BCSM: A BlockChain-based Security Manager for Big Data

Hanan E. Alhazmi[1], Fathy E. Eassa[2]
Department of Computer Science, Faculty of Computing and Information Technology
King Abdulaziz University (KAU), Jeddah, Saudi Arabia[1,2]
Computer Science Department, Umm Al-Qura University, Makkah 21955, Saudi Arabia[1]

***Abstract*—The amount of data generated globally is increasing rapidly. This growth in big data poses security and privacy issues. Organizations that collect data from numerous sources could face legal or business consequences resulting from a security breach and the exposure of sensitive information. The traditional tools used for decades to handle, manage, and secure data are not suitable anymore in the case of big data. Furthermore, most of the current security tools rely on third-party services, which have numerous security problems. More research must investigate protecting user-sensitive information which can be abused and altered from several sides. Blockchain is a promising technology that provides decentralized backend infrastructure. Blockchain keeps track of transactions indefinitely and protects them from alteration. It provides a secure, tamper-proof database that may be used to track the past state of the system. In this paper, we present our big data security manager based on Hyperledger Fabric, which provides end-to-end big data security, including data storage, transmitting, and sharing as well as access control and auditing mechanisms. The manager components and modular architecture are illustrated. The metadata and permissions related to stored datasets are stored in the blockchain to be protected. Finally, we have tested the performance of our solution in terms of transaction throughput and average latency. The performance metrics are provided by Hyperledger Caliper, a benchmark tool for analyzing Hyperledger blockchain performance.**

Keywords*—*Big data security; blockchain; access control; hyperledger fabric

I. Introduction

Since 2011, five Exabytes (10^{18}) of data have been generated every two days. Nowadays, this is done in less than ten minutes [1]. Social media data, videos, server logs, and sensor data are among the many types of data that have been generated. Compared to a traditional relational database management system, big data technologies are more equipped to deal with large volumes and diverse types of data. Large amounts of information can be gathered from various big data applications. For instance, these massive amounts of data always contain sensitive information that might disclose a person's identity. Although all of the information required to identify a person may not be present in the same dataset, a combination of data sources may be able to reveal their identity. Because of this, these sensitive data must be protected. When it comes to storing large amounts of data, distributed storage like Hadoop Distributed File System (HDFS) [2] is commonly used. Multiple nodes must cooperate to complete a single task in distributed storage. Consequently, the reliability of computing results will be affected if an attack targets one or more nodes. Distributed data storage significantly raises the storage node's obligation to protect the data. Key management becomes more difficult in the case of encrypted data storage. As a result, the traditional symmetric and asymmetric encryption techniques cannot be directly applied in big data schemes [3]. In the existing Hadoop implementation [4], the Portable Operating System Interface (POSIX) architecture is used to enable access to folders and files stored in HDFS where users may or may not be granted access to a whole dataset. However, this does not prevent authorized users from misusing or abusing the data. It also provides system security auditing [5]; however, there is no standard format for this auditing, making it difficult to read and analyze. Our previous work [6] presented the way for implementing a security framework in Hadoop. We proposed integrating blockchain technology with new fragmentation and encryption techniques to increase big data security. We have tested the performance of our techniques which imposed negligible computation overhead in contrast to the security and privacy improvements. Once the data are fragmented and stored, the next step is to test the performance when integrated with blockchain. This paper presents a new security solution for big data, called BCSM, that leverages the unique security by design and tamper-proof properties of blockchain technology in contemporary domains[7], [8]. Data is stored in HDFS, and the related metadata and permissions will be held as assets inside the blockchain. We used Hyperledger Fabric [9], a permissioned blockchain with a distributed ledger that allows smart contracts [10]. Unlike other public blockchains like Ethereum, Bitcoin, or Monero, the data in Fabric can only be accessed by those who have been authorized. Paper contributions are summarized below: 1) proposing a new architecture of integrating big data (Hadoop) with blockchain (Hyperledger fabric); 2) enforcing access control policies based on data permissions; 3) protecting metadata and permissions to be stored and accessed by blockchain. 4) evaluating the performance of the proposed solution in terms of throughput and latency for reading and writing operations. The remainder of the paper is structured as follows. Section 2 is a compilation of related work. The proposed BCSM manager is presented in depth in Section 3. Section 4 presents the findings of the BCSM manager's testing and evaluation. The conclusion is addressed in Section 5.

II. State of the Art

A. Blockchain Technology

Blockchain is a decentralized system for exchanging digital currencies that were first introduced by bitcoin [11]. Blockchain is managed by a peer-to-peer network. It is a distributed ledger that records and stores transactions in blocks that are linked using cryptography. An untrusted party submits a transaction block, which is then confirmed by the other participants in the chain of transactions without any central authority. The chain expands indefinitely from the first block, the genesis block, as each subsequent block ties to the previous one via its hash value. In other words, the hash value of the preceding block is considered when calculating the hash value of a new block. As a result, any attempt to alter the hashes of connected blocks will cause the shared ledger to be tampered with, making the blockchain tamper-resistant. All members of the blockchain will have access to a shared ledger. Each peer will have access to the latest version of the blockchain after being updated to its unique state.

Tractability is one of the important features of the blockchain. Transactions on the blockchain are tagged with a timestamp once they've been validated. Thus, allowing users to track the history of all transactions to facilitate auditing, which is essential in data management and applications that need access to a tamper-proof log history.

1) Consensus Algorithms: Consensus algorithms are used to obtain consensus on the new state of the blockchain. A consensus algorithm uses a group of participants who are directly participating in the system to make agreements instead of using third-party decision-making. Practical Byzantine Fault Tolerance (PBFT), PBOT, Proof of work (PoW), and proof of stake (PoS) are some of the most well-known examples of consensus algorithms on the blockchain. They differ in identity management mechanisms, adversary power, and energy savings [12].

2) Smart Contract: The smart contract is stored on a blockchain which is a piece of code executed when some conditions are fulfilled. Smart contracts are often used to automate the execution of business logic. All participants are instantly receiving the outcome without the engagement of an intermediary or the loss of time. On the other hand, smart contracts eliminate the need for a centralized authority. Aside from just exchanging digital currency, smart contracts can also be used to build applications in the supply chain, business process management (BPM), and healthcare, all of which are areas where blockchain technology has the potential to have a significant influence.

3) Blockchain Types: Bitcoin was the first public blockchain. That is, anyone with an anonymous identity can join and read the blockchain, submit transactions, and participate in the consensus process. Although public blockchains have the advantage of being accessible to anyone with unknown identities, the rise of private blockchains is more suited from an intra-organizational viewpoint to incorporating blockchain into several products. Users who want to join the private or permissioned blockchain must be authenticated by an additional permission layer. As a result, the main distinction between public and private blockchain is participating in the system. Furthermore, there is a third form of blockchain known as a consortium blockchain, which can be considered a hybrid because only certain nodes can participate in consensus, and access to read or write on the blockchain [13]. Fabric, Sawtooth, Burrow, and Iroha are examples of open-source industrial blockchain frameworks under the Hyperledger projects hosted by Linux. In order to build permissioned blockchain platforms, Hyperledger Fabric provides a modular design and contains a Membership component. It contains the "chaincode" which is used to implement the application logic, and transaction functionalities in several programming languages. The Fabric uses an execute-order-validate approach instead of an order-execute [14] to solve the drawbacks of permissioned blockchains, like the non-deterministic execution of concurrent transactions, inflexible trust model, execution on all nodes, and hard-coded consensus. The Transaction Log and the World State are parts of Fabric's ledger. All transactions are recorded in a transaction log. By utilizing the world state, a program may obtain the current value of a state without searching through the entire transaction log. Key-value pairs are the default representation for ledger states. When a transaction updates any value that was previously entered in the ledger or adds new data, this is a new state for the blockchain that will be preserved in everlasting; it is impossible to return the last state of the blockchain [13].

B. Big Data Security and Privacy Issues

1) Access Control: Access Control is a critical aspect of big data. Organizations and users working with Big Data must implement access control policies. An access control mechanism governs the connectivity of various nodes to the system. A weak access control method can enable attackers to get unauthorized access to data storage, bringing security and privacy concerns [15]. Access control lists (ACL) and policies help protect data by granting nodes and devices privacy and security permissions. Although numerous research studies have focused on access control mechanisms, specific difficulties still need to be solved. For example in cloud, data owners who outsource their data to the cloud may selectively seek to make it visible/accessible to other users. Such a feature necessitates access control in order to enforce the authorizations provided by the data owners properly [16], [17]. For instance, these authorizations cannot be implemented in a cloud either by the data owners or cloud service providers (CSPs). However, making the outsourced data self-enforce the access permissions is a promising solution to this problem. The automation provided by smart contract will make this feasible [18].

2) Data Integrity: Data integrity is another consideration in maintaining big data privacy and security. Data integrity entails ensuring the consistency and accuracy of data. In the big data era, data must ensure its integrity properties from origination to final destination in analysis reports to provide valuable outcomes for business and decision-making.

3) Metadata and Policy Protection: Metadata is a type of data that describes other data with information that makes it easier to find, use and manage. Policies are sets of rules that are used to regulate data access. One of the attack methods is when attackers access or alter metadata and policies to compromise or get unauthorized access to data. Most of the time, data owners are unaware of whether these metadata or policies are accessed and changed by attackers. As few prior research discuss metadata and policy protection, thus there is a critical need to focus on this issue.

4) Data Privacy: Data privacy guarantees that only those with authorization may access the data. Big data may contain person sensitive data. Thus these data must not been revealed without the person permission. Obtaining approval from a person is also restricted to specific causes. Consequently, protecting people sensitive attributes such as social security numbers and addresses is necessary to guarantee data privacy.

5) Data Auditing: Data integrity is not always possible to be guaranteed. Data loss due to malicious activity or system failure poses a significant security risk. For instance, numerous cloud-based big data auditing approaches have been proposed to maintain the integrity of big data stored in cloud storage[19], [20]. A third-party auditor (TPA) is commonly used in these approaches to perform auditing tasks on behalf of data owners. Although TPA is considered a trustworthy entity that always acts honorably, it may not be as trustworthy as it appears. Cloud service providers (CSPs) may even hire a TPA to assist them in concealing data corruption incidents. Furthermore, because of centralization, the single point of failure might have disastrous impacts. TPA system disruptions can be caused by external attacks and internal abuses flaws. Decentralized schemes are more reliable and robust than TPA-based ones.

Our research proposes a decentralized big data security manager based on blockchain technology to address all the above issues.

C. Related Work

Several research studies have explored the use of blockchain technology in healthcare to allow patients to own and control their medical information. Blockchain technology has the potential to enable secure electronic health record (EHR) sharing in which patients are the real owners. The authors of [21] suggested that the blockchain simply stores metadata relevant to medical events to avoid overwhelming blockchain limited storage due to storing the entire health records.

In the work of [22] authors presented a privacy-preserving framework for EHR by using blockchain technology with a zero-knowledge proof cryptographic protocol named Identity Mixer. Their solution aims to protect private data and maintain anonymity.

L. Yue et al. [23] introduced a blockchain-based big data-sharing architecture and used smart contracts to facilitate big data sharing. The Access control mechanism is used for addressing big data's privacy and security issues.

A supply chain is a network that transfers products from suppliers to customers, generating a huge amount of data in the process. Authors in [24] proposed integration of blockchain-based supply chain management with big data technology. However, their contribution is limited to protecting the consumers from the risk of food fraud, and they used big data to enhance the analysis process for business profits.

I. Makhdoom et al. [25] suggested a ”privacy sharing” on a blockchain system for the secure and private preservation of IoT data in smart city environments. Data privacy is managed by blockchain, and some limited users have access to the blockchain data, which is encrypted and governed by an embedded access control mechanism.

To overcome blockchain synchronization time and storage space limitations, authors in [26] proposed a blockchain-based personnel management system that provides a new on-chain and off-chain data storage model to address the problem of insufficient storage space. However, they used a central database for out-of-chain storage, which has a risk of a single point of failure.

To improve Hadoop security, authors in [27] presented a big data access control approach that maintains metadata security by enhancing the heartbeat model.

Adopting blockchain technology by means of big data security and management necessitates more efforts. Previous research exploited blockchain for limited big data applications, for example, data sharing and access control. Furthermore, there is a lack in the state of the art to provide end-to-end big data security solutions based on blockchain technology which integrates data security at rest while transmitting and offering auditing and access control mechanisms. This research intends to solve the above constraints by presenting a comprehensive and general blockchain-based security manager for big data.

III. Proposed Security Manager

In this section, the proposed security manager for big data is presented. First, we describe the architecture of the proposed solution, then the details of the components and processes of our manager are provided.

A. Manager Architecture

As shown in Fig. 1, the manager elements consist of the following:

- **Data Owner (DO)** is the entity that owns the data and wants to access or store it. DO has full control over his data. DO needs to define policy for his data access, including data access permissions for others.
- **User (U)** is the entity that grants access to request data.
- **BlockChain-based Security Manager (BCSM)** ensures the legitimacy of the system events. The events involve storing big data and metadata and accessing the ledger's assets and logs. In addition, the BCSM is responsible for managing blockchain. BCSM will communicate with other entities via a secured SSL/TLS connection.
- **Big data Distributed Storage (BDS)** BDS is in charge of storing big data after fragmentation and encryption.
- **The BlockChain (BC)** is responsible for recording security events on the blockchain ledger. It includes the following:
    - Smart Contract: represents the following logic:
      1) creating MD and PL and inserting them as assets into ledger DB
      2) managing and accessing these assets according to user or data owner actions.
      3) managing the ACL rules used for the authorization process. The smart contract is used to interact with the ledger to read or modify the assets (MD and PL).

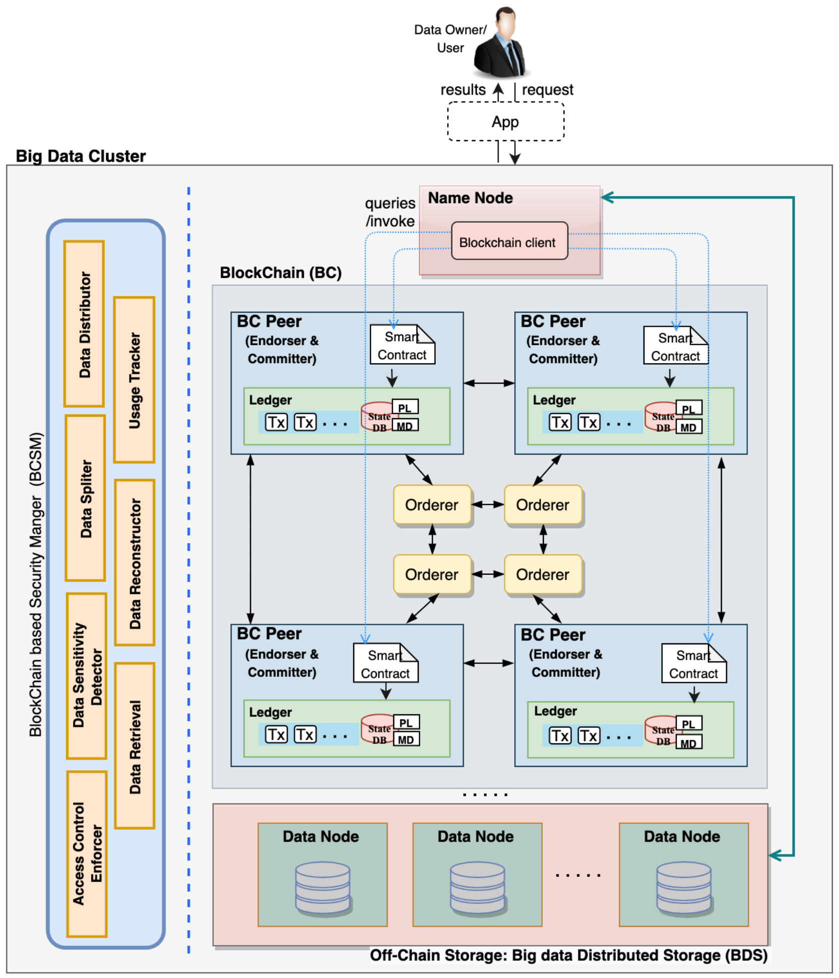

Fig. 1. Manager Architecture.

- ○ BC Peers: There are two types of peers endorser which hold the smart contract and committer; both peers host a copy of blockchain ledger.
- ○ Orderers: are a collection of many nodes in charge of generating an ordered list of transactions and creating blocks. Orderer is responsible for transaction hashing and block creation. The separation of smart contract execution and ordering transactions derived from Fabric architecture provides better performance and solves scalability issues compared to other blockchain platforms.

Furthermore, BC is responsible for keeping track of the system auditing logs.

- **On-chain and Off-chain Storage** Recent studies advocate storing the highly critical transactions that must be approved via blockchain consensus in order to avoid overwhelming the blockchain ledger[28]. Due to the limitations in blockchain storage, it is recommended to store the necessary critical data which require tamper-proof. These blockchain difficulties can be improved by using additional mechanisms applied on off-chain data such as fragmentation, scrambling, and calculating the hash of the dataset to preserve it on blockchain for checksum purposes.

Specifically, there are the following components that make up the proposed Security Manager BCSM:

1) Data Sensitivity Detector (DSD): The approaches of sensitivity detection are classified as automated, semi-automated, or manual. Our sensitivity detection relies on the data owner's (DO) policies and requirements. DO needs to specify the level of data sensitivity (high, low, or none) and indicate the sensitive attributes that must be protected.

2) Data Splitter (DS): We take advantage of fragmentation techniques to give an extra layer of data security. According to the user requirements, data is divided into sensitive and non-sensitive collections. By computing the SHA-256 for the original file and comparing the hashing result to the result of the file after the reconstruction process, the checksum is utilized to confirm data integrity. The security of sensitive data is handled by our manager based on the level of sensitivity. Scrambling is used to harden the fragmentation process for low-sensitive data, and this is complemented with distributed big data storage partitioning. Furthermore, to minimize the enormous cost of encrypting the entire data volume, our method performs encryption on the high-sensitive part of the dataset. The details of our fragmentation algorithm is presented in [6].

3) Data Distributor (DD): DD assigns dataset-id for each uploaded dataset to be referred to in merged files. It creates MD and PL based on a specific structure and inserts the merged files into big data storage. Moreover, DD sends MD and PL to be kept on the blockchain ledger and managed by the smart contract.

4) Data Retrieval (DR): DR gets data-hash and metadata from the blockchain using dataset-id. After that, it requests merged files from the BDS. Finally, the DR decrypts the metadata and sends it to the Data Reconstructor.

5) Data Reconstructor (DRE): DRE returns the data to its original version according to the metadata stored in the blockchain. This component applies decryption and defragmentation techniques in order to reconstruct the original data. Furthermore, the data-hash is retrieved to perform the checksum needed to check data integrity.

6) Access Control Enforcer (ACE): ACE handles data owner and user authentication and authorization processes. Once the ACE has authenticated the client, the authorization process is started. To verify the identity of a user, ACE employs multi-factor authentication. Data can only be accessed with the privileges specified in PL using ACL rules defined in the blockchain smart contract. Under the PL, only a selected group of users have access to the required data.

7) Usage Tracker (UT): This component is responsible for responding to data owner/auditor requests of acquiring auditing information. Auditing information related to data access and usage is retrieved from the blockchain utilizing the traceability feature given by the blockchain.

Fig. 2 illustrates the communication flow during the write operation, highlighting the interactions between different components starting from the Data Owner's request to upload the dataset to inserting in big data distributed storage. Fig. 3 illustrates the communication flow between several components throughout the reading process for sensitive data.

IV. Experimental Results and Analysis

A. Performance Measurement Tool

The process of performance evaluation means measuring the system performance, which is under test. This basically includes measuring what occurs when dependent variables are changed. Measuring blockchain network performance has been a significant concern among researchers and developers. The blockchain network consists of several peers that communicate with each other in order to collaborate to perform transactions.

We used Hyperledger Caliper v0.4.2 [29] to evaluate our solution performance. Hyperledger Caliper is a unified blockchain benchmark tool that integrates with different blockchain platforms. It allows us to test the performance of our manager components running under blockchain when interacting with client application requests for dataset read and write operations

TABLE I. Experiments Setup

Components	**Values**
Number of Organizations	2
Number of Endorsor Peers	2
Ordering Service	RAFT
Endorsement Policy	”OR(’Org1MSP.peer’)” , ”OR(’Org1MSP.peer’, ’Org2MSP.peer’)”
Block Size	10 transactions per block
Programming Language for smart contract	Nodejs
Ledger Database	CouchDB
Number of clients	2
Transaction duration	30s
Send Rates	50-650 tps

B. Performance Metrics

- **Transaction Throughput** is not measuring only at one node but across all nodes in the network. It is the rate at which the blockchain commits valid transactions in a specific time period, represented in transactions per second (tps).

- **Transaction Latency** is the time it takes for the whole network to validate a transaction, including broadcasting and allocation time used by the consensus algorithm.

- **Fail Rate** is described as the amount of failed transactions performed out of the total transactions

C. Experiment Environment Setup

Our blockchain platform is Hyperledger Fabric v2.3.3. The experiments were conducted on a host machine equipped with Intel Core i9 2.3 GHz, 16GB DDR6 memory, and a 1TB SSD hard disk. Table 1 discusses the default Fabric experiment setup. In this experiment, read or write operation is performed to virtual Hadoop cluster based on our previous study experiment [6]. The Hadoop cluster consists of one Name node and three Data nodes using the virtual machine manager VirtualBox 6.1.26.

D. Experiments Profile

To conduct our experiments, we have developed a Fabric Chaincode (smart contract), which is in charge to represent some functions of our manager components, including the authorization process of ACE, Data Distributor (DD), and Data Retrieval (DR). Moreover, the client application sending write and read requests is also implemented as part of the Caliper workload module to submit the transactions. Our previous experiment [6] started with preparing the dataset by fragmentation and encryption to ensure the efficiency of using the off-chain data storage security. Along with each operation, there is a call to blockchain to handle the management of that operation. In this paper, our concern is to test the performance of blockchain network during reading and writing operations.

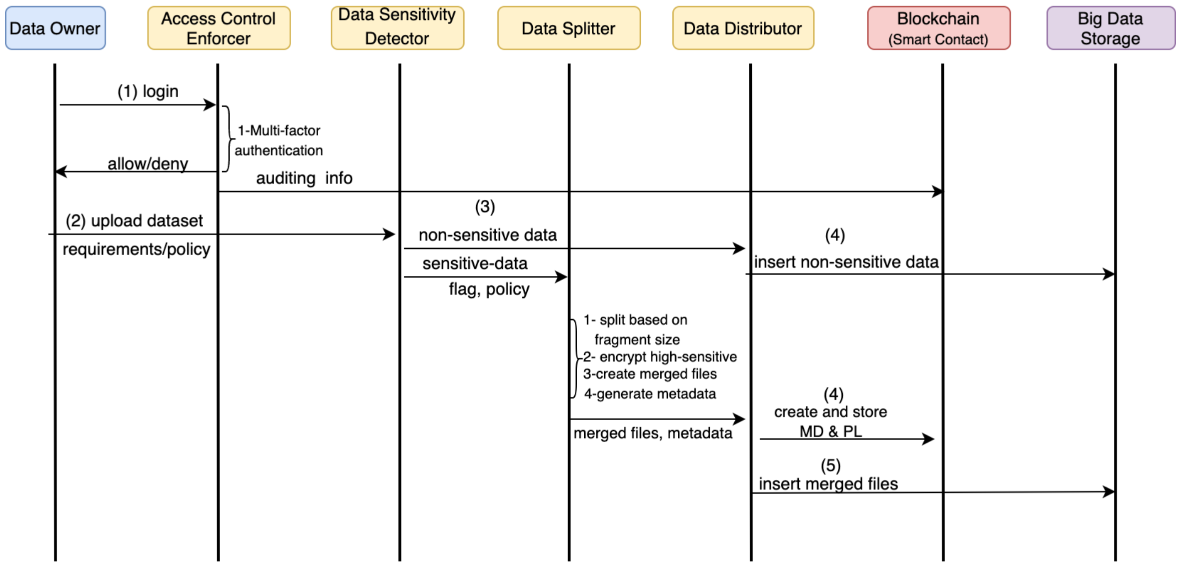

Fig. 2. Sequence Diagram for Writing Operation.

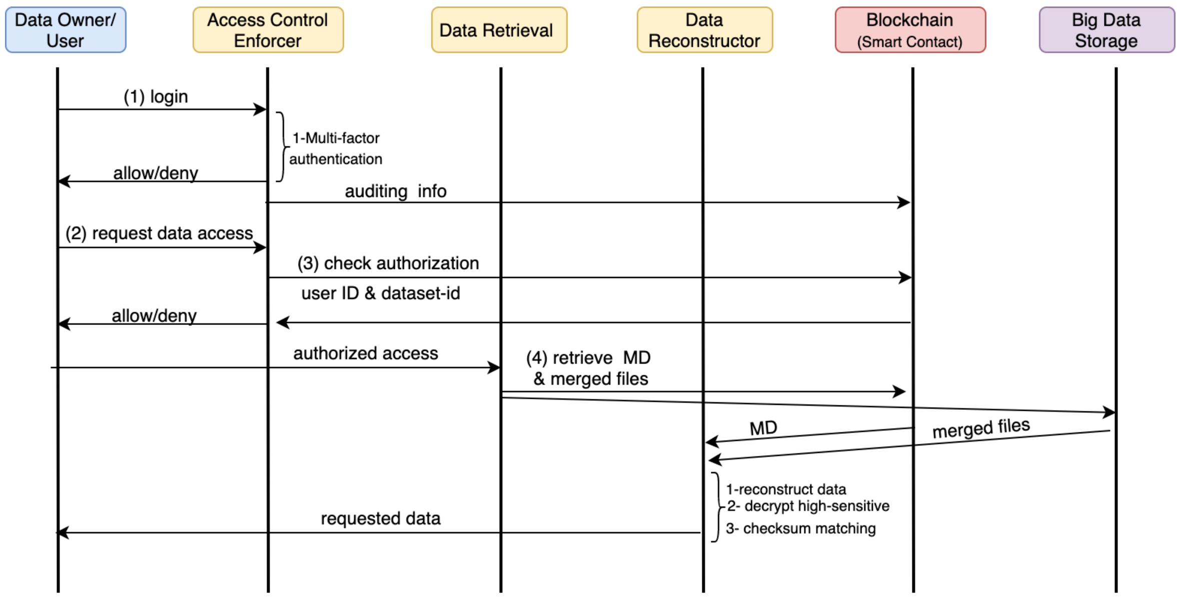

Fig. 3. Sequence Diagram for Reading Operation.

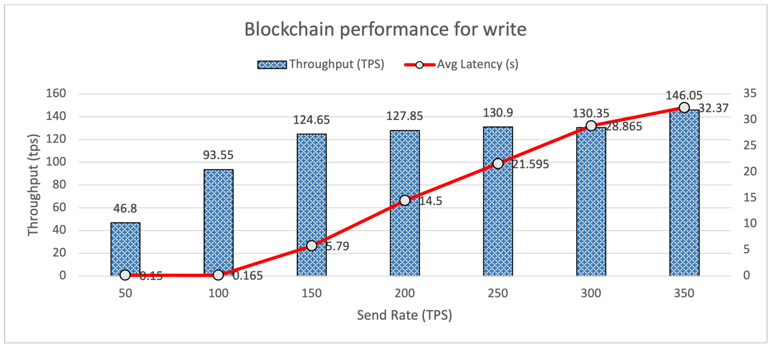

Fig. 4. Write Experiments Results.

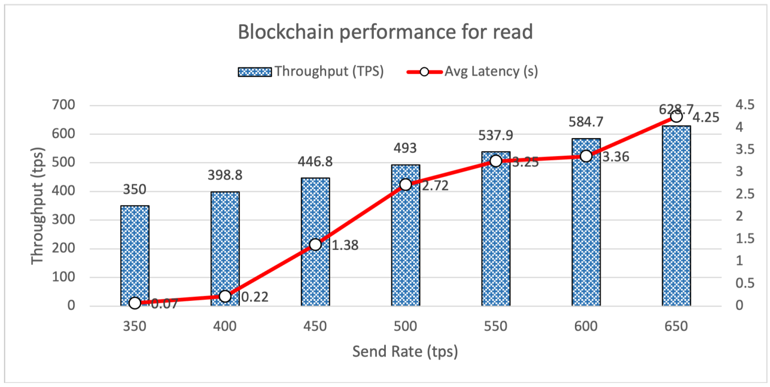

Fig. 5. Read Experiments Results.

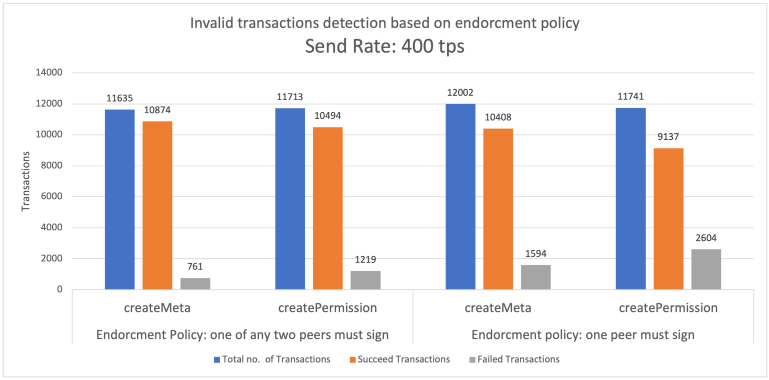

Fig. 6. Failed Transactions with Different Endorsement Policies.

E. Results and Analysis

Write Experiments: Fig. 4 plots the experimental results in terms of transaction throughput and latency for write operation. This operation involves two functions which are creating the metadata and permission lists to be stored in ledger CouchDB. The throughput increased linearly with the increase in send rate. The results show a remarkable drop in throughput when the send rate reaches 150 tps. However, the growth of transaction throughput had significant decreased approximately to half of the send rate value when the send rate was above 250 tps. Also, the figure plots the experimental results of transaction latency. When the send rate was above 150 tps, the latency had a significant increase.

Read Experiments: We evaluated the performance of our manager by varying transaction send rates (350 tps to 650 tps) to measure transaction throughput and latency. The Fabric has robust performance for reading and accessing assets stored in its ledger. In our read experiment scenarios, the results show no impact on throughput and average latency for send rates from 50 to 300 tps. The throughput reaches the same value as send rate with the same latency equals 0.01 seconds. Consequently, we started our performance evaluation for reading experiments from 350 tps when the performance showed a significant impact. We configured the test with a different number of transactions in each round of testing. Even though the average latency grows with the number of transactions, the rise is not sharp and growing very slowly. Fig. 5 plots the experimental results in terms of average transaction throughput. As shown in Fig. 5, the solution can process a throughput of around 350 to 628 transactions per second (authorization decisions) with an average latency of 0.07 to 4.25 seconds. Fig.5 shows the transaction throughput increased linearly with the increase in send rate. However, the transaction throughput increased until the send rate reached around 500 tps. The transaction throughput growth decreased when the send rate was above this point. Fig. 5 also shows the results of transaction latency. The transaction latency increases with the increase in the send rate. There is a small growth of latency for send rates from (350 tps to 400 tps). However, the growth of latency is increased from (450 tps to 650 tps).

Endorsement Policy in Write Experiments: Moreover, we evaluate our solution with different endorsement policies. Write experiments include the execution of two functions: (1) createMeta, which creates and inserts metadata into the ledger state (2) createPermission, which creates the permission list then inserts it into the ledger state. As depicted in Fig. 6, the experiment shows an impact on the number of failed transactions. In the case of ”OR(’Org1MSP.peer’)”, the peer from organization1 must sign. In the other case, ”OR(’Org1MSP.peer’, ’Org2MSP.peer’)”, one of any two peers can sign. This experiment indicates that the choice of endorsement policy has a significant impact on the number of invalid transactions.

V. Conclusion

This study presents our proposed manager BCSM, which aims to enhance big data security and privacy. Our security manager is based on blockchain technology, and we have developed a prototype manager using Hyperledger Fabric and Hadoop to test the feasibility of this solution. We have defined several big data security and privacy issues and proposed our manager to address these issues. Moreover, our solution takes into account the limitation of blockchain insufficient data storage. This solution can effectively solve problems such as big data leakage and tampering. Blockchain technology is still in its early stages. There is a limitation in state of the art to leverage blockchain for improving security for large-scale data application scenarios, particularly in the big data industry. The non-tampering and traceability of blockchain are expected to have significant benefits in the field. Our manager provides a secure environment for big data sharing, storage, and transmission. The blockchain is in charge of ensuring the security of big data storage and retrieval procedures, as well as access control and auditing mechanisms. Previous studies have not sufficiently addressed big data security issues; for example, they mainly focused on access control, data sharing, and auditing for specific big data applications such as smart homes and healthcare. We believe that almost all big data fields can refer to our suggested solution, which better solves big data security problems and the potential of blockchain technology in the future.

REFERENCES

[1] D. Laffly, "Big data in geography," *TORUS 1–Toward an Open Resource Using Services: Cloud Computing for Environmental Data*, pp. 45–54, 2020.

[2] "The apache^{TM} hadoop® project." [Online]. Available: https://hadoop.apache.org/docs/ (2022/02/20).

[3] D. Lv, S. Zhu, H. Xu, and R. Liu, "A review of big data security and privacy protection technology," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, pp. 1082–1091, 2018.

[4] T. A. Kumar, H. Liu, J. P. Thomas, and X. Hou, "Content sensitivity based access control framework for hadoop," *Digital Communications and Networks*, vol. 3, no. 4, pp. 213–225, 2017.

[5] P. Koopman, "Embedded system security," *Computer*, vol. 37, no. 7, pp. 95–97, 2004.

[6] H. E. Alhazmi, F. E. Eassa, and S. M. Sandokji, "Towards big data security framework by leveraging fragmentation and blockchain technology," *IEEE Access*, vol. 10, pp. 10768–10782, 2022.

[7] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, H. Karimipour, G. Srivastava, and M. Aledhari, "Enabling drones in the internet of things with decentralized blockchain-based security," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6406–6415, 2021.

[8] S. Yaqoob, M. M. Khan, R. Talib, A. D. Butt, S. Saleem, F. Arif, and A. Nadeem, "Use of blockchain in healthcare: A systematic literature review," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019.

[9] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, *et al.*, "Hyperledger fabric: a distributed operating system for permissioned blockchains," in *Proceedings of the thirteenth EuroSys conference*, pp. 1–15, 2018.

[10] I. Mokdad and N. M. Hewahi, "Empirical evaluation of blockchain smart contracts," in *Decentralised Internet of Things*, pp. 45–71, Springer, 2020.

[11] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." [Online]. Available: https://bitcoin.org/bitcoin.pdf (2019/07/02).

[12] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, "A survey on the security of blockchain systems," *Future Generation Computer Systems*, vol. 107, pp. 841–853, 2020.

[13] L. S. Sankar, M. Sindhu, and M. Sethumadhavan, "Survey of consensus protocols on blockchain applications," in *2017 4th international conference on advanced computing and communication systems (ICACCS)*, pp. 1–5, IEEE, 2017.

[14] M. Vukolić, "Rethinking permissioned blockchains," in *Proceedings of the ACM Workshop on Blockchain, Cryptocurrencies and Contracts*, pp. 3–7, 2017.

[15] P. Centonze *et al.*, "Security and privacy frameworks for access control big data systems," *Comput. Mater. Continua*, vol. 59, no. 2, pp. 361–374, 2019.

[16] S. D. Capitani di Vimercati, P. Samarati, and S. Jajodia, "Policies, models, and languages for access control," in *International Workshop on Databases in Networked Information Systems*, pp. 225–237, Springer, 2005.

[17] E. Damiani, S. D. C. Di Vimercati, and P. Samarati, "New paradigms for access control in open environments," in *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology, 2005.*, pp. 540–545, IEEE, 2005.

[18] Y. Xiao, N. Zhang, J. Li, W. Lou, and Y. T. Hou, "Privacyguard: Enforcing private data usage control with blockchain and attested off-chain contract execution," in *European Symposium on Research in Computer Security*, pp. 610–629, Springer, 2020.

[19] C. Wang, S. S. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for secure cloud storage," *IEEE transactions on computers*, vol. 62, no. 2, pp. 362–375, 2011.

[20] F. Zafar, A. Khan, S. U. R. Malik, M. Ahmed, A. Anjum, M. I. Khan, N. Javed, M. Alam, and F. Jamil, "A survey of cloud computing data integrity schemes: Design challenges, taxonomy and future trends," *Computers & Security*, vol. 65, pp. 29–49, 2017.

[21] A. J. N. Gupta and P. Roy, "Adopting blockchain technology for electronic health record interoperability," tech. rep., Cognizant Technology Solutions, New Jersey, U.S, 2016.

[22] C. Stamatellis, P. Papadopoulos, N. Pitropakis, S. Katsikas, and W. J. Buchanan, "A privacy-preserving healthcare framework using hyperledger fabric," *Sensors*, vol. 20, no. 22, p. 6587, 2020.

[23] L. Yue, H. Junqin, Q. Shengzhi, and W. Ruijin, "Big data model of security sharing based on blockchain," in *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)*, pp. 117–121, IEEE, 2017.

[24] M. R. Amin and M. F. Zuhairi, "Review of fscm with blockchain and big data integration," *Indian J. Comput. Sci. Eng*, vol. 12, pp. 193–201, 2021.

[25] I. Makhdoom, I. Zhou, M. Abolhasan, J. Lipman, and W. Ni, "Privysharing: A blockchain-based framework for privacy-preserving and secure data sharing in smart cities," *Computers & Security*, vol. 88, p. 101653, 2020.

[26] J. Chen, Z. Lv, and H. Song, "Design of personnel big data management system based on blockchain," *Future Generation Computer Systems*, vol. 101, pp. 1122–1129, 2019.

[27] C. Zhang, Y. Li, W. Sun, and S. Guan, "Blockchain based big data security protection scheme," in *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pp. 574–578, IEEE, 2020.

[28] J. Eberhardt and S. Tai, "On or off the blockchain? insights on off-chaining computation and data," in *European Conference on Service-Oriented and Cloud Computing*, pp. 3–15, Springer, 2017.

[29] "Hyperledger caliper—a blockchain benmark tool." [Online]. Available: https://www.hyperledger.org/use/caliper (2021/12/02).