Towards Achieving Reliable Digital Forensics in IaaS and StaaS Clouds Using the Open Cloud Forensics Model

Shams Zawoad*, Ragib Hasan*, and Anthony Skjellum**
{zawoad, ragib}@uab.edu, skjellum@auburn.edu
*Department of Computer and Information Sciences
University of Alabama at Birmingham, AL 35294, USA
**Department of Computer Science and Software Engineering
Auburn University, AL 36849, USA

Abstract
The rapid adoption of cloud computing to meet storage and computation needs has changed the way computing services and resources are used. However, because of the black-box nature and multi-tenant usage models of clouds, existing digital forensics science cannot cope with current state-of-the-art cloud architectures. Because of the fundamental characteristics of such clouds, many assumptions of digital forensics are invalidated. In the digital forensics process involving clouds, the role of cloud service providers (CSPs) is extremely important; a role which needs to be considered in the science of cloud forensics. In this paper, we define cloud forensics while considering the role of the CSP and propose the Open Cloud Forensics (OCF) model. The OCF model focuses on preserving sufficient electronically stored information (ESI) required to investigate cases involving clouds and ensuring the trustworthiness of such ESI. Based on this we propose two cloud architectures: the OCF-supported Infrastructure-as-a-Service (IaaS) cloud and the OCF-supported Storage-as-a-Service (StaaS) cloud. We validate our proposed models using a case study, which is inspired from an actual civil lawsuit.

Keywords: Cloud Forensics, Forensics Science, Digital Investigation, Digital Forensics, Cloud Security

1. Introduction
Cloud computing has opened a new horizon in computing. Individuals, business, industry, and government are moving towards clouds because of the high degree of scalability, convenient pay-as-you-go service, and low-cost computing provided by the cloud. The rapid adoption of cloud computing has effectively increased the market value of clouds, which crossed the $100 billion milestone in 2013 (Deeter, 2013) and will continue to grow in the future (IDC, 2012), (INPUT, 2009), (Market Research Media, 2014). According to a report from Market Research Media, the cloud computing market is expected to grow at a compound annual growth rate (CAGR) of 30% and will reach $270 billion in 2020 (Market Research Media, 2014).

On the other hand, since in today’s world most business records (92-99%) are stored electronically (Ruhnka, 2007), the Federal Rules of Civil Procedure (FRCP) have broadened the scope of evidence in the 2006 amendment to include Electronically Stored Information (ESI) to be used in civil litigation (“Rule 34”, n.d.). Because of the rapid adoption of clouds, it is clear that a significant portion of the ESI will be stored in clouds. Notably, there have already been incidents in which the availability of the massive computation power and storage facility of clouds motivated malicious actors to launch attacks from machines inside the cloud (Infosecurity-magazine, 2014), (The Register, 2011), and to store contraband documents in the cloud (Dist. Court SD Texas, 2014), (bbc.com, 2013). It was reported recently that in order to launch Distributed Denial of Service (DDoS) attacks, adversaries have been placing a Linux DDoS Trojan – Backdoor.Linux.Mayday.g in compromised Amazon EC2 virtual machines (VM) and launching attacks from those VMs (Infosecurity-magazine, 2014). For these types of attacks, the forensic examiner needs to execute digital forensics procedures in the cloud in order to determine facts concerning a given incident. These types of forensic investigations are known as cloud forensics.

However, many of the assumptions of traditional digital forensics are invalidated in the cloud computing model. One of the major assumptions of digital forensics procedures and tools is that investigators (or users) have certain physical access to the evidence, which is an invalid assumption in clouds; sometimes it is even impossible to identify the
physical location of cloud-hosted data. In a cloud, each server contains files from many users. Hence, it is infeasible to seize servers from a data center without potentially violating the privacy of other legitimate users. The trustworthiness of such evidence would also be questionable, because other than the Cloud Service Provider’s (CSP) word/warranty, there is no routine way to determine the integrity of the evidence so obtained. To provide on-demand services, cloud providers do not typically support persistent storage for terminated VMs. Hence, data residing in cloud VMs will typically become unavailable after terminating such VMs. This, in turn, renders it almost impossible to perform forensics investigation if some illegal activities have allegedly occurred using VMs that have subsequently been terminated. Finally, cloud providers and investigators can potentially collude with a malicious user to hide traces of an illegal activity or to frame an innocent user. For these reasons, we need to take special care to provide support for reliable forensics in current cloud infrastructures.

While there are several research works that addressed the challenges of cloud forensics (Birk, 2011), (Dykstra, 2011), (Grispos, 2012), (Reilly, 2010) and proposed solutions to overcome certain of the problems (Birk, 2011), (Dykstra, 2012), (Dykstra, 2013), (Thorpe, 2012), a formal model of reliable cloud forensics does not yet exist. To address this gap, we offer a redefinition of the cloud forensics process and propose in particular the Open Cloud Forensics (OCF) model, first proposed in our previous work (Zawoad, 2015). This model considers a new role of the CSP to support reliable digital forensics in the cloud. We argue that to support such reliable digital forensics, a continuous process flow should be executed by the CSP, which is a part of the cloud forensics process and is referred to as continuous forensics (Zawoad, 2015). Based on the OCF model, we then propose forensics-aware cloud computing architectures for Infrastructure-as-a-Service (IaaS) and Storage-as-a-Service (StaaS) clouds. We also validate the proposed architecture for the StaaS cloud using a case study.

Contributions. The contributions of this work are as follows:

1. We extend the existing definition of digital forensics and redefine it in the context of clouds to support reliable digital forensics in the cloud. The new definition of cloud forensics will ideally guide future research in this area.

2. We propose the open cloud forensics (OCF) model, which includes continuous forensics support by the CSP – an integrated part of our cloud forensics definition. The proposed OCF model will ideally inspire future researchers to design forensics-aware cloud computing architectures.

3. While many architectures can be derived and extrapolated from the OCF model, we present two forensics-aware cloud architectures designed on top of OCF model: an OCF-supported IaaS cloud and OCF-supported StaaS cloud. The proposed architectures support the new cloud forensics definition as well as the OCF model. The design of the OCF-supported StaaS cloud is validated by a case study, which is inspired by an actual civil lawsuit.

Organization. The remainder of the paper is organized as follows: Section 2 provides background knowledge of digital forensics and cloud forensics while presenting the motivation behind our work. Section 3 presents our proposed cloud forensics process and the OCF model. In Section 4, we present two OCF-supported cloud architectures. Section 5 discusses how the proposed model can work in a real-life scenario of cloud forensics. In Section 6 we discuss some contemporary research works on cloud forensics and, finally, we conclude in Section 7.

2. BACKGROUND AND MOTIVATION

In this section, we present a brief overview of digital forensics and cloud forensics, and justify why we need a new forensics model for the cloud.

2.1 DIGITAL FORENSICS

![FIGURE 1: DIGITAL FORENSICS PROCESS FLOW (Zawoad, 2013)](Zawoad, 2013)

The National Institute of Standards and Technology (NIST) defines digital forensics as “an applied science to identify an incident, collection, examination, and analysis of evidence data” (Kent, 2006). Maintaining the integrity of the information and a strict chain of custody for the data are mandatory. Several other researchers define digital forensics as the procedure of examining a computer system to
determine potential legal evidence (Lunn, 2000), (Robbins, 2008). From the above working definitions, we can state that digital forensics comprises four main processes: identification, collection, organization, and presentation (Zawoad, 2013). Figure 1 illustrates the flow of the processes of digital forensics, which is briefly described below.

- **Identification**: There are two main steps in the identification phase — identification of an incident and identification of the evidence, which will be required for successful investigation of that incident, with potential correlation to another incident.

- **Collection**: In the collection process, an investigator extracts the digital evidence from different types of media (e.g., hard disk, cell phone, e-mail, and analogous types of data). Additionally, the investigator preserves the integrity of the evidence.

- **Organization**: There are two main steps in the organization process: examination and analysis of the digital evidence. In the examination phase, an investigator extracts and inspects the data and its characteristics. In the analysis phase, he or she interprets and correlates the available data to come to a conclusion, which can serve to prove or disprove civil, administrative, or criminal allegations (when interpreted legally).

- **Presentation**: In this process, an investigator makes an organized report to state his or her findings of the case. This report should be appropriate for presentation to the judge and jury.

### 2.2 Cloud Forensics

The National Institute of Science 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 technology that will mitigate the challenges of cloud forensics, which cannot be handled with current technology and methods (NIST, n.d.), (Mell, 2014). NIST defines cloud forensics as “the application of scientific principles, technological practices and derived and proven methods to reconstruct past cloud computing events through identification, collection, preservation, examination, interpretation and reporting of digital evidence (Mell, 2014).”

Different steps of digital forensics (as shown in Figure 1) and the control over evidence can vary in the cloud according to the service and deployment models of cloud computing. For example, evidence collection procedures in Software-as-a-Service (SaaS) and Infrastructure-as-a-Service (IaaS) are different. In the private cloud deployment model, we can have physical access to the digital evidence, but we rarely can get physical access to the public deployment model.

### 2.3 Motivation

Certain fundamental characteristics of clouds render digital forensics more challenging as compared to dedicated server and system environments. Hence, we need to model the digital forensics in such a way as to overcome the new challenges imposed. Below, we present a hypothetical case in order to illustrate the challenges of digital forensics imposed by clouds.

**Motivating case study.** The motivating (hypothetical) case study is inspired by the Quantlab Technologies Ltd. v. Godlevsky case (Dist. Court SD Texas, 2014). In this case, the plaintiffs brought suit against the defendants for copyright infringement, breach of contract, misappropriation of trade secrets, and fraud. We present the scenario of this real-life case in the following hypothetical case study.

_Mallory worked in a software development company BISoft and there she developed a business analysis algorithm and a business intelligence system for high volume business data. The software proved popular in industry and BISoft reaped huge profits from the system. Though the company maintains strict rules to protect their intellectual property, Mallory managed to export the source code of the developed system to CloudCo’s storage – an arms-length 3rd party CSP. Later, Mallory formed her own company and used substantially the same designs and code to develop a business intelligence system. BISoft filed a case against Mallory accusing her and her company of stealing intellectual property and Bob, a digital forensics investigator, was assigned to determine the facts._

However, the following characteristics of clouds can serve to hinder Bob’s investigation:

**Physical inaccessibility.** Evidence collection procedure is harder in the cloud because of the physical inaccessibility of digital evidence. Existing state-of-the-art digital forensic procedures and tools that the investigator Bob could use in traditional computing systems proves a poor fit with the cloud because the assumption of having physical access to the computing resources (e.g., hard disk, a network router, etc.) is invalid in clouds. Sometimes, we would not even know where the data is located in a large, distributed cloud infrastructure. Location of data is important for many reasons: for instance, a warrant must specify a location, but in a cloud, data may not be located at a precise location or a particular storage server and may transition over time as well. A number of researchers address this issue in their
work (Birk, 2011b), (Dykstra, 2011), (Guo, 2012), (Ludwig, 2012), (Reilly, 2011), (Wolthusen, 2009). Because of the physical inaccessibility, Bob will have less control over the evidence and needs to rely on CloudCo to collect the digital evidence from the cloud computing environment on his behalf. This is a serious bottleneck in the collection phase (and could arguably raise doubts about the chain of custody as well as data integrity and completeness).

**Volatile data.** Data that reside in a virtual machine (VM) are volatile since these data cannot be sustained without power. After terminating a VM, no such data will be preserved, except that which was written to disk. Volatile data can comprise documents, network logs, operating system logs, and registry logs stored on volatile volumes or in memory. In order to provide on-demand computational and storage services, CSPs need not provide persistent storage to a VM instance. Hence, if Mallory chose VMs running on CloudCo to store the stolen code and then terminated the VMs after she developed her own software, that would lead to a complete loss of crucial evidence, such as logs, information about data possession, and/or provenance. Although there is a way to preserve VM data by storing an image of the VM instance, Mallory would definitely not elect to use it in order to reduce her digital footprint, assuming “knowledge of guilt” or simply an avoidance of potential exposure to future forensics drove her behavior while using the CSPs resources.

**Multi-tenancy.** Cloud computing is a single-owner, multi-tenant system, while traditional computing is single owner system (potentially multi-tenant or otherwise single-tenant). To offer an analogy, a cloud can be compared to a motel, while the other can be compared to a person’s home, or to an apartment complex. In clouds, multiple Virtual Machines (VM) routinely share the same physical infrastructure. Hence, in our hypothetical scenario, Mallory’s allegedly stolen code and other documents of legitimate users can be co-located in and on the same storage device(s). Given this property of clouds, it is difficult if not impossible for the investigator to confiscate such a shared storage device without violating the privacy and usage rights of other CloudCo users. Mallory might also repudiate data contained on such storage device as evidence that contains information of other users, rather than her own. In such a case, if Bob finds any trail of the stolen intellectual property from the cloud, he also needs to prove to the court that such evidence presented actually owns to Mallory. Conversely, if Mallory would have stored the documents in her personal computer, that would constitute prima facie evidence that she would be responsible for all the evidence found in her computing system (which Mallory would have to argue against).

**Collusion between different entities.** In traditional digital forensics, investigators have full control over the evidence (e.g., router logs, process logs, and hard disks). Whereas users or investigators have limited control over the evidence stored in clouds. Hence, one of the major challenges of establishing trustworthy forensics support in cloud infrastructures is the dependency on the cloud providers who are not necessarily completely honest (or may inadvertently employ dishonest actors). With state-of-the-art frameworks for collecting evidence from a cloud, Bob needs blindly to take CloudCo’s asssentation as valid, since he cannot verify whether CloudCo is providing valid evidence or not. Such gaps provide opportunities for a defense to raise objections of reasonable doubt in criminal investigations and to impair any “more likely than not” standard in civil litigation.

The CloudCo’s employee, who will collect data on behalf of Bob is most likely not a licensed forensics investigator and it is impossible in any event to guarantee his or her integrity in a court of law. Mallory could potentially collude with that employee of CloudCo to hide important evidence or to inject invalid evidence to mitigate her guilt or help establish her innocence. Such a malicious employee could provide incomplete logs, remove documents without keeping any trace, could maintain false timestamps, and could tamper with various provenance data or meta-data. Conversely, Bob could also be malicious and could alter any kind of evidence before presentation to the court. In a traditional system, only the suspect and the investigator can collude. The potential for three-way collusion in clouds certainly increases the attack surface and makes cloud forensics more challenging.

### 3. Open Cloud Forensics

State-of-the-art digital forensics models do not presently consider CSPs in the investigation process. However, we argue that without defining the role of CSPs in forensics investigations, cloud forensics cannot be defined properly and it may not be possible to execute digital forensics procedures in a trustworthy manner. We introduce the notion of *Continuous Forensics* (Zawoad, 2015) in the cloud forensics model to facilitate the digital forensics procedures. In this section, we first amend cloud forensics by considering the important role of CSPs. We present the threats that exist in the cloud forensics process. Based on our definition of cloud forensics and the threat model, we then propose the Open Cloud Forensics (OCF) model. Following that, two cloud architecture are proposed, which supports this OCF model in order to ensure reliable forensics in the cloud.

#### 3.1 Cloud Forensics Process

We define cloud forensics as the science of preserving all evidence possible while ensuring the privacy and integrity of the information, identification, collection, organization, presentation, and verification of evidence data to determine
the facts about an incident involving clouds. Figure 2 illustrates the proposed cloud forensics process flow.

From Figure 2, we observe that the preservation stage of all possible evidence and the verification of such evidence are introduced with the digital forensics process flow presented in Figure 1. Because of the volatile nature of cloud data and possible manipulation of evidence by malicious cloud providers, we need to include the preservation and verification steps. The preservation stage should always be online/running and, hence, we refer to this as a continuous forensics process.

![Cloud Forensics Process Flow Diagram](image)

**FIGURE 2: CLOUD FORENSICS PROCESS FLOW**

In the verification stage, the court authority will needfully verify the cloud-based evidence provided by an investigator. The verifier will use the information stored in preservation stage to decide on the integrity of the evidence. Trustworthiness of evidence and availability of the volatile data depends on how efficiently and securely we preserve such data.

3.2 Threat Model

A cloud provider may be honest but it might employ disgruntled or malicious personnel with super user access or its machines could be compromised by malicious users, yielding them unauthorized access. The incentive for an employee of the cloud provider to be malicious can have monetary value. Hence, unlike the existing body of work on cloud forensics (Dykstra, 2012), (Thorpe, 2012), (Dykstra, 2013b), we do not consider a priori that the CSP will be honest. In our threat model, users, investigators, and CSPs—all three entities—can be malicious and hence can collude to provide fake or falsified evidence to the court authority.

Collusion between different entities increases the capability of the attackers and hence, expands the attack surface. Ensuring reliability of the evidence becomes more challenging when different malicious entities collude rather than acting singly. For example, a malicious user acting alone cannot modify the evidence stored under the control of CSP unless he/she colludes with the CSP or compromises the CSP in order to alter the evidence. A user can delete evidence that is under his/her control (intentionally or unintentionally) or he/she can provide false evidence to an investigator. However, when the CSP is honest, the investigator can detect at least some of all such alterations of evidence made by a malicious user. On the other hand, if the CSP and the user both provide the same falsified evidence to the investigator, it will be difficult to verify its integrity. It may simply be accepted as valid.

Likewise, an investigator can present false incriminating evidence to the court to frame an honest user, or to save a malicious user from conviction. However, if the dishonest investigator acted alone, such malicious behavior could be detected, when the court authority verifies the evidence provided by the investigator with the evidence stored in the cloud. However, when the investigator colludes with the CSP, it will be difficult for an auditor to determine the trustworthiness of such evidence. This gap itself provides a logical basis for potential defense claims as well.

Moreover, after providing evidence to an investigator, a CSP can potentially repudiate any evidence. As data are commingled in the cloud, a malicious user could claim that particular evidence does not belong to him or her. An intruder, as well as a malicious cloud employee, could acquire the evidence of a user to learn the user’s activity or confidential information.

3.3 The OCF Model

Based on our definition of cloud forensics and the threat model, we propose the Open Cloud Forensics (OCF) model, which is depicted in Figure 3. Below we describe different entities and components of the OCF model.

**Stakeholders.** In the OCF model, four entities are involved: the CSP, user, investigator, and court authority.

- **Cloud Service Provider (CSP):** A CSP provides various services to its customers. In the OCF model, a CSP in its entirety or an employee of the CSP can be malicious.
- **User:** A user enjoys various services provided by the CSP. Let $U$ be the set of all users who are using...
services provided by the CSP. If there are \( n \) such users, \( U = \{ u_1, u_2, \ldots, u_n \} \). A user can be malicious and can also collude with a dishonest CSP or a dishonest investigator.

- **Investigator**: An investigator is a professional forensic expert, who needs to collect necessary electronic information from cloud infrastructures in case of any malicious incident and present the required evidence to the court authority. An investigator could be malicious and if so may collude with users or CSPs.

- **Court authority**: Based upon the evidence provided by an investigator, the court authority decides the innocence or guilt of a suspect.

\[
E_{u_j} = \bigcup_{1 \leq i \leq m} E_{ts_{u_j}}^{s_i}
\]

The complete set of ESI between time \( t_s \) and \( t_e \) for \( n \) number of users, \( E_{ts_{u_j}}^{s_i} \), can be defined as

\[
E_{ts}^t = \bigcup_{1 \leq j \leq n} E_{u_j}
\]

- **Verifiable ESI**: To ensure the integrity and privacy of ESI, the CSP translates all the ESI to verifiable ESI. Hence, for every ESI, \( E_i \), there will be a corresponding verifiable ESI denoted as \( VE_i \).

Let user \( u_j \) be a malicious user, who executed an illegal activity using service \( s_i \) provided by the CSP. An investigator subsequently gathers relevant verifiable ESI from the set of verifiable ESI of user \( u_j \) for service \( s_i \), \( VE_{s_i}^{s_j} \), analyze the ESI, and present evidence to the court.

- **Evidence**: When a set of verifiable ESI is presented to the court to establish the facts about a criminal incident, we refer such verifiable ESI as evidence. Therefore, evidence is a subset of verifiable ESI. Let us assume that user \( u_j \) accessed a service \( s_i \), which produces some possible evidence. In this case, the set of verifiable evidence presented to the court is denoted as \( PE_{s_i}^{s_j} \) and \( PE_{s_i}^{s_j} \subset VE_{s_i}^{s_j} \). The court authority later verifies the integrity of the evidence \( PE_{s_i}^{s_j} \) and rules based on the evidence.

**Components.** The components of the OCF model are as follows: cloud services, ESI, verifiable ESI, and evidence. These are presented below:

- **Cloud Services**: Let us assume that there are \( m \) services provided by the CSP and \( S \) is the set of all services. Hence, \( S = \{ s_1, s_2, \ldots, s_m \} \). In the OCF model, the set of services \( S \) includes but is not limited to software, computing and storage resources, platforms, etc.

- **ESI**: Any types of services (when used by a user) create ESI such as documents, activity logs, file system provenance, and many others. An ESI generated for accessing service \( s_i \) by user \( u_j \) at time \( t \) is described as \( E_{ts_{u_j}}^{s_i} \). Hence, if the user \( u_j \) has access to \( q \) number of services, where \( q \leq m \), then all the ESI of user \( u_j \) between time \( t_s \) and \( t_e \) can be defined as

\[
E_{u_j} = \bigcup_{1 \leq i \leq q} E_{ts_{u_j}}^{s_i}
\]

**Continuous forensics.** The sequence of actions — where a user accesses a cloud service, which in turn generates ESI, and is finally translated into verifiable ESI — are referred to as a continuous forensics process (Zawoad, 2015). The continuous forensics process is marked with a blue dotted line in Figure 3. Without the presence of this continuous forensics support in the cloud, the evidence presented to court may not be considered trustworthy.

4. **OCF-SUPPORTED CLOUDS**

In this section, we present the OCF-supported cloud architectures for Infrastructure-as-a-Service (IaaS) and Storage-as-a-Service (StaaS) cloud models.
4.1 OCF-SUPPORTED IaaS

Crucial evidence in the IaaS model. The IaaS model allows a customer to rent processing power to launch his or her own virtual machine. One of the important features is that the customers can scale up according to their requirement. It allows their applications to handle high load smoothly. On the other hand, they can save cost when demand is low. An example of IaaS is Amazon EC2 ("Amazon EC2", n.d.). EC2 provides users with access to virtual machines (VM) or instances running on its servers. Customers in principle can install most any operating system and run any application in that VM.

A malicious user can utilize the low-cost and highly distributed nature of the IaaS model and launch various attacks, such as hosting a botnet server, spam email server, or phishing websites in a cloud VM, launching DDoS attacks by renting hundreds of VMs or by compromising the VMs running in clouds (Infosecurity-magazine, 2014).

Activity logs of cloud users can be served as crucial evidence to investigate such malicious activities. For example, a digital forensics investigator can analyze the network logs of VMs to identify the source(s) of a DDoS attack. Similarly, registry and process logs can be used to detect a malware or to analyze how a VM may have been compromised. Therefore, while designing the OCF-supported cloud, we focus on preserving various activity logs of cloud users, translating the logs to verifiable ESI, and making the logs available to various stakeholders of a lawsuit.

Architecture. The proposed OCF-supported IaaS cloud is illustrated in Figure 4. Various logs generated in the VM, such as process logs and registry logs are volatile, whereas, data in the host machine are persistent. We introduce the following features with the existing IaaS framework to support the OCF model:

- First, to prevent the loss of volatile logs of the VMs, we propose a continuous synchronization feature, which will store various volatile logs efficiently in a persistent storage without hampering the CSP’s business model.
- Second, to translate the logs to verifiable ESI, we propose a new cryptographic proof generator and proof publisher module that will create cryptographic proofs of all the logs and publish to the Internet, so that neither a dishonest cloud provider nor a dishonest investigator can alter the logs after-the-fact.
- Third, all the activity logs of VMs and cloud host machines will be made available to the investigators through APIs so that investigators need not have physical access to the cloud infrastructure in order to acquire potential evidence.
- Finally, the court authority can use the published proofs of the logs to verify the integrity of the evidence provided by an investigator.

In the following, we describe each of these key features.

1) Continuous Synchronization. Persistent activity logs, which are generated on the host machine will be directly stored in the persistent record storage. Since CSPs do not provide persistent storage to VMs, turning off or rebooting a VM will ultimately lose all the (volatile) data residing in that VM, which may well include crucial activity logs of the VMs, such as process and registry logs. Therefore, we need continuous synchronization
scheme to preserve various VM-generated logs. By preserving the volatile activity logs in a persistent storage, we can gather required evidence even once a malicious user terminates a VM after some illegitimate activities.

One possible solution for continuous synchronization is that CSPs will provide a continuous synchronization API to customers to preserve logs generated in the VM. Using this API, customers can preserve the synchronized data to any cloud storage, e.g., Amazon S3 (“Amazon S3”, n.d.), or to their local storage. However, if the adversary should be the owner of a VM, this mechanism will fail. Trivially, he or she will conveniently choose not to be interested in synchronizing the activity logs of his or her malicious VMs. Therefore, we propose that the CSP will constantly monitor all the VMs running on cloud host machines and store the logs that are volatile in a persistent database.

Logs that are volatile include network, OS, and registry logs. However, we need to carefully select which of the volatile data will be preserved and to what extent. Storing all the volatile logs for a long period of time may not be economical for the CSP. Hence, based on the business model of the CSP and government regulations, we can select the crucial pieces of volatile logs and define a retention period for those data.

When a VM is in its active state, its host machine can track which data belongs to which VM. Hence, while preserving the activity logs of the VMs through continuous synchronization, the CSP can take care of segregating the logs by VM owner. Thus, multiple VM owners’ activity logs will not be commingled. The CSP can preserve the confidentiality of the logs from external adversaries by using public-key encryption, where private keys are only accessible to users and law enforcement agencies. This will also ensure the confidentiality of data from malicious cloud employees.

2) Cryptographic Proof Generator and Publisher. According to the OCF model, we need to translate the activity logs of VMs and cloud host machine to verifiable ESI, which preserve the trustworthiness of the logs even when the CSP, investigators, and cloud users collude in order to mislead a case. For translating the logs stored in the persistent storage to verifiable ESI, we propose a cryptographic proof generator and proof publication module, which will jointly be responsible for generating and publishing the proofs of the logs stored in the persistent storage. The proof will not be the log records; rather we propose to use cryptographic accumulators such as a One-Way (or cryptographic) accumulator (Benaloh, 1994) in order to preserve the proof. Using a cryptographic accumulator has certain benefits. Using accumulators, we can preserve the proof of thousands of activity logs in a single data structure. This will reduce the storage overhead for preserving the proof to a great extent. At the end of each day (or another audit period), the accumulator is signed by the private key of the CSP and the publisher module publishes the generated proof publicly on the Internet so that the cloud provider cannot modify any generated proof after-the-fact.

3) ESI Disclosure. We propose to provide secure read-only APIs to law enforcement agencies and other narrowly authorized parties to disclose various activity logs of VMs and cloud hosts. Only the users, investigators, and the court will have access to these APIs. They can collect the logs stored in the persistent storage through these APIs. To implement this feature, the CSP needs to accommodate an ESI disclosure module, which will communicate with the previously described persistent log storage in order to collect requested logs via an API call. The ESI disclosure module can provide a Representational State Transfer (RESTful) API, where the requested logs are the resource. To retrieve required evidence from such APIs, GET operations can be used on the resources. The caller of a REST service can pass different search parameters to retrieve his desired logs, such as process logs of a particular VM at a specific time.

4) Integrity Verification by the Court. A malicious CSP could alter various activity logs stored in the persistent storage. Moreover, a dishonest investigator can alter the evidence before presenting such information to the court. Since we cannot guarantee the integrity of the evidence presented to the court, the court authority needs to verify the data’s integrity. In this regard, the court will collect the cryptographic proofs available on the Internet. If a log record is valid, it should be presented in the cryptographic proofs. Similarly, proofs of any faked instances of logs will not exist in the published proofs. If a log record is removed, that should also be detected from the proofs.

Once a proof is published, none of the entities can modify or deny that proof. Therefore, when the proof is made publicly available, neither the CSP nor investigator can alter any log records or provide falsified evidence. Thus, using the periodically published cryptographic proofs, a verifier can determine the integrity of the evidence even when three entities (user, CSP, and investigator) collude with each other.

4.2 OCF-SUPPORTED STaaS

Crucial evidence in the StaaS model. Consumers are moving towards the storage-as-a-service (StaaS) cloud for their storage needs. According to Gartner, consumers will store more than one-third of their digital content in the cloud by 2016 (Gartner, 2012). A few examples of current StaaS clouds include Dropbox (“Dropbox”, n.d.), OneDrive (“OneDrive”, n.d.), and Amazon S3 (“Amazon S3”, n.d.).
A criminal can keep some secret files, for example, child pornography (bbc.com, 2013) or stolen intellectual property (Dist. Court SD Texas, 2014) in cloud storage to keep the personal computer clean. Information about data possession and provenance can be used as important evidence to investigate cases involving a StaaS cloud.

1. Data Possession. In order to prove that a suspect had some specific files in a given storage at a particular time, it is important to preserve information about data possession. A malicious user can delete a file from his/her cloud storage to remain free of incriminating evidence. Therefore, from the proof of data possession, we need to detect whether a user possessed a file at a particular time even after removing the file from the cloud storage.

2. Provenance provides the history of an object. Hence, from the provenance of a file stored in the cloud, investigators can analyze the creation, evolution, and use of any evidence. By preserving provenance records, CSPs can provide the support of chain of custody. Provenance of a file can also reveal whether a litigation hold (Araiza, 2011) was violated or not. A litigation hold is a legal notice to a defendant that triggers the preservation of ESI, which may require the termination of the routine operation of an information system to suspend the normal destruction of ESI (Araiza, 2011). From the provenance information, it will be possible to identify whether a file under a litigation hold is deleted or modified during the litigation hold period.

Architecture. Figure 5 illustrates the proposed OCF-supported StaaS cloud architecture. This architecture provides the following features to support the OCF model for the aforementioned crucial ESI for data possession and provenance.

- First, we introduce a forensics layer, which generates and preserves data possession and provenance records. All the communications between a user and a cloud storage occur through the forensics layer.
- To translate the data possession and provenance information to verifiable ESI, the cryptographic proof generator, and publisher module respectively creates proofs of data possession and provenance and publishes them to the Internet.
- Data possession and provenance information can be collected by a read-only API published by the ESI disclosure module.
- The court authority can verify the integrity of the data possession and provenance information by using publicly available proofs.

Below, we describe various components of the architecture.

1) Forensics Layer. In the OCF-supported StaaS cloud, all communications between users and cloud storage occur through the forensics layer. This module can be implemented in such a way that it supports any type of cloud storage, such as Amazon S3 or a private cloud built on top of open source cloud framework, such as OpenStack Swift (“Swift”, n.d.). To handle data possession and provenance information the forensics layer includes two modules: a provenance manager and a data possession manager. (Should there be a necessity of preserving any new type of ESI, a new module could in future be integrated with the forensics layer.)
• Provenance Manager: Whenever a user creates, updates, or deletes a file from his/her cloud storage, the provenance manager module creates appropriate provenance records and preserves them in the provenance database. A provenance record of a file can include the description of operation (create/edit/delete), time of operation, change of contents of the file, information about the user who initiates the action. Maintaining the chronological ordering of provenance records is also important. The provenance manager applies secure provenance management protocol, such as a secure provenance chaining (Hasan, 2009) to maintain the chronological order of the provenance records.

• Data Possession Manager: This module is responsible for creating data possession information when a user uploads a file to the cloud storage through the forensics layer. One way to preserve the data possession is to preserve a file even after a user deletes a file. However, this will increase storage cost significantly. An efficient way to store the data possession information is using accumulator data structures (Benaloh, 1994) (Bloom, 1970). A single accumulator can store the data possession information of thousands of files of a single user, which lowers the required amount of space for data possession significantly. Moreover, such a scheme can preserve data possession information without disclosing the original data.

2) Cryptographic Proof Generator and Publisher. This module is responsible for creating proof of provenance and data possession.

• Proof of provenance: The provenance chain can preserve the integrity of the provenance records. However, since the provenance records along with the chain are stored within the control of the CSP, a malicious CSP can tamper with the provenance chain. The proof of provenance will ensure that a malicious CSP cannot modify the chain after-the-fact. To create proofs of provenance, the proof generator module retrieves the head of the chain at the end of an epoch and sign with CSP’s private key. The publisher then publishes this proof to the Internet.

• Proof of Data Possession: A malicious CSP can tamper with the data possession by manipulating the accumulator stored in the DP database. To prevent such malicious activity, the proof generator module creates proof of data possession by signing the accumulator at the end of each epoch and the publisher module publishes this signed accumulator as proofs of data possession

3) ESI Disclosure. Data possession and provenance records can be made accessible by the users, investigators, and the court authority by secure read-only APIs provided by the ESI disclosure module. The ESI disclosure module needs to communicate to the data possession database and provenance database to collect the data possession and provenance records. The APIs exposed by the ESI disclosure module are RESTful APIs. Data possession or provenance records of a particular user for specific duration can be obtained through the RESTful APIs using query parameters.

4) Integrity Verification. Using the publicly available proofs, the court authority can verify whether the data possession or provenance records provided by an investigator or CSP are valid. When the court authority needs to verify whether certain incriminating documents pertain or belong to a suspect, it first collects the proof of data possession of the suspect. Then, using the membership checking method of the accumulator, the court authority can verify whether or not the documents in question exist in the proof.

5. DISCUSSION

In this section, we show how existing forensics procedures fail for the hypothetical scenario presented in Section 2.3 and how our proposed model can ensure reliable forensics under the same scenario.

5.1 UNRELIABLE FORENSICS PROCESS IN CURRENT CLOUDS

Using traditional digital forensics methods, we cannot execute a reliable forensic investigation in current clouds. This situation is illustrated in Figure 6. In the motivating case study presented in Section 2.3, Mallory removed the stolen codes after using or further transferring them. When Bob requested Mallory’s ESI from CloudCo, the firm will fail to provide such removed documents. In current clouds, there is no way for Bob to recover those deleted files. If Mallory used a VM running on clouds to store the documents, the situation will be more complicated for Bob.

Since the storage of a VM is volatile, there will be no guaranteed trail of such files in the cloud. In this situation, the evidence presented to the court will be incomplete and the forensics process will not be reliable. Moreover, the court authority trusts Bob’s and CloudCo’s honesty to validate such evidence. However, such honesty is not guaranteed since either or both CloudCo and Bob could collude with Mallory to remove evidence or to provide false evidence in order to exonerate her. There is no reliable way for the court to verify the evidence in the current situation.
First, let’s assume that CloudCo, the cloud service provider that Mallory used to store the stolen intellectual properties, deployed an OCF-supported cloud infrastructure. Then, the sequence of action presented in Figure 6 will be executed. Mallory uses the storage as a service provided by CloudCo. Mallory sent a stolen code file to CloudCo at time $t$. According to the OCF model, this action will create an ESI, $E_{storage \_Mallory}$. CloudCo stores this ESI and creates a proof of the ESI, $VE_{storage \_Mallory}$, and publishes it to the Internet, which will make the evidence verifiable. Even if Mallory stores the stolen codes in a VM and terminates the VM after using the codes, the continuous synchronization scheme will store the proof $VE_{storage \_Mallory}$, which cannot be altered by any of the entities.

Later, Bob collects files from CloudCo using the secure API, analyzes and presents the evidence to the court. The court authority collects proof of evidence from the Internet and verifies the integrity of the evidence provided by Bob. Mallory can remove the incriminating files and collude with CloudCo. However, neither Mallory nor CloudCo can alter the proofs that are already available on the Internet. Similarly, Bob cannot modify these proofs. Hence, even if Mallory were to collude with Bob, then the court authority could detect any alteration of the evidence using the proofs. Hence, the proposed cloud forensics model can support reliable forensics in a strong adversarial scenario.

5.2 RELIABLE DIGITAL FORENSICS BY OCF SUPPORTED CLOUD

FIGURE 6: RELIABLE CLOUD FORENSICS PROCESS ON AN OCF-SUPPORTED CLOUD
6. RELATED WORK

Cloud forensics is a relatively new area of study. Since cloud computing is based on extensive network access, and since network forensics handles forensic investigation in private and public networks, Ruan et al. defined cloud forensics as a subset of network forensics (Ruan, 2011). They also identified three dimensions of cloud forensics – technical, organizational, and legal. However, this definition is not complete since analyzing a hard drive stored in Amazon’s infrastructure is not an instance of network forensics.

Logs are in heterogeneous formats in clouds and hence, it is difficult to examine and analyze log evidence. Marty proposed guidelines to overcome this problem (Marty, 2011). The proposed guidelines instruct us to focus on three things: when to log, what to log, and how to log. At the minimum, he suggests logging the timestamps record, application, user, session ID, severity, reason, and categorization, so that we can get the answer of what, when, who, and why (the “4 Ws”). He also recommended syntax for logging, which was represented as a key-value pair and used three fields to establish a categorization schema – object, action, and status (Marty, 2011).

Dykstra et al. illustrated the difficulty of data acquisition by using a hypothetical case study of child pornography (Dykstra, 2011). To investigate this case, the forensics investigator needed bit-for-bit duplication of the data to prove the existence of contraband images and/or video. In a cloud, the investigator could not collect such data independently. Furthermore, the data could not be seized by confiscating the storage server in a cloud, as the same disk might contain data from many legitimate users.

Zafarullah et al. were able to monitor the activity of a Eucalyptus-based cloud and log all internal and external interaction of Eucalyptus components (Zafarullah, 2011). From logs, they were able to track a DDoS attack launched from their Eucalyptus cloud. To make the network, process, and access logs available to customers, Bark et al. proposed to expose read-only APIs by CSPs (Birk, 2011). By using these APIs, customers can gather valuable information and provide this to investigators. Some of the authors of this paper and others (Zawod et al.) proposed a Secure Logging-as-a-Service (SecLaas) securely to store VM activities, a procedure that ensures integrity and confidentiality of logs from a malicious CSP and investigators (Zawod, 2013). To detect temporal inconsistencies in a VM’s timeline, Thorpe et al. developed a log auditor by using the ‘happened before’ relation (Lampport, 1978) in the cloud environment (Thorpe, 2012).

Delport et al. focused on isolating an instance to mitigate the multi-tenancy issue (Delport, 2011). Isolation (White, 2010) is necessary because it helps to protect evidence from contamination. Virtual Machine Introspection (VMI) can also be helpful in a forensic investigation. In (Hay, 2008), 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.

Patrascu et al. proposed a cloud architecture to monitor the activities in a cloud environment (Patrascu, 2014). 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 (Dykstra, 2013b). Using FROST, cloud users/investigators can acquire an image of the virtual disks associated with any of a user’s virtual machines, and can validate the integrity of those images with cryptographic checksums. It is also possible to collect logs of all API requests made to the CSP and OpenStack firewall logs for VMs. These works mainly focused on making the logs more easily available. While these two efforts are big steps towards providing forensics support in the cloud, these works treated cloud service providers as both honest and reliable. It had not been shown in these works how to protect users’ privacy and integrity of logs in the event of either or both a malicious CSP and investigator(s). However, in an adversarial situation, CSPs as well as investigators, can be malicious, non-compliant, and/or could tamper with the logs. Hence, the trustworthiness of the data collected through those proposed architectures could prove questionable. Moreover, these solutions only focused on logs, which is relatively narrow and does not span all the information one might wish to collect in a forensic investigation.

7. CONCLUSION

Because of the widespread adoption of clouds, it is more likely that adversaries will use clouds for various malicious activities. Certain incidents of cloud-based malicious activities have already been reported. Therefore, it is becoming increasingly important to ensure that clouds support reliable digital forensics investigations. Making the cloud forensics-aware also has the broader impact of bringing regulatory compliance to the realm of clouds. In this paper, we first identified the limitations of digital forensics in current cloud infrastructures. By examining cloud architectures and various entities involved in a cloud, we defined the cloud forensics process flow and proposed the Open Cloud Forensics (OCF) model. Cloud architects can use such model to design clouds that support trustworthy cloud forensics investigations. We proposed OCF-supported cloud architectures for the IaaS and SaaS cloud models and showed how such architectures can support reliable digital forensics in a realistic scenario.
the future, we plan to implement the proposed OCF supported, forensics-aware cloud infrastructure.

8. ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation EARLY CAREER Award #CNS-1351038 and Grants Nos.~1547245 and 1229282. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Additional support was provided by the Department of Homeland Security Grant #FA8750-12-2-0254, and Sigma Xi Grants-in-Aid of Research #G201503151198201.

9. REFERENCES


Dist. Court, SD Texas. (2014). Quantlab technologies ltd. v. godlevsky, Civil Action No. 4: 09-cv-4039.


Authors

Dr. Shams Zawoad is a Sr. Information Security Analyst at the Advanced Threat Detection department of Visa Inc, where he is involved in designing innovative security solutions. He received his Ph.D. in Computer Science from the University of Alabama at Birmingham (UAB) in April 2016 and received his B.Sc in Computer Science and Engineering from Bangladesh University of Engineering and Technology (BUET) in January 2008. His research interest is in cloud forensics, trustworthy cloud computing, cyber defense, big data security, secure provenance, IoT forensics, and distributed systems.

Dr. Ragib Hasan is a tenure-track Assistant Professor at the Department of Computer and Information Sciences at the University of Alabama at Birmingham. Hasan is the founder of the SECuRE and Trustworthy Computing Lab (SECRETlab) at UAB. He is also a member of the UAB Center for Information Assurance and Joint Forensics Research. Prior to joining UAB, He received his Ph.D. and M.S. in Computer Science from the University of Illinois at Urbana Champaign in October, 2009, and December, 2005, respectively, and was an NSF/CRA Computing Innovation Fellow post-doc at the Department of Computer Science, Johns Hopkins University. Hasan is the founder of Shikhkhok.com, an award-winning online education platform serving impoverished students in South Asia. Hasan has received multiple awards in his career, including the 2014 NSF CAREER Award, 2013 Google RISE Award, 2011 Google Faculty Research Award, and 2009 NSF Computing Innovation Fellowship.

Dr. Anthony (Tony) Skjellum is the lead cyber scientist for Auburn University. He is the Director of the newly formed Charles D. McCrary Institute for Critical Infrastructure Protection and Cyber Systems (http://mccrary.auburn.edu). His research activity focuses on: cyber security and forensics, applications of high performance computing to intrusion detection, message passing systems, embedded high performance computing, scientific computing and parallel computing, and exascale storage systems. He received his PhD in 1990 from the California Institute of Technology, and has been a professor of computer science for over 23 years.