International Journal of Multidisciplinary Research and Growth Evaluation.

Secure social media application system using block chain and cloud computing

Magesh A [1*], **Sakthi Ganesh C** [2], **Sanjay S** [3]
[1-3] Computer Science and Engineering, Egs Pillay Engineering College Nagapattinam, Tamil Nadu, India

* Corresponding Author: **Magesh A**

Article Info

ISSN (online): 2582-7138
Volume: 04
Issue: 03
May-June 2023
Received: 27-03-2023;
Accepted: 18-04-2023
Page No: 160-167

Abstract

Information generated by humans and machines is a major concern in the security and protection of information. However, currently the tools and processes designed to manage big data cannot meet these requirements. Most large security data is provided by third parties in the middle, making them vulnerable to security threats. Blockchain technology has emerged as a solution to these problems by providing decentralization, transferability, trust, proprietary information and traceability. In this study, we propose a new big data security solution that combines sharing, encryption and access control strategies and powered by blockchain technology. Our proposed shredding algorithm takes into account the need for data subjects to be encrypted to ensure optimum security. Additionally, we provide an additional layer of data protection by storing corrupted data in a distributed format. Our solutions increase the security of large files with minimal overhead, avoiding the encryption overhead for low-value and low-cost data. We present the results of our application, which shows the low up time load and important security and privacy considerations. Overall, our solutions provide a powerful and reliable way to protect big data.

Keywords: Big data security, blockchain, fragmentation, access control, auditing

1. Introduction

The volume of data in our world is increasing rapidly. According to International Data Corporation (IDC) prediction [1], global data generation is expected to reach more than 180 zettabytes by 2025. In the big data environment, data is regularly being collected and analyzed. Companies and organizations usually use the collected data to personalize services, improve decision-making optimization, predict future trends, etc. Nowadays, data is an essential factor in the industry [2]. When big data is stored, the data is usually stored in a distributed file system or the cloud. In distributed storage, multiple nodes cooperation is needed to accomplish a specific task. Thus, attacking one or several nodes will affect the reliability of computing results. Distributed data storage dramatically increases the burden of protection on storage nodes. If data encryption storage is applied, key management will be more complicated. Accordingly, in big data schemes, it is hard to apply the traditional asymmetric, and symmetric encryption techniques directly [3]. In the case of cloud storage, due to large scale volume, significant increment, and rapid changes, encrypting big datasets directly will raise the risk of secret key management and require a considerable computation overhead [4]. Once the secret key is exposed, the whole dataset can be corrupted and stolen. Blockchain technology has attracted academics and industry due to several properties such as immutability, confidentiality, integrity, and authorization [5]. It is based on public and private key infrastructure (PKI), cryptography, peer-to-peer network, and consensus algorithms that keep the ledger's immutability.

Combining blockchain with big data brings many benefits. For instance, the automation provided by blockchain workflow will help companies become more efficient and productive [6, 7]. The two technologies are significant to the industry, and the blockchain is still new to be used in data science; therefore, more efforts are required. Any effective security system should meet the main security properties [8], which are confidentiality (preventing unauthorized access and infringement), availability (assuring access to resources by authorized users in both normal and disastrous situations), and integrity (avoiding resources to be modified without authorization). Information protection can be carried out in several ways: by 1).

Strict access control techniques (using authentication or access control policies), 2) hiding information (using encryption, steganography, and scrambling), and 3) making it understandable and available only by people who own a map or key to reverse transformation efficiently and return the original version.

Data fragmentation by means of data security is not new and has been used in numerous studies [9, 10]. Moreover, fragmentation is combined with encryption [11, 12] to enable parallel encryption of information pieces, which is more efficient than executing complete sequential encryption. More recent studies suggested using fragmentation to eliminate encryption for data protection in order to avoid significant overhead [13, 14]. However, Any data security strategy should include encryption as a cornerstone [15]. Consequently, our solution recommends encryption for data at rest, especially for high-sensitive data. The contributions of this paper can be summarized as follows:

- This paper's main contribution is to propose a new blockchain-based framework architecture for securing a big data system.
- We propose new fragmentation techniques which facilitate the encryption regarding user preferences. • We designed a lightweight metadata structure to be stored in the blockchain ledger.
- We propose a new access control mechanism based on blockchain.
- We propose a blockchain-based big data auditing method that eliminates the third-party auditing, hence improving the dependability and stability of auditing methods.
- We show that the security, efficiency, and reliability of our solution are favorable.

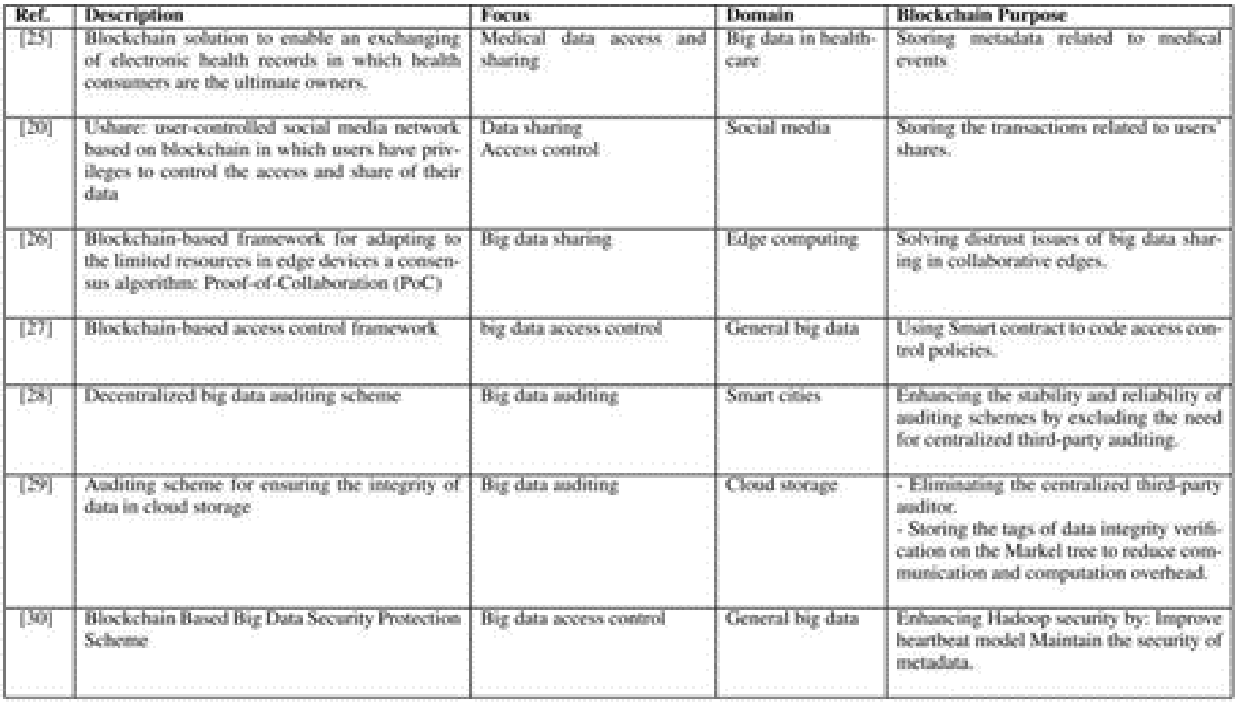
2. Related Works

Many studies discussed blockchain technology with medical data access in healthcare, which calls that people must own and access their health records. Blockchain has the potential to provide secure electronic health record (EHR) sharing in which the patients are the actual owners. In [25], authors proposed that the blockchain stores only the metadata related to medical and health events. Thus, avoiding enormously scaled blockchain infrastructure to store the entire health records.

Human activities in social media networks produce largescale databases. There is a high risk of relying on third parties to protect personal and sensitive data. Hence, there is a need to let users track and control their online activities. Blockchain can be used as a permission-filtering technique in social networks as proposed Ushare [20]. Ushare is a social media framework based on blockchain and provides user privileges to control his data access. PCA (personal certificate authority) is created for each post to manage the user's circles and encrypt data before broadcast. Blockchain is used to store the transactions related to users' shares. In spite that Ushare as a user-centric blockchain application could enable end-users to control their data, their solution is difficult to implement in terms of producing a huge number of transactions stored in the blockchain as well as encrypting the entire content of user's data.

The work in [16] aims to introduce a blockchain framework for smart mobility transactions. The purpose is to secure and protect the collected data of individuals for different participants such as companies, governments, and universities. In this scenario, each participant shares their encrypted data to a blockchain network issued with a smart contract containing transaction rules. This framework is limited to smart mobility scenarios.

Table 1: A Summary of Blockchain-based solutions for big data



Ref.	Description	Focus	Domain	Blockchain Purpose
[25]	Blockchain solution to enable an exchanging of electronic health records in which health consumers are the ultimate owners.	Medical data access and sharing	Big data in health-care	Storing metadata related to medical events
[20]	Ushare: user-controlled social media network based on blockchain in which users have privileges to control the access and share of their data	Data sharing Access control	Social media	Storing the transactions related to users' shares.
[26]	Blockchain-based framework for adapting to the limited resources in edge devices a consensus algorithm: Proof-of-Collaboration (PoC)	Big data sharing	Edge computing	Solving distrust issues of big data sharing in collaborative edges.
[27]	Blockchain-based access control framework	big data access control	General big data	Using Smart contract to code access control policies.
[28]	Decentralized big data auditing scheme	Big data auditing	Smart cities	Enhancing the stability and reliability of auditing schemes by excluding the need for centralized third-party auditing.
[29]	Auditing scheme for ensuring the integrity of data in cloud storage	Big data auditing	Cloud storage	- Eliminating the centralized third-party auditor. - Storing the tags of data integrity verification on the Markel tree to reduce communication and computation overhead.
[30]	Blockchain Based Big Data Security Protection Scheme	Big data access control	General big data	Enhancing Hadoop security by: Improve heartbeat model Maintain the security of metadata.

Z. Guan *et al*. [31] proposed a big data collection and trading system based on the Blockchain and Trusted Security Model (TSM). They combined different technologies such as Physical Unclonable Function (PUF), which uses the fingerprint as a sensor identifier while the TSM model ensures the data collection process's trust. They aim to allow trading private data and moderate any attack by using blockchain to facilitate accounting and trading processes. Trust is an essential issue in edge computing among edge devices that share big data. However, their solution lacks considering the end-to-end data protection.

In another paper by C. Xu *et al.* [26], a blockchainbased framework in collaborative edges is proposed to allow trusted big data sharing while maintaining efficient resource usage in edge devices. Moreover, this work proposed a consensus algorithm called Proof-of-Collaboration (PoC) to achieve optimized computational power. Smart environments produce a large amount of data, typically are personal and sensitive data that require great attention to secure and protect.

To avoid centralization in big data auditing, which the third party regularly provides, the study in [28] proposed a scheme for decentralized big data auditing in smart cities based on blockchain to improve stability and reliability to participate in smart city construction.

Another paper by S. Li [32] integrates big data, the energy Internet, IoT, and blockchain to construct a smart city. They exploit blockchain features that are compatible with the nature of the energy Internet. Therefore, the issue of expensive maintenance of centralized databases in big data centers is solved. A smart contract is a self-executed computer code stored in the blockchain and executed based on certain conditions evaluation. Smart contracts in blockchain technology offer a new solution to trust issues in big data by automatically executing default instructions. Even though the work offers redundant and distributed storage, it suffers from high cost, poor recovery capability, high cost of maintenance of IoT equipment.

In [33], the authors presented a big blockchain-based datasharing framework and exploited smart contracts to secure big data sharing.

Access control and authentication are considered key technologies to solve the privacy and security problems in big data. Es-Samaali *et al.* [27] use blockchain technology to develop a big data access control solution. They use smart contracts to code access control policies to check authorization for big data access requests. They adopt blockchain to enforce access policies in distributed environments where there is no central authority.

In [28], the authors proposed decentralized big data auditing scheme based on blockchain for smart cities. Their goal is to enhance auditing schemes' stability and reliability by excluding the need for centralized third-party auditing but the overhead to the users is significant. Their method produces extra costs in blockchain ledger navigation during the auditing process. Similar work has been presented in [29]; however, they designed their scheme for cloud storage. Moreover, they proposed to store the tags of data integrity verification on blockchain to reduce the overhead of communication and computation produced by the integrity verification process

Authors in [30] proposed a big data access control scheme to enhance Hadoop security by maintaining metadata security and improving the heartbeat model.

Generally, exploiting blockchain technology in the era of big data security and management is new and requires more effort. Even though previous studies employed blockchain for data sharing and access control solutions, it was restricted to specific domains and applications in big data. Furthermore, the proposed solution must also be integrated with access control, data security at rest, in transit, and auditing to improve big data security. However, this integration is limited in prior research. This study aims to address these limitations by proposing a general and comprehensive blockchain-based solution for managing and securing big data.

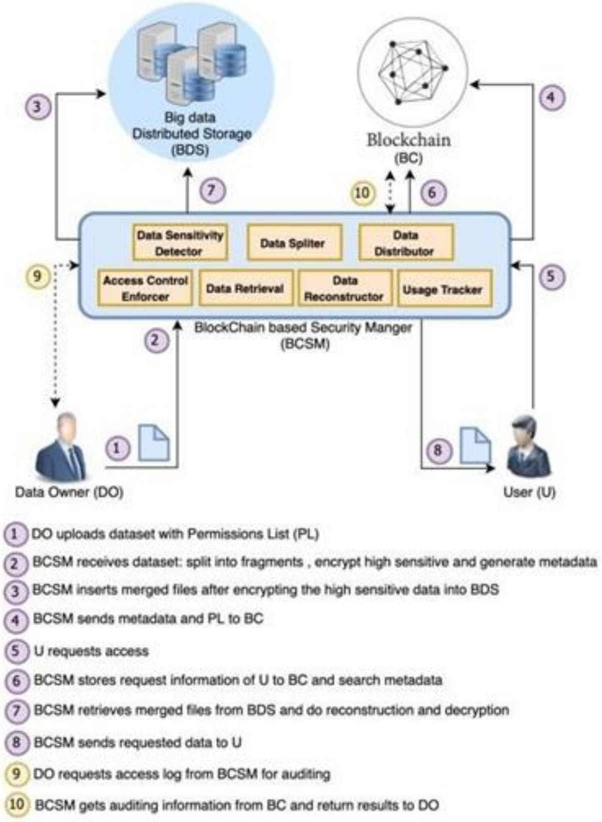
3. Proposed Framework

The proposed framework will be discussed in this section. Firstly, the system components will be introduced. Next, the workflow will be described. Finally, we will present our proposed techniques.

Framework Architecture

In our framework, data will be analyzed to define sensitive parts. According to user preferences, the sensitivity level will be determined. Depending on the sensitivity level, data will be treated differently. Data will be fragmented and encrypted before insertion into big data distributed storage. The metadata (MD) generated during the fragmentation process and the permission list (PL) will be held on the permissioned blockchain to facilitate search and tamperresistant capabilities. The structure of MD and PL are shown in Tables 2 and 3.

Figure 1 illustrates the overall architecture of the proposed framework. The roles and responsibilities of entities in the proposed framework are described as follows: • The Data Owner (DO) refers to the entity that owns the data and seeks to access or store it. DO has complete authority over his/her data. DO must create an access control policy for his/her data, including authorization for others to access his/her data. • The User (U) is the entity requesting data access with granted authorization. • The BlockChain based Security Manager (BCSM) ensures the authenticity of all events performed within the system. The events involve storing big data, storing metadata, and accessing the assets and logs on the ledger. The BCSM is also in charge of blockchain management. BCSM and other entities will communicate through a secure SSL/TLS channel. • Big data Distributed Storage (BDS) After fragmentation and encryption, BDS is in charge of storing the big data. • The blockchain (BC) is in charge of storing MD and the PL in the ledger. Moreover, BS is responsible for recording the access log and the other events of the system.



1 DO uploads dataset with Permissions List (PL)
2 BCSM receives dataset: split into fragments , encrypt high sensitive and generate metadata
3 BCSM inserts merged files after encrypting the high sensitive data into BDS
4 BCSM sends metadata and PL to BC
5 U requests access
6 BCSM stores request information of U to BC and search metadata
7 BCSM retrieves merged files from BDS and do reconstruction and decryption
8 BCSM sends requested data to U
9 DO requests access log from BCSM for auditing
10 BCSM gets auditing information from BC and return results to DO

Fig 1: The proposed framework architecture.

The proposed framework consists of the following key components:

1) Data Sensitivity Detector (DSD)

Sensitivity detection techniques can be categorized as automated, semi-automated, and manual. Due to the large volume of data, manual data sensitivity detection needs intensive efforts. Therefore, automated solutions are recommended. These solutions involve complex techniques that are out of this paper's scope and remain as future work. Examples include domain experts and neural networks such as in [34–36]. In our case, sensitivity detection is subjective to the data owner (DO) policy and requirements. The DO determines the sensitivity level of data (high, low, and not) and identifies the sensitive attributes to be protected. The flow of DSD is shown in Figure 2.

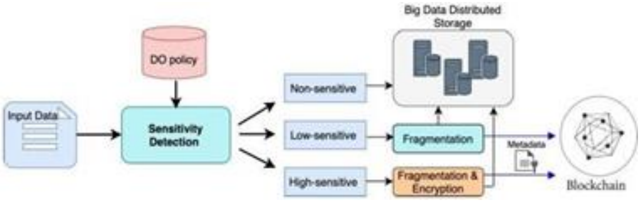


Fig 2: The flow of data sensitivity detector

2) Data Splitter (DS)

Our solution exploits the fragmentation techniques for providing an additional layer of securing sensitive data. Data is split up into sensitive and non-sensitive collections based on user requirements. The checksum is used to ensure the data integrity by calculating the SHA-512 [37] for the original file then comparing the result of hashing to the result of the file after the reconstruction process. Our framework handles the security of sensitive data according to the sensitivity level. For low-sensitive data, we use scrambling to harden the fragmentation process, which is combined with a distributed big data storage partitioning [14]. Additionally, our solution encrypts only the high-sensitive portion of the dataset to avoid the significant overhead associated with encrypting the whole volume of data. Further details of the proposed fragmentation techniques are presented in subsection B.

3) Data Distributor (DD)

This component assigns the data-id and maps it to the merged files. Then DD creates and encrypts the MD. The size of the MD data structure depends on the number of fragments. Furthermore, this component sends the merged files to BDS. Also, it sends MD and PL to be stored on the blockchain ledger.

4) Data Retrieval (DR)

This component retrieves the information (metadata and data-hash) related to the data from the blockchain using the data-id, and the requested merged files are retrieved from the BDS. The DR then performs the decryption process of metadata in order to send them to the Data Reconstructor.

5) Data Reconstructor (DRE)

This component returns the data to its original form using the metadata retrieved from the blockchain. It performs the decryption and defragmentation techniques to reconstruct the original file.

6) Access Control Enforcer (ACE)

This component is responsible for the authentication and authorization of the data owner and user. The client applies for authentication to the ACE, and once authentication is granted, the authorization process is initiated. ACE checks the user authenticity by using multi-factor authentication. The ACL rules are used to enforce that the data is only accessed with privileges listed in PL. The requested data can only be accessible by a group of authorized users based on the PL. If the user is granted, the ACE records the request in the blockchain for auditing purposes.

7) Usage Tracker (UT)

This component retrieves auditing information from blockchain regarding data access and usage upon data owner or auditor request leveraging the traceability feature provided by the blockchain.

5. Implementation

Our fragmentation techniques aim to increase the security for two types of sensitive data, namely low or highsensitive data. Based on the user's preference for handling his/her data, the degree of sensitivity is determined. This section presents our proposed fragmentation techniques which are fragmentation, defragmentation, fragmentation & encryption, and defragmentation & decryption. The first two techniques are applied for low-sensitive data and the other for high- sensitive data.

-

1) Fragmentation

Fragmentation tolerates efficient scrambling and encryption mechanisms utilizing parallelism. It is indispensable to consider the size of fragments; a too-large fragment may include too much information to reveal, whereas a too small fragment may cause agonizing overhead. To achieve optimal fragmentation, fragmentation must abide by the following rules:

- Completeness: Data should not be lost during the fragmentation process. Every data item must exist in at least one fragment.
- Reconstruction: If the data breaks down, the data can still be combined without any modification to the data structure. For all n fragments: f1 U f2 U . . . fn = Originalfile.

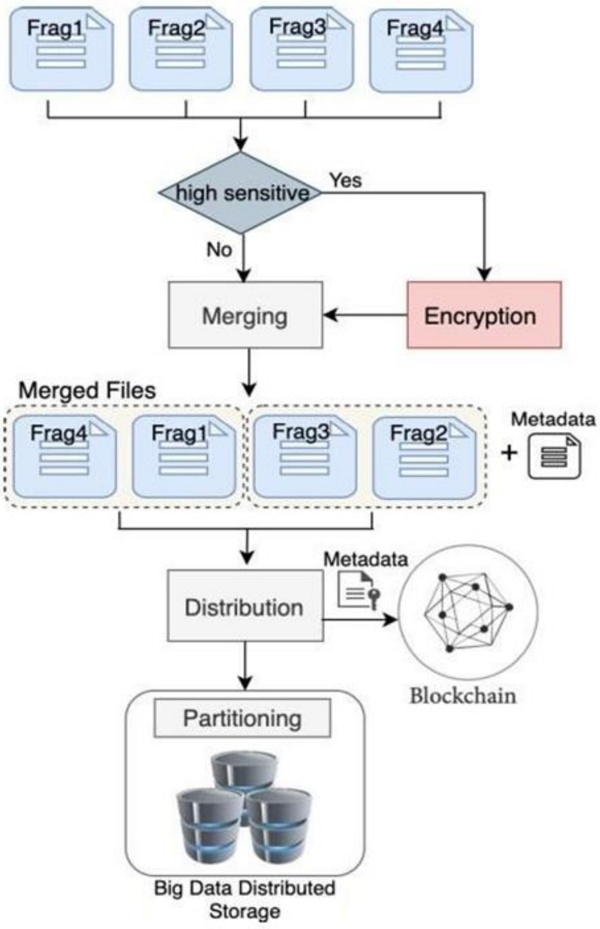


Fig 3: Data Upload process.

- Disjointness: Data in the fragment should not be included in the other fragments to prevent data duplication.

For each two fragments: fi and fj, where i 6= j, fi ∩ fj = φ. The checksum is applied to ensure meeting the above rules by calculating the SHA-512 for the original file then comparing the result of hashing to the result of the file after the reconstruction process. Our framework handles the security of sensitive data according to the sensitivity level. For low-sensitive data, we use scrambling to harden the fragmentation process, which is combined with a distributed big data storage partitioning [14].

Our technique encrypts only the high-sensitive part of a dataset to avoid the high overhead produced by encrypting large-scale data. In any scenario, fragmentation for security

raises latency inside the system. As a result, to achieve acceptable overall performance, a good fragmentation technique must be combined with processing parallelization. The fragmentation stores metadata of the dataset and its fragments. We propose a new data structure of metadata to be stored in blockchain ledger utilizing blockchain immutability and tamper resistance in order to assist the data integrity checks. The data-hash is calculated via the hashing algorithm (SHA-512) as: message digest (md) = H(m).
The md is calculated and added to the MD.
The pseudo-code for the fragmentation is illustrated in Algorithm 1. This algorithm has two major procedures. Firstly, splitting the sensitive data file into fragments based on predefined size suitable for the original file size. The original order of the fragments will be kept in the MappingArray by storing the ID of each fragment.

Creating the fragments will be based on the following:

- ∀ fi, fj ∈ F, ∃ MappingArray[i] A MappingArray[j]
- ∀fi 6= fj, MappingArray[i] 6= MappingArray[j]

Algorithm
Input Senstv Data File, fragment Size, File Path, Is High Sensitive
Output M:{ m1, m2, m3, . . . mn},
MappingArray [ ]

```
Procedure Fragmentation
  Senstv Data File ← File Input Stream (File Path)
  F ← Senstv Data File. Split File (fragment Size)
  For i ← 0, fragments Number do
    Mapping Array[i] =fi .id
  if Is High Sensitive then
    for i ← 0, fragments Number/2 do
      mi = Senstv Data File. merg(en(fj), en(fk ))
      F fj, fk are selected arbitrarily
    end for
    merge File Number = i
  else
    for i ← 0, fragments Number/2 do
      mi = Senstv Data File. merg(fj, fk )
      F fj, fk are selected arbitrarily
    end for
    merge File Number = i
  end if
procedure Store
  for i ← 0, merge File Number do
    insert HDFS (mi)
  end for
  send Mapping Array [] to block chain
```

Then, the algorithm creates the merged files which consists of two fragments fj and fk as follows:

- fj and fk are selected arbitrarily.
- ∀ mi 3 M, ∃ fj ∪ fk = mi and fj ∩ fk = φ,

where M is the merged files set and mi, mj are any two elements in the M set.
The high-sensitive fragments will use AES encryption [38].
Finally, each merged file m will be inserted in HDFS.

2) Defragmentation

Defragmentation is an inverse process of fragmentation. It follows the same rules of fragmentation. Algorithm 2 describes our pseudo-code of defragmentation. The algorithm starts by getting the Mapping Array embedded in MD from block chain. Next step is searching for the fID in M to reconstruct the original file follows the original order found in Mapping Array. For high-sensitive fragments, decryption is needed to form the original version.

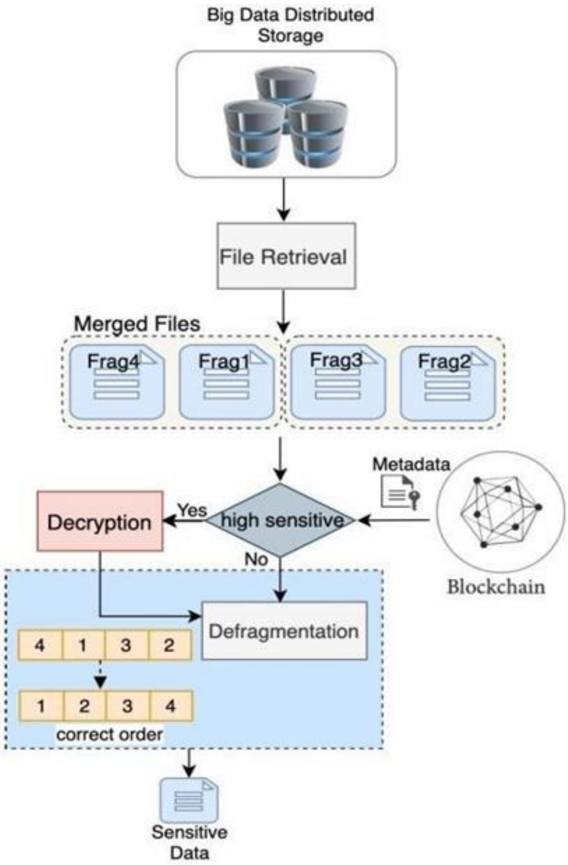


Fig 4: Data retrieve process

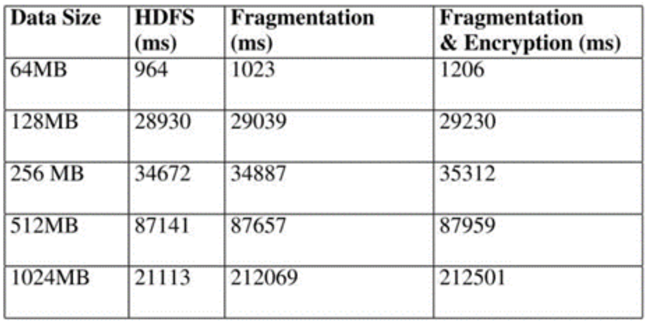
Algorithm
Input M: {m1, m2, m3,... mn}, Is High Sensitive,
Mapping Array [ ] Output Sensitive Data File

```
procedure Defragmentation
  get Mapping Array[ ] from block chain
  Outputstream = File Output Stream (Sensitive Data File)
  for i ← fragmentsNumber, 0 do
    fID = MappingArray[i]
    Search fID in M if match, read and send
    to Output stream
    f = File Input Stream (fID)
    f .read ()
    if Is High Sensitive then
      outputstrem. Write (decrypt (f ))
    else
      outputstrem. Write (f)
  end for
  outputstream. Close ()
```

5. Result Analysis

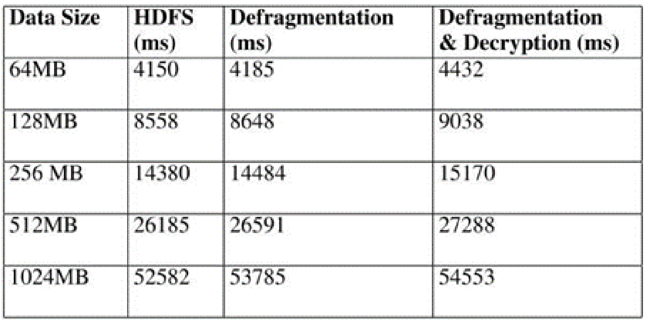
The experiments were conducted to evaluate the performance overhead caused by introducing our framework into HDFS. We measured the writing and reading speeds of the HDFS with and without the proposed techniques.We have developed a prototype of our framework in Java that is able to read and write to HDFS. We considered different structured files sizes to evaluate the performance of our framework. We have simulated the parallelism of fragmentation by implementing multithreading. Each experiment was repeated five times, and we took the average as the measured result.

Table 2: Data write experiment results



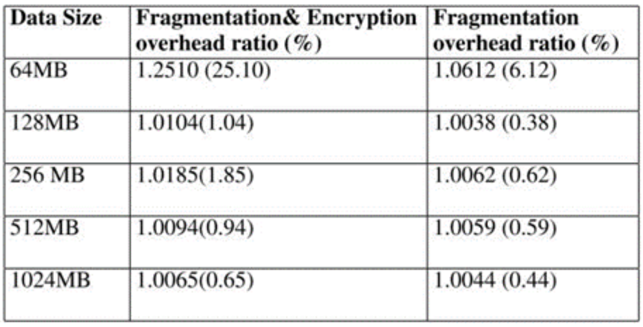
Data Size	HDFS (ms)	Fragmentation (ms)	Fragmentation & Encryption (ms)
64MB	964	1023	1206
128MB	28930	29039	29230
256 MB	34672	34887	35312
512MB	87141	87657	87959
1024MB	21113	212069	212501

Table 3: Data read experiment results



Data Size	HDFS (ms)	Defragmentation (ms)	Defragmentation & Decryption (ms)
64MB	4150	4185	4432
128MB	8558	8648	9038
256 MB	14380	14484	15170
512MB	26185	26591	27288
1024MB	52582	53785	54553

Table 4: Overhead ratio in writing experiments



Data Size	Fragmentation& Encryption overhead ratio (%)	Fragmentation overhead ratio (%)
64MB	1.2510 (25.10)	1.0612 (6.12)
128MB	1.0104(1.04)	1.0038 (0.38)
256 MB	1.0185(1.85)	1.0062 (0.62)
512MB	1.0094(0.94)	1.0059 (0.59)
1024MB	1.0065(0.65)	1.0044 (0.44)

6. Conclusions

The security and privacy issues of big data systems are noteworthy and require demand attention. For instance, big data models such as Hadoop are built without any secure assumption. Furthermore, most of the existing tools rely on third parties, which imposes significant security issues. In this work, we proposed a big data security framework that provides data security using blockchain technology and fragmentation. The framework offers a secure environment for big data sharing, storage, and transmitting. Blockchain is responsible for providing the necessary security for big data storing and retrieving processes as well as access control and auditing mechanisms. Prior research has not adequately addressed big data security issues. For instance, previous studies have primarily focused on access control, data sharing, and auditing in specific big data domains such as smart homes and healthcare. However, our proposed framework is a generic solution that can be utilized in a wide range of big data domains. This study is still progressing, and the authors believe that additional details should be explored and published in a future paper. For future work, we are going to implement the complete scenario of our framework with blockchain technology. The Hyperledger fabric [40] platform will be used to implement the solution, which is a permissioned blockchain with higher transaction throughput and security than other blockchain platforms [41, 42].

7. References

1. M Zwolenski, L Weatherill. The digital universe: Rich data and the increasing value of the Internet of Things, J Telecommun. Digit. Economy. 2014; 2(3):1-47.
2. X Jin, BW Wah, X Cheng, Y Wang. Significance and challenges of big data research, Big Data Res. 2015; 2(2):59-64.
3. D Lv, S Zhu, H Xu, R Liu. A review of big data security and privacy protection technology, in Proc. IEEE 18th Int. Conf. Commun. Technol. (ICCT). 2018, 1082-1091.
4. López-Alt, E Tromer, V Vaikuntanathan. On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption in Proc. 44th Symp. Theory Comput, 2012, 1219-1234.
5. Q Zhou, H Huang, Z Zheng, J Bian. Solutions to scalability of blockchain: A survey, IEEE Access. 2020; 8:16440-16455.
6. R Norvill, M Steichen, WM Shbair, R State. Demo: Blockchain for the simplification and automation of KYC result sharing, in Proc. IEEE Int. Conf. Blockchain Cryptocurrency (ICBC), 2019, 9-10.
7. M Kassen. Blockchain and E-government innovation: Automation of public information processes, Inf. Syst., 2022, 103:101862.
8. R von Solms, J van Niekerk. From information security to cyber security,'' Comput. Secur. 2013; 38:97-102.
9. G Aggarwal, M Bawa, P Ganesan, H Garcia-Molina, K Kenthapadi, R Motwani, U Srivastava, D Thomas, Y Xu. Two can keep a secret: A distributed architecture for secure database services, in Proc. CIDR, 2005, 1-5.
10. N Santos, GL Masala. Big data security on cloud servers using data fragmentation technique and NoSQL database, in Proc. Int. Conf. Intell. Interact. Multimedia Syst. Services. Cham, Switzerland: Springer, 2018, 5-13.
11. G Memmi, K Kapusta, H Qiu. Data protection: Combining fragmentation, encryption, and dispersion, in Proc. Int. Conf. Cyber Secur. Smart Cities, Ind. Control Syst. Commun. (SSIC), 2015, 1-9.
12. H Heni, MB Abdallah, F Gargouri. Combining fragmentation and encryption to ensure big data at rest security, in Proc. Int. Conf. Hybrid Intell. Syst. Cham, Switzerland: Springer, 2017, 177-185.
13. K Kapusta, G Memmi. A fast fragmentation algorithm for data protection in a multi-cloud environment, 2018, ar Xiv: 1804.01886.
14. DE Bakken, R Rarameswaran, DM Blough, AA Franz, TJ Palmer. Data obfuscation: Anonymity and desensitization of usable data sets,'' IEEE Security Privacy. 2004; 2(6):34-41.
15. C Tankard. Encryption as the cornerstone of big data security, Netw. Secur. 2017; 3:5-7.
16. D Lopez, B Farooq. A blockchain framework for smart mobility, in Proc. IEEE Int. Smart Cities Conf. (ISC2), Sep. 2018, 1-7.
17. N Deepa, Q Pham, DC Nguyen, S Bhattacharya, PBTR Gadekallu, PKR Maddikunta, F Fang, PN Pathirana. A

survey on block chain for big data: Approaches, opportunities, and future directions,'' CoRR, vol. abs/2009.00858, 2020, 1-4.

18. J Anuradha, A brief introduction on big data 5VS characteristics and Hadoop technology, Proc. Comput. Sci. 2015; 48:319-324.
19. M Crosby, P Pattanayak, S Verma, V Kalyanaraman, Blockchain technology: Beyond bitcoin, Appl. Innov. 2016; 2(6-10):71.
20. Chakravorty, C Rong. Ushare: User controlled social media based on blockchain, in Proc. 11th Int. Conf. Ubiquitous Inf. Manage. Commun, Jan. 2017, 1-6.
21. LS Sankar, M Sindhu, M Sethumadhavan. Survey of consensus protocols on blockchain applications, in Proc. 4th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS), Jan. 2017, 1-5.
22. M Castro, B Liskov. A correctness proof for a practical byzantinefault-tolerant replication algorithm, MIT Lab. Comput. Sci., Cambridge, MA, USA, Tech. Rep. MIT/LCS/TM-590, 1999.
23. PL Aublin, SB Mokhtar, V Quema. RBFT: Redundant byzantine fault tolerance,'' in Proc. IEEE 33rd Int. Conf. Distrib. Comput. Syst., 2013, 297-306.
24. Y Xiao, N Zhang, W Lou, YT Hou. A survey of distributed consensus protocols for blockchain networks,'' IEEE Commun. Surveys Tuts. 2020; 22(2):1432-1465.
25. N Gupta, A Jha, P Roy. Adopting blockchain technology for electronic health record interoperability, Cognizant Technol. Solutions, Teaneck, NJ, USA, White Paper, 2016. [Online]. Available: https:// dokumen.tips/documents/adopting-blockchain-technology-for- electronichealth-record-.html
26. C Xu, K Wang, G Xu, P Li, S Guo, J Luo. Making big data open in collaborative edges: A blockchain-based framework with reduced resource requirements, in Proc. IEEE Int. Conf. Commun. (ICC), 2018, 1-6.
27. H Es-Samaali, A Outchakoucht, JP Leroy. A blockchain-based access control for big data, Int. J. Comput. Netw. Commun. Secur. 2017; 5(7):137.
28. H Yu, Z Yang, RO Sinnott. Decentralized big data auditing for smart city environments leveraging blockchain technology, IEEE Access. 2019; 7:6288-6296.
29. J Li, J Wu, G Jiang, T Srikanthan. Blockchain-based public auditing for big data in cloud storage, Inf. Process. Manage. 2020; 57(6):102382.
30. C Zhang, Y Li, W Sun, S Guan. Blockchain based big data security protection scheme, in Proc. IEEE 5th Inf. Technol. Mechatronics Eng. Conf. (ITOEC), 2020, 574-578.
31. Z Guan, Y Zhao, D Li, J Liu. TBDCT: A framework of trusted big data collection and trade system based on blockchain and TSM, in Proc. IEEE Int. Conf. Smart Cloud (SmartCloud), 2018, 77-83.
32. S Li. Application of blockchain technology in smart city infrastructure, in Proc. IEEE Int. Conf. Smart Internet Things (SmartIoT), 2018, 276.
33. L Yue, H Junqin, Q Shengzhi, WRuijin. Big data model of security sharing based on blockchain, in Proc. 3rd Int. Conf. Big Data Comput. Commun. (BIGCOM), 2017, 117-121.
34. BK Adhikari, W Zuo, R Maharjan, X Han, S Liang, Detection of sensitive data to counter global terrorism, Appl. Sci. 2019; 10(1):182.
35. G Xu, C Qi, H Yu, S Xu, C Zhao, J Yuan. Detecting sensitive information of unstructured text using convolutional neural network, in Proc. Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discovery (CyberC), Oct. 2019, 474-479.
36. Z Yang, Z Liang. Automated identification of sensitive data from implicit user specification, Cybersecurity, 2018; 1(1):1-15.
37. CIS Ubuntu Linux 18.04 BenchmarksT Resources. Accessed: Oct. 30, 2021. [Online]. Available: https://www.cisecurity.org/cisbenchmarks