IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 November 2023, accepted 30 December 2023, date of publication 8 January 2024, date of current version 23 February 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3351119

RESEARCH ARTICLE

Dynamic AES Encryption and Blockchain Key Management: A Novel Solution for Cloud Data Security

MOHAMMED Y. SHAKOR[1,2], **MUSTAFA IBRAHIM KHALEEL**[2], **MEJDL SAFRAN**[3], **SULTAN ALFARHOOD**[3], **AND MICHELLE ZHU**[4], **(Member, IEEE)**

[1]Department of English, College of Education, University of Garmian, Kalar 46021, Iraq
[2]Department of Computer, College of Science, University of Sulaimani, Sulaymaniyah 46001, Iraq
[3]Department of Computer Science, College of Computer and Information Sciences, King Saud University, Riyadh 11543, Saudi Arabia
[4]Department of Computer Science, College of Science and Mathematics, Montclair State University, Montclair, NJ 07043, USA

Corresponding author: Mustafa Ibrahim Khaleel (mustafa.khaleel@univsul.edu.iq)

This research is funded by the Researchers Supporting Project Number (RSPD2024R1027), King Saud University, Riyadh, Saudi Arabia.

ABSTRACT In the rapidly evolving realm of cloud computing security, this paper introduces an innovative solution to address persistent challenges. The proliferation of cloud technology has brought forth heightened concerns regarding data security, necessitating novel approaches to safeguarding sensitive information. The issue centers on the vulnerability of cloud-stored data, usually necessitating enhanced encryption and key management strategies. Traditional methods usually fall short in mitigating risks associated with compromised encryption keys and centralized key storage. To combat these challenges, our proposed solution encompasses a two-phase approach. In the first phase, dynamic Advanced Encryption Standard (AES) keys are generated, ensuring each file's encryption with a unique and ever-changing key. This approach significantly enhances file-level security, curtailing an attacker's ability to decrypt multiple files even if a key is compromised. The second phase introduces blockchain technology, where keys are securely stored with accompanying metadata, bolstering security and data integrity. Elliptic Curve Cryptography (ECC) public key encryption enhances security during transmission and storage, while also facilitating secure file sharing. In conclusion, this comprehensive approach enhances cloud security, providing robust encryption, decentralized key management, and protection against unauthorized access. Its scalability and adaptability make it a valuable asset in contemporary cloud security paradigms, assuring users of data security in the cloud.

INDEX TERMS AES, blockchain, cloud computing, cloud storage, dynamic encryption, ECC.

I. INTRODUCTION

Given that cloud computing stands as one of the pervasive technologies within the Information Technology (IT) sector, it presents a set of advantages which are: encompassing virtualization, extensive scalability, cost-efficiency, remote data processing, and the provision of on-demand clients-centric sharing services [1]. It is useful for different areas in IT, including business applications, educational platforms, and especially, data storage and sharing enabled by cloud services [2].

Cloud storage exhibits several salient characteristics, including immediate availability, affordability, accessibility, ease, reliability, flexibility, and a wide range of leasing choices [3]. Furthermore, cloud computing is underpinned by critical attributes such as security, scalability, economic efficiency, accessibility, data recovery capabilities, and optimized resource utilization. Trust emerges as a pivotal concern when contemplating the transfer of user data to cloud environments, representing a significant challenge in the relationship between users and cloud service providers [4]. Users of cloud storage services necessitate clear visibility and assurance regarding the security and integrity of their data stored in the cloud, given the limited means for

The associate editor coordinating the review of this manuscript and approving it for publication was Mueen Uddin.

monitoring stored data. To address this imperative and foster broad user acceptance, a multitude of data and resource protection strategies have been introduced and integrated within the domain of cloud security, leveraging contemporary cryptographic algorithms.

Data protection through encryption in the cloud entails the implementation of robust security measures to safeguard customer data within server centers against external and internal threats, facilitated by encryption algorithms [5]. Two primary categories of encryption methods, supported by cryptographic keys, are symmetric and asymmetric cryptography [6]. The selection between these methods hinges on the number of keys employed: one key for symmetric cryptography and a pair of keys for asymmetric encryption/decryption. The use of larger and more intricate keys enhances the security of encryption algorithms and renders attacks more formidable.

Conversely, cloud users have the opportunity to bolster trust and enhance data protection when engaging in outsourcing and cloud services by harnessing the innovative and emerging technology of Blockchain [7]. Blockchain security offers a more complex and reliable paradigm than centralized database security. Blockchain works by keeping track of documents in a ledger that are safely connected to earlier blocks using cryptographic hash algorithms. A blockchain is a type of distributed ledger that is used to record transactions and prevent tampering. Usually run via a peer-to-peer network, the Blockchain is designed specifically to prevent unwanted manipulation. As a result, Blockchain can furnish security measures on par with those found in central database storage, effectively averting potential attacks and data breaches from a managerial perspective.

Furthermore, in scenarios where data transparency is imperative, Blockchain's inherent attribute of openness can facilitate the necessary level of data transparency [8]. Because of these unique benefits, Blockchain is used in a variety of industries, such as finance and the Internet of Things (IoT) ecosystem, and its use is expected to grow dramatically. In light of its effectiveness and accessibility, numerous IT environments have embraced cloud computing. Consequently, there has been a heightened focus on exploring critical security facets concerning cloud security and privacy issues.

This paper introduces a novel approach aimed at enhancing file storage security within the cloud infrastructure. This approach leverages a hybrid dynamic encryption technique, incorporating elements of Elliptic Curve Cryptography, Advanced Encryption Standard, and Blockchain technology. The primary objective is to establish a highly secure environment conducive to elevating the overall security of cloud-based storage solutions.

The article's primary contributions are encapsulated within the following key points:

- Dynamic AES File Encryption: The article introduces an innovative approach to file encryption utilizing the Advanced Encryption Standard (AES). This method is characterized by its dynamic and efficient key generation mechanism, which bolsters the security of file storage in the cloud.
- Blockchain-Powered Key Security: A notable contribution lies in the integration of Blockchain technology to secure cryptographic keys within the cloud environment. This ensures the robust protection of encryption keys and safeguards against potential security breaches.
- User-Friendly Key Management: The article streamlines the process of key management for end-users. This simplification empowers users to efficiently manage the substantial volume of dynamic keys required for encryption tasks, thereby enhancing usability and security in cloud-based storage systems.

II. BACKGROUND AND METHODOLOGY

A. DYNAMIC ENCRYPTION

In contrast to the conventional practices of encrypting data either at rest (i.e., during storage) or during transmission (i.e., while traversing a network), dynamic encryption, also referred to as "runtime encryption" or "real-time encryption," encompasses the process of encrypting data as it is generated or accessed. Dynamic encryption ensures the protection of data from the moment of its creation or access until it is no longer required.

Key characteristics and principles associated with dynamic encryption include the following:

- Encryption in Real-Time [9]: Dynamic encryption secures data while it is in use, typically employing encryption keys generated or derived in the active process. This ensures data security during processing, transfer, or utilization.
- Data-in-Use Protection [10]: Dynamic encryption safeguards data during its active utilization, ensuring encryption even when authorized users access it or applications process it. That sets it apart from data-in-transit encryption (like file encryption on storage devices) and data-at-rest encryption (like network transmission encryption).
- Granular Access Control [11]: is frequently used in tandem with dynamic encryption, gives businesses the ability to specify who can access data and under what conditions. Permissions granted to the user, the time, the place, and other pertinent variables can all be used to restrict access.
- Adaptive Security [12]: Dynamic encryption demonstrates flexibility in reaction to changing security scenarios. For example, according on the perceived danger level or the sensitivity of the material, the encryption strength and key management may be changed.
- Robust Authentication [13]: Strict authentication procedures are often included with dynamic encryption to guarantee that only authorized entities-individuals or systems-are able to access encrypted data. Techniques

such as digital certificates and multi-factor authentication may be deployed for this purpose.
- Key Management [14]: Managing keys well is essential when it comes to dynamic encryption. To maintain the security of encrypted data, encryption keys must be generated, stored, cycled, and destroyed on time and securely.

In order to protect sensitive data, dynamic encryption adds an extra layer of security, reducing the danger of data breaches and illegal access. It is frequently used in situations like secure communications, financial transactions, healthcare, and cloud computing, where data security and privacy are crucial.

B. AES

In 2000, the NIST intentionally selected Rijndael as the advanced encryption standard due to its outstanding qualities in terms of security, performance, and elegance. As per NIST guidelines, the symmetric encryption method AES has a block size of 128 bits. A key feature is that AES can vary the number of encryption rounds according to the size of the encryption key. More specifically, For a 128-bit key, the AES uses 10 rounds of encryption; for 192-bit and 256-bit keys, it uses 12 rounds and 14 rounds, respectively [15].

The fundamental building blocks of each encryption round in AES encompass SubBytes, ShiftRows, MixColumns, and AddRoundKey operations. Among these, the AddRoundKey operation assumes paramount importance as it executes an exclusive OR (XOR) operation between the input state matrix and the cryptographic key. It is noteworthy that in the traditional AES framework, each round key is generated by means of a predetermined key expansion process.

The selection of Rijndael as the advanced encryption standard, its block length, the variable number of encryption rounds, and the integral components of AES rounds, including the critical AddRoundKey operation, collectively contribute to the robustness and effectiveness of this widely adopted encryption algorithm [16].

C. ELIPTIC CURVE CRYPTOGRAPHY

Elliptic Curve Cryptography, which also know as ECC, is a method for encrypting and decrypting data that creates a pair of keys by mathematically connecting each point on an elliptic curve to a specific set of public and private keys [17]. But the public key is distributed, the private key remains private. To guarantee the security of data being transmitted through ECC the sender has to get the recipient's public key first. The data is then encrypted using the public key and can be unencrypted only with the recipient's private key. The data can be viewed only by the intended receiver when the encryption method is applied.

Many applications from Virtual Private Networks (VPNs) to file transfers and secure email protocols employ the currently popular methods, the so-called ECC. It is also utilized in the design of cryptographic protocols such as the TLS protocol, that establish a secure internet connection. The prime modulus p, the generator point G, the elliptic curve's coefficients a and b, and the order of the generator point n are input parameters for the ECC method. The public key Q is determined as dG, while the private key d is produced as a random integer between 1 and $n - 1$.

Considering the aforementioned points, ECC is a secure and efficient encryption algorithm which can be used for a wide range of applications consisting of those that require the use of mobile gadgets.

An equation: $y^2 = x^3 + ax + b$ is an equation of an elliptic curve. In this equation, the constants a and b represent the shape of the curve which looks like an elongated circle or oval. The curve contains point at infinity which is involved in the point addition operation and it also has locations where $y^2 = x^3 + ax + b$.

The algorithm starts with a point P and perform a point doubling or point addition operation to create points on the curve. A point P on the curve is used as input for the point doubling operation, which outputs a new point $2P$. When two points P and Q are added together, a third point R, which is also on the curve, is produced.

D. BLOCKCHAIN TECHNOLOGY

Blockchain technology has recently garnered potential to revolutionize several industries, including cloud computing recently [18]. The urgency with which this problem must be solved in order to improve cloud data storage security is highlighted. Because blockchain technology is known for its immutable, transparent, and secure record-keeping, it appears to be a viable solution. Blockchain integration with cloud computing systems seems to be a good fit because of its decentralized architecture, which protects against fraud and manipulation.

The application of blockchain technology holds promise in addressing several critical issues within the realm of cloud security research. Blockchain technology may provide a strong answer to this issue by utilizing smart contracts that are able to confirm device identities and authorize network access in accordance with predetermined standards [19].

Numerous research endeavors have explored the utilization of blockchain technology to enhance cloud security. These studies include the use of blockchain-based solutions to protect the privacy and integrity of data, enable private communication in cloud services, and provide secure device identification. Still, more study is required to fully understand how blockchain technology might improve cloud security and to determine the best ways to put it into practice.

Public and private blockchains exhibit marked disparities in terms of their decentralization paradigms. While private blockchains act as closed, limited networks, public blockchains are open, decentralized, and welcome participation from everybody interested [20]. Consequently, private blockchains, in theory, offer superior efficiency and security attributes when compared to their public counterparts.

However, more centralization and decreased transparency are the cost paid for this improved performance.

In this paper, a private blockchain has been employed as the chosen framework for implementing blockchain technology to enhance the security of data within cloud storage systems.

III. RELATED WORKS

The growing adoption of cloud storage can be attributed to its convenient accessibility, resource efficiency, and cost-effectiveness. However, ensuring user privacy during data transfers to the cloud requires implementing technologies that guarantee data privacy and integrity. This aspect holds particular significance within the related work context, where investigating diverse techniques for enhancing security in data migration to the cloud remains a central focus.

In 2021, [21] introduced a new Lightweight Cryptographic Algorithm named (NLCA), which operates as a 16-byte block cipher and utilizes a 16-byte key for encryption within cloud environments. The objective of this proposal is to enhance data security. Notably, the algorithm exhibits a flexible nature while concurrently achieving optimal encryption speed and an elevated level of security which is accomplished by incorporating supplementary logical operations, distinguishing NLCA from other encryption algorithms.

In 2021, Hybrid algorithms have demonstrated their effectiveness in enhancing data protection within the cloud environment, corroborated by researchers in [22]. This research proffered a hybrid algorithm that capitalizes on the synergistic attributes of Elliptic Curve Cryptography (ECC) and AES algorithms. An ECC algorithm was enlisted for AES key generation to harmonize the imperatives of data security, computational efficiency, and implementation expediency. The algorithm's key size, notable for its compact dimensions, is an additional strength of the proposed system. A comprehensive comparative analysis involving diverse encryption algorithms and alternative proposed systems was conducted. The outcomes firmly establish that the AES-ECC hybrid algorithm attains superior levels of security and exhibits reduced energy consumption in contrast to its counterparts, rendering it a quintessential choice for data-safeguarding endeavors in the cloud.

In 2022, Blockchain technology has been employed to tackle the shortcomings and obstacles inherent in conventional medical cloud storage systems and establish trust, audibility, and data-sharing interoperability as employed in [23]. The proposed solution incorporates a consensus algorithm for validating new blocks, authenticating healthcare providers, and enhancing data management in the cloud.

In 2022, the Fine-Grained Access Control (FGAC) system has been proposed to enhance the trustworthiness and confidentiality of users and service providers by leveraging a fuzzy logic framework [24]. The system creates three groups of keys which are the public, private, and session keys. The proposed solution utilizes such an elaborate management scheme to deliver an array of security functions. Thus, it encompasses various aspects of possible threats emerging from different forms of cyberattacks. These featured processes are enabled by the system's remarkable capability to use biometric authentication correctly that was achieved by developing a strong approval procedure which follows the rule of permissions and requirements to the latter.

In 2023, An novel Non-Deterministic Cryptographic Scheme (NCS) solution has been proposed to ensure data confidentiality and privacy in cloud environments, incorporating Sliding Window Algorithm (SWA), Linear Congruential Generator (LGC), and XOR implementation. The suggested method's strength was compared with the encryption algorithms of AES, RSA, and DES to show its superiority in terms of execution time. The resolution presented in [25] also emphasized striking a balance between the encryption algorithm's strength and efficiency in relation to the volume of data.

In 2023, the solution in [26] proposed utilizing AES, RSA, DES, and Blowfish encryption algorithms to elevate data security within a cloud environment. The solution is including computing time, strength of encryption, and resource use is carried out to undertake a thorough performance evaluation of these methods. The comparison analysis's findings demonstrate the AES algorithm's superiority in terms of cryptographic resilience and encryption speed.

IV. PROPOSED DYNAMIC ENCRYPTION SOLUTION

The proposed solution relies primarily on three fundamental elements, as shown in Figure 1, to secure data at three levels: when it is transferred from the client to the server, when it is stored and managed on the server, and when it is shared among clients. The utilized components are the AES encryption algorithm, the ECC public key encryption algorithm, and blockchain technology.

Therefore, we will elucidate the processes of securing data through data encryption and decryption, as well as how to manage file sharing operations by creating branches in the blockchain, thereby enhancing blockchain management.

A. KEY GENERATION AND FILE ENCRYPTION

In the initial stage of the proposed solution, the client initiates the blockchain if they do not already possess any previous blocks. The first block is initialized with random data, and the block number and creation date are added to it, as depicted in Figure 2. However, if the client already has a pre-existing blockchain, they have two options. They can either fetch the latest block from the server, should it not be available locally, or retrieve it from their device, if it is locally available.

This flexibility allows the user to access the necessary data even if it is not present locally or to leverage the data available on their device to expedite operations. Following this, the client inputs the file they wish to upload to the cloud storage service into the SHA-256 algorithm to obtain the file's hash code. Subsequently, the client inputs the hash code of the last block in the blockchain into the same algorithm to acquire a hash code. These two codes are then combined using XOR to yield a final code, which serves as the encryption key for the

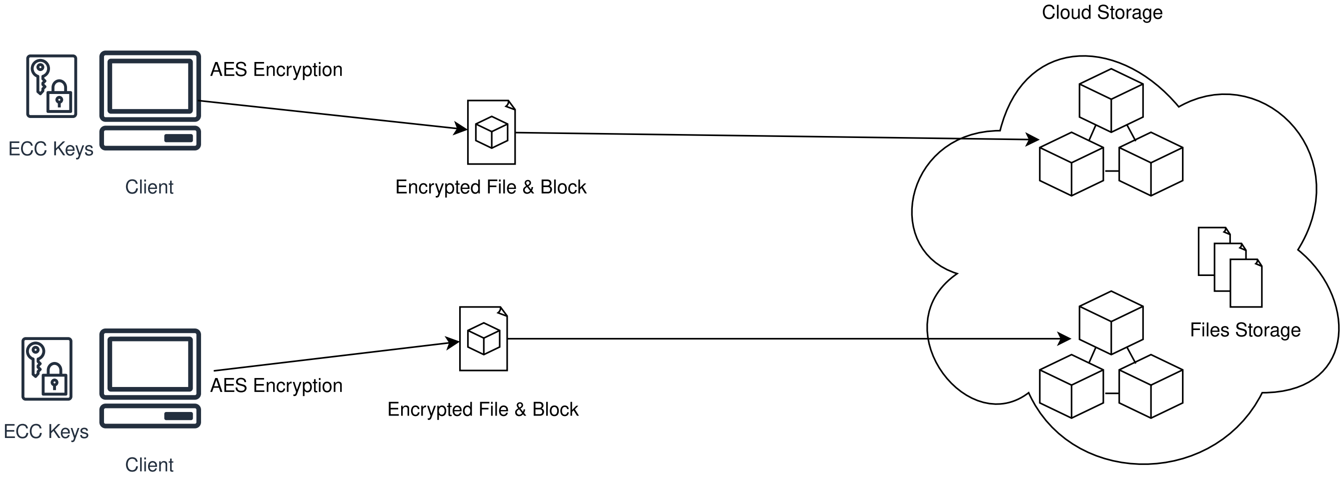

FIGURE 1. **Proposed solution architecture.**

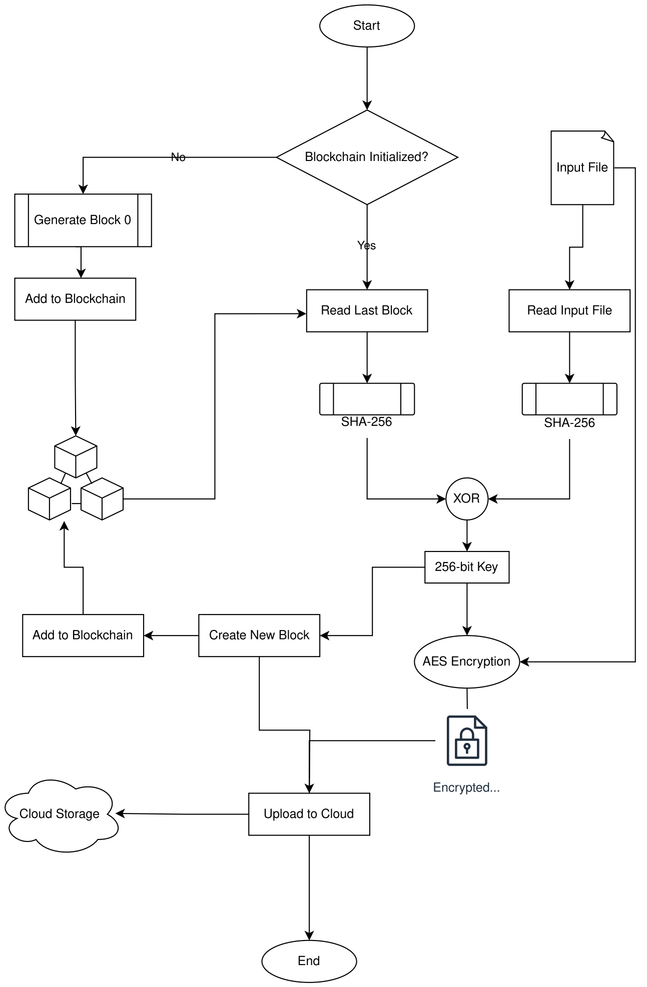

FIGURE 2. **Dynamic encryption flowchart.**

file. This process helps fortify the security of the stored file within the Blockchain.

Algorithm 1 outlines the primary steps involved in the key generation and file encryption process.

Algorithm 1 Key Generation and File Encryption

```
Input: Plain File
Output: Encrypted File and Encrypted Block
if Blockchain is Empty then
    Block ← New Block
    Block Data ← Random Data
    Block ID = 1
else
    Block Data ← Random Data
    Block ID = Block ID + 1
    Block DateTime = DateTimeNow
end if
if Local(Blockchain) is Null then
    Request Block[Last] ← Server
else
    Hash Code = Hash(Block[Last])
end if
File Hash Code = Hash(Plain File)
key = Hash Code ⊕ File Hash Code
encrypted file = Encrypt(Plain File, key)
```

The process of generating blocks for the purpose of adding them to the blockchain comprises the following stages, as illustrated in Figure 3. It commences with the retrieval of the dynamic encryption key used to encrypt the file, which is generated as part of the key generation process. Simultaneously, the contents of the latest block in the blockchain are read. Subsequently, the block counter is incremented by one, and the current time and date on the client's device are recorded.

The next step involves constructing the block, which will contain the encryption key as data and the hash of the previous block. Following this, the user encrypts the entire block using their private encryption key, and the new block, along with the encrypted file, is transmitted to the server. Additionally,

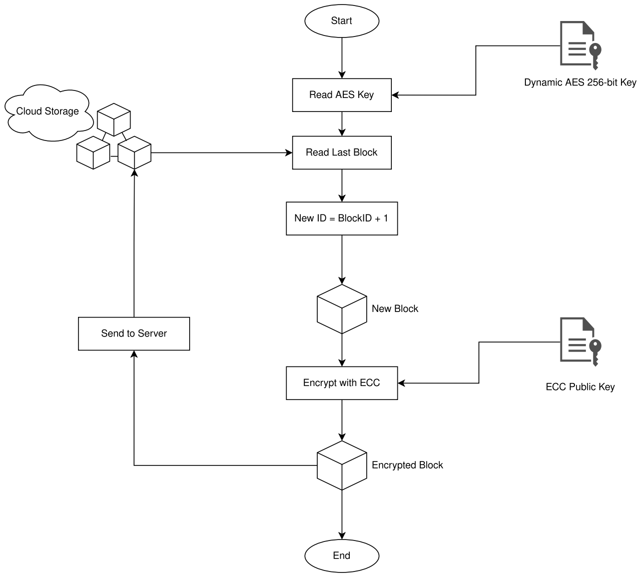

FIGURE 3. Blockchain generating flowchart.

a local copy of the block may be retained for the purpose of expedited retrieval when necessary.

Every client is required to possess a pair of ECC keys, comprising a both public and private keys. The public key serves the dual purpose of being disseminated to the general public and stored alongside the client's unique identifier on the server when sharing with other clients is necessitated. Conversely, the private key is retained by the client to facilitate the decryption of blocks and to obtain the access key for each individual file exclusively. The client's public key plays a pivotal role in encrypting block data on the client side prior to transmission to the server, thereby rendering the block data impervious to inspection by the server's administrators.

Algorithm 2 outlines the primary steps involved in the key securing using blockchain and ECC algorithm.

Algorithm 2 Block Generating and Securing Keys

```
Input: AES Key 256-bit, ECC Public Key
Output: Encrypted Block
  New Block Data = AES Key
  New Block ID = Block ID + 1
  New Block DateTime = DateTimeNow
  Encrypted Block = Encrypt(New Block, ECC Public Key)
  Send Encrypted Block $\Longrightarrow$ Server
  Send Encrypted File $\Longrightarrow$ Server
  Server Save New Block and Encrypted File
```

The novel approach outlined herein offers several distinct advantages, which can be delineated as follows:

- **Enhanced File-Level Security:** By employing key generation based on individual files, this approach facilitates the encryption of each file with a unique and dynamically changing key. Consequently, this dynamic key generation mechanism substantially elevates the security of files. Even in the event of key compromise, an attacker's ability to decrypt multiple files is restricted, as each file is encrypted with a distinct key.
- **Dynamic Key Generation:** The utilization of two distinct hashes, namely the file hash and the block hash, to derive encryption keys ensures the generation of different keys for each encryption instance, even when the content remains the same. This feature not only bolsters security but also thwarts any attempts by service providers to acquire or deduce the key based solely on block content, as it necessitates both hash codes for key derivation.
- **File Sharing with Asymmetric Key Encryption:** Encrypting blocks with an asymmetric key, such as ECC (Elliptic Curve Cryptography) public key, affords two significant advantages. Firstly, it shields the blockchain from unauthorized access by service providers or potential attackers during transmission to or storage within the cloud. Secondly, it enables efficient file sharing with clients. Clients can request block permissions from recipients and subsequently employ the blocks to generate encryption keys, facilitating the secure addition of files to the recipient's cloud-based file chain. This dual-pronged benefit enhances both security and user functionality.

B. DECRYPTION AND FILE DOWNLOADING

In the process of decryption and file retrieval, the client initiates a request for the file they wish to decrypt as depicts in Figure 4. Consequently, the server transmits the file along with its associated block. Using the private key of the ECC algorithm, the client decrypts the block to access its specific data, which includes the encryption key specific to the file. The absence of the key in an explicit form on the server enables the service provider to create multiple copies of the user's blockchain and provide them in a decentralized manner. Only authorized individuals possessing the ECC decryption key will be able to access the keys needed for any given file. Furthermore, users will have the capability to manage millions of files with distinct keys using a single key stored on their device.

C. FILE SHARING WITH BLOCKCHAIN

One of the most critical services required by the client in cloud file storage is the secure sharing of files with other clients, and this has been taken into account in the proposed system. In this section, a new modification has been imposed on the blockchain, which is the multi-branch blockchain. It allows the user to add new files to their secure files, which have been shared with them by other clients.

Initially, as illustrated in Figure 5, the sender needs a copy of the recipient's hash code and the block number associated with the hash code. Here, the recipent needs to choose one of the blocks to start a branching blockchain and send it along with its number to the client they want to share the

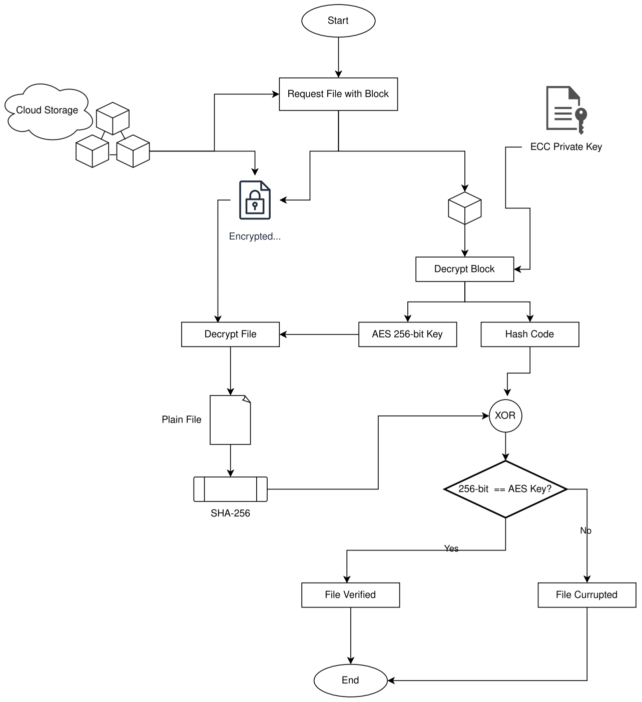

FIGURE 4. File decryption flowchart.

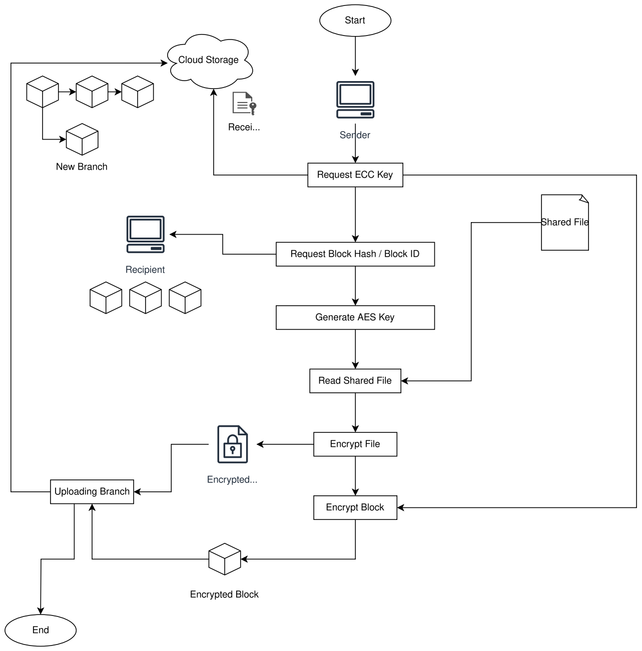

FIGURE 5. File sharing mechanism.

file with. The recipient hashes the file they want to share and then creates a new block and a new encryption key in the same way previously explained for file encryption and block construction. They assign a sequence to the block directly after the number of the block received from the client they intend to share with. Then, they encrypt the file and the block with the recipient's ECC public key, which can be directly requested from the server that holds the clients' public ECC algorithm keys.

The ability to modify the blockchain mechanism and add new branches to it provides the advantage of verifying that a file has been shared from a trusted source. This is achieved by regenerating the block from the file after decrypting it. Additionally, revoking or deleting a file from the blockchain will not affect the main chain that contains the user's primary files uploaded to the server. This flexibility allows for the creation of branches to any extent, with each branch containing multiple files in a hierarchical manner.

V. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed solution has been assessed and analyzed across several stages, encompassing both statistical and detailed mathematical evaluations. This comprehensive analysis extends to its resilience against data analysis attacks and key guessing. These evaluations serve to underscore the significance and robustness of the proposed solution in the requisite encryption scenarios. The performance measurements were conducted using randomly generated synthetic data, created algorithmically for the purpose of simulating various text inputs. Additionally, image data was collected from well-known online sources, serving the purpose of simulating multimedia data.

A. HISTOGRAM ANALYSIS OF IMAGE ENCRYPTION

As depicted in Figure 6, the system under consideration demonstrates its proficiency in the encryption and decryption of grayscale images, specifically the "Cameraman" image, both sized at 256 × 256 pixels.

These images are accompanied by their respective histograms, illustrating their pixel intensity distributions both before and after encryption, employing both the proposed AES model and the conventional AES model. Upon closer examination, it becomes evident that while the pixel values of the encrypted images are uniformly distributed, the histograms of the original, unencrypted images exhibit non-uniform distributions with noticeable variations. Importantly, the suggested encryption method exhibits reduced fluctuations and a more balanced distribution compared to the traditional AES approach. This discernible distinction underscores the enhanced security and greater resilience against statistical attacks offered by the proposed approach.

However, it is worth noting that the computational complexity of the suggested algorithm results in somewhat prolonged computation times.

B. SENSITIVITY ANALYSIS

The proportion of '1's within a binary data stream relative to the total number of bits in the data is commonly denoted as sensitivity or bit density. Bit density serves as a metric to assess the entropy or predictability of a given data stream. In the case of highly random data, it is anticipated that the bit density will closely approach 0.5, signifying an equitable distribution of "0" and "1" bits.

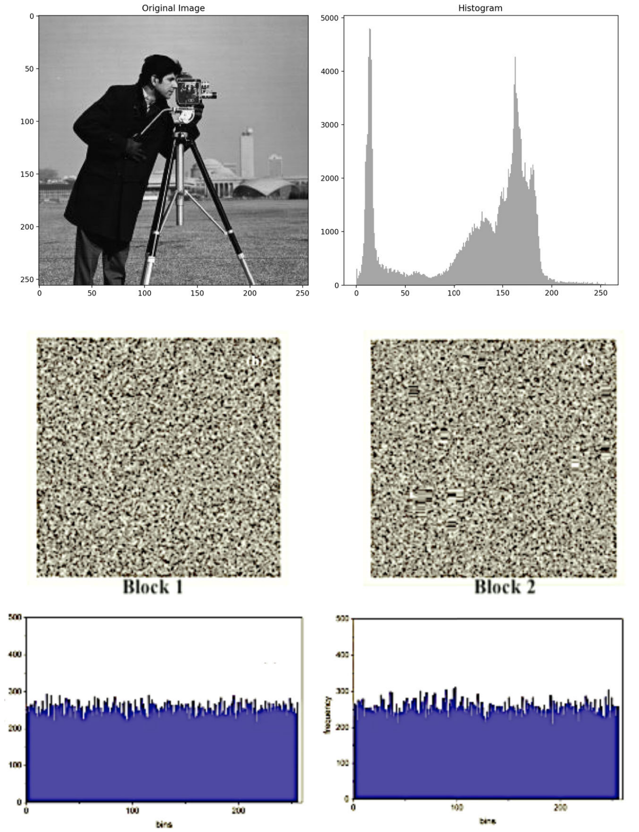

FIGURE 6. Image histogram analysis.

TABLE 1. Encrypted file sensitivity.

File Title	Block No.	File Size	Sensitivity
File1.txt	1	10 KB	57.3
File2.txt	2	100 KB	57.1
File3.txt	3	1024 KB	55.59
File4.txt	4	10 MB	55.58
File5.txt	5	100 MB	55.6

To ensure the security of the proposed solution, the level of sensitivity has been systematically computed for various encrypted files and dynamic encryption keys.

As shown in Table 1, which represents the sensitivity values of encrypted data with different sizes, it is evident that the values are high compared to the benchmark solution. This indicates the effectiveness of the algorithm against data analysis attacks due to the randomness of the data after encryption. Furthermore, dynamic encryption rearranges the positions of bits in the resulting text, even with the same input data, which increases the complexity of data analysis for potential attackers.

Conversely, a comprehensive sensitivity analysis of the encryption keys has been conducted. As illustrated in Table 2, which portrays the sensitivity values of the keys utilized for encryption across diverse block sizes and distinct files, it becomes evident that these values notably exceed those of the benchmark solution. This disparity underscores the algorithm's efficacy in thwarting data analysis attacks, owing to the inherent unpredictability of the data post-encryption. Additionally, the dynamic encryption process, which alters the bit positions within the resulting text, even when employing identical input data, significantly enhances the complexity of data analysis for potential attackers.

TABLE 2. Dynamic keys sensitivity.

File Title	Block No.	File Size	Sensitivity
File1.txt	1	10 KB	58.45
File2.txt	2	100 KB	57.42
File3.txt	3	1024 KB	58.69
File4.txt	4	1 MB	59.04
File5.txt	5	10 MB	57.50

TABLE 3. Dynamic keys sensitivity comparison.

Algorithm	Sensitivity
Proposed Solution	59.04
Dynamic AES Cryptosystem [27]	51.46

TABLE 4. Dynamic keys sensitivity comparison.

Algorithm	Sensitivity
Proposed Solution	60.54
Chaotic Encryption Schemes [1]	42.5

Compared with [27], the proposed solution outperformed the dynamic encryption in that paper in terms of sensitivity which it was produced 51.46 at max compared with 59.04 in the proposed solution.

As shown in Table 3, Compared with [27], the proposed solution outperformed the dynamic encryption in that paper in terms of sensitivity which it was produced 51.46 at max compared with 59.04 in the proposed solution.

Table 4 demonstrated superior performance of the proposed solution in sensitivity compared to the dynamic encryption method presented in [1]. The sensitivity attained a maximum value of 42.5 in the referenced paper, whereas the proposed solution achieved a notably higher sensitivity of 60.54.

C. STATISTICS ANALYSIS

Entropy, also referred to as information density, serves as a metric for quantifying uncertainty within a dataset or a series of bytes. It is a mathematical concept that characterizes the probability or level of difficulty associated with accurately predicting each individual number within a given sequence. Given that genuinely random data is infrequent in typical user data, entropy finds applications in various contexts, with a predominant role in the realms of encryption and compression.

This significance becomes particularly pronounced when dealing with executables that have intentionally undergone encryption through real-time decryption processes [28]. In such cases, the very nature of this encryption approach renders it challenging for antivirus programs to perform an

TABLE 5. File entropy with dynamic key.

File Title	Block No.	File Size	Entropy
File1.txt	1	10 KB	5.564
File2.txt	2	100 KB	5.797
File3.txt	3	1024 KB	5.9680
File4.txt	4	10 MB	5.9971
File5.txt	5	100 MB	6.013

in-depth analysis of these executables while they are stored on disk. This is because the encryption effectively conceals the internal structure of the executable, preventing the inspection for specific strings or patterns.

Moreover, entropy analysis as shown in Table 5 proves invaluable in the detection of files characterized by a high degree of unpredictability [29]. Such files often signify the presence of an encrypted volume or container, a detail that might otherwise remain concealed without the aid of entropy-based identification techniques.

VI. CONCLUSION

In this paper, a comprehensive and innovative solution to address critical security concerns in cloud computing environments has been introduced. The suggested approach utilizes an ECC, AES, and Blockchain hybrid dynamic encryption method, which being a multi-layered defense mechanism ensures high degree of security for sensitive data. In the process, widely regarded security issues in cloud computing have been explained, specifically, the deficiency of centralized key management and the necessity of privacy reinforcement. The proposed answer, acting in two stages, is quite solid to the problems that have been discovered. At first, dynamic AES keys are created to make sure each file is encrypted differently and changes often. This dynamic key generation greatly enhances file-level security, mitigating the risk of compromise. The second phase introduces the use of blockchain technology, providing an immutable and decentralized ledger to securely store encryption keys. By encrypting these blocks with ECC public keys, we ensure that unauthorized access is effectively prevented during both transmission and storage. The combined strength of these components not only bolsters the security of cloud-stored data but also enhances user trust. Users can confidently manage an array of files with distinct encryption keys using a single key stored on their device, while service providers benefit from decentralized key management. In simple words, the suggested way makes a strong and flexible security system that matches with the changing needs of cloud computing. It works well to solve security problems in cloud environments. It makes sure that the data is safe and private, while meeting the different wants of people using it or service providers.

ACKNOWLEDGMENT

The authors extend their appreciation to King Saud University for funding this research through Researchers Supporting Project Number (RSPD2024R1027), King Saud University, Riyadh, Saudi Arabia.

REFERENCES

[1] R. Anandkumar, K. Dinesh, A. J. Obaid, P. Malik, R. Sharma, A. Dumka, R. Singh, and S. Khatak, "Securing e-health application of cloud computing using hyperchaotic image encryption framework," *Comput. Electr. Eng.*, vol. 100, May 2022, Art. no. 107860.

[2] Z. Bashir, T. Rashid, and S. Zafar, "Hyperchaotic dynamical system based image encryption scheme with time-varying delays," *Pacific Sci. Rev. A, Natural Sci. Eng.*, vol. 18, no. 3, pp. 254–260, Nov. 2016.

[3] W. Y. Chang, H. Abu-Amara, and J. F. Sanford, *Transforming Enterprise Cloud Services*. Berlin, Germany: Springer, 2010.

[4] B. Alouffi, M. Hasnain, A. Alharbi, W. Alosaimi, H. Alyami, and M. Ayaz, "A systematic literature review on cloud computing security: Threats and mitigation strategies," *IEEE Access*, vol. 9, pp. 57792–57807, 2021.

[5] N. M. Sultana and K. Srinivas, "Survey on centric data protection method for cloud storage application," in *Proc. Int. Conf. Comput. Intell. Comput. Appl. (ICCICA)*, Nov. 2021, pp. 1–8.

[6] F. Thabit, O. Can, S. Alhomdy, G. H. Al-Gaphari, and S. Jagtap, "A novel effective lightweight homomorphic cryptographic algorithm for data security in cloud computing," *Int. J. Intell. Netw.*, vol. 3, pp. 16–30, 2022.

[7] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Integration of blockchain and cloud of things: Architecture, applications and challenges," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 4, pp. 2521–2549, 4th Quart., 2020.

[8] S. N. G. Gourisetti, Ü. Cali, K.-K.-R. Choo, E. Escobar, C. Gorog, A. Lee, C. Lima, M. Mylrea, M. Pasetti, F. Rahimi, R. Reddi, and A. S. Sani, "Standardization of the distributed ledger technology cybersecurity stack for power and energy applications," *Sustain. Energy, Grids Netw.*, vol. 28, Dec. 2021, Art. no. 100553.

[9] S. Banani, S. Thiemjarus, K. Wongthavarawat, and N. Ounanong, "A dynamic light-weight symmetric encryption algorithm for secure data transmission via BLE beacons," *J. Sensor Actuator Netw.*, vol. 11, no. 1, p. 2, Dec. 2021.

[10] I. Keshta, Y. Aoudni, M. Sandhu, A. Singh, P. A. Xalikovich, A. Rizwan, M. Soni, and S. Lalar, "Blockchain aware proxy re-encryption algorithm-based data sharing scheme," *Phys. Commun.*, vol. 58, Jun. 2023, Art. no. 102048.

[11] O. A. Khashan, N. M. Khafajah, W. Alomoush, M. Alshinwan, S. Alamri, S. Atawneh, and M. K. Alsmadi, "Dynamic multimedia encryption using a parallel file system based on multi-core processors," *Cryptography*, vol. 7, no. 1, p. 12, Mar. 2023.

[12] K. Bhalla, D. Koundal, S. Bhatia, M. Khalid Imam Rahmani, and M. Tahir, "Dynamic encryption and secure transmission of terminal data files," *Comput., Mater. Continua*, vol. 71, no. 1, pp. 1221–1232, 2022.

[13] X. Liu, R. Zhang, and M. Zhao, "A robust authentication scheme with dynamic password for wireless body area networks," *Comput. Netw.*, vol. 161, pp. 220–234, Oct. 2019.

[14] M. Yousefpoor and H. Barati, "Dynamic key management algorithms in wireless sensor networks: A survey," *Comput. Commun.*, vol. 134, pp. 52–69, Jan. 2019.

[15] R. K. Chaurasiya, B. Acharya, and P. Singh, "A comparative survey on lightweight block ciphers for resource constrained applications," *Int. J. High Perform. Syst. Archit.*, vol. 8, no. 4, p. 250, 2019.

[16] S. Hussain, T. Shah, and A. Javeed, "Modified advanced encryption standard (MAES) based on non-associative inverse property loop," *Multimedia Tools Appl.*, vol. 82, no. 11, pp. 16237–16256, May 2023.

[17] S. Ullah, J. Zheng, N. Din, M. T. Hussain, F. Ullah, and M. Yousaf, "Elliptic curve cryptography; applications, challenges, recent advances, and future trends: A comprehensive survey," *Comput. Sci. Rev.*, vol. 47, Feb. 2023, Art. no. 100530.

[18] G. Habib, S. Sharma, S. Ibrahim, I. Ahmad, S. Qureshi, and M. Ishfaq, "Blockchain technology: Benefits, challenges, applications, and integration of blockchain technology with cloud computing," *Future Internet*, vol. 14, no. 11, p. 341, Nov. 2022.

[19] M. Rashmi, P. William, N. Yogeesh, and D. K. Girija, "Blockchain-based cloud storage using secure and decentralised solution," in *Proc. Int. Conf. Data Anal. Insights (ICDAI)*, in Lecture Notes in Networks and Systems, vol. 727, N. Chaki, N. D. Roy, P. Debnath, and K. Saeed, Eds. Singapore: Springer, 2023. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-99-3878-0_23, doi: 10.1007/978-981-99-3878-0_23.

[20] P. Sharma, R. Jindal, and M. D. Borah, "A review of blockchain-based applications and challenges," *Wireless Pers. Commun.*, vol. 123, pp. 1201–1243, 2022. [Online]. Available: https://link.springer.com/article/10.1007/s11277-021-09176-7, doi: 10.1007/s11277-021-09176-7.

[21] F. Thabit, A. P. S. Alhomdy, A. H. A. Al-Ahdal, and P. D. S. Jagtap, "A new lightweight cryptographic algorithm for enhancing data security in cloud computing," *Global Transitions Proc.*, vol. 2, no. 1, pp. 91–99, Jun. 2021.
[22] S. Rehman, N. Talat Bajwa, M. A. Shah, A. O. Aseeri, and A. Anjum, "Hybrid AES-ECC model for the security of data over cloud storage," *Electronics*, vol. 10, no. 21, p. 2673, Oct. 2021.
[23] S. K. Dwivedi, R. Amin, J. D. Lazarus, and V. Pandi, "Blockchain-based electronic medical records system with smart contract and consensus algorithm in cloud environment," *Secur. Commun. Netw.*, vol. 2022, pp. 1–10, Sep. 2022.
[24] S. Virushabadoss and T. P. Anithaashri, "Enhancing data security in mobile cloud using novel key generation," *Proc. Comput. Sci.*, vol. 215, pp. 567–576, 2022.
[25] J. K. Dawson, F. Twum, J. B. Hayfron Acquah, and Y. M. Missah, "Ensuring confidentiality and privacy of cloud data using a non-deterministic cryptographic scheme," *PLoS ONE*, vol. 18, no. 2, Feb. 2023, Art. no. e0274628.
[26] Y. Alemami, A. M. Al-Ghonmein, K. G. Al-Moghrabi, and M. A. Mohamed, "Cloud data security and various cryptographic algorithms," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 13, no. 2, p. 1867, Apr. 2023.
[27] Y. A. Liu, L. Chen, X. W. Li, Y. L. Liu, S. G. Hu, Q. Yu, T. P. Chen, and Y. Liu, "A dynamic AES cryptosystem based on memristive neural network," *Sci. Rep.*, vol. 12, no. 1, p. 12983, Jul. 2022.
[28] K. Lee, S.-Y. Lee, and K. Yim, "Machine learning based file entropy analysis for ransomware detection in backup systems," *IEEE Access*, vol. 7, pp. 110205–110215, 2019.
[29] C.-M. Hsu, C.-C. Yang, H.-H. Cheng, P. E. Setiasabda, and J.-S. Leu, "Enhancing file entropy analysis to improve machine learning detection rate of ransomware," *IEEE Access*, vol. 9, pp. 138345–138351, 2021.

MOHAMMED Y. SHAKOR received the Master of Science degree from the Computer Science Department, College of Science, University of Sulaimani, in 2019. He is currently a Lecturer with the University of Garmian. He is also an accomplished academic professional with a profound expertise in computer science. During this tenure, he exhibited a remarkable aptitude for advanced concepts and demonstrated a keen interest in cutting-edge developments within the field. His research interests include cloud security, cryptography, deep learning, and cloud computing. He has developed innovative methods and techniques to enhance accuracy and efficiency in these fields.

MUSTAFA IBRAHIM KHALEEL received the Ph.D. degree in computer science from Southern Illinois University, USA. He is currently an Assistant Professor with the Computer Department, University of Sulaimani. Since joining the university in 2006, he has made notable contributions to computer science through various research projects and scholarly articles. His research interests include wireless networks, high-performance computing, cybersecurity, cryptography, and cloud computing. He has developed innovative methods and techniques to enhance accuracy and efficiency in these fields. Recently, his work has focused on game theory, artificial intelligence optimizers, and energy-efficient solutions for challenges in these areas. His research, which has significantly enriched computer science and cloud computing domains, is recognized in many esteemed journals and conferences.

MEJDL SAFRAN received the bachelor's degree in computer science from King Saud University, in 2007, and the master's and Ph.D. degrees in computer science from Southern Illinois University Carbondale, in 2013 and 2018, respectively. He is currently a Passionate Researcher and an Educator in the field of artificial intelligence, with a focus on deep learning and its applications in various domains. He is also an Assistant Professor in computer science with King Saud University, where he has been a Faculty Member, since 2008. His doctoral dissertation was on developing efficient learning-based recommendation algorithms for top-N tasks and top-N workers in large-scale crowdsourcing systems. He has published more than 20 articles in peer-reviewed journals and conference proceedings, such as *ACM Transactions on Information Systems*, *Applied Computing and Informatics*, *Mathematics*, *Sustainability*, *International Journal of Digital Earth*, IEEE ACCESS, *Biomedicine*, *Sensors*, IEEE International Conference on Cluster, IEEE International Conference on Computer and Information Science, International Conference on Database Systems for Advanced Applications, and International Conference on Computational Science and Computational Intelligence. He has been leading grant projects in the fields of AI in medical imaging and AI in smart farming. He has been an AI Consultant for several national and international agencies, since 2018. His current research interests include developing novel deep learning methods for image processing, pattern recognition, natural language processing, predictive analytics, and modeling and analyzing user behavior and interest in online platforms.

SULTAN ALFARHOOD received the Ph.D. degree in computer science from the University of Arkansas. He is currently an Assistant Professor with the Department of Computer Science, King Saud University (KSU). Since joining KSU, in 2007, he has made several contributions to the field of computer science through his research and publications. His research interests include machine learning, recommender systems, linked open data, text mining, and the ML-based IoT systems. His work includes proposing innovative approaches and techniques to enhance the accuracy and effectiveness of these systems. His recent publications have focused on using deep learning and machine learning techniques to address challenges in these domains. His research continues to make significant contributions to the field of computer science and machine learning. His work has been published in several high-impact journals and conferences.

MICHELLE ZHU (Member, IEEE) is currently a Professor and the Associate Director of the School of Computing, Montclair State University, NJ, USA. She has published about 150 peer-reviewed articles in various journals and conferences. Her research interests include parallel and distributed computing and big data. Her research projects have been funded by various agencies, such as NSF, DOE, and Oak Ridge National Laboratory.