A BLOCK-CHAIN APPROACH TO GUARANTEE THE PROBITY OF THE CLOUDDATA DELIVERED IN A DISTRIBUTED ENVIRONMENT

Dr. A. Satyanarayana S. Bhanu Charan M. Naga Rikit M. Yashwanth M. Niharika MD. Zaid Siddiqui

Abstract* — Cloud servers allows data owners to transmit and storesecurely encrypted information that numerous users can access. However, once data is exported to the cloud, data owners have limited access to their data, and external tools are used for managing it. Several scientific approaches use algorithms for encryption to restrict unauthorized access to data, but they neglect the difficulty of maintaining track of valid changes performed on the data. As provenance data contains private information, it should be unchangeable and protected from adversaries because it can be used to determine the integrity of data. This paper proposes using block-chain network to secure access logs in an efficient way. A flexible framework is created, tested, and evaluated, with the results demonstrating that our model may accurately improve the data provenance security. This work takes into account two types of data users, as well as their separate roles and permissible behaviour on the outsourced data. In short, this job ensures the data's trustworthiness, as well as the verification and administration of the outsourced data. The experimental findings demonstrate our solution's efficiency and scalability.

Keywords —*Cloud Servers, Unauthorized Access, Provenance Data*

1. INTRODUCTION

With the rising amount of data generated daily through the useof various apps, services to store and manage the data are required [1]. With the advent of cloud computing, it is now possible to obtain storage and processing services [2]. In general, cloud service providers (CSPs) make infrastructures available for data storage and processing by charging a fee per use. These services reduce the expenses of establishing and maintaining systems designed to fulfill in-house data requirements indicated by a data stakeholder [3]. As a result, people and organizations are drawn to use computer and storage services provided by third parties and made available on demand. As a result, stakeholders must have faith in the service. As a result, stakeholders must rely on service providers to securely preserve their data and corresponding metadata [4]. Users generally encrypt data before storing it since the data can be vulnerable to unauthorized usage [5]. Adopting outsourced storage with CSPs presents a hurdle in terms of access control. This problem stems from the use of data in which various users are permitted varied access privileges, posing a problem in the development and management of decryption keys. The potential remedy to this issue is to create fine-grained access control over outsourced encrypted data [6]. Attribute-based encryption (ABE) [7] and secret-sharing [8] approaches provides an array of options to determine which users have access to the data. However, this method falls short of providing means to ensure trust in the use (origin and alterations) of data throughout its life cycle.

The development of data provenance systems to ensure confidence in data interchange systems could be one answer to this problem. A provenance system offers information indicating where the data originated, who owns the data, and the many alterations the data has undergone. These include the location of the data as well as the numerous timestamps associated with its generation and use. The deployment of a provenance system, on the other hand, does not completely undermine trust. The key challenge is gathering, storing, and maintaining the confidentiality and privacy of provenance information. Implementing a method or technology that assures the security and privacy of provenance information is critical. Furthermore, provenance information should be verifiable without jeopardizing the privacy and security of individuals thedata [9].

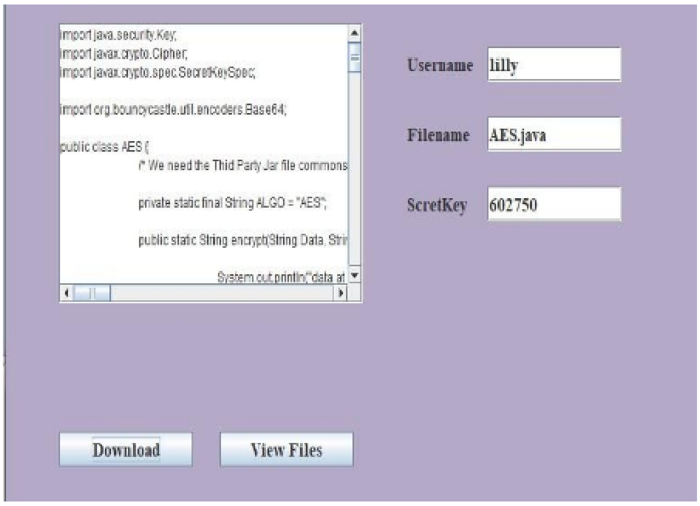
In this paper, we propose a block chain-based provenance system for a data-sharing ecosystem. Our solution uses the block chain network and intelligent contracts to eternally store and validate metadata aggregated as logs from events and may be applied to a wide range of use cases. The suggested approach ensures user verifiability when obtaining data from CSPs. Our system's design enables authorized system participants to perform write operations on data while also giving the data owner with access and control over the outsourced data. The paper's contributions are summarized below.

Based on an on-chain and off-chain process classification, we propose and build a block chain architecture for attaining provenance in a data-sharing ecosystem.

We emphasize a data owner's ability to retain control over outsourced data. This is accomplished by ensuring that all data modifications are confirmed with permission from the data owner based on access policies that specify actions that are applied to the data.

We provide a block chain data view that underlines the importance of a tamper-proof log in enabling traceability through the aggregate of data transactions that are part of event logs.

We give a performance evaluation and analysis of the system on an ethereum block chain proof of concept, confirming the feasibility of implementing the proposed solution.



```
import java.security.Key;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;

import org.bouncycastle.util.encoders.Base64;

public class AES {
                /* We need the Thid Party Jar file commons

                private static final String ALGO = "AES";

                public static String encrypt(String Data, Stri

                                System.out.println("data at
```

Username: lilly

Filename: AES.java

ScretKey: 602750

Figure 1: Example of a generated cloud data file, composedof an username, filename and the secret key.

2. LITERATURE SURVEY

1. A. Related Work

Over the past few decades, a great deal of work has been done in the area of cloud computing regarding secured data, which is typically characterized as a two-stage process consisting of Owner Name and User Name. Extracting significant data from owner so they may be stored at Data Provenance System.

Several block-chain-based provenance solutions have been presented to address traceability and log aggregation in Internet of Things (IoT). However, these works are deficient in providing insight into the block structure of the block chain network, therefore inhibiting the feasibility of the proposed solutions.The following research works helped us in working on this project.

"Big data analytics," in Springer Briefs in Computer Science.

Authors: B. N. Silva, M. Diyan, and K. Han,BerlinPublication: 2016 in Springer.

Summary: Big data is a new force in global economic and societal change. The world's data collecting is approaching a tipping point for big technological advances that could usher in new approaches to decision making, city management, finance, and education. While data challenges such as volume, variety, velocity, and veracity are increasing, the real impact is dependent on our ability to identify the 'value' in the data using Big Data Analytics tools. Big Data Analytics presents a significant challenge in the design of highly scalable algorithms and systems to integrate data and find substantial hidden values from diverse, complicated, and massive datasets. Potential breakthroughs in Big Data Analytics include innovative algorithms, techniques, systems, and applications that reveal meaningful information. extracting hidden insights from Big Data in an efficient and effective manner.

Security issues in cloud computing.

Authors: M. Vijaya kumar, V. Sunitha, K. Uma, and A.Kannan, J. Adv. Res. Dyn.

Publication: Journal of Advanced Research in Dynamical and Control Systems 2017.

Summary: There is no doubt that cloud computing has numerous benefits, but it also has certain security concerns. The following are some security issues in cloud computing: Data Loss.

Interference of hackers and Insecure API's. User Accounting Hijacking. Changing Service Provider. Lack of Skill. Denial of Service (DoS) attack.

"Towards trusted cloud computing," in Proc.

Authors: N. Santos, K. P. Gummadi, and R. Rodrigues. Publication: USENIX-SS'06, Berkeley, CA, USA, 2006.

Summary: Cloud computing infrastructures enable companies to cut costs by outsourcing computations on-demand. However, clients of cloud computing services currently have no means of verifying the confidentiality and integrity of their data and computation. To address this problem we propose the design of a trusted cloud computing platform (TCCP). TCCP enables Infrastructure as a Service (IaaS) providers such as Amazon EC2 to provide a closed box execution environment that guarantees confidential execution of guest virtual machines. Moreover, it allows users to attest to the IaaS provider and determine whether or not the service is secure before theylaunch their virtual machines.

"Data provenance in the cloud: A block-chain based approach"

Authors: D. Tosh, S. Shetty, X. Liang, C. Kamhoua, and L. L. Njilla

Publication: IEEE Consumer Electronics Magazine (Volume: 8, Issue: 4, July 2019)

Summary: Ubiquitous adoption of cloud computing and virtualization technology has necessitated the need for strong security mechanisms. Multiple entities are involved in creating, exchanging, and altering data objects in the cloud environment,making it challenging to track malicious activities and security violations. To address these issues, there is a need for a data provenance framework, with which each data object in the federated cloud environment can be tracked and recorded. Although log-based provenance provides the ability to track operations conducted on digital assets, the provenance data are not transparent and immutable. Block-chain technology offers apromising mechanism for building a tamper-proof informationsystem backed by strong cryptographic primitives. In this article, we propose Block-Cloud, a block-chain empowered data provenance architecture for the cloud computing platform. In addition, we present a proof-of-stake (PoS) consensus mechanism for Block-Cloud to alleviate the overhead of computational requirements that the traditional proof-of-work (PoW) consensus needs. Finally, we discuss several research challenges and vulnerabilities that need to be addressed to realize Block-Cloud.

"Using block-chain and smart contracts for secure data provenance management"

Authors: A. Ramachandran and D. Kantarcioglu

Publication: 2017 in arXiv.

Summary: Blockchain technology has evolved from being an immutable ledger of transactions for cryptocurrencies to a programmable interactive the environment for building distributed reliable applications. Although, block-chain technology has been used to address various challenges, toour knowledge none of the previous work focused on using block-chain to develop a secure and immutable scientific data provenance management framework that automatically verifies the provenance records. In this work, we leverage block-chain as a platform to facilitate trustworthy data provenance collection, verification, and management. The developed system utilizes smart contracts and open provenance model (OPM) to record immutable data trails. We show that our proposed framework can efficiently and securely capture and validate provenance data, and prevent any malicious modification to the captured data as long as the majority of the participants are honest.

1. B. Proposed Model

We propose a block-chain-based provenance system for a data-sharing ecosystem in this paper. Our solution uses the block-chain network and smart contracts to eternally store andcheck metadata aggregated as logs from events and can be applied to a wide range of use cases. The suggested approach ensures user verifiability when acquiring data from CSPs. Oursystem's framework allows authorized system participants to perform write operations on data while also giving the data owner with access and control over the outsourced data. This article's contributions are summarized below.

Based on on-chain and off-chain processcategorization, the system proposes and implements a block-chain architecture for attaining provenance ina data-sharing environment. The system emphasises a data owner's capacity to retain control over outsourced data. This is accomplished by ensuring that all data modifications are confirmed with consent from the data owner based on access policies that specify actions that are applied to the data. The solution provides a block-chain data view that highlights a tamper-proof log in establishing traceability through the aggregation of data transactions that form part of event logs. The system gives a performance evaluation and analysis of the system on an ethereum block-chain proof of concept, confirming the feasibility ofImplementing the proposed solution.

Our proposed model provides insight into the structure and generation of blocks in a block-chain network classified under views. This structure enhances the visibility and traceability of provenance logs in the provenance system. The smart contract models account for dynamic policy updates for user data to efficiently provide a layer of access control on the outsourced-data.

METHODOLOGY

For ease of understanding, this section presents an overview of the provenance framework for our proposed solution. We consider a healthcare environment where a patient’s data is stored on a cloud server and shared among healthcare practitioners. This is depicted in Fig. 2. Access to the outsourced encrypted data is managed through policies andpublic keys established by the owner of the data. Write access is needed to activate provenance protocols to keep track of document changes as part of aggregating provenance information for our solution. Versioning allows system entities (nodes) to keep track of the current data as each accepted change enforced on the data is mapped to its parent, from which the current data was formed. The change of the document is stored as the current view of the data. We emphasize the fact that the latest version of a document is the current view of the file for future access or modifications by data stakeholders; however, a data owner has visibility of the entire data from its genesis to its current view. Finally, documents with changes not logged in the provenance data are ignored by the system. This section highlights, in summary, thesteps for which modified data is changed from its modified state to the state of current view. The first step starts with the initialization of system parameters for the data owner and stakeholders through the processes of outsourcing the encrypted data and the generation of decryption keys for the encrypted data. Fig. 3 describes the data state managementmentioned in this section.

Step 1: A data owner stores data on a cloud repository.This is the current view Dcur of the data to be assessed.

Step 2: A validated stakeholder with write permissions, access data from the CSP for modification. The data accessed is the current view Dcur of the data to be modified.

Step 3: A stakeholder submits the modified version D cur to the data owner and the blockchain provenance system for validation.

Step 4: A data owner sends an acceptance or rejection, which is a transcript of an agreement, to change or reject the modification, to the provenance system.

Step 5: The provenance system rejects or accepts D cur based on the transcript received from the data owner. The blockchain provenance system computes the digest of the modified version H(D cur).

Step 6: The provenance system performs computationsto compare the digest of the hash of the current view of the data, H(Dcur) = H(D cur). If H(Dcur) = H(D cur), the change is accepted and D curbecomes Dnew cur ; else, the change is rejected because no modifications have been applied on the data.

Step 7: All actions performed on the data areconsidered as data logs and are processed into blocks and stored on the blockchain network.

Step 8: Dnew cur is finally stored on the cloud. Transcripts of these actions sent to the data owner as a reference to the data chain. These are done so that different versions of the data can be easily maintained and errors reverted to the old state

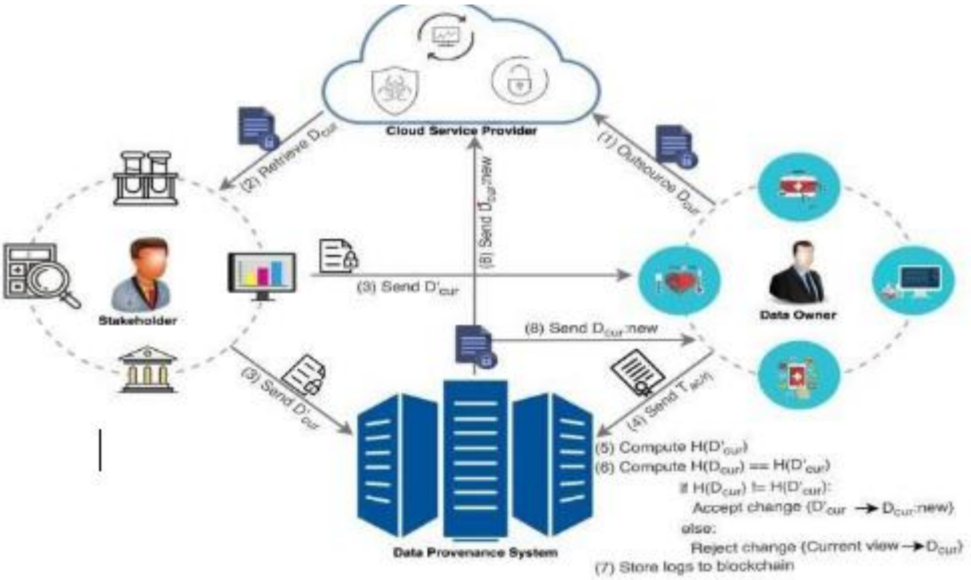


Figure 2: Provenance procedure: data state management

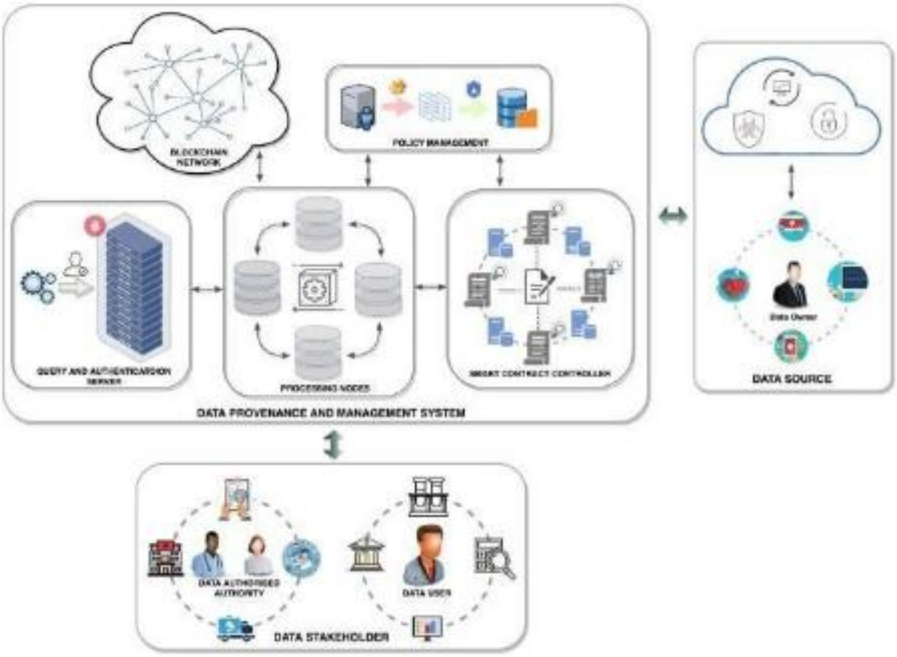

BLOCKCHAIN NETWORK
POLICY MANAGEMENT
QUERY AND AUTHENTICATION SERVER
PROCESSING NODES
SMART CONTRACT CONTROLLER
DATA PROVENANCE AND MANAGEMENT SYSTEM
Data Owner
DATA SOURCE
DATA AUTHORISED AUTHORITY
DATA USER
DATA STAKEHOLDER

Figure 3: Subsystems of the provenance architecture

Data Owner will first register in the cloud service provider and then login into it. Now Data Owner will upload all of his data into cloud service provider. With the help of Stakeholder, all the data which is uploaded by Data Owner will be stored inData Provenance System.

Now User will register in the cloud service provider and logins into it. User wants to retrieve some data from the Data Provenance System, so the user requests the cloud service provider to provide the data.

Now User will get to know the secret key and file name. Using these secret key and file name, the end user will enter it and get the required data or file to download and access it. Once the file is accessed or downloaded, that file will no longer be present or available in the cloud service provider.

If the end user doesn't enter the right secret key or the file name, then the cloud service provider will block the end user. This is because to make use the confidentiality, integrity and authentication as this is done in block chain network.

Now let’s discuss about the proposed framework for the data provenance system. The framework is interoperable with cloud storage services for various application scenarios. For our design, the data provenance structure consists of an overall block-chain, data block-chain, and storage block- chain. The storage block-chain in turn consists of application and storage models. The storage block-chain is responsible forstoring provenance data, related to a specific application. These structures are modified based on the requirements for the related application.

Data Model: The generalized data model for our provenance solution is adopted from. We state definitions for our model as a data point d for a provenance record. A provenance record is a tuple defined as {Did, loc(Did), Δ(Did), in(Δ(Did))}. Did is the unique identifier of the data created from the data owner's identifier, a string of sequence characters, and the version of the data. Versioning allows every entity to know the current view of the data. loc(Did) is the address of the data; this is a pointer to where the data was retrieved to ensure that data from other sourcesare not presented as the original data from the cloud storage. Δ(Did) is a reason-set of logs stating the purpose for performing write operations on the data. This is validated by the user and processing servers in the provenance system. in(Δ(Did)) defines the specific write operations on the data. Write operations can be validated by the data owner before the view of the data is changed since this is the point of information interest for stakeholders. This model allows for the creation, modification, and deletion of data points inthe provenance system. This model merged with the blockchain, achieves data integrity for the recorded data.

Storage Model: This model is responsible for the storage of provenance records. It is the representation for provenance defined under the data model and has create, retrieve, update, and delete functions. The storage model consists of two layers, that is, overall block (log) storage and data log storage. Operational functions on provenance records are applied to transactions before they are committed to the storage network. An unapproved or invalid record is neglected and not stored in the provenance system. Smart contracts are enforced to allow the storage of provenance data in the storage network.

In this work, we specify the structure of transactions for the blocks and the formation of the chain in the block-chain network. The block-chain network is structured under two separate views, namely user view and data view. This is specified in Fig. 4. For the data view, the root of the chain is thedata identifier which extends to form two genesis blocks defined as the branches of the identifier. These initiate the start of the formation of blocks on the block-chain network. These blocks have information on read and write operations, respectively. The root, therefore, forms part of an identity chain specifying the data identifier and the various stakeholders related to the data in the provenance system. Transactions in theroot are made up of the multi signatures and the various timestamps on when the data was created and stored in the cloud. Appended to the genesis blocks are the user blocks whichspecify the identities of stakeholders for the data. These blocks form part of an aggregation of data trials for the use of the data. Transactions that form the user block on the read genesis block is the user identity, timestamp of access for the data, and multi signatures of the stakeholder and owner.

Transactions that form the user block on the write genesis blockare the status of affiliation for the stakeholder and the data, the user identity, timestamp of access for the data, and multi signatures of the stakeholder and owner. Transactions of logs represent the data operations on the data and the time of access and operations on the data.

The user view is formed from two genesis blocks which is the oot for the identifier of the user. The root consists of read andwrite blocks that denote the initialization of the formation of blocks on the separate chains. This can be considered as an identity chain of the provenance system for user identity validations. A combination of the identifier of the users, user signatures, and event timestamps defines transactions for the root of the user view. Fig. 4 describes data trails, which are based logs on the use of data, appended to the genesis blocks. Transactions in the read block are generated from the data accessed, the various timestamp of events related to the accessof the data, and the digital signatures of the stakeholder and owner. These are different from write transactions because transactions should relate to a data owner's affiliation to form a multi signature.

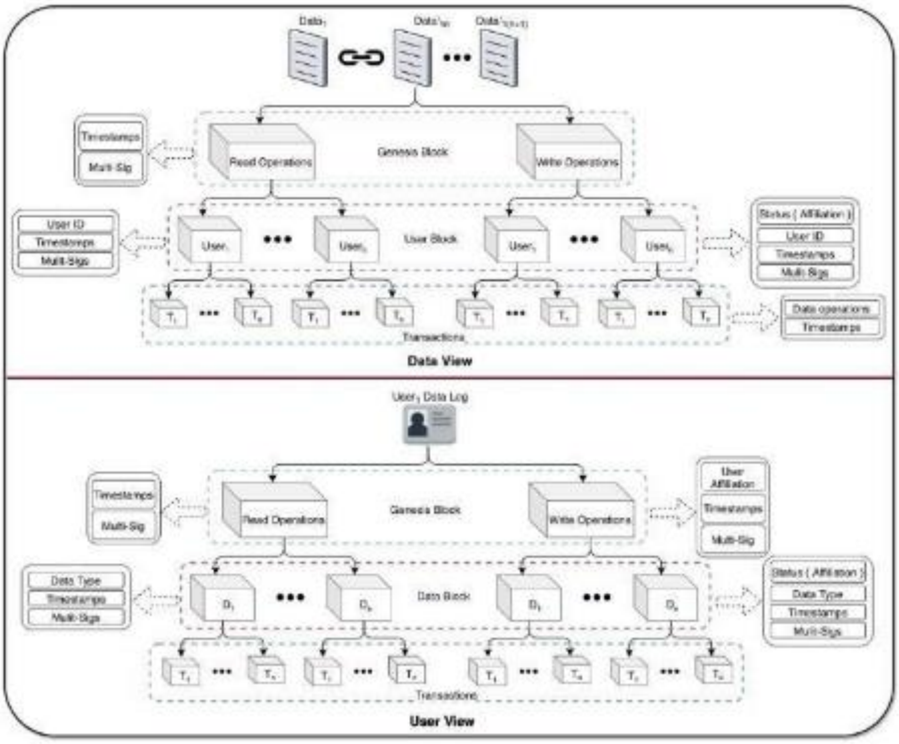


Figure 4: Block-chain data views

CONCLUSION

Block chain technology, in conjunction with smart contracts, is employed in this work to provide efficient access control to outsourced data in provenance systems. An owner can control and monitor an outsourced encrypted health record using the technology offered. The designed system paradigm assures that user verifiability is efficiently realized and that data is immutably saved and validated. The implementation of smart contracts allows for penalties to be levied to system defaulters by constant monitoring of activities performed on data by system members, together with enforced revocation. Finally, our solution assures data confidentiality, integrity, and authorization, thereby making the system secure. Experiment results suggest the efficiency and scalability of our proposed method based on its performance solution.

FUTURE SCOPE

More study is required in order to try this with high quantity and quality of data in future work. Presently, we have used AES algorithm but in future this concept can be approached with different block chain algorithms to provide atmost high – level integrity, confidentiality and authentication to secure the data. More study can be done to ensure the secret keys and private keys to be more confidential by not letting the hackers or other persons to know the private details.

REFERENCES

[1] B. N. Silva, M. Diyan, and K. Han, "Big data analytics," in *Springer Briefs in Computer Science*. Berlin, Germany: Springer, 2019.

[2] A. Lele, "Big data analytics," in *Smart Innovations, Systems, and Technologies.*Berlin, Germany: Springer, p. 3–3, 2019.

[3] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing," in *Proc. Workshop Hot Top. Cloud Comput.*, HotCloud'09,2020.

[4] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey," *J. Netw. Comput. Appl.* vol. 79, pp. 88– 115, 2017.

[5] M. Vijayakumar, V. Sunitha, K. Uma, and A. Kannan, "Security issues in cloud computing," *J. Adv. Res. Dyn. ControlSyst.*, vol. 4, no. 1, pp. 1–13,2017.

[6] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving fusion of IoT and big data for e-health," *Futur. Gener. Comput. Syst.*, vol. 86, pp. 1437–1455, 2018.

[7] V. Goyal, O. Pandey, A. Sahai, and B.Waters, "Attribute- based encryption for fine-grained access control of encrypted data," in *Proc. 13th ACMConf. Comput. Commun. Secur.*, 2007, pp. 89–98.

[8] B. Fabian, T. Ermakova, and P. Junghanns, "Collaborative and secure sharing of healthcare data in multi-clouds," *Inf. Syst.*, vol. 48, pp. 132–150,2015.

[9] D. Tosh, S. Shetty, X. Liang, C. Kamhoua, and L. L. Njilla, "Data provenance in the cloud: A blockchain-based approach," *IEEE Consum. Electron. Mag.*, vol. 8, no. 4, pp. 38–44, Jul. 2019.

[10] R. K. Lomotey, J. C. Pry, and C. Chai, "Traceability and visual analytics for the Internet-of-Things (IoT) architecture," *World Wide Web*, vol. 21,pp. 7–32, 2018.

[11] E. Nwafor, A. Campbell, D. Hill, and G. Bloom, "Towards a provenance collection framework for internet of things devices," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov.*, 2017, pp. 1–6

[12] M. H. Chia, S. L. Keoh, and Z. Tang, "Secure data provenance in home energy monitoring networks," in *Proc. 3rdAnnu. Ind. Control Syst. Secur. Workshop*, 2017, pp. 7–14.

[13] X. Liang, S. Shetty, D. Tosh, C. Kamhoua, K. Kwiat, and

L. Njilla, "Provchain: A blockchain-based data provenance architecture in cloud environment with enhanced privacy and availability," in *Proc. 17th IEEE/ACM Int. Symp. Cluster, Cloud Grid Comput.*, May 2017, pp. 468–477.

[14] A. Ramachandran and D. Kantarcioglu, "Using blockchain and smart contracts for secure data provenance management," 2017, *arXiv:1709.10000.*

[15] M. Sigwart, M. Borkowski, M. Peise, S. Schulte, and S. Tai, "A secure and extensible blockchain-based data provenance framework for the Internet of Things," *Pers. Ubiquitous Comput.*, 2020, pp. 1–15.

[16] H. Olufowobi *et al.*, "Data provenance model for Internetof Things (IoT) systems," in *Proc. Serv.-Oriented Comput.- ICSOC Workshops: Revised*