International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Jun-2022)

http://dx.doi.org/10.12785/ijcds/1101117

Blockchain Driven Secure and Efficient Logging for Cloud Forensics

Sagar Rane[1], Sanjeev Wagh[1, 2] and Arati Dixit[1, 3]

[1] Department of Technology, Savitribai Phule Pune University, Pune, MH, India
[2]Government College of Engineering, Shivaji University, Karad, Maharashtra, India
[3]Applied Research Associates Inc., Raleigh, North Carolina, USA

Received 16 Feb. 2021, Revised 11 Apr. 2022, Accepted 18 Apr. 2022, Published 15 Jun. 2022

Abstract: Cloud computing is one of the most holistically used technology nowadays. To do the forensics in the cloud, it is essential to accumulate as well as safeguard the justifiable facts of different events to find out the culprit. However, logging the events that take place in the cloud and securing them while preserving the privacy of cloud consumers is a big challenge. Currently, cloud consumers are relying on cloud service providers to get the logs of events that take place on their data despite multi-stakeholder collusion. Thus, there is a strong requirement of publicly verifiable secure logging which will play a vital role in criminal investigations without depending on a third party. In this paper, we developed a Blockchain-driven Secure Logging-as-a-Service (BlockSLaaS) scheme that supports, privacy-preserving secure logging and eclectic verification. To make a serene forensic investigation, the proposed scheme guarantees the trustworthiness and reclamation of logs in case of tampering. The scheme proposes integrating the Interplanetary File System (IPFS), a decentralized off-chain data storage platform with blockchain for efficient logging and its visualization. The extensive experiments on the number of transactions, storage requirement, uploading, reading, and downloading of log files for varying node count and file size are performed. The proposed method is compared with nine existing methods based on 9 security and performance features. The response, proof insertion, and proof-verification times of the proposed BlockSLaaS are 38.3, 29.7, and 26.3 milliseconds respectively which outperform the existing methods.

Keywords: Blockchain, Cloud Computing, Cloud Forensics, Secure and Efficent Logging, Forensic Investigation.

1. INTRODUCTION

Today, cloud computing services are widely used in various industries due to a tremendous efficiency of cost over conventional storage services [1] [2]. Currently, the market of cloud-based data storage is on the upsurge due to the successful espousal of cloud facilities in almost all companies. In India, cloud computing market will be valued at seven billion dollars by 2022 and expected to cross 1 trillion dollars by 2025 globally as per the NASSCOM report [3] [4]. However, the shift from onsite storage techniques to cloud storage services is a big challenge due to the rise in the issues of data security [5-11]. Certain malevolent cloud consumers can utilize the cloud storage to stock illegitimate information including but not limited to stolen Intellectual Property Rights (IPR) documents, pornographic content, and contraband documents or can target other cloud consumers by hosting the malware injection attacks, denial of service attacks, wrapping attacks, structured query language injection attacks, abuse and hijacking of services on cloud computing environment [12]. Once the attackers accomplish the unethical goal, they can smoothly wipe out the hints, traces, and remain unidentified [13] [14]. Therefore, there is a strong requirement of procedures and scientific methods to ensure trustworthiness and confidentiality of data in cloud computing environment for effective forensic investigations [2] [15]. Consequently, a new branch of forensics came into existence i. e. Cloud Forensics. Federal Bureau of Investigation (FBI) report of 2017 on internet crime statistics depicted that, over 3 lakh online misconduct complaints have been registered which amounted to around fifteen thousand-million-dollar loss in the year of 2017 itself [12]. The count of digital forensic belongings is on the upsurge [16]. The existing forensic methods and techniques cannot be applied to the cloud directly due to the nature of the cloud. Also, they require to be modernized to be competent and suitable for the cloud environment [6] [15]. In cloud computing, virtual

E-mail: scholarsaggy@gmail.com

machines (VMs) may be located out of the jurisdiction and they may consist of volatile data which could be lost once the VM turns off. Thus, forensic investigations of virtual machines' volatile data pose a grave challenge from the technical, organizational, and legal perspective [17].

The behavior of cloud consumers which is captured into the activity logs has the ability to report as to what events happened in the cloud [13]. Hence, logs are a crucial element to gain insight into any malicious activity. As of today, once the malevolent action got reported law enforcement agency has to depend on the activity logs that are given through the Cloud Service Provider (CSP) [18]. The most experienced crime perpetrators majorly focus and destroy the logging facilities and service stations to do away with the hints and traces of their nasty actions [7]. Therefore, integrity and confidentiality of responsible log marks is a leading worry to conduct forensic investigations in the cloud environment. Some invader may utilize the cloud environment to do the denial of service attack on the collocated apps successively on cloud system & can also lay off their virtual machines or attempt to fabricate the logs to remain untraceable. Figure 1. shows some advantages of using blockchain in secure logging.

Researchers have proposed various techniques to increase reciprocal trust among stakeholders of cloud systems i.e. Cloud Service Provider (CSP), Cloud Service Consumer (CSC), and Cloud Forensic Investigator (CFI). Few are in the field of secure logging [12] [19] but, using probabilistic data structures. Cloud data auditing schemes was presented like dynamic [18] [20] [21], shared [18] [22] [23] and privacy protected [24] [25] in existing work. Unlike the above-mentioned techniques and schemes whose focus was to protect cloud users' data, we aim to develop a privacy-preserving secure logging scheme for cloud users' activity logs. Extending SecLaaS [12] and CLASS [19], we develop blockchain-driven secure logging-as-a-service scheme (BlockSLaaS), a two-step privacy preservation scheme which certifies (a) CSP is putting correct information into the log file for associated cloud consumer event and (b) eclectic verification of logs without having actual logs. This paper is an extended version of our previous work [26] [27].

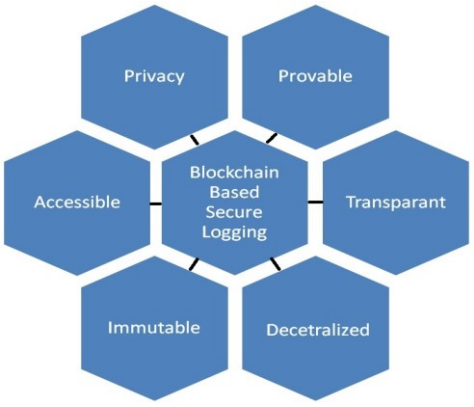


Figure 1. Advantages of blockchain in Secure Logging

According to the authors' knowledge, there is no work carried out on privacy-preserving secure logging for cloud forensics. The privacy of cloud consumers' is a very important aspect while considering the cost of a logging system. Thus, we propose an integration of blockchain and IPFS for efficient and cost-effective logging while maintaining the privacy, integrity, and confidentiality of cloud consumers' data. Our proposed scheme is publicly verifiable which means any authorized peer can verify the eclectic logs, unlike existing methods. Further, BlockSLaaS provides original logs via IPFS in case of tampering. This research work gives better access control to the cloud consumers on their data. Previous works have not comprehensively considered the multi-stakeholder collusion problem in the context of cloud forensics. There is no published analysis of the factors affecting the performance of secure logging. The extensive experiments on the number of transactions, storage requirement, uploading, reading, and downloading of log files for varying node count and file size are performed. Log visualization mechanism has been developed on top of IPFS for a better understanding of the process of cloud forensics. The following hypothesis describes the precise concern and apprehensive research problem which we planned to solve.

Hypothesis: Imagine Sagar is the proprietor of an international firm that provides cloud services. He has set up a cloud in different continents to manage the data for finance and insurance applications. He has demanded from the technical team a backup of every second for all the applications for financial and insurance data. A technical team of the firm configured the system to take the backup which guarantees that there is adequate space available to save the dealings.

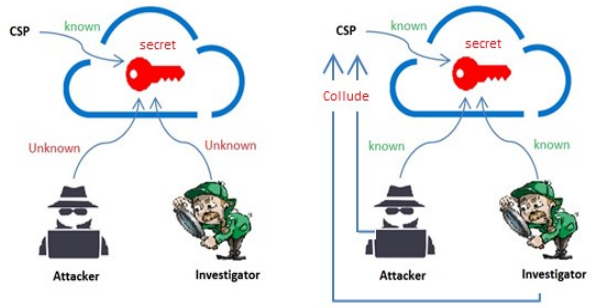


Figure 2. Collusion of Multiple Parties in Cloud

After few days, due to hardware failure, the firm's main storage got affected. A technical team discovers that the physical backup arrangement not working properly for the last few days and had sent many alerts to the technical team that no one observed. The technical team has the responsibility for this work. Thus, the technical team may be motivated to fabricate the firm's logs to be away from the issue.

The firm's consumers' operations, transactions data, personal data, and all relevant finance and insurance statistics are entirely managed by Sagar. Manasi is one of the consumers of cloud services offered by Sagar doesn't have direct access to the logs of their activity which is unfair. She has to be dependent on Sagar who is the service provider. As it is financial data, Manasi initially asked the cloud provider to update the security services on the cloud. Amol who is the competitor of Manasi did a DDOS attack on her application for his personal gains. Imagine that due to non-updated security services Manasi receives complaints related to their transactions for which Amol is responsible. Sagar knows that he has not updated the security patches. Hence, Sagar can collude with the technical team or Amol or forensic investigator to cover up the tracks and be motivated to change the logs before sending them to Manasi. As shown in Figure 2. At first, the lone CSP acquainted with the initial covert, as he is supervising all the cloud activities. However, the hypothesis talks about the untrusted system and CSP can be motivated to conspire through the malevolent invader or forensic investigator en route for tamper with data. To solve this type of problem we did some contributions which are listed in the next section.

Our Contributions: Following are the main contributions of our paper:

a) We developed an access control mechanism for different parties of the cloud while preserving data integrity and confidentiality.
b) We developed a forensic-enabled secure logging scheme in cloud despite multiple-party collusion.
c) We implemented a proof publication method for publishing cloud users' data for forensic investigations in a distributed environment where every authorized peer can check the signs of tampering and verify the eclectic logs in terms of better accuracy and performance.
d) We designed an efficient and cost effective decentralized off-chain log storage mechanism by engaging IPFS to reduce the data packing load of blockchain.

Organization: We have presented the background and the challenges of cloud log forensics in Section 2. Section 3 presents the literature review. The threat model & security properties are deliberated in Section 4. Our proposed technique is presented in Section 5. System setup, security analysis, evaluation results are discussed in Sections 6 and 7 respectively. The conclusion and future scope are provided in Section 8.

2. BACKGROUND

In this segment, we introduce a succinct background and overview of the forensic field, components, and characteristics of the blockchain. In section 3(A), (B), (C), we acquaint the branches of forensics i.e. digital forensics, cloud forensics, and cloud log forensics. In section 3(D), (E) we provide the overview of block chain, IPFS, and their usage to devise a solution to the cloud collusion problem.

A. Digital Forensics (DF)

As per the National Institute of Standards and Technology (NIST) document, DF is a systematic procedure of discovering & interpreting computing information using detection, assembling, investigation, and analysis. The penultimate objective is to maintain the data in some specific way that will be beneficial for the reintegration of old actions chronologically [5]. The information which is existent on machines in the form of images, audios, etc. can act as pieces of potential evidence. This information can help us to investigate the crimes, by providing supportive shreds of evidence. To procure, stock, and examine digital information, robust methods are required in the cloud [1].

B. Cloud Forensics (CF)

The CF is a subdivision of DF. It is firstly presented by author Keyun [28] as an inter-discipline of two popular fields i.e. cloud computing and digital forensics. It deals with and applicable to digital forensics, but the working atmosphere is different. However, this can lead to several defiances for pursuing forensics inside the cloud

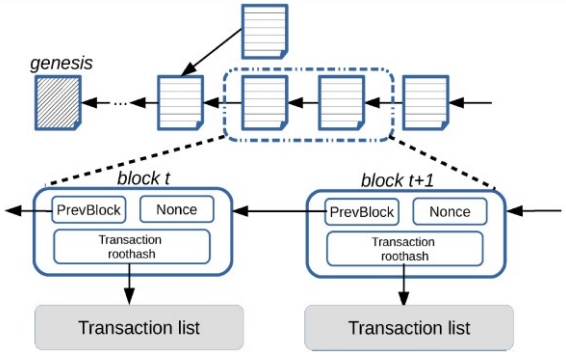


Figure 3. Structure of Blockchain

computing environment [16] [29]. The final aim remains unchanged, reintegration and ensuring the safety of ancient chronological actions [12] [30].

C. Cloud Log Forensics (CLF)

The third branch of forensics is quite challenging to implement since of the distributed nature and less control of consumers over the cloud. This can lead to a conundrum while accessing cloud logs and maintaining the integrity of consumers' behavior logs. In anticipation of getting multiple logs generated within the system [1], all stakeholders of the system are dependent on the CSP to get the logs and as previously pointed out in our proposition, there is no assurance that the cloud provider will make available justifiable and correct logs. Espousal of cloud computing is rising day by day consequences the assaults on the cloud have also been on the rise and there is a dire urgency of implementing log forensics in the cloud [31].

D. Blockchain

In the blockchain, there are three components as shown in Figure 3. The genesis block is the first component of any blockchain. This block states the start of the blockchain. The second component of blockchain is the data block which consists of transaction data, a hash of transaction data, a hash of the previous block, nonce, and timestamp. A hash is a unique value generated from hashing algorithm, e.g. SHA-256. A nonce is a random number that is used to verify the hash. The timestamp is a time of block creation in UTC. The genesis block is also a data block as it stores the above-mentioned information. Miners are some specific nodes that plays important role in the block verification process. The third core component of blockchain is chain i.e. sequence of blocks in a specific order. With the hash and chain of blocks, data is promised immutable. The linking of the data blocks with the previous hash creates a blockchain that assures data integrity [32]. Algorithm-1 from the appendix shows the creation of a single block in the blockchain.

E. IPFS

The Interplanetary File System (IPFS) is a decentralized network that is developed to style the internet faster without harm. In the IPFS system, if one machine goes down, other machines in the network can provide the required data. If we add any document on the IPFS network, it creates a hash value of the document. Other machines in the network can access this document through hash value which is published on the network. IPFS is a versioned file system like Git, it can stock records and track changes by checking versions of those records for any prescribed chronological period. IPFS is essentially a scattered organization that stocks and shares the documents based on content addressing [33].

3. Related Work

The overall goal of this extensive literature survey was to understand the current related work going on cloud forensics field. This section clearly describes the previous work and its limitations and the need for new techniques and methods for cloud forensics. We tried to address secure logging and related problems of the cloud computing domain. Both centralized and decentralized types of systems have been studied. This work was inspired by [12] [19] as they tried to solve the same problem using different methods and techniques. Several research efforts have been undertaken to various extents for doing forensics in the cloud by a multitude of people [31]. The early work [34] encapsulates questions to answers on application logging ranging from enabling logging on to possible sources, setting up safe log transference, and fine-tuning logging formations. With the help of these pieces of advice, logging responsibilities can be completed successfully. A trustworthy and protected transference layer is essential in logging cloud applications. To obtain pieces of prospective evidence, a virtual contract of reliance amongst diverse cloud computing layers is a requisite. However, only guidelines have been given in the present research for how what, and when to do logging. Forensic tools and methods for doing forensics in cloud computing have been studied and evaluated successfully [17]. To gather different application peripheral interfaces (API) and firewall logs the FROST, a forensic enabled gizmo has been established by researchers [14] [17]. But, a strong trust is needed in different layers of the tool. Correspondingly, to deliver several activity logs to cloud consumers securely, novel APIs have been proposed which consider data in different phases like rest, motion, and execution [6]. In [35] Distributed denial of service (DDoS) attack detection mechanism has been implemented using Syslog on the eucalyptus cloud computing platform. The task of inner and outer activities observation by means of transmission capacity and CPU utilization is feasible in eucalyptus cloud computing software [35]. To procure the various logs from network segments, virtual file systems (VFS), and system call interfaces the supervision level over the Infrastructure as a Service is needed [36]. However, this

aforementioned layer may be a residence of weakness for enemies [12] [30]. A lot of forensic experts have put forward the idea to use Trusted Platform Module (TPM) to execute procedures and methods of digital forensics assuming the cloud service provider is a reliable participant in the cloud computing environment [29]. Few researchers were worked on the investigation of logs for several attack recognition [35]. Delegation of log controlling and its supervision instead of building activity log supervision for cloud environment is a price-saving job. Mysterious networks like TOR can be utilized for mockups [37] [38]. To maintain the order of VMs dealings, happened-before relationships are positioned in [41]. However, this relationship is not suitable in each criminal study. Zawoad and Ahsan have established Secure Logging as a Service (SecLaaS) [12] and CLASS [19], integrity safeguarding and verification way with the help of probabilistic data structures in cloud computing which is raising false positives and they are not publicly verifiable.

Cucurull et. al [42] proposed a secure logging method by using Bitcoin to record the history of events on local log chains. Sutton et. al [43] engage graphs to store the information of logs and offload the integrity proof digests on the Bitcoin network for audit trail. Since these methods depend on the Bitcoin blockchain, they cannot scale much by their block size and throughput. Other researchers explored the usage of permissioned blockchain framework i.e. Hyperledger Fabric. Ahmad et al. [44] worked on the database tracking by uploading the changed evidence information on Hyperledger Fabric. Shekhtman et. al [45] store the log files information straight away on the Hyperledger blockchain. Permissioned blockchain has been used for secure logging; however, the implemented methods are not scalable because the scalability parameters like throughput and storage are not undertaken. On the other hand, scalability is the utmost parameter to handle the huge number of log data in real-world secure logging management. SDN and blockchain has been used for secure logging with graphs in [46].

Cloud is a mixture of multifarious virtual systems and their linkages. Thus, it is exposed to cyber-attacks [28] [39] [40]. Currently, establishments can manage to pay for the additionally incurred care of safety and secure logging facilities. Though, a great amount of study is being accompanied in this region, for safeguarding reliable indications and marks in the cloud; but without considering collusion of multiple parties. Building a mechanism to make verifiable pieces of evidence presented to everyone in the system is what requires more research [12]. Trustworthiness and privacy maintenance specifically with logging in a virtual setting like a cloud is still an imperative topic to work upon [16]. Few schemes like SecLaaS and CLASS are based on centralized systems which can be down due to a single point of failure. As much existing work is based on internet protocols like HTTP may get affected due to DDOS attacks which lead to data unavailability. Privacy of the cloud consumers' data is also a major issue. Till now, cloud consumers have to depend on cloud providers to get the logs of their activities. Many of the approaches are assuming logger as a trusted entity.

Above mentioned research works precisely motivated us to work on the coherent readiness of logs; by looking at compelling problems of secure logging in spite of the collusion of multiple parties in a cloud that means no stakeholder of the cloud system is trusted and collusion may happen. Thus, there is a strong need for a scalable, efficient, and provable logging model in the distributed environment. The kind of system needs the proper procedures and forensic practices to be applied. A summary of the most recent and relevant approaches is given on the next page TABLE I. We considered the paper name, publication year, paper objective, service, advantages, and limitations for comparison.

4. THREAT MODEL & SECURITY PROPERTIES

In this section, we define the important notations used in the proposed secure logging model. After that, we narrate the threat model and security properties required in our projected system.

A. Threat Model

1. Confidentiality Violation: The confidentiality violation occurs when the cloud consumers' logs information retrieved by the invaders without consumers' consent. Though some confidentiality violations are unintentional, cloud consumers can suffer from financial losses.

2. Integrity Violation: The integrity violation of cloud consumers' logs is misconduct on their data to damage it. Such misconduct can be achieved by a single or group of entities by colluding with each other.

3. Availability Violation: The availability violation takes place by using unauthorized access to data which makes the cloud consumers' logs unavailable to the actual stakeholder of cloud systems.

4. Privacy Violation: The privacy violation of cloud consumer' logs occur when exuded log files can lead to creating a link or direct identification of cloud consumers' identity and data.

5. Repudiation by CSP: The cloud service provider can deny the logs and the proofs of consumers' actions in cloud computing.

TABLE I. A SUMMARY OF THE MOST RECENT AND RELEVANT APPROACHES FOR SECURE LOGGING

Ref.	Year	Paper Objective	Services	Advantages	Limitations
[12]	2016	Secure Logging as a Service	Centralized	Preservation of Integrity and Confidentiality, Method for publishing and verifying the logs.	The system may get down due to a single point of failure, Suffers from false positives, privacy issues of the cloud users' logs, and the possibility of collusions may lead to data leak. A trusted third party is required for forensic analysis.
[42]	2016	A Bitcoin-based secure logging technique to record the history of events on local log chains.	Decentralized	Provides a method for data auditability.	Since this method depends on the Bitcoin blockchain, it cannot scale much by its block size and throughput. Optimized implementation is required, Required huge amount of gas and transaction fee.
[47]	2017	Graphs to store the information of logs and offload the integrity proof digests on Bitcoin network for audit trail.	Decentralized	Maintenance of Integrity proof using graph approach.	Logging data transactions will be very huge which can place a storage burden on the blockchain. Off-chain storage methods are not used, Since this method depends on the Bitcoin blockchain, it cannot scale much by its block size and throughput. Optimized implementation is required. Required a huge amount of gas and transaction fee.
[43]	2017	Database operations tracking using graphs and Bitcoin network	Decentralized	Discussed the Database operations tracking and its security.	They have shown the viability of logging audit method using permissioned blockchain; however, the implemented methods are not scalable because the scalability parameters like throughput and storage are not undertaken, Required huge amount of gas and transaction fee.
[19]	2018	To develop cloud log assurance soundness and secrecy	Centralized	Preservation of log integrity, confidentiality, and privacy of consumers' data through encryption.	The storage is completely centralized and thus may suffer from a single point of failure which leads to data unavailability; Use of probabilistic data structures which may give wrong results. The log inspection process was very tedious.
[46]	2018	Secure storage of the log files on the Hyperledger Fabric.	Decentralized	Metadata is stored on the on-chain storage of blockchain.	Permissioned blockchain has been used for secure logging; however, the implemented methods are not scalable because the scalability parameters like throughput and storage are not undertaken.

[45]	2018	Data Integrity Validation	Decentralized	Validation of data integrity using blockchain is valuable.	Only data validation method is presented. The method works on few assumptions. Data Storage on a blockchain is very expensive. IPFS data storage is suggested in the future scope of the paper.
[48]	2019	Challenges and Opportunities in Distributed Cloud Storage Forensics	Decentralized	Explain Cloud Storage Forensics using STORJ case study.	The guidelines have been given for cloud storage forensics but implementation details are missing.
[49]	2020	Implementation for secure Logging Service	Decentralized	Isolation of logging service, reconstruction of event timeline, Resiliency to log manipulation, privacy, and data confidentiality.	Vulnerable to DDOS attacks. Only the on-chain storage layer is discussed; Data packing load is very high on a blockchain which is uneconomical solution.
[50]	2020	concrete transparency and compliance architecture	Centralized	List of data processing events and their sharing and compliance. Special Architecture for compliance checking.	A policy vocabulary and language has been given for logging. Integration of it is not described anywhere. Does not directly deal with the integrity and privacy of the data. Dashboard creates a new place for attackers.
[51]	2020	Network Log Management	Decentralized	This technique provides transparency, and preserves confidentiality & data accountability.	Not presented off-chain storage of records and also provided only on-chain storage for pre forensic data files. Solution is not cost effective.
[52]	2020	Method for collection of evidence and Preservation of Provenance in Cloud	Decentralized	Authentication, Encryption and optimal key generation algorithms have been implemented.	Not presented off-chain storage for evidences thus it is uneconomical solution, Logical graph of evidence visualizations is not up to the mark.

6. Repudiation by CSC: A cloud consumer may assert that the said consumers' log information is not his own and can pin the ownership on some other user considering the fusion of cloud data.

B. Security Properties

1. Admissibility (A): Prospective pieces of proof have to be safeguarded, for viable enough to be admissible in the law court for criminal examinations.

2. Correctness (C1): A Correctness possession upholds the eminence of restricted to faults, errors & conformism towards the acceptance of the proof for criminal inquiries.

3. Tamper Resistance (TR): TR is the property of security which quantifies the confrontation to interfering of cloud consumers' log files or whichever reliable proofs on cloud computing.

4. Confidentiality (C2): This property is a state of being kept secret of the cloud consumers' logs to authorized entities only, while others cannot ensure secured access.

5. Verifiability (V): Every single secured proof of activity must be verifiable while considering virtuous precision and performance.

TABLE II. NOTATIONS

Notation	***Meaning***	***Notation***	***Meaning***
CSP	Cloud Service Provider	H(M)	Hash Function
CSC	Cloud Service Consumer	E	Events
CFI	Cloud Forensics Investigator	O	Objects
PK_{CSP}	Public Key of CSP	L	Activity LogInfo
SK_{CSP}	Private Key of CSP	R	Event Request
PK_{CSC}	Public Key of CSC	D = {accept, reject}	Decision
SK_{CSC}	Private Key of CSC	$M = R^I$	Request Queue
PK_{CFI}	Public Key of CFI	$N = D^I$	Decision Queue
SK_{CFI}	Private Key of CFI		

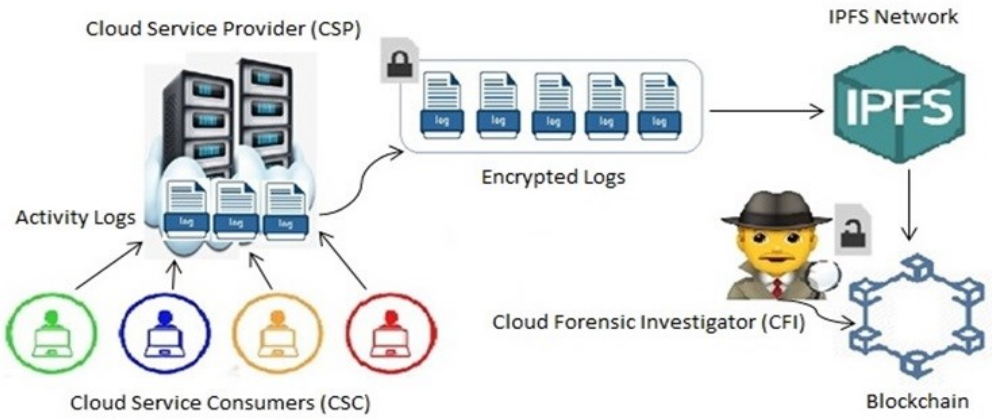


Figure 4. Proposed Blockchain Driven Secure and Efficient Cloud Logging

5. PROPOESED TECHNIQUE

Cloud service consumers (CSCs) make use of the cloud daily, for the consumer as well as business activities. For every single action happening in the cloud computing environment, we capture produced action occurrence into the log file. Afterward, we collect these logs, encrypt them and accumulate them on the IPFS network to reduce the storing load of the blockchain. Finally, the highest hash will get stored on to the blockchain network which is essentially publicly verifiable as shown in Figure 4. The flow of the scheme is shown in Figure 5. We assume that, CSP is honest for collecting the logs, publishing the proof of past logs, but during an investigation they can collude with other parties as discussed in hypothesis. We also assume that, in the Service Level Agreements (SLA), CSP is agreed that he will be responsible for logging all the events.

A. System Model

Definitions, Notations, and Assumptions: We take for granted the nature of the entire cloud stakeholders is untrusted. PK_{CSP}, $SK_{CSP,}$ $PK_{CSC,}$ $SK_{CSC,}$ $PK_{CFI,}$ and SK_{CFI} *are* public and private keys of CSP, CSC, and CFI respectively. H (D): This generates the hash for a given data. $Encypt_{PK}$ (D): This is the encipherment of data D using the public key PK. $Sign_{SK}$ (D): This signifies the signature over data D using the private key SK. D1 || D2: shows the consistency between two data/ messages/proofs.

B. Proof Creation

Cloud service consumers $CSC = \{CSC_1, CSC_2, CSC_3 \dots CSC_n\}$ send an event $E = \{e_1, e_2, e_3 \dots e_n\}$ for requesting R (which will be initially resides in the queue, $M \rightarrow R^I$) to data objects $O = \{o_1, o_2, o_3 \dots o_n\}$ in cloud to perform various actions for which system captures the activity logs $L = \{l_1, l_2, l_3 \dots l_n\}$.

a) Log File Creation

The scheme catches every incident that happened in this environment into the log file LF, $E_i \rightarrow L_i$ in which i is starting from 1 for the very first action. The scheme will be incrementally captured the events and creates a log file over the time T. Index i is assigned to each event E. Thus the scheme is maintaining the sequence of events $L.Seq\{L_i, L_{i+1}, L_{i+2}, \dots L_{i+n}\} \rightarrow PS$; in which PS is the proof of sequence where $i \leq i+1$.

b) Partial Proof Generation

In the second step of proof creation, the scheme is storing m number of events in one file. This will create a partial proof of events in the system. Partial Proof Generation is represented as $LF.insert \{E_1, E_2, E_3 \dots E_m\} \rightarrow PP1$; where PP1 is the partial proof of events E_1 to E_m and so on for remaining events.

c) Partial Proof Encryption

In the third step, considering data privacy concerns of cloud customers, the scheme encrypts the partial proofs using CSCs' public key, $Encrypt_{CSC}(PP1) \rightarrow EP1$ where EP is the Encrypted Proof which is signed and as per its sequence number i. Thus only CSC will be able to decrypt the logs which will preserve their privacy. The creation of EP1 and other partial proofs accumulatively will give P_n. it will get added into the IPFS network. Therefore, EP1 || EP2, EP2 || EP3 and the rest similar for all the proofs.

d) Add Encrypted Proofs on IPFS Network

In the end, all the proofs from $P_1 \dots P_i, P_{i+1}, P_{i+2} \dots P_n$ which are reliable and scrambled are pushed on an interplanetary file system network. For insertion of a piece of proof, interplanetary file system returns a hash which CSCs' may use for auxiliary checking in future. Command to add proof on IPFS is *$ ipfs add P.log*. At the end of the day, for one IP, the scheme uploads the final hash or root hash of all the proofs onto the blockchain. Algorithm-2 shows

that how the system starts the IPFS daemons and when it is ready system uploads the log files into the IPFS. IPFS return the content hash of the file. With the help of geth console; after completion of mining process content hash get offloaded on blockchain which results the transaction hash.

Algorithm-2: Storing log hash in ethereum blockchain

```
Input: log-hash from IPFS
Output: Transaction hash
Begin
  Start all Daemons D in the IPFS nodes M
  if D == Ready then
  Upload the file in one of the IPFS nodes
  M (1)
  Get the log-hash from IPFS node M (1)
  end if
  Start geth console in one of the Ethereum
  nodes N (1)
  if account == locked then
    Unlock the private account
  end if
  Start the mining process
  while mining == True do
      Make a transaction to store the log-hash
      Wait for the block to get mined
      Get transaction hash
  end while
End
```

C. *Eclectic Proof Verification*

The second half of our scheme offers an eclectic verification of the integrity of the potential items as well as trustworthy proofs. In our work, every block consists of four things. 1. TS is the block creation timestamps; 2. PH stand for the previous hash is a hash address that locates the preceding block. 3. L_Root is the topmost hash that comes from the hash of all the transactions of the logs in a block. 4. Nonce is the number to the block which is an arbitrary number and only uses once. It also maintains the difficulty level restrictions.

In our proposed work, the scheme is built to provide two-step integrity verification. It comprises engagement of Interplanetary File System and blockchain. IPFS has the feature of versioning the data. The first phase of the scheme stocks the encrypted log files into the IPFS as shown in Algorithm-2. If some adversary tries to tamper it generates the different modified copy by keeping the actual log file unaltered. Therefore, the scheme can definitely trail the hashes of altered and unaltered files. In the same situation, the logs are encrypted with CSCs public key so only authorized CSC can decrypt that log with forensic investigators at the time of the investigation. Thus, the privacy of the cloud consumers gets preserved. As shown in Figure 6, in the subsequent period the scheme is pushing the topmost combined log hash L_Root to the blockchain. Even minuscule alterations in the log data will consequence the completely dissimilar hash. Therefore, eventually succeeding hash data of the hierarchy become altered. In this phase, the cloud forensic investigator is authorized to do the verification in the cloud. When CSC wants to verify the integrity of their activity logs, they send a request to CFI. After receiving the request CFI asks CSP to give the logs of the period in which malicious activity happened.

When any malicious activity happens cloud forensic investigator randomly take few logs L = $\{l_1,\ldots, l_k\}$ for k logs, where $1 \leq l_i \leq n$. First, Cloud Forensic Investigator takes MHT root from IPFS and blockchain. CSP generates the new hash for asked logs. Here, the eclectic verification process starts which is the novelty of our scheme. After getting the root value from CSP, CFI matches that root value with blockchain root. If that matches all logs stored in the cloud system are not tampered otherwise logs get tampered with. And therefore forensic investigators can release decision D in terms of either positive or negative. Decision D will depict the reliability of proofs. We signify the queue of results by N -> D^I. Hence, CFI can smoothly confirm the truthfulness of log files & other pieces of evidence. We demonstrate the safety of the proposed scheme with below-mentioned security properties. Algorithm-3 describes that how we system is retrieving L-Root log-hash from blockchain and Algorithm-4 shows the log validation function so as to get accurate transactions.

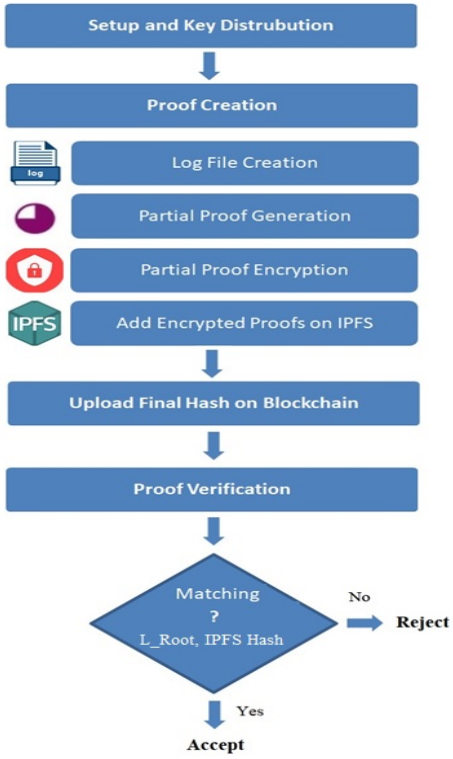


Figure 5. Flowchart for working of a BlockSLaaS

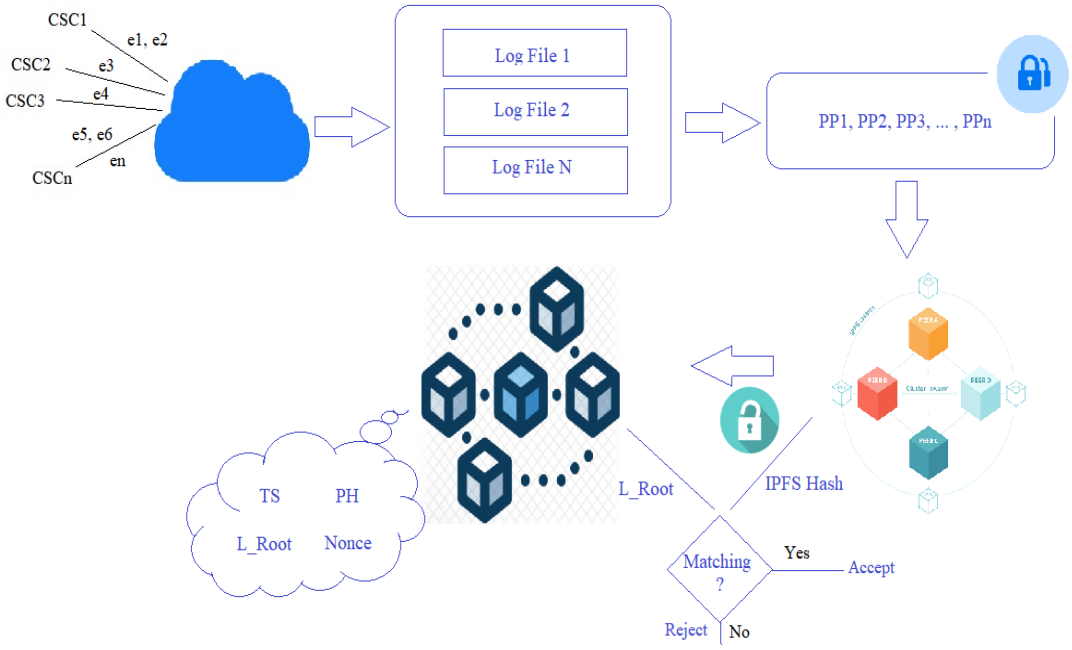


Figure 6. Secure Logging and Eclectic Proof Verification

Algorithm-3: Retrieving log-hash from blockchain

Input: Contract address, Blockchain nodes N
Output: log-hash stored in blockchain
Begin
Start geth console in one of the Ethereum nodes N (1)
if account == locked **then**
Unlock the private account
end if
Start the mining process
while mining == True **do**
if N (1) requires log-hash **then**
Make a transaction to retrieve log-hash by submitting the contract address
else
Send bin, abi and the contract address to other peers
Call the function which retrieves the log-hash
end if
Wait for the block to get mined
Return log-hash
end while
End

Algorithm-4: Validating log-hash stored in blockchain

Input: Transaction hash Tx, IPFS log-hash i
Output: Validation Result
Begin
Start geth console
Start mining process
if mining == True **then**
receipt ← Retrieves the transaction receipt of Tx from blockchain
endif
h ← receipt [input]
if i == h
return true
else
return false
end if
End

6. SECURITY ANALYSIS

In this paper, as shown in Figure 1. Multi-stakeholder collusion model, there are basically three entities are involved: Cloud Service Provider, Cloud Service Consumer, and Cloud Forensics Investigator. All above mentioned entities can do malicious activities individually or can collude with each other for their personal benefits as mentioned in hypothesis. As a host cloud service providers have complete control over the logs stored in interplanetary File System. Thus, CSP can be able to add / modify / delete the activity logs. After obtaining logs from CSP, Cloud Forensic Investigator may be motivated to modify the log data beforehand providing in front of court. Thus, we presented the interference stopping scheme using block chain and IPFS. Violation of the any property by CSC, CFI and CSP, as mentioned in threat model, can be reveal in the verification process.

In this segment, we converse on how our novel scheme guarantees all the security properties and essential to defend against collusion between CSP, CSC & CFI. In Figure 11, we have shown the multiple-party collusion model, the variety of probable assaults which can take place over cloud & the necessity of security property. In this, we symbolize a truthful cloud service provider as P, a fraudulent cloud service provider as $\bar{P}$, a truthful cloud consumer as C, a fraudulent cloud user as $\bar{C}$, a truthful forensic investigator as I, and a fraudulent forensic investigator as $\bar{I}$.

(Correctness, Tamper Resistance, Confidentiality):
Security is always an important concern in a cloud-based system where sensitive data should maintain in an appropriate way to tackle different attacks and guarantee cloud service consumers' data integrity and confidentiality. Assume that a malicious insider or external entity can get into the IPFS data storage, such a malicious insider or entity struggles for getting the sensitive information using BlockSLaaS. Because, in our proposed scheme, all log file records $L = \{l_1, l_2, l_3 \dots l_n\}$ are hashed using a one-way hash function i.e. H ($l_1, l_2, l_3 \dots l_n$) where $l_i \in L$. One way hash functions are irreversible and thus, difficult to find the content of log file. Also, the log files are enciphered with the public key of CSC, PK_{CSC} for uploading to the IPFS. To get these trustworthy logs data, any attackers should know the secret key PK_{CSC} of a particular CSC for decryption operation. Thus, through BlockSLaaS, it's very hard for the malicious insider or outside invader to guess/find the private key PK_{CSC} to decrypt and obtain the sensitive logs data. Only authorized users with the keys can access the log files. It ensures confidentiality property.

In the Interplanetary File System, activity log files LF are get chunked and saved on various storage machines. Now, the log files are already encrypted using the RSA algorithm and stored in IPFS storage machines. Other machines of the system can only see a part of scrambled data, $Encrypt_{CSC}$ (PP1) -> EP1 and cannot obtain any useful insights from the log files. The root of these proof files $P_1 \dots P_i$, P_{i+1}, $P_{i+2} \dots P_n$ are stored on the blockchain network nodes. This is computationally impractical to alter the log information on blockchain. If adversary tries to tamper t block, hash value of t+1 block will automatically change and so on. Thus, our scheme ensures tamper resistance property. Authentic stakeholders of the cloud system can get access to complete log records and retrieve correct logs using our system which provides the access control mechanism by preserving integrity and confidentiality. After applying one way function H (L) to logs; log contents will be irreversible. As shown in Figure 6, adversary cannot tamper the logs; and checks for the proof hash available with blockchain. Forensic Investigator will get correct logs through our system and thus ensures correctness property.

(Verifiability, Admissibility):
In our proposed scheme, Cloud Service Provider constructs the Merkle Hash Tree (MHT) with the hash of different leaf and intermediate nodes using IPFS log files and finally creates the root node say L_Root and stores it into the blockchain network. If any adversary tries to alter the IPFS logs, it generates a new copy of that log file, say IPFS_Root. It happens due to the versioning nature of IPFS. Still, if someone tampers the logs data MHT hash values will get change, and subsequently, the root of MHT will get changed. So, the root of the IPFS logs file and associated blockchain root will either match or not. If L_Root = IPFS_Root then content is not changed otherwise got tampered. Using this principle system can easily verify that the IPFS logs are authentic or not. Thus, the scheme ensures verifiability property as shown in Algorithm-4. Each and every secure evidence is verifiable while considering good accuracy and performance. Potential items of evidence i.e. logging proofs $P_1 \dots P_i$, P_{i+1}, $P_{i+2} \dots P_n$ are got secured using our blockchain-driven secure logging scheme in such a way that they are viable enough to be acceptable for law enforcement. Thus, it guarantees the admissibility property. With these security properties, the significance of security properties can be seen as stated in the threat model. The BlockSLaaS scheme empowers the security belongings by conserving & confirming integrity, the confidentiality of responsible indications and marks in the cloud system, hence making the cloud more secure & forensic friendly.

7. IMPLEMENTATION AND RESULTS

For the implementation purpose, we used the open-source OpenStack software to form a cloud with I9 Machine, 32GB Random Access Memory, 2TB Hard Disk, Linux 18 versioned OS, and Oracle VirtualBox 6.1.18. RSA public-key cryptography algorithm and Secure Hash Algorithm of 256 bits, a hash function has used for encipherment and hash creation correspondingly.

Figure 7 depicts the integrity and signature proof verification of logs through the proposed BlockSLaaS scheme. The overall time required for these operations is also calculated and shown in the same figure. The time taken to complete both the verification processes are also shown. X and Y-axis display the number of actions in a chiliad and the time required to finish the proof verification in seconds respectively. Our results indicate that the projected novel method is efficient. So, this BlockSLaaS setup on OpenStack open-source cloud computing platform and its results reflect the feasibility for giving safety to reliable pieces of evidence and marks in the cloud.

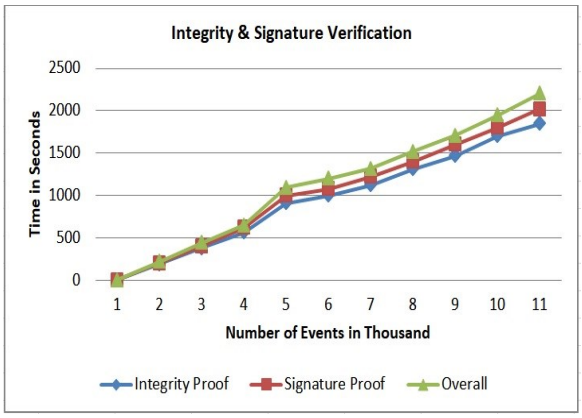
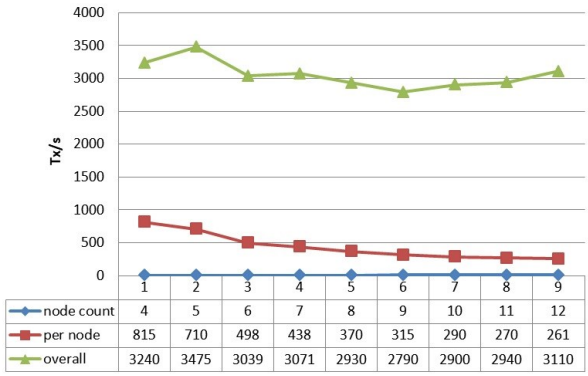


Figure 7. Integrity and Signature Proof Verification



	1	2	3	4	5	6	7	8	9
node count	4	5	6	7	8	9	10	11	12
per node	815	710	498	438	370	315	290	270	261
overall	3240	3475	3039	3071	2930	2790	2900	2940	3110

Figure 8. Transaction per second for varying node count

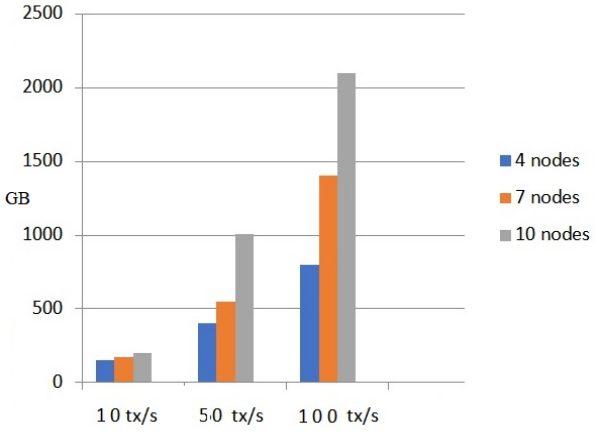


Figure 9. Storage requirements for varying average throughput and node count

In this implementation ethereum testnet blockchain was used to serve the purpose. Figure 8 shows the transactions per second completed for varying node count. It is showing that the system is performing fewer transactions per second after increasing few nodes and then it is stable. Overall transaction per second reflects the same analysis but transaction per second increases slightly at the end. Figure 9 shows the storage requirement for varying average throughput from 10tx/s to 100tx/s and node count from 4 to 7. Integration of IPFS is with blockchain greatly reducing the storage burden of blockchain and thus its efficient and cost-effective scheme. The above implementation results and security analysis proves that our system is efficient & secure.

Sample setter function statistics
Without IPFS
Gas used 12355552
Transaction cost 12355552 gas
Data size = 37kb
Mining time = 59 sec

With IPFS
Gas used 75896
Transaction cost 75896
Data size = 37kb
Mining time = 11 sec

We measured the time required to send the log data to IPFS and then on Ethereum. We also calculated the transaction validation time and the total time required for the whole execution process. The analysis has been done for 20 times; 4 times each for the following number of files.

TABLE III. EXECUTION TIME OF PROOF CREATION AND VERIFICATION IN SECONDS FOR VARYING NO. OF LOG FILES

No. of Log Files	IPFS	Etherium Sent	Etherium Confirmed	Total
5	2.85	1.284	32.125	36.259
10	1.323	1.191	49.411	51.925
15	3.774	1.273	25.992	31.039
20	2.124	1.194	27.004	30.322
30	1.395	1.282	30.939	33.616

Above mentioned result is an average execution time in seconds. The TABLE III. depicts the etherium confirmation time is high in the validation process by varying number of log files.

We did some experiments to check the performance of the BlockSLaaS scheme by uploading, reading and downloading the file by varying the node count. The results are shown in Figure 10, 13 and 14. The log files were uploaded using one machine and accessed by another two machines. At each step we increased the number of nodes in the BlockSLaaS as shown in figure. We observed that if the number of increases the transaction time slightly increases. This signifies that the extra overhead does not degrade the overall performance of the system. Thus, our system nature is scalable. The logging process generates a tremendous amount of logs on daily basis. Inspecting these logs for forensic purposes is a hectic process. To overcome this problem we have developed a consumer activity log visualization mechanism on top of IPFS using BlockLaaS. Figure 12. shows a sample visualization of few events that happened in our system. These events can be retrieved easily through the content hash of a log file. This will be a great help for forensic investigators in identifying the suspects early stages.

We did a comparison with the well know techniques CFLOG [51] and DFeSB [53] in terms of complete response time, proof insertion time and proof verification time as shown in TABLE IV. For inserting the evidence CFLOG and DFeSB taken 88.5 ms and 63 ms time respectively considering 100 users on the system. BlockSLaaS get the response from the system in 38.3 ms only which is very less time as compare to other two. Evidence insertion time for CFLOG and DFeSB was 71 ms and 43.5 ms respectively. BlockSLaaS requires only 29.7 ms as it is based on off-chain data storage.

TABLE IV. PERFORMANCE COMPARISON WITH EXISTING TECHNIQUES

Performance Metric (ms)	CFLOG	DFeSB	BlockSLaaS
Response Time	88.5	63	**38.3**
Proof Insertion Time	71	43.5	**29.7**
Proof Verification Time	70	42.8	**26.3**

Verification time in BlockSLaaS is 26.3 which is also very less as compare to CFLOG and DFeSB for 10 users. The major impacts on results were due to integration of IPFS. Thus, our system performs better in all the aspects. Our scheme reduced block chain storage burden up to large extent. Thus, our proposed system is more efficient than existing systems. performance Comparison with existing techniques.

TABLE V. COMPARISON WITH EXISTING TECHNIQUES BASED ON SECURITY AND PERFORMANCE FEATURES

Ref.	Year	VL	I	C	A	P	O	D	V	S
[19]	2018	×	✓	✓	×	×	×	×	✓	×
[46]	2018	×	✓	✓	×	×	×	✓	✓	×
[48]	2018	×	✓	×	×	✓	×	✓	✓	×
[53]	2018	✓	✓	✓	✓	✓	×	✓	✓	✓
[50]	2019	×	✓	×	×	×	✓	✓	✓	✓
[44]	2020	×	✓	✓	✓	✓	×	✓	✓	×
[47]	2020	×	✓	×	✓	×	×	×	✓	×
[49]	2020	×	✓	✓	✓	×	×	✓	✓	×
[51]	2020	×	✓	✓	✓	✓	×	✓	✓	×
Our	2021	✓	✓	✓	✓	✓	✓	✓	✓	✓

VL- Visualization I - Integrity, C - Confidentiality, A - Availability, P - Privacy, O - Off-Chain Storage, D - Decentralized, V - Verifiability, S - Scalability

In this section, we have shown TABLE V. comparative analysis of most recent and relevant approaches used for secure logging. Most of the above-mentioned schemes are not scalable as they are centralized. Very few authors used off-chain data storage with blockchain. The proposed model addressed the limitations of existing work. We are the first to develop and point out the usefulness of log activity visualization for calm forensic analysis. Our scheme facilitates the secure, efficient, provable model for cloud forensics. The above analysis shows that our scheme is performing better as compared to existing techniques and methods.

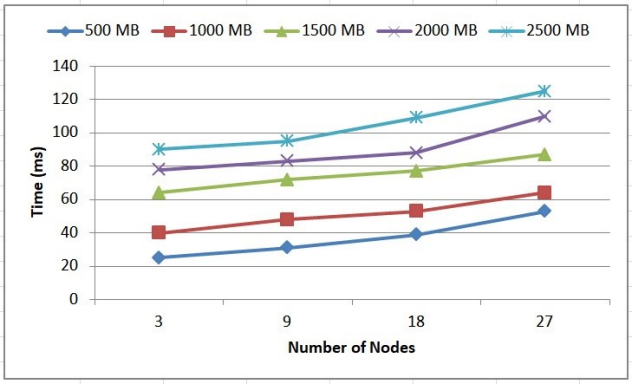


Figure 10. Time required uploading the file

Is Honest?			Notation	Possible Attacks	Required Security Properties
CSP	CSC	CFI			
Y	Y	Y	P C I	Attack free	None
N	Y	Y	$\overline{P}$ C I	Consumers activity disclosure from logs	C2
Y	N	Y	P $\overline{C}$ I	Other consumers' log recovery from proofs	C2
Y	Y	N	P C $\overline{I}$	Add, modify, delete logs	C1, TR, V, A
Y	N	N	P $\overline{C}$ $\overline{I}$	Add, update, delete logs, other consumers' log recovery	C1, C2, TR, V, A
N	Y	N	$\overline{P}$ C $\overline{I}$	Add, modify, delete logs, repudiate proofs, and disclose consumer activity	C1, C2, TR, V, A
N	N	Y	$\overline{P}$ $\overline{C}$ I	Add, modify, delete logs, repudiate proofs, other consumers' log recovery and consumers' activity disclosure	C1, C2, TR, V, A
N	N	N	$\overline{P}$ $\overline{C}$ $\overline{I}$	Add, modify, delete logs, repudiate proofs, other consumers' log recovery and consumers' activity disclosure	C1, C2, TR, V, A

Figure 11. A Cloud Stakeholders Collusion, Possible Attacks and Required Security Properties

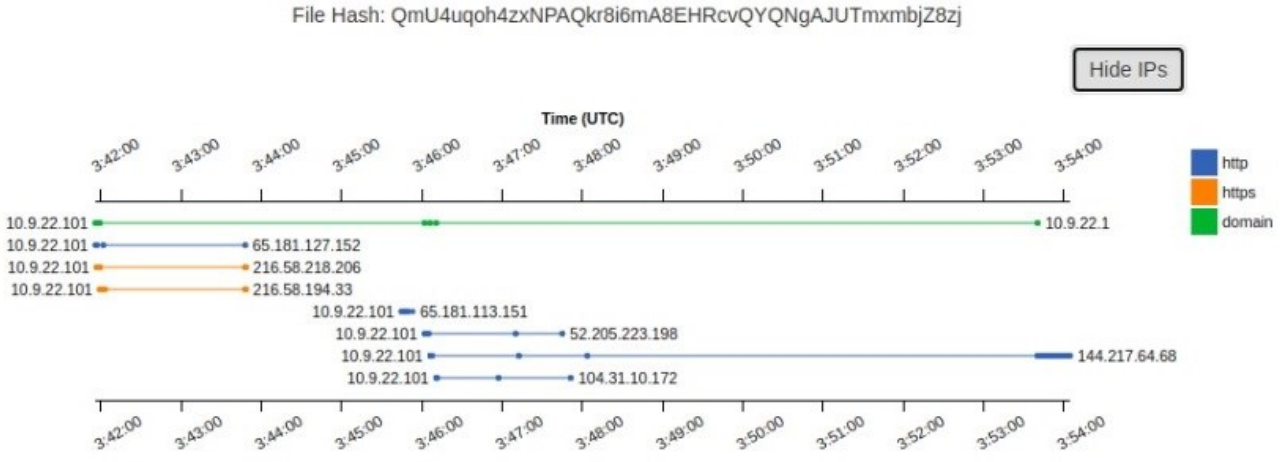


Figure 12. Forensic Data Visualization on top of IPFS in BlockSLaaS

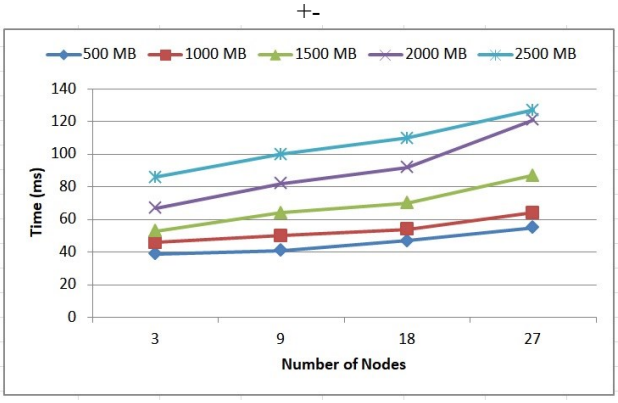


Figure 13. Time required reading the file

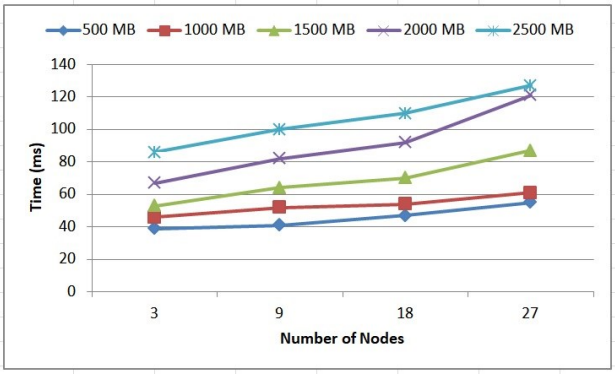


Figure 14. Time required downloading the file

8. Conclusion and Future Work

In this paper, we introduced BlockSLaaS, a blockchain-driven efficient and tamper-proof distributed log storage model while preserving the confidentiality of cloud consumers' logs. Unlike other centralized techniques, the proposed method integrated the IPFS with a fully decentralized blockchain that takes control from a single authority and provides a fair service. The IPFS along with smart contracts preserves an immutable record of cloud consumers' activities on the blockchain. With this scheme only authorized stakeholders can check the historical records stored in the blockchain. The proposed scheme is resilient as no malicious stakeholder of the cloud system can alter cloud consumers' logs despite the collusion of multiple parties.

Our analysis of the number of transactions per second, storage requirements, uploading, reading, and downloading of log files for varying node count and file size validates the viability of the proposed scheme. This work compared the proposed method with nine existing systems based on 9 security and performance features. The BlockSLaaS response, proof insertion, and proof verification times are 38.3, 29.7, and 26.3 milliseconds respectively which beat existing CFLOG and DFeSB methods. Our method adds security which enables a transparent system to obey the audit requirements. The visualization mechanism on top of IPFS gives better readability to the log evidence while doing forensic investigations. A comprehensive performance evaluation shows that our scheme is scalable, efficient, and assures high performance than existing schemes.

As future work, the evidence visualization techniques can be strengthened by implementing a better user interface and mapping logical shreds of evidence. Machine learning algorithms can be explored for the same. Attribute and role-based encryption techniques can be applied for better access control management over forensic data. The current scheme is completely based on IPFS so the drawbacks of IPFS need to be taken care of and other off-chain data storage models can be explored.

References

[1] P.Melland, T.Grance, "Nist Cloud Computing Forensic Science Challenges," NIST Cloud Computing Forensic Science Working Group, Information Technology Laboratory, Draft NISTIR 8006, June 2014.

[2] Kolhar, M., Abu-Alhaj, M. M., & El-atty, S. M. A. (2017). Cloud data auditing techniques with a focus on privacy and security. IEEE Security and Privacy, 15(1), 42–51.

[3] Nasscom, "India's cloud market to cross 7 billion dollar by 2022", https://economictimes.indiatimes.com/tech/internet/indias-cloud-market-to-cross-7-billion-by-2022-nasscom/articleshow/68689359.cms. [Accessed May 08, 2020].

[4] MRM, "Market Research Media, Global Cloud Computing Market Forecast 2019-2024", https://marketresearchmedia.com/global-cloud-computing-market/ [Accessed May 08, 2020].

[5] P. J. Sun, "Privacy Protection and Data Security in Cloud Computing: A Survey, Challenges, and Solutions," in IEEE Access, vol. 7, pp. 147420-147452, 2019, doi: 10.1109/ACCESS.2019.2946185.

[6] D. Birk and C. Wegener, "Technical issues of forensic investigations in cloud computing environments," in SADFE. IEEE, 2011, pp. 1-10

[7] M. Balduzzi, J. Zaddach, D. Balzarotti, E. Kirda, and S. Loureiro, "A security analysis of amazon's elastic compute cloud service," in Symposium on Applied Computing. ACM, 2012, pp. 1427-1434.

[8] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, vol. 34, no. 1, pp. 1\textendash11, 2011.

[9] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," Future Generation Computer Systems, vol. 28, no. 3, pp. 583\textendash592, 2012.

[10] Tang, J., Cui, Y., Li, Q., Ren, K., Liu, J., & Buyya, R. (2016). Ensuring security and privacy preservation for cloud data services. ACM Computing Surveys, 49(1), 1–39.

[11] Tari, Z. (2014). Security and privacy in cloud computing. IEEE Cloud Computing, 1(1), 54–57.

[12] S. Zawoad, A. Dutta, and R. Hasan, Towards Building Forensics Enabled Cloud through Secure Logging-as-a-Service, IEEE Transactions on Dependable and Secure Computing, preprint, \DOI:10.1109/TDSC.2015.2482484. 13(2):148-162 (2016)

[13] K. Kent and M. Souppaya, "Guide to computer security log management," Tech. Rep. 800-92, NIST Special Publication, 2006.

[14] J. Dykstra, A Sherman, Understanding Issues in Cloud Forensics: Two Hypothetical Case Studies Cyber Defense Lab, Department of CSEE, University of Maryland, Baltimore County (UMBC).

[15] Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. Future Generation Computer Systems, 79, 849–861.

[16] S. Zawoad and R. Hasan, Cloud Forensics: A Meta-Study of Challenges, Approaches, and Open Problems arXiv: 1302.6312v1 [cs.DC] 26 Feb (2013)

[17] J. Dykstra and A. Sherman, Acquiring forensic evidence from infrastructure-as-aservice cloud computing: Exploring and evaluating tools, trust, and techniques, Journal of Digital Investigation 9 S90-S98, DOI: 10.1016/j.diin.2012.05.001 (2012).

[18] Wang, B., Li, B., & Li, H. (2015). Panda: public auditing for shared data with efficient user revocation in the cloud. IEEE Transactions on Services Computing, 8(1), 92–106.

[19] M. A. M. Ahsan et al., "CLASS: Cloud Log Assuring Soundness and Secrecy Scheme for Cloud Forensics," in IEEE Transactions on Sustainable Computing, doi: 10.1109/TSUSC.2018.2833502.

[20] Wang, Q., Wang, C., Ren, K., Lou, W., & Li, J. (2011). Enabling public auditability and data dynamics for storage security in cloud computing. IEEE Transactions on Parallel and Distributed Systems, 22(5), 847–859.

[21] Zhu, Y., Ahn, G. J., Hu, H., Yau, S. S., An, H. G., & Hu, C. J.(2013). Dynamic audit services for outsourced storages in clouds. IEEE Transactions on Services Computing, 6(2), 27–238.

[22] Jiang, T., Chen, X., & Ma, J. (2016). Public integrity auditing for shared dynamic cloud data with group user revocation. IEEE Transactions on Computers, 65(8), 2363–2373.

[23] Tian, H., Nan, F., Jiang, H., Chang, C. C., Ning, J., and Huang, Y. (2019). Public auditing for shared cloud data with efficient and secure group management. Information Sciences, 472, 107–125.

[24] Hao, Z., Zhong, S., & Yu, N. (2011). A privacy-preserving remote data integrity checking protocol with data dynamics and public verifiability. IEEE Transactions on Knowledge and Data Engineering, 23(9), 1432–1437.

[25] Wang, C., Chow, S. S. M., Wang, Q., Ren, K., & Lou, W. (2013). Privacy-preserving public auditing for secure cloud storage. IEEE Transactions on Computers, 62(2), 362–375.

[26] Rane S., Dixit A. (2019) BlockSLaaS: Blockchain Assisted Secure Logging-as-a-Service for Cloud Forensics. In: Nandi S., Jinwala D., Singh V., Laxmi V., Gaur M., Faruki P. (eds) Security and Privacy. ISEA-ISAP 2019. Communications in Computer and Information Science, vol 939. Springer, Singapore.

[27] Sagar Rane, Sanjeev Wagh and Arati Dixit (2020) Design of Forensic Enabled Secure Logging, in Proceedings of the 21st International Conference on Distributed Computing and Networking, Article No.: 61, Pages 1, DOI: https://doi.org/10.1145/3369740.3373803

[28] Keyun Ruan, Jeo Carthy, Tahar Kechadi, Mark Crosbie ,Cloud Forensics Chapter 2, Advances in Digital Forensics.

[29] S. Thorpe and I. Ray, "Detecting temporal inconsistency in virtual machine activity timelines." Journal of Information Assurance and Security, vol. 7, no. 1, pp. 24-31, 2012.

[30] S. Thorpe, I. Ray, T. Grandison, A. Barbir, and R. France. 2013b. Hypervisor event logs as a source of consistent virtual machine evidence for forensic cloud investigations. In Data and Applications Security and Privacy XXVII. Springer Berlin Heidelberg. 97-112.

[31] S Khan, A Gani, and A Wahab et al., Cloud Log Forensics: Foundations, State of the Art, and Future Directions ACM Computing Surveys, Vol. 49, No. 1, Article 7, \DOI:http://dx.doi.org/10.1145/2906149 (2016).

[32] Satoshi Nakamoto, Bitcoin: A peer-to-peer electronic cash system. Consulted, 1:2012, 2008.

[33] Juan Benet, "IPFS - Content Addressed, Versioned, P2P File System draft 3, arXiv:1407.3561v1, July (2014)"

[34] R. Marty, Cloud application logging for forensics, Proc. of the 2011 ACM Symposium on Applied Computing (SAC11), Taichung, Taiwan. ACM, March 2011, pp. 178-184.

[35] Z. Zafarullah, F. Anwar, and Z. Anwar, "Digital forensics for eucalyptus," in FIT. IEEE, 2011, pp. 110-116.

[36] A. Patrascu and V.-V. Patriciu, "Logging system for cloud computing forensic environments," Journal of Control Engineering and Applied Informatics, vol. 16, no. 1, pp. 80\textendash88, 2014.

[37] Dıaz, M., Martı´n, C., & Rubio, B. (2016). State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. Journal of Network and Computer Applications, 67, 99–117.

[38] I. Ray, K. Belyaev, M. Strizhov, D. Mulamba, and M. Rajaram, Secure logging as a service delegating log management to the cloud, IEEE Syst. J., vol.7,no. 2,pp.323-334, Jun. (2013)

[39] A. Khajeh-Hosseini, D. Greenwood, and I. Sommerville, Cloud migration: A case study of migrating an enterprise it system to iaas, in proceedings of the 3rd International Conference on Cloud Computing (CLOUD) IEEE, 2010,pp. 450-457.

[40] George Grispos, W Glisson, Tim Storer, Calm before the Storm: The Emerging Challenges of Cloud Commuting in Digital Forensics University of Glasgow.

[41] S. Thorpe and I. Ray, "Detecting temporal inconsistency in virtual machine activity timelines." Journal of Information Assurance and Security, vol. 7, no. 1, pp. 24\textendash31, 2012.

[42] Cucurull J., Puiggalí J. (2016) Distributed Immutabilization of Secure Logs. In: Barthe G., Markatos E., Samarati P. (eds) Security and Trust Management. STM 2016. Lecture Notes in Computer Science, vol 9871. Springer, Cham. https://doi.org/10.1007/978-3-319-46598-2_9

[43] Sutton A., Samavi R. (2017) Blockchain Enabled Privacy Audit Logs. In: d'Amato C. et al. (eds) The Semantic Web – ISWC 2017. ISWC 2017. Lecture Notes in Computer Science, vol 10587. Springer, Cham. https://doi.org/10.1007/978-3-319-68288-4_38

[44] Ahmad, Ashar & Saad, Muhammad & Bassiouni, Mostafa & Mohaisen, David. (2018). Towards Blockchain-Driven, Secure and Transparent Audit Logs. MobiQuitous '18: Proceedings of the 15th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services. 443-448. 10.1145/3286978.3286985.

[45] Shekhtman, Louis & Waisbard, Erez. (2018). Securing Log Files through Blockchain Technology. 131-131. 10.1145/3211890.3211921.

[46] Ameer Pichan, Mihai Lazarescu, Sie Teng Soh, Towards a practical cloud forensics logging framework, Journal of Information Security and Applications, Volume 42, 2018, Pages 18-28.

[47] I. Zikratov, A. Kuzmin, V. Akimenko, V. Niculichev and L. Yalansky, "Ensuring data integrity using blockchain technology," 2017 20th Conference of Open Innovations Association (FRUCT), 2017, pp. 534-539, doi: 10.23919/FRUCT.2017.8071359.

[48] J. Ricci, I. Baggili and F. Breitinger, "Blockchain-Based Distributed Cloud Storage Digital Forensics: Where's the Beef?," in IEEE Security & Privacy, vol. 17, no. 1, pp. 34-42, Jan.-Feb. 2019, doi: 10.1109/MSEC.2018.2875877.

[49] Rosa, M., Barraca, J.P. & Rocha, N.P. Blockchain structures to guarantee logging integrity of a digital platform to support community-dwelling older adults. *Cluster Comput* **23,** 1887–1898 (2020). https://doi.org/10.1007/s10586-020-03084-4.

[50] Kirrane, Sabrina & Fernández, Javier & Bonatti, Piero & Milošević, Uroš & Polleres, Axel & Wenning, Rigo. (2020). The SPECIAL-K Personal Data Processing Transparency and Compliance Platform.

[51] M. H. Rakib, S. Hossain, M. Jahan and U. Kabir, "Towards Blockchain-Driven Network Log Management System," 2020 IEEE 8th International Conference on Smart City and Informatization (iSCI), 2020, pp. 73-80, doi: 10.1109/iSCI50694.2020.00019.

[52] M. Pourvahab and G. Ekbatanifard, "Digital Forensics Architecture for Evidence Collection and Provenance Preservation in IaaS Cloud Environment Using SDN and Blockchain Technology," in IEEE Access, vol. 7, pp. 153349-153364, 2019, doi: 10.1109/ACCESS.2019.2946978.

[53] A. Patil, A. Jha, M. M. Mulla, D. G. Narayan and S. Kengond, "Data Provenance Assurance for Cloud Storage Using Blockchain," 2020 International Conference on Advances in Computing, Communication & Materials (ICACCM), 2020, pp. 443-448, doi: 10.1109/ICACCM50413.2020.9213032.

APPENDIX

Algorithm-1: Creation of a new block in blockchain

Input: BI, TS, data, PH
Output: CH, nonce

```
nonce = 0;
BD = concatenate (BI, TS, data, PH);
BD = H (BD);
Repeat
    nonce = nonce++;
    CH = H (concatenate (nonce, BD) );
until prefix of CH = DT;
return CH, nonce;
```

BI – block _index, TS – timestamp, PH – previous_hash, CH – current_hash, BD – block_data, H – Hasher, DT – difficulty_target

Sagar Rane received the B.E. and M.Tech. degrees in computer engineering from Savitribai Phule Pune University, India in 2013 and 2015 respectively. He is pursuing Ph.D. degree in computer and information technology at Savitribai Phule Pune University, India. He worked at Government Polytechnic Pune as a Lecturer in 2014. He was research intern at Center for Development of Advanced Computing (CDAC) headquarters, Pune, India in 2015. Since 2015 he is working as an Assistant Professor in Army Institute of Technology, Pune, India. His current research interest is in Information Security, Cloud Forensics, Blockchain and Web Development. He has published 20+ research papers in reputed journals and conferences. He has three books on his name. He has received various grants from national and international organizations for conducting research activities. He has delivered many sessions in information security and digital forensics domain. Mr. Sagar Rane was a recipient of Sahara Welfare Foundation Fellowship.

Sanjeev Wagh, working as Professor and Head in Department of Information Technology at Govt. College of Engineering, Karad. He has completed his BE(1996), ME(2000) and PhD(2009) in Computer Science & Engineering from Govt. College of Engineering, Pune & Nanded. He was full time Post Doctorate fellow at Center for Tele-Infrastructure, Aalborg University, Denmark during 2013-14. He has also completed MBA in IT from NIBM 2015, Chennai. Also, honored as D.Sc. Award (Honorary). He has total 25 years of experience in academics & research. His research interest areas are Network Security, Natural Science Computing, Internet technologies & Wireless Sensor networks, Data Sciences & Analytics. He has 100+ research papers to his credit, published in International/National Journals & conferences. He has visited various universities in different countries.

Arati Dixit received the B.E. and M.Tech. degrees in computer science and engineering from Pune University and IIT Bombay, India, in 1994 and 1999 respectively. She has completed the certificate in scientific computing program and Ph.D. degree in computer engineering at Wayne State University (WSU), Detroit, MI, USA in 2007 and 2010 respectively. Currently she is a Senior Scientist, Predictive Analytics and Modeling Group at Applied Research Associates Inc., Raleigh and a Teaching Associate Professor, Electrical and Computer Engineering Department, North Carolina State University. Until January 2018, she worked as a Professor at the Department of Computer Engineering, PVPIT, Pune, India. She has published more than 80 research papers in reputed refereed national/international conferences and journals. Her areas of interest are AI and Security Analytics, Cyber Security, Digital Forensics, Cyber Physical Systems and Education Technology. She is ACM Senior member, Chairperson of ACM-W(ACM Council on Women in Computing) India, supporting and advocating full engagement of women in computing. She has been active volunteer and founding Vice-Chairperson of ACM iSIGCSE(India SIGCSE) and Chairperson, ACM Pune Professional Chapter. She has been on ACM India Eminent Speaker Program (ESP) since June 2015, with more than 50 talks delivered. She has received many prestigious awards.