FECloud: A Trustworthy Forensics-Enabled Cloud Architecture

Shams Zawoad and Ragib Hasan
{zawoad, ragib}@cis.uab.edu
University of Alabama at Birmingham
Birmingham, Alabama 35294-1170, USA

Abstract

The rapid migration from traditional computing and storage model to the cloud model creates the necessity of supporting reliable forensics in the cloud. However, today's cloud computing architectures often lack support for forensic investigations because many of the assumptions that are taken for granted in traditional digital forensics do not apply to clouds. Hence, the existing digital forensics tools cannot handle the dynamic and black-box natures of clouds. Moreover, trustworthiness of evidence can be questionable because of the possibility of collusion between dishonest cloud providers, malicious users, and investigators. Since reliability and accuracy of evidence are very important factors while evaluating evidence during a criminal investigation and prosecution, we need to preserve the integrity of evidence before and after collecting from clouds.

In this paper, we first identify the required properties to support trustworthy forensics in clouds. Based on the requirements, we propose a forensics-enabled cloud architecture (FECloud) to preserve and provide required evidence while protecting the privacy and integrity of the evidence. FECloud is designed on top of Openstack – a popular open source cloud computing platform. Incorporating architectures like FECloud may impose significant business impacts on Cloud Service Providers (CSP) as well as customers. CSPs can attract more customers with the assurance of providing proper forensics support. Likewise, customers do not require extreme investment on establishing their own forensics friendly infrastructures.

Keywords: Cloud Forensics, Digital Forensics, Cloud Security, Cloud Architecture

1 Introduction

Cloud computing is one of the major forces behind many services. Today, consumers are enjoying the services provided by clouds when they access Gmail, Google Calendar, Dropbox, Microsoft Office Live, or run hundreds of Amazon Elastic Compute Cloud (EC2) instances for processing large-scale data. According to Gartner, consumers will store more than one third of their digital content in the cloud by 2016 [1]. Due to the high demand for cloud-based services, cloud computing has emerged as the dominant computing paradigm in recent years. A recent research by Market Research Media states that the global cloud computing market is expected to grow at a 30% Compound Annual Growth Rate (CAGR) reaching $270 billion in 2020 [2].

However, the highly-scalable computing and storage resources offered by the cloud can be misused by malicious users to perform attacks from machines inside the cloud [3, 4]. A criminal can also keep some secret files (e.g., child pornography, terrorist documents, stolen intellectual properties) in cloud storage to keep the personal computer clean. There have been incidents where suspects stored child pornographic materials and stolen intellectual properties in clouds [5, 6]. It was reported recently that to launch Distributed Denial of Service (DDoS) attacks, adversaries are now placing a new Linux DDoS Trojan – *Backdoor.Linux.Mayday.g* in the compromised Amazon EC2 virtual machines (VM) and launch attacks from those VMs [4]. To investigate such crimes involving clouds, investigators have to carry out a digital forensic investigation in the cloud environment. This particular branch of forensics has become known as *Cloud Forensics*.

Unfortunately, many of the assumptions of traditional digital forensics are not valid in the cloud computing model. One of the major hurdles is that neither users or nor investigators have physical access to the cloud. Even with a subpoena, law enforcement agents cannot confiscate a suspect's computer and get access to the digital evidence. In clouds, each server contains files from many users. Hence, it is not feasible to seize servers from a data center without violating the privacy of many other users. The trustworthiness of the evidence is also questionable, because other than the cloud service provider's (CSP) word, there is no usual way to determine the integrity of the evidence. To provide on-demand services, cloud providers do not support persistent storage for terminated VMs. Hence, data resides in the cloud VMs will be unavailable after terminating the VMs. This in turns makes it

almost impossible to do forensics investigation if some illegal activities have been occurred using such terminated VMs. Finally, cloud providers and investigators can collude with a malicious user to hide the trace of an illegal activity or to frame an honest user. For these reasons, we need special cares to provide reliable forensics supports in current cloud infrastructures.

We argue that to support trustworthy forensics in clouds, we need to preserve logs, proof of data possession, provenance information, and timestamp securely. The required evidence should also be easily available to users, investigators, or the court authority. Based on the requirements, we propose a forensics-friendly architecture – *FECloud*, which is designed on top of the current OpenStack[1] architecture. *FECloud* introduces five new components in the existing OpenStack architecture namely: Logger (Themis), Data Possession (DP) Manager (Metis), Timestamp Manager (Chronos), Provenance Manager (Clio), and Proof Publisher (Brizo). We also add new modules with the OpenStack Block Storage (Cinder) and Nova Compute to communicate with the new components. OpenStack Dashboard (Horizon) and Identity manager (Keystone) are augmented to provide user interface (UI) and authentication to the proposed components. Finally, we design a forensics-enabled image for VMs to support the forensics related features. By incorporating the proposed architecture, cloud providers may attract more customers with the assurance of reliable forensics support. Customers also do not need to establish a privately-owned, costly forensics-enabled computing/storage infrastructures for critical business applications.

While there are several research works, which addressed the challenges of cloud forensics [7, 8, 9, 10, 11, 12] and proposed solutions to overcome some of the problems [7, 13, 14, 15], we do not see any complete architecture that can preserve and provide trustworthy evidence to law enforcement agencies. Recently, Dykstra *et al.* implemented FROST, a tool for OpenStack to collect virtual disks, API logs, and guest firewall logs [16]. Patrascu *et al.* proposed a logging framework for cloud architecture [17]. These works mainly focus on making the logs easily available. However, there are other important evidence besides logs, such as data possession, provenance, timestamp of evidence, which have not been addressed in these works. Moreover, availability of evidence is one of the many challenges of cloud forensics; there are several other issues that should be considered while

[1] http://www.openstack.org/

preserving information securely, e.g., malicious cloud providers, collusion between different stakeholders, which have not been addressed in these works.

Contribution: The contributions of this work are as follows:

1. We systematically analyze the requirements and challenges to establish trustworthy forensics supports in current cloud architectures, which can help researchers to focus on specific research sub-problems of the large cloud forensics problem domain.
2. We present the *FECloud* architecture which provides the required properties for reliable forensics investigations in clouds. To the best of the authors' knowledge, this is the first work that proposes a complete, trustworthy, and forensics-enabled cloud architecture on top of a popular open source cloud computing platform.
3. We show how various proposed components of the architecture acquire different types of evidence from the existing architecture, preserve them securely, and make them available to users and law enforcement agencies. We also show how the trustworthiness of various evidence can be verified by the court authority.

Organization: The rest of the paper is organized as follows: Section 2 presents the research motivation. In Section 3, we present the required properties for a forensics-enabled cloud infrastructure. Section 4 discusses the challenges to establish a forensics-enabled cloud architecture. In Section 5, we propose the *FECloud* architecture. Section 6 presents the related work and finally, we conclude in Section 7.

2 Research Motivation

Through out history, it has been observed that general technological developments have continually created new opportunities for criminal activity, and this is also true for the emergence of cloud computing. The black-box nature of clouds can attract criminals to launch various kinds of malicious activities using clouds. While digital evidence is being used to implicate or exonerate a person, unreliable and inaccurate data can also impact an individual's liberty and life. Hence, when prosecuting any criminal incident, having incorrect information can sometimes be more damaging than having no information. In an ideal situation, we should evaluate digital evidence based on the reliability of the system and process that generated the evidence. As suggested by Strong *et al.*, digital evidence are not the counterpart of statements provided by humans, which should ideally be tested by

cross-examination, whereas admissibility of the digital evidence should be determined on the basis of the reliability and accuracy of the process involved [18]. Hence, preservation and collection of trustworthy evidence is utterly important in digital forensics and as well as in cloud forensics.

However, the very black-box nature of cloud computing also raises questions about the trustworthiness of evidence, when the evidence are collected from the cloud. Because of the importance of reliable forensics in the cloud, the National Institute for Standards and Technology (NIST) recently established the NIST Cloud Computing Forensic Science Working Group (NCC-FSWG) to research cloud forensic science challenges and to develop solutions, standards, and technologies to mitigate the challenges that cannot be handled with current technologies and methods [19, 20]. Many of the challenges of reliable forensics are new in the cloud, while others are already faced by digital forensics examiners. According to Ruan *et al.*, challenges to cloud forensics can broadly be categorized into technical, legal, and organizational challenges [21].

Currently, extensive research is ongoing to protect cloud form external or internal attackers. However, in case of an attack, we need to investigate the incident. Besides protecting the cloud, it is important to focus on this issue. The goal of this work is to enable forensics investigation in clouds while ensuring the trustworthiness of evidence in the complete life cycle of cloud forensics, from evidence preservation to presentation.

3 Properties of a Trustworthy Forensics-Enabled Cloud

Based on the unique characteristics of clouds and existing forensics framework, we identify the following five crucial properties that a trustworthy forensics-enabled cloud should ensure.

3.1 Trustworthy Log Management

Activity logs of cloud users can reveal the actions taken by a user using cloud infrastructures. Therefore, activity logs are crucial pieces of evidence to prosecute a suspect. While the necessity of logs is indisputable in forensic investigation, the trustworthiness of this evidence will remain questionable if we do not take proper measures to secure them, because it is often the case that experienced attackers first attack the logging system [22, 23]. An adversary can try to host a botnet server, spam email server, or phishing websites in cloud VMs and he can remove all the traces of these malicious activities later by tampering

with the logs. We have already mentioned that attackers are now placing a new Trojan in the compromised Amazon EC2 virtual machines (VM) and launch DDoS attacks from those VMs [4]. For these attacks, we need appropriate logs to investigate how an attack occurred, when it occurred, and by whom. Hence, a forensics enabled cloud should acquire all types of activity logs from VMs and store them in a persistent storage while ensuring the integrity and confidentiality of the logs.

3.2 Proof of Data Possession

Preserving proof of data possession is important for two reasons: to prove the presence of a particular file in a given storage system at a particular time and to ensure the preservation of litigation hold.

To keep the personal computer clean, malicious users can keep their contraband documents in clouds. Some incidents of storing contraband documents in cloud storage have already been reported [5, 6]. In [5], the criminal stored child pornographic documents and in [6], the suspect stored some stolen intellectual properties in cloud-baser storage. In order to prove that the suspect actually had some specific files in a given storage at a particular time, a forensics-enabled cloud should preserve the proof of data possession.

A *litigation hold* is a legal notice to a defendant that triggers the preservation of electronically stored information (ESI), which may require the termination of the routine operation of an information system to suspend the normal destruction of ESI [24]. A major difference for litigation holds on cloud-based ESI is that users' data is now under the direct control of a third party – the CSP [25]. FRCP 34(a)(1) states about the preservation of evidence, which are under control or possession of the defendants [26]. However, there are various cases, where a third party was involved to store ESI, but the court determined that the defendant was still in possession or control of the ESI, since they had or should have had the ability to obtain the requested data from the third-party [27, 28, 29]. According to Information Law Group, though the cloud-based ESI is under the possession of the CSP, relevant ESI within the possession or control of a third party may be obtained by serving a subpoena upon the third party, including a CSP [30]. Moreover, the CSP may be the subject of a civil subpoena, government agency demand, or a governmental subpoena directly [25]. The need of litigation hold in clouds has appeared in two recent cases [6, 31]. Hence, if a case involves cloud-based ESI, customers and CSPs share a mutual need and duty to ensure

the litigation hold on cloud data. Whether a litigation was maintained or violated can be ensured by preserving trustworthy proof of data possession.

3.3 Secure Timestamp

It has been seen in the past that the time associated with digital evidence can be very crucial to discharge or condemn a suspect. For instance, the primary suspect of a 1995 homicide case claimed that he was at work at the time of the murder and his alibi mostly depended on the last configuration time of a Fastpath network device. [32]. The suspect had full control over the management console and the log identifying the last configuration time of the device was only recorded on the management console. Hence, it was possible for the suspect to reset the time on the device from the management console that supports his alibi. Therefore, though the last configuration time of the network device agreed with the suspect's alibi, the investigators believed that the evidence had been altered after the crime. Since the management console was not collected during the initial search and seizure, there was a high degree of uncertainty in the timestamp, the suspect's alibi could not be confirmed, and he was convicted of the murder.

A similar situation can happen with the cloud where a crime has occurred directly using cloud computing resources, or where cloud activities can be used as evidence for any other crime. The existing schemes on secure event logs assume that the timestamps provided by the system clock of cloud host machines and guest VMs are trustworthy. However, an attacker can change a VM's system clock before launching any attack and later reset it to the original time. Tampering with the system clocks of the host and VM will provide a set a events which are temporally coherent but occurred in a different time domain than the actual. Any timeline of events generated with the assumption of trusted system clock of host and VM suffers from this vulnerability. Hence, the timeline will not be admissible in a court of law. To provide a trustworthy timeline of events for a reliable forensics investigation, we need to make sure that the system clocks of the host and guest VMs have not been tampered with.

3.4 Secure Provenance

Provenance provides the history of an object. Hence by implementing cloud provenance, investigators can reason about the origins, collection or creation, evolution, and use of any evidence. Besides preserving the integrity of evidence, CSPs also need to provide a proper

chain of custody information, which can be accomplished by maintaining proper provenance. CSPs can provide the chronological access history of evidence, how it was analyzed, and preserved, which can ensure the chain of custody for cloud forensics. However, as all the evidences and the access histories are under the control of CSPs, they can always tamper with the provenance record. Moreover, from the provenance data of the clouds, an attacker can learn confidential information about the data stored in the cloud. To protect provenance information from these types of attack, we need a secure provenance [33] scheme.

3.5 Availability of Evidence

The physical inaccessibility of the evidence makes evidence acquisition a challenging task in the cloud. Considering CSPs are preserving all the evidence, investigators will be still dependent on CSPs to collect evidence, since all the ESI reside in the cloud providers' data center. CSPs can play a vital role in this step by providing a web based management console or providing secure Application Programming Interface (API) to law enforcement agencies. Using a web console or API, customers and investigators can collect network, process, and database logs, as well as other digital evidence and the provenance records of that evidence.

4 Challenges

Integrating the required properties in current cloud infrastructures is challenging due to several characteristics of cloud computing.

Collusion Between Different Entities: In traditional computer forensics, investigators have full control over the evidence (e.g., router logs, process logs, and hard disk). Whereas, users or investigators have very limited control over the evidence stored in clouds. Hence, one of major challenges of establishing trustworthy forensics support in cloud infrastructures is the dependency on the cloud providers, who are not necessarily completely honest. With the state-of-the-art frameworks for collecting evidence from a cloud, investigators need to believe the CSPs blindly, as they cannot verify whether the CSPs are providing valid evidence or not.

The employee of a cloud provider, who collects data on behalf of investigators is most likely not a licensed forensics investigator and it is not possible to guarantee his integrity in a court of law. An employee of a CSP can collude with a malicious user to hide important evidence or to inject invalid evidence to prove the malicious user as innocent.

Such a malicious CSP can provide incomplete logs, remove documents without keeping any trace, can maintain false timestamp, and can tamper with various provenance information. Conversely, investigators can also be malicious and can alter any kind of evidence before presenting to court. In a traditional system, only the suspect and the investigator can collude. The three-way collusion in clouds certainly increases the attack surface and makes cloud forensics more challenging.

Volatile Data: Volatile data cannot be sustained without power. Data that reside in a VM are volatile, as after terminating a VM, no data will be preserved. The volatile data can be documents, network logs, operating system logs, and registry logs. In order to provide the on demand computational and storage services, CSPs do not support persistent storage to a VM instance. Hence, if an adversary terminates VMs after doing a malicious activity (e.g., launch DoS attack, send spam mail) that will lead to a complete loss of the crucial evidences, such as logs, information about data possession or provenance. Though there is a way to preserve VM data by storing a image of the VM instance, an attacker will not use it in order to remain clean.

CSPs can constantly monitor all the running VMs and store the volatile data in a persistent storage so that they can provide logs or proofs of data possession when needed. However simply preserving all data of a terminated VM can overwhelm the storage of cloud providers. Hence, we need to find an effective way to preserve the logs, data possession history or the provenance records.

Multi-tenancy: Cloud computing is a multi-tenant system, while traditional computing is single owner system. To give an analogy, a cloud can be compared to a motel, while the other can be compared to a personal house. In clouds, multiple Virtual Machines (VM) can share the same physical infrastructure, i.e., data for multiple customers can be co-located. An alleged user may claim that the evidence contains information of other users, not her's. In this case, the investigator needs to prove it to the court that the evidence presented actually belongs to the suspect. Conversely, in a traditional computing system, the owner is solely responsible for all the ESI located in her computing system. Moreover, in clouds, we need to preserve the privacy of other tenants. The multi-tenancy characteristic also brings the possibility of side-channel attacks [34] that are difficult to investigate.

5 FECloud Architecture

To preserve and provide cloud-based ESI to forensics investigators securely, we propose *FECloud*, a forensics-enabled cloud architecture. *FECloud* introduces following five new components with the existing OpenStack architecture:

1. *Logger (Themis)* collects logs from VMs, Cinder, and Nova compute nodes and preserves them securely;
2. *Data Possession (DP) Manager (Metis)* collects proofs of data possessions from Cinder and stores the proof securely;
3. *Timestamp Manager (Chronos)* handles timestamp verification cycles between VMs, compute node, and itself and preserves the information about the verification phase securely;
4. *Provenance Manager (Clio)* collects various provenance records (e.g., data, application, state) from VMs, Nova compute, and Themis to securely create and preserve provenance chain;
5. *Proof Publisher (Brizo)* makes the proof of different types of ESI publicly available so that any alteration of evidence by users, CSPs, or investigators after-the-fact can be detected at the court.

Besides the aforementioned new components, we augment the OpenStack Dashboard (Horizon) and Identity manager (Keystone) to support UI and authentication to the proposed components. We also design a forensics-enabled image for VMs to provide the forensics related features. Figure 1 illustrates the proposed *FECloud* architecture and below we describe the design in detail.

5.1 Logger (Themis)

The Logger (Themis) communicates with the OpenStack compute node (Nova), block storage (Cinder), and the running VMs to collect all possible activity logs. To communicate with Themis, a new log provider module is added to the existing Nova and Cinder node of OpenStack and with the VM image.

The log provider module of Nova compute constantly monitors some of the activities (e.g., network activity, processor usage by VMs) of VMs running in the compute node and sends the logs to Themis. Some of the logs of VMs cannot be gathered from Nova compute, e.g., operating system logs. Such logs are directly sent from VMs to Themis. The

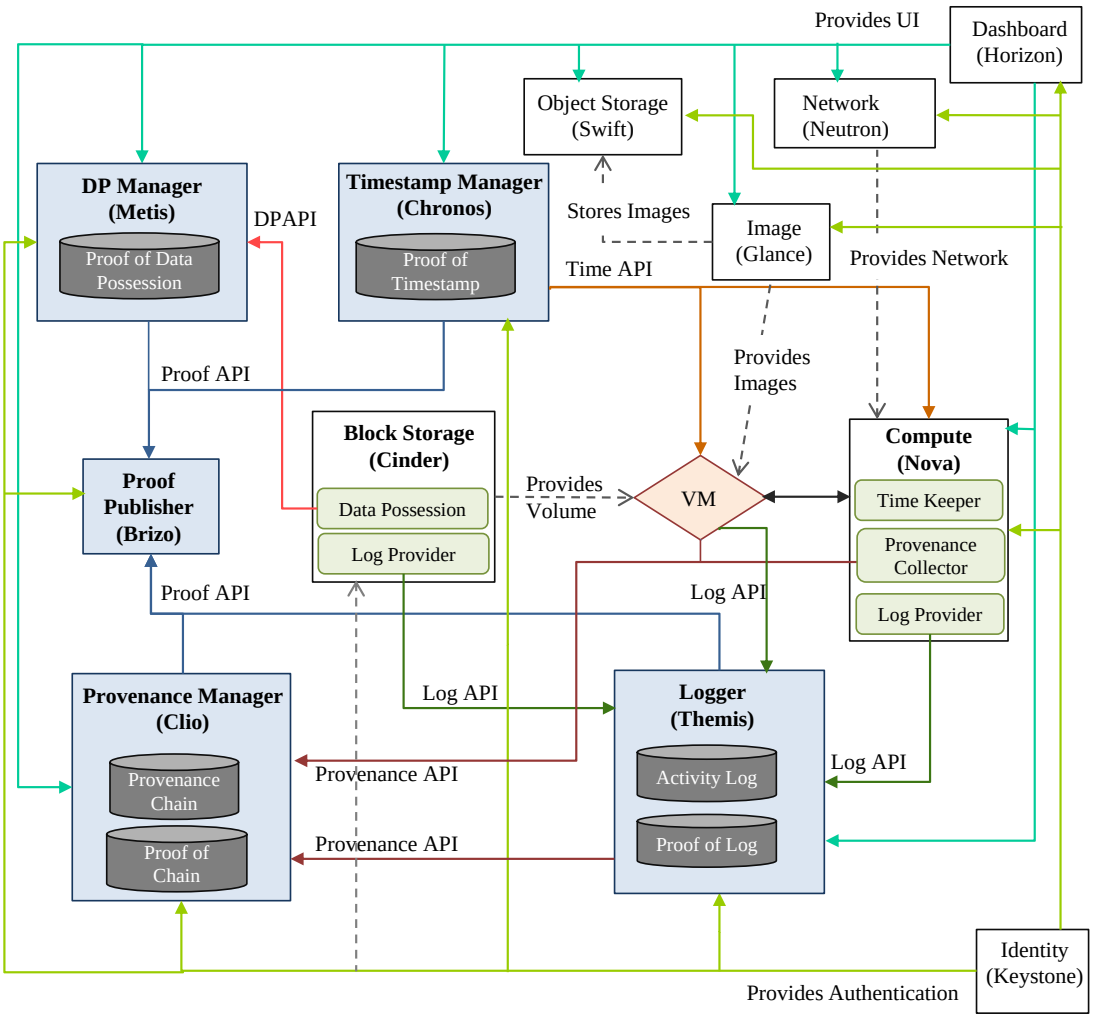

Figure 1: *FECloud* Architecture

log provider module of Cinder sends logs of block storage usage to Themis. Logs from different entities are sent to Themis by the Log API exposed by Themis.

Whenever Themis receives any logs via the Log API, it stores the logs in a persistent storage, so that we will not lose any log after terminating the VMs. When a VM is in active state, Themis can track which data belongs to which VM. Hence, while preserving the data, Themis can take care of segregating the data according to VM owners. In this way, multiple VM owners' data will not be co-mingled. We also preserve the confidentiality of the data from malicious cloud employee by using public-private key based encryption, so that only users and law enforcement agencies can view the data.

To prevent the collusion between CSP, investigators, and cloud users, Themis creates cryptographic proof of the logs using an accumulator data structure, such as a One-Way accumulator [35] or Bloom filter [36]. The proof of the logs is stored in the proof of log

database. By using such proofs, the court authority can verify whether the logs presented to the court are valid or not.

5.2 DP Manager (Metis)

The data possession manager (Metis) collects information about data possession (DP) from Cinder and stores the proof of data possession in a proof of data possession database. Cinder is augmented with a new data possession module to communicate with this new component. Metis takes care of two issues: preserve proof of past data possession and preserve proof of litigation hold maintenance.

A naive way to preserve the proof of DP is to store all the data in a persistent storage. However, this will increase the storage cost significantly. An efficient way to preserve the proof of data possession is using accumulator data structures [36, 35]. Using accumulators, we can store the proof of a thousand blocks of files in a single data structure, which requires significantly lower amount space compared to preserve the blocks. Moreover, using an accumulator, Metis can preserve the proof of data possession without revealing the original data. When the court authority needs to verify whether some incriminating documents belongs to a suspect, first it collects the proof of DP of the suspect. Then using the membership checking method of the accumulator, they can verify whether the documents in question exist in the proof or not.

The proof of DP can also be used in the court to determine the violation of litigation holds. A defendant needs to present all the documents that are under a litigation hold to the court. The verification method first creates DP information of the documents provided by the defendant. It then compares the generated DP information with the proof of DP collected from the cloud. Any deletion of documents by the defendant can be detected if the generated DP information does not match.

5.3 Timestamp Manager (Chronos)

Since it is not possible to block a VM owner to change the system time of a VM, or a malicious system administrator to change the system time of a compute node, we propose a tamper-evident scheme using the Timestamp Manager (Chronos) to protect alterations of a VM's or Nova compute node's timestamp.

A secure timestamp verification protocol runs between three entities: Nova compute node, running VMs, and the timestamp manager (Chronos). In this stage, each entity

verifies the timestamp of others so that timestamp alteration by any entity can be detected by others. Later, traces about the verification phase are stored securely using hash-chain scheme in a proof of timestamp database. Before beginning the verification cycle, VM and Chronos determine the Round Trip Time (RTT) with Nova compute. Validity of a requestor's timestamp depends on the current timestamp of verifier and the RTT values. The timestamp of one requestor is attested by two other entities and each attestation is later certified by an entity other than the verifier and requestor. A new Time Keeper module is added to the Nova compute node to handle the timestamp verification phase. Public key encryption and signature generation take place in all the communications to preserve the integrity and confidentiality of the scheme.

With this new feature included with the cloud, investigators can now present the trace of timestamp verification stage to the court besides the evidence collected from the cloud. As the timestamp verification information is preserved using hash-chain scheme, adversaries cannot change the system time without breaking the chain of the verification information. However, alteration of the verification chain can be detected at the court while verifying the integrity of the chain. Hence, integrating Chronos will ensure that the timestamp associated with the evidence is trustworthy.

5.4 Provenance Manager (Clio)

The secure provenance manager (Clio) extracts provenance record of data, application, and VM state from the log database as well as from the provenance layer of Nova compute, and the running VMs. Since the Logger node (Themis) collects logs of data modification from the block storage, Clio collects necessary log records to build the data provenance from Themis through the Provenance API. Provenance records for the virtual file system and applications running inside the VMs are directly collected from VMs using the same API. Finally, provenance records to create system level provenance of Nova compute, is collected from the provenance layer of Nova.

After collecting various provenance records, Clio applies secure provenance chaining [37] to preserve the integrity of the provenance records. The provenance chain database stores the secure provenance information. To ensure that a malicious CSP cannot modify the chain after-the-fact, head of the provenance chain will be stored periodically to the proof of chain database after some certain epoch.

5.5 Proof Publisher (Brizo)

The proof publisher node (Brizo) periodically publishes the proof of logs, data possession, timesstamp verification, and provenance chain publicly on the web. When the proof is publicly available, CSP or investigator cannot alter any ESI or provide fake evidence, since proofs of those fake evidence will not exist in the published proof. Hence, regularly publishing the proof ensures the forward integrity of the evidence, i.e. none of the ESI for which the proof is already published cannot be altered by any of the entities.

Information published by Brizo will be available by RSS feed to protect it from manipulation by the CSP after publishing the proof. We can also build a trust model by engaging other CSPs in the proof publication process. Whenever one CSP publishes a proof, that proof will also be shared among other CSPs. Therefore, we can get a valid proof as long as more than 50% of the CSPs are honest.

5.6 Access to Evidence Through Horizon

To mitigate the problem of the availability of cloud-based ESI, all the ESI can be collected from the OpenStack dashboard (Horizon). Hence, investigators will not need the physical access to cloud infrastructures to acquire logs, data possession, or provenance information. Four new modules are added with Horizon to provide user interface (UI) for Metis, Chronos, Clio, and Themis, where one module is dedicated for one component. Using these modules, users, investigators, and the court authority can collect activity logs, provenance, proof of data possession, and proof of timestamp.

5.7 Forensics-enabled Image

From only Nova compute and Cinder, we cannot acquire all the evidence required for prosecuting various types of criminal incidents. Hence, without introducing new capabilities to the VMs, it will not be possible to develop a forensics-enabled cloud completely. We proposes a forensics-enabled image for VMs, which is illustrated in Figure 2. A VM, launched using such image can support the required forensics features. Some of the modules that we propose will be inside the kernel, while some others are inside the application layer. Below we describe the functionalities of these modules:

• *Virtual File System (VFS) Monitor:* This module is placed inside the kernel to trace the operations on VFS, which are mostly important to build the data provenance of the VM.

• *System Call Tracer:* This module is also placed inside the kernel to track all the system call. System call information can reveal the activity of cloud users and are also important to build the application and state provenance.

• *Kernel Communicator:* This module resides inside the application layer and performs as a bridge between the kernel and the application layers. This module is responsible to collect information from the virtual file system monitor and the system call tracer module of the kernel and feeds the information to the other modules of the application layer.

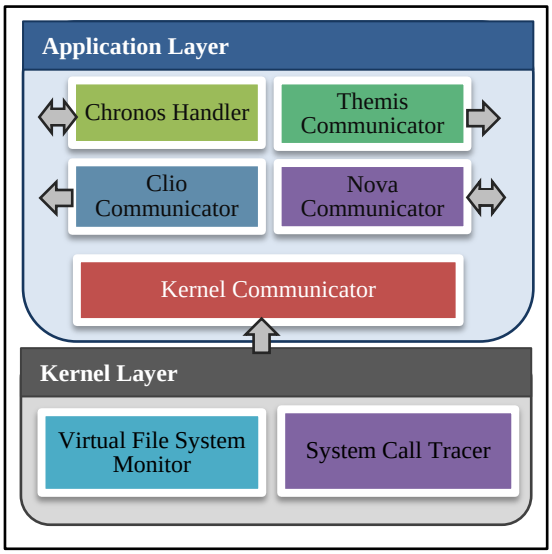

Figure 2: A Forensics-enabled VM Image

• *Chronos Handler:* This application layer module participates in the timestamp verification step to verify the timestamp of Nova compute node and Chronos node and also to get its own timestamp verified by the other two entities.

• *Themis Communicator:* This module first collects the system call information and VFS activity from the Kernel communicator and sends these logs to Themis using the log API.

• *Clio Communicator:* This module sends provenance records for applications, VFS, and VM state to Clio using the provenance API. The provenance records are collected from the VFS monitor and system call tracer via the kernel communicator module.

• *Nova Communicator:* Nova communicator is required to communicate between the compute node and VMs. This communicator module is required in the timestamp verification phase where Nova compute node and a VM verifies each others' timestamp.

5.8 Preliminary Results

Till now, we developed cryptographic frameworks for Metis [12] and Themis [15]. In [12], we designed a Bloom filter based data possession scheme – PPDP for storage-as-a-service cloud. We determined that after integrating the proposed scheme, a user can face 3.73 to 0.13% overhead in terms of time to upload files based on the file size and security properties. This overhead actually decreases with the increase in file size and becomes

almost constant when the file size crossed 6 MB. The storage overhead on cloud provider's side is also low. Regardless of the file size, PPDP requires approximately 1262 bytes to preserve the proof of 1000 files.

In [15], we proposed a secure logging scheme for Themis. The required security properties are ensured by the use of accumulator and hash-chain scheme. We used Bloom filter and RSA accumulator for this work. We found that while RSA accumulator provides better security than Bloom filter, it suffers from higher overhead in terms of log insertion, verification, and storage. Our design supports O(n) time and space complexity for log insertion and storage. The verification algorithm takes constant amount of time to verify logs using both of the accumulator schemes.

Our ongoing work on securing the system time of Nova compute and VMs using the help of Chronos reveals that we can execute the timestamp verification cycle between these three entities in every 60 seconds while introducing less than 1% system overhead on each entities. We run the verification protocol between 20 VMs and one Nova compute and one Chronos for 24 hours with a verification frequency of 60 seconds and found that the system is 99.98% stable.

6 Related Work

Cloud forensics is a relatively new topic. Several researchers have proposed solutions to overcome some of the challenges of cloud forensics. Delport *et al.* focused on isolating an instance to mitigate the multi-tenancy issue [38]. Isolation is necessary because it helps to protect evidence from contamination. Virtual Machine Introspection (VMI) can also be helpful in forensic investigation. In [39], Hay *et al.* showed that if a VM instance is compromised by installing some rootkit to hide the malicious events, it is still possible to identify those malicious events by performing VMI.

As a solution for forensic investigation in clouds, Zafarullah *et al.* proposed logging provided by OS and the security logs [40]. They were able to monitor the Eucalyptus behavior and log all internal and external interaction of Eucalyptus components. To make the network, process, and access logs available to customers, Bark *et al.* proposed to expose read-only APIs by CSPs [7]. By using these APIs, customers can gather valuable information and can provide this to investigators. Zawoad *et al.* proposed a Secure Logging-as-a-Service to securely store VM activities, which ensures integrity and confidentiality

of logs from a malicious CSP and investigators [15]. To detect temporal inconsistencies in a VM's timeline, Thorpe *et al.* developed a log auditor by using the 'happened before' relation [41] in the cloud environment [14].

Patrascu *et al.* proposed a cloud architecture to monitor the activities in the cloud environment [17]. Using the proposed framework they collected logs from different layers of the cloud. They also presented a data center topology to deploy the proposed architecture. Recently, Dykstra et al. implemented FROST, a forensic data collection tool for OpenStack [42]. Using FROST, cloud users/investigators can acquire an image of the virtual disks associated with any of the user's virtual machines, and validate the integrity of those images with cryptographic checksums. It is also possible to collect logs of all API requests made to CSP and OpenStack firewall logs for VMs. FROST is integrated with the OpenStack Horizon. While these two works are big steps towards providing forensics support in the cloud, these works considered the cloud service providers as honest. However, in an adversarial situation, CSPs as well as investigators can be malicious and can tamper with the logs. Hence, trustworthiness of the data collected through the proposed architectures will be questionable. Moreover, data possession, provenance, and timestamp are crucial pieces of evidence to determine the facts about a criminal incidents, which were not addressed in the proposed architectures.

Hasan et al. first introduced secure data provenance [43] and later they proposed a solution to ensure secure data provenance [37]. Provenance for cloud computing is relatively new research area and was first proposed by Muniswamy-Reddy et al. [44]. The same research group proposed a solution for gathering provenance data from Xen Hypervisor [45]. Lu et al. introduced the concept of secure provenance in cloud [46]. They proposed a Trusted Third Party (TTP) based scheme for secure cloud provenance, which can ensure the confidentiality of the data, unforgeability and full anonymity of the signature, and full traceability from a signature. Bates et al. showed how we can use provenance metadata to ensure secure access control of cloud data [47]. In [48] author proposed to build a legal hold framework in clouds. The framework receives legal hold information indicating a legal hold applicable to modification or deletion of a document. However, this patent did not focus on trustworthy management of litigation hold to protect hold from dishonest CSP, defendant, or plaintiff.

7 Conclusion and Future Work

Collecting trustworthy evidence from the cloud is challenging as we have very little control over clouds compared to traditional computing systems. Till now, investigators have to depend on the CSP to collect evidence from clouds. To make the situation even worse, there is no way to verify whether the CSP is providing the correct evidence to the investigators, or the investigators are presenting valid evidence to the court. To enable trustworthy forensics support in current cloud architectures, we identified the required properties and the challenges associated with these properties. We designed *FECloud* – a forensic enabled cloud architecture – on top of OpenStack, which is a widely used open source cloud computing framework. Implementing our proposed architecture will preserve the trustworthiness of evidence and will eliminate the dependency on CSP. However, the CSPs need to come forward to adapt such forensics-enabled architecture.

Currently, we are working to design an efficient secure cloud provenance scheme for Clio and integrate all the proposed components with OpenStack. After integrating all the components, we will evaluate the overhead and stability of different components of OpenStack using standard benchmark tools of OpenStack, such as Rally[2], which will determine the feasibility of using the proposed architecture in a real cloud scenario.

Acknowledgment

This research was supported by the National Science Foundation CAREER Award CNS-1351038, a Google Faculty Research Award, and the Department of Homeland Security Grant FA8750- 12-2-0254.

References

[1] Gartner, "Gartner says that consumers will store more than a third of their digital content in the cloud by 2016," http://goo.gl/39a0y, 2012.

[2] Market Research Media, "Global cloud computing market forecast 2015-2020," http://goo.gl/AR3FBD, [Accessed May 5, 2014].

[3] The Register, "Amazon cloud hosts nasty banking trojan," http://goo.gl/xGNkNO, 2011, [Accessed July 9th, 2014].

[4] Infosecurity-magazine, "Ddos-ers launch attacks from amazon ec2," http://goo.gl/vrXrHE, July 2014, [Accessed September 25, 2014].

[5] www.bbc.com, "Lostprophets' Ian Watkins: 'Tech savvy' web haul," http://goo.gl/C8FVnC, December 2013.

[2] https://wiki.openstack.org/wiki/Rally

[6] Dist. Court, SD Texas, "Quantlab technologies ltd. v. godlevsky," Civil Action No. 4: 09-cv-4039, 2014.

[7] D. Birk and C. Wegener, "Technical issues of forensic investigatinos in cloud computing environments," *Systematic Approaches to Digital Forensic Engineering*, 2011.

[8] J. Dykstra and A. Sherman, "Understanding issues in cloud forensics: Two hypothetical case studies," *Journal of Network Forensics*, vol. b, no. 3, pp. 19–31, 2011.

[9] G. Grispos, T. Storer, and W. Glisson, "Calm before the storm: The challenges of cloud computing in digital forensics," *International Journal of Digital Crime and Forensics (IJDCF)*, 2012.

[10] D. Reilly, C. Wren, and T. Berry, "Cloud computing: Forensic challenges for law enforcement," in *proceedings of the Internet Technology and Secured Transactions (ICITST) Conference*. IEEE, 2010, pp. 1–7.

[11] S. Zawoad and R. Hasan, "Digital forensics in the cloud," *The Journal of Defense Software Engineering (CrossTalk)*, vol. 26, no. 5, 2013.

[12] ——, "Towards building proofs of past data possession in cloud forensics," *ASE Science Journal*, 2012.

[13] J. Dykstra and A. Sherman, "Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques," *DoD Cyber Crime Conference*, January 2012.

[14] S. Thorpe and I. Ray, "Detecting temporal inconsistency in virtual machine activity timelines." *Journal of Information Assurance & Security*, vol. 7, no. 1, 2012.

[15] S. Zawoad, A. K. Dutta, and R. Hasan, "SecLaaS: Secure logging-as-a-service for cloud forensics," in *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security (ASIACCS)*. ACM, 2013.

[16] J. Dykstra and D. Riehl, "Forensic collection of electronic evidence from infrastructure-as-a-service cloud computing," *Rich. JL & Tech.*, vol. 19, p. 1, 2012.

[17] 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. 80–88, 2014.

[18] J. W. Strong and K. S. Broun, *McCormick on evidence*. West Publishing Company, 1992.

[19] P. Mell and T. Grance, "Nist cloud computing forensic science challenges," *Draft NISTIR 8006*, June 2014.

[20] collaborate.nist.gov, "Cloud forensic science," http://collaborate.nist.gov/twiki-cloud-computing/bin/view/CloudComputing/CloudForensics, [Accessed July 5th, 2014].

[21] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, "Cloud forensics: An overview," in *proceedings of the 7th IFIP International Conference on Digital Forensics*, 2011.

[22] M. Bellare and B. Yee, "Forward-security in private-key cryptography," *Topics in Cryptology, CT-RSA 2003*, pp. 1–18, 2003.

[23] ——, "Forward integrity for secure audit logs," Technical report, Computer Science and Engineering Department, University of California at San Diego, Tech. Rep., 1997.

[24] A. G. Araiza, "Electronic discovery in the cloud," *Duke L. & Tech. Rev.*, p. 1, 2011.

[25] K. N. Rashbaum, B. B. Borden, and T. H. Beaumont, "Outrun the lions: A practical framework for analysis of legal issues in the evolution of cloud computing," *Ave Maria L. Rev.*, vol. 12, pp. 71–149, 2014.

[26] Federal Rules of Civil Procedure, "Rule 34," http://goo.gl/NfL61.

[27] J. Smith, "Electronic discovery: The challenges of reaching into the cloud," *Santa Clara L. Rev.*, vol. 52, p. 1561, 2012.

[28] Dist. Court, SD New York, "Dietrich v. bauer," F. Supp. 2d, vol 76, page 312, No. 95 Civ. 7051 (RWS), p. 312, 1999.

[29] Court of Appeals, 10th Circuit, "Tomlinson v. el paso corp." F. 3d, Volume 653, pages 1281, no. 10-1385, p. 1281, 2011.

[30] InfoLawGroup LLP, "Legal Implications of Cloud Computing Part 4.5," http://goo.gl/rzJe2e.

[31] Dist. Court, SD Ohio, "Brown v. tellermate holdings ltd." Case No. 2: 11-cv-1122, 2014.

[32] E. Casey, "Error, uncertainty, and loss in digital evidence," *International Journal of Digital Evidence*, vol. 1, no. 2, pp. 1–45, 2002.

[33] R. Hasan, R. Sion, and M. Winslett, "Preventing history forgery with secure provenance," *ACM Transactions on Storage (TOS)*, vol. 5, no. 4, pp. 1–43, 2009.

[34] T. Ristenpart, E. Tromer, H. Shacham, and S. Savage, "Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds," in *Proceedings of the 16th ACM conference on Computer and communications security*, 2009, pp. 199–212.

[35] J. Benaloh and M. De Mare, "One-way accumulators: A decentralized alternative to digital signatures," in *Proceedings of Advances in Cryptology, EUROCRYPT*. Springer, 1994, pp. 274–285.

[36] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.

[37] R. Hasan, R. Sion, and M. Winslett, "The case of the fake Picasso: Preventing history forgery with secure provenance," in *Proceedings of the 7th USENIX Conference on File and Storage Technologies (FASTi09)*. USENIX Association, 2009, pp. 1–12.

[38] M. K. Waldo Delport, Martin S. Olivier, "Isolating a cloud instance for a digital forensic investigation," in *Information and Computer Security Architecture (ICSA)*, 2011.

[39] B. Hay and K. Nance, "Forensics examination of volatile system data using virtual introspection," *ACM SIGOPS Operating Systems Review*, vol. 42, no. 3, pp. 74–82, 2008.

[40] Z. Zafarullah, F. Anwar, and Z. Anwar, "Digital forensics for Eucalyptus," in *Proceedings of Frontiers of Information Technology (FIT)*. IEEE, 2011, pp. 110–116.

[41] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, 1978.

[42] J. Dykstra and A. T. Sherman, "Design and implementation of frost: Digital forensic tools for the OpenStack cloud computing platform," *Digital Investigation*, vol. 10, pp. S87–S95, 2013.

[43] R. Hasan, R. Sion, and M. Winslett, "Introducing secure provenance: problems and challenges," in *Proceedings of the 2007 ACM workshop on Storage security and survivability*. ACM, 2007, pp. 13–18.

[44] K. Muniswamy-Reddy, P. Macko, and M. Seltzer, "Making a cloud provenance-aware," in *Proceedings of the 1st Workshop on the Theory and Practice of Provenance*, 2009.

[45] M. Seltzer, P. Macko, and M. Chiarini, "Collecting provenance via the Xen hypervisor," in *Proceedings of the 3rd USENIX Workshop on the Theory and Practice of Provenance (TaPP'11)*, 2011.

[46] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure provenance: The essential of bread and butter of data forensics in cloud computing," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. ACM, 2010, pp. 282–292.

[47] A. Bates, B. Mood, M. Valafar, and K. Butler, "Towards secure provenance-based access control in cloud environments," in *proceedings of the 3rd ACM Conference on Data and Application Security and Privacy*, 2013.

[48] O. Schmidt, "Managing a legal hold on cloud documents," 2012, uS Patent App. 13/543,254.