brute-force attacks, replay attacks, and tampering attacks. All of these findings prove the efficiency of the complex AES-blockchain model in providing efficient, decentralized, and secure cloud image management.

Implementation code:

```
// Step 1: User Authentication
Authenticate_User(blockchain_identity)
// Step 2: Image Encryption
Input: Original_Image
Dynamic_Key ← Generate_Key()
Encrypted_Image ← Enhanced_AES_Encrypt(Original_Image, Dynamic_Key)
```

```
// Step 3: Fragmentation
Fragments ← Split(Encrypted_Image, n)
// Step 4: Hash Generation
FOR each Fragment_i in Fragments:
    H_i ← SHA3(Fragment_i)
    Metadata ← {Fragment_ID, Node_ID, Timestamp}
    Store(Fragment_i, Cloud_Node[Node_ID], Metadata)
    Record_Transaction(Blockchain, {H_i, Node_ID, Timestamp})
// Step 5: Retrieval Process
Authenticate_User(blockchain_identity)
Retrieved_Fragments ← Fetch_Fragments(Cloud_Nodes)
FOR each Fragment_i in Retrieved_Fragments:
    Current_Hash ← SHA3(Fragment_i)
    Blockchain_Hash ← Get_Hash(Blockchain, Fragment_ID)
    IF Current_Hash ≠ Blockchain_Hash:
        Raise Integrity_Error
    ELSE
        Verified_Fragments ← Append(Fragment_i)
// Step 6: Reassembly
Encrypted_Image_Reassembled ← Merge(Verified_Fragments)
// Step 7: Decryption
       Decrypted_Image ← Enhanced_AES_Decrypt(Encrypted_Image_Reassembled, Dynamic_Key)
// Step 8: Output
       Display(Decrypted_Image)
```

Description of smart contract; i) image registration; ii) integrity validation; iii) access control; and iv) transaction login.

Blockchain integration provides great resilience to the system against tampering and replay attacks. The decentralized registry guarantees that all unauthorized alterations to image fragments can be immediately identified by checking the hash of fragments. The images used for testing were selected from publicly available benchmark datasets representing three distinct categories such as medical, satellite, and natural images each 20 to evaluate the robustness of the proposed AES–blockchain model across diverse image types.

- Medical images (JPEG): selected from the Kaggle chest x-ray dataset (https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia).
- Satellite images (TIFF): taken from the UC Merced land use dataset (https://www.kaggle.com/datasets/apollo2506/uc-merced-land-use-dataset).
- Natural images (PNG): chosen from the Berkeley segmentation dataset (BSDS500) https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html).

2.6.1. Statistical analysis report

To increase the rigor and reproducibility of the experimental evaluation, every single look was transformed into repeat multiples under the same hardware and software conditions. For each photo length and photo class, encryption time, throughput, PSNR, and SSIM were measured in repeat runs. 95% self-confidence periods were calculated to quantify the uncertainty within the suggested effects. Statistical significance testing was performed to evaluate the proposed approach against the normal AES at a degree of significance of 0.05. If the size distribution was normal, a paired t-test transformed was applied.

2.6.2. Blockchain simulation environment and limitations

The blockchain factor of the proposed framework turned out to be evaluated in the adjacent private blockchain surrounding the use of ganache and truffles. It was decided to validate smart contract execution, hash registration, access control, and integrity verification in putting controlled experimental conditions. The simulation environment enabled practical testing of the proposed blockchain layer without introducing external network variations. However, the mentioned blockchain latency values represent the closest execution latency. Additional overheads including community propagation delay, transaction queuing, node synchronization, distributed consensus, and block confirmation time in a fully allocated blockchain environment can increase the break-end latency therefore, in this pronounced blockchain timing results below implementation phase there is a view of the need to interpret as a performance measurement.

2.6.3. Scalability and storage overhead evaluation

The storage overhead is measured by comparing the size of the authentic photograph with the combined length of the encrypted fragments and associated metadata. The percentage of storage overhead was calculated as:

$$\text{textStorageoverhead}(\%) = \text{frac}S_t\text{exttotal} - S_t\text{extoriginal}S_t\text{extoriginaltime}$$

In which $S_t$extunique enter indicates the dimensions of the image and $S_t$extgeneral indicates the overall size of the encrypted photograph blocks, metadata information, and associated storage artifacts. Transaction cost is the additional payment cost needed to write updates on models and provide audit records to blockchain-based networks.
Ledger Growth is given as:

$$LG = \text{Stored_On_Blockchain}(\theta, \text{Training}, \text{ContractDetails}$$

where θ represents the time delay between time to sign a contract and its execution.

For blockchain overhead analysis, the size of each transaction file written to the ledger was measured, such as block hashes, node identities, timestamps, access metadata, and the ledger was expected to grow by the sum of the transaction lengths accumulated across all processed images. Changes in transaction costs were also recorded to determine the amount of computational garage load distributed with the help of a smart settlement layer. By increasing processed-image variety and checking multiple image sizes in benchmark datasets, scalability turned into evaluated. The evaluation recorded encryption time, fragmentation time, blockchain transaction time, laser length increase, and overall storage amplification under small, medium, and large-scale workloads The employee's proposed framework makes it viable to decide whether or not to stay green.

3. RESULTS AND DISCUSSION

The presented design of combining the improved AES encryption with authentication based on blockchain was implemented and tested on Python (encryption and fragmentation) and Solidity (blockchain smart contract). The experimental system was implemented on an Intel i7 processor, 16 GB RAM, and Ubuntu 22.04 system. Simulation of the blockchain network was done using Ganache and Truffle.

3.1. Performance metrics

The evaluation focused on key performance parameters. improved as well [14]. The blockchain timestamps have been integrated, which means that a transaction can only occur once, preventing the replay attacks and ensuring the chronological integrity [15]. Hash authentication using blockchain offers a powerful system of identifying any form of unauthorized changes or interference of encrypted fragments of the image, and data immutability is ensured among distributed nodes [16]. Moreover, dynamic key scheduling supports encryption since it keeps key parameters updated during the run time. This minimizes the susceptibility of key-dependent cryptanalytic attacks. The blockchain ledger is decentralized, which means that it does not rely on the authority, and as a result, distributed trust, transparency, and fault tolerance are ensured in the context of the cloud image security model [17]. Approximately 18–20% improvement in encryption time is achieved through optimized key scheduling, precomputed S-box substitution, and efficient block level processing. These enhancements reduce redundant computations and memory access delays, allowing the enhanced AES-256 algorithm to perform encryption faster than standard AES without compromising security. Table 2 represents encryption time, blockchain verification time and throughput.

The reported blockchain verification time of 3.2–8.9 ms was obtained in a controlled private blockchain environment using local simulation tools such as Ganache and Truffle. In real-world public or consortium blockchain networks, verification latency would be higher due to distributed consensus, block validation, and synchronization overhead. Therefore, the reported values reflect the computational efficiency of the proposed smart contract logic rather than full network latency.

Table 2. Encryption time, blockchain verification time, and throughput of the proposed framework

Image size (pixels)	Standard AES encryption time (ms)	Enhanced AES encryption time (ms)	Blockchain verification time (ms)	Throughput (MB/s)	Statistical summary
256×256	12.8	10.4	3.2	24.6	Mean±SD; 95% CI
512×512	28.5	23.1	4.8	26.3	Mean±SD; 95% CI
1024×1024	61.7	49.2	6.5	27.8	Mean±SD; 95% CI
2048×2028	132.4	108.6	8.9	29.1	Mean±SD; 95% CI

3.2. Visual quality assessment

Decrypted images were evaluated using PSNR [18] and SSIM [19] to confirm the fidelity of photo reconstruction after encryption and decryption. These metrics are here used strictly as reconstruction-satisfactory measures. A higher PSNR cost indicates that the decrypted image is numerically closer to the original image, even as a higher SSIM value confirms that the structural material content,

brightness, and contrast of the image were preserved. The results in Table 3 show always high PSNR and SSIM values in the clinical, satellite, and herbal image classes, showing that the proposed framework appropriately reconstructs the authentic image without visible distortion or loss of information. This confirms that the enhanced AES decryption method is reliable and that photograph fragmentation and blockchain-primarily based verification steps do not now degrade the overall reconstructed image [20]. However, PSNR and SSIM should not be interpreted as proof of encryption security. These measures very effectively confirm that the authentic photograph can be effectively recovered after decryption, which is anticipated for an effectively used symmetric encryption scheme so the proposed framework evaluates security residences one by one using cryptographic integrity-based metrics in the following section.

Table 3. Visual quality assessment of decrypted

Image type	PSNR (dB)	SSIM	Statistical summary
Medical image	58.7	0.998	Mean±SD; 95% CI
Satellite image	56.9	0.996	Mean±SD; 95% CI
Natural image	57.3	0.997	Mean±SD; 95% CI

High PSNR (>55 dB) and SSIM (>0.99) values confirm that the decryption process preserves image quality without distortion or data loss [21]. PSNR and SSIM were used only to confirm bit-level integrity of the decrypted images, not to evaluate compression or visual quality. Table 4 compares the proposed work with existing work.

Standard AES implementation refers to the conventional AES-256 encryption algorithm executed using the PyCryptodome library. The enhanced AES version, by contrast, integrates session-dependent dynamic key generation derived from SHA-256-based seeds [22].

Table 4. Proposed work versus existing works

Criteria	Proposed model	Ref [8]	Ref [10]	Ref [11]
Blockchain integration	Private blockchain is used for decentralized authentication	No blockchain is used; paper focuses on AES algorithm	No blockchain is used in this paper	Blockchain is included with FHE
Security enhancements	AES used dynamic key scheduling and optimized using S-box SHA 256 key along with blockchain	Improved AES algorithm using key expansion (S-box)	Security enhancements	AES used dynamic key scheduling and optimized using S-box SHA 256 key along with blockchain
Speed/ encryption time	Faster than standard AES (e.g., 10.4 ms for 256×256 images vs. 12.8 ms standard AES)	Moderate improvement over standard AES	Slower due to multi-layer encryption (AES+RSA+HMAC)	High computational cost due to FHE operations
Integrity verification	Blockchain ledger stores SHA-3 hashes of image fragments	Integrity verification limited to AES decryption validation	Integrity done using via HMAC but centralized verification	Blockchain helps in verifiable integrity through immutable transaction records
Throughput	Higher 24.6–29.1 MB/s	Significantly enhanced	Lower	Lower

The result show that proposed implementation provides a balanced trade-off between both security as well as performance. The dynamic key scheduling is useful in encryption process and uses blockchain-based integrity and verification that cannot be changed. The parameters of latency and scalability make is appropriate candidate for real time applications [23].

3.3. Security analysis

A proper security analysis is provided in this section to substantiate the security claims of the proposed framework. The assessment includes comprehensive photographic encryption metrics including number of pixels change rates (NPCR), unified average changing intensity (UACI), correlation coefficients, key sensitivity, differential cryptanalysis, linear cryptanalysis [24]. It is to evaluate the cryptographic power and the integrity protection provided through the blockchain layer. The proposed more cost-effective AES module is evaluated using entropy, NPCR, correlation coefficient analysis, major sensitivity, and avalanche impact evaluation. High NPCR and UACI values show that even a completely small change within plaintext leads to large adjustments within ciphertext, demonstrating strong resistance to differential cryptanalysis Correlation evaluation between neighboring pixels in an encrypted image. In addition to encryption-level analysis, the blockchain element provides tamper detection, traceability and non-repudiation. Each encrypted block is hashed and its hashed document is stored in the blockchain ledger. During recovery, the recalculated

hash of each fragment is compared to the stored hash to verify integrity. If a block has been modified, the mismatch is detected right now. This makes the device resilient against unauthorized modifications, replay attempts, and block substitutions.

3.3.1. Threat model and assumptions

The proposed mechanism assumes that an attacker may additionally have access to encrypted photo blocks saved in the cloud and adversary underlying can try and compromise confidentiality, integrity, or authenticity through brute force search, replaying past transactions, tampering with blocks, or statistical cryptanalysis Cryptographic primitives, i.e., SHA-256, AES-256, and the blockchain used within the non-public deployment are assumed not to break the consensus mechanism The blockchain layer is assumed to provide immutable storage of hash facts and transaction metadata, so any changes to stored blocks are detected at some stage of verification The proposed framework addresses those threats through three mechanisms. First, the AES key is dynamically generated from person credentials, photo metadata, and consultation nonce, making the ciphertext session specific. Second, blocks of encrypted photos are individually hashed and anchored to the blockchain ledger, allowing for tamper detection and traceability. Third, the use of consultation-based key assignment and substitution behavior increases resistance to statistical attacks against the encryption layer [25].

3.3.2. Entropy analysis

Entropy measures the randomness of the encrypted image. A secure encryption scheme should produce ciphertext whose pixel values are uniformly distributed, resulting in entropy close to the ideal value of 8 for 8-bit grayscale or color channels.
The entropy (H) is computed as:

$$\mathrm{H} = -\sum_{\mathrm{i}=0}^{255} \mathrm{p(i)} \log_2 \mathrm{p\ (i)}$$

where $p(i)$ is the probability of occurrence of grey level (i) in the encrypted image.

For the proposed framework, the encrypted image should ideally exhibit entropy values close to 8, indicating high unpredictability and near-uniform distribution of pixel intensities. Higher entropy implies stronger resistance to information leakage through statistical analysis.

3.3.3. Number of pixels change rates and unified average changing intensity analysis

To evaluate sensitivity to small changes in the plaintext, the NPCR and UACI are calculated. These metrics are commonly used to measure resistance to differential attacks in image encryption. If $\mathrm{C}_1$and C_2 are two encrypted images corresponding to two plaintext images that differ by only one pixel, then NPCR is defined as:

$$\mathrm{NPCR} = \frac{1}{\mathrm{W}\times\mathrm{H}} \sum_{\mathrm{i}=1}^{\mathrm{W}} \sum_{\mathrm{j}=1}^{\mathrm{H}} \mathrm{D(i,j)} \times 100$$

where:

$$\mathrm{D(i,j)} = \ [\![\{\mathrm{cases}\}\ 1, \&\ "\{\mathrm{if}\}\ \mathrm{C_1(i,j)} \neq \mathrm{C_2(i,j)}\ \ 0, \&\ "\{\mathrm{otherwise}\}\ ]\!]\{\mathrm{cases}\}$$

UACI is defined as:

$$\mathrm{UACI} = \frac{1}{\mathrm{W}\times\mathrm{H}} \sum_{\mathrm{i}=1}^{\mathrm{W}} \sum_{\mathrm{j}=1}^{\mathrm{H}} \frac{|\mathrm{C_1(i,j)} - \mathrm{C_2(i,j)}|}{255} \times 100$$

A high NPCR value and a UACI value close to the ideal benchmark indicate that a one-pixel change in the original image causes significant and widespread changes in the encrypted image. This demonstrates strong resistance to differential cryptanalysis.

4. CONCLUSION

The proposed work combines authentication using blockchain, AES-based encryption for higher performance, and secure image information. The proposed model makes use of traditional centralized models by integrating dynamic key assignment, S-box based substitution, and distributed leger verification. The experimental findings illustrate that the improved AES algorithm has a short encryption and decryption time, less complexity in key management, and increased throughput rates over the normal AES implementations.

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
Vishnu A. Suryawanshi	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Nakul Sharma		✓				✓		✓	✓	✓	✓	✓		
Ramesh Yashwant Mali	✓		✓	✓			✓			✓	✓		✓	✓
Anil Tukaram Lohar					✓		✓		✓				✓	✓
Nandkishor Padmakarrao Karlekar					✓		✓			✓		✓		✓
Vishal Ambaji Bogam		✓	✓							✓	✓	✓		✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data will be available on request by the journal.

REFERENCES

[1] F. Thabit, O. Can, A. O. Aljahdali, G. H. Al-Gaphari, and H. A. Alkhzaimi, "Cryptography algorithms for enhancing IoT Security," *Internet of Things*, vol. 22, p. 100759, Mar. 2023, doi: 10.1016/j.iot.2023.100759.

[2] Z. Yan, R. H. Deng, and V. Varadharajan, "Cryptography and Data Security in Cloud Computing," *Information Sciences*, vol. 387, pp. 53–55, May 2017, doi: 10.1016/j.ins.2016.12.034.

[3] K. Pavani, J. R. S. Sree, A. S. S. Rani, K. Rohini, T. P. Kumar, and P. Yellamma, "Data Security and Privacy Issues in Cloud Environment," in *2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2023, pp. 788-793, doi: 10.1109/ICSSIT55814.2023.10060925.

[4] E. Elgeldawi, M. Mahrous, and A. Sayed, "A Comparative Analysis of Symmetric Algorithms in Cloud Computing: A Survey," *International Journal of Computer Applications*, vol. 182, no. 48, pp. 7–16, Apr. 2019, doi: 10.5120/ijca2019918726.

[5] G. S. Mahmood, D. J. Huang, and B. A. Jaleel, "Achieving an effective, confidentiality and integrity of data in cloud computing," *International Journal of Network Security*, vol. 21, no. 2, pp. 326-332, 2019, doi: 10.6633/IJNS.201903_21(2).17.

[6] B. Seth *et al.*, "Secure Cloud Data Storage System using Hybrid Paillier-Blowfish Algorithm," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 779–798, 2021, doi: 10.32604/cmc.2021.014466.

[7] A. Kumar, P. Singh, D. P. Kamble, and I. Singh, "Hybrid cryptographic approach for strengthening IoT and 5G/B5G network security," *Scientific Reports*, vol. 15, no. 1, p. 37971, 2025, doi: 10.1038/s41598-025-21861-2

[8] Z. A. Mohammed, H. Q. Gheni, Z. J. Hussein, and A. K. M. Al-Qurabat, "Advancing Cloud Image Security via AES Algorithm Enhancement Techniques," *Engineering, Technology & Applied Science Research*, vol. 14, no. 1, pp. 12694–12701, Feb. 2024, doi: 10.48084/etasr.6601.

[9] S. Liu, J. Sun, and Z. Xu, "An Improved Image Encryption Algorithm based on Chaotic System," *Journal of Computers*, vol. 4, no. 11, pp. 1091–1100, Nov. 2009.

[10] E. S. I. Harba, "Secure Data Encryption Through a Combination of AES, RSA and HMAC," *Engineering, Technology & Applied Science Research*, vol. 7, no. 4, pp. 1781–1785, Aug. 2017, doi: 10.48084/etasr.1272.

[11] A. H. Al-Omari, "Lightweight Dynamic Crypto Algorithm for Next Internet Generation," *Engineering, Technology & Applied Science Research*, vol. 9, no. 3, pp. 4203–4208, June 2019, doi: 10.48084/etasr.2743.

[12] K. E. Makkaoui, A. Beni-Hssane, A. Ezzati, and A. El-Ansari, "Fast Cloud-RSA Scheme for Promoting Data Confidentiality in the Cloud Computing," *Procedia Computer Science*, vol. 113, pp. 33–40, 2017, doi: 10.1016/j.procs.2017.08.282.

[13] A. El-Yahyaoui and M. D. E.-C. El Kettani, "A Verifiable Fully Homomorphic Encryption Scheme for Cloud Computing Security," *Technologies*, vol. 7, no. 1, p. 21, Feb. 2019, doi: 10.3390/technologies7010021.

[14] M. S. Yousefpoor, E. Yousefpoor, H. Barati, A. Barati, A. Movaghar, and M. Hosseinzadeh, "Secure data aggregation methods and countermeasures against various attacks in wireless sensor networks: A comprehensive review," *Journal of Network and Computer Applications*, vol. 190, p. 103118, Sept. 2021, doi: 10.1016/j.jnca.2021.103118.

[15] H. Zang, H. Kim, and J. Kim, "Blockchain-Based Decentralized Storage Design for Data Confidence Over Cloud-Native Edge Infrastructure," *IEEE Access*, vol. 12, pp. 50083–50099, 2024, doi: 10.1109/ACCESS.2024.3383010.
[16] T. T. Huynh, T. D. Nguyen, and H. Tan, "A Survey on Security and Privacy Issues of Blockchain Technology," in *2019 International Conference on System Science and Engineering (ICSSE)*, July 2019, pp. 362–367, doi: 10.1109/ICSSE.2019.8823094.
[17] S. Zhang, W. Zhang, W. Liang, K. Li, and C. Dobre, "Blockchain-based secure and verifiable storage scheme for IPFS-assisted IoT with homomorphic encryption," *Computing*, vol. 107, no. 7, p. 148, July 2025, doi: 10.1007/s00607-025-01506-y.
[18] S. M. Kareem and A. M. S. Rahma, "An innovative method for enhancing advanced encryption standard algorithm based on magic square of order 6," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1684-1692, 2023, doi: 10.11591/eei.v12i3.4498
[19] Z Katılmış and Ş. Koç, "Performance Evaluation of Autoencoder and JPEGBased Image Compression Schemes Integrated with AES and RSA Encryption," *El-Cezeri*, vol. 13, no. 3, pp. 477-496, doi: 10.31202/ecjse.1796961.
[20] B. A. Mohammed *et al.*, "Security challenges and solutions in Internet of Medical Things (IoMT) communication: A review," *Journal of King Saud University Computer and Information Sciences*, vol. 38, p. 254, 2026, doi: 10.1007/s44443-026-00658-x.
[21] M. Martini, "A Simple Relationship Between SSIM and PSNR for DCT-Based Compressed Images and Video: SSIM as Content-Aware PSNR," in *2023 IEEE 25th International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–5, Sept. 2023, doi: 10.1109/mmsp59012.2023.10337706.
[22] E. N. Kadhim and K. A. Hussein, "A New Hybrid Lightweight LEA-PRESENT (HLP) Algorithm to Encrypt Color Images," *International Journal of Safety & Security Engineering*, vol. 16, no. 5, pp. 1009–1017, 2026, doi: 10.18280/ijsse.160507.
[23] K. Quist-Aphetsi and I. B. Senkyire, "Validating of Digital Forensic Images Using SHA-256," in *2019 International Conference on Cyber Security and Internet of Things (ICSIoT)*, Accra, Ghana, 2019, pp. 118-121, doi: 10.1109/ICSIoT47925.2019.00028.
[24] H. Liu and X. Wang, "Color image encryption based on one-time keys and robust chaotic maps," *Computers & Mathematics with Applications*, vol. 59, no. 10, pp. 3320–3327, 2010, doi: 10.1016/j.camwa.2010.03.017.
[25] I. Arioua, S. Hraoui, and H. El Fadili, "Semantically conditioned hyperchaotic encryption for medical images: U-Net and SHA-256 key binding against chosen plaintext attacks," *Cybersecurity*, vol. 9, no. 1, p. 203, 2026, doi: 10.1186/s42400-026-00638-8.

BIOGRAPHIES OF AUTHORS



Dr. Vishnu A. Suryawanshi working as an assistant professor at the MIT Art Design and Technology University Pune. He received his M.Tech. degree in VLSI and System Design in from J.N.T. University Kakinada with distinction and Ph.D. from Rashtra Sant Tukdoji Maharaj University Nagpur, Maharashtra in 2009 and 2020 respectively. He has more than 23 years teaching experience in electronics and telecommunication/computer engineering. His current research interests VLSI, cryptography and network security, artificial intelligence and machine learning, and systems security. He has published more than 12 papers in national and international journals and conferences, out of 2 are in Scopus. He published two book chapters which are in Scopus indexed. He has published 2 patents at academic level. He can be contacted at email: vishnusam2007@gmail.com.



Dr. Nakul Sharma is an assistant professor at the Department of Computer Science and Engineering (IoT CS inc. Blockchain Technology) at Vishwakarma Institute of Technology. He has been assigned various administrative duties and has vast experience in both teaching as well as research. He completed his B.Tech. from Bharati Vidyapeeth College of Engineering and Master of Engineering from Thapar University. He has completed his Ph.D. from Koneru Lakshmiah Education Foundation. He has to his credit several national and international research papers and patents. His area of specialization includes source code mining, IoT, machine learning, and human health. He can be contacted at email: nakul777@gmail.com and nakul.sharma@vit.edu.



Dr. Ramesh Yashwant Mali is a professor (asso. dean) at the School of Computing, MIT-ADT University Pune with over 26 years of teaching experience. He specializes in areas such as wireless technology, image processing, signal processing and computer science theory, and methods. He can be contacted at email: Ramesh.mali@mituniversity.edu.in.



Prof. Anil Tukaram Lohar is working as an assistant professor in the Department of Computer Engineering at Anantrao Pawar College of Engineering and Research, Parvati, Pune. He has more than 15 years of academic experience in teaching, research, and academic administration. His areas of research interest include computer vision, image processing, and smart computing. He has contributed to various reputed international journals and conference publications, reflecting his active involvement in research and innovation in emerging technologies with more than 40 international publications on his credit. He can be contacted at email: loharanil04@gmail.com.



Dr. Nandkishor Padmakarrao Karlekar is an associate professor at the School of Computing, MIT-ADT University. Pune with over 23 years of teaching experience, he specializes in areas such as biometric security, cloud computing, artificial intelligence, machine learning, and big data analytics. He holds a Ph.D. in computer engineering and has authored numerous research papers and patents in emerging technologies. He is also recognized for his contribution to academic development and innovation initiatives. He can be contacted at email: nandkisor.karlekar@mituniversity.edu.in.



Vishal Ambaji Bogam is assistant professor in Department of Computer Science and Engineering, School of Computing, MIT Art Design and Technology University, Pune, India. Has experience: teaching about 14.5 years and in industry about 3.5 years. His research interests are IoT, AI, and networking. He can be contacted at email: vishal.bogam@gmail.com.