International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-8 Issue-6, August, 2019

Design of Security Technique through Secure Logging for Cloud Forensics

Jyoti Rao, Aboli Deshpande, Pramod Patil, Swati Nikam

Abstract*: Cloud computing has a new edge computing paradigm these days. Sometimes cloud computing architectures don't support for computer forensics investigations. Analyzing various types of logs and logging mechanism plays an important role in computer forensics. Distributed nature and the multi-tenant cloud models, where many users share the same processing and network resources, collecting, storing and analyzing logs from a cloud is very hard. User activity logs can be a valuable source of information in cloud forensic investigations. Generally, Cloud service providers have access to activity logs of cloud user and CSP can tamper the logs so that investigator cannot reach to the real culprit. In such an environment, log security is one of challenge in the cloud. Logging technique is used to monitor employee's behavior, to keep track of malicious activities and prevent cloud networks from intrusion by well-known organizations. Ensuring the reliability and integrity of logs is crucial. Most existing solutions for secure logging are designed for traditional systems rather than the complexity of a cloud environment.

In the proposed framework secure logging environment is provided by storing and processing activity logs and encrypting using advanced encryption method. It detects DDoS (distributed denial of service) attack on cloud infrastructure by using the published logs on cloud and thus helpful in cloud forensics. It is detected by the investigator using available application activity logs in the cloud server. Searchable encryption algorithm will be used to increase the security of the logging mechanism and to maintain confidentiality and privacy of user data. Proof of past (PPL) logs is created by storing logs at more than one place. This PPL helps in the verification process of changed logs by CSP the actual implementation of this application on AWS Infrastructure as a service (IAAS) cloud shows real-time use of this structure.

Keywords: Activity logs, Cloud forensics, Distributed denial of service, Searchable encryption.

I. INTRODUCTION

Cloud computing offers unlimited infrastructure resources, very convenient pay-as-you-use service, and low-cost computing. As a result, cloud computing has become one of the most dominant computing technology in recent years. These day's maximum industries are adopting cloud computing because it does not necessary to set up any type of local infrastructure, and has maximum cost benefit. Cloud computing opens a new horizon of computing for business and IT organizations [1]. However, at the same time, a malicious user easily exploits the power of cloud computing. An attacker can attack applications running inside the cloud. Also, they can launch attacks from machines inside the cloud. These issues are the primary concerns of Cloud Forensics.

There are several forensic analysis schemes and tools available in the market but very few are suitable for the dynamic nature of cloud computing. Traditional digital forensic procedures cannot be directly applicable to the cloud due to its inherent nature; it needs to be updated to retain the same usefulness [2]. Unlike a conventional client device, cloud virtual machines (VMs) can be supported by hardware that might be located remotely and thus would not be physically accessible to an investigator. In addition, VMs can be distributed across multiple physical on the same physical components. Therefore, holding the machine for forensic analysis is not possible in most investigations. Furthermore, data residing in a VM may be volatile and could be lost once the power is off or the VM terminates. Hence, the cloud service provider (CSP) plays a crucial role in the collection of evidential data (e.g. cloud user's activity log from the log). For example, the CSP writes the activity log (cloud log) for each user. Thus, preventing modification of the logs, maintaining a proper chain of custody and ensuring data privacy is crucial [3].

Each log is recorded computer event that corresponds to a specific user. Privacy, integrity, and Confidentiality of such data are crucial while designing secure logging system. Such data must be preserved, to maintain user privacy and to facilitate potential investigative activities.

The objective of this paper is to analyze existing logging mechanism, to propose enhanced technique using searchable encryption and detect DDoS attack and helps in cloud forensics

A. Challenges of Log Forensics

The inherent nature of cloud like its scattered infrastructure, multiple-client resources, enormous running applications, thousands of cloud users, on demand response and virtual environment makes log forensics very difficult. Collecting, analyzing, storing of logs in real time makes forensics investigation harder than digital forensics. Log forensics challenges are as follows [3].

Revised Manuscript Received on October 30, 2019.

* Correspondence Author

Dr. Jyoti Rao*, Associate Professor in D.Y. Patil Institute of Technology, Pimpri.

Aboli Deshpande, computer engineering in 1999 from government college of engineering Aurangabad and is pursuing M.E. from D.Y. Patil Institute of Technology, Pimpri, Pune.

Dr. Pramod Patil, HOD of Computer Engineering in DIT Pimpri, Pune Padmashree

Dr. D.Y. Patil, Assistant Professor in Department of Computer Engineering since August 2001. Perusing PhD from University of Pune.

Log Security: As the content of log may provide evidence in log forensics, the security of log file is at most important in forensic investigation. Confidentiality, integrity, and availability (CIA) of log files is crucial in log forensics [4]. Investigator can reach up to malicious individual by analyzing log files. If log files are tampered, then reaching the real culprit is a difficult task. An attacker can attack at log file storage, a network, where log files are transferred from one place to other or at cloud log analysis server where investigations are carried out. Investigations produce biased results if the CIA of the log file is not properly managed.

CSP plays an important role in maintaining CIA triad of a log file. By encrypting log files and by ensuring proper access of a log file to the proper individual, CSP can able to maintain a CIA of a log file and thus helps in cloud forensics procedure [3].

Cloud Log Accessibility: In cloud computing environments the generation of cloud log files is easy, but gaining access with the proper requirement is difficult [5]. It has to be accessed with a clear objective. Legal authority approves access to an authorized individual only if they have a valid reason. Full access to required cloud logs is given to each forensic investigator to investigate malicious activities. Authorized access to logs leads to correct log investigation.

Sometimes due to privacy and security issues, CSP does not allow log access to third-party agency or forensic examiner.

Nonstandard Log Format: There are many cloud log formats depending on requirements as various cloud log files being generated in a cloud computing environment. There is no standard log format which expresses various types of cloud logs. For example, network log, application logs have their own log format. Same application on a different cloud may have different log format. It is very difficult for forensic examiner to analyze logs with a different format. If there is a standard log format, a forensic examiner easily understands data in log and accurately identifies malicious activator [3].

Decentralize Logs: In a decentralized cloud environment, there is no central location where logs are accumulated [6]. There are various layers such as operating system, applications, database, and network layer. These layers generate various different logs with their own log formats [5]. Access of all logs in every layer is provided to the investigator as it may prove to be helpful in the forensic process. One application in a virtual machine has multiple logs, reside on different servers placed at different clouds. Log forensics is difficult due to the accessibility of logs, network delays, and access to servers, involvement, and availability at a different cloud. Investigation in real time of such scattered cloud logs for malicious activities is challenging.

Log as Big Data: Analyzing a huge amount of cloud log data generated at various resources is a hard task for CLF investigator. The volume of cloud log data is " big data" problem [7]. It is a time-consuming activity to analyze such big data and find malicious individual than in traditional digital forensics [8].

Volatile Virtual Machine (VM) Logs: VMs in the cloud are volatile so its logs and data. Data and logs of VM will be lost if switched off, shut down or terminated [9]. Attackers use these VMs to do malicious activities and then terminate the VM. Generated logs disappear with VM termination. For log forensics, preservation of such logs is important to reach a real attacker.

Multiple Tenant and Jurisdictions: Every jurisdiction has its own rule for access and retrieval of data or log, admissibility of evidence, recovery of evidence without harming tenant rights, and chain of custody. As there is no worldwide regulatory body or any federation of national bodies, it significantly affects the effectiveness of cloud log forensics investigations [6].

II. LITERATURE SURVEY

Ahsan MM et al. [9] proposed a secure logging scheme called Cloud Log Assuring Soundness and Secrecy (CLASS) based on SecLaas scheme. This scheme is specially designed to help in cloud forensics in which privacy of a user is preserved even though there is collusion among CSP, investigator and cloud employee. The privacy of cloud users is preserved by encrypting cloud logs with a public key of the individual user while also facilitating log retrieval in the event of an investigation.

It also allows the user to identify any log modification. Once proof of past log (PPL) is established even user cannot refuse his own log in case of an investigation. Unauthorized modification is prevented by generating proof of past log (PPL) using Rabin's fingerprint and Bloom filter. This approach reduces verification time and ensures that CSP is writing correct logs. To prove that class is applicable in real-world author tests it by deploying it on Open Stack. Limitation of this system is at least one CSP should be honest.

Shams Zawoad et al. [1] has drafted Secure-Logging-as-a-Service (SecLaaS), which stores virtual machine's logs and permit access to forensic examiners by maintaining the privacy of the cloud customers. The confidentiality of the cloud logs from malicious investigators or CSPs is maintained by generating proof of past logs. Publishing proof of past log regularly by CSP and ash chain of logs is used to maintain integrity.

To generate PPL, a new accumulator scheme called Bloom Tree is used which proved to be better in time, integrity verification and space requirement than bloom filter and also provide negligible false positive. Eventually, Author successfully ran the system using network logs in the cloud on the open stack and checks the feasibility on the cloud. Limitation of this system is user privacy is in danger if cloud employee colludes with forensics examiner. CSP can able to enter false entries before proof of past logs is created. This system is unable to verify that CSP is writing true information in the log or modify the log or eliminate some information from logs.

Anwar et al. [10] proposed the solution for cloud forensics by providing secure logging with the operating system and the security logs. Cloud computing environment of Eucalyptus was set up using Snort, Syslog and Log Analyzer. They examined the characteristic of Eucalyptus and preserved all the logs of Eucalyptus objects.

The author created their own database by attacking Eucalyptus with different attack then finds the log entries made by an attacker and thus reduce the effort of forensics examiner to identify the attacker and helps in the investigation. They launched a DDoS attack from two virtual machines.

They analyzed generated logs on the Cloud Controller (CC) machine and easily identify the attacking machine IP from which attack is carried out, the type of browser and requested content. Security, access control and verification of log were not considered in this work.

Ma and G Tsudik [11] analyzed existing insecure logging system and identify the problems in it. They have given two approaches based on Forward-Secure Sequential Aggregate (FssAgg) authentication technique. Both of these schemes provide practical secure logging as they removed dependence on trusted third party or secure hardware. They demonstrated schemes using one private variable and one public variable. In one approach, two tags are appended with each log entry, one is for trusted verifier and other is for semi-trusted log accumulator. They tried to maintain forward secure stream integrity instead of forwarding security.

This FssAgg tag can confirm any log prior to it in a particular epoch. The end tag of an epoch can `testify' the entire chain of log entries up to that epoch. Both schemes provide security and space efficiency. The drawback of this system is that additional computation cost required during the verification phase.

In Ray et al. [12] approach authenticated channel is created so that the series of logs are collected at CSP from a logger or log accumulator. Then, the cloud server tries to maintain confidentiality, integrity, availability, and verifiability of secure logs. Special types of logs are used at start and endpoints of each block of logs to avoid truncation error from both ends. To save logs from privacy violation and security breaches, sequentially generated keys are used to encrypt log entries.

The integrity of logs is maintained by generating another set of keys in the same way. There is no option for public verifiability as symmetric key encryption is used to encrypt log to protect privacy and confidentiality.

Tian et al. [13] recently proposed a scheme for public auditing of operational behavior in the cloud. All basic requirements for security and performance like non-repudiation of operational behavior tamper resistance of logs and selective verification of log blocks are satisfied by block based logging. An unlimited number of auditing operations are performed by the public auditing method for single log block. Selective verification for multiple log blocks is supported by binary auditing tree-based public auditing method. They also used widely recognized hash-chain schemes for forwarding security and append-only property. To reduce the burden faced by forensic investigators and for log credibility, they introduced the idea of a trusted third party.

One novel approach they used is in the authentication structure is Merkle Hash Tree (MHT) for tamper resistance. They stressed on effective verification of the integrity of operation behavioral logs. MHT is vulnerable to pre-image attack as a direct result of how it functions.

Lokhande and Mane [14] proposed robust and forensic enable architecture in which bloom filters are attached at R-tree nodes and this data structure is used to store or accumulate the logs. This accumulator is used by CSP to publish proof of past logs and to check that CSP is providing correct logs to the forensic investigator. Logs integrity is protected by a hash chain scheme.

The confidentiality and integrity of the extracted logs are preserved so that logs become proved to be authenticated proof in the investigation. CSP can add, modify, change and delete activity logs. This system works for those logs which are not altered before handover to an investigator.

Sheik Khadar et al. [15] suggested framework based on trusted third party (TTP) along with a cloud forensics investigation team (CFIT) proves to be a better solution to enhance the trustworthiness of the service provider and thereby facilitate the cloud providers to trap cyber attackers with strong collection of pieces of evidence which might help in further legal process.

TTP is the central authority of the cloud environment total responsibility and security lies with TTP. As TTP handle all request from cloud customer(CC) and CSP, CC's are unaware of their CSP's and vice versa. The TTP takes care of the privacy of CC.

Aswathy Mohan [16] suggested the log-based model which helps to decrease the complexity of forensic for non-repudiation of user behavior on a cloud. This is a model for SAAS (Software as a Service) service in which consumer uses an agent to send commands to SAAS. And after SAAS processes the commands and creates logs for that, it will send back a response. While consumer gets there response the agent may make its own logs or just processes the response to the user. It means that it should keep another log locally and synchronously. Thus this model can be used to check the activities on SAAS cloud without the help of the CSPs.

Birk et al. [17] outlined the basic issues of forensic investigations in cloud environments. They also listed current challenges and provided possible solutions. The author also mentioned that these issues are occurring as there is no global standard in cloud computing related to proper deployment, security, and compliance. These issues make forensics examination hard. They also suggested read-only APIs exposed by CSPs in which network, access and process logs are available to the user.

A tool for Open Stack to collect virtual disks, API logs, and guest firewall logs called FROST [18]. Forensic tools are integrated into the management plane of cloud architecture. Cloud user, investigator and law enforcement agency can collect forensics data without the permission of cloud provider. FROST stores logs in hash trees and proof of old logs are generated using crypto-logic hashes. In this system, without the interference of the provider, all the logs are provided to the investigator. FROST is used for auditing, metrics and real- time monitoring beside incident response and forensics.

It also provides forensic capabilities to the cloud user. One limitation of FROST is it needs trust in the cloud provider. [17][18]These works could not focus on protecting user's privacy and integrity of logs from malicious CSP and investigators.

Saibharath & Geethakumari [19] proposed the system in which an engine handles multiple cloud provider platforms for cloud acquisition and pre-processing is implemented. The collected evidence from different cloud provider platform is analyzed and preprocessed from which features and values are built. These are used in clustering as evidence files. Cloud forensic clustering is done across multiple virtual machine instances. These are validated by testing data set from ten virtual machines obtained from the open nebula and open stack cloud. Every virtual machine has a virtual machine disk and its associated RAM image.

This forensic clustering solution reduces the search space, enables multi-drive correlation and forms a social network of virtual machine instances. Addressing different cloud architectures, open source cloud platforms Open Nebula and Open Stack are compared with respect to the location of evidence artifacts.

Arsalan Ali et al. [20] try to address the issue of digital forensics in the cloud environment. They suggested an architecture which helps forensics examiner to collect the logs of suspected user. To secure the data in the cloud from malicious attacks Host-based Intrusion Detection System (HIDS) is proposed. Email alerts and Secure Shell (SSH) messages are generated from the web server after getting intimation from HIDS to stop further malicious activities. In this way, the digital forensic investigators get reliable evidence of suspected user. Thus HIDS helps in log collection for digital forensics in the cloud.

Schneier and Kelsey [21] proposed a log management scheme based on forward integrity and provided various real-world example applications. The forward integrity property is ensured using a secret key which is the initial point of a one-way hash chain and message authentication code rather than PRF in Bellare and Lee scheme [23].

Both schemes rely on the fact that, keeping a small and secret piece of information with each log entry which cannot be generated without a secret key, and this secret key changes with each new log. A log entry can be verified using secret information later on its integrity. Both schemes require the presence of an online trusted server to maintain the secret key and to verify its integrity.

To remove the need for an online trusted server, Holt [22] proposed using public key cryptography instead of private keys residing in the trusted server and keeping the encrypted keys with the log entries. Thus, it was proposed to use an elliptic curve cryptosystem. Instead of using the one-way hash function, the author used a digital signature. More computations are required when public key cryptography is used. This approach does not consider privacy protection from an honest but curious logger.

III. RPROPOSED ARCHITECTURE AND METHODOLOGY

The main aim to create a framework for a distributed Cloud system, which has the ability to detect DDOS attack, is to validate designed logging system whether it is secure and reliable and proves to be helpful in cloud forensics. This is done by flooding system with multiple requests. When the incoming request exceeds the limit then the server will not be able to process the request and may crash or not able to provide service to a legitimate user.

In the proposed mechanism, the user generates activity logs, analyzing available cloud logs that will help to detect DDoS attack on cloud infrastructure. To implement DDoS attack, java code will be used to start multiple requests to Cloud server. Multiple requests flooded onto the server in such a way that it cannot serve to the legitimate user request.

A. Proposed Methodology

In this framework file storage system is designed. Activity logs of user are generated by performing file related activities like uploading, sharing, updating and deleting. Generated logs records the information like IP address of cloud user, user ID assigned automatically by the system, email id of user, MAC (Media Access Control) address of machine, previous ID of a log and saved on the cloud. File related activity performed by user date and time of log, the hash value of IP address of a user by applying MD5 algorithm to IP address. Logs are fully encrypted using symmetric standard encryption/AES (Advanced Encryption Standard) and saved on the cloud.

As part of project, DDoS database is generated by randomized request from the entire user at a time so that the server is unable to handle the requests and crash. After the DDoS attack even legitimate user request is not served by the server until it is started manually. Investigator module can be able at this point to investigate the matter. After the investigation, this system displays the IP address and the names of those users from which DDoS attack is performed. Cloud service provider publishes it on the internet to preserve for future investigation.

Logs are stored at three places to generate proof of past logs named as LOGS, LOGS1, and LOGS2. CSP can access logs stored at LOGS. At the time of verification of log of particular cloud user logs stored at LOGSs, LOGS1 and LOGS2 play an important role. If user colludes with CSP and CSP changes the logs at LOGS, this system's investigator module can able to detect changed logs by CSP and display it on screen.

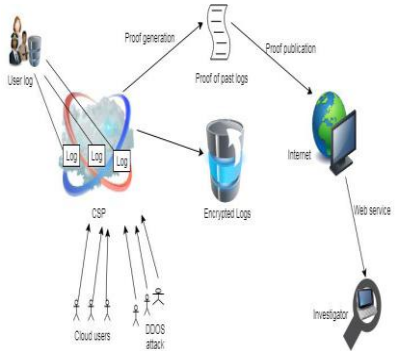

Fig. 1 Proposed DDoS Framework

When any malicious activity is to be detected like DDoS attack, corresponding logs are encrypted and saved with a time stamp of a particular user. The investigator compares these logs with the logs in a database which were already published by CSP. If logs are matched then IP address corresponding to logs are fetched and decrypted. Hence the IP address of malicious users is identified. Thus this system helps in cloud forensics procedure.

The central outline of the proposed method is as follows.

1) Activity logs of user are generated.
2) Tags are generated by applying message digest (MD5) algorithm to IP address of cloud user machine. These tags are used to search DDoS attacker.
3) Generated logs encrypted using Advanced Encryption Standard (AES) with key size 128 bit.
4) Generated logs are saved in SQL database.
5) Cipher text of same IP is different every time it is created.
6) Proof of past log (PPL) is created by fetching current system date and database entry of logs for that day.
7) Cloud service provider publishes it on internet to preserve for future verification.
8) Java code will be used to start DDoS attack by starting multiple requests to Cloud server.
9) Log database is created for corresponding requests.
10) Multiple requests flooded onto the server in such a way that it cannot serve legitimate user request or unable to provide service.
11) The investigator compares DDoS logs in database with the logs already published by CSP.

The proposed framework of cloud model contains following modules.

1) Cloud User
2) CSP
3) Cloud
4) Investigator
5) Server Test

1) Cloud User:

Cloud user allows new user to register who want service from cloud. User is registered on cloud by providing details like Email, name, mobile no. and password. User can able to select the cloud resource plan and send a request to CSP according to his own requirement. User can check the status of the request either accepted or pending from “Response from CSP” tab.

2) CSP:

CSP has given authority either accept or reject a user request. If CSP rejects the user he has not eligible to get the particular service from CSP. The user becomes authorized once his request has been accepted. CSP published log data to the internet to preserve the activity logs of a particular user which are become useful for the investigator to detect DDoS attack.

3) Cloud:

Once the user is authenticated by CSP, then only he is able to use computer resources provided by CSP and can log in as a valid user in cloud. If CSP rejects the user’s request, cloud resources are not accessible to that user. In this file storage model user can upload, share, delete and updates the files which are stored on the cloud.

4) Investigator:

An investigator is responsible for investigating DDoS attack by analyzing log entries in the server database. Secure and reliable logs are provided to the investigator to detect the attack. Logs are encrypted by using advance encryption methods so that the confidentiality of logs is retained.

5) Server Test:

Server test is done after DDoS attack is executed and to check
Server is unable to serve any request unless and until it is manually started.

A. Structure and field of an activity log

A record of the events occurring within an organization’s information systems and networks is called log. Logs are used to increase system and network performance, to keep track of actions of users, and providing data useful for investigating malicious activities [10]. Actions taken by a user in cloud infrastructures is revealed by activity logs of cloud users [3]. In this scheme, activity logs are generated by performing file related activities to the cloud. Format of a log will be set according to the requirement of particular organization. Structure of a log for this system is as follows.

< Log id, From IP, Email, UID, Dt, Ip digest, Activity, PID Filename>

It records the following information about the user. Log id is serial number in cloud database, IP address of user (From IP), email ID of user from which user is uniquely identified, user Id (UID), Date/time of log (Dt), previous ID of log is saved in current log entry to form log chain and maintain integrity of logs.

B. Log storage algorithm and PPL generation

First, framework for secure logging by using log storage algorithm will be created and then the framework will be tested against DDoS attack.

Searchable encryption is an encryption technology for searching data in an encrypted state. In this Scheme for searchable encryption that uses tags to determine whether data correspond to search item is a match or not. These tags are generated from MD5 algorithm. Message digest is a hash value created from IP address [26].

The use of random numbers to create cipher text makes different cipher text of the same word or phrase. So every time, the IP address of a same user appear differently in encrypted format in the database. Due to this encryption, CSP cannot able to change or modify the logs of a particular user. Tampering of logs is not possible even if CSP collude with user or investigator. Even if logs are modified by CSP, it is verified by investigator in the later stage. Non tampered logs are provided to investigator thus this scheme proves to be helpful in cloud forensics.

During a search of an attacker, "message digest of IP address of cloud user" and "message digest of IP address of an attacker is compared. If the comparison indicates that the two kinds of digest conform, the search is considered to be a hit. Attacker user is found. IP address of an attacker and name is displayed on the screen. In this cipher text is different every time it is created.

In proposed work to create secure logging environment activity logs are fully encrypted using symmetric encryption technique like AES with a key size of 128 bit, a tag for search purposes is created by using MD5 (Message Digest 5) algorithm and is attached to their respective cipher texts. MD5 and AES based hybrid cryptographic algorithm called searchable encryption is used for providing security.

Algorithm 1 is Logstorage algorithm as shown below, which takes input as tag and recorded log (LE) and create output as encrypted log in the database as database log entry (DBLE). EncryptAES is standard AES algorithm applied to the current log.

Log Storage (log entries LEs, TG)

$for\ i \leftarrow$ 1 to size (LEs)

Encrypted_log i= encryptAES (log $\_entry\ i$)

Log$_chain$ i= (encrypted_log$i$ || log$_chain\ i-1$);

Database_log_entry i = <log$_chain$, TGi >;

Store database_log_entry$i$ into DBLE;

End for;

Algorithm: 1 Logstorage pseudocode for processing log [9]

Step1:

User Logs in into the system, using login id and password. Login id of user is its valid email address using which User is registered.

Step2:

User places request to cloud for one of the available different virtual machine configurations as per his requirements. Unless and until CSP accept the request of virtual machine user is not able to login to the cloud and does not do any activity.

Step 3:

After the requested grant by CSP user can upload, share, delete and update the files and corresponding activity logs are generated. Logs record some important information like email id of the user, IP address of user machine and date and time of activity logs and stored in the database after some processing. LE is recoded log entry which will be stored in the database. Log id is serial number of log in the database. From IP is IP address of cloud user machine. Dt is date and time of generated activity log.

LE= < Log id, FromIP, Email, UID, Activity, Dt >

For ease of understanding, only few fields are shown in LE.

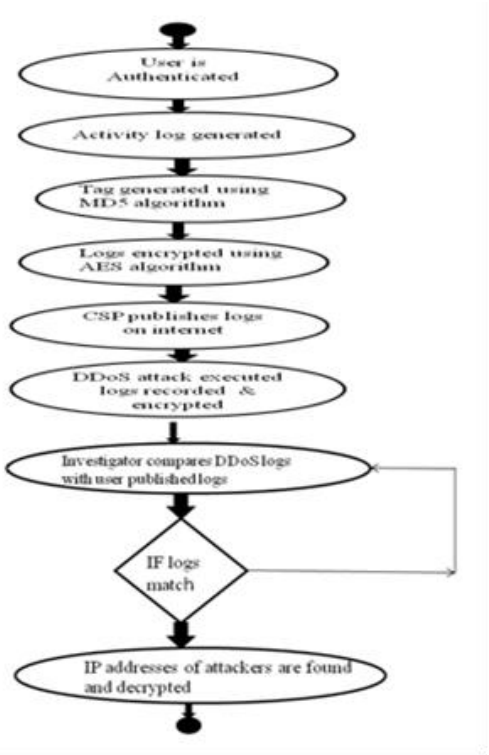

Fig. 2 Flowchart DDoS System

Step 4:

TAGs (TG) is generated by applying MD5 algorithm to IP address of the user. TAG is hash value generated from IP address. Investigator may use these TAGs to locate DDoS attacker's machine after the attack.

TG = H (FromIP)

Step 5:

To preserve the privacy of the user, logs are encrypted using AES algorithm and encrypted log entry (ELE) is created in the database. The AES symmetric key is denoted by Ese.

ELE= < Ese. (Logid, FromIP, UID, Email, Dt) >

Step 6:

To prevent potential manipulation and protect the integrity of logs, the log chain is created by storing previous log ID (PID) with a current log entry.|| shows concatenation between two messages.

LC= (ELE || LCPID)

Step 7:

After generating tag and log chain.log entry is made to a database. This scheme generates database log entry (DBLE) with LC and TG. While creating an entry of log in a database, PID is stored with it. It will help us in creating a log chain so that the integrity of logs is not violated.

DBLE = < LC, TG>

Step 8:

Proof of past log (PPL) is created by fetching current system date and database entry of logs for that day which is then made available to the investigator by publishing it on the cloud. We will generate PPL in a batch of the logs at each end of the day.

PPL= < signatureCSP (DBLEDt)>

This PPL is accessible to the investigator if court permit to check past logs of a particular user in case of malicious activity.

Step 9:

To validate the designed secure logging system, DDoS attack is

executed by running a java code to start multiple mouse click requests from multiple users to a server to exhaust its communication channel or bandwidth resources. Server's bandwidth is exhausted and it is not able to provide legitimate request.

To exhaust a communication channel threshold of 150 requests are kept. Continuously 150 requests from different cloud user can be handled by server but after that count server gets crashed and unable to serve the request from legitimate request.

Step 10:

All DDoS logs are recorded and stored in encrypted form using AES symmetric encryption method and generate message digest of IP address.

Step 11:

At investigator site, investigator investigates DDoS An attacker by comparing tags i.e. message digest of IP address of the user with tags in DDoS logs. If both tags matches with each other that means that is the IP address of a malicious user which contributed to DDoS attack.

PPL Generation

In this scheme, generated logs are fully encrypted and after processing stored according to the system date and time in database as LOGS, LOGS1, and LOGS2. A batch of a log which is stored according to the chronological order of date and time in the database is used by CSP to create a proof and then publish on a cloud to avail investigator.

CSP publishes logs at the end of the day by creating a proof by using a digital signature on that particular date. Digital signature is used to authenticate the logs generated on a particular date. Database entry of particular date is created as DBLEDt. If there are "n" logs generated on that day, a log chain created is as follows

DBLEDT = (LCPID TG1|| LCPID-1 TG2 ||-------|| LCPID-n TGn)

PPL= < signatureCSP (DBLEDt)>his journal uses double-blind review process, which means that both the reviewer (s) and author (s) identities concealed from the reviewers, and vice versa, throughout the review process. All submitted manuscripts are reviewed by three reviewer one from India and rest two from overseas. There should be proper comments of the reviewers for the purpose of acceptance/ rejection. There should be minimum 01 to 02 week time window for it.

IV. ANALYSIS

We evaluated our proposed approach with respect to security properties such as confidentiality, integrity, availability, and privacy.

Confidentiality: Logs are generated, fully encrypted and then stored in the database, logs even not visible to logger i.e. CSP thus confidentiality retained.

Integrity: To prevent potential manipulation and protect the integrity of logs, log chain is created by storing previous log ID (PID) with current log entry and also publishes its proof on the internet i.e. stored on the cloud so integrity can be protected.

Tamper Resistance: The IP address of the same user appears differently in encrypted format in the database. Due to searchable encryption CSP cannot able to change or modify the logs of a particular user. Tampering of logs is not possible even if CSP collude with user or investigator.

Privacy: This framework preserves the privacy of cloud users by encrypting total cloud logs while also facilitating log retrieval in the event of an investigation.

Availability: Logs are available to the investigator if a Court of Law orders the accessibility of logs for the investigation of malicious activity. CSP published the logs on the internet which are always available to the investigator at the time of the investigation.

Admissibility: This scheme has the ability to verify if there is any change or modification of logs so logs are reliable to be admitted in the Court of Law and also publishes the logs on the internet thus admissibility assured.

Verifiability: Investigator can verify the logs of particular cloud user if CSP changed the logs of that user and able to display changed logs.

V. RESULT AND DISCUSSION

In this secure architecture, Applications activity logs of user are taken as input. By applying MD5 algorithm, a hash value for the IP address of the user is generated which are recorded in logs called tags. Logs are encrypted using AES algorithm and saved in the database.CSP publishes the logs on internet so that they are accessible to the investigator.

Logs are published on cloud at three places so that it can be helpful in verification phase to investigator. If logger modified the logs of particular cloud user, investigator checks the integrity of logs by verifying it with published logs which are not accessible to logger.

If a malicious activity like DDoS attack has happened, the investigator compares tags associated with DDoS logs and database logs. If a hash value of tags matches with each other, then IP address in encrypted form is considered to be same. An IP address is extracted and the attacker is identified.

Results are verified on AWS IAAS Service by creating two instances one using Elastic beanstalk for deploying the application and another using AWS RDS (Relational Database Service).SQL workbench is connected to AWS RDS using endpoints.

Following is proposed secured cloud system in which file storage system is implemented and logs related to file activities are stored and processed. Following are the screenshots of implemented system

Fig.3 cloud plans

Multiple combinations of configurations are available at the cloud server as shown in Fig. 3 cloud plans.

User has a choice of selecting one of them as per their requirement.

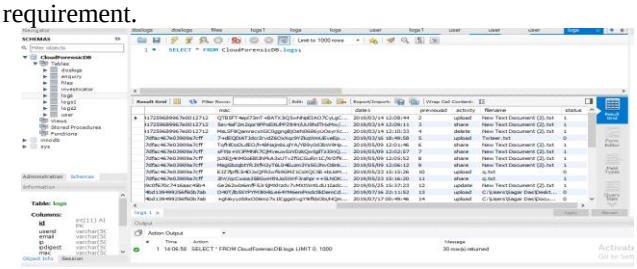

Fig.4 logs

This is log database as shown in Fig.4 logs, in which logs corresponding to cloud user's activities are stored. Recorded logs contains following fields

LOG=(ID, USERID, Email, IP Add, Tag ,MAC Add, System date, Previous ID, Activity, File).

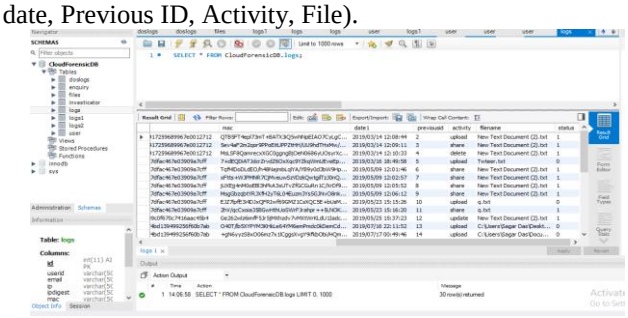

Fig.5 halflog

This is half part of above log recorded in a database which contains the field date and time of generated log, the previous id of log; activity done by user, file name on which activity is done.

Fig.6 DDoSIP

Above Fig. 6 DDoSIP shows the IP address and name of attacker who executed the attack. Thus this scheme helps in forensic investigation for this application.

The confidentiality to data is provided while it is stored and transmitted through cryptography techniques. In this system, AES which is symmetric encryption block cipher of key size 128 bits. The AES encryption is better than other encryption methods like RSA and 3DES.

For text files [24]:

1. The encryption time for AES is almost half that of DES system and one fifth that of RSA.
2. The Decryption time for AES is almost half that of DES system.

Table1: execution times (sec) of AES and 3DES [25]

Input size in (bytes)	3DES	AES
20527	7	4
36002	13	6
45911	17	8
59852	23	11
Average Time(sec)	15	7.25

AES execution time for 16 kib (16384 bytes) file for 128 Bit encryption on 2.4 MHz machine was 4 sec as against 7 sec for 3DES . Table-I shows the execution time to encrypt text file of different sizes. The average time for AES was 7.25 seconds as against 15 seconds for 3DES.The time for 3DES is almost double than AES[25].

In this way approximate execution time to encrypt 16 kib log file in this system is 100 percent faster than 3DES.Performance v/s security trade-off: There is no limit for security measure and no security measure is completely secure. The security and encryption/decryption time will go together. There is a trade-off between security and performance in real world [25]. To make the system more secure no of rounds in encryption algorithm is increased but it affects the performance by increasing the encryption time

REFERENCES

1. Z S. Zawoad, A. K. Dutta, R. Hasan, "SecLaaS: secure logging-as-a-service for cloud forensics," In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, 2013, pp. 219-230.
2. B. Martini, K. K. R Choo, "An integrated conceptual digital forensic framework for cloud computing," Digital Investigation, vol. 9,2012, pp. 71-80.
3. S. Khan *et al*,"Cloud log forensics: foundations, state of the art, and future directions," ACM Computing Surveys (CSUR), vol. 49, 2016
4. p. 7.
5. L. K. Ryan, M. Kirchberg, B. S. Lee, " From system-centric to data-centric logging-accountability, trust & security in cloud computing. In Proceedings of the IEEE Defense Science Research Conference and Expo (DSR). Singapore, 2011b, pp. 1–4.
6. S. Zawoad, A. K. Dutta, R. Hasan, "Towards building forensics enabled cloud through secure logging-as-a-service," IEEE Transactions on Dependable and Secure Computing, vol. 1, 2016, pp. 1-1
7. K. Ruan, J. Carthy, T. Kechadi, M. Crosbie, "Cloud forensics: An overview," In proceedings of the 7th IFIP International Conference on Digital Forensics, 2011, pp. 16-25.
8. V. Wesley, T. Harris, L. Long Jr., and R. Green. 2014 , " Hypervisor security in cloud computing systems, " ACM Comput. Surv., 2014, pp. 1–22.
9. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, and S. U. Khan, " The rise of "big data" on cloud computing: Review and open research issues. Inform. Syst. Vol. 47 , 2015, pp 98–115.
10. Ahsan *et al.* , " CLASS: Cloud Log Assuring Soundness and Secrecy Scheme for Cloud Forensics, " IEEE Transactions on Sustainable Computing. May 2018.
11. F. Anwar, Z. Anwar, "Digital forensics for eucalyptus." In 2011 Frontiers of Information Technology, " IEEE, 2011, pp. 110-116.
12. D. Ma, G. Tsudik, "A new approach to secure logging." ACM Transactions on Storage (TOS) , Vol. 5,2009, p. 2.
13. I. Ray, K. Belyaev , M. Strizhov , D. Mulamba,M. Rajaram, "Secure logging as a service—delegating log management to the cloud." IEEE systems journal, Vol. 7, 2013, pp. 323-334..
14. H. Tian et al, "Enabling public auditability for operation behaviors in cloud storage." Soft Computing , vol. 21 , 2017, pp. 2175-2187.
15. P. Lokhande, V. Mane, "Log based privacy preservation in cloud forensic," 2016.
16. S. K. A. Manoj, D. L. Bhaskari, "Cloud Forensics-A Framework for investigating Cyber Attacks in cloud environment," Procedia Computer Science, vol. 85,2016, pp. 149-154.
17. Aswathy Mohan V et al,"A Log-Based Approach to Make Digital Forensic Easier on Cloud Computing", International Research Journal of Latest Trends in Engineering and Technology (IRJLTET), Vol.3 , May/June 2016.
18. D. Birk , C. Wegener, "Technical issues of forensic investigations in cloud computing environments," in SADFE. IEEE, 2011, pp. 1–10.

19. J. Dykstra , A. T. Sherman, “Design and implementation of frost: Digital forensic tools for the OpenStack cloud computing platform,” Digital Investigation, vol. 10, 2013, pp. S87–S95.
20. S. Saibharath, G. Geethakumari, "Pre processing of evidences from cloud components for effective forensic analysis," Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on. IEEE, 2015, pp. 394 -339.
21. A. A Shaikh,Qi Heng , W. Jiang, T. Muhammad "A novel HIDS and log collection based system for digital forensics in cloud environment," In 2017 3rd IEEE International Conference on Computer and Communications (ICCC), IEEE, 2017, pp. 1434-1438.
22. B. Schneier ,J. Kelsey, "Secure audit logs to support computer forensics," ACM Transactions on Information and System Security (TISSEC), vol. 2, 1999, pp. 159-176.
23. J. E. Holt, "Logcrypt: forward security and public verification for secure audit logs," In ACM International Conference Proceeding Series, vol. 167, 2006, pp. 203-211.
24. M. Bellare, B. Yee. "Forward-security in private-key cryptography." In Cryptographers’ Track at the RSA Conference, Springer, Berlin, Heidelberg, 2003, pp. 1-18.
25. M. Panda, "Performance analysis of encryption algorithms for security. In 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), IEEE,Oct 2016, pp. 278-284.
26. A. Nadeem, M. Y. Javed, . "A performance comparison of data encryption algorithms. In 2005 international conference on information and communication technologies, IEEE, Aug 2005, pp. 84-89.
27. http://www.hitachi.com/rd/portal/contents/story/searchable_encryption

AUTHORS PROFILE

Dr. Jyoti Rao has total 18 years of Teaching Experience in Computer Engineering. She is working as Associate Professor in D.Y. Patil Institute of Technology, Pimpri. She is approved Post Graduate Teacher at SPPU. She completed her PhD in 2016 on topic "Novel and efficient Visual Cryptography scheme for privacy protection". She has 2 Patent Published in IPR. She has more than 20 papers are published in International journals

Aboli Deshpande, passed computer engineering in 1999 from government college of engineering Aurangabad and is pursuing M.E. from D.Y. Patil Institute of Technology, Pimpri, Pune.

Dr. Pramod Patil An alumnus of COEP Pune, holds Masters in Computer Engineering and Ph.D from COEP. He has total 14 years of experience in Academics, Research and Industry. He has 20 research articles in National & International Journals and Conferences to his credit. He is currently the HOD of Computer Engineering in DIT Pimpri, Pune and is member of professional society such as CSI, ACM, IEEE, and ISTE.

Dr. D.Y. Patil, Working at Padmashree Institute of Engineering & Technology, Pimpri, Pune as an Assistant Professor in Department of Computer Engineering since August 2001.
Perusing PhD from University of Pune. She has Published 8 Research papers in National /International Journals, 10 Research papers in National /International conferences and also attended 27 Workshops, FDP, STTP etc. and also has the experience of Organizing 2 workshops. Guiding students at UG as well as PG level.