Moussa et al. Journal of Cloud Computing: Advances, Systems and Applications (2018) 7:1
DOI 10.1186/s13677-017-0102-3

Journal of Cloud Computing: Advances, Systems and Applications

RESEARCH Open Access

CFaaS: bilaterally agreed evidence collection

Ahmed Nour Moussa[1*], Norafida Ithnin[2] and Anazida Zainal[3]

Abstract

A common cloud forensic model proposed by researchers is 'Cloud-Forensic-as-a-Service' where consumers have to access it as a service to collect forensic data from cloud environments. The 'Cloud-Forensic-as-a-Service' model raises the question of how it collects digital evidence pertaining to an incident which occurred in the cloud. Currently, types of 'Cloud-Forensic-as-a-Service' systems in the literature show that the system is controlled and implemented by the cloud provider, where they unilaterally define the type of evidence that can be collected by the system. A serious limitation of this approach is that it does not offer the consumer sufficient means of performing reasonableness checks to verify that the provider is not accidentally or maliciously contaminating the evidence. To address the problem, the paper proposes a conceptual bilateral Cloud-Forensic-as-a-Service model where both consumers and providers can independently collect, verify the equity of the forensic analysis process and try to resolve potential disputes emerging from the independently collected results. The authors have developed a cloud forensic process model to lead common and significant aspects of a bilateral Cloud-Forensics-as-a-Service model. The paper explicitly discusses the concept of a bilateral Cloud-Forensic-as-a-Service model.

Keywords: Cloud forensics, Forensics as a service, Trust, Model

Introduction

The focus of this research is on cloud forensic services provided remotely to Cloud Service Consumers (CSCs) over the internet. According to the National Institute of Standards and Technology (NIST), cloud services can either be offered as an Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) or Software-as-a-Service (SaaS) model [1]. Leaving aside specific technical details, the authors consider there to be Cloud Service Providers (CSPs) that sell the three basic IaaS services including storage, compute power and network to remote CSCs [2]. The authors are interested in providing a Cloud-Forensics-as-a-Service (CFaaS) model that is integrated into cloud architectures for the purpose of forensic investigations involving cloud environments. Consequently, an implemented CFaaS interface is made available to the CSCs, via the Service Level Agreements (SLAs) signed with their CSPs, to assist them in their investigation of their adopted cloud services.

Central to this CFaaS model is the issue of accountability for the digital evidence: who performs the investigation and decides what kind of digital evidence is required for a specific cloud forensics case –– the provider, the consumer, a trusted third party, or some combination of them? Traditional digital forensics investigations (non-cloud), investigators had the ability to seize any suspected device. However, in contrast to traditional digital investigations, the infrastructure responsible for the CFaaS model is deployed at the premises of the CSPs. In other words, the CSPs have a higher degree of control over most of the critical evidence needed for investigations involving cloud environments.

In this light, CSCs and Law Enforcement Agents (LEAs) are heavily dependent on the CSPs to obtain the evidence required for a cloud forensics case, as they have limited control on the cloud systems and data residing in it. This dependency further leads to serious issues surrounding trust in the CSPs, like originality of the evidence and timely response to litigation holds [3–6]. The distinguishing feature of the CSP side evidence collection is that it seems to be unilateral. CSP side forensics evidence collection can be acceptable when the CSC has good reasons to trust the CSP of not accidentally or maliciously contaminating the critical forensic evidence. However, the authors contend that there may be cases

* Correspondence: nmahmed3@live.utm.my; ahmedinour45@gmail.com
[1]Faculty of Computing, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia
Full list of author information is available at the end of the article

where this assumption does not hold, or there is a dispute between the CSCs and CSPs, and where other models are needed. Consequently, unilateral CSCs side evidence collection can be implemented, but, a trusted third party acting on behalf of both CSCs and CSPs would be more practical [7]. However, in this paper the authors consider a hitherto unexplored alternative of bilateral forensics in the collection and analysis of cloud based evidence. A new model, where the CSCs and CSPs independently collect and analyze digital evidence, compare their outputs and agree on a mutually trusted output, is proposed. The problem of achieving mutual trust in cloud forensics investigations is not currently covered in the literature but is becoming important as CSCs organizations increasingly rely on cloud computing for their needs. In this regard, technical issues in bilateral cloud forensics is explored and a model that is abstract and general enough to be applied to the different types of IaaS resources is developed.

The rest of the paper is organized as follows: the next section discusses an overview of cloud forensics. Section 3 highlights existing digital and cloud forensic process models reviewed in this study. In section 4 the live cloud forensic process model intended to guide the development of the bilaterally trusted CFaaS model is discussed. Section 5 describes the state of the art of the related work. In section 6 the CFaaS model is presented, and finally in section 7 a case study scenario is formulated to discuss the feasibility of the model.

Cloud forensics

Cloud Forensics (CF) covers more than one area of knowledge, and can be referred to as the application of digital forensics in cloud computing environments [8]. Another definition introduced by the newly established NIST Cloud Forensic Working Group, states that *"Cloud Computing Forensic science is the application of scientific principles, technology practices and derived and proven methods to process past cloud computing events through identification, collection, preservation, examination and reporting of digital data for the purpose of facilitating the reconstruction of these events"* [1]. Consequently, cloud forensics is more complicated due to the default nature of its characteristics such as multi-tenancy, multi-jurisdiction, data duplication and high degree of virtualization. Similarly, the chain of dependency or the trade of services among CSPs have made it difficult to follow the continuity of digital evidence in cloud. Therefore, in the case of cloud, the traditional forensic process models that were applicable in non-cloud environments are no longer practical. As a result, those traditional methods can be modified so as to adapt to the cloud environments or else cloud specific steps may precede existing methods in order to utilize such methods in cloud environments [9–11]. In general, the digital forensic investigation process is a post incident activity as it is mostly initiated after an incident happens. It follows few pre-defined steps, and in a cloud environment, can be implemented in two areas, that is, CSCs side forensics and CSPs side forensics.

CSCs side forensic

Forensic data pertaining to security incidents happening in cloud environments are left behind both on the CSCs and CSPs sides. Investigating cloud security incidents generally starts at the CSCs side. Identifying sources of forensic data and collecting them on the CSCs side is therefore deemed as a vital part of the cloud investigation process [12]. In order for the forensic data to be used as evidence, in cloud forensics, data have to be collected as early as possible in their sterile state. That is, forensic data residing in the cloud can purposefully or inadvertently be erased by the stakeholders. Similarly, it can also be erased by the cloud resources due to system configuration. For instance, the web browser history and session logs can be configured to be overwritten or erased after a specific period or when the file size reaches the configured maximum limit.

In the meantime, the rapid increase of the consumer's side endpoints, especially devices for Bring Your Own Device (BYOD) scenarios makes identification and collection of forensic data even more challenging [13]. Hence, to reconstruct a timeline of events for a cloud security incident, identifying and collecting those BYOD endpoints in a timely manner and keeping evidence integrity intact is crucial.

CSPs side forensic

It is essential that forensic data are similarly collected from the CSPs side. There are many forensic data created and made available on the cloud resources, which form a critical part of the forensic data. The lack of investigators' accessibility to the physical infrastructure of the cloud and the unknown location of the cloud data centers make it much harder, if not impossible, to identify, isolate and collect forensic data in the cloud. Furthermore, in a highly decentralized and virtualized cloud environment, it is quite common that the forensic data may be scattered across multiple data centers situated in different geographic locations [3, 14]. Even if the location is known, seizing systems at the data centers may endanger the availability of the cloud, and may as a result affect other co-tenants. This issue has been widely raised in the literature and some researchers have suggested possible solutions [2–4, 10, 14, 15].

The loss of data ownership for CSCs organizations to CSPs is another major issue in CF. As a result, CSCs are entrusting the data stewardship to the CSPs. According to the European Network and Information Security Agency (ENISA)'s, 'loss of data stewardship to the CSPs' is one of

the top risks of cloud computing. This loss of stewardship thus poses another big bottleneck for collecting forensic data from the cloud. Nevertheless, this loss of control depends on the type cloud deployment and service model, as outlined in Table 1. For example, in IaaS, CSCs have more control and relatively unfettered access to the system logs and data. Conversely, in PaaS and SaaS, CSCs have either limited or no access to such data. Hence, as the CSCs lose control of their data, at the same time, they are losing the identification and collection of forensic data for any subsequent forensic needs [10, 14]. In other words, as the degree of control of the CSCs decreases, less forensic data is available for CSCs. Subsequently, this creates more dependency on the CSPs to get access to such forensic data. This is illustrated in Table 2.

Table 1 A comparison of typical cloud delivery model control levels

Cloud Delivery Model	Typical Level of Control Granted to CSCs	Typical Functionality Made Available to CSCs
SaaS	Usage and usage-related configuration	Access to front-end user-interface
PaaS	Limited administrative	Moderate level of administrative control over IT resources relevant to cloud consumer's usage of platform
IaaS	Full administrative	Full access to virtualized infrastructure-related IT resources and, possibly, to underlying physical IT resources

Table 2 Cloud consumers and cloud providers responsibilities in relation to the cloud delivery models

Cloud Delivery Model	Common CSCs Activities	Common CSPs Activities
SaaS	Use and configure cloud.	Implement, manage, and maintain cloud service. Monitor usage by CSCs.
PaaS	Develop, test, deploy, and manage cloud services and cloud-based solutions.	Pre-configure platform and provision underlying infrastructure, middleware, and other needed IT resources, as necessary. Monitor usage by CSCs.
IaaS	Set up and configure bare infrastructure and install, manage, and monitor any needed software.	Provision and manage the physical processing storage networking and hosting required. Monitors usage by CSCs.

Digital and cloud forensic process models

Not only the digital evidence is needed to succeed in any court of law, the process followed in conducting the investigation will also be needed to prevail in courts. In response, researchers and forensic practitioners have proposed several digital forensic process models. Different researchers have refined previously published process models and proposed new ones, resulting in a variety of digital forensic process models and terminologies. A number of digital forensic process models selected for the design of a new process model, intended to guide in the development of the CFaaS model, are presented in this section. In this paper, in order to capture and visualize the flow of the processes in the existing digital forensic process models, the ordering of the processes have been represented by a Sequential Logic as adopted from [16, 17].

On this sequential logic representation of the digital forensic process models, the outcome of the circuit is dependent on the input and current internal states. For the circuit to evaluate true, all conditions of previous states must be true. The circuit will fail if the current state is not positively completed. That is, the investigator should revisit previous states in the process for completeness. If any of the previous states fail to complete, the investigator will not be able to continue the investigation. For example, in a digital investigation process 'data extraction' from the suspected digital media must be completed before evidence can be discovered during 'examination and analyses' or in a more simple way data must be extracted before examination and analyses could start. Therefore, the adopted sequential notation is illustrated as:

$$Digital\ Forensics\ Process\ Model = \{start \Rightarrow next \Rightarrow then \ldots\ldots end\}$$

In certain models where sub-processes are indicated, the same sequential notations will be illustrated for each process and its corresponding sub-processes. Parallel processes are indicated by ‖, while iterations or a previous processes that should be repeated are indicated by ⇔. This is to show similarities and differences within the sequence of activities when conducting a digital forensic investigation. The selected digital and cloud forensic process models include:

The Abstract Digital Forensic Process Model (ADFPM) [18] was developed based on the Digital Forensic Research Workshop model [19]. It consists of nine processes including Identification, Preparation, Approach strategy, Preservation, Collection, Examination, Analysis, Presentation and Returning evidence. It adds three more processes and describes what each one of them is concerned with. This ADFPM is represented as:

$$\mathbf{ADFPM} = \{Identification \Rightarrow Preparation \Rightarrow Approach\ strategy \Rightarrow Preservation \Rightarrow Collection \Rightarrow Examination \Rightarrow Analysis \Rightarrow Presentation \Rightarrow Returning\ Evidence\}.$$

The Integrated Digital Investigation Process Model (IDIPM) [20] was introduced based on the crime scene theory for physical investigations. It allows technical

requirements for each phase to be developed and for the interactions between the physical and digital investigations to be identified. This framework consists of 17 phases organized into five groups: Readiness, Deployment, Physical crime scene investigation, Digital crime scene investigation and Review. Those groups and phases of IDIPM are listed as:

$$\begin{aligned}\textbf{IDIPM} = \{&Readiness \Rightarrow Deployment \Rightarrow Physical\ Crime\ Investigation \| \\ &Digital\ Crime\ Investigation \Rightarrow Review\}\end{aligned}$$

Where

$$Readiness = \{Operations\ Readiness \Rightarrow Infrastructure\ Readiness\}$$

$$\begin{aligned}Deployment = \{&Detection\ and\ Notification \Rightarrow Confirmation\ and \\ &Authorization\ phase\}\end{aligned}$$

$$\begin{aligned}Physical\ Crime\ Invest = \{&Physical\ Preparation \Rightarrow Physical\ Survey \Rightarrow \\ &Physical\ Documentation \Rightarrow Physical\ Search \Leftrightarrow \\ &Physical\ Reconstruction \Rightarrow Physical\ Presentation\}\end{aligned}$$

$$\begin{aligned}Digital\ Crime\ Invest = \{&Digital\ Preservation \Rightarrow Digital\ Survey \Rightarrow \\ &Documentation \Rightarrow Digital\ Search \Leftrightarrow Digital \\ &Reconstruction \Rightarrow Digital\ Presentation\}.\end{aligned}$$

The Enhanced Digital Investigation Process model (EDIPM) [21] separates the investigations at primary and secondary crime scenes while depicting the processes as iterative, instead of linear. It was introduced based on the IDIPM model and expands the Deployment process into Physical and Digital crime investigations while introducing the Primary crime scene process. However, the reconstruction is only made after all investigations have been taken place. This EDIPM is given as:

$$\textbf{EDIPM} = \{Readiness \Leftrightarrow Deployment \Leftrightarrow Trackback \Leftrightarrow Dynamite \Leftrightarrow Review\}$$

Where

$$Readiness = \{Operations\ Readiness \Rightarrow Infrastructure\ Readiness\}$$

$$\begin{aligned}Deployment = \{&Detection\ and\ Notification \Rightarrow Physical\ Investigation \Rightarrow \\ &Digital\ Investigation \Rightarrow Confirmation \Rightarrow Submission\}\end{aligned}$$

$$Trackback = \{Digital\ Investigation \Rightarrow Authorization\}$$

$$\begin{aligned}Dynamite = \{&Physical\ Investigation \Rightarrow Digital\ Investigation \Rightarrow \\ &Reconstruction \Rightarrow Communication\}.\end{aligned}$$

The Hierarchical, Objectives Based Process Model (HOBPM) [22], proposes a multi-layer, hierarchical model, which includes objectives-based processes and sub-processes that are applicable to various layers of abstraction, and to which additional layers of detail can easily be added, as needed. The model includes the processes of Preparation, Incident response, Data collection, Data analysis, Presentation of findings and Incident closure. This HOBPM is sequentially represented as:

$$\begin{aligned}\textbf{HOBPM} = \{&Preparation \Rightarrow Incident\ Response \Rightarrow Data\ Collection \Rightarrow \\ &Data\ Analysis \Rightarrow Presentation \Rightarrow Incident\ Closure\}\end{aligned}$$

Where

$$Data\ Analysis = \{Survey \Leftrightarrow Extraction \Leftrightarrow Examination \Leftrightarrow Survey\}.$$

The Digital Forensic Process Model (DFPM) [23] proposed by NIST. It consists of four processes including Collection, Examination, Analysis and Reporting. In this model, the investigation process transforms media into evidence for law enforcement or for organization's internal usage. First, the collected data is examined, extracted from media and transformed into a format that can be processed by forensic tools. Subsequently, the data is transformed into information through analysis and finally the information is transformed into evidence during the reporting process. The DFPM is given as:

$$\textbf{DFPM} = \{Collection \Rightarrow Examination \Rightarrow Analysis \Rightarrow Reporting\}$$

Where

$$\begin{aligned}Data\ collection = \{&Identifying\ Possible\ Sources\ of\ Data \Rightarrow Acquiring\ the \\ &Data \Rightarrow Incident\ Response\ Considerations\}.\end{aligned}$$

The Digital Forensic Investigation Process Model (DFIPM) proposed in [24] groups and merges the same activities that provide the same output into an appropriate process. The model simplifies the existing complex process models. It can be used as a generic model for investigating all incident cases without tampering with the evidence and protecting the chain of custody. The model consists of five processes including Preparation, Collection and Preservation, Examination and Analysis, Presentation and Reporting and Disseminating the case. This DFIPM is given as:

$$\begin{aligned}\textbf{DFIPM} = \{&Preparation \Rightarrow Collection\ and\ Preservation \Rightarrow Examination \\ &and\ Analysis \Rightarrow Presentation\ and\ Reporting \Rightarrow Dissemination\}.\end{aligned}$$

The Digital Forensic Evidence Examination Process Model (DFEEPM) [25] defined nine processes including Identification, Collection, Preservation, Transportation, Storage, Analysis (interpretation and attribution), Reconstruction, Presentation and Destruction. All of these should be done in a manner that meets the legal standards of the jurisdiction and the case. The DFEEPM is a linear model and is represented as:

$$\begin{aligned}\textbf{DFEEPM} = \{&\textit{Identification}\Rightarrow\textit{Collection}\Rightarrow\textit{Preservation}\Rightarrow\textit{Transportation}\\&\Rightarrow\textit{Storage}\Rightarrow\textit{Analysis}\Rightarrow\textit{Reconstruction}\Rightarrow\textit{Presentation}\Rightarrow\\&\textit{Destruction}\}\end{aligned}$$

Where

$$\textit{Analysis} = \{\textit{Interpretation}\Rightarrow\textit{Attribution}\}.$$

The Harmonized Digital Forensic Investigation Process Model (HDFIPM) [26] was introduced in 2012 and proposed several actions to be performed constantly in parallel with the processes of the model in order to achieve efficiency of investigation and ensure the admissibility of digital evidence. This model is categorized into five higher abstraction levels of digital investigation process classes including readiness class, initialization class, acquisition class and investigative class. The readiness class deals with pre-investigative processes aimed at digital forensic preparedness within an organization. Sixteen other processes are categorized among the remaining three classes in terms of scope, functions and order. These include incident detection, first response, planning and preparation referred to as initialization process class. Potential digital evidence identification, Potential digital evidence collection, Potential digital evidence acquisition, Potential digital evidence transportation and Potential digital evidence storage are grouped under the Acquisition class. Finally, the Investigative class contains Evidence acquisition, Evidence examination and analysis, Evidence interpretation, Reporting, Presentation and Investigation closure processes. The HDFIPM is considered as a standard model in some researches and is therefore represented as follows:

$$\begin{aligned}\textbf{HDFIPM} = \{\{&\textit{Readiness class}\Rightarrow\textit{Initialization class}\Rightarrow\textit{Acquisitive class}\Rightarrow\\&\textit{Investigative class}\}\|\{\textit{Obtaining authorization}\|\textit{Documentation}\|\\&\textit{Managing information flow}\|\textit{Preserving chain of custody}\|\\&\textit{Preserving digital evidence}\|\textit{Interaction with physical investigation}\}\}\end{aligned}$$

Where

$$\textit{Readiness class} = \{\textit{Planning}\Rightarrow\textit{Implementation}\Rightarrow\textit{Assessment}\}$$

$$\begin{aligned}\textit{Initialization class} = \{&\textit{Incident detection}\Rightarrow\textit{First response}\Rightarrow\textit{Planning}\\&\Rightarrow\textit{Preparation}\}\end{aligned}$$

$$\begin{aligned}\textit{Acquisitive class} = \{&\textit{Evidence identification}\Rightarrow\textit{Evidence Collection}\Rightarrow\\&\textit{Evidence acquisition}\Rightarrow\textit{Evidence transportation}\Rightarrow\\&\textit{Evidence Storage}\}\end{aligned}$$

$$\begin{aligned}\textit{Investigative class} = \{&\textit{Evidence acquisition}\Rightarrow\textit{Evidence examination}\\&\textit{and analysis}\Rightarrow\textit{Evidence interpretation}\Rightarrow\\&\textit{Reporting}\Rightarrow\textit{Presentation}\Rightarrow\textit{Investigation closure}\}.\end{aligned}$$

The Forensic Investigations Process Model in Cloud Environments (FIPMCE) [15] was introduced based on the DFPM model. Due to the evolution of cloud computing the processes were changed to apply basic forensic principles and processes. It consists of four processes including Determining the purpose of the forensics requirement, Identifying the types of cloud services (SaaS, IaaS, PaaS), Determining the type of background technology used, and Examining the various physical and logical locations, (client, server or developer sides). The FIPMCE is given as:

$$\begin{aligned}\textbf{FIPMCE} = \{&\textit{Determine purpose of the investigation}\Rightarrow\textit{Identify type of}\\&\textit{cloud service used}\Rightarrow\textit{Determine type of technology used}\\&\Rightarrow\textit{Examine various physical and logical location}\}\end{aligned}$$

Where

$$\begin{aligned}\textit{Examine various physical and logical locations} = \{&\textit{Consumer side}\|\\&\textit{Provider side}\|\textit{Developer side}\}.\end{aligned}$$

The Integrated Conceptual Digital Forensic Process Model for Cloud Computing (ICDFPMCC) [27] was proposed based on the frameworks of McKemmish and National Institute of Standards and Technology. Its focus is on the differences in the preservation of forensic data and the collection of cloud computing data for forensic purposes. It consists of four stages, which are, Identification and Preservation, Collection, Examination and Analysis, and finally Reporting and Presentation. The utility of the ICDFPMCC is validated in a number of researches and is represented as follows:

$$\begin{aligned}\textbf{ICDFPMCC} = \{&\textit{Identification and Preservation}\Leftrightarrow\{\textit{Collection}\Rightarrow\\&\textit{Examination and Analysis}\}\Rightarrow\textit{Reporting and Presentation}\}.\end{aligned}$$

The Cloud Storage Forensics Process Model (CSFPM) [28] was introduced based on the intelligence analysis cycle and DFPM model. It includes processes such as Commence (Scope), Preparation and Response, Identification and Collection, Preservation, Analysis, Presentation, Feedback and Complete. The CSFPM is given as:

$$\begin{aligned}\textbf{CSFPM} = \{&\textit{Commence}\Rightarrow\textit{Prepare and Respond}\Leftrightarrow\textit{Identify and collect}\Rightarrow\\&\textit{Preserve}\Rightarrow\textit{Analyze}\Rightarrow\textit{Present}\Rightarrow\textit{Feedback}\Rightarrow\textit{Complete}\}.\end{aligned}$$

The Cloud Network Forensics Process Model (CNFPM) [9] has five horizontal processes that interact with a management process, which is needed as a central point of control. The processes of the model are adopted from the DFPM model introduced by NIST [23]. The processes represent independent tasks with regards to the investigation of an incident. This generic process model is developed to remotely analyze network traffic in the IaaS environment. All processes of this model take place within the cloud, and the consumers are provided with an interface to request forensics data. The processes of this model include Data Collection, Separation, Aggregation, Analysis and Reporting controlled by a Management Console. Authors of this CNFPM has implemented its prototype in

an OpenNebula-based IaaS environment and is represented as follows:

$$\mathbf{CNFPM} = \{\{Data\ Collection \Rightarrow Separation \Rightarrow Aggregation \Rightarrow Analysis \Rightarrow Reporting\} \Leftrightarrow Management\}.$$

The Open Cloud Forensic Process Model (OCFPM) most recently proposed in [29] is supposed to continuously be supported by the CSPs. The model is built based on the DFPM model and defines six processes such as Preservation, Identification (Incident and Evidence), Collection, Organization (Examination and Analysis), Preservation and Verification. Both Identification and Organization processes have sub-processes. This OCFPM is given as:

$$\mathbf{OCFPM} = \{Preservation \Rightarrow Identification \Rightarrow Collection \Rightarrow Organization \Rightarrow Presentation \Rightarrow Verification\}$$

Where

$$Identification = \{Incident \Rightarrow Evidence\}$$
$$Organization = \{Examination \Rightarrow Analysis\}.$$

Based on the above reviewed digital and cloud forensic process models, the authors propose the live forensic process model intended to lead the CFaaS model. The process model is briefly discussed in the following section.

Proposed live cloud forensic process model

In order to identify a suitable cloud forensic process, existing digital and cloud forensic process models have been reviewed in Section 3. In this light, it is obvious that some existing models follow similar methods while others move into different methods of investigation, but the outcomes in most occasions are almost the same. Based on the suggestions and drawbacks highlighted in the reviewed models, authors have designed a new 'Live Cloud Forensic Process Model' shown in Fig. 1. The model consists of the following processes:

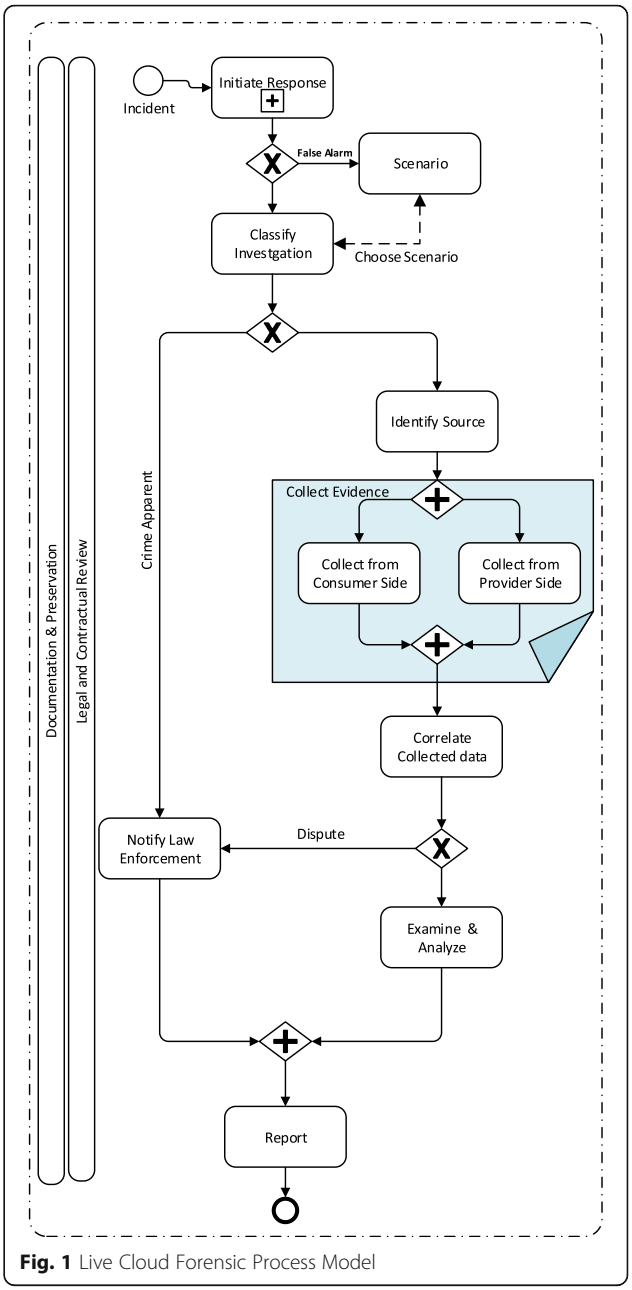

Fig. 1 Live Cloud Forensic Process Model

Initiate response

CSCs organizations will either need dedicated first responders, or a staff awareness system backed up by a helpline, so that when an incident occurs, a local immediate assessment can be done to its resource. At this stage, the process of confirming whether the incident is real or a 'false alarm' should begin, even though this can be a conclusion at any later process in the investigation.

Scenario

The scenario consists of a predefined proper risk assessment conducted to the local infrastructure of the CSCs, the history of breaches that have occurred on the infrastructure of the CSPs, and legal policies and bylaws that the cloud CSPs abides by, so that the CSCs can achieve an optimal scenario definition. Based on the scenario, a decision is made about which specific approach can be used for the case being investigated. This also equips the investigator with the best practices and procedures which are usually adopted by cloud forensics practitioners. To specify the case, the scenario may define the evidence that is most likely related to that precise digital crime. The scenario also specifies locations to look for and tools to be run for specific digital evidence.

Classify investigation

In classifying the investigation, the incident response team should confer with local staff (and with legal advisers if necessary), in order to correctly classify the investigation type, as early as possible. This will help to select the appropriate plan from the scenario at hand. It is especially important to identify the overall incident, as this tends to help in the decision making process.

Notify law enforcement

Meanwhile, the notification of law enforcement involves referring a case to the appropriate LEAs or other authorities as soon as that course of action is considered appropriate. That decision should usually be made by the management of the organization after consulting with their legal advisers if necessary.

Identify sources

Identify Sources refers to the first process of cloud investigation, and deals with identifying all possible sources of evidence in a cloud environment in order to prove that the incident took place. This process is crucial, because the next processes depend upon the evidence sources identified here.

Collect evidence

In this process forensics local investigators of the CSCs are supposed to start collecting digital evidence after their sources have been identified and isolated. Based on the delivery type and technology of the utilized cloud service model, the data collection can be implemented in the following two areas:

Consumer side evidence collection

The two primary areas for evidence collection on the cloud consumer side are devices and networks used by the users to access cloud services. In other words, forensic data that can be found at the consumer premises may either be data stored in a device or data flowing through local network of the consuming organization. Usually, the kind of forensic data that can be extracted from a device residing on the premises of the CSCs may include history logs, temp data, the registry, access logs, chat logs, session data and persistent cookies that can be found on the web browsers through which the CSCs are accessing their adopted cloud services [15]. Similarly, network forensic data may also include filesystems, processes and network traffic. Therefore, once potential sources of digital evidence in the CSCs is identified, live digital evidence is immediately collected in a forensically sound manner.

Provider side evidence collection

Likewise, many forensic data that form a critical part of forensic data are created at the CSPs side. These forensic data, which are at the custody of the CSPs may include system logs, application logs, user authentication and access information, database logs etc. CSPs side forensic data may also include virtual disk data and physical memory data pertaining to particular virtual machine, host logs and Application Programming Interface (API) logs. Therefore, it is essential that this evidence is collected remotely by an investigator situated at the premises of the provider.

Correlate data

The next process is Correlation, where due to the distributed nature of cloud computing, perfect time synchronization between all cloud environments is almost impossible. As a result, the time of an incident may differ from device to device. Thus, timestamps from different sources of forensic data in the cloud environment can be misleading or deceptive. Hence, this process correlates and visualizes the logical relationship among forensically interesting digital objects collected from CSCs and CSPs sides. The overall vision behind this process is to create a uniform timeline for the collection of forensic data in response to incidents pertaining to cloud.

Examine and analyze

In the meantime, Examine – Analyze, involves inspection and extraction of crucial digital evidence from the huge amount of forensic data collected in previous processes. In this process, extracted crucial digital evidence is analyzed with different tools to reveal any useful information that may prove if someone is guilty or not. Preliminary event reconstruction may also take place in this process.

Report

Reporting is the last process which deals with the presentation of the evidence in a court of law. In this regard, a well-documented report that contains the findings and an expert testimony should be produced on the analysis of the evidence. Since cloud computing is a very complicated environment to understand, especially for ordinary Internet users, the findings should hence be presented in a way that the jury would understand.

Legal and sontractual review

Meanwhile, the process of Legal and Contractual Review runs concurrently with all other processes, here, CSCs should develop a set of SLA requirements that may improve the qualities of forensics readiness, and most importantly may render the ability of consumers to rely on

CSPs to collect and store valid and admissible evidence. These requirements should enable contractual obligations that are demanded by CSCs from their CSPs. If an SLA does not state the type of process or forensic data that will be provided for its CSCs, then, the CSPs have no contractual duties to offer such information. Consequently, lack of a comprehensive SLA may lower the quality of the best evidence available and cause lack of access to forensics data. Thus, the SLA contract should govern what type of forensic evidence should be collected and the process of its storage.

Documentation and preservation

The Documentation and Preservation process runs concurrently with all other processes. It includes listing the organization's digital evidence by category, location, and the custodian or stewardship. Similarly, it includes taking notes about details on digital evidence storage, accessibility, associated retention policies and procedures created to preserve the chain of custody. Creating this list is increasingly important as the volume of evidence within the CSCs grow. Therefore, organizations should consider developing the appropriate evidence preservation plan in advance, with flexibility built in for exceptional circumstances. These may include preserving potential evidence locally or preserving them in another cloud on a dedicated litigation hold server.

Related work

Cloud forensics solutions have been proposed by many researchers over the past five years. Some researchers simply present concepts while others provide details on how a solution can be implemented in practice in cloud environments. This section discusses some research works that are aimed at investigating cloud environments. The focus of current studies on cloud forensics can be classified into five categories, including cloud forensic readiness, cloud forensics process models, evidence collection and acquisition, evidence examination and analysis and cloud-based technical and conceptual solutions.

Cloud forensics readiness

Consumer and provider side cloud forensic readiness is a solution for most of the issues and challenges pertaining to cloud forensic. Kirsten and Barbara [30] proposed a theoretical approach to proactively collect evidence from cloud environments claiming that cloud forensics readiness calls upon technological and organizational strategies to address the risks that threaten organizational information. The researchers offered a conceptual framework for making decisions about how to identify and manage the increasing quantity of evidence collected in clouds. They integrated best practices from the Record Management discipline and an Organizational Network Forensics Readiness model proposed by [31] to achieve the proposed cloud forensics readiness. A cyclic model proposed in their research consists of four phases, including Continuity of evidence, Patterns of evidence, Weight of evidence, and Manifestation of evidence. The model is not validated by practically applying it to the cloud, instead the researchers applied the model to a set of cloud forensic issues raised in a research conducted in [13].

Sibiya et al. proposed an integrated cloud forensics readiness model that uses a security as a service model, a cloud model, and a digital forensics as a service model. In this model the consumer should always access cloud resources through security as a service model and invoke digital forensics as a service model to achieve cloud forensics readiness. The authors implemented their model in a private cloud scenario (Nimbula directory) to demonstrate its potential utility [32].

Philip M Trenwith and H.S. Venter proposed a cloud forensics readiness model that uses remote and centralized logging in an attempt to improve the integrity of stored evidence, and to overcome jurisdiction issues that a cloud forensics investigator may face. The researchers implemented the proposed model in a proof of concept prototype on a client and server applications that consist of a windows service [33].

Lucia De Marco et al. presented a cloud forensics readiness system to provide a manner of implementing forensics readiness capability in cloud environments. The system includes several modules that perform dedicated operations interconnected via dedicated open virtualization format communication channels. The researchers implemented the model in OpenStack project. In another study, the researchers considered formalization of a co-signed SLA for cloud forensics readiness. The SLA is composed of a set of clauses that are fed as an input to their cloud forensics readiness system [34, 35].

Makutsoane et al. proposed a Cloud Capability Decision Framework (C^2DF) that can be used by CSCs to achieve a certain level of confidence in the CSPs they select, with regard to their forensics readiness. The framework employs risk analysis tools and techniques as resource for cloud consuming organization, when migrating to the cloud, to assess the cloud forensics readiness solution of CSPs. The framework follows four sequential phases, including evaluation of the CSCs, evaluation of the CSPs, alignment of the CSPs with a digital forensics readiness model, and lastly selection of the CSPs. The researchers did not pragmatically test the framework within an industry environment [36].

Evidence collection and acquisition

Evidence collection and acquisition deals with identification of potential sources of digital evidence and the means of acquiring it from cloud computing resources. Several

researchers have participated in efforts to define tools and techniques for the use of digital forensics in the cloud environments. Dykstra and Sherman discussed a method for collecting digital evidence from Amazon's Elastic Computing Cloud (EC2) service in the context of conducting cloud forensics using both conventional digital forensics tools and Amazon's export features. They also used Eucalyptus to inject forensics tools into running virtual machines via the hypervisor layer using virtual machine introspection techniques. In addition, by using tools such as Encase and AccessData's Forensics Tool Kit (FTK), they successfully collected evidence from both Amazon and Eucalyptus. They also explained levels of trust required in the cloud service models to execute evidence collection procedures [5].

Corrado Federici proposed a cloud data imager and used it for collecting digital evidence from Dropbox, Google Drive and Microsoft SkyDrive. With the cloud data imager he could display the file hierarchy, and was able to logically collect data from those three cloud storage products. The tool achieved this, by providing read-only access to the cloud service primarily and, then, by using Secure Hypertext Transfer Protocol (HTTPs) requests and Open Authentication (OAuth) tokens, each specific to the individual cloud storage product being examined [37].

Kurt Oestreicher proposed an effective means for forensic acquisition of data stored on a cloud service. He collected and examined forensic data from Apple's iCloud when used on an OS X personal computer. He created two identical virtual machines representing the subject computer and examination computer. A new iCloud account was created, new iCloud data was created on the subject machine, and the data were synchronized with the cloud service. The second virtual machine acting as the examiner was synchronized with the newly created iCloud account. Representing an examiner performing live forensic acquisition, files created on the subject system were downloaded and analyzed. The downloaded files were found complete and forensically sound because Message Digest 5 (MD5) hash values and timestamps matched [38].

In an attempt to provide digital forensic practitioners with an overview of the capability of mobile forensics tools in acquiring forensic data from cloud-of-things devices, Cahyani et al. have undertaken a research on the extent that three popular mobile forensics tools may acquire forensics data from three Windows phone devices. They examined the Nokia Lumia 900, the Nokia Lumia 625 and the Nokia Lumia 735 against the Paraben Device Seizure v7.0, XRY forensic Pack v6.13.0 and the Cellebrite UFED Touch v4.1.2.8. In this study, researchers also examined the effects of the settings modification and alternative acquisition processes on the acquisition results. As a result, the researchers have determined that the power status of the mobile devices matter because a data alteration issue may occur if the device power state was not correctly handled at the beginning of the acquisition process. The result also revealed that support for Windows Phone devices by the tested mobile forensic tools was still limited [39].

Evidence examination and analysis

Evidence examination and analysis deals with the tools and techniques used to discover relevant evidence from forensic data and reconstructing a sequence of events to answer questions pertinent to cloud forensics cases. The cloud forensics community has contributed to a number of tools and techniques to cope with the issues related to analyzing digital evidence collected from cloud environments. Anwar et al. generated their own dataset by attacking open source Eucalyptus with known cloud attacks and further analyzed built-in logs and third-party-application logs. They simulated an HTTP Denial of Service (DoS) attack in a virtualized environment on the Eucalyptus cloud controller. They proposed and discussed potential snort rules for detecting an HTTP DoS attack on the Eucalyptus. They also highlighted relevant log entries on the cloud controller as a result of the attack [40].

Fabio et al. formulated a case study to analyze forensics evidence stored in a Window 7 personal computer when Google documents, PicasaWeb, Flickr, and Dropbox are accessed via web browser. The researchers also analyzed when the Dropbox client is installed in a local synchronized folder. They performed the test twice, the former with live forensics tools on a powered on laptop running Windows 7 and the latter with postmortem forensic tools on a physical image of its hard drive [41].

Jason Hale analyzed forensic data created by an Amazon Cloud Drive accessed via web browsers and its desktop application. Potential digital evidence was found in the web history, windows registry and log files on the test installations and uninstallations. As a result, he proposed methods to follow when determining what files were transferred to and from an Amazon Cloud Drive [42].

Quick and Choo conducted three studies to identify evidence that is likely to remain after the use of major cloud storage products, such as SkyDrive, Google Drive and Dropbox on Windows and on iOS devices using common forensic tools. They located a range of evidence across the devices, including SQLite databases, log files, registry entries, thumb cache data, link files and browser history. They also conducted live forensics to determine whether artifacts of interest could be located via memory analysis and network interception [43–45].

Shariati et al. analyzed the possible forensic data left behind on the consumer devices when the Ubuntu One cloud storage service is utilized. This study gave a special focus on the analysis of both volatile and nonvolatile data present after utilizing Ubuntu One on different platforms, such as Windows 8.1, Mac OS X 10.9, and

iOS 7.04. The researchers showed that they were able to locate a range of distinct data, but they found that access to valuable forensic data, such as authentication and user action logs, varied between platforms [46].

Cloud forensics process models

Cloud forensics processes deal with establishing theoretical forensic processes and procedures that must be put in place to guarantee the integrity of evidence throughout an investigation. The cloud forensic process models may also define fundamental forensics principles for the development of new tools and techniques. Chae Cho et al. proposed a high-level guidance on the forensic analysis of the well-known Hadoop Distributed File System (HDFS). Processes suggested in this model include identification, live collection and analysis, and static collection and analysis [47].

Hong and Ting proposed a process model based on a traditional digital forensics model [23]. The process model proposed in this study consists of four processes, including determine the purpose of the forensic requirements, identify the types of cloud services, determine the type of background technology and finally examine physical and logical locations of digital evidence. The researchers divided process number four into three location of focus, such as client side, provider side and developer side [15].

Hyunji et al. proposed relatively detailed procedures for investigating cloud storage services to determine the forensic data that can be collected from four popular public cloud services, such as Amazon's Simple Storage Service (S3), Dropbox, Evernote and Google Docs. Each cloud service has been analyzed with four major operating systems, including Windows, Mac OS, iOS, and Android. The researchers prioritized four sources of digital evidence for cloud forensics investigations, including log file of web browser, data of client application in personal computer, data in smartphones, and physical memory of the system in question [48].

Theodoros and Vasilios proposed an acquisition process and scenarios to meet challenges pertaining to the acquisition of evidence from cloud environments. Different from other proposed process models the researchers claim the deployment of evidence correlation right from the beginning of a cloud storage forensics is a plausible scenario that will prove or disprove the allegation. The researchers also discussed that cloud forensics readiness is vital and mandatory to significantly decrease the cost and time of digital evidence acquisition. However, they highlight that cloud forensics readiness could only happen from the CSPs side, which comes as a result of a lack of awareness of the cloud consumers to interfere. The acquisition process proposed in this research starts by identifying the evidence source followed by evidence collection. After identifying and collecting the valuable digital evidence, investigators can deploy traditional digital forensics processes in order to recover, examine, and record conclusions drawn from the examination. Finally, digital evidence recovered from CSCs and CSPs sides can be correlated in order to draw conclusions about the suspect's activities to prove the case [49].

Martin et al. also proposed an integrated cloud forensics process model with which they conducted a server and client analysis of an ownCloud private storage as a service product. With this four-stage cloud forensics model they collected a range of client software data, including file data and metadata and authentication credentials of the cloud consumer. Using the credentials collected from the client, they analyzed server components and were able to decrypt files stored on the server. In a recent study, for further validation, they analyzed the XtreemFS distributed file system to determine the proper methods for forensic examination of file systems that commonly underlie cloud systems. The researchers have shown that there are significant complications introduced by the use of distributed file systems, when collecting evidence from cloud systems. Finally, they suggested that practitioners should analyze the cloud system directory to locate relevant metadata and object storage components for collection of particular volumes or user files [27, 50, 51]. Guided by this process model Thethi and Anthony evaluated AccessData's FTK as a forensics tool for collecting evidence from Amazon's EC2 cloud. They conducted an experiment using a case study that could be analogous to a real-world case involving cloud evidence [52]. Finally, Daryabar et al. in their recent work guided by this process model analyzed a range of forensic artifacts arising from user activities which could be forensically recovered from smartphones when the MEGA cloud client app is utilized. Their main focus remained on identifying the potential forensic evidence that can be recovered from the Android and iOS platforms. In this light, they also studied the modifications that may happen to the file content or metadata during uploading and downloading process, which may affect preservation of evidence, on Android and iOS platforms [53].

Quick et al. proposed a seven phased process model, upon which they conducted analysis to client-side artifacts for three cloud storage products to determine whether files were inadvertently modified during collection from these cloud storage services. They realized that file content remained unchanged; however, timestamp data changed based on download sources and timestamp type [28].

Gebhardt and Reiser proposed a cloud network forensics process model. The researchers implemented the model using OpenNabula platform and the Xplico analysis for validation. The model provides remote network forensics mechanism to CSCs, and ensures separation of consumers in a multi-tenant environment [9].

Martini and Choo proposed another six-step process, upon which they conducted an in-depth identification

process to establish the forensic data and metadata that could be collected by utilizing vCloud's RESTful API. The researchers concluded that the majority of the forensic data that they identified could be collected via API in a programmable way [54].

D. Povar and G. Geethakumari proposed a heuristic cloud forensics process model, based on the models proposed in [55] and NIST [23], aimed at enabling forensic investigators to perform investigations and tool developers to come up with forensic tools for cloud environments. This model suggests areas of data collection as CSCs side data collection and CSPs side data collection. Likewise, they propose that forensic analysis can either be done on the CSCs side or at CSPs side [56].

Ab Rahman et al. proposed a cloud incident handling model by integrating principles and practices from incident handling and digital forensics. The researchers validated the model using ownCloud as a case study. In an extra effort in a recent work, to respond to the increased volume of forensic data and the sophisticated attacks targeting cloud services, the model was later enhanced to a cloud incident handling and forensics-by-design model. The efficacy of the model was then demonstrated by using Google Drive, Dropbox, and OneDrive [57–60].

Zawoad et al. defined 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 to determine the facts about an incident involving clouds'. The researchers discussed the reliability of digital evidence, considering that the CSCs, CSPs and the forensics investigators involved in a case can all be malicious entities. Therefore, they defined a continuous model that verifies the reliability of forensic data extracted from cloud environments [29].

Simou et al. proposed a cloud forensics process model similar to the digital investigation process model proposed by Digital Forensic Research Workshop (DFRWS) [19]. Based on the suggestions and drawbacks located from the investigation of similar approaches presented before it. The researchers propose the inclusion of a collection process into the preservation process, secondly, they grouped the analysis process and examination process and finally the decision process is excluded [61].

Cloud-based technical and conceptual solutions

These technical and conceptual solutions are proposed by academia and industry researchers to equip cloud forensics practitioners and cloud providers with systems. Most of the systems are mostly designed for postmortem investigations. Similarly, the systems are mostly designed in a way that they can be integrated into the cloud environments.

Cheng Yan proposed a conceptual framework that is similar to anti-virus software [62]. Considering the cloud as the core of the forensics system, the researchers installed an analysis engine in the cloud to communicate with every server in the cloud to collect both volatile and non-volatile data. The engine has a real-time monitor with data acquisition and analysis functions to collect and analyze cybercrime activities in the cloud. When unusual behavior is detected, evidence will be collected on the relevant cloud server or client.

In addition, in order to protect the digital evidence, from contamination and loss of continuity, Delport et al. proposed methods to isolate suspected cloud instances in cloud environments. The researchers discussed the benefits and challenges of several means of isolating crime scene (instance, under investigation) in the cloud, including relocating the instance manually or automatically, server farming, collecting evidence from a failover instance, address relocation, sandboxing instances, using man-in-the-middle analysis on cloud virtual machines, and following a "let's hope for the best" approach, which involves imaging the relevant cloud nodes in a similar manner to traditional forensics [63].

Meanwhile, Raffael Marty discussed the logging challenges associated with cloud computing and proposed a guideline to address those challenges. In addition to discussing the logging challenges a logging architecture, and a set of guidelines which are applicable to multiple types of cloud implementations were proposed. The guideline is mainly on when, what and how to log. Marty implemented the conceptual logging architecture in an SaaS model, providing detailed configuration requirements to ensure that the log data in target are properly logged for forensics [64].

Dykstra and Sherman proposed a Forensics acquisition suite for OpenStack Tools (FROST), a toolkit for OpenStack IaaS cloud model. FROST allows remote CSCs to collect an image of their virtual machines. Similarly, it allows to the CSCs to retrieve log events for API requests and firewall logs for virtual machines. With the use of FROST, the CSCs can authenticate all of the evidence collected by using cryptographic hashes. Consequently, FROST has been integrated with the various OpenStack dashboard and compute components. After validation, the researchers suggested that the toolkit performance overhead is acceptable. Other researchers of this domain referred FROST as one of the most advanced IaaS data collection toolkit published to date [65].

Zawoad et al. proposed a Secure-Logging-as-a-Service (SecLaaS) system designed for collecting forensic logs from the cloud [66]. The researchers implemented the logging systems with OpenStack and Snort. They suggested that the logging system requires minimal overhead.

In a new study, Zawoad et al. also defined a model of trustworthy litigation hold management for cloud storage systems. Based on the model, a trustworthy litigation hold enabled cloud storage system (LINCS) has been proposed. The system verifies any deliberate destruction of evidence after a litigation hold is triggered. According to [7], LINCS can be implemented with low system overhead.

Patrascu and Patriciu proposed a method to monitor the activities in cloud environments by running a secure cloud forensics framework. The main goal of this framework is to gather forensic log data from all virtual machines running inside the virtualization layer. It provides an interface between the forensic investigator and the monitored virtual machines. This cloud logging system consists of five layers, including Management, Virtualization, Storage, Processing and Data layers [67].

Finally, Alqahtany et al. proposed a framework claimed to enable CSCs to collect and analyze forensics evidence without the consent of the CSPs. The model uses an agent-based approach that is held in each virtual machine. The agents send the required evidence to a central cloud forensic acquisition and analysis system in the CSPs environment. The communication between the two components is provided through a communication engine. The system uses cryptography to ensure the confidentiality and integrity of the forensic data. The researchers believe that the framework might cause additional performance overhead. However, this framework has not been tested in a cloud environments [11].

Proposed cloud forensics as a service

In this study, the authors conceived a CFaaS system composed of three basic components, Forensics Data Manager (FDM), Forensic Application/Analysis Manager (FAM) and Forensics Workflow Manager (FWM) as shown in Fig. 2. In this regard, the CFaaS model is exposed as a service through one or more service interfaces. With the use of this system, consumers, providers and LEA investigators can process, analyze and archive forensics data with improved efficiencies and increased productivity.

The FDM retrieves, uploads, and stores forensic data for the use by the FAM. The FDM should be able to extract the relevant data and logs with the help of live, as well as static data acquisition techniques. Inherently, it collects forensic data from all sources in the cloud infrastructure.

Consequently, the FAM receives and analyzes forensics data through collaboration processes in a manner that can be used by the FWM to generate a forensic report. Here, the application manager uses dynamic data mining techniques to segregate the relevant evidence for proving crime and delivers them to the workflow manager. In addition, the analysis software utilized in this stage are created by forensic software vendors and are legally supported.

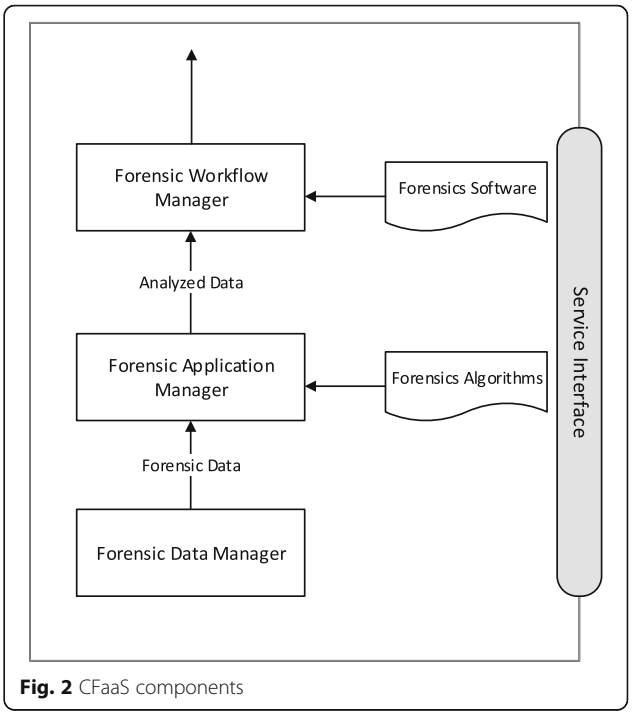

Fig. 2 CFaaS components

Furthermore, it is assumed that FAM as well as FWM components use algorithms and software known to the involved parties. Thus, when given the same forensic data, any party can compute the output of the relevant forensics analysis.

Trust assumptions and root of trust

Regarding the trustworthiness of the forensics analysis produced by the components of the CFaaS system, the difference between unilaterally trusted and mutually trusted outputs has to be distinguished. A unilaterally trusted output is produced by a party with the help of its own component services and is not necessarily trusted by other parties. The components could be located either within or outside the party's environment. In the latter case, the component's owner may need to take additional measures, such as the use of tamper-resistant mechanisms, to protect the component and its outputs against modification by other parties [68, 69]. There are two approaches to producing a mutually trusted output: (i) A Trusted Third Party (TTP) produces the output using its own certified infrastructure, or (ii) the parties concerned use their respective unilaterally trusted outputs as the basis for agreement on a valid, mutually trusted output [70]. The latter approach is the main focus of interest in this study. In this approach, the CSCs and CSPs need to execute some protocols between them to produce an agreed upon and non-repudiation output. Inherently, the mutually trusted outputs produced in this approach form the "root of trust" for developing trusted CFaaS system. The source of mutually trusted outputs can

be rooted in any one of the three components constituting the CFaaS system shown in Fig. 2. Therefore, once a mutually trusted output source is available, the trustworthiness of the component services above it becomes irrelevant. Given the determinacy assumption, a party can always resort to the mutually trusted output to compute and verify the results produced by other parties. For example, given an FDM that produces mutually trusted forensic data, the FAM and FWM services can be provided by any of the parties in any combination; their outputs are verifiable by any other party.

Bilateral cloud-forensics-as-a-service system

The bilateral cloud forensics system model is an attractive solution in applications where mutually untrusted CSCs and CSPs are reluctant, or unable, to use a trusted third party and therefore, agree to deploy their own component services. A distinguishing feature of this CFaaS system is that the CSCs and CSPs own and run their own independent but functionally equivalent component services to produce unilaterally trusted outputs. In this regard, a bilateral agreement between the pair of component services results in the trusted output needed to build the CFaaS system. This approach leads to two fundamental issues: (i) how do the CSCs and CSPs collect the forensic data that are essential to compute unilaterally trusted outputs that can form the basis for agreement on the forensic investigation, and (ii) how do the CSCs and CSPs resolve conflicts over a forensic investigation. Answers to these are explicitly discussed in Sections 6.3 and 6.4 respectively.

Forensic data collection

In the CFaaS system illustrated in Fig. 2, the FDM represents the instrumentation that performs the collection of forensic data. It is assumed that a Forensic Data Collector (FDC) is the component of the FDM that is responsible for doing digital evidence acquisition. Hence, the FDC is a piece of software that is possibly in combination with some hardware components which are used to collect and store the forensic data that are crucial for an incident under investigation. In this regard, the FDM has different FDCs that hold responsibility for various forms of the digital evidence acquisition.

As stated in Section 4.6, it is natural that forensic data pertaining to a cloud computing environment should be collected from the CSCs and CPSs sides [2, 27, 28, 49, 56]. To collect its unilaterally trusted forensic data with respect to a cloud incident: (i) the CSCs can collect and analyze forensic data left behind on their client machine by invoking commands at the CFaaS service interface, and (ii) the CSCs must collect and analyze forensic data on the provider side by invoking commands at the CFaaS service interface.

In contrast, as can be seen in Fig. 3, the CSPs have access to the CFaaS system interface, which they expose to the CSCs, as well as to the resource interface, which they use for operations on their cloud resources. Consequently, the CSPs can retrieve the forensic data from its resources by either using the CFaaS system interface or by directly collecting the forensic data from its data center through the resource interface. The CSPs should also be able to collect the forensic data left on the CSCs side. This makes eminent that the CSPs should remotely connect to the CSCs environments and collect forensic data with the help of the CFaaS system. At this point, it is assumed that the CSPs can add some FDCs to the browser or to the cloud client software commonly used by the CSCs to get access to the cloud services. These FDCs log and preserve the forensic data, including communication logs and other sensitive data, on the client's machine.

The preceding discussions highlighted that the ability of cloud forensics investigation relies on the ability of the parties involved to independently collect the forensic data necessary to produce unilaterally trusted output. Furthermore, as noted before, FDCs may need other protection mechanisms such as tamper-resistance. In this light, the discussion will now be generalized.

Figure 4 shows a provider (p), which offers some service to a consumer (c). FDM_C is a consumer's Forensic Data Manager. FDM_P, refers to a provider's forensic data manager, while TR stands for Tamper-Resistant protection. Therefore, FDM_C^{TR} is a consumer forensic data manager that is tamper-resistant to provider modification, whereas, FDM_P^{TR} is a provider forensic data manager that is tamper-

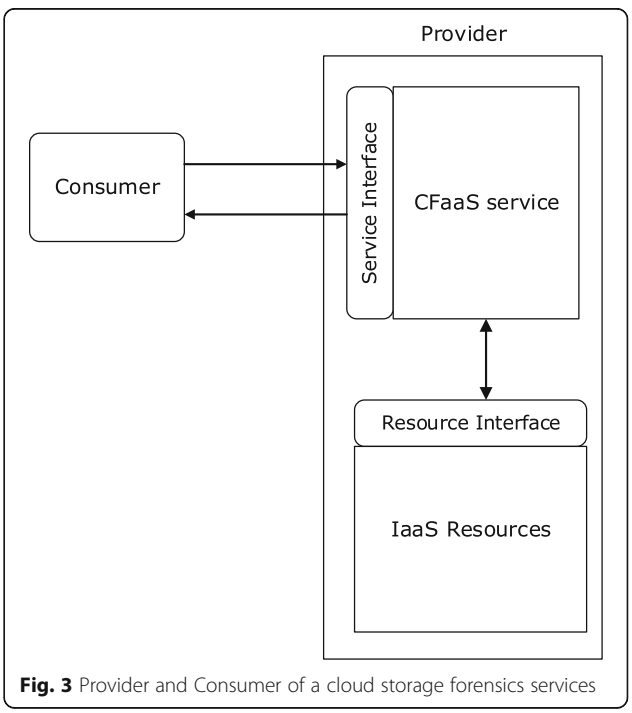

Fig. 3 Provider and Consumer of a cloud storage forensics services

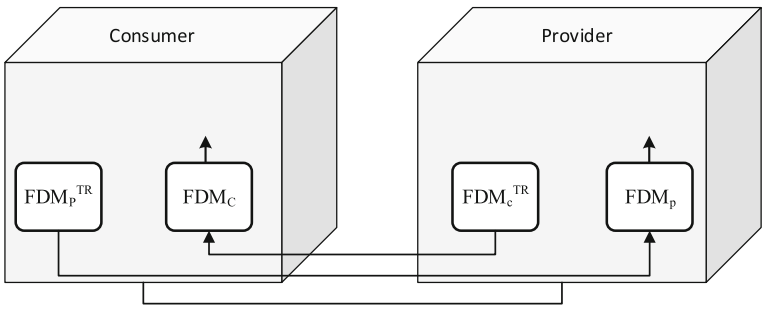

Fig. 4 Model of bilateral data collection

resistant to consumer modification. The outputs of the FDMs are unilaterally trusted, the FDM_C by the consumer and FDM_P by provider, and made available to their respective forensic application managers in the bigger model of the CFaaS system. This model implies that the forensic data should be collected from within the infrastructure of the consumer and from within the provider side, where the forensic data of a required degree of accuracy can be collected from within the consumer and provider environments. In this case, the consumer deploys its FDM_C locally, and FDM_C^{TR} to remotely collect forensic data from the cloud. Likewise, the provider deploys its FDM_P locally in the cloud, and FDM_P^{TR} to remotely collect forensic data at the infrastructure of the consuming organization.

Agreement on mutually trusted forensics output

Though they are functionally equivalent, consumer and provider components in the CFaaS system do not necessarily use the same algorithms or input forensic data to analyze their unilaterally trusted output. As discussed in Section 6.3, they may use data collected at different interfaces to analyze an output. There is then the possibility of divergence between the unilaterally trusted outputs. To address this problem, the authors propose the use of a Comparison and Conflict Resolution Protocol (CCRP). A suitable protocol that supports:

i. The comparison of independently produced and unilaterally trusted outcomes to detect potential divergences;
ii. Where possible, for example, when $| Output_p - Output_c | \leq d$, ($c$, p, d stand for consumer, provider and agreed-upon acceptable divergence, respectively), the immediate declaration of absence of conflict;
iii. Where the divergence is greater than d, the execution of a negotiation protocol between consumer and provider with the intention of reaching agreement on a single output;
iv. When the negotiation protocol fails to sort out the conflict automatically, the declaration of conflict for off-line resolution and search for neutral body, which is in this study referred as the LEA.
v. Production of non-repudiable mutually trusted output to the LEA.

The non-repudiation property is necessary to ensure that neither the consumer nor the provider can subsequently deny execution of the CCRP or the results of that execution. That is, both sides could not subsequently deny their agreement or otherwise to a given forensic output that they submitted to the LEA. Consequently, the LEA needs to correlate or fuse the different outputs to normalize the dispute.

In the meantime, the overall vision behind the correlation and fusion services, in the high level architecture of the proposed CFaaS model depicted in Fig. 5, is to help the LEA correlate and visualize the divergence pertaining to the unilaterally trusted forensic outputs discovered by the CSCs and CSPs. It provides the parsing of multiple forensically interesting objects from the trusted forensics outputs and correlation between them. Finally, it pulls together the results to present a hyperlinked interactive report. The correlation and fusion service use an internal database to store and relate the various conceptual units of the outputs discovered by both sides.

The authors now try to complete the model for bilateral cloud forensics by combining the component services with execution of the CCRP. Figure 6 shows the model when forensic data is collected at both CSCs and CSPs (as in Fig. 4). As before, c and p stand for consumer and provider. FDM, FAM and FWM stand for forensic data manager, forensic application manager and forensic workflow manager, respectively. The root of trust is established at the forensic data manager. This is determined by the level at which the parties execute the CCRP to agree on a mutually trusted output. As indicated in section 6.1, the rationale here is that once a bilaterally trusted output is available, the CSCs can verify any subsequent forensic analysis. This verification is orthogonal to the operation of the bilateral CFaaS system. The model allows for the combination of data from mixed FDM deployments. That is, the consumer deploys

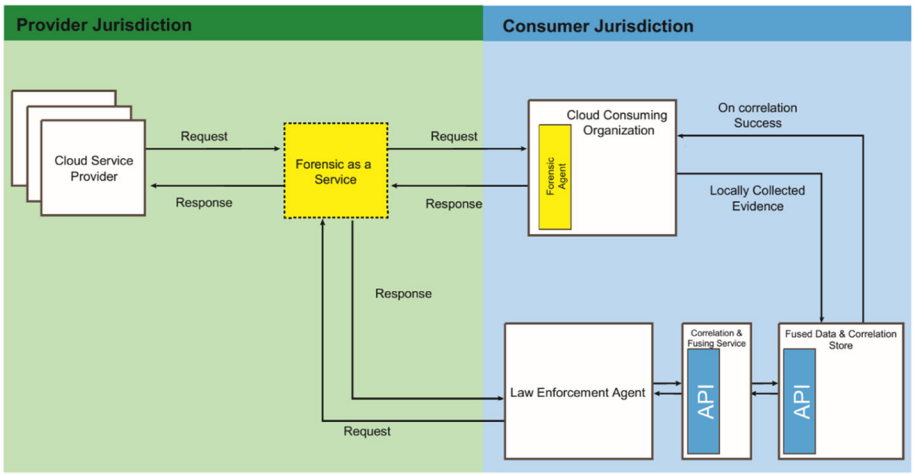

Fig. 5 High Level View of the Bilateral CFaaS

FDM_C and tamper-resistant FDM_P, while the provider deploys FDM_P and tamper-resistant FDM_C (as in Fig. 4).

Case study: On the feasibility of bilateral cloud forensic as a service

A Cloud Service Provider (CSP) offers Storage as a Service (StaaS) and Cloud Forensics as a Service (CFaaS) to thousands of consumers. In this light, a Cloud Service Consumer (CSC) is using the StaaS to store its data on the cloud through the CSP, where an attacker has managed to break into the server hosting the data and stolen confidential information. After conducting the initial response to remediate the incident, the CSP should notify the CSC about the incident, and data collection is initiated to investigate the incident (it is assumed that the CSP and CSC are in the same legal jurisdiction).

Stakeholders collect the forensics data, analyze and present the concerned bodies with the report using the CFaaS system. According to the previous discussion, the CFaaS systems proposed in the literature are based on unilaterally trusted out-comes. Ideally, CSC should have a mechanism to independently investigate the incident and verify that the CSP is not accidentally or maliciously modifying the evidence. This would result in a bilateral CFaaS system.

For this specific scenario of investigating the StaaS model consumed by an application deployed within the consumer's infrastructure, this study envisages a bilateral CFaaS system based on the abstract model illustrated in Fig. 1. In this light, forensic data pertaining to an incident occurred in the StaaS model can literally be collected from the AAA (Authentication, Authorization, Accountability) logs, hypervisor event logs and storage server [49]. As discussed in Section 6.1, both CSC and CSP can collect the necessary forensic data at the CFaaS interface. In the case of FDM_C and FDM_P can independently collect the forensic data and provide to FAM_C and

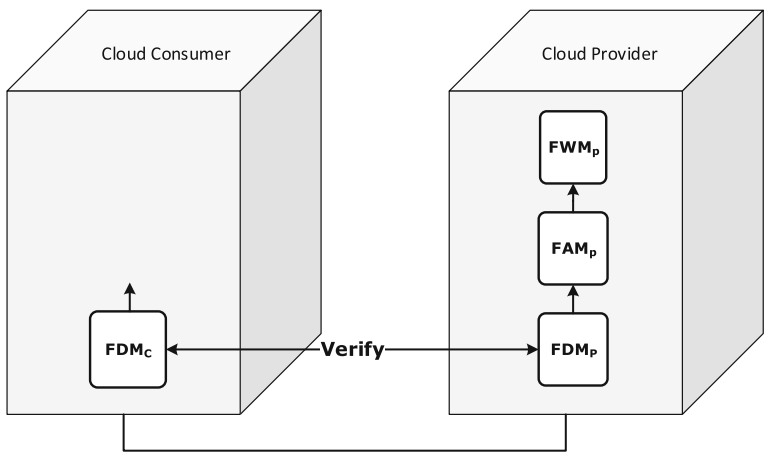

Fig. 6 Bilateral CFaaS model

FAM_P. Given that the forensic data collecting agents are collecting similar data, the Forensic Analysis Modules (FAM) of the CFaaS system should arrive at the same result. Thus, it is more straightforward to bilaterally collect and analyze evidence from all sources of forensic data pertaining to the StaaS model.

In the case of the CSP it is collecting the potential evidence through the resource interface, this allows the CSP to more directly collect potential evidence. This does not allow the FDM_C to collect the same forensic data independently. As suggested in section 6.3, this may lead to different strategies for the collection of forensic data by FDM_C and FDM_P. For example, FDM_C of a CSC can proactively collect provenance metadata of the data transferred to and from the StaaS. The FAM_C locally implemented in the premises of the CSC can then use the forensic data which has been proactively collected by the FDM_C to analyze the incident. However, it is very likely that the results of the analysis produced by the FAM_C will diverge from those produced by the FAM_P, because the FAM_P is able to rely on data collected by the FDM_P, which has access to the resource interface within the CSP premises. In summary, the only independent forensic data available to the CSC are based on proactively collected provenance metadata. These metadata alone cannot provide the CSC with sufficient information to perform own forensic analysis and produce the same results that will be compatible with those produced by the CSP. In such cases, FAM_C and FAM_P may produce divergent unilaterally trusted outcomes.

Therefore, CSC and CSP should execute the Comparison and Conflict Resolution Protocol (CCRP) by first negotiating to resolve the dispute. Here, the CCRP is a peer to peer dispute resolution protocol in a sense that it will be executed between the CSC and CSP without the intervention of a third party. The model utilizes a non-repudiable object sharing middleware for sharing the forensic data among CSC and CSP to form the basis of CCRP [68, 71–73]. The middleware can provide multi-party, non-repudiable agreement to a shared forensic data with the CSC and CSP holding their own copies. Fundamentally, one party proposes new forensic data pertaining to a cloud forensic case, and the other party involved in that same case subjects the proposed forensic data to a case-specific validation. Subsequently, when all parties involved in the case agree upon the validity of the forensic data, the state of the shared forensic data will accordingly be changed into mutually trusted. For the non-repudiable agreement to a change of state: (i) there must be evidence that any proposed forensic data originated at its proposer, and (ii) there must be evidence that all parties agreed to the proposed forensic data and therefore share the same (agreed) view of the state of the forensic data. In other words, there must be evidence that the CSC and CSP involved in the case received the proposed forensic data and that they agreed on the state change (state is either mutually or unilaterally trusted).

Based on this case study, it is considered that the CSP has collected forensic data, FD_P, through the resource interface of the StaaS, and that the CSC has proactively collected forensic data FD_C. Currently, the FD_C and FD_P are unilaterally trusted by the CSC and CSP respectively. Meanwhile, the problem is for the CSC and CSP to reach an agreement on the forensic data that will ultimately be used to investigate this particular cloud forensic case. Hence, the basic consumer-provider agreement process is that the CSP proposes FD_P and the CSC performs case specific validation by comparing the FD_P with FD_C. Subsequently, the CSC returns a decision on the validity, or otherwise, of the FD_P. The middleware utilizes a signed two-phase commit protocol (2PC), an automatic commitment protocol, with application level validation to accomplish the preceding basic agreement.

Figure 7 illustrates the execution of the protocol for the CSP's proposal of the FD_P that has to be agreed with the CSC. Initially, the CSP proposes FD_P with evidence of non-repudiation of its origin ($NRO(FD_P)$). Subsequently, the CSC validates FD_P and returns a decision on its validity or otherwise (decn), non-repudiation of receipt of FD_P ($NRR(FD_P)$) and non-repudiation of origin of the decision (NRO(decn)). The decision is a binary of either Yes or No values. Consequently, the middleware annotates the decision with a case specific information. For instance, the decision may be annotated with the degree of divergence of the proposed FD_P from the CSC's view of the forensic case. Finally, the protocol terminates with the CSP sending non-repudiation of receipt of the CSC's validation decision back to the CSC (NRR(decn)).

If the CSC decides that the FD_P is valid then the mutually agreed set of forensic data that has to be used for the forensic analysis is FD_P. Otherwise, the failure to

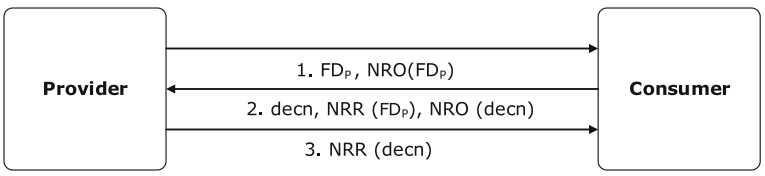

Fig. 7 Non-repudiable sharing of forensic data

agree to the proposed FD_P will be signaled to both CSC and CSP. As with annotations to decisions, this failure signal can be used to initialize reasonable negotiations. At the end of the protocol run, both CSC and CSP have the same irrefutable view of the set of agreed FD_P. Similarly, both parties have an irrefutable validation decision made with respect to proposed FD_P that have been rejected by either CSC or CSP.

If the negotiation fails to resolve the dispute, the two parties will seek help from the LEA to correlate the unilaterally produced outcomes to normalize the dispute. In this regards, the LEA uses the non-repudiated FD_C and FD_P.

Conclusion

As shown, consumers are increasingly relying on ready-made resource services from cloud service providers. Some sources of potential digital evidence are on the premises that the provider unilaterally producing forensic data. In cloud forensic there is a need for investigators to adapt and develop a cloud forensic process model that would enable forensic investigators to collect and analyze forensic data on the consumer and provider sides. The live cloud forensic process model presented in this article illustrates this issue and provides a basis for the development of new tools in cloud forensics.

Hence, with this process model, the authors believe that the practice of provider-side data collection may need to be supplemented by measures that enable cloud consumers to produce their own forensic data, minimally, to verify the reasonableness of the provider produced data. Bilaterally collection of digital evidence may become a next logical step: the consumer and the provider independently collect and analyze the forensics data, compare their outcomes and agree on a mutually trusted outcome. We took a plausible cloud forensic scenario, which employs a bilateral cloud forensic as a service model to investigate an incident involved in a storage as a service model, to highlight the issues involved. The success of the bilateral cloud forensic as a service (CFaaS) to a large extent will depend on two factors: the quality of forensic data consumers can collect and the availability of a relatively simple comparison and conflict resolution protocol (CCRP) to enable production of mutually agreed outcomes. Service providers can help consumers by providing: (i) a suitable service interface to enable consumer side forensic analysis, and (ii) a reference analysis tool to enable consumers to estimate source of the security incident. Further, as we discussed, sometimes there is also a need for a consumer (provider) to collect forensic data directly at the provider's (consumer's) premises, so suitable forensic analysis techniques will need to be developed. CCRP procedures will also need to be developed and agreed as a part of the service level agreement.

Abbreviations

AAA: Authentication, Authorization, Accountability; API: Application Programming Interface; BYOD: Bring Your Own Device; CCRP: Comparison and Conflict Resolution Protocol; CF: Cloud Forensics; CFaaS: Cloud-Forensics-as-a-Service; CSCs: Cloud Service Consumers; CSPs: Cloud Service Providers; DoS: Denial of Service; ENISA: European Network and Information Security Agency; FAM: Forensic Application/Analysis Manager; FDC: Forensic Data Collector; FDM: Forensics Data Manager; FTK: AccessData's Forensics Tool Kit; FWM: Forensics Workflow Manager; HDFS: Hadoop Distributed File System; HTTPs: Secure Hypertext Transfer Protocol; IaaS: Infrastructure-as-a-Service; LEAs: Law Enforcement Agents; MD5: Message Digest 5; NIST: National Institute of Standards and Technology; OAuth: Open Authentication; PaaS: Platform-as-a-Service; S3: Amazon's Simple Storage Service; SaaS: Software-as-a-Service; SLAs: Service Level Agreements; TTP: Trusted Third Party

Acknowledgements

The first author is supported by a scholarship from the Ministry of Higher Education of Malaysia (MOHE) under the Malaysian International Scholarship (MIS) scheme.

Funding

Not applicable

Availability of data and materials

Not applicable

Authors' contributions

All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]Faculty of Computing, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia. [2]Faculty of Computing, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia. [3]Faculty of Computing, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia.

Received: 7 February 2017 Accepted: 21 December 2017
Published online: 05 January 2018

References

1. Group, (2014) N.C.C.F.S.W., NIST Cloud Computing Forensic Science Challenges (Draft NISTIR 8006)
2. Moussa, A.N., N. Binti Ithnin, and O.A. Miaikil (2014) Conceptual forensic readiness framework for infrastructure as a service consumers. In Systems, Process and Control (ICSPC), 2014 IEEE Conference on. IEEE
3. Hay B, Nance K (2008) Forensics examination of volatile system data using virtual introspection. ACM SIGOPS Operating Systems Review 42(3):74–82
4. Birk, D. and C. Wegener (2011) Technical issues of forensic investigations in cloud computing environments. In Systematic Approaches to Digital Forensic Engineering (SADFE), 2011 IEEE Sixth International Workshop on. IEEE
5. Dykstra J, Sherman AT (2012) Acquiring forensic evidence from infrastructure-as-a-service cloud computing: exploring and evaluating tools, trust, and techniques. Digit Investig 9:S90–S98
6. Zawoad S, Hasan R (2013) Cloud forensics: a meta-study of challenges, approaches, and open problems. arXiv preprint arXiv 1302:6312
7. Zawoad S, Hasan R, Grimes J (2015) LINCS: towards building a trustworthy litigation hold enabled cloud storage system. Digit Investig 14:S55–S67
8. Ruan K et al (2013) Cloud forensics definitions and critical criteria for cloud forensic capability: an overview of survey results. Digit Investig 10(1):34–43
9. Gebhardt T, Reiser HP (2013) Network Forensics for Cloud Computing. In Distributed Applications and Interoperable Systems. Spring
10. Pichan A, Lazarescu M, Soh ST (2015) Cloud forensics: technical challenges, solutions and comparative analysis. Digit Investig 13:38–57
11. Alqahtany S et al (2015) A forensic acquisition and analysis system for IaaS. Clust Comput:1–15

12. Damshenas, M., et al (2012) Forensics investigation challenges in cloud computing environments. In Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on. IEEE
13. Ruan K, et al (2011), Cloud forensics, in Advances in digital forensics VII, Springer, Dordrecht. p. 35–46
14. Hay, B., K. Nance, and M. Bishop (2011) Storm clouds rising: security challenges for IaaS cloud computing. In System Sciences (HICSS), 2011 44th Hawaii International Conference on. IEEE
15. Guo, H., B. Jin, and T. Shang (2011) Forensic investigations in cloud environments. In Computer Science and Information Processing (CSIP), 2012 International Conference on. IEEE
16. Roussev V, Quates C, Martell R (2013) Real-time digital forensics and triage. Digit Investig 10(2):158–167
17. Kohn MD, Eloff MM, Eloff JH (2013) Integrated digital forensic process model. Computers & Security 38:103–115
18. Reith M, Carr C, Gunsch G (2002) An examination of digital forensic models. International Journal of Digital Evidence 1(3):1–12
19. Palmer, G (2001) A road map for digital forensic research. in First Digital Forensic Research Workshop, Utica, New York.
20. Carrier B, Spafford EH (2003) Getting physical with the digital investigation process. International Journal of digital evidence 2(2):1–20
21. Baryamureeba, V. and F. Tushabe (2004) The enhanced digital investigation process model. In Proceedings of the Fourth Digital Forensic Research Workshop. Citeseer
22. Beebe NL, Clark JG (2005) A hierarchical, objectives-based framework for the digital investigations process. Digit Investig 2(2):147–167
23. Kent K et al (2006) Guide to integrating forensic techniques into incident response. NIST Special Publication:800–886
24. Selamat SR, Yusof R, Sahib S (2008) Mapping process of digital forensic investigation framework. International Journal of Computer Science and Network Security 8(10):163–169
25. Cohen FB (2010) Fundamentals of digital forensic evidence. In: Handbook of Information and Communication Security. Springer, Dordrecht, pp 789–808
26. Valjarevic, A. and H.S. Venter (2012) Harmonised digital forensic investigation process model. in Information Security for South Africa (ISSA), 2012. IEEE
27. Martini B, Choo K-KR (2012) An integrated conceptual digital forensic framework for cloud computing. Digit Investig 9(2):71–80
28. Quick D, Martini B, Choo R (2013) Cloud storage forensics. Syngress, Waltham
29. Zawoad, S., R. Hasan, and A. Skjellum (2015) OCF: An Open Cloud Forensics Model for Reliable Digital Forensics. In Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on.IEEE
30. Ferguson-Boucher, K. and B (2012) Endicott-Popovsky, Forensic Readiness in the Cloud (FRC): Integrating Records Management Cybercrime and Cloud Forensics: Applications for Investigation Processes: Applications for Investigation Processes, p. 105
31. Endicott-Popovsky B, Frincke DA, Taylor CA (2007) A theoretical framework for organizational network forensic readiness. Journal of Computers 2(3):1–11
32. Sibiya, G., et al (2013) Digital forensic readiness in a cloud environment. in AFRICON, 2013. IEEE
33. Trenwith, P.M. and H.S. Venter (2013) Digital forensic readiness in the cloud. in Information Security for South Africa, 2013. IEEE
34. De Marco, L., F. Ferrucci, and T. Kechadi (2014) Reference architecture for a cloud forensic readiness system.
35. De Marco, L., et al (2014) Formalization of SLAs for Cloud Forensic Readiness. in Proc. ICCSM Conference.
36. Makutsoane, M.P. and A. Leonard (2014) A conceptual framework to determine the digital forensic readiness of a Cloud Service Provider. In Management of Engineering & Technology (PICMET), 2014 Portland International Conference on. IEEE
37. Federici C (2014) Cloud data imager: a unified answer to remote acquisition of cloud storage areas. Digit Investig 11(1):30–42
38. Oestreicher K (2014) A forensically robust method for acquisition of iCloud data. Digit Investig 11:S106–S113
39. Cahyani NDW et al (2016) Forensic data acquisition from cloud-of-things devices: windows smartphones as a case study. Concurrency and Computation, Practice and Experience
40. Anwar, F. and Z. Anwar (2011) Digital forensics for eucalyptus. in Frontiers of Information Technology (FIT), 2011. IEEE
41. Marturana, F., G. Me, and S. Tacconi (2012) A case study on digital forensics in the cloud. In Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2012 International Conference on. IEEE
42. Hale JS (2013) Amazon cloud drive forensic analysis. Digit Investig 10(3): 259–265
43. Quick D, Choo K-KR (2013) Dropbox analysis: data remnants on user machines. Digit Investig 10(1):3–18
44. Quick D, Choo K-KR (2013) Digital droplets: Microsoft SkyDrive forensic data remnants. Futur Gener Comput Syst 29(6):1378–1394
45. Quick D, Choo K-KR (2014) Google drive: forensic analysis of data remnants. J Netw Comput Appl 40:179–193
46. Shariati, M., et al., (2015) Ubuntu one investigation: detecting evidences on client machines.
47. Cho C, Chin S, Chung KS (2012) Cyber forensic for hadoop based cloud system. International Journal of Security and its Applications 6(3):83–90
48. Chung H et al (2012) Digital forensic investigation of cloud storage services. Digit Investig 9(2):81–95
49. Spyridopoulos, T. and V. Katos (2012) Data Recovery Strategies for Cloud Environments Cybercrime and Cloud Forensics: Applications for Investigation Processes: Applications for Investigation Processes,: p. 251
50. Martini B, Choo K-KR (2013) Cloud storage forensics: own cloud as a case study. Digit Investig 10(4):287–299
51. Martini B, Choo K-KR (2014) Distributed filesystem forensics: XtreemFS as a case study. Digit Investig 11(4):295–313
52. Thethi, N. and A. Keane. Digital forensics investigations in the cloud. In Advance Computing Conference (IACC), 2014 IEEE International. 2014. IEEE
53. Daryabar F, Dehghantanha A, Choo K-KR (2016) Cloud storage forensics: MEGA as a case study. Australian Journal of Forensic Sciences:1–14
54. Martini, B. and K.-K.R. Choo. Remote programmatic vCloud forensics: a six-step collection process and a proof of concept. In Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on. 2014. IEEE
55. McKemmish, R., What is forensic computing? 1999: Australian Institute of Criminology Canberra
56. Povar D, Geethakumari G (2014) A Heuristic Model for Performing Digital Forensics in Cloud Computing Environment, in Security in Computing and Communications. Springer, Dordrecht, pp 341–352
57. Ab Rahman NH, Choo K-KR (2015) A survey of information security incident handling in the cloud. Computers & Security 49:45–69
58. Ab Rahman N, Choo K (2015) Integrating digital forensic practices in cloud incident handling: A conceptual cloud incident handling model. Cloud Security EcosystemR
59. Ab Rahman, N.H., N.D.W. Cahyani, and K.K.R. Choo (2016) Cloud incident handling and forensic-by-design: cloud storage as a case study. Concurrency and Computation: Practice and Experience
60. Ab Rahman NH et al (2016) Forensic-by-design framework for cyber-physical cloud systems. IEEE Cloud Computing 3(1):50–59
61. Simou, S., et al (2015) Towards the Development of a Cloud Forensics Methodology: A Conceptual Model. In Advanced Information Systems Engineering Workshops. Spring
62. Yan, C (2011) Cybercrime forensic system in cloud computing. In Image Analysis and Signal Processing (IASP), 2011 International Conference on. IEEE
63. Delport, W., M. Köhn, and M.S (2011) Olivier. Isolating a cloud instance for a digital forensic investigation. in ISSA.
64. Marty, R (2011) Cloud application logging for forensics. In Proceedings of the 2011 ACM Symposium on Applied Computing. ACM
65. Dykstra J, Sherman AT (2013) Design and implementation of FROST: digital forensic tools for the OpenStack cloud computing platform. Digit Investig 10:S87–S95
66. Zawoad, S., A.K. Dutta, and R. Hasan (2013) SecLaaS: secure logging-as-a-service for cloud forensics. In Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM
67. Patrascu A, Patriciu V-V (2015) Logging for cloud computing forensic systems. International Journal of Computers Communications & Control 10(2):222–229
68. Molina-Jimenez, C., N. Cook, and S. Shrivastava (2008) On the feasibility of bilaterally agreed accounting of resource consumption. In Service-Oriented Computing–ICSOC 2008 Workshops. Spring
69. Van Oorschot PC (2003) Revisiting software protection, in Information Security. Springer, Dordrecht, pp 1–13
70. Adolph M, Sutherland E, Levin A (2009) Distributed computing: utilities, grids & clouds. International Telecommunication Union-Technology Watch Report 9

71. Cook, N., S. Shrivastava, and S. Wheater (2002) Distributed object middleware to support dependable information sharing between organisations. In Dependable Systems and Networks, 2002. DSN 2002. Proceedings. International Conference on. IEEE
72. Robinson, P., N. Cook, and S. Shrivastava (2005) Implementing fair non-repudiable interactions with web services. In EDOC Enterprise Computing Conference, 2005 Ninth IEEE International. IEEE
73. Cook N, Robinson P, Shrivastava SK (2006) Design and implementation of web services middleware to support fair non-repudiable interactions. International Journal of Cooperative Information Systems 15(04):565–597