FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 37, Nº 2, June 2024, pp. 317- 342
https://doi.org/10.2298/FUEE2402317K

Original scientific paper

TOWARDS DATA STORAGE SCHEME IN BLOCKCHAIN BASED SERVERLESS ENVIRONMENT: AES ENCRYPTION AND DECRYPTION ALGORITHM APPROACH

Meenakshi Kandpal[1], Yash Pritwani[2], Chinmaya Misra[3], Amrendra Singh Yadav[4], Rabindra Kumar Barik[3]

[1]School of Computer Engineering, KIIT Deemed to be University, India
[2]Odisha University of Technology and Research, India
[3]School of Computer Applications, KIIT Deemed to be University, India
[4]Department of Computer Science & Engineering, ABV-IIITM Gwalior, India

ORCID iDs:	Meenakshi Kandpal	https://orcid.org/0000-0002-8974-0229
	Yash Pritwani	https://orcid.org/0009-0003-8899-7377
	Chinmaya Misra	https://orcid.org/0000-0001-7043-1349
	Amrendra Singh Yadav	https://orcid.org/0000-0003-0241-3661
	Rabindra K Barik	https://orcid.org/0000-0003-3086-3782

Abstract. *In today's digital age, data storage has become an increasingly important issue. The exponential growth of data has led to the need for secure and private storage solutions. Unfortunately, without a standardized protocol, the security and privacy of sensitive data can be a challenge. However, blockchain technology has emerged as a promising solution for secure data storage. Blockchain's decentralized and immutable nature provides a comprehensive solution for the security and privacy of all types of data. In this research, we propose an innovative framework that leverages the benefits of blockchain technology to securely handle and store data in a serverless environment of distributed nodes. To ensure the highest level of security, we evaluated four encryption algorithms - Blowfish, RC4, DES, and AES - for storing data in a permissioned blockchain network. We found that AES encryption and decryption algorithms provide the best solution for creating a decentralized, immutable coordinate system. Our proposed framework is based on a permissioned blockchain network that enables multiple users to join the network through suitable identity verification processes, and each user is assigned certain special and designated permissions to perform actions. This framework provides a secure and efficient solution for the storage of all types of data, ensuring privacy and security. Our proposed framework offers an innovative solution for secure data storage and management in a decentralized environment. This research has practical implications for organizations that need to store sensitive data securely, and it also contributes to the ongoing development of blockchain technology.*

Key words: *Blockchain, cryptography, AES, Serverless framework, data storage*

Received November 18, 2023; revised January 14, 2024; accepted January 28, 2024
Corresponding author: Rabindra K. Barik
School of Computer Applications, KIIT Deemed to be University, India.
E-mail: rabindra.mnnit@gmail.com

1. INTRODUCTION

The decentralized nature of blockchain enables it to have multiple records of data over thousands of devices and provides a complete solution for highly demanding and secure data. Each transaction on the blockchain network can be accessed by specifying some access methods [1]. Blockchain has allowed organizations to transmit a large amount of data in a very secure and synchronous way with minimal transmission cost, capable of processing millions of transactions each day. This has also effectively reduced the latency and enhanced the throughput of the network. Organizations can now independently process transactions without the intervention of any third-party intermediaries [2]. Blockchain technology also facilitates the exchange of cross-border data which involves organizations or communities connected in a network from different countries. The Inter Planetary File System (IPFS) is a popular framework that can facilitate the transmission of files in decentralized file systems. With the advantage of the blockchain, several communities around the world have achieved much amiability when it comes to the sharing of mutual resources or the transparency of information. The blockchain provides a frictionless infrastructure for the storage of complex and sensitive data, allowing efficient scalability and availability [3].

The use of big data in the past few years has become a global trend as many industries have adopted easier and more efficient techniques for collecting and, at the same time, significantly storing massive amounts of data. Switching to serverless infrastructure in a blockchain network is possibly win-win situation for all the organizations, which are looking forward to build and deploy blockchain-based data transmission and management applications without worrying about the limitations inherent in other approaches [4]. Serverless storing and management of these data reduce the complexity of data handling, thereby providing sustainable services in the blockchain network. Serverless architecture enhances scalability and security by simplifying the integration of multiple secure nodes in the network and reducing the overall exposure of the architecture surface to possible threats [5].

In some cases, the process of negotiation requires global centralization, but groups of supply chain people at a local scale certainly perform well [6,7,8]. Many organizations would rather outsource based on the competency levels of the professionals, but outsourcing being centralized has its own consequences, both good and bad. Scalability remains a crucial challenge for blockchain applications, necessitating ongoing research and innovation [9,10,11]. Solutions such as sharding, off-chain transactions, and consensus mechanism improvements are being explored to address these challenges effectively. The idea of an organization being centralized or decentralized has its own consequences [12,13,14,15]. The emergence of blockchain promised to abolish the desideratum of trusted third parties in multiple domains that require it. The property of virtual immutability is vital for a distributed ledger technology such as blockchain [16,17,18]. The proficiency of Bitcoin technology initiated the abolition of trusted third parties in the field of digital payments and demarcated the first successful case study of blockchain technology in action. An important aspect of achieving decentralization in organizations is to abide by the phrase “Think Global, Act Local”. This aids in capturing most of the benefits that blockchain can bring to the transactions carried out in an organization [19,20,21,22].

Various factors, including network size, transaction throughput, and consensus algorithms, significantly impact blockchain scalability. Addressing these factors is essential for enhancing the efficiency and widespread adoption of blockchain technology [23, 24, 25, 26]. Both

technologies can prove to complement each other, since IoT is facing security challenges and blockchain has become very popular in providing security in multiple domains. The Server Less Approach is a method for businesses to develop and execute applications without having to maintain infrastructure. It allows organizations to delegate architecture tasks to others, including provisioning, scalability, and maintenance. Scaling can be done automatically, and organizations just pay for the services they utilize. Microsoft Azure, AWS Lambda, and Google Cloud Functions are some examples [27,28,29]. The range of capabilities of blockchain and serverless can be the basis for the merger of the two technologies. Blockchain being a decentralized platform is inevitable and requires proof in all fields wherever businesses think of implementing blockchain at the backend [30,31,32,33]. Thus, blockchain can also aid in securing IoT sensory geospatial data and can prove itself as an indispensable ledger technology. With the assistance of distributed storage mechanisms along with encryption techniques, blockchain can immensely empower the IoT sensory data with security. Furthermore, without the interference of humans, the integrity of the data can be well-preserved through this merger technology [34,35,36,37].

1.1. Motivation and objectives

The data that the application contains is the pivotal and core component of any real-world application. So, we are working on this research paper to develop a model that secures sensitive data storage using blockchain and serverless framework. Hence, the integrity of data and users' privacy will never be compromised. Our proposed model's main goal is to create a blockchain-based data storage framework for storing and securely encrypting and decrypting data to and among serverless frameworks. The secure data storage in the blockchain can be achieved by utilizing AES encryption and decryption algorithm which we have chosen after thorough comparison of 4 industry standard encryption algorithms - Blowfish, RC4, DES and AES.

1.2. Problem Formulation

The problem with storing big sensitive data in the blockchain lies in the lack of encryption, which can leave sensitive data vulnerable to unauthorized access. With serverless frameworks being accessible to the public, there is an increased likelihood of data breaches, making it imperative to secure the data. One potential solution to this problem is the implementation of the Advanced Encryption Standard (AES), which is widely considered to be a more secure encryption algorithm than Blowfish.

AES uses a symmetric key algorithm, which means that the same key is used for both encryption and decryption of the data. In comparing AES to other encryption algorithms, we have considered that AES's symmetric key framework provides a higher security level than asymmetric key algorithms, due to its unified key approach, and shows superior efficiency and security over other symmetric algorithms such as Blowfish. This dual-edge of better security management and operational efficiency makes AES an optimal choice for our proposed data storage solution in serverless blockchain frameworks. Additionally, AES is a widely recognized encryption standard, which means that it is likely to be more widely implemented than other algorithms, making it a more practical solution for securing big data in the blockchain.

In evaluating alternatives, we considered algorithms such as Blowfish, RC4, and DES. However, AES was selected for its unparalleled security characteristics, notably its resistance

to known types of cryptographic attacks, and its efficiency in processing, which is critical in the serverless blockchain environment we focus on. Overall, using AES encryption and decryption algorithms can help to minimize the risks associated with storing data in the blockchain. By providing a secure method of encrypting the data, the risk of unauthorized access can be significantly reduced, ensuring the integrity and privacy of the user and company data.

1.3. Contributions

The following contributions, which make up the current research article, are organized as follows:

- Processing blockchain environment within serverless framework has been briefly described.
- A model based on the AES encryption and decryption algorithm has been proposed for secure data storage in the blockchain.
- Further, the industry standard algorithms are compared along with implementation results to better understand why AES was chosen as most suitable algorithm for data storage.

1.4. Limitations

Permissioned blockchain network requires internet connection to establish connection with the end user. In such cases, when standard security protocols are not adopted, they become vulnerable to various attacks through immutable techniques such as validation through consensus mechanisms and cryptographic security measures. By adopting and implementing a strong network design, we may overcome this issue.

The user authentication information is the sole approach to access the Blockchain as a Service (BaaS), network. If the information or private keys are stolen by someone, they will be able to access the network. This can also be handled by implementing MFA (Multi Factor Authentication) at login validation on the network. Though most of the blockchain networks are considered impregnable, there are still many weaknesses to be handled.

1.5. Organizations

The structure for the rest of the paper is as follows: Section 2 explains the background studies for the present research objectives. Section 3 outlines the related work about data storage in a blockchain environment using the serverless framework for data, and the same is also represented in tabular form. Section 4 provides the proposed framework for data storage using the AES algorithm and a comparison of all industry-standard encryption algorithms. This section elaborates on the encryption and decryption of the AES algorithm along with its key parameters. Section 5 provides the results, discussions, and performance evaluation of all algorithms on various characteristics. Section 6 draws the summary and concluding remarks of the present research paper.

2. BACKGROUND

2.1. Serverless with Blockchain Paradigm

The Server Less Approach (SLA) is a method for businesses to develop and execute applications without having to maintain infrastructure. It allows organizations to delegate architecture tasks to others, including provisioning, scalability, and maintenance. Scaling can be done automatically, and organizations just pay for the services they utilize. Microsoft Azure, AWS Lambda, and Google Cloud Functions are some examples [38,39,40,41]. Fig. 1 shows the operational workings of the serverless approach.

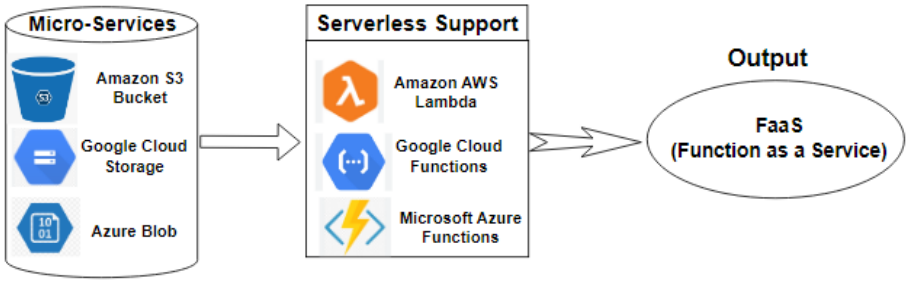

Fig. 1 A generalized diagram for Serverless Approach [39]

Benedict [40] focused on the fact that now a day's Serverless Application Architectures are getting popular, and so is serverless application development. Here," serverless" doesn't mean without using a server; it is a misnomer here. Serverless means developing application programs and deploying them into a cloud environment without worrying about the backend, such as managing the infrastructure, scaling the services, managing databases, etc., which are all provided by the cloud services provider. Again, societal IoT-enabled applications like smart city applications, agricultural applications, environment monitoring systems, etc. are of government interest, which is slowly moving towards serverless computing. This research article proposes an innovative Blockchain-Enabled Serverless Architecture for societal IoT applications, which can overcome the existing issues in current IoT architectures such as inefficiency in energy utilization, resource underutilization, etc. Fig. 2 visually represents the framework of a serverless blockchain system for Internet of Things enabled societal applications, with a focus on the AQMS application. Architecture has been used to characterize the entities centered at the Edge level, Fog level, and Cloud-Blockchain/serverless level. The Edge Level (Sensor and Actuator Devices) have limited computing power and are power-constrained. These sensor devices process the trigger for sending sensor data to the computer nodes whereas Fog-Level Nodes mostly based on containers or docker systems, these nodes use serverless capabilities to minimize the need for constant powering on of the containers. This level of implementing serverless operations lowers the overall system cost in addition to saving energy. At the cloud blockchain level, transactions pertaining to the accountable manufacturing units or automobiles linked to the air quality data are sent to blockchain networks. A selected group of peers and signing authorities work together on the blockchain network to validate and authenticate transactions that rely on chaincodes. Moreover, at the cloud-level of AQMS, a number of additional features could be implemented as serverless features and other end-to-end IoT societal applications would undoubtedly save energy and use computer resources more effectively if serverless-based services were implemented [42,43,44,45].

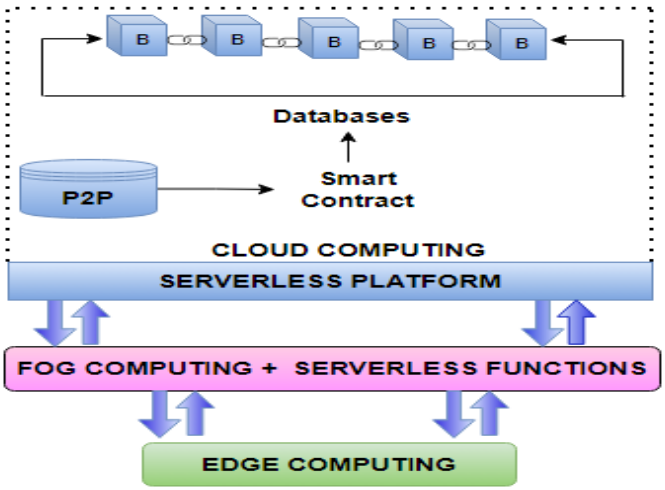

Fig. 2 Integration of Serverless and blockchain for IoT societal applications [40]

The serverless framework plays a pivotal role in our proposed solution, as it allows for efficient resource utilization and scalability. This framework, by its nature, eliminates the need for dedicated server management, leading to cost reduction and enhanced flexibility. Within the context of blockchain data storage, this translates to a more agile and adaptable infrastructure, capable of responding to varying workload demands without compromising security or performance [46,47,48,49].

3. Related Work

3.1. Data Storage in Blockchain

The cloud database is a conventional and most popular method to store data, but it's highly misused. The centralization of data is the main shortcoming of cloud storage. Data is exposed because it was not encrypted at the time of transactions. The most significant unit on the planet is data. Recent technologies are developing for the safe storage, processing, and analysis of data [50,51,52]. The probability of hacking increases due to the centralization of information. Blockchain technology requires decentralization in order to provide a complete solution for highly demanding and sensitive operations. Several modern-day computing infrastructures require the secure processing, storage, and retrieval of multiple records over distributed platforms [53,54,55,56]. The decentralized nature of blockchain enables it to have multiple records across thousands of devices in the network [57,58,59]. Earlier research in the domain of data storage on blockchain networks by some researchers primarily focused on the implementation of the three major data security principles: privacy, integrity, and security shown in Table 1. The methodology undertaken by the researchers is as follows:

Jyoti and Chauhan [1] used Fuzzy Extractor, ECCST & SHA-3 algorithms for data security. Blockchain-enabled applications in serverless computing provide a secure way of provenance data collection and storage in a cloud environment. This research paper is mainly about how we can collect and store provenance data securely in a cloud environment using Blockchain-based smart contracts. It claims that the proposed model is efficient in performance and time compared to the other traditional approach and has been experimentally verified.

Alsayegh et al. [8] make use of conditional proxy re-encryption and asymmetric searchable encryption to achieve privacy, access control, and data security. Proof of permission ensures the availability of the system on consortium blockchains. The security analysis shows that the suggested protocol is capable of achieving the stated security objectives. Additionally, fundamental cryptography created the protocol on the Ethereum platform. The analysis's findings imply that the protocol that was created is computationally effective.

Park et al. [10] focused on a proxy re-encryption mechanism for safe storage of EHR. Additionally, the blockchain uses smart contracts for access management, enabling secure and effective EHR sharing. The suggested strategy resolves the aforementioned security issues through security analysis and experiments that demonstrate how well the system works.

Bebortta et al. [14] discussed many well-known, scalable serverless frameworks, such as Google Cloud Functions, Microsoft Azure Functions, and Amazon Web Services (AWS) Lambda, intended for handling of big data. Several prominent methods now in use for evaluating large data have been discussed and pointed out their drawbacks. The author also discussed how the suggested framework can be used with the Cloud Geographic Information System (GIS). A description of several cutting-edge tools and technologies that are pertinent to the problem domain is discussed. The suggested architecture incorporates serverless support to ease time restrictions and enhance the efficiency of processing high-dimensional hyperspectral data.

Liang et al. [18] proposed a blockchain-based 4th revolution in the safe storage of enormous data. The huge amount of data stored and communicated lack in maintaining the veracity and altering of data. This scheme refines the decentralization, altering proof and concurrent observation of storage structure. Moreover, when a few nodes become unsuccessful in storing the data, at that time local nodes heal up the data and bring current updated information about the authenticity and integrity of data. The characteristics of blockchain like consensus and smart contracts method help in speedily healing up of data and decrease the resources overhead in the storage mechanism. The outcome of the scheme increases the heal-up rate by 9% and 8.6% encouraging the safety of concurrent data.

Chen et al. [21] used a threshold proxy re-encryption mechanism. The findings of this research work demonstrate that the suggested design may accommodate high data privacy and security requirements without introducing a tolerable temporal lag.

Tosh et al. [28] in their research paper published use of SHA-256 and Block clouds and was able to accomplish data security. The author outlined many research obstacles and potential weaknesses that must be fixed in order to implement Block cloud. The PoS consensus mechanism must be included in the future for examining the privacy and safety features.

Table 1 Literature Review on data storage in blockchain ecosystem

Author	Year	Method	Privacy	Integrity	Security
Jyoti and Chauhan [1]	2022	Fuzzy Extractor, ECCST and SHA-3	✓	✓	✓
Alsayegh et al. [8]	2022	Searchable encryption and Proxy re-encryption	✓	✓	✓
Park et al. [10]	2021	EHR Model, Proxy re-encryption	✓	✓	✓
Bebortta et al. [14]	2020	Geospatial Serverless	✗	✓	✗
Liang et al. [18]	2020	Secure data storage & recovery scheme	✓	✓	✓
Chen et al. [21]	2021	Threshold Proxy re-encryption mechanism	✓	✓	✓
Tosh et al. [28]	2019	SHA 256, Blockclouds	✗	✓	✗
Liu et al. [29]	2019	Double chain scheme	✗	✓	✗
Zhao et al. [35]	2022	Location based services	✓	✓	✓
Do and Ng [37]	2017	BlockDS	✓	✓	✓
Alsulbi et al. [42]	2021	Ethereum Blockchain, Mobile Agent	✓	✓	✗
Jimenez et al. [48]	2022	SIMILE, DGGS	✓	✓	✓

Liu et al. [29] explain the difficulties of storage imbalance in blockchain systems. The author proposed a double chain safety scheme with a heuristic, genetic and tabu search algorithm to resolve the issue. The performance is appraised by parameters like running time, storage fitness, and memory cost.

Do and Ng [37] present a proposal known as Block DS. The blockchain framework imposes the integrity of data assisted by a consensus-based blueprint. Conventional cloud used to rely upon many storage providers, considered as reliable third parties and this type of storage has issues of availability and security of data. The blockchain framework helps in providing distributed and decentralized data storage with the provision of security and searching. The main focus of the paper is on keyword searches done on encrypted data.

Alsulbi et al. [39] in their research paper used Ethereum blockchain and Mobile agents, for storing the big data in a secured the empirical findings and framework assessment demonstrate that the suggested framework offers a practical approach to safe data storage in a Big Data environment.

Yadav and Kushwaha [43] proposed a trust value-based efficient consensus mechanism based on land registry management. However, the parties involved can also undertake traditional operations at any moment, which helps them strike a balance between the process's reliability and cost-effectiveness.

Ako Muhamad Abdullah [52] in his research paper provided an overview of AES algorithm and explain several crucial features of this algorithm in details and demonstration some previous research that have done on it with comparing to other algorithms such as DES, 3DES, Blowfish etc.

P. Princy [53] in his research paper discussed about comparison of various symmetric key algorithms DES, AES, Blowfish, RC4 and RC6 through their characteristic and properties like key size, block size, round, structure, flexibility and features along with a survey.

Jawahar Thakur and Nagesh Kumar [54] in their paper provided a fair comparison between most common symmetric key cryptography algorithms: DES, AES, and Blowfish. Main concern they covered was the performance of algorithms under different settings, their presented comparison takes into consideration the behavior and the performance of the algorithm when different data loads are used. The comparison is made based on speed, block size, and key size and the simulation was implemented using Java.

Masram et al. [55] in their paper provided an analysis and comparison of symmetric key ciphers like RC4, AES, Blowfish, RC2, DES, Skipjack and Triple DES based on encryption time along with file features like data types, data size, data density and key sizes.

Douglas Selent [56] in his research paper discussed in details mathematical implementation of encryption and decryption using Advanced Encryption Standard (AES) algorithm along with code examples and steps. Furthermore, authors proposed a efficient search mechanism using sidechain and reduces the storage problem of blockchain data. The author suggested a blockchain application in various fields [60,61,62,63].

4. METHODOLOGY

4.1. Proposed algorithm for data storage

A data block containing sensitive information requires high security measures for protection. To ensure safe storage, we propose the use of the AES (Advanced Encryption Standard) algorithm. The data block is divided into smaller divisions known as shards, which are individually encrypted and hashed using the AES algorithm. The encrypted shards are then synced with the blockchain ledger, ensuring data integrity and immutability. To ensure the safe storage of data, it is distributed across multiple cloud providers. The AES algorithm is applied to each shard before being saved in the blockchain, offering robust encryption with a high level of security. In contrast to previous research conducted by other scholars, we propose the use of the AES algorithm on a permissioned blockchain. This introduces unique functionalities such as Blockchain-as-a-Service (BaaS), allowing many users to join the network through an identity verification process. Each individual is assigned special and designated permissions for performing actions. The permissioned blockchain offers significant advantages over public blockchains in terms of cost and power efficiency, as well as the ability to authorize individuals to join the network. To further enhance the security of sensitive data, we propose a multiple encryption approach. This involves encrypting the already encrypted data, either using the same or different algorithms. This makes it difficult for any system to decrypt the data in transit.

The AES algorithm is a widely used encryption method known for its strength, efficiency, and robustness. It is a symmetric encryption method, meaning that the same key is used for both encryption and decryption. The encryption key and the decryption key of the AES algorithm turn private data into ciphertext, ensuring that the data remains confidential. In summary, the use of the AES algorithm on a permissioned blockchain offers a secure and efficient way to store and protect big sensitive data. The combination of multiple encryption and distribution across cloud providers ensures that the data remains secure and tamper-proof. The AES (Advanced Encryption Standard) algorithm

utilizes block cipher encryption. However, AES has a larger block size of 128 bits, adding a higher level of security and complexity to the encryption process. The key size for AES is variable, with options for 128, 192, or 256-bit sizes. The AES algorithm utilizes subkeys, known as round keys, generated from the original key. In the standard implementation of AES, there are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Each round involves a series of substitutions and permutations, providing a high level of security. In terms of substitution boxes, AES has four tables known as S-boxes, each with 256 entries that are 32 bits long. These tables are used in the substitution process during encryption.

Our proposed model as shown in Fig. 3 for secure data storage involves distributing incoming data across multiple databases, which can then be stored on multiple machines as shard data. The data is then distributed across multiple cloud providers to maintain decentralization. The AES algorithm is applied to the shard data, providing a high level of security, before being stored in the blockchain offered by different cloud providers. Overall, the use of the AES algorithm for data storage provides a high level of security and efficiency, with a larger block size and variable key sizes, making it difficult to break the encryption. The combination of multiple cloud providers and the use of blockchain technology provide a secure and decentralized approach to data storage.

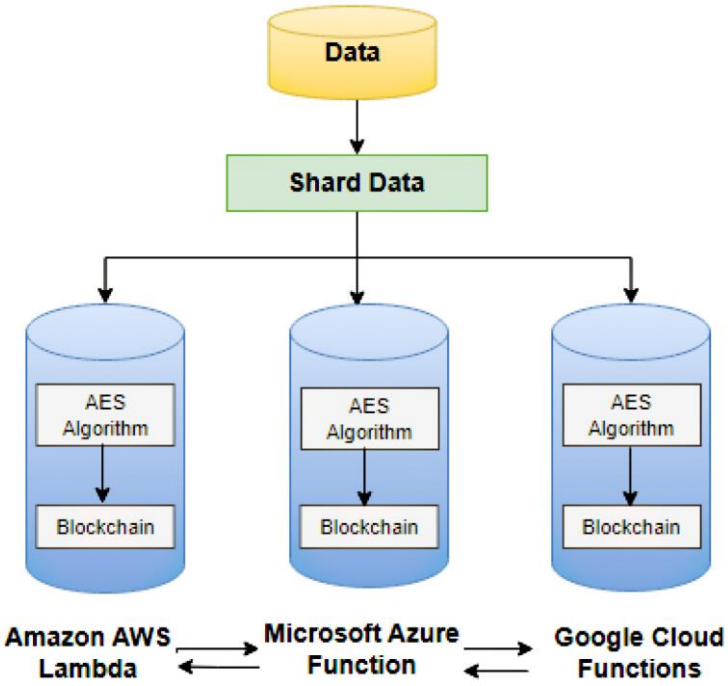

Fig. 3 Secure Data storage using AES algorithm

4.1.1. Sub Key Generation for AES Algorithm:

1. **Key Expansion:**
   Start with a 128-bit (16-byte) original encryption key, denoted as `$Key[0]$` .
2. **Round Constants:**
   AES uses a set of round constants denoted as `$Rcon[i]$` , where `$i$` is the round number.
   These round constants are defined as follows:
   `$Rcon[1]$ = 0x01`, ` $Rcon[2]$ = 0x02`, ` $Rcon[3]$ = 0x04`, ..., ` $Rcon[10]$ = 0x80`.
3. **Subkey Generation Core:**
   For each round i from 1 to 10 (AES-128 has 10 rounds), perform the following steps:
   a) **Rotate Word:**
      Rotate the previous word `$Key[i-1]$` left by one byte.
      Let's denote this rotated word as **`$temp$`**.
   b) **Substitute Bytes:**
      Apply a byte substitution (S-box) operation to each byte in **`$temp$`.**
      The S-box is a predefined substitution table.
      Denote the result as **`$SubBytes(temp)$`.**
   c) **Round Constant XOR:**
      XOR **`$SubBytes(temp)$`.** with a round constant **`$Rcon[i]$`.**
      Denote the result as **`$temp2$`.**
   d) **XOR with Previous Word:**
      XOR `$temp2$` with the previous word in the key schedule,
      `$Key[i-4]$`
      Denote the result as `$Key[i]$`.
4. **Key Schedule:**
   After completing the above steps for each round, you have generated 11 round keys in total: **`$Key[0], Key[1], Key[2], ....., Key[10]$`**.These keys are used for the 10 rounds of AES encryption.
   Here's a more concise mathematical expression for subkey generation:
   For each round ` $i$ ` from 1 to 10:
   - ` $temp = RotateWord(Key[i-1])$`
   - `$temp = SubBytes(temp)$`
   - `$temp2 = temp\ XOR\ Rcon[i]$`
   - `$Key[i] = temp2\ XOR\ Key[i-4]$`

   Where:
   - `$RotateWord(word)$` rotates the bytes in `$word$` to the left by one byte.
   - `$RotateWord(word)$` substitutes each byte in `$word$` using the AES S-box.
   - `$Rcon[i]$` is the round constant for round ` $i$ `.
   - `$Key[i]$` is the round key for round ` $i$ ` .

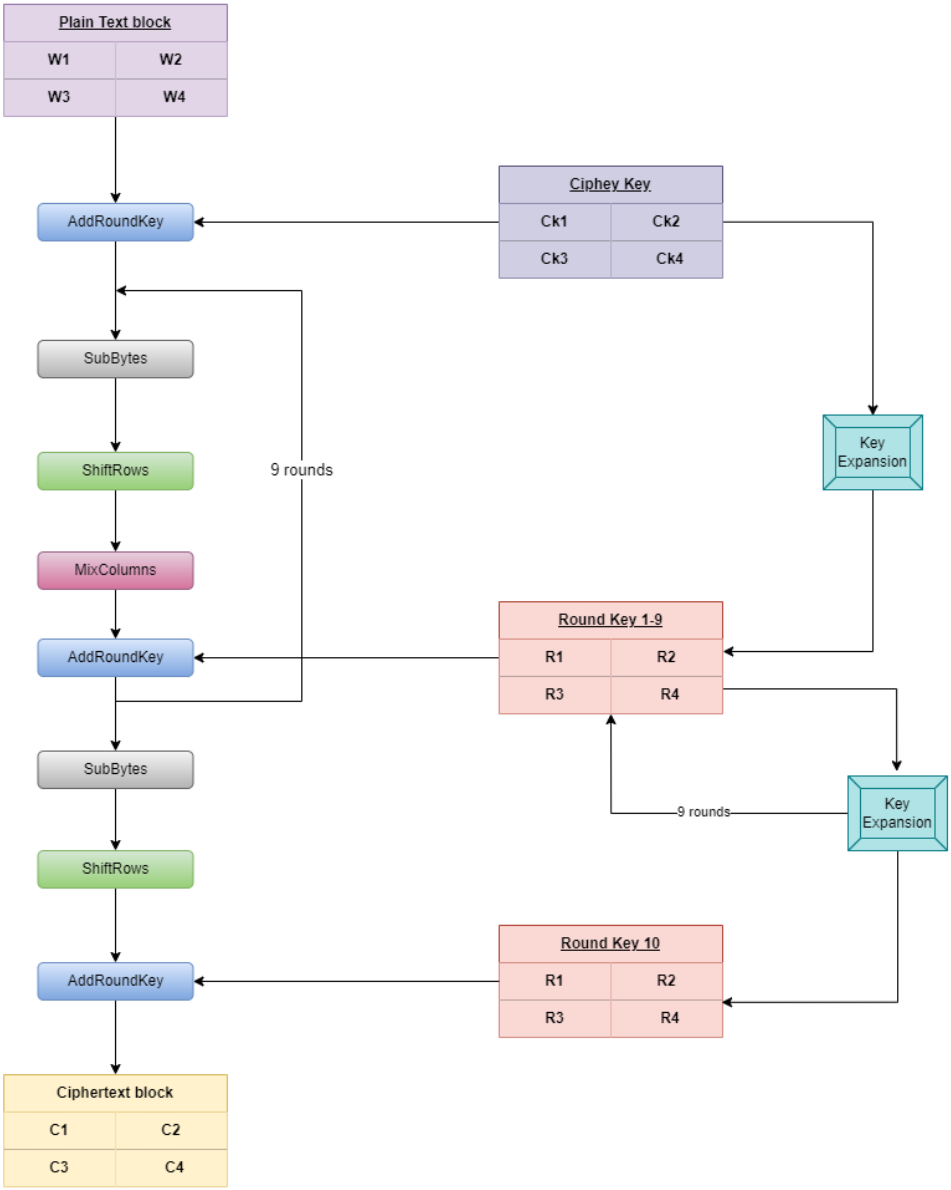

Fig. 4 Subkey array generation phase

This process generates the round keys used in the AES encryption rounds, ensuring that each round key is derived from the original encryption key and provides the necessary security for the algorithm as shown in Fig. 4.

Fig. 4 provides a detailed representation of the AES encryption process. At the outset, the 'Plain Text block' contains the data to be secured, divided into four distinct word blocks (W1, W2, W3, W4). This data undergoes the initial 'AddRoundKey' stage, where it is combined with the 'Cipher Key' (Ck1, Ck2, Ck3, Ck4) through a key expansion process. Subsequently, the data passes through a series of transformations - 'SubBytes', 'ShiftRows', and 'MixColumns' - across nine rounds. Each round utilizes a set of round keys (R1, R2, R3, R4), derived from the main cipher key, to further scramble the data. The final round omits the 'MixColumns' phase and concludes with an 'AddRoundKey' step, outputting the 'Ciphertext block' (C1, C2, C3, C4), which is now securely encrypted. This comprehensive diagram underscores the complex yet systematic approach AES employs to ensure data security.

4.1.2. Initialization Substitution Boxes

An S-Box (Substitution-box) is a fundamental building block of symmetric key algorithms in cryptography. It performs substitution and is often used to create a nonlinear relationship between the key and the cipher text. Most symmetric key algorithms, including Blowfish, use S boxes. Both the encryption and decryption processes in Fig. 5 need the use of 4 substitution boxes (S-boxes), each of which has 256 entries (S[i][0] through S[i][255], 0&lei&le4 and each of which is 32 bits).

These are the following steps for the S-Box Generation for AES Algorithm.

1. An S-Box (Substitution-box) is a critical component of symmetric key algorithms in cryptography. It is used to perform substitution to create a non-linear relationship between the key and the ciphertext.
2. The AES algorithm uses a fixed set of S-Boxes, with each S-Box containing 256 entries.
3. There are a total of 10 S-Boxes used in AES, one for each round of the encryption process.
4. Each entry in an S-Box is represented by an 8-bit value, which is substituted for the corresponding 8 bits in the input data.
5. The S-Boxes are generated using a combination of substitution and permutation operations applied to a fixed set of input values.

In summary, the AES algorithm uses a fixed set of S-Boxes, with each S-Box containing 256 entries. There are 10 S-Boxes used in the algorithm, one for each round of the encryption process. Each entry in an S-Box is represented by an 8-bit value, which is substituted for the corresponding 8 bits in the input data. The S-Boxes are generated using a combination of substitution and permutation operations applied to a fixed set of input values. Algorithm 1 and Pseudocode 1 show the detailed workflow of process.

Algorithm 1: AES algorithm

Input:
- For encryption: None
- For decryption: Ciphertext, Set of subkeys, S-boxes

Output:
- For encryption: A 256-byte initialized S-box
- For decryption: Plaintext

Encryption (S-box Initialization):
- Initialize an empty 256-byte array, referred to as the S-box.
- Fill the first 16 bytes of the S-box with the values 0x00 to 0x0F.
- **For each byte i from 16 to 255 do:**
  1.1. Compute the multiplicative inverse of i in the finite field GF(2^8) using the Extended Euclidean Algorithm.
  1.2. Apply an affine transformation to the result, which involves a bitwise XOR with a fixed constant and the inverse function of Galois field multiplication.
  1.3. Assign the resulting byte to the corresponding position in the S-box.
- **End for**
- Return the completed S-box.

Decryption:
- To decrypt the ciphertext, it must first be translated from its encrypted format into plaintext.
- Use the same set of subkeys and S-boxes as used in encryption, but in reverse order.
- **Start decryption with the last subkey (from the 10th) and proceed in reverse order to the first subkey.**
- Continue the decryption process using each subkey sequentially.
- Return the plaintext when all subkeys have been applied in reverse order.

Pseudocode 1: AES algorithm

Start
Declare S-Box [16][16] array
Declare temp and constant variables
for i=0 to i < 16 do
for j=0 to j < 16 do
*temp = i*16 + j*
constant = 0x63
for k=0 to k < 4 do
byte = temp & 0xff
temp = temp >> 8
if k == 0 then
subByte = byte
else
subByte = subByte XOR byte
end if
end for
S-Box[i][j] = subByte XOR constant
end for
end for
Display S-Box array
End

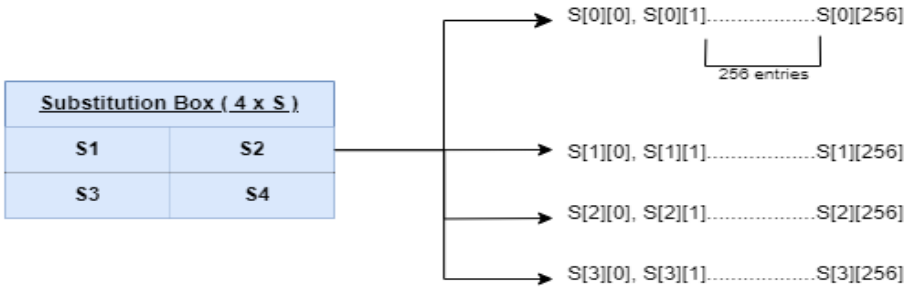

Fig. 5 Substitution box Initialization

4.1.3. AES Encryption Algorithm

In the AES encryption process, the plain text is divided into blocks of 128 bits each. The AES algorithm uses a substitution-permutation network (SPN) structure. In the substitution phase, the 16 bytes (128 bits) of the block are substituted using a substitution box (S-box) and in the permutation phase, rows and columns of the block are shifted and mixed using a matrix multiplication operation. *(Code link can be checked in reference section)*

The AES algorithm has 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. In each round, a round key is generated from the original key using key expansion. The generated round key is then used in the substitution and permutation phases. In the final round, the permutation phase is omitted. The AES encryption process can be summarized as follows:

- Key expansion: The original key is expanded into a set of round keys.
- Initial round: The first-round key is XORed with the plaintext block.
- Rounds: The AES algorithm performs a specified number of rounds, each consisting of four operations:
  a. SubBytes: Bytes of the block are substituted using an S-box.
  b. ShiftRows: Rows of the block are shifted by a certain number of bytes.
  c. MixColumns: Columns of the block are mixed using a matrix multiplication operation.
  d. AddRoundKey: The round key is XORed with the block.
  e. Final round: The final round is similar to the other rounds, except that the MixColumns operation is omitted.

The resulting block after the final round is the cipher text. The decryption process is the inverse of the encryption process, where the round keys are used in reverse order. Fig. 6 and 7 can be referred for the AES encryption algorithm to represent the flow and function used in the algorithm, respectively. Fig. 8 illustrates the flow of Decryption in AES algorithm. Pseudocode 2 describes the AES encryption process. Pseudocode 3 presents the AES decryption workflow.

Pseudocode 2: AES Encryption

Start

- Input: Plain Text 128-bits
- Output: Cipher Text 128-bits

- The Plain Text (128-bits) is divided into 4 parts of 32-bits each.
- AddRoundKey operation is performed on the input data with the first round key.
- The data is passed through the SubBytes transformation using the S-box.
- The data is then shifted row-wise in the ShiftRows transformation.
- The MixColumns transformation is applied on the shifted data.
- Steps 2-5 are repeated for each round key until the final round.
- In the final round, the SubBytes transformation and ShiftRows transformation are applied
- AddRoundKey operation is performed on the output data with the final round key.
- The resulting 128-bit data is the Cipher text.

End

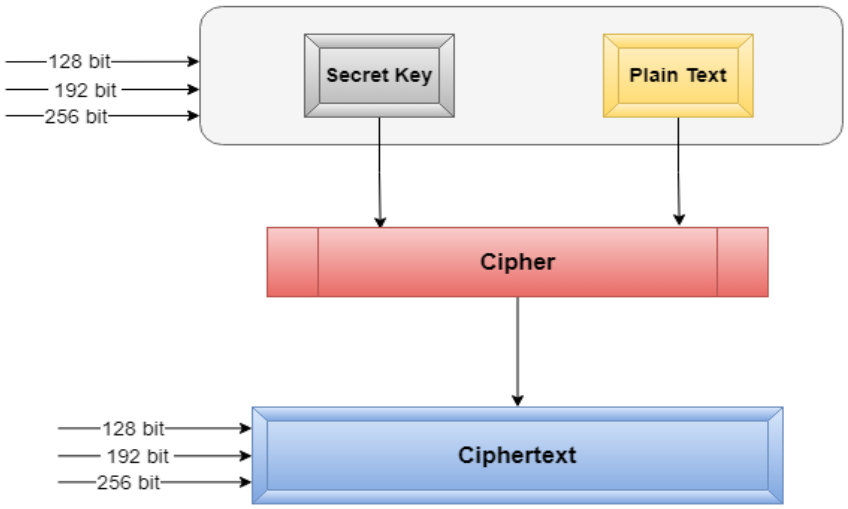

Fig. 6 AES Based Encryption flow

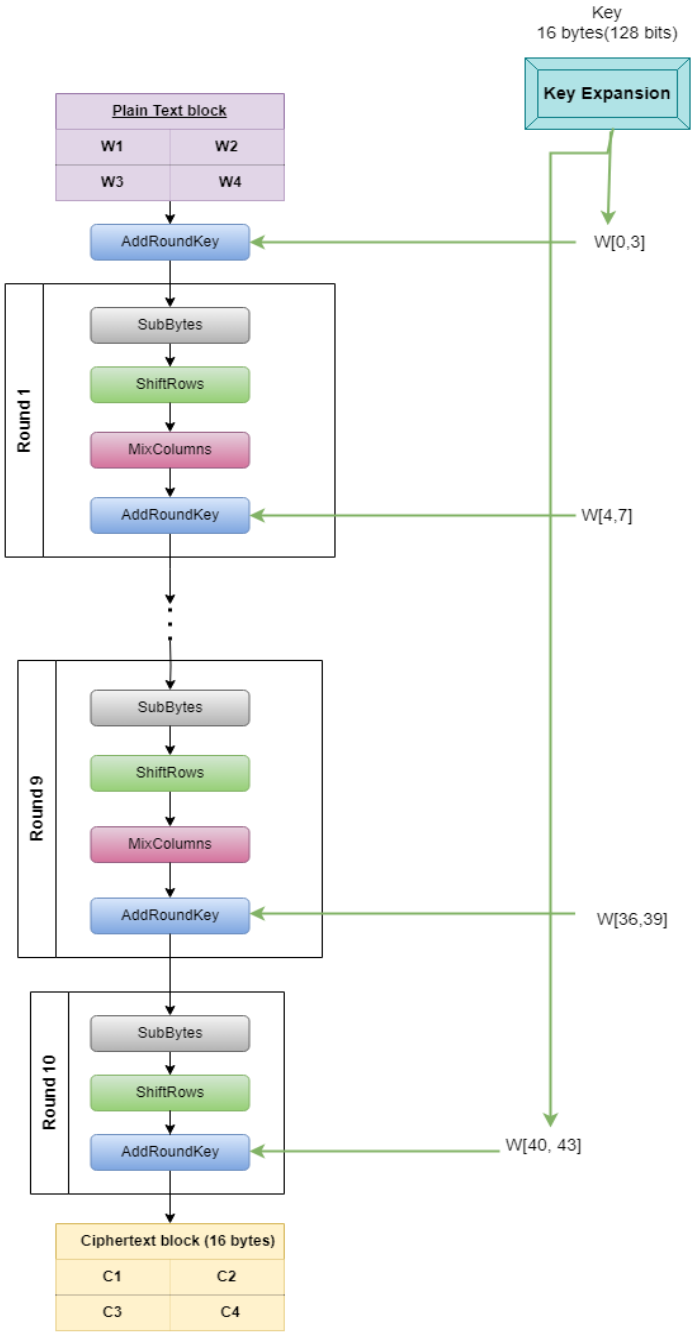

Fig. 7 AES encryption Function used in the above flow diagram

Pseudocode 3: AES Decryption

Start

- Input: The file is in the form of cipher text.
- Output: It is converted into plain text.

a) The decryption process is similar to the encryption process but with the use of the same set of keys in reverse order.
b) The subkeys are generated by the key schedule algorithm in reverse order, starting with the last subkey.
c) The cipher text is divided into blocks of 128 bits.
d) The first block is XORed with the last subkey.
e) Then the block goes through the inverse rounds of substitution, shifting, and mixing using the inverse S-box, inverse shift rows, and inverse mix columns operations.
f) After completing the inverse rounds, the second last subkey is XORed with the output.
g) This process continues until the first subkey is used.
h) The final block is the plain text.

End

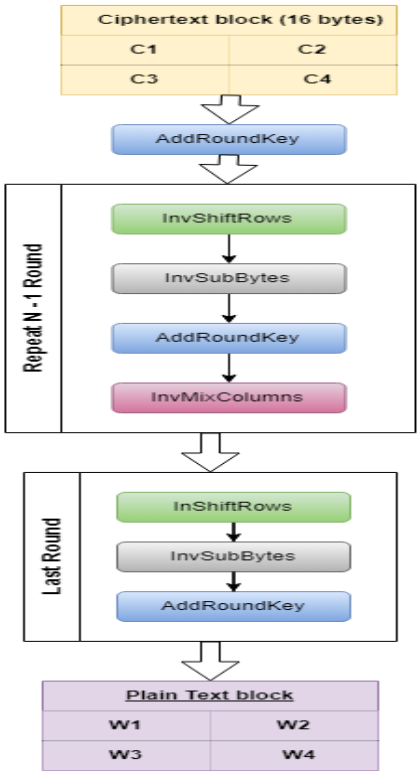

Fig. 8 AES based decryption flow

4.1.4. Comparison with other encryption algorithms

Table 2 presents a comparison of four encryption algorithms - Blowfish, RC4, AES, and DES. The purpose of this is to examine the performance, security, and efficiency of these algorithms in securing sensitive data and understand why AES is chosen over the other industry standard algorithms *(code link can be checked in reference section)*.

Table 2 Comparison of various symmetric key encryption algorithms

Feature	Blowfish	RC4	DES	AES
Block size	64 bits	N/A	64 bits	128 bits
Key size	32 to 448 bits	40 to 2048 bits	56 bits (56-bit version) or 168 bits (Triple DES)	128, 192, or 256 bits
Rounds	16	N/A	16	10, 12, or 14
Mode of operation	ECB, CBC, CFB, OFB, PCBC	Stream cipher	ECB, CBC, CFB, OFB	ECB, CBC, CFB, OFB, CTR, GCM
Key schedule	Precomputed S-boxes and a complex key-dependent permutation	Key scheduling algorithm that generates a keystream	Key-dependent permutation of the input data	Key expansion using Rijndael's key schedule
Encryption latency	Fast software implementation on a variety of platforms	Fast software implementation on a variety of platforms	Relatively slow due to the complexity of the algorithm and low data throughput	Fast on modern CPUs with hardware support for AES
Decryption latency	Fast software implementation on a variety of platforms	Fast software implementation on a variety of platforms	Relatively slow due to the complexity of the algorithm and low data throughput	Fast on modern CPUs with hardware support for AES
Security level	Strong resistance against known cryptanalytic attacks	Vulnerable to biases in the key scheduling algorithm and poor random number generation, leading to possible attacks	Vulnerable to brute-force attacks due to small key size	Strong resistance against known cryptanalytic attacks
Performance	Fast software implementation on a variety of platforms	Fast software implementation on a variety of platforms	Relatively slow due to the complexity of the algorithm and low data throughput	Fast on modern CPUs with hardware support for AES
Use cases	Widely used in software applications that require strong encryption, such as secure communication protocols and file encryption tools	Widely used for wireless networks and online encryption protocols	Widely used in the past but now considered obsolete due to small key size	Widely used in software applications that require strong encryption, such as secure communication protocols and file encryption tools
Algorithm complexity	More complex algorithm than RC4 and DES, may require more effort to implement and understand	Simple algorithm that is easy to implement and understand	Relatively simple algorithm that is easy to implement and understand	More complex algorithm than Blowfish and DES, but still widely used and well-understood

5. Result and Discussion

For testing the efficiency and reliability of algorithms for storage of files we have implemented it on functional basis over a NodeJS project, we had used Crypto module to handle a major part of mathematical implementation on the encryption algorithms. As the crypto module supports only string encryption thus the supplied file irrespective of its size extension and other characteristics must be converted to string; for that we have implemented buffer conversion which is then converted to string after the complete read process, and it is then returned to the next function for processing the metadata. Apart from the encryption and decryption of file content we have also focused on overall security and privacy of stored files by encrypting the filename and other metadata of files using the SHA-512 algorithm. The supplied file is allocated a random id and its filename and metadata are encrypted and assigned to the encryption function to use for storing the encrypted file content, The above step ensures privacy and security of the stored file while going for storage in P2P mode by encrypting all metadata along with its contents.

5.1. Block Size comparison

The bar chart below compares the block size of four encryption algorithms: Blowfish, RC4, DES, and AES. Blowfish, RC4, and DES all have a block size of 64 bits, while AES has a larger block size of 128 bits. The chart displays the data in vertical bars, with the height of each bar representing the block size of the corresponding algorithm. This chart is useful for comparing the block sizes of different encryption algorithms and understanding the impact of block size on encryption security. Fig. 9 illustrates the bar chart for comparison of the block size of four encryption algorithms.

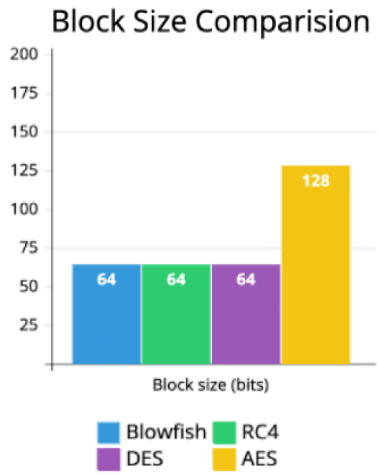

Fig. 9 Block Size comparison of four encryption algorithms

5.2. Key Size Comparison

The bar chart below compares the key size in bits of four encryption algorithms: Blowfish, RC4, DES, and AES. Blowfish has the smallest key size at 448 bits, followed by DES at 168

bits. AES has the largest key size at 256 bits, while RC4 has a relatively large key size of 2048 bits. The chart visually illustrates the differences in key sizes among these encryption algorithms, highlighting the importance of selecting an algorithm with an appropriate key size for the desired level of security. Fig. 10 illustrates the bar chart for comparison of key of four encryption algorithms.

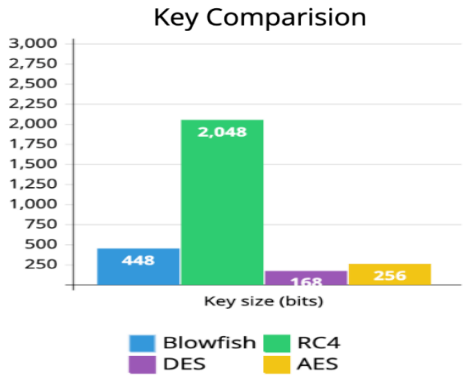

Fig. 10 Key comparison of four encryption algorithms

5.3. Performance comparison

The bar chart below compares the encryption and decryption latencies (in milliseconds) of four different algorithms - Blowfish, RC4, DES, and AES. Blowfish and DES take the least amount of time for both encryption and decryption, while AES takes the most. RC4 has moderate latency for both operations. The chart also provides information on the number of rounds used and the mode of operation for each algorithm. Blowfish and DES use 16 rounds, while AES uses 14 rounds. The mode of operation for Blowfish is 5, for RC4, it is 1, for DES, it is 4, and for AES, it is 6. Fig. 11 illustrates the bar chart for performance comparison of four encryption algorithms based on characteristics like encryption and decryption latency, rounds and mode of operation.

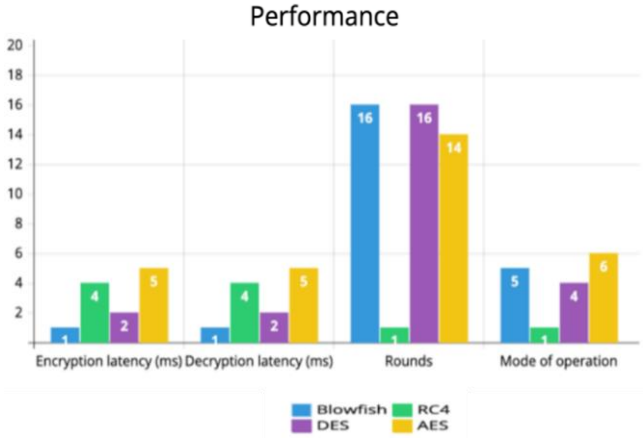

Fig. 11 Performance comparison of four encryption algorithms

5.4. Algorithm comparison

The bar chart compares the performance, security level, key schedule, and complexity of four different encryption algorithms: Blowfish, RC4, DES, and AES. The horizontal axis represents the four algorithms, while the vertical axis shows the level of each metric on a scale of 1 to 5. RC4 is the fastest algorithm with a performance level of 5 and the least complex with a complexity level of 1. AES is the most secure algorithm with a security level of 5 and the most complex with a complexity level of 4. Blowfish and RC4 have the same key schedule level of 3 and 4, respectively, while DES has the lowest key schedule level of 2. Fig. 12 illustrates the bar chart for algorithm comparison of four encryption algorithms based on characteristics like key schedule, security level, performance and algorithm complexity.

Technically, we have tried to analyze and worked on the implementation of various algorithms *(code link can be checked in reference section)* to enhance security and reduce the cost in terms of computational and financial resources. In this paper, we proposed use of AES encryption algorithm to ensure added security to the big sensitive data in transit, and use of permissioned blockchain as the network framework. There has been continuous research by many researchers throughout the world in this domain using private or public blockchain networks in recent days, but none of them have referred to, or suggested the implementation of these two approaches. Permissioned blockchain has been around for a while, but it is the least explored domain as people and organizations are more oriented towards private or public blockchain. Blockchain as a Service (BaaS), as a pay-as-u-go service, offers more functional and pocket-friendly benefits than any other blockchain network.

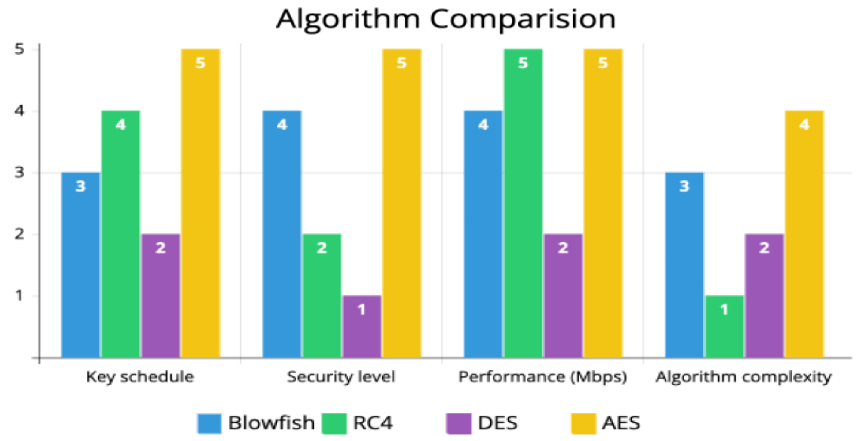

Fig. 12 Encryption algorithms' comparison for Security and Complexity parameters

6. Conclusion and Future Work

The present paper focuses on an enhanced framework for securely handling and storing data in a serverless architecture model using the Advanced Encryption Standard (AES) encryption and decryption algorithms. The developed model proposes the integration of blockchain technology within a serverless framework for secure data storage, enabling

organizations to delegate architecture-related tasks to others, including provisioning, scalability, and maintenance's encryption is employed in the proposed model due to its high level of security, fast data storage transactions, and a variable key size that can be adjusted according to the data sensitivity level. To assess the performance of the proposed AES encryption algorithm compared to other commonly used encryption algorithms, we evaluated four encryption algorithms, including Blowfish, RC4, AES, and DES. The study revealed that AES encryption provided superior security, faster data storage transactions, and better overall performance compared to the other three encryption algorithms.

In future research, the proposed system will be implemented in the real world with different big data sets, focusing on improving system performance through experimentation and analysis of results related to the proposed system. Additionally, the research will explore the use of different encryption and decryption algorithms on sensitive geospatial data to enhance system efficiency and security. Overall, the proposed framework offers a secure, decentralized, and immutable coordinate system for geospatial data, ensuring data integrity, privacy, and security within a serverless architecture model.

7. Supplementary Data

Code used in this article can be found online at blockChainDataStorage - https://github.com/yashpritwani/blockChainDataSharingAndTransfer

The following abbreviations are used in this manuscript:

RC4	Rivest Cipher 4
DES	Data Encryption Standard.
AES	Advanced Encryption Standard
IPFS	Inter Planetary File System
AWS	Amazon Web Services
MFA	Multi-factor authentication
AQMS	Air Quality Management System
SHA	Secure Hash Algorithm
ECB	Electronic Code Book
CBC	Cipher block chaining
CFB	cipher feedback encryption
OFB	output feedback
PCBC	Propagating cipher block chaining
CTR	Counter Mode
GCM	Galois/Counter Mode
ECCST	Elliptic-curve key-based cyclic shift transposition
ECC	Elliptic curve cryptography
CST	Cyclic Shift transposition
CSTA	Cyclic shift transportation algorithm
PT	Plain Text
CT	Cipher Text
BaaS	Blockchain as a Service

REFERENCES

[1] A. Jyoti and R. K. Chauhan, "A blockchain and smart contract-based data provenance collection and storing in cloud environment", *Wireless Networks*, vol. 28, no. 4, pp. 1541–1562, 2022.

[2] H. G. Do and W. K. Ng, "Blockchain-based system for secure data storage with private keyword search", In Proceedings of the 2017 IEEE World Congress on Services (SERVICES), 2017, pp. 90–93.

[3] Q. Zheng, Y. Li, P. Chen and X. Dong, "An innovative IPFS-based storage model for blockchain," In Proceedings of the 2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI), 2018, pp. 704–708.

[4] E. Karaarslan and E. Konacaklı, "Data storage in the decentralized world: Blockchain and derivatives", 2020. arXiv preprint arXiv:2012.

[5] K. Wüst and A. Gervais, "Do you need a blockchain?", In Proceedings of the 2018 Crypto Valley Conference on Blockchain Technology (CVCBT), 2018, pp. 45–54.

[6] G. Brambilla, M. Amoretti, F. Medioli and F. Zanichelli, "Blockchain-based Proof of Location", 2016. arXiv preprint arXiv:1607.00174.

[7] Y. Ren, Y. Leng, Y. Cheng and J. Wang, "Secure data storage based on blockchain and coding in edge computing", *Math. Biosci. Eng.*, vol. 16, no. 4, pp. 1874–1892, 2019.

[8] M. Alsayegh, T. Moulahi, A. Alabdulatif and P. Lorenz, "Towards secure searchable electronic health records using consortium blockchain", Network, vol. 2, no. 2, pp. 239–256, 2022.

[9] U. Bodkhe, S. Tanwar, K. Parekh, P. Khanpara, S. Tyagi, N. Kumar and M. Alazab, "Blockchain for industry 4.0: A comprehensive review", *IEEE Access*, vol. 8, pp. 79764–79800, 2020.

[10] Y. H. Park, Y. Kim, S. O. Lee and K. Ko, "Secure outsourced blockchain-based medical data sharing system using proxy re-encryption", *Applied Sciences*, vol. 11, no. 20, p. 9422, 2021.

[11] A. Chauhan, O. P. Malviya, M. Verma and T. S. Mor, "Blockchain and scalability", In Proceedings of the IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C), 2018, pp. 122–128.

[12] www.forbes.com, "3 ways blockchain is a supply chain's best friend," Available online: https://www.forbes.com/sites/oracle/2019/03/08/3ways-blockchain-is-a-supply-chains-best-friend/

[13] B. Marr, "30+ Real examples of Blockchain technology in practice," 2018. Forbes.

[14] S. Bebortta, S. K. Das, M. Kandpal, R. K. Barik and H. Dubey, "Geospatial serverless computing: Architectures, tools and future directions", *ISPRS International Journal of Geo-Information*, vol. 9, no. 5, p. 311, 2020.

[15] S. Ali, G. Wang, B. White and R. L. Cottrell, "A blockchain-based decentralized data storage and access framework for pinger", In Proceedings of the 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications/12th IEEE International Conference on Big Data Science and Engineering (TrustCom/BigDataSE), 2018, pp. 1303–1308.

[16] Q. Xu, K. M. M. Aung, Y. Zhu and K. L. Yong, "A blockchain-based storage system for data analytics in the internet of things", *New Advances in the Internet of Things*, pp. 119–138, 2018.

[17] Y. K. Peker, X. Rodriguez, J. Ericsson, S. J. Lee and A. J. Perez, "A cost analysis of internet of things sensor data storage on blockchain via smart contracts", *Electronics*, vol. 9, no. 2, p. 244, 2020.

[18] W. Liang, Y. Fan, K. C. Li, D. Zhang and J. L. Gaudiot, "Secure data storage and recovery in industrial blockchain network environments", *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6543–6552, 2020.

[19] M. Kandpal, R. K. Barik and C. Misra, "Towards Data Storage, Availability and Scalability with the Aid of Blockchain", *Trends in Wireless Communication and Information Security*, pp. 189–196, 2021.

[20] H. P. K. and S. G. Kumar, "Scalability challenges and solutions in blockchain technology," In Proceedings of the ICICIT Inventive Computation and Information Technologies, Singapore: Springer Nature Singapore, 2022 pp. 595–606.

[21] Y. Chen, B. Hu, H. Yu, Z. Duan and J. Huang, "A Threshold Proxy Re-Encryption Scheme for Secure IoT Data Sharing Based on Blockchain", *Electronics*, vol. 10, p. 2359, 2021.

[22] A. I. Sanka and R. C. Cheung, "A systematic review of blockchain scalability: Issues, solutions, analysis and future research", *Journal of Network and Computer Applications*, vol. 195, p. 103232, 2021.

[23] G. Karame, "On the security and scalability of bitcoin's blockchain", In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 1861–1862.

[24] P. W. Eklund and R. Beck, "Factors that impact blockchain scalability", In Proceedings of the 11th International Conference on Management of Digital Ecosystems, 2019, pp. 126–133.

[25] S. Ghaemi, H. Khazaei and P. Musilek, "Chainfaas: An open blockchain-based serverless platform", *IEEE Access*, vol. 8, pp. 131760–131778, 2020.

[26] H. Shafagh, L. Burkhalter, A. Hithnawi and S. Duquennoy, "Towards blockchain-based auditable storage and sharing of IoT data," 2017. In Proceedings of the 2017 on Cloud Computing Security Workshop, pp. 45–50.
[27] G. Rathee, et al., "On the design and implementation of a secure blockchain-based hybrid framework for Industrial Internet-of-Things", *Information Processing & Management*, vol. 58, no. 3, p. 102526, 2021.
[28] D. Tosh, S. Shetty, X. Liang, C. Kamhoua and L. L. Njilla, "Data provenance in the cloud: A blockchain-based approach", *IEEE Consumer Electronics Magazine*, vol. 8, no. 4, pp. 38–44, 2019.
[29] T. Liu, J. Wu, J. Li and J. Li, "Secure and balanced scheme for non-local data storage in blockchain network", In Proceedings of the IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2019, pp. 2424–2427.
[30] Md. A. Uddin, et al., "A survey on the adoption of blockchain in IoT: Challenges and solutions", *Blockchain: Research and Applications*, vol. 2, no. 2, p. 100006, 2021.
[31] A. Lakhan, M. Ali Dootio, A. H. Sodhro, S. Pirbhulal, T. M. Groenli, M. S. Khokhar and L. Wang, "Cost-efficient service selection and execution and blockchain-enabled serverless network for internet of medical things," *Mathematical Biosciences and Engineering*, vol. 18, no. 6, pp. 7344–7362, 2021.
[32] A. Kaplunovich, K. P. Joshi and Y. Yesha, "Scalability analysis of blockchain on a serverless cloud", In Proceedings of the IEEE International Conference on Big Data (Big Data), 2019, pp. 4214–4222.
[33] M. Stefanović, D. Pržulj, S. Ristić and D. Stefanović, "Blockchain and land administration: Possible applications and limitations", In Proceedings of the 5th International Scientific Conference on Contemporary Issues in Economics, Business and Management EBM, 2018, pp. 1-8.
[34] M. N. Kamel Boulos, J. T. Wilson and K. A. Clauson, "Geospatial blockchain: promises, challenges, and scenarios in health and healthcare", *International Journal of Health Geographics*, vol. 17, no. 1, pp. 1–10, 2018.
[35] P. Zhao, J. R. Cedeno Jimenez, M. A. Brovelli and A. Mansourian, J. R. Cedeno Jimenez, P. Zhao, A. Mansourian, M. A. Brovelli, "Geospatial Blockchain: review of decentralized geospatial data sharing systems", *AGILE: GIScience Series*, vol. 3, no. 29, pp 1–6, 2022.
[36] L. Junghyun, T. Kwon, and S. Jung, "An effective spatial join method for blockchain-based geospatial data using hierarchical quadrant spatial LSM+ tree", *The Journal of Supercomputing*, pp. 1–32, 2024.
[37] H. G. Do and W. K. Ng, "Blockchain-based system for secure data storage with private keyword search", In Proceedings of the 2017 IEEE World Congress on Services (SERVICES), 2017, pp. 90–93.
[38] J. Xie, F. R. Yu, T. Huang, R. Xie, J. Liu and Y. Liu, "A survey on the scalability of blockchain systems", *IEEE Network*, vol. 33, no. 5, pp. 166–173, 2019.
[39] R. Reisman, "Blockchain serverless public/private key infrastructure for ADS-B security, authentication, and privacy," In Proceedings of the AIAA Scitech 2019 Forum, 2019 p. 2203.
[40] S. Benedict, "Serverless blockchain-enabled architecture for IoT societal applications", *IEEE Transactions on Computational Social Systems*, vol. 7, no. 5, pp. 1146–1158, 2020.
[41] C. Siemon, D. Rueckel and B. Krumay, "Blockchain technology for emergency response", In Proceedings of the 53rd Hawaii International Conference on System Sciences, 2020.
[42] K. A. Alsulbi, M. A. Khemakhem, A. A. Basuhail, F. E. Eassa, K. M. Jambi and K. A. Almarhabi, "A proposed framework for secure data storage in a big data environment based on blockchain and mobile agent", *Symmetry*, vol. 13, no. 11, p. 1990, 2021.
[43] A. S. Yadav and D. S. Kushwaha, "Blockchain-based digitization of land record through trust value-based consensus algorithm", *Peer-to-Peer Networking and Applications*, vol. 14, no. 6, pp. 3540–3558, 2021.
[44] A. S. Yadav and D. S. Kushwaha, "Digitization of land record through blockchain-based consensus algorithm", *IETE Technical Review*, pp. 1–18, 2021.
[45] A. S. Yadav, N. Singh and D. S. Kushwaha, "A scalable trust-based consensus mechanism for secure and tamper-free property transaction mechanism using DLT", *International Journal of System Assurance Engineering and Management*, vol. 13, no. 2, pp. 735–751, 2022.
[46] R. Vasantha and R. Prasad, "A secured blockchain technology by using Blowfish algorithm for new broadcast proxy provisional re-encryption & its application to cloud e-mail", *International Journal of Research*, vol. 8, no. 4, pp. 2712–2719, 2019.
[47] V. Rudramalla and R. Prasad, "An efficient secured system by using Blowfish with Blockchain technology," 2019. The International Journal of Analytical and Experimental Modal Analysis, vol. 11, no. 12, pp. 2965–2975.
[48] J. R. Cedeno Jimenez, A. Folini and M. A. Brovelli, "Collaborative Validation of User-Contributed Data Using a Geospatial Blockchain Approach: the Simile Case Study," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 48, pp. 89–95, 2022.

[49] B. Schneier, "Description of a new variable-length key, 64-bit block cipher (Blowfish)", 2005. In Fast Software Encryption: Cambridge Security Workshop Cambridge, UK, December 9–11, 1993 Proceedings, Springer Berlin Heidelberg, pp. 191–204.
[50] M. Y. Alshahrani, "Implementation of a blockchain system using improved elliptic curve cryptography algorithm for the performance assessment of the students in the e-learning platform", *Applied Sciences*, vol. 12, no. 1, p. 74, 2022.
[51] F. Gao, "Data encryption algorithm for e-commerce platform based on blockchain technology", *Discrete and Continuous Dynamical Systems-S*, vol. 12, no. 4&5, pp. 1457–1470, 2019.
[52] A. M. Abdullah, "Advanced encryption standard (AES) algorithm to encrypt and decrypt data", Cryptography and Network Security, vol. 16, no. 1, p. 11, 2017.
[53] P. Princy, "A comparison of symmetric key algorithms DES, AES, Blowfish, RC4, RC6: A survey", *International Journal of Computer Science & Engineering Technology (IJCSET)*, vol. 6, no. 5, 2015.
[54] J. Thakur and N. Kumar, "DES, AES and Blowfish: Symmetric key cryptography algorithms simulation-based performance analysis", *International Journal of Emerging Technology and Advanced Engineering*, vol. 1, no. 2, pp. 6–12, 2011.
[55] R. Masram, V. Shahare, J. Abraham and R. Moona, "Analysis and comparison of symmetric key cryptographic algorithms based on various file features", *International Journal of Network Security & Its Applications*, vol. 6, no. 4, p. 43, 2014.
[56] D. Selent, "Advanced encryption standard", *Rivier Academic Journal*, vol. 6, no. 2, pp. 1–14, 2010.
[57] Y. Ren, Y. Leng, J. Qi, P. K. Sharma, J. Wang, Z. Almakhadmeh and A. Tolba, "Multiple cloud storage mechanism based on blockchain in smart homes", *Future Generation Computer Systems*, vol. 115, pp. 304–313, 2021.
[58] A. Dorri, S. S. Kanhere, R. Jurdak and P. Gauravaram, "Blockchain for IoT security and privacy: the case study of a smart home", In Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops, 2017, pp. 618–623.
[59] E. Fernandes, J. Jung and A. Prakash, "Security analysis of emerging smart home applications", In Proceedings of the IEEE Symposium on Security and Privacy (SP), 2016, pp. 636–654.
[60] Y. Ren, Y. Leng, Y. Cheng and J. Wang, "Secure data storage based on blockchain and coding in edge computing", *Math. Biosci. Eng.*, vol. 16, no. 4, pp. 1874–1892, 2019.
[61] A. S. Yadav, N. Singh and D. S. Kushwaha, "Evolution of Blockchain and consensus mechanisms & its real-world applications", *Multimedia Tools and Applications*, pp. 1–46, 2023.
[62] A. S. Yadav, S. Agrawal and D. S. Kushwaha, "Distributed Ledger Technology-based land transaction system with trusted nodes consensus mechanism", *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 8, pp. 6414–6424, 2022.
[63] A. S. Yadav, V. Charles, D. K. Pandey, S. Gupta, T. Gherman and D. S. Kushwaha, "Blockchain-based secure privacy-preserving vehicle accident and insurance registration", *Expert Systems with Applications*, p. 120651, 2023.