Faculty of Science and Engineering
School of Electrical Engineering, Computing and
Mathematical Sciences

Digital Forensics Investigation Frameworks for Cloud Computing and Internet of Things

Ameer A Pichan

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

June 2022

Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any University.

Ameer A Pichan

"***Connecting the dots....*** *it (i.e., events in life) is impossible to connect the dots looking forward, but it is very clear when connecting the dots backward. Again, you cannot connect the dots looking forward; you can only connect them looking backward (Steve Jobs)*
Digital forensics is all about connecting the dots backward so that every connection of dots makes sense to prove some events occurred in the past. "

— AUTHOR

Acknowledgements

My research journey, which I ventured as a very mature student, would not have been possible without the support of many people. In particular, I would like to express my sincere gratitude and thanks to my supervisors, Associate Professor Mihai Lazarescu and Dr. Sie Teng Soh. Their guidance and invaluable advice motivated and set me in the right direction. Further, their meticulous attention to detail encouraged me to strive for perfection in my research and thesis writing. It was a privilege and pleasure to have both of you as my supervisors.

I would also like to sincerely thank Professor Ba-Tuong Vo, an inspirational force whose suggestions were awesome. My special thanks also go to research colleagues Tony, Lely Hiryanto, Yuthika Punchihewa, Dat Nguyen for their help, collaborations and intellectual discussions. I also want to express my sincere gratitude to the school administrative staff for their timely support.

I gratefully acknowledge the financial support received from Curtin University for part of my study. I also want to thank all the anonymous reviewers and editors of journals and conferences for their time, effort, and comments. Their suggestions have certainly raised the quality of my publications.

More than anyone else, my deepest gratitude goes to my wife, Shahitha Ameer, for her care, encouragement, and unwavering support throughout my Ph.D journey. She endured long days of separation, partly because of my work elsewhere,

and long hours at the University, when I was immersed in the research. Without her prayers and steadfast support, this doctoral program would not have been possible to complete.

My sincere gratitude also goes to my son, Irfan, his wife, Nilofar, and little granddaughter, Ifza. Their constant encouragement during family time kept me motivated. Also, I am grateful to my brothers, sisters, and in-laws for their push to get to the finish line. On this occasion, I fondly remember my late father, who was always an inspiration and role model, and firmly stood behind my every ambitious journey in my life and career.

Above all, I praise the Almighty God. We repose our trust in him and always seek his righteous guidance, whom his generosity and everlasting blessing showered upon me throughout my life.

Abstract

Cloud computing and Internet of Things (IoT) are two technologies that have drastically changed how Information Technology (IT) delivers services. However, the rapid growth in these technologies potentially introduces new vulnerabilities that can be exploited to mount cyber-attacks. A digital forensics investigation is then commonly used to find the culprit responsible for the attacks as well as to help expose the vulnerabilities. Unfortunately, due to the unique characteristics of cloud computing and IoT technologies, traditional forensics tools and methods are not suitable for use in both technologies. Hence, more specific digital forensics investigation frameworks and methodologies are required. This deficiency motivates research in this thesis on frameworks and methods for digital forensics in cloud computing and IoT platforms.

The *first* research work in this thesis focuses on identifying and recommending suitable solutions to address cloud forensic challenges. In this work, we systematically survey the forensic challenges and analyze their most recent solutions. In particular, we describe the issues using the phases of traditional digital forensics as the base. For each phase of digital forensics process, we include a list of challenges and analysis of the solutions addressing the challenges. Our description helps identify the differences between the problems and solutions for traditional and cloud digital forensics. Further, the presentation is expected to help the

investigators better understand the problems in the cloud environment.

The *second* work in this thesis focuses on developing a cloud forensics framework, which can be used to detect cyber-crimes on cloud platforms. In this work, we present a practical log architecture framework by analyzing it from forensic practitioners' needs and perspectives. We demonstrate the framework on an ownCloud platform. We assess usability and applicability of the framework by validating it against two set of requirements, one of which is specified by Association of Chief Police Officers (ACPO), and the other by National Institute of Standards and Technology (NIST). Further, the relevance of the framework has been demonstrated by comparing it against other recent frameworks.

The *third* work in this thesis focuses on developing a systematic methodology for assessing the forensics readiness and compliance levels of public Cloud Service Providers (CSPs). More specifically, this work (i) proposes a systematic approach that specifies the cloud forensic requirements, (ii) defines the process for evaluating the forensics compliance levels, and (iii) shows **where** to look for and **how** to extract the crime evidence. We evaluated the methodology on three major public CSPs , viz., Amazon Web Services, Microsoft Azure, and Google Cloud platforms by using a case study of a crime incident. Further, our assessment (i) identifies and quantifies the gaps which the CSPs failed to satisfy, and (ii) provides a comprehensive view of the cloud forensics maturity levels of the three providers.

The *fourth* work of this thesis focuses on designing a framework to enhance the IoT forensics capabilities. More specifically, this work (i) summarizes IoT attack taxonomy and forensics challenges (ii) defines IoT forensics requirements, and (iii) proposes a conceptual IoT event logging model supporting forensics. Finally, this work provides the design architecture for the framework, using an integrated Cloud-IoT environment. The framework has been assessed by evaluating it against the requirements.

List of Publications

This thesis is based upon several works that have been published in journals and presented at conferences over the course of the author's PhD. They are listed as follows:

1. **A. Pichan**, M. Lazarescu, and S. Soh, "Cloud forensics: Technical challenges, solutions and comparative analysis," Digital investigation, vol. 13, pp. 38–57, 2015 (cited 158 times)

2. **A. Pichan**, M. Lazarescu, and S. Soh, "Towards a practical cloud forensics logging framework," Journal of Information Security and Applications, vol. 42, pp. 18-28, Oct. 2018. (cited 20 times)

3. **A. Pichan**, M. Lazarescu, and S. Soh, "A case study on major cloud platforms digital forensics readiness – are we there yet?" International Journal of Cloud Computing, (Accepted, in press).

4. **A. Pichan**, M. Lazarescu, and S. Soh, "A logging model for enabling digital forensics in IoT, in an inter-connected IoT, cloud eco-systems," in 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4). IEEE, 2020, pp. 478–483. (cited 8 times)

5. **A. Pichan**, M. Lazarescu, and S. Soh, "Can nuclear installations and research centers adopt cloud computing platform," in Symposium on International Safeguards Linking Strategy, Implementation and People. IAEA, 2015, vol. IAEA-CN–220, pp. 1–9. (cited 1 time)

Table 1: Attribution statement for all published works

	Conception, Problem Design & Model	**Problem Solution**	**Review**	**Validation**	**Interpretation and Discussion**	**Final Approval**
Primary Author Ameer Pichan	✓	✓		✓	✓	
Primary Author Acknowledgement: I acknowledge that these represent my contribution to the above research output Signed:						
Co-Author 1 Mihai Lazarescu			✓			✓
Co Author 1 Acknowledgement: I acknowledge that these represent my contribution to the above research output Signed:						
Co-Author 2 Sieteng Soh	✓		✓		✓	✓
Co Author 2 Acknowledgement: I acknowledge that these represent my contribution to the above research output Signed:						

Contents

List of Figures

List of Tables

Acronyms

CSP	Cloud Service Provider
IoTs	Internet of Things
CSU	Cloud Service User
IaaS	Infrastructure-as-a-Service
PaaS	Platform-as-a-Service
SaaS	Software-as-a-Service
DoC	Degree of Control
NIST	National Institute of Standards and Technology
ACPO	Association of Chief Police Officers
AWS	Amazon Web Services
GCP	Google Cloud Platform
AD	Active Directory
VM	Virtual Machine
EC2	Elastic Cloud Computing
SQL	Structured Query Language
SLA	Service Level Agreement
LEA	Law Enforcement Agency
API	Application Programming Interface
CSA	Cloud Security Alliance
ENISA	European Network and Information Security Agency
DIP	Digital Investigative Process
S3	Simple Storage Service
JSON	Java Script Object Notation
ISO	International Standards Organization
IEC	International Electrotechnical Commission
UTC	Coordinated Universal Time
IAM	Identity and Access Management
5W1H	Who, What, When, Where, Why, and How elements of a cyber-crime

Chapter 1

Introduction

Cloud computing is an innovative way of delivering computing services. Services such as computing power, servers, storage, databases, networking, and application services, all over the internet ("the cloud"), where consumers pay for their service usage. Cloud computing offers faster innovation, speed and agility, and economies of scale, revolutionizing the computing industry in recent years. As a result, there is tremendous growth in the cloud computing market space. Recent market study statistics show that the cloud computing market in 2021 was USD 445.3 billion and is forecasted to grow to USD 791.5 billion by 2028, as reported by leading market research organizations [1, 2]. The Covid pandemic outbreak at the beginning of 2020 fuelled the growth as more business and educational institutions suddenly moved to online mode, subscribing to more cloud-hosted services [1].

As it exists now, cloud computing platforms have drastically improved since their inception. Today's cloud environment can provide much more advanced services, with many facets and a customer-oriented service portfolio, making cloud platforms very diverse. Cloud computing faced a trust issue initially, and organizations were reluctant to shift their data to the cloud [3]. Also, customers were concerned about the security, privacy, legal, and jurisdictional aspects of the cloud environment and how cloud computing stores and processes customers'

data [4], causing inhibition for cloud adoption, primarily during the early days of cloud computing. However, the technological advancements led to more sophisticated implementations of cloud services, increasing the customers' confidence in the cloud. Further, the advantages and benefits that the cloud offers outweigh its shortcomings. The availability and service delivery models in any size and configuration made the cloud an appropriate platform for any organization or institution. It also helped avoid any capital expenditure and replace it with pay-for-use or an operational expenditure model. Additionally, the increasing availability of human resource expertise in the cloud domain curbed a major cloud adoption challenge. Cloud computing paved the way for innovations and advancements in two other parallel technologies, i.e., IoT and Big data analytics [3], resulting in the rapid growth of the cloud computing adoption rate.

Increasing adoption of the cloud also attracted more sophisticated adversaries to the cloud. Cloud platforms offered an extensive value proposition for cyber criminals, using the technology to scale up their operations. Though the size and magnitude of the attacks varied, the attacks often resulted in enormous costs for organizations and often caused significant disruptions to human life. (E.g: Ransomware attacks on a (i) major gas supply pipeline and (ii) big health care service provider during May 2021). A leading information security study reports that cyber crime costs organizations nearly $1.79 million per minute [5]. There has been a surge in attack since the start of Covid-19, as there was a big surge in online business [5]. However, most cloud-based cybercrimes do not even go to court because of various issues surrounding forensics investigations. Such as blurred jurisdictional boundaries, lack of physical access to storage media, reliance on CSPs for data acquisition, questions regarding evidence admissibility, uncertainties surrounding data ownerships, and weak chain of custody proofs [6]. For example, as per Australian law, for cybercrime to be admissible in an Australian court, the perpetrator must be in Australia while committing the crime, or an Australian citizen in an overseas country, with which Australia has

an extradition treaty [7]. If not, Australian courts have no jurisdiction. Another major reason which acts as an inhibition to undertake digital investigation is the huge cost of investigation. The 2021 report by IBM security research team reports that the total cost of post-breach investigation amounts to 27% and detection and escalation amounts to 29% of the total cost of the breach [8].

Digital forensics includes the technology, processes, and methods carried out as a post-crime activity, with the primary objective of finding the culprit responsible for the crime based on credible evidence [9, 10]. Digital forensics applied to the cloud environment is referred to as cloud forensics. Though, we are using digital forensics and cloud forensics terminology interchangeably. Extending and developing digital forensics capabilities to the cloud environment is essential. The reason is because the traditional digital forensics methods and processes are unsuitable for cloud environments [11–13]. The core attributes of cloud computing, such as multi-tenancy, unknown data location, data volatility, and jurisdiction, threaten forensics data visibility since the critical information, including evidence artifacts, can be stored in unidentified virtual servers "somewhere in the cloud". Therefore, typical search & seizure operation is not practical for the cloud since they rely on unrestricted access to the relevant systems and user data, which is not available in the cloud.

Cloud forensics investigation serves two fundamental objectives. First, it can help find out the criminal and do justice. Second, to find out the modus operandi of the crime action, i.e., how the cloud system was exploited to conduct the crime. By studying the modus operandi, one can design better systems for the future, such that the repeat of the same breach using the same scenario can be stopped, helping to improve the system defense. Furthermore, given that seldom cloud-based cybercrimes go to court, it is essential to understand the method of exploitation. Therefore, the primary motivation to undertake this research project is to enhance further cloud forensics capabilities, such that our work contributes to resolving the fundamental objectives mentioned above.

During the first stage of this research, the extensive literature study revealed that the academic community has taken up and provided due importance to the development of cloud forensics. However, we observed that many diverse sets of processes and solutions had been proposed addressing various cloud forensics challenges, all, as a disjoint set of activities. From a forensics investigation point of view, the disjoint set of activities provided little value. To this end, the *first* important issue taken up in this research project was to assess the current state of cloud forensics by doing a comprehensive analysis of related research work. Then, based on the study, recommend suitable solutions to address cloud forensics challenges faced during the different stages of an investigative process, along with the relative merits of the solutions explained. The information will be much valuable to the forensics practitioners and security professionals.

The *second* important issue of forensics interest is the necessity to produce an event trail, recording key investigative parameters. Any forensics investigation is a post-incident activity, the same applies to digital forensics. The key investigative parameters that any forensics seeks to answer are **what** the crime was, **who** did it, **when** it was done, **where/which** systems were compromised, **how** it was done, and finally, **why** it was done (or the motivation) for the crime. In digital forensics, this key investigative parameter is also referred to by its abbreviation 5W1H. Therefore, recording an event trail by capturing sufficient information to answer the 5W1H questions and storing it in persistent storage would help the forensics investigation and help to trace the modus operandi of the crime. However, it is worth mentioning that the **why** aspect of the crime (i.e., the motivation) may not be directly evident from an event log, but can be deduced by evaluating other factors presented in the log. We are using the 5W1H scenario throughout this research project.

As already noted, cloud services adoption has been steadily increasing, and major service providers have been releasing innovative services. Therefore, it is worth assessing major CSPs, cloud forensics readiness, or compliance levels. This

is the *third* important issue addressed here, i.e, developing a repeatable methodology and process to assess the forensic readiness of cloud platforms. This part of the research project assessed three major CSPs' cloud forensics readiness and compliance levels using the defined methodology and a digital crime case scenario. We chose Amazon AWS, Microsoft Azure, and Google Cloud Platform (GCP) for the assessment. These CSPs were chosen based on their market share and services they offer. Leading market research companies identified that AWS, Azure and GCP holds the major market share to date [14–16]. The assessment focused on the evidence available in the CSPs prime logging services, viz., AWS CloudTrail, Azure Activity Logs, and Google Stackdriver logs. Though the assessment provided the forensics readiness at the time of the study, one can assess the readiness any time following the defined methodology and process.

Internet of Things (IoT) is another significant wave of new technology making inroads to every aspect of human life. IoT allows billions of intelligent devices to connect, communicate and control "things", providing many value-added services, improving quality of life and human health. The fundamental objective is to have the IoT devices that self report in real-time, improving efficiency and quickly bringing important information to the surface without depending on human intervention. The rapid growth and innovations in the IoT technology space are happening at a tremendous rate, reaching to trillions of dollars within next five years [17, 18]. However, IoT systems are also prone to security vulnerabilities and exploitation [19–21]. Also, the limited capacity of an IoT device restricts designing it with higher levels of security capabilities embedded in it [22]. This leads to the *fourth* essential and challenging issue undertaken in this study, i.e., contributions to the advancement of IoT forensics capabilities. The IoT poses a new set of challenges to forensics. The proprietary and heterogeneous architecture, limited computing and memory capacity, small physical size, deployed locations, (including hazardous, inaccessible locations) and mobility makes IoT forensics more complex. Due to such unique challenges, typical digital forensics methods

and traditional search and seizure are impractical for IoT [23–25]. Therefore, in this work, we are proposing a new IoT forensics framework.

Several IoT forensics frameworks have been proposed, e.g., [26–33]; see Section 2.3.2 for their detailed discussion. Though each framework has its merit, none addresses the investigative forensics requirements and key challenges of IoT forensics. The proposed model in this thesis defines and describes IoT forensics requirements and challenges, then solves the issue. Further, the model generates, secures, and preserves the evidence in a log repository, making the acquisition of evidence much easier in an investigation. The framework also provides design guidelines to make IoT systems forensic ready.

1.1 Research Objectives

The aims of this research project are as follow:

Objective 1: To conduct an analytical study on cloud forensics process, challenges and solutions, and then propose suitable solutions and their relative merit addressing every cloud forensics challenges. The aim is to enable forensics investigators and security professionals to quickly understand the cloud forensics issues to be aware of during every phase of a digital investigation process and provide a detailed explanation of the solutions presented.

Objective 2: To propose and demonstrate a forensics logging framework for cloud computing. The framework has been crafted, considering the cloud forensics requirements and the 5W1H investigative questions. The aim is to guide mainly the CSPs in developing forensics capable cloud services and to demonstrate the suitability of the framework as a practical cloud forensics framework.

Objective 3: To develop a methodology and process to assess the forensics readiness of cloud platforms systematically. Further, demonstrate the usability

and applicability of the methodology. Following the methodology, we assessed the forensics compliance and maturity levels of three major public CSPs. It describes, **what** evidence are available, **where** to find the evidence, and **how** to extract them. The aim is to help the investigators and cloud service users (CSUs) guide them in investigations and let them know what forensics support can be expected.

Objective 4: To propose an IoT forensics framework. The framework has been designed by considering the IoT forensics challenges and requirements. The framework uses an integrated cloud-IoT platform. Further, it proposes the design architecture and parameters to log. The aim is to help the IoT system designers to build the forensic capable IoT, such that IoT-based attacks can be traced.

1.2 Significance and Contributions

The main contributions and significance of this thesis are described below. The specific contributions and importance of each project are further detailed in the respective chapter.

1. It presents various digital forensics investigative models. Using a well-known digital investigative model [34], it identifies the challenges associated with each process phase. Further, it provides a comprehensive analysis of the existing cloud forensic solutions and recommends suitable solution(s) to address specific cloud forensics challenges based on an analytical study. Finally, it analyses the relative merit of every solution presented.

2. It identifies and presents forensics investigative requirements from a practitioner's point of view, proposes a cloud forensics logging framework addressing the identified needs, and demonstrates the framework on an ownCloud platform. It validates the framework against a well know investigative guidelines i.e., ACPO guidelines [35] and NIST Cloud Computing Forensic

Science Challenges (NISTIR 8006) [36], demonstrating the relevance and suitability of the framework as a practical framework. Further, the relative merit of the framework has been shown by comparing it against other recent frameworks.

3. It develops a systematic methodology to evaluate the forensics readiness of cloud computing platforms. Using the methodology it evaluates and presents the cloud forensics readiness and compliance level of three major public CSPs. Further, it identifies and quantifies the gaps in forensics compliance and presents a comprehensive analysis of the state of cloud forensics maturity levels at the time of the study. More importantly, it presents a repeatable methodology to evaluate cloud forensics readiness. Given that new cloud services are released more often, one can assess or re-assess the forensics readiness at any given time using the methodology.

4. It identifies IoT forensics investigative requirements. Based on the requirements, it presents an IoT event logging and data collection framework, specifies the log architecture, and identifies the elements to log. Further, it proposes the log collection design and describes how to design of forensics-capable IoT.

1.3 Thesis Organization

The contents of each chapter in this thesis are as follows.

Chapter 2: Background

Chapter 2 discusses the background information on cloud computing, forensics, digital forensics, and cloud forensics. Further, it describes the cloud forensics dimensions and different forensics models. The chapter also details the related work in the same domain. Finally, the chapter provides an overview of IoT, IoT

forensics, including the IoT forensics landscapes and frameworks.

Chapter 3: Cloud Forensics: Process, Challenges, and Solutions

Chapter 3 describes the cloud forensics process and challenges commonly faced in a digital investigation. Further, it provides a comprehensive analysis of the challenges and solutions. Finally, it identifies and proposes solutions to the challenges, describing their relative merits. The material and contributions presented in this chapter have been published in the author's journal article [37]. Since its publication, the paper has been cited many times. Therefore, this chapter also discusses the research impact of this work by analyzing how the cited researchers have used the contributions made in [37].

Chapter 4: Forensics Logging framework for cloud Computing

Chapter 4 formulates the cloud forensics requirements and models the cloud forensics logging framework. It also designs and develops the proposed framework, termed as CFLOG, on an ownCloud platform. Further, it demonstrates and validates the CFLOG framework, establishing its suitability for practical use. Finally, the chapter compares the CFLOG framework with three other forensics frameworks, taking NIST cloud forensics challenges as the key for comparison. The purpose of the comparison is to highlight the significance and relevance of our model and its relative merit.

Chapter 5: A Method to Assess Forensics Readiness of Cloud Computing

Chapter 5 develops a formal and repeatable methodology for assessing the cloud forensics readiness and compliance levels of CSPs. It demonstrates the methodol-

ogy, by assessing the readiness of AWS, Azure, and GCP, using the methodology, and subscribing to the live cloud services from the respective CSPs. The chapter contains the acquired cloud services, their configurations, and the testing process. The chapter also describes the crime use case scenarios used in the study. It also describes where to find the evidence, the format and develops scripts to extract the relevant evidence artifacts from the log archive. Finally, the chapter provides the compliance results in a matrix format in the Appendix.

Chapter 6: Forensics Logging Framework for IoT

Chapter 6 proposes a logging framework to resolve IoT forensics issues. The chapter describes IoT forensics challenges and attack vectors, and models the framework on an integrated cloud-IoT platform. Finally, it provides the architecture and design stack, such that forensics-enabled IoT can be developed.

Chapter 7 : Conclusion and Future Work

Chapter 7 summarizes this research, lessons learnt from each project and discusses possible areas for future research.

Appendices

The results of the various studies and the effectiveness of the solutions are illustrated in the thesis Appendix, which is summarized below.

Appendix A – presents the forensics compliance levels of AWS, Azure, and GCP, against use case scenario events described in Chapter 5.

Appendix B – presents the log snapshots with forensics artifacts and identifies 5W1H evidence elements in the log, produced as a result of the case studies explained in Chapter 5.

Appendix C – presents cloud forensics logging results produced as a result of the execution of the use case scenario explained in Chapter 4, i.e., the framework's output.

Appendix D lists the copyright information to reuse the published works.

Chapter 2

Background

As already noted in Chapter 1, digital forensics applied in cloud computing is known as *cloud forensics*. Cloud forensics intersects many fields and subject domains. This chapter covers cloud computing, forensics, digital forensics, cloud forensics, and all the recent related works in the cloud forensics domain. For that, first, we start with a primer on cloud computing in Section 2.1. Then it proceeds with the discussion on forensics, followed by digital forensics and cloud forensics in Section 2.2. Cloud forensics details and related works are described in the subsequent Sections. Since part of our research work also includes Internet of Things (IoT) forensics, Section 2.3 describes an IoT overview and an introduction on IoT forensics. We conclude this chapter with a summary in Section 2.4.

2.1 Cloud Computing

Cloud computing is a broad, generic term with many meanings and definitions. The NIST defines cloud computing as "a model for enabling ubiquitous, convenient, on demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service

provider interaction." [38].

In simple terms, the cloud computing is a service delivery model, in which IT services are offered as a service to consumers and billed as per usage. The cloud computing architecture has the following core attributes [39]:

- **Elasticity:** the ability to scale up or down cloud computing resources dynamically as required.
- **Connectivity:** the ability to connect and access cloud computing services anytime from anywhere, typically using a thin client, such as a web browser.
- **Multi-tenancy:** the ability to host multiple tenants on the same physical resources, by sharing physical storage, memory, and networks.
- **Visibility:** the ability for consumers to have full visibility and control of their cloud deployment parameters, usage and cost.
- **Measured service:** the ability to meter the services and pay as per usage.

2.1.1 Cloud Computing Deployment Models

Cloud computing comes in several deployment and service delivery models. The deployment model includes [39]:

- **Public cloud:** A computing environment made available to the public over the internet. The public cloud is owned and operated by an external provider that sells computing resources and services.
- **Private cloud:** A computing environment, exclusively owned and operated by an organization or a third party. By virtue, the private cloud provides greater control of all computational resources and services and is intended for a single tenant.

- **Community cloud:** A computing environment is similar to a private cloud, but more than one organization shares the computational resources with equivalent privacy, security, and regulatory rules and requirements.

- **Hybrid cloud:** A computing environment comprising of two or more clouds that are bound together by standardized or proprietary technology, enabling interoperability.

2.1.2 Cloud Computing Service Models

There are three well-known cloud service models [38, 39]:

- **Software-as-a-Service (SaaS):**A model of software deployment whereby one or more applications and the computational resources to run them are provided for use on demand as a turnkey service, accessed using a thin client. This model aims to reduce the total cost of hardware and software development, maintenance, and operations. In this model, the control of the applications and the underlying infrastructure lie with CSPs. Consumers have minimal privileges, such as managing application settings and their data.

- **Platform-as-a-Service (PaaS):** A model of software deployment whereby the computing platform is provided as an on-demand service upon which applications can be developed and deployed. Its main purpose is to reduce the cost and complexity of buying, deploying, licensing and managing the underlying hardware and software components of the platform, such as database, operating system and development tools.

- **Infrastructure-as-a-Service (IaaS):** A model of software deployment whereby the basic computing infrastructure of servers, software, and network equipment is provided as an on-demand service upon which a platform to develop and execute applications can be established. Cloud Service Users

(CSUs) of IaaS avoid purchasing, housing, and managing basic hardware and infrastructure software components; they instead obtain those resources as virtualized objects controllable via a service interface.

This thesis addresses the public cloud deployment model and considers all service models.

2.1.3 Cloud Computing Architecture

The architecture of a cloud computing is unique, regardless of its deployment and service delivery models. Figure 2.1 depicts a typical high level cloud architecture stack [37].

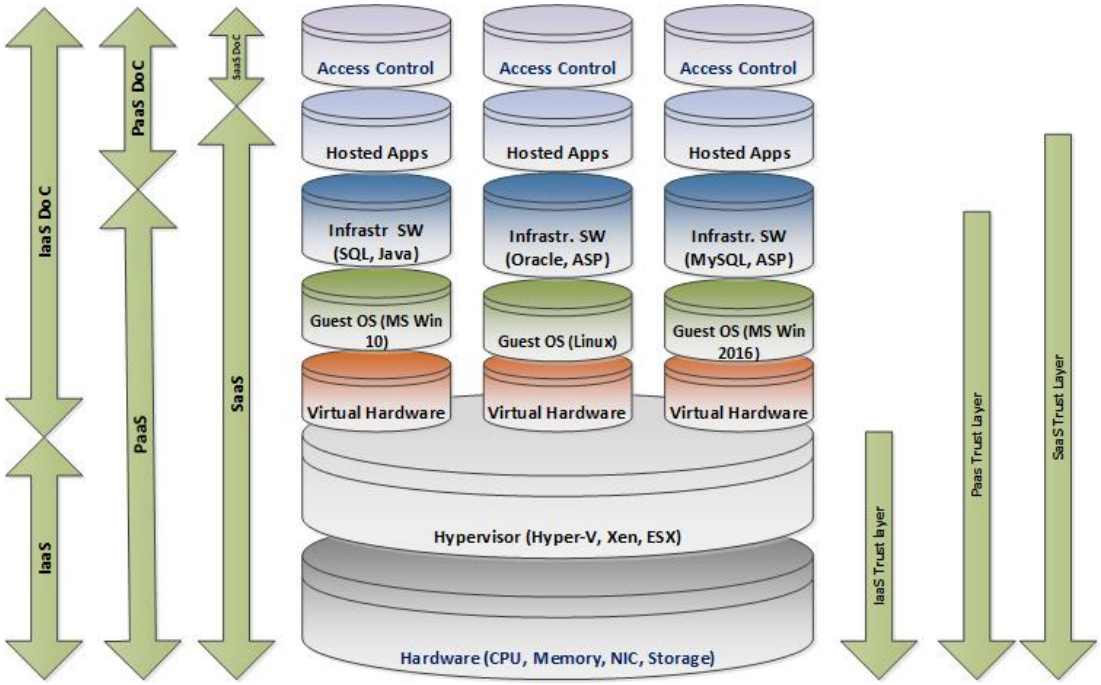


Figure 2.1: Cloud architecture stack, degree of control (DoC) and trust layer

As shown in Figure 2.1, the architecture comprises of the following six layers [37, 38]:

- **Hardware:** The lowest layer of the cloud stack is the physical hardware, consisting of the Central Processing Unit (CPU), memory, networking components, and storage.

- **Hypervisor:** Software that creates and runs Virtual Machines (VMs) and runs directly on the computing hardware. A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its physical resources. The hypervisor makes the VMs independent of the host hardware, providing flexibility in IT resource mobility, which means that the VMs can easily be moved between different servers [40].

- **Virtual Machine (VM):** A compute resource (typically called an image) with custom configurable CPU, storage, memory and networking resources, that uses software instead of physical hardware to run programs and deploy applications. Each VM can have its own operating systems and functions independently from other VMs. Multiple VMs can run on the same host. The virtual hardware and guest OS, as shown in Figure 2.1 together forms a VM.

- **Infrastructure Software:** Databases (e.g., MS SQL, Oracle), and development tools (e.g., .NET, ASP, Java), are deployed on the VMs, which are used to develop, deploy and run custom specific applications.

- **Hosted Apps:** The applications deployed on the VM instances providing specific services to the CSUs.

- **Access Control:** Software modules that authenticate the CSUs' identity when requesting to access cloud resources and authorize the access by enforcing the access control policies.

Figure 2.1 also shows degree of control (DoC) measure for each cloud service model across the six layers. The DoC measure shows each CSU's level of control on its own data asset. The figure shows that CSUs have a higher DoC level in IaaS than in PaaS model; SaaS has the lowest DoC. The trust layer for each service model, shown in the figure, defines a CSU's level of confidence in using the cloud and entrusting its CSPs as the custodian of its data assets [41]. The figure shows that a higher level of DoC means more trust.

2.2 Forensics, Digital Forensics and Cloud Forensics

Forensics describes the scientific methods used to investigate crime following a well defined process. Forensics includes collecting, examining, analysing physical evidence and presenting the results of the findings in a court of law. These activities are carried out as a post-incident or crime to find the culprit responsible for the crime.

Digital forensics is a branch of forensics science encompassing the recovery and investigation of material found in digital device, often conducted as a response to digital crime. Any form of digital data with any forensics value is also referred to as artifacts. The first Digital Forensics Research Workshop held in New York in 2001 [10] provided the following working definition of digital forensics: “The use of scientifically derived and proven methods toward the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources to facilitate or further the reconstruction of events found to be criminal, or helping to anticipate unauthorized actions shown to be disruptive to planned operations.” The NIST provided another definition for digital forensics in their special publication [42]: “the application of science to the identification, collection, examination, and analysis of data while preserving the integrity of the information and maintaining a strict chain of custody for the data."

Digital forensics comprises of three main areas viz., (i) client forensics (ii) server forensics, and (iii) network forensics. Client forensics deals with the collection, preservation and acquisition of evidence artifacts on the client side. On the other hand, server forensics deals with the same on the end point servers and network forensics deals with the forensics on the network logs and artifacts available on the network devices to further establish that the communication has indeed happened between the client and server.

Cloud forensics can be defined as the application of digital forensics in cloud computing. It is a cross discipline area. The recently published NISTIR 8006 report [36] defines that "cloud computing forensic science is the application of scientific principles, technological 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". More importantly, in non cloud-based (or *traditional*) digital forensics, all of digital devices systems involved in the crime can be easily acquired by the investigators, which is no longer applicable in cloud forensics. This point has been expanded later in subsequent paragraphs, in particular Section 2.2.4. In this work we consider cloud forensics as a sub-domain of digital forensics. Section 2.2.1 further elaborates on cloud forensics.

2.2.1 Cloud Forensics

Security incidents are generally the trigger for digital forensics investigations and therefore the forensics process is often treated as a post incident activity. It follows through pre-defined and well crafted procedure. Cloud forensics also comprises of three main areas similar to digital forensics, i.e., client forensics, server forensics, and network forensics [4]. The challenge is that, in a cloud environment, servers are VM instances in the cloud stack. Further, network forensics also includes internal network protocols and communications between the VMs within cloud infrastructure.

2.2.1.1 Client forensics

Digital crimes on the cloud are initiated and often carried out from the client side; their artifacts, however, are left on both the client and server sides. Client side evidence identification and collection is a vital part of the forensics process [43]. In the traditional client side forensics, the client system will hold the client side evidence artifacts, primarily in the application logs, operating system event

history logs, configuration files and remote systems communication logs etc. On the other hand, in the cloud environment, the services are accessed using thin client, usually a web browser. Therefore, the evidence data, such as browser history logs, registry content, access logs, chat logs, session data and persistent cookies, can be found on the web browser [44]. It is critical that the data should be collected as early as possible in its *sterile* state, i.e., the data is in its original state. The reason is because there is a potential risk that the data could be erased either purposefully by the crime actor, or inadvertently by the system due to system configuration. For example, the web browser history and session logs can be configured to be overwritten or erased after a specified time or when the file size reaches the configured maximum limit. The trash files could also be erased for resource optimization. However, it is to be noted that both web browser history sessions and trash files are critical forensics artifacts.

The proliferation of client-side endpoints, especially to the mobile endpoints, makes the forensics data identification and collection even more challenging [4]. Devices at endpoints can be remotely located on mobile or IoT devices that send data to cloud infrastructure. It is critical for client-side forensics to identify those end points and collect timely data.

2.2.1.2 Server forensics

Digital traces of user actions and events that are created and available on the servers form a critical part of forensics artifacts. The artifacts include system logs, application logs, user authentication, access information, database logs etc. The physical inaccessibility, multi-tenancy, and unknown location of the data in the cloud environment make it much harder to conduct the evidence identification, separation, and collection. In a highly decentralized and virtualized cloud environment, it is quite common that the data may be located across multiple data centers situated in different geographic locations [45]. The traditional approach to seizing the system is no more practical either, even if the location is known, as

it could bring down the whole data center, affecting other CSUs. Consequently, making the evidence data acquisition from multi-tenant cloud platforms a major challenge for cloud forensics. [44–48].

Loss of governance is another major issue in cloud forensics [49]. The CSUs are entrusting the governance to the CSPs. This was also flagged by the European Network and Information Security Agency (ENISA)'s cloud computing risk assessment report, which includes the loss of governance as one of the top risks of cloud computing, especially in IaaS platform [50]. Loss of governance inadvertently leads to loss of control of information assets of the CSU's own data. The loss of control depends on the cloud model as outlined in Figure 2.1. In IaaS model, CSUs have more control and relatively unfettered access to the system logs and data. On the other hand, in PaaS model, CSUs' access is limited to the application logs and what pre-defined API provides, while in SaaS model CSUs have either little or no access to such data. As the CSUs increasingly rely on the CSPs to provide the functionality and services, they correspondingly give the CSPs more control of their data assets. Thus, resulting in more dependence on the CSPs for any future forensics needs [45].

Movement of VM instances within the cloud environment is another significant forensics issue. The VM instances are subject to movement across physical hardware, situated within a data center, or to centers, located elsewhere, including across jurisdictions. VM movements, done by the CSPs, are completely outside the control of the CSUs. Consequently, finding the evidence data location is almost impossible, adding additional challenges to the cloud server side forensics.

2.2.1.3 Network forensics

Traditional network forensics deals with the analysis of network traffic and logs for tracing events that have occurred in the past. Network forensics is theoretically also possible in the cloud. VMs instances within the cloud are configured with

networking protocols to use. The different protocol layers can provide several sets of information on communication between VM instances within the cloud and instances elsewhere. The CSPs ordinarily do not provide the network traces or communication logs, although such logs are a critical element of forensics data [46]. For example, if someone used an IaaS instance to distribute malware, routing information and network log are crucial parts of forensics data collection, but they are difficult to obtain. This becomes more challenging for PaaS and SaaS cloud models, and the acquisition of the data depends heavily on the support investigators receive from the CSPs.

2.2.2 Cloud Forensics Dimensions

Ruan *et al.* [4] described that cloud forensics is multi-dimensional, viz., Technical, Organisational and Legal, defined as follows;

- **Technical dimension:** "encompasses the procedures and tools that are needed to perform the forensics process in cloud computing environments. These include evidence identification, segregation, collection, live forensics, virtualized environments and proactive measures".

- **Organizational dimension:** "encompasses the policies, governance and service level agreements (SLAs) that facilitate communication and collaboration in forensics activities".

- **Legal dimension:** "encompasses the regulations and international agreements to ensure that forensics activities do not breach laws and regulations in the jurisdictions where the data resides, including privacy and ethical factors. Legal dimension also includes co-operation between nations to facilitate cross-border forensics investigations".

All of these dimensions contribute to the forensics readiness of the cloud environment. Alenezi *et al.* [51] further provided explanations to these dimensions,

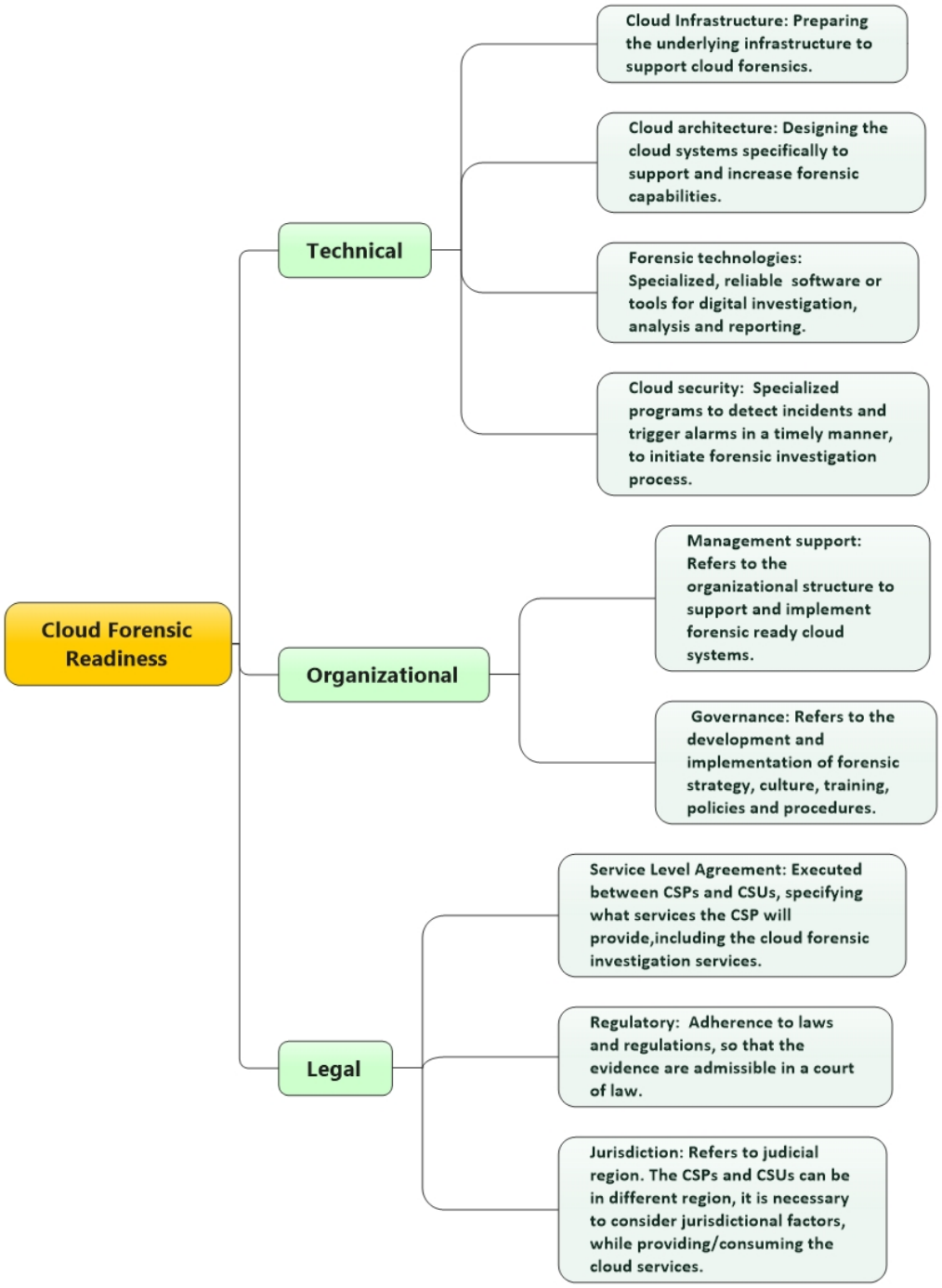


Figure 2.2: Cloud forensics dimensions.

which is described in Figure 2.2. This thesis addresses only the technical dimension.

2.2.3 Post-incident forensics vs. Pre-incident forensics

In general, security incidents are the trigger for a forensics investigation, hence the base of post-incident forensics. Post-incident forensics is an evidence-based investigation initiated after the incident. In contrast, pre-incident forensics is a continuous process and is referred to also as 'forensics readiness' or 'forensics-by-design' of the cloud infrastructure [49].

Post-incident forensics has been extensively studied. Many frameworks and methods have been proposed by researchers, although a number of challenges remain [52–62]. In general, post-incident forensics solutions can be grouped into two categories viz., (i) Log based solutions and (ii) Agent based solutions. Though the two solutions are relatively similar, Log based solution focuses more on logging as a continuous activity that records every event in the system in support of forensics (aka 'forensics logging'). On the other hand, Agent based solution has an additional layer of an Agent application running elsewhere (in the cloud or stand alone) which harvests the digital events and streams to a repository for further processing. Table 2.1 summarizes related works of Post-incident forensics.

Table 2.1: Research works in post-incident cloud forensics

Publications	**Contribution**
	Log based solutions
Marty [52]	Proposed a cloud application logging framework and guidelines describing what, when, and how to log the data needed for forensics investigation for SaaS cloud service model.
Patrascu *et al.* [53]	Presented a novel way of monitoring activity in cloud environments and data centers using a secure cloud forensics framework. The proposed architecture can be applied on top of cloud computing deployments. The authors proved the architecture by integrating it with a previously developed cloud computing system.

Continued on next page

Publications	Contribution
Zawoad *et al.* [63]	Proposed a Secure Logging as a Service Scheme (SecLaaS) that collects and preserves the activity logs in a cloud infrastructure. Confidentiality of the logs has been ensured thru RESTful APIs only access and integrity by hash-chain scheme and proofs of past logs published periodically by the cloud providers.
Rane *et al.* [55]	Proposed Blockchain assisted secure logging-as-a-service (Block-SLaaS) in storing and processing logs securely. The method leveraged the immutable property of Blockchain to ensure the confidentiality and integrity of the cloud logs.
Raju *et al.* [56]	Proposed a framework for event re-construction of cloud service logs, using two variant forms of log aggregation Leader-Follower algorithm.
Khan MN *et al.* [57]	Proposed a log aggregation model by combining logs from the client-side and CSP-side logs to a central log repository and then applying indexing normalization, correlation, and timeline sequencing of evidence data.
Awuson-David *et al.* [58]	Designed and implemented a Blockchain Cloud forensics Logging (BCFL) framework, based on Blockchain distributed ledger technology. The framework uses Blockchain smart contract to maintain the transaction log immutability and establish the chain of custody of evidence assets.
	Agent based solutions
Dykstra *et al.* [59]	Developed FROST, a forensics tool kit, that supports the Infrastructure-as-a-Service (IaaS) cloud service platform and provides trustworthy forensics acquisition of virtual disks, API logs, and guest firewall logs.
Alqahtany *et al.* [60]	Presented a forensics acquisition and analysis system (FAAS) for IaaS model. The FAAS seeks to provide a richer and more complete set of admissible evidences than what current CSPs provide, with no requirement for CSP involvement or modification to the CSP's underlying architecture.

Continued on next page

Publications	Contribution
Alex *et al.* [61]	Presented a cloud forensics framework for monitoring all activity that gathers forensics and log data from the cloud environments. The solution addresses the data collection issues by introducing a centralized forensics server and a forensics layer called forensics monitoring plane (FMP) outside the cloud Infrastructure, so that the investigators need not have to depend on the CSPs for collecting data.
Kebande *et al.* [62]	Proposed an agent based solution for evidence harvesting from the cloud. The agent application runs on every VM instance, streams into a logging repository, hashes the data blocks, and preserves the evidence. The model was proved by creating a prototype on Linux VM in an ownCloud environment.

Pre-incident forensics involves the design of the cloud infrastructure for future forensics support [49]. Digital forensics readiness consists of pro-active measures whereby the cloud environment is being continuously prepared by building and maintaining the environment's capabilities to support forensics. So that, when needed, the organization can comply with the investigation with sufficient preparedness [64, 65]. One of the primary objectives of forensics readiness is to maximize the environment's capability of collecting digital forensics data while minimizing the cost of the investigation during an incident response [64]. This topic also has been widely studied. A summary of related works is listed in Table 2.2. This thesis addresses both Post-incident and Pre-incident forensics.

Table 2.2: Research works in pre-incident cloud forensics

Publications	Contribution
Continuous forensics solutions	
De Marco *et al.* [66]	Laid out the foundations of cloud forensics readiness and proposed a Cloud forensics Readiness Reference Architecture.

Continued on next page

Publications	Contribution
Ab Rahman *et al.* [67]	Presented conceptual forensics-by-design framework for a cyber-physical cloud system (CPCS), which ensures that a CPCS is designed to facilitate forensics investigations. The framework consists of many factors, such as risk management, forensics readiness, incident-handling, laws and regulations, and specific requirements.
Kebande *et al.* [68]	Presented a functional architecture for cloud forensics readiness large scale digital evidence analysis and proved the architecture using MapReduce.
Kebande *et al.* [64]	Presented a novel idea of cloud forensics readiness model using 'botnet as a service'. The model implements a botnet with non-malicious code. The botnet infects instances of VMs within the cloud, but with good intention. The botnet then harvest data which can be used for investigation.
Kebande *et al.* [65]	Designed and implemented a technique for performing Digital Forensics Readiness (DFR) in cloud computing environments. The design used an Agent-Based Solution (ABS) in a cloud environment with capabilities of performing forensics logging for DFR purposes.
Zhengwei Qi *et al.* [69]	Developed a forensics hypervisor, namely ForenVisor, as a live forensics tool to perform continuous forensics on a cloud platform. The ForenVisor collects evidence directly from the cloud environment and protects the evidence data.
Trenwith *et al.* [70]	Proposed a remote central logging method as a cloud forensics readiness model, which shortens DFR process and does not require CSP assistance.
Sibiya *et al.* [71]	Presented a digital forensics service model, which can be provisioned as a service. CSPs can use the service to make their environment forensics ready.
Alenez *et al.* [51]	Investigated and presented the factors that influence the forensics readiness of an organization, looking from the angle of technical, legal, and organizational dimensions.

Continued on next page

Publications	Contribution
Fei Ye *et al.* [72]	Developed a new Cloud Forensics Tamper Proof Framework (TamForen) aimed at an untrusted cloud environment. The framework used a data structure based on the Compressed Counted Bloom Filter to generate, transmit and store evidence and provenance data. Additionally, the model introduced a new concept of injecting noise data to increase the difficulty of tampering.

2.2.4 Summary of Cloud Forensics

In general, the investigation of a cyber-crime involving the cloud as a subject, object, or environment is cloud forensics [3], and it is an emerging area that continues to develop. We discussed several cloud forensics process models, frameworks and solutions, addressing the challenges and issues of cloud forensics. Ruan *et al.* [4] provided one of the earlier pioneering work on cloud forensics. They defined the cloud forensics terminology, described the forensics dimensions (listed in Figure 2.2), challenges and opportunities. Subsequently, the authors expanded the work and presented the cloud forensics maturity model and the two inter-related parts i.e., the cloud forensics investigative architecture and the cloud forensics capability matrix [73]. Recently NIST presented a report, i.e., NISTIR 8006, to define, identify and list cloud forensics challenges [36]. Further, the digital forensics investigative guideline produced by the ACPO [35] have become a good standard for digital investigation in UK and other countries. The ACPO guideline describes the digital investigative process and provides guidance on how to conduct digital forensics investigations. The ACPO guideline also is a good source to draw digital forensics requirements, as it is a practitioners' guideline. This thesis uses both NISTIR 8006 publication and ACPO guideline, as a source to guide the research and further to validate our models and proposals. Below we are summarizing some key findings in the cloud forensics domain.

- **Evidence location:** Unknown physical location of forensics artifacts and duplicate copies of the data being spread across different virtual servers, possibly in different countries, in a cloud environment, not only renders traditional digital forensics techniques inapplicable, but also causes significant hurdles in the evidence identification, acquisition and preservation phases. Often Law Enforcement Agencies (LEA) and CSUs have to depend upon CSPs for full data recovery and that requires stronger SLAs.

- **Decentralized data:** The decentralized and ephemeral nature of cloud environment produce not only a technical challenge, but also a legal issue, requiring support from other countries, since it is possible that the victim, the perpetrator and the cloud platform are located in different jurisdictions. Though there exists legal framework for cooperation among some countries, authors identified a lack of international framework and agreements, suggesting urgent attention by lawmakers.

- **Dependency chain:** Often CSPs trade service among themselves, creating an array of dependencies and trust issue. Investigators have to follow each link in the chain in order to collect the evidence. Moving forward, there is a strong need for a forensically valid uniform log framework, including the ability to capture the logs that are segregated per user account level (to protect the privacy of co-tenants, in a multi-tenant platform), and the ability to track the movement of user files in intra-cloud. Such logging mechanism will greatly help the traceability and provide transparency.

- **Data integrity:** The CSUs are adopting encryption to ensure data confidentiality and integrity and satisfy the policy or regulatory requirements. Ensuring the integrity of the evidence data is critical to have trustable evidence. However, encryption causes the biggest challenge to forensics, especially in evidence identification, acquisition, and segregation.

- **Provenance:** Properly implemented secure provenance can play critical

part in cloud forensics, as it provides ownership, process history, and comprehensive security features, thereby enriching the trustability of evidence [74, 75].

- **Security vs. forensics:** It is our observation that major CSPs are all giving much more importance to security and having their service portfolio getting security accreditation and less (or nil) importance to forensics needs. For example AWS has a strong security compliance program and aligned with security practices and various security standards [76]. Similar compliance with digital forensics are yet to come. In addition, forensics requirements and privacy rules often contradict each other. In an attempt to give maximum respect to privacy laws, cloud providers make the cloud platform more secure, inadvertently making the forensics tasks much harder.

2.3 Internet of Things (IoT): Overview

Internet of Things (IoT) is a relatively new wave of technology that is increasingly becoming popular, serving many aspects and needs of human life. IoT digitizes physical assets and connects 'things' (aka 'smart devices') to 'things' and 'things' to the world, forming a network of billions of connected 'things'. These smart devices communicate with each other to share information, monitoring and controlling services such as home automation systems, security monitoring, health care, agriculture, transportation, and critical infrastructure monitoring and control. The recent advancement and breakthroughs in the development of wireless sensor networks have further accelerated the growth of IoT usage [77, 78]. They help build intelligent environments, intelligent buildings, and cities [79]. One example is the large-scale deployment of IoT technologies within a city to make city operations efficient while improving the quality of life for the inhabitants. Same time, mission-critical smart city data, captured by IoT and carried over IoT networks, must be secured to prevent cyber attacks. Otherwise, that might

cripple city functions, steal personal data and inflict catastrophic harm [80].

The IoT also refers to a broad spectrum of information sensing and control devices. This includes s radio frequency identification devices, global positioning systems, industrial control systems, and small devices planted inside the human body (e.g.: pacemakers), providing many life-supporting services. In general, IoT is usually driven by purpose-built programs which can sense, control, and alter the state of objects. For example, turning on/off a ventilator system in an underground train station, based on the air quality. Further, IoT has become a robust set of connected, intelligent devices in an interconnected world, enabling surgeons to perform remote operations, power companies to efficiently manage the grid, security companies to monitor security, and defense personnel to control national security. The capability of the IoT embedded technology to sense and interact with the external environment has created a multitude of use of the IoT systems, promoting a sharp increase in the deployment of these smart devices and will continue to increase rapidly.

However, IoT devices are very much prone to security vulnerabilities too. IoT manufacturers often provide greater importance and emphasis to the cost, size, and usability than to the security [23, 81, 82]. The vulnerabilities of the smart interconnected IoT devices can be exploited and subjected to cyber-attacks. Such attacks can significantly change the IoT's behavior, which can cause significant damage and life-threatening consequences. Therefore, the significance of IoT forensics and it is crucial that the capabilities and support for post-incident forensics are to be integrated into the design and development of IoT devices.

2.3.1 IoT Forensics

IoT forensics could be perceived as an extension of digital forensics to the IoT domain. The purpose of IoT forensics is similar to that of digital forensics, which is to identify, collect, preserve and present digital information in a legally valid and forensically sound manner. IoT forensics is young and unexplored branch

of digital forensics and relatively understudied. Ghosh *et al.* [81] coined the word IoT forensics as a budding field of forensics. Although, we observed that there is a concerted effort by the research community to bridge the gap. IoT forensics is interdisciplinary, as the evidence data for investigation may have to be collected from sensors, controllers, smart devices etc., connected to a crime scene. Since modern IoT deployments use cloud services, it naturally becomes essential to extend IoT forensics investigations to the cloud environment, too [81]. IoT forensics and cloud forensics intersect in some areas, but there are more complexities in IoT forensics. Mainly the additional complexities arise from the diverse architectural complexity, varying communication protocols, data formats, and more importantly, the physical location and mobility of the IoT [23, 25, 82, 83]. While issues like locating and identifying the evidence, enforcing chain of custody, and ensuring data integrity are related in the cloud and IoT forensics domain. Jurisdictional issues are also similar. However, the magnitude and context vary widely. Detailed analysis on IoT forensics are listed in Section 6.3. A snapshot preview of a cross-section of cloud and IoT forensics can be summarized as follows as background information.

- Uncertainty about the origin and source of evidence data, both in IoT and cloud environments, where and how the data is stored, locating and identifying the evidence;
- Proving the data provenance, ensuring integrity and authenticity of the data remains to be a challenge in both, but more challenging in IoT, mainly due to the capacity constraints;
- Difficulty in securing the chain of custody of evidence data, mainly due to data volatility for both IoT and cloud, but for IoT, it is even more complex due to various data transmission protocols and transit routes used;
- Unlike the cloud, there is no common or standard architecture or deployment model used for IoT;

- Digital forensics extractions or data collection techniques are less mature for IoT than cloud, and the traditional data extractions techniques are inapplicable for IoT;
- Diverse and proprietary data formats and limited storage space in IoT, produces additional constraints for timely data collection;
- Jurisdictional issues are common to both; while in cloud, jurisdictional problems are associated with data locations and movement of data across jurisdictions, but for IoT, the physical location and mobility of the IoT cause additional layers of jurisdictional complexity.

There is no unique methodology to investigate in a digital environment. The same applies to IoT. In the end, the choice of approach mainly depends on the assessment of the investigative body [25]. Also, IoT forensics is even more challenging due to the complexity, heterogeneity, and diversity of IoT devices and data transfer protocols [83, 84]. To address these issues, at least partially, researchers have proposed various IoT forensics frameworks and solutions. The following sections summarize the recent contributions in the field of IoT forensics.

2.3.2 IoT forensics frameworks

The work of Zawoad and Hasan [26] is one of the first efforts to define IoT forensics formally. They proposed a Forensics-aware IoT (FAIoT) model for supporting reliable forensics investigations in IoT infrastructure. FAIoT uses a centralized trusted evidence repository to ease the evidence collection and analysis process. The evidence repository applies a secure logging scheme [54] and ensures the reliability of the evidence. The two main components of FAIoT are (i) Secure Evidence Preservation Module and (ii) Secure Provenance Module. The Secure Evidence Preservation Module constantly monitors all registered IoT devices and stores evidence securely in the evidence repository. The Secure Provenance Module ensures proper chain of custody by preserving the access history of the evi-

dence [26]. The FAIoT model allows access to the evidence material via secure read-only APIs.

Kebande and Ray [27] presented a framework called Digital forensics Investigation Framework for IoT (DFIF-IoT). The framework complies with the international IT standards, security techniques, incident investigation principles and process (ISO/IEC 27043:2015). Kebande *et al.* [28] further expanded their work into the Integrated Digital Forensics Investigation Framework for IoT (IDFIF-IoT). The main contribution is the inclusion of digital forensics techniques that can analyse potential digital evidence from IoT-based systems that may be used to prove a fact.

Digital forensics in IoT space would definitely demand doing forensics with large amounts of devices and very large volume of data. If the foot print of the data are reduced to the forensics relevant subset of data and devices, it would then create a manageable scenario for forensics investigation in IoT. To this effect, Quick and Choo [29] proposed a novel IoT device forensics and data reduction method by selective imaging, automated with bulk data extraction techniques. The methodology demonstrates the capability to reduce the volume of the forensics data, while preserving its native source file format and its original metadata.

Internet of Vehicles (IoV) are an extension of IoT where 'things' are part of a moving vehicle and forms a complex and dynamic mobile network system that enables information sharing between vehicles, their surrounding sensors, and cloud environments [31]. Conventional IoT forensics techniques are insufficient in IoV environments due to the mobile, dispersed nature of the nodes. Rahman *et al.* [30] proposed a framework, called Mobility Forensics Framework, to address forensics in IoV environments. The framework is similar to the traditional digital forensics frameworks, which deals with the finding answers tot he 5W1H parameters. However, the model lacks methods to establish trust in the evidence. Hossain *et al.* [31] further improved the model, called Trustworthy forensics Investigation

Framework for the Internet of Vehicles (Trust-IoV). The Trust-IoV model maintains a secure provenance of the evidence to ensure the integrity of the stored evidence and allows investigators to verify the integrity during an investigation. The experimental results in [31] show that Trust-IoV can operate with minimal overhead while ensuring the trustworthiness of the evidence.

Many other models also have been proposed by researchers. One of them is a theoretical model that was presented by Harbawi *et al.* [85], namely The Last-on-Scene (LoS) Algorithm. In LoS, each device, representing the last node in the communication chain, must be the first to be investigated. Thus, the challenge in identifying the main source of digital evidence in IoT starts by identifying the 'thing' that produced the initial trace of the evidence. Then the process is carried on to the next element by moving through the network zones. Another framework, namely forensics State Acquisition for IoT (FSAIoT) [32], emphasizes the evidence acquisition process. The main component of the model is the centralized forensics State Acquisition Controller (FSAC). The FSAC is responsible for acquiring forensics evidence from the set IoT which the controller is responsible for and consumes cloud services for evidence preservation. The model was proved using Open Home Automation Hub (openHAB) [1], a vendor and technology agnostic open source home automation platform.

Designing and building IoT system with forensics readiness is one of the best options to address IoT forensics challenges. In this regard it is worthwhile to explain the framework proposed by Babun *et al.* [33], namely IoTDots. The novelty of the framework is the capability to analyze smart IoT apps source code, detect forensically relevant data points inside the source code and insert specific evidence collection and logging code at compile time. At run time, the apps then would log the forensics evidence which are sent to a remote IoTDots server. In a case of a forensics investigation, the framework app applies the data processing and machine learning techniques to extract valuable and usable

[1]https://www.openhab.org

forensics information from the IoTDots logs. The framework validation proved that over 96% accuracy of detecting user activities and behaviour with very low overhead to the IoT cloud resources [33].

An integrated IoT with Blockchain technology provides an innovative platform that could solve many of the IoT security and forensics issues [86]. The immutable, distributed nature of Blockchain technology may also suit the demands of the IoT Forensics. Digital evidence could be collected and stored in the ledger where the immutability of the Blockchain will ensure its validity and integrity [25]. Thus, making the chain of custody process more easier and valid. In addition, the forensics relevant information could be extracted by the investigation authorities from the nodes and the Blockchain could be used to timestamp to maintain the integrity of the digital evidence [25]. Subsequently, a wide variety of research contributions have been presented to cope with the forensics challenges by using Blockchain technology [87–90]. One recent work in this area is that of Hossain *et al.* [88], in which the authors proposed an IoT forensics investigation framework using decentralized and distributed Blockchain technology, namely 'A forensics Investigation Framework for IoT Using a Public Digital Ledger (FIF-IoT)'. The FIF-IoT model collects interactions among various IoT entities, e.g., cloud, users, and IoT devices, as evidence and stores them in the decentralized Blockchain network, thereby eliminating 'single point of failure'. FIF-IoT framework ensures integrity, confidentiality, anonymity, and non-repudiation of the evidence stored in the public digital ledger. Le *et al.* [87] proposed 'Blockchain based IoT forensics framework (BIFF)', to record the events of the entire life cycle of digital evidence in a transparent and traceable way to to record the chain of custody for digital evidence during a cyber attack. Further, Brotis *et al.* [89] used Blockchain technology to deal with collecting and preserving IoT digital evidence in the smart home domain. Their proposal is based on the advanced intrusion detection and distributed ledger technology solutions and the collected data are stored in the evidence database hosted by the internet service provider. The metadata are

published on a Blockchain and maintained by the service provider, and thus the proposal provides the means to law enforcement agencies to effectively trace back an attack to its source [89]. The model proposed by Rye *et al.* [90] aims to solve the heterogeneity and distribution characteristics in IoT environment. In this model, all communications of IoT devices are also stored in the Blockchain as transactions. Thus the model makes the integrity, non-repudiation and chain of custody factors much stronger.

2.3.3 IoT forensics landscape

Digital investigation in IoT is marred with numerous complexities, created by the unique nature of IoT and the challenges they pose. To mention a few, (i) lack of standardisation in IoT architecture, resulting in the proliferation of heterogeneous IoT devices, (ii) no standard communication protocols, resulting in the development of a multitude of IoT communication protocols, including some vendor specific proprietary protocols and (iii) IoT devices are usually very small with limited memory and storage space, but they are the source of big data [25, 29, 91]. In the IoT forensics space Hou *et al.* [91] sketched the IoT forensics from a three dimensional model. The three dimensions are (i) temporal dimension, (ii) spatial dimension, and (iii) technical dimension. The temporal dimension walks through the standard digital forensics process, i.e., Collection, Examination, Analysis and Reporting. The spatial dimension explores where to identify the evidence source, e.g., device, network or cloud. Finally. the technical dimension guides the exploration of tools and techniques, e.g., forensics readiness, evidence extraction, and live forensics techniques. The 3-D view of the IoT landscape provides a holistic overview of the digital forensics in IoT as shown in Figure 2.3 [91].

One key characteristic of IoT forensics is to comply with the laws and regulations applied to digital forensics investigations. However, the multi-jurisdictional nature of IoT environment makes it unclear regarding which law can be applied and how [92]. That leads to the argument that it is necessary to create an interna-

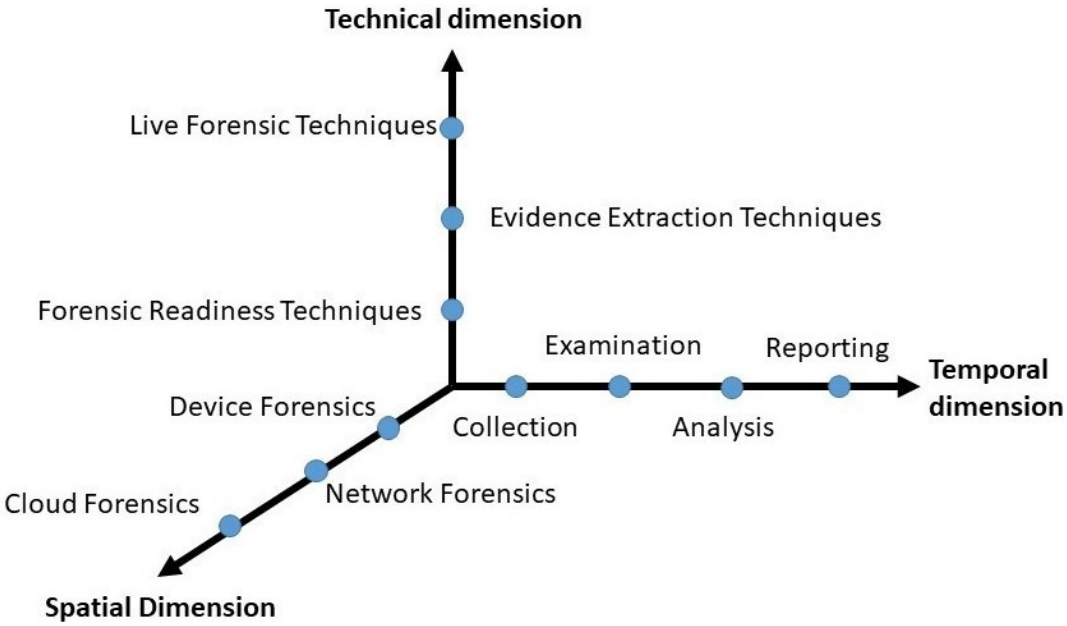


Figure 2.3: Holistic view of the IoT landscape.

tional commission, which updates the current legislation and defines an uniform governance structure for IoT Forensics [25].

2.4 Chapter Summary

Cloud computing has changed the way IT services are being delivered and consumed. There has been a tremendous growth of cloud adoption and that trend is expected to continue. Correspondingly there is growing concern by the consumers about the security and privacy of data assets stored in the cloud. At the same time there is also a significant concerns on the potential to use the cloud as a platform to conduct cyber crimes. With immense computing power and storage offered by cloud environments, major attacks can be conducted in shorter time periods and at low cost. The criminals can then terminate their account completely and disappear into ether without leaving any traces. The deleted account holder's storage and memory are quickly re-allocated to other tenants, thereby exacerbating the difficulties in finding the evidence and associating it with the actor.

This chapter begins with cloud computing characteristics and service models. Further, it describes digital forensics in detail, particularly cloud forensics. It also summarizes related and recent cloud forensics research contributions and summarizes the findings. Given the ever-growing ubiquity of digital technology at a remarkable rate, it is natural to expect that digital crime, exploiting the pitfalls in the developing technology, would also increase, posing new challenges to digital investigations. Therefore, digital forensics research will continue to develop to address the emerging issues [93].

The large scale usage of IoT in every day life and the potential that vulnerabilities in IoT can be exploited and can create a major cyber security menace, increased the interest in IoT forensics studies. However, the fundamental characteristics of IoT poses real challenges to IoT forensics. Regardless, researchers have made many attempts to address and solve IoT forensics issues, and it continues to develop. This chapter also summarizes IoT forensics landscape and forensics frameworks.

Chapter 3

Cloud Forensics: Process, Challenges and Solutions

Recall that in Chapter 2 we described cloud forensics in detail and highlighted that cloud forensics poses its own unique set of challenges. Moreover, typical digital forensics processes and methods cannot be directly applied to cloud forensics either. This chapter presents a comprehensive analysis of the cloud forensics process and challenges and recommends solutions to address the challenges as we walk through the different cloud forensics phases. In detail, the contributions of this chapter are as follows:

1. It systematically presents the cloud forensics process and lists the challenges per different phases or stages of the process. The systematic approach would enable forensics practitioners and information security professionals to quickly comprehend and understand its problem as they go through the various stages of the forensics process;

2. It provides a comprehensive analysis of the existing cloud forensics solutions. Based on the complete analysis, we recommend suitable solutions to address specific cloud forensics challenges and provide an evaluation of the suggested solutions;

3. It identifies the area where the solutions are still immature or not yet fully developed and lists the pros and cons of the solutions, where applicable.

The organization of this chapter is as follows. Section 3.1 outlines the motivation to conduct this research study and Section 3.2 describes the methodology applied to conduct the study. The cloud forensics process is described in Section 3.3. Then it identifies, analyses and lists the challenges associated with each forensics process phases and recommends the solutions to overcome the challenge with evidence and reasoning in Section 3.4. This chapter also conducts a study on the recent works in the cloud forensics domain since our initial study in 2015 and establishes the research impact and relevance of this work in Section 3.5. The chapter contributions are summarized in Section 3.6. Parts of this chapter, specifically Sections 3.3 and 3.4, has been published in the author's journal article [37].

3.1 Motivation

As previously noted in Chapter 2, cloud forensics is a multi-disciplinary field. The architectural complexity of cloud computing makes cloud forensics more complex too. During the period of this study, we observed that cloud forensics is in its infancy state [73, 94]. However, as evidenced by the number of published research work in cloud forensics, we found that there has been active research happening. Many researchers started highlighting various cloud forensics challenges, and some proposing solutions addressing the challenges. We also observed a lack of compatibility and reliability of the tools, technology, or solutions proposed by the cloud forensics research community. The solutions also lacked a consistent approach. The challenges and solutions mainly were classified along the lines of cloud deployment models [13, 94]. However, it is to be noted that the forensics challenges nor the solutions pertain only to the cloud deployment model. Also, the suggested solutions stack lacked comprehensive coverage too. From foren-

sics practitioners' or investigators' point of view, they want to know what cloud forensics challenges are, which solutions are most appropriate to overcome the challenges, and how they can be applied during an investigative process. From forensics researchers' point of view, they will be more interested in knowing the pros and cons of different solutions to help them develop better solutions. We found that the magnitude of cloud forensics has not been addressed from that angle, which is the focus of this study.

Therefore, understanding the cloud forensics challenges and how they can be tackled would be better if presented along with the cloud forensics process phase. The cloud forensics process phases are described in Section 3.3. We believe that such an approach would help the audience, more specifically the forensics practitioners, to know the challenges to expect and what possible solutions are available during an investigative process. This would also help organizations better prepare pro-actively for supporting investigations, knowing the issues in advance, and the research community to understand the current state and deficiency of the forensics solutions.

3.2 Methodology

This section describes the methodology applied to conduct this research study.

1. Conducted a thorough review of the related work, research outputs and the material available in this field. We used Google Scholar and existing databases (e.g: ACM digital library, Science Direct, IEEEXplore, Springerlink) for finding out the related materials. In the first phase of the search, we collected the existing research publications related to cloud computing, digital forensics, and cloud forensics. In the next search phase, we specifically looked into cloud forensics challenges and solutions. Our search process was iterative and narrowed down to individual journals (e.g., Digital investigation, Computers & Security, etc.) or conferences dedicated to

this field. In the third round, we also collected resources and materials from standards organizations like NIST, ISO, and professional organizations like Cloud Security Alliance (CSA). Finally, we also collected resources from public CSPs like AWS, Azure, specifically looking for the solutions they are offering.

2. We studied these publications and categorized them into the following:

   (a) Contributions related to digital forensics, regardless of the computing platform.

   (b) Contributions focusing on cloud forensics, including forensics technology, challenges, issues and solutions.

   (c) Contributions focusing on specific tool(s) or solutions addressing a forensic challenge (e.g: Resource tagging (to address evidence location identification), encryption (to address confidentiality and trust).

3. We systematically studied the cloud forensics challenges and grouped them per forensics phase.

4. Finally, we studied the solutions collected from the research resources addressing every cloud forensics challenge and assessed the solutions' suitability and applicability. Based on our assessment, we proposed a suitable solution(s) to address a specific challenge. We further described the solutions and highlighted the benefits or drawbacks.

3.3 Cloud Forensics Process

A cloud forensics investigation follows thru a set of pre-defined processes such that the outcome of a digital crime investigation can prevail in any court of law. An investigation must satisfy at least two required fundamental criteria: (i) immutability of the crime's evidence and (ii) the crime's investigation process.

Immutability refers to the integrity of evidentiary items and proving that they are accurate to the original [58, 86]. The investigation process refers to the systematic procedure followed in a digital investigation process. For example, an investigation process that follows ACPO guidelines [95] ensures its result is legally accepted in a country that complies to the guidelines. Researchers and forensics practitioners have proposed several digital forensics frameworks and explained the process. Some research work focused on refining previously published processes and frameworks and proposing new ones, resulting in a variety of digital forensics process models and terminology. A selected few digital forensics process models are:

1. Digital Investigative Process (DIP) model [34] comprising of (i) Identification (ii) Preservation (iii) Collection or Acquisition (iv) Examination & Analysis and (v) Presentation phases.

2. McKemmish model [96], a linear process comprising of (i) Identification (ii) Preservation (iii) Analysis and (iv) Presentation phases.

3. NIST forensics model [42] consisting of (i) Collection, (ii) Examination, (iii) Analysis and Reporting phases.

4. Integrated Digital forensics Process Model (IDFPM) [97] that consists of (i) Preparation, (ii) Incident, (iii) Incident response, (iv) Digital forensics investigation and (v) Presentation.

5. Digital Forensics Analysis Cycle Model (DFACM) [98] that consists of (i) Commence, (ii) Prepare and respond, (iii) Identify and collect, (iv) Preserve, (v) Analyze, (vi) Present, (vi) Feedback, and (vii) Complete or further task identified phases. This is a cyclic and iterative model.

6. Integrated Conceptual Digital forensics Framework (ICDFF) [99] for cloud computing that consists of (i) Evidence source identification and preservation, (ii) Collection, (iii) Examination and analysis, and (iv) Reporting and

presentation phases .

7. Consumer-oriented Cloud Forensic Process Model (CCFPM) [100] presents a process model that takes consumer perspective into account. The process model has three main components (i) a forensic readiness component, (ii) Live forensic component and a (iii) Postmortem component.

8. Standardised Digital Forensic Investigation Process Model (SDFIPM) [101]. This model argues that digital forensics practitioners have developed all the existing models based on their personal experience and therefore lack standardization. This model attempts to establish standardization within the field, resulting in a heavier process model consisting of many process steps, viz., examination, analysis, interpretation, event reconstruction, reporting, presentation, and investigation closure.

9. Common Investigation Process Model for Internet of Things Forensics [102]. This digital investigation process is primarily aimed at IoT forensics, considers the heterogeneity of IoT infrastructure and consists of four processes, viz., preparation process, collection process, analysis process and final report process.

The digital process investigation models listed from 7 to 9 are relatively new, and our studies could not find evidence of practical use or applications of the models yet. In a traditional server based computing environment, where physical locations of the systems are known, investigators can have complete control over forensics artifacts. However, the intrinsic nature and characteristics of the cloud ecosystem make the physical data location an unknown identity and produce additional challenges of mapping each traditional forensics framework to the cloud environment. For example, IDFPM framework recommends seizing the digital evidence during incident response, which is impossible in a cloud environment. Citing another example, consider the ICDFF model proposed by Martini and Choo in [99]. In this model, during the examination and analysis phase, i.e.,

step (iii), if more data or evidence is required, the process iterates back to the evidence source identification and preservation phase, i.e., step (i) and collection, i.e., step (ii). There is a high probability of evidence being erased or modified at any given time in a cloud environment since the cloud platforms are constantly subject to rapid changes. So more evidence may not be available by the time the process iterates back to step (i). This highlights the importance of preserving the evidence as soon as it is identified, using proper preservation techniques regardless of the evidence source. Martini and Choo [103] emphasized such an important step in their work on distributed file system forensics. The distributed file system forensics highlighted forensically sound methods, processes and clear guidance.

The DIP model [10] is applicable to any digital investigation, including forensics investigations in cloud environment. The model has been further expanded and applied in distributed file system forensics [103]. The DIP model is also very popular among forensics practitioners [104, 105]. This thesis uses the DIP model to describe the cloud forensics process, associated challenges, and solutions. The DIP model is illustrated in Figure 3.1.

Different phases of the cloud forensics process pose a different set of challenges. However, there are also common challenges applicable to more than one stage of the process, but the characteristics of the challenges are often unique to a particular stage. If so, the challenge is included in all applicable stages. The challenges span all three cloud forensics dimensions, viz., technical, organizational, and legal. However, in this thesis, more focus is given to challenges related to the technical dimension. Challenges related to other dimensions are discussed based on merit and importance in a given situation.

3.4 Cloud Forensics: Challenges and Solutions

It has already been mentioned that the traditional digital forensics process and techniques cannot be directly applied to cloud forensics. In particular, distributed processing, the multi-tenancy nature of cloud computing, and its highly virtual-

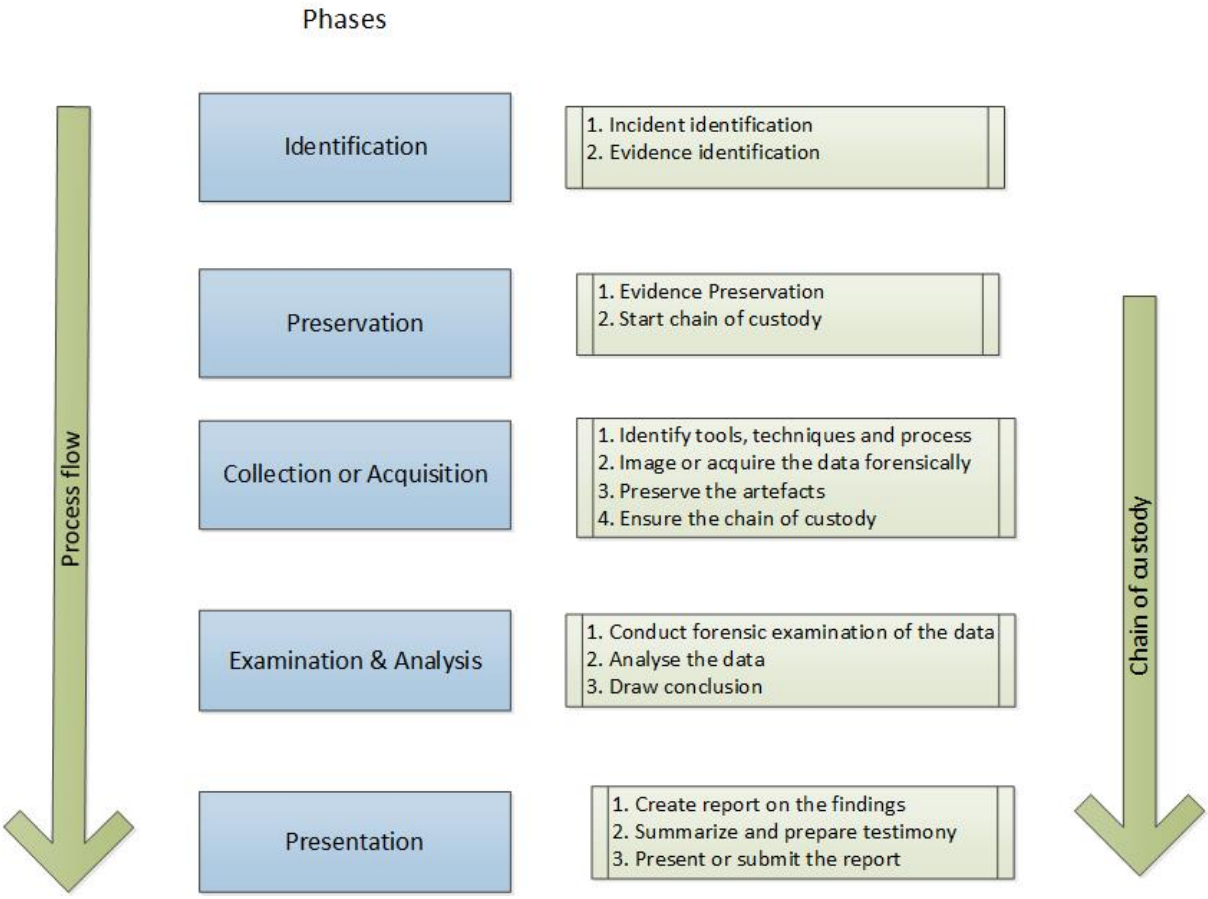


Figure 3.1: Digital investigative process

ized and dynamic environment make digital evidence identification, preservation and collection, needed for forensics a daunting and challenging task. This section analyzes cloud forensics challenges and recommends appropriate solutions to resolve the challenges per cloud forensics stages defined in the DIP model. We derived the solutions after studying the work and the contributions in the same space by previous researchers.

3.4.1 Identification

According to the DIP model (Figure 3.1) the cloud forensics process begins with the identification phase. ISO 27037 standard defines the identification process as a "process involving the search, recognition, and documentation of potential

digital evidence" [106]. The identification phase includes identifying systems, media, mobile devices, etc., likely to contain potential digital evidence. In reality, identification is a two-step process: (i) identification of the incident and (ii) identification of necessary evidence and its location to prove the incident. Step (i) requires identifying all machines and system files suspected of containing related evidence, and step (ii) requires identifying the evidence in the media or storage. Traces of evidence can be found in media such as cloud servers, network devices, and mobile devices [96, 107]. Proper evidence identification requires the knowledge of its present location, type, and format, and in the cloud, it is almost impossible to identify the physical location of data assets at a given time [99, 105].

The default settings for the public cloud deployment model are multiple jurisdictions and multi-tenancy in a highly decentralized data processing environment. CSPs often intentionally hide data location, so they have the freedom to move the data across VMs and data centers. This done mainly to facilitate business continuity, increase operational efficiency, and enhance data availability. The settings pose additional difficulties in data identification and subsequent collection [4, 45, 46]. System and application logs form a vital part of forensics investigation and finding out logs' location also poses an equal challenge [43]. Table 3.1 outlines the challenges in the data identification phase and their recommended solutions.

Table 3.1: Identification phase: challenges and recommended solutions

No.	Challenges	Recommended Solutions	Comments
1	Unknown physical location	Resource tagging [45]	Adversely affects CSPs ability to ensure flexibility, service availability and manageability.
		Robust SLA with CSPs in support of cloud forensics [46, 108, 109]	Most of the SLA guidelines are mainly focused on security requirements and less on forensics requirements.
		System level logs	System level logs can contain prime information regarding the access, creation and deletion of system level objects.
2	Decentralized data	Log framework [52, 110]	Logs including the hypervisor level logs would help the forensics process and time lining of events.
3	Data duplication	Resource tagging [45]	Can adversely affect the system performance.
4	Jurisdiction	SLA, specifying where the data can be stored or migrated [38, 108, 109]	Can adversely affect CSPs ability to ensure service availability flexibility and cost benefits to consumers.
		Reverse look up for networked devices to study the network topology [111]	This is a very time critical action due to dynamic nature of cloud.
5	Dependency Chain	None	Lack of solutions in the form of software tools, standards & processes.
6	Encryption	Key management system within cloud [111] and legal authority	Enables efficient investigations.
7	Dependence on CSP	SLA specifying the specific forensics services [108, 109, 112]	Good SLA ensures service availability and compliance [109].

Detailed explanations of the recommended solutions, including their relative merit are provided in the Sections 3.4.1.1 through to 3.4.1.7.

3.4.1.1 Unknown physical location

As already noted, the intrinsic nature of cloud computing abstracts the data locations. From an investigation angle, it means that cloud data can be stored out of the jurisdiction of the investigating authority. Or the data may be split across several storage devices within a cloud environment, with some part of the data remaining within the jurisdiction. Some other parts outside the jurisdiction [36, 113]. All of this produces challenges in identifying the evidence artifacts.

- **Resources tagging:** The CSUs can "tag" their resources to mark the location of their data assets. The CSPs can then use the tag information to determine whether the data assets can be migrated, and if so, provide the allowed regional boundary of migration [45]. It is common for the CSPs to move the VM instances and associated files between different physical machines and sometimes across various data centers located in other geographical locations. Resource tagging can be used in such cases to inform the CSPs "what can be" and "what cannot be" moved. Therefore, resource tagging can address the legal issue by dictating the resources that cannot be transferred to different jurisdictions. However, the solution may adversely affect the CSPs' ability to efficiently manage their resources and provide exceptional services, such as availability, service continuity, and acceptable performance [45, 46].

- **Robust SLA:** In general, CSPs do not provide an option to choose the location where the user data will be stored. Having a solid SLA stipulating the data location is an option to address this challenge. Some CSPs , e.g., AWS, provide an option to choose a geographic location from a list of available regions around the globe to host the VM instance and data. AWS offers public cloud services in different areas around the world [76]. This

scheme partially solves the jurisdictional or data location issue for AWS customers. However, such an option is not a standard feature.

Jansen *et al.* [38] provided guidelines on incorporating data location into SLA. Alhamad *et al.* [108] proposed a conceptual SLA framework for cloud computing. Other researchers have suggested the importance of having a robust SLA with CSPs, which can be enforced [4, 38, 46, 112]. Ruan *et al.* [109] provided key terms and conditions covering forensics services in SLAs.

- **System level logs:** System logs provide detailed access reports on data assets, including privileged user access, creation, deletion, and modification of system-level objects. For example, the logs produced by AWS CloudTrail logs are a prime piece of forensics information [114].

3.4.1.2 Decentralized data

As already noted, the decentralized nature of data processing is one of the default attributes of cloud computing. There is no central location for files, database artifacts, system artifacts, and logs where the investigators can identify, preserve and collect the artifacts. The CSPs seldom provide the details of how the logs are created and where they are stored. In addition, the CSPs use their own proprietary log formats, and there exists no uniform log format either [36]. Having a uniform and forensically sound practical log framework is one way to solve the issue. Many researchers have identified the importance of keeping end-to-end and comprehensive transaction logs [46, 52, 110, 115]. Marty [52] provided a business-oriented logging framework and guidelines suggesting 'what to log' and 'when to log' and proposed a proactive approach to application logging. However, there is no research so far regarding a "pre-defined forensically valid log structure and location" that can be easily located, retrieved, and verified for its integrity, using which the investigators can produce an end-to-end temporal analysis of events.

3.4.1.3 Data duplication

Duplicating data to multiple locations is another inherent feature of cloud computing. From a forensics perspective, this is a good feature because it will be tough to destroy all the evidence from the cloud [116]. However, data identification is equally complex because the data is spread out. Nevertheless, one can use the resource tagging mechanism [45], described in Section 3.4.1.1, to locate deleted files needed for forensics by following the files' logical chain.

3.4.1.4 Jurisdiction

Since the cloud, by definition, enables broad network access connectivity. Storing customer data outside of the customer's jurisdictional area is quite common in the cloud. CSPs need not have to inform the CSUs of the location details of their data. Depending upon the location, different laws would apply, which would significantly impact the forensics process. Although it is theoretically possible for networked devices to trace back or perform a reverse look-up to produce overall topology and thereby obtain essential information, the step is quite tricky and cumbersome due to the fast dynamic nature of the cloud systems. The topology information (such as the allocated IP address, storage space, etc.) is subjected to rapid changes, and therefore faster response is often required to obtain meaningful information [111]. In addition, the CSPs usually keep migrating the VM instances between different physical machines, spreading them across other jurisdictional locations [45] and eventually creating legal challenges. The possible solutions to address the jurisdictional issue are:

- **Specific SLA:** Create SLAs that clearly specify where the data can be stored, re-located or duplicated [109, 117].

- **Reverse Look up:** To find the location of networked devices, and conduct a reverse look up of network topology [111].

3.4.1.5 Dependency chain

It is prevalent for CSPs to trade services among them. For example, a CSP providing email as a SaaS service may depend upon a third party CSP offering PaaS to host the mail platform services, depending on another IaaS provider to store the mail content and log files. The correlation of activities across the CSPs is a significant challenge, creating a chain of dependencies. Moreover, different providers might be hosting their services in other locations. Lack of transparency is another issue associated with multiple levels of outsourcing. Investigators need to trace and follow each link in the chain to trace the link and lock the evidence for collection. However, there is no easy way to perform this process. To date, procedures, policies, and guidelines related to cross provider forensics examination are virtually nonexistent, exacerbated by a lack of interoperability framework among cloud providers [36].

3.4.1.6 Encryption

Encryption is becoming increasingly relevant for cloud computing. Most CSPs provide encryption as a feature in their security services portfolio. CSPs provide the service either by providing an API for encryption. At the same time, customers use their key management system and keys, or by applying encryption when the data is stored in the cloud and storing the encryption key, which is often linked with user access password [111]. Major CSPs like AWS provide end-to-end encryption of storage volumes. AWS provides server-side encryption where AWS manages the encryption process. Alternately, customers who prefer to manage their data encryption can use a client encryption library to encrypt data before uploading to AWS S3 storage [76]. This method offers 'zero knowledge privacy,' meaning that the CSP can never know the stored data content. Regardless of the technique used for encryption, encrypted data appears as a continuous byte stream, making evidence identification very difficult and creating challenging problems in later phases of the forensics investigation.

Cloud Security Alliance (CSA) research group suggested using proper key management infrastructure like public key infrastructure (PKI). So that the data assets can be decrypted without the need to share keys, same time preventing unapproved access to customer data [118]. However, no published guidelines or standards have been found mandating the process.

3.4.1.7 Dependence on CSP

Due to the intrinsic nature of the cloud environment and the lack of administrative control a CSU has over their data in the cloud, the CSUs and investigators have to depend upon the CSPs to identify, locate and lock forensics evidence. Therefore, the best solutions is incorporating the essential forensics services required from the CSPs in the SLA. The CSPs are becoming more aware of it, and some do offer such services.

Dependence on CSP will continue to be an issue until the providers start offering tools to collect forensics artifacts on demand using a provided portal or similar applications. For example, AWS provides memory dumps and means to ship the memory anywhere for a fee. In addition, the AWS CloudTrail logging application allows the logs to be retrieved using AWS portal or CloudTrail can be configured to send events to AWS CloudWatch logs. Finally, for a rapid response, CloudWatch events can be configured to send notifications or alarms as a trigger to kick start incidence response or forensics activity [76]. Such features ease the dependence on CSPs.

3.4.2 Preservation

Once the evidence has been identified, the next phase is evidence preservation. ISO 27037 defines preservation as the "process to maintain and safeguard the integrity and/or original condition of the potential digital evidence" [106]. The preservation phase deals with locking or freezing the evidence, making it ready for collection, and this phase encompasses all activities that protect the integrity

of the evidence throughout the process. In reality, evidence preservation is not a one-step process; the process continues until the evidence is presented in court. Therefore, the preservation phase deals with locking or freezing the evidence, making it ready for collection, and maintaining legally admissible evidence is a vital objective of this phase. As the cloud platform is very dynamic, this phase is very critical. Table 3.2 identifies the challenges in the preservation phase and their possible solutions.

Table 3.2: Preservation phase: challenges and recommended solutions

No.	Challenges	Recommended Solutions	Comments
1	Chain of custody	RSA signature [119]	Can be used to validate the chain of custody and data integrity.
		Blockchain forensics logging framework [58]	Uses data immutability feature of Blockchian technology.
2	Evidence segregation	Sandboxing [120, 121]	Running programs are separated by virtual enclaves.
3	Distributed storage	VM instance tagging [45]	The tagged VM instances can be used to identify the location.
4	Data volatility	Persistent storage [43, 46]	Providing persistent storage defeats the elastic nature of cloud computing.
5	Data integrity	Checksum, hashing (e.g., MD5,SHA1,SHA256)	Used to preserve and verify the integrity of data.

3.4.2.1 Chain of custody

Chain of custody is the chronological documentation of access and handling of evidentiary items to ensure the authenticity and integrity of the evidence. It is required to avoid allegations of evidence tampering or misconduct [36]. As a part of the chain of custody process, any access to the evidentiary items should be recorded in an access log [43, 107]. Maintaining a strict chain of custody log is

crucial for evidentiary items to be valid and admissible in the eye's of the law [35, 122].

Researchers and legal practitioners have highlighted the importance of maintaining a proper chain of custody log. For example, Principle 3 of the ACPO guideline states the necessity of keeping an audit trail of all processes [35]. Basic premises of digital evidence collection include collecting the data in a manner consistent with the law, verifying the data to ensure that the data collected is comprehensive, and maintaining a proper chain of custody of evidence data. Although there is no single way to enforce chain of custody in digital forensics, the use of techniques such as time stamping, hashing, and e-signatures are central to all methods [122, 123].

One way of establishing the chain of custody for digital evidence is by using the RSA signature. RSA signature is a widely used public-key crypto system to secure data transmission. Lin *et al.* [119] proposed a cloud aided RSA signature scheme to seal and store the digital evidence in the cloud. The proposed technique would greatly assist in securely collecting and storing evidence, especially from mobile or IoT devices with limited computational and storage power. The digital signature can also be used to enforce the data integrity and establish the chain of custody of the evidence post seizure. For example, an investigator can perform a checksum on the artifacts and digitally sign the checksum using their private key. Another recent work is that of Awuson *et al.* [58], in which they proposed a novel method for maintaining the integrity and chain of custody using the data immutability feature of Blockchain technology.

3.4.2.2 Evidence segregation

By default cloud computing, is a multi-tenant environment. The multi-tenant characteristics possess difficulties in isolating and preserving evidence without hindering other tenants from sharing the same resources. One solution to evidence segregation is by sandboxing each user instance [120, 121]. Sandboxing is a

mechanism by which the running programs are separated into virtual enclaves, and each of them uses its enclave such that no instance knows the existence of its neighbor. Neighbors behave as if they are on separate hosts. Capturing the entire sandbox instances provides the running state of CSUs virtual machine instances then, which can be loaded onto another VM instance for analysis [37].

However, sandboxing addresses the problem only partially. For example, sandboxing of VM instances does not capture events such as the creation and deletion of VM instances. The hypervisor event logs capture such system level info, and those event logs are not accessible from a standard user account, and in addition, such logs would potentially contain information about other tenants too.

3.4.2.3 Distributed storage

Due to the distributed and elastic nature of the cloud environment, it is often impossible to ascertain where the piece of data is stored, as the data could be distributed among many hosts in multiple data centers. Resource tagging of the virtual instances [45], described in Section 3.4.1.1, is a potential solution.

3.4.2.4 Data volatility

The highly volatile nature of data is a significant concern for evidence preservation and collection in a cloud environment. One of the solutions suggested by researchers is to use persistent storage. Having persistent storage and keeping the storage synchronized frequently between the VM instances and persistent storage can counter the data volatility issue [43, 46]. However, the data on the running system compromised by an adversary cannot be mitigated, although traces of such ill-action will be available on the persistent storage as evidence [114, 124]. Note that CSPs usually do not offer persistent storage as a generic service. Further, providing persistent storage works against the on-demand, low cost, and elastic nature of the cloud. While synchronizing the volatile data storage to non-cloud storage is theoretically possible, it is not practical to implement and could defeat

the whole purpose of adopting cloud systems. Therefore, it remains very critical to collect forensics data as soon as the incident happens [4, 49]. Data volatility issue is further explained in Section 3.4.3

3.4.2.5 Data integrity

Data Integrity ensures that the evidence accurately represents the data found in the computer system. Several aspects of the cloud environment affect the data integrity, but maintaining the integrity remains to be a crucial aspect of cloud forensics. The known method to preserve data integrity is using proven hash techniques such as MD5, SHA1, SHA256. More recently, other novel methods to address data integrity in the cloud have been suggested, such as tamper proof cloud forensics framework using Bloom Filter algorithm [72]. Blockchain forensics logging is another proposed method. In this method the data integrity has been proved by using the data immutability and smart contract features offered by the Blockchain distributed ledger technology [58].

3.4.3 Collection or Acquisition

In digital forensics, collection refers to the "process of gathering items that contain the potential digital evidence," and acquisition refers to the "process of creating a copy of the data within a defined set" [111]. However, evidence collection is complex due to the transient nature of the cloud environment and the inaccessibility to the operating system files and artifacts such as temporary internet files and registry entries. In addition, public and hybrid cloud systems might operate across jurisdictions, making it much more challenging to acquire artifacts. Finally, unless cloud computing applications provide a complete audit trail, it may not be easy to extract the evidence in an admissible manner, or there may be little evidence available to collect [125].

Legal collection refers to the seizure of physical evidence under the authority of a legal or court order. Due to the multi-tenancy and jurisdictional issue associated

with the cloud environment, the collection is not practical, so the acquisition is the recommended process. Note that collection requires CSP support, whereas acquisition can be performed remotely using valid methods and tools described later. The process of acquisition (i.e., making a legally valid copy of all forensics artifacts) should be done using a well-defined, well-tested, and repeatable process using trusted tools. Therefore, the acquisition is a more challenging process than collection. According to ISO 27037, as illustrated in Figure 3.2, collection and acquisition are two parallel processes [106, 111].

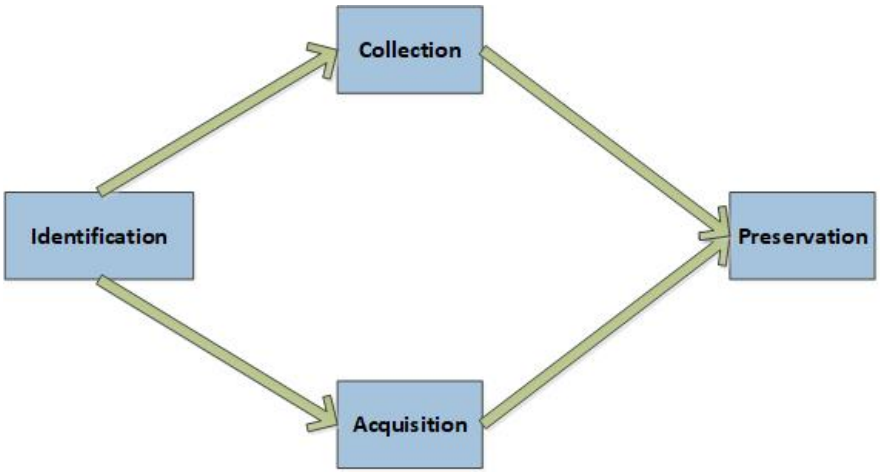


Figure 3.2: Evidence collection and acquisition process

Table 3.3 provides a summary of challenges in the Acquisition phase in cloud computing and recommended solutions.

Table 3.3: Acquisition phase: challenges and recommended solutions.

No.	Challenges	Recommended Solutions	Comments
1	Inaccessibility	Remote data acquisition [126]	By data imaging tools such as EnCase, FTK Imager, X-Ways, F-Response, Paladin etc., over a secure network.
		Management plane [94, 126]	Preferred option, removes the dependency on CSP.

Continued on next page

No.	Challenges	Recommended Solutions	Comments
		Live forensics [127]	Provides running system info, like process list, open ports etc., which are not available in offline forensics.
		Snapshot analysis [46]	Captures the whole system info at the instant of taking the snapshot.
2	Dependence on CSP	Management plane [94, 126]	Preferred option, but requires an additional level of 'trust' of the management plane.
		Stronger SLA [108, 112]	Preferred option for CSUs.
3	Ephemeral nature of data	Snapshot analysis [46]	Provides the point in time picture of the whole system.
4	Trust	Hardware trusted platform model (TPM)	Designed to work with single operating system, single machine. Fails to scale up to a virtualized cloud environment.
		Virtual TPMs [128]	TPM instances are obtained on demand. Solves scalability issue.
		Trusted virtual environment module [129]	Modular and extensible approach which supports persistent storage of keys.
		Trusted cloud computing platform [130].	Provides a closed box execution environment. Ensures confidentiality and integrity.
		Detective controls [41].	Complements formal preventive approach, and can address the risk that arise from within CSPs.

Continued on next page

No.	Challenges	Recommended Solutions	Comments
5	Multi-tenancy	Isolating cloud instance [120]. Sandboxing [120, 121]	Discussed various methods of isolating cloud instances. Most popular method of isolating the instance and widely supported.
6	Jurisdiction	SLA International cooperation in the form of agreements and treaties	Helps to address jurisdictional issue. Partially addressed in [108, 112]. E.g.: International treaties and agreements.
7	Deleted data	Frequent snap shots User signature [131]	Difficult to achieve and manage due to the sheer volume of snap shot images. Another piece of metadata is added with every data block which persists.
8	Lack of specialized tools	Cloud data imager [132]	The solutions remains to be commercialized.

3.4.3.1 Inaccessibility

As already mentioned, un-restricted access to cloud storage is not possible, something that is guaranteed in a traditional client-server environment. Note that the data in the cloud can be duplicated to multiple locations, resulting in decentralized artifacts. Some cloud providers, e.g., AWS, allow users to choose their geographical location while creating VM instances. However, physical acquisition is impossible due to multi-tenancy even if the location is known.

Various methods have been proposed for evidence acquisition from cloud, such as:

- **Remote data acquisition:** Refers to acquiring the evidence remotely over

a trusted and secure channel. Widely used forensics tools such as Guidance EnCase and Access Data FTK support remote data acquisition. Dykstra and Sherman [126] reported successful retrieval of volatile and nonvolatile data from AWS EC2 active instance platform using the tools, despite citing many layers of trust requirements. They validated the data integrity by computing and comparing the hashes of the images before and after downloading data.

- **Management plane:** Controlling the virtual assets in the cloud using a web interface is often referred to as the Management Plane. Using the interface, e.g., AWS Management Console, CSUs can conduct data acquisition of forensics artifacts, such as VM images, logs, disk images, user access information, etc. For example, one can use the AWS management console to extract CloudTrail logs without helps from the CSPs [114]. However, one more level of 'trust', i.e., trust in the management console application, is required. Despite the trust issue, researchers have recommended the use of a management plane for remote data acquisition, especially for IaaS model [94, 126]. Zafarullah *et al.* [133] showed that it is possible to collect necessary logs from cloud infrastructure using open source tools. One can acquire the disk image from a cloud server using cryptographic tunneling protocol, e.g., a virtual private network (VPN), to guarantee the confidentiality and integrity of the data, which can also help to address the chain of custody problem, described in Section 3.4.2.1.

- **Live forensics:** Forensics on a running system is referred to as live forensics, in which an investigator performs a forensics examination of a system in a running state. Such forensics comes with an added advantage as it can gather a wealth of information, such as process list, kernel modules, open network ports, volatile memory data, etc., from a running system and the information stored in persistent storage. Virtual Machine Introspection (VMI) is a live forensics technique where a CSU can interact with a running

system from another virtual machine. Hay and Nance [127] proposed a virtual introspection solution. They demonstrated their solution using Virtual Introspection for Xen (VIX) set of tools as a proof of concept. Their work has been further enhanced as an introspection library known as LabVMI (VMITools). However, the live forensics tools are yet to be incorporated and provided as a commercial service by the CSPs.

- **Snapshot analysis:** Snapshotting is the process of making a clone of a virtual image in a running state, including all the system memory, and saving the clone to persistent storage. Snapshot technology enables CSU to freeze a specific form of VM [46]. Major hypervisor vendors, e.g., Xen, VMWare, ESX, and Hyper-V, support snapshot features. The snapshot images provide valuable information regarding the running state of a system. They can be restored by loading them to a target VM for analysis. The snapshot feature can capture live VM instances and works across decentralized regions as long as the instances remain in the same logical infrastructure. Since the cloud environment is subject to rapid changes, a series of snapshot images can provide valuable information regarding changes to the data assets, which can be used to analyze and map onto a timeline of events. Therefore, for the cloud to be forensically ready, one should have an inbuilt feature to dump virtual machine snapshots automatically at configurable intervals since it is impossible to know when the security breach occurs. On the downside, this feature would require more storage space and can create significant performance issues. Nevertheless, the system can be configured to either purge or overwrite unwanted or older image dumps.

3.4.3.2 Dependence on CSP

Many researchers have cited the dependence on CSPs during the forensics investigation process [94, 134]. There is a move by some CSPs to provide management tools so that CSUs or investigators can collect the artifacts, by themselves. One

example is the use of *Management Plane* supported by *specific SLAs*, which is described in Section 3.4.3.1 on *Inaccessibility* and Section 3.4.1.4, on *Jurisdiction* respectively. Clearly drafted and executed SLA between the CSPs and CSUs is one of the critical elements to address the CSP dependence challenge. The SLA should specify: (i) specific forensics elements, e.g., monitoring, (ii) forensics support services, (iii) data ownership (specifically of the data under investigation), responsibility, (iv) the right to retain consumer data for investigative purpose even when the consumer decides to change the cloud provider, and (v) any applicable regulatory compliance requirements [108, 109].

3.4.3.3 Ephemeral nature

The transient nature of cloud data is another major issue facing data acquisition. For example, registry files, temporary files, internet access history logs, etc., are key forensics artifacts, and it is crucial to collect them sooner the incident happens, though it is difficult [111]. The periodic snapshotting of VM instances, described in Section 3.4.3.1 is a recommended solution.

3.4.3.4 Trust

In general, trust means an act of faith in confidence and reliance on something that is expected to behave or deliver something as promised [135]. In the cloud computing context, trust is the belief in the competence and expertise of the CSPs, and the underlying cloud architecture and systems, to reasonably care for the valuable information assets of the users. Trust and control go together, e.g., we trust a system less if it has poor control. Trust also is a function of ownership, e.g., you trust your data assets. Note that in a public cloud model, the CSPs are the custodian of CSUs data assets and CSUs have neither ownership nor control of the environment. When an enterprise adopts the cloud and consigns its data (belonging to the enterprise and its clients) to the cloud, it creates an array of complex trust relationships. First, the enterprise must trust the cloud provider.

Second, the enterprise should ascertain that their clients have enough reason to trust the same provider. In cloud forensics, the lack of transparency and trust results in untrustworthy evidence data [46, 135].

Researchers have highlighted the problem associated with trust in cloud forensics [43, 45, 46, 94, 126, 127, 136]. For the evidence to be valid, there is a need to establish trust in the layers of the cloud architecture. The layers of trust increase cumulatively as more services are subscribed from the CSP, i.e., the layers of trust are the highest for SaaS model and lowest for IaaS model. Figure 2.1 describes the trust layers.

Solving the trust issue remains a big challenge. Trust cannot be solved by using technology alone; rather, the solution should combine process, people, and technology. Following an established forensics process, e.g., ACPO guidelines, having experienced or certified people undertaking forensics collection and evaluation, and using approved forensics software or hardware tools would strengthen the trust in evidence.

Trust can also be treated as a function of security. Consumers will trust the more secure systems. One of the widely accepted approaches to solving security issues is the Trusted Platform Model (TPM), which is briefly described as follows.

- **Hardware TPMs:** Where a TPM chip is integrated into the motherboard, which can ensure data security. However, TPM chips are usually designed to work with a single OS on a single machine and typically would not scale, therefore, not practical for a cloud environment [137].

- **Virtual TPM (VTPM):** Virtual TPM is a software TPM providing TPM functionality. A TPM instance can be obtained from TPM cloud on demand. This technique is scalable and suitable for a cloud environment. [128].

- **Trusted virtual environment module (TVEM):** TVEM is a software appliance that helps to solve the trust issue by using a trust relationship

model. The TVEM architecture is modular and extensible, and also provides persistent storage for the encryption keys [129].

- **Trusted cloud computing platform (TCCP):** TCCP provides a closed box execution environment by extending the concept of a trusted platform to IaaS environment, guaranteeing confidentiality and integrity. Even privileged administrators cannot inspect or tamper with its content [130].

- **Detective controls:** These are controls based on policy and process. The benefit of this approach is that it is non-invasive and enforces the need for policy and governance structure to establish accountability and trust [41].

3.4.3.5 Multi-tenancy

One of the prime characteristics of cloud computing is that multiple VMs, hosting multiple tenants instances, can share the same physical hardware, which can spread across different data centers. This model is very different from a single owner system, where it is easy to seize the hardware. The multi-tenancy aspect adds to the complexity of forensics data collection in the cloud. Though the VMs operate in their sandboxes without knowing their neighbors' existence, doing a physical seizure is not at all practical as it can hold other tenant's VM instances and data. CSPs are bound to protect the customers' privacy and abide by the regulations. For example, a 2012 report by ENISA emphasized that multi-tenant outsourced services should protect the confidentiality of co-tenants [138]. Further, Ruan *et al.* [109] highlighted that SLAs must address privacy issues and noted that "the cloud provider to accurately and comprehensively filter forensics data sources that contain data belonging to multiple tenants and release only the data related to the specific tenant."

Researchers have recommended using the management plane for forensics data acquisition. Dykstra *et al.* [126] used forensics tools, such as EnCase and FTK, to successfully return the evidence from AWS EC2 cloud instance without violating the privacy of other tenants. Some CSPs,(e.g., AWS), offers a single tenant

option while creating an instance for an additional fee. Other suggested solutions to address multi-tenancy are:

- **Isolating cloud instance:** Delport *et al.* [120] introduced a new concept of isolating the cloud instance to facilitate the forensics investigation, using different methods such for instance relocation, address relocation, server farming etc. The isolated instances can prevent further contamination or tamper with possible evidence.

- **Sandboxing:** Creating sandbox image of a virtual machine instance is another way of isolating and protecting the evidence [120, 121]. Sandboxing is an easily executable option, and most of the vendors support sandboxing features. Sandboxed VM images can then be acquired using remote acquisition methods.

3.4.3.6 Jurisdiction

The CSPs often perform data mirroring to ensure high availability and business continuity. The mirrored databases can be in a different jurisdiction than the primary location, causing a lack of real-time information about the data location and introducing a high degree of difficulties for data acquisition. The jurisdictional issue associated with data location is one of the major concerns of customers. The cloud consumer should be aware that it could be challenging to conduct an investigation when the data does not reside in jurisdictions with proper regulations [116].

One possible solution to the problem is using specific SLA, described in Section 3.4.1.4, in which CSUs could specify where the data could be stored or relocated. The CSP should also accurately track the jurisdiction where the data resides during a given period [109]. If the data assets are spread across logical infrastructures around different locations, they can be acquired using proven techniques such as Remote Data Acquisition or commercially available tools. Further, if the data crosses geopolitical borders, stronger international cooperation and bilateral

agreements will also be required for evidence artifacts collection, establishing the chain of custody [7, 105].

3.4.3.7 Deleted data

From a forensics perspective, recovering deleted data and attributing the deleted data to the doer(s) are vital sources of evidence. Typically, the deleted data can be collected using data carving methods supported by forensics tools. However, in the case of the cloud, the volatility and elasticity of cloud environments make it much harder to collect the deleted data. Dykstra and Sherman [126] showed how to remotely acquire hardware and memory images from AWS cloud instances. They could also prove that it is possible to collect the deleted data (provided that the data volume is not overwritten) by the same tenant, excluding data or residual data from the previous tenant(s) who probably had the same hardware space earlier. However, this is a hypothetical situation, which proves that in theory, it is possible to recover deleted data from the cloud, but in practice, it depends upon how quickly the recovery operation is done since the re-allocation of the vacated data volume space is an indeterministic factor. Those with criminal intent can carry out the crime using cloud resources, delete all the files, logs, and trash folders, delete VM instances, terminate their account, and disappear. Consequently, no traces of their actions will be left around. In an attempt to provide the highest respect to privacy, CSPs delete the data entirely once confirmed by the users. For example, Google's current policy on deleted data states the following:

"After a Google Apps user or Google Apps administrator deletes a message, account, user, or domain, and confirms deletion of that item (e.g., empties the trash), the data in question is removed and no longer accessible from that user's Google Apps interface. The data is then deleted from Google's active servers and replication servers. Pointers to the data on Google's active and replication servers are removed. De-referenced data will be overwritten with other customer data

over time" [139]. Though this satisfies the privacy regulations, it works negatively from a forensics perspective. It highlights the fact that there is a disconnection between privacy regulations and forensics needs in the cloud.

Alsadhan *et al.* [131] proposed a "user signature" approach to attribute deleted files to a specific user. In this model, a unique ID (termed as user signature) is created for every CSU during the user registration process. The signature will be added to all data blocks in every writing process. When the CSU delete files, their signature will not be deleted. The only difference here is that the user signature piece of metadata persists while all other metadata is deleted when a file is deleted in cloud systems. Taking frequent snapshots of the virtual image is another possible solution, which has been explained in Section 3.4.3.1, under bullet point *Snapshot Analysis*.

3.4.3.8 Lack of specialist commercial tools

There is a lack of certified commercial tools for e-discovery and data acquisition of all artifacts in their entirety, including metadata, network logs, and hypervisor logs from a cloud environment. Metadata is a rich source of forensics information, as it contains the full revision history of files and the changes done to the file content and registry. Network logs provide traffic patterns and routing information. Hypervisor level logs provide critical information, including user account creation, virtual resource acquisitions, deletion, etc. However, researchers could develop tools and prove that remote acquisition is possible from an active user account. In addition, they proved that a wide range of forensics artifacts, including metadata and data remnants of deleted files, can be collected [98, 126, 140]

Further to that, Federici [132] extended the work outlined by Quick and Choo [98] and presented a Cloud data imager. The motivation for the work is that the traditional approach of bitstream copying of mass storage may not be possible in an investigating crime related information hosted on a cloud platform. Cloud data imager is a dedicated forensics software tool to log the entire conversation

with the cloud platform at the application level and in cleartext. In addition, the tool can support remote data collection from cloud storage, conforming to the principle of reliability and integrity of digital evidence by enforcing read-only access.

However, such holistic and certified tools that provide end-to-end forensics data collection, are not commercially widespread to date. Therefore, the CSUs and investigators often have to depend upon the CSPs to get the evidence.

3.4.4 Examination and Analysis

Once the digital artifacts are acquired, the next logical step is the examination and analysis phase. NIST Guide to Integrating Forensic Techniques into Incident Response (SP 800-86) defines forensics examination as: "forensics tools and techniques appropriate to the types of data that were collected are executed to identify and extract the relevant information from the collected data while protecting its integrity". The forensics examination may use a combination of automated and manual processes [42].

NIST SP 800-86 defines Analysis phase as "involves analyzing the examination results to derive useful information that addresses the questions that were the impetus for performing the collection and examination" [42]. Typically, in an Analysis phase, the significance of information artifacts is evaluated, and a narrative produced is supported by the evidence and a timeline of events. The narrative would help understand the case better and can be easily explained to a jury. Table 3.4 lists the challenges and recommended solutions in Examination and Analysis phase as applicable to the cloud platform.

Table 3.4: Examination and analysis phase: challenges and recommended solutions

No.	Challenges	Recommended Solutions	Comments
1	Lack of log framework	Comprehensive log management system [52, 54, 110, 126]	A good log helps to timeline the events, understand the case and eventually present the analysis better.
		AWS CloudTrail [114, 141]	Provides a comprehensive event log trail.
2	Evidence time lining	AWS CloudTrail can provide a partial solution [114, 141]	AWS CloudTrail provides comprehensive event logs in UTC, enabling time lining.
		Timestamped secure logs	End-to-end log helps to create an event time line.
		Secure provenance [74]	Provides the ownership and time history of data objects.
3	Encrypted data	Cloud key management infrastructure [118]	It helps to decrypt data volume.
4	Evidence data integration	AWS CloudTrail supports aggregation of log files [114, 141]	Requires third party tools for processing and analysis.
		Security information and event management [142]	Supported by tools like ArcSight.
		Data tracking [75]	Data tracking in the cloud, using provenance.

3.4.4.1 Lack of log framework

In general, cloud service providers use their own logging policy and format [114, 139, 141]. The lack of a proper forensically valid log framework applicable to cloud computing produces challenges in the time lining of events. However, logs are not mandatory for investigative purposes, and investigations can be conducted by examining file contents, access timestamps, and data remnants. Nevertheless, logs help an investigator to connect the dots. In Section 3.3 we discussed various digital forensics process and frameworks. Log framework forms a subset

of the comprehensive forensics framework. Recommended solutions proposed by researchers are briefly discussed below:

- **Comprehensive log management system:** The need for a comprehensive log management system, which contains enough information satisfying the forensic needs, has been flagged by many researchers [52, 54, 110, 126]. Marty [52] proposed a cloud application logging framework and provided detailed guidelines regarding 'when' to log, 'where' to log, and exactly 'what' to log to enable forensics investigation, reporting, and correlation for SaaS platform. In the Secure-Logging-as-a-Service (SecLaaS), the authors proposed a scheme to securely store and provide logs for forensics purposes. This scheme will allow the CSPs to store the logs in the cloud while preserving the confidentiality of the CSUs, and maintaining the integrity, while at the same time making it available publicly in a secure way [54].

- **AWS CloudTrail:** As a part of security operational best practice and to comply with industry and regulatory compliance, AWS has recently provided CloudTrail audit logging feature. This feature is a web service that logs the API calls to support AWS services and delivers the log file to a predefined AWS Simple Storage Service (AWS S3) bucket. AWS has created the audit trail web service by considering various logging and compliance and regulatory requirements and satisfying industry standards. However, the log data can also be used for forensics purposes. The log files are written in Java Script Object Notation (JSON) format. The log files can be extracted from the defined S3 bucket using AWS management plane without needing support from the CSP. AWS provides a comprehensive solution to restrict access controls to logs and allows enforcing integrity using S3 service side encryption techniques [114, 141]. The CloudTrail is a regional service aggregating log files across different regions and multiple accounts to a single S3 bucket. The CloudTrail logs events in UTC time format and provides comprehensive information including *who* performed the activity,

what they did, *when* and from *where*, which will be very useful in incident investigations as well as in evidence time lining [141].

3.4.4.2 Evidence time lining

Time lining provides an association of timestamps with each event or data item of interest to reconstruct a sequence of events. For timelining, the actions performed on objects have to be time stamped. Time lining assists in understanding evidence and data, putting information into context, which is potentially easier to understand. Further time lining also helps to explain the case better to a jury. Researchers have suggested the following methods to help evidence time lining:

- **Timestamped secure logs:** Such as AWS CloutTrail logs or Secure Logging as a Service proposed by Zawoad *et al.* [54].

- **Secure provenance:** Lu *et al.* [74] proposed secure provenance, citing it as the bread and butter of data forensics in cloud computing. Secure provenance records ownership and process history and provides trusted evidence of data objects; therefore, it plays a crucial role in cloud forensics. In addition, a correctly implemented secure provenance helps in evidence time lining because ownership and process history attributes provide information regarding 'who' owned the data object at a given time and 'who' updated the objects, respectively.

3.4.4.3 Encrypted data

Cloud customers are widely using encryption as a measure of securing their data. Data encryption is also done to satisfy legal and compliance requirements. However, criminals can also use encryption for illegal purposes. McKemmish [96] pointed out the widespread usage of encryption by criminals to hide unlawful images. Biggs and Vidalis [117] mentioned that 70-80% of an investigator's workload in UK law enforcement agency is spent on monitoring cloud computing usage by

pedophiles. Therefore, from a forensics perspective, encryption produces a significant barrier for an investigator. The report by Cloud Security Alliance on security guidance for cloud computing suggests the necessity of key management infrastructure with options to make the key accessible for forensics examiners as a possible future solution [118]. However, such an option should be supported by proper regulations and governance structure to avoid potential privacy violations, and misuse [4].

3.4.4.4 Evidence data integration

The evidence data in the cloud is spread across many devices across different locations, including mobile endpoints, middle-tier proxy servers, and the virtual cloud environment itself. Often, CSPs trade services among themselves, creating a complex array of intra-cloud dependency chains [4]. The trading produces additional challenges to collate and integrate the evidence data from multiple sources as investigators have to follow each link in the dependency chain. Combining all these pieces of data and creating the sequence of events are crucial parts of the forensics process. Suggested methods are discussed below:

- **AWS CloudTrail:** Supports aggregation of log files to a single AWS S3 bucket [141]. Since the CloudTrail is a dump of all events and user activities, third-party tools or specialized programs are required to do event correlation and forensics analysis. But, any additional layer of third-party tools adds another layer of 'trust' issue.

- **Security information and event management:** Tools, such as ArcSight, provide log integration from multiple sources and can be used for high volume data management, evidence time lining, correlation, and analysis [142].

- **Data tracking:** Zhang *et al.* [75] provided a mechanism of data tracking in the cloud using data provenance software tools implemented utilizing data

tracking principles would help to integrate user artifacts and draw event timeline.

3.4.5 Presentation

The final stage of the digital investigation process is the Presentation phase. This phase is also known as Reporting. To conclude an investigation, evidence collected during the previous stages is analyzed, and investigative reports are produced. NIST (SP 800-86) defined Presentation or Reporting as a process in which "Evidence collected during the collection or acquisition phase and the analytical reports are presented to the court of law during this phase of the forensics process." [42]. In this phase, the presenter's expertise and qualification and the credibility of the process can play a crucial role, since they are critical in determining the probative value of evidence.

Table 3.5 lists challenges and their recommended solutions in Presentation phase.

3.4.5.1 Jurisdiction

In Section 3.4.1.4, we discussed the jurisdictional issue related to evidence acquisition in the cloud. Jurisdiction is also a challenge while presenting the case because the law applicable to the country/region is different from place to place. As already noted in Chapter 1 that as per Australian law, the perpetrator must be in Australia or an Australian citizen overseas for cybercrimes to be accepted by the Australian court. However, suppose the perpetrator is overseas but not an Australian citizen, and there is no extradition treaty between the host nation and Australia. In that case, the Australian court has no jurisdiction, further strengthening the argument for an international framework. A subsequent study on critical criteria for cloud forensics capability found that lack of law, regulations, international cooperation, and legislative mechanism in cross-nation data access and exchange is by far the most forensics challenges in cloud [116]. Cloud foren-

sics, being a multi-dimensional issue and consisting of technical, organizational, and legal domains, requires collaboration between international law enforcement agencies and legal framework to conduct and present crimes conducted using cloud computing [4].

Table 3.5: Presentation phase: challenges and recommended solutions.

No.	Challenges	Recommended Solutions	Comments
1	Jurisdiction	Cross border law, International relations	Legal Agreements.
2	Chain of custody	Well defined process and guidelines on evidence handling [104, 105, 117]	Essential to establish the trustability of the evidence.
3	Crime scene reconstruction	Framework, process and guidelines, supported by tools and technology	There is a lack of such tools.
4	Complexity of cloud	Time lining of events	Difficult to explain the complexity of cloud to jury. Time lining would help to describe case.
5	Compliance	Established principles, process and procedures (e.g., ACPO guidelines)	An essential factor in strengthening the validity of the case.

3.4.5.2 Chain of custody

Proving the chain of custody in cloud forensics is a more complex process than traditional digital forensics in case presentation. A survey study conducted by Ruan *et al.* [116] stated that "a procedure and a set of toolkits to record and maintain the chain of custody in an investigation is critical" to consumers. Following established guidelines, e.g., ACPO guidelines, is one way to establish the chain of custody [35]. Following the guidelines would provide all relevant information with high standards for the case to stand in the court and to establish trust in the evidence presented [104, 117, 125]. However, our study finds a lack of certified tools to establish the chain of custody of evidence data.

3.4.5.3 Crime scene reconstruction

Reconstructing a digital crime requires an exact time lining of evidence, which further involves reconstructing virtual storage from physical disk images. NIST identifies that in the cloud environments, imaging of media has an added level of complexity that could cause damage to the original media [36]. Further certified tools, standards and processes are yet to be developed, especially for the cloud [36]. Studies also identified that, although it is possible to acquire cloud images from some public cloud (for example, AWS EC2 environment), it is not proven to be scalable to other cloud models (like MS Azure and Google) [126]. Further, special scripts or tools are required to extract the evidence from the image, demanding more layers of trust [126]. Therefore, reconstructing a crime scene in the cloud remains a challenge.

3.4.5.4 Complexity of cloud

Juries in the common law system are made of individuals from the general public, often with very limited or no understanding of cloud computing technology. Therefore expert witnesses will be faced with the daunting task of ensuring juries understand the principles and technology of cloud computing [104].

3.4.5.5 Compliance

Compliance, generally refers to the satisfactory adherence to a set of regulatory or legal rules in an investigation process. For example, for a digital investigative case to be legally valid in a court, it must meet the compliance requirements. One way to meet the compliance is by following an established standard procedure throughout the forensics process. An example of established procedures is the ACPO guidelines [35]. ACPO guidelines are more mature, and practical [95]. Though the guidelines were developed prior to the advent of cloud computing, the guidelines can be applied to cloud computing also [122]. Lallie *et al.* [95] provided guidelines on applying ACPO principles in public cloud forensics investigations.

It is to be noted that if an organization creates and follows its own standard operating procedures, it may not stand up in a court of law.

3.5 Research Impacts

As already stated, some parts of this chapter, specifically Sections 3.3, and 3.4 have been published in [37]. Since its publication, the paper [37] has been cited more than 150 times till date. The material and contributions presented in [37] have been used by researchers to advance the field further. This section provides an analytical summary of all important and recent research outputs which used the contributions from [37] and establishes the relevance of the authors' contributions made in [37].

Roman *et al.* [143] applied the concepts presented in [37] to fog computing, mobile edge computing, and mobile cloud computing. The mobile, edge, and cloud computing technology platforms intersect boundaries, and there are cross-platform dependencies. Hence, the authors argued the necessity for a common forensics approach instead of a compartmentalized approach per domain and extended the contributions presented in [37] to address the security and forensics challenges in cloud/mobile or edge computing space.

Cloud services providers assure a high level of security compliance by certifying against specific standards. However, the long-term certifications are not enough, as the cloud is a constantly changing platform. Therefore, continuous auditing of the cloud environment is necessary to increase the trustworthiness of the certifications. The work of Lins *et al.* [144] used data collection methods and solutions presented in [37] for data collection of relevant data for continuous auditing of cloud services and thereby to create a trustable cloud platform. Their subsequent work on designing monitoring systems for continuous certification of cloud services utilizes the methods used for evidence collection explained in [37] to collect the necessary data for cloud service certifications [145].

Alqahtany *et al.* [60] used the acquisition methods described in [37], further ex-

tended the work for forensics acquisition and analysis for IaaS platform, using an agent application running on VMS to collect the data. Further in the subsequent work of Pichan *et al.* [146] used the forensics challenges and gaps highlighted in [37] and proposed framework to solve the issues related to forensics logging and unified log formats to help crime detection. Again, in the log aggregation forensics analysis framework proposed by Khan *et al.* [57] used the systematic and thorough study conducted in [37] and extended it to develop their log aggregation framework. Manral *et al.* [49] used the same five stage digital forensics process presented in [37], and further enriched the cloud forensics challenges by proposing a cloud forensics solution taxonomy and classifying the solutions.

Alsadhan *et al.* [131] took the VM snapshotting solution presented in [37] and conducted experiments to find out the effectiveness and overhead of the VM snapshot solution. While Alenezi *et al.* [147] argued that cloud security and digital forensics in the cloud is a converging field and used the forensics challenges listed per investigation phase in [37] to connect cloud security with cloud forensics. In the later work of Alenzi *et al.* [148] provided an experts reviews of cloud forensics readiness framework for organizations and used the cloud forensics challenges described in [37] as the prime source for preparing the questionnaire for the expert survey. Moussa *et al.* [149] while proposing a CSU and CSP bilaterally agreed 'Cloud-forensics-as-a-Service' evidence collection model, primarily used the data acquisition and degree of control vs. trust layer explained in [37] to define the parameters for their model. In the subsequent work of Moussa *et al.* [100] on a consumer oriented cloud forensics process model, used the cloud forensics challenges highlighted in [37] to draw the requirements for their proposed model.

Further, in the work of Kumar *et al.* [150], citing the magnitude of the cloud forensics challenges described in [37] argued the necessity for a standardized procedure for assuring security and privacy in the cloud. While Montasari *et al.* [151] in their work on next generation digital forensics, challenges and future directions

used the contributions from [37] to analyze and list the most difficult technical challenges and propose future research directions. The contributions presented in [37] have been used as a significant input in the recent works; such as (i) for the development of design strategies of evidence collection framework [152], (ii) to propose Blockchain based solutions to address cloud forensics challenges [153], (iii) to develop digital evidence case management tool [154], and (iv) to prepare the requirement engineering guidelines [155]; all, related to cloud forensics domain. Recently, the contributions made in [37] has been also used to design or to develop process models or frameworks such as (i) to design and enforce a reliable timeline [156], (ii) to develop digital evidence case management tool for collaborative forensics investigation [154], (iii) to develop and propose cloud forensics taxonomy [3], and (iv) to model cloud forensics-by-design framework [157]. In addition, some of the recent survey papers or work highlighting the current state of cloud forensics and future directions presented in [13, 158–161] used the main contributions from [37].

Further, there have been significant contributions to IoT forensics space also, which used the contributions made in [37]. Few to mention are: (i) Alenezi *et al.* [23] extended the digital forensics process described in [37] by connecting with IoT forensics. Islam *et al.* [162] in their work on comprehensive data security and forensics investigation framework for cloud-iot ecosystem, used [37] and extended to cloud-iot ecosystem, and established that there are commonalities in cloud forensics and IoT forensics domains. Hou *et al.* [91] mentioned the successful models for evidence acquisition presented in [37] and how similar approaches can be applied to assist IoT forensics. In the study to understand IoT forensics challenges and future directions conducted by Wu *et al.* [163] used the contributions from [37] as a key input to frame the online survey and to connect cloud forensics with IoT forensics. While Jahankhani *et al.* [164] highlighted the need for a digital forensics investigation process model for medical IoT devices. The work and forensics process model presented in [37] has been used as one of the main input

sources of information while arguing and establishing the necessity for a similar forensics process model for medical IoT (IoMT) devices.

The work presented in [37] has been referenced in many Doctoral and Master's program theses, or part of it incorporated into book chapters by different authors [165–168].

The above cited examples establish the relevance and currency of the contributions made by the authors in their publication [37].

3.6 Chapter Summary

The cloud computing environment is an attractive platform for hackers to commit digital crimes because it is economical, easy to acquire and release resources/services, and has much less chance of conviction. However, proving a crime and finding the culprit behind the action is a digital forensics task, and performing the task in a cloud computing platform is complex and challenging. Traditional digital forensics technologies and the process cannot be applied to the cloud either. Though cloud computing offers tremendous benefits to customers, it also offers similar attractions to those with evil intent. As a result, cybercrime on the cloud continues to increase.

This chapter begins with a brief description of various cloud forensics process models. Subsequently, the Digital Investigation Process (DIP) has been discussed in detail, identifying every digital forensics process and sub-process. The main contributions of this work are the systematic analysis of cloud forensics challenges, their possible solutions applicable to different phases of the forensics process, and a detailed analysis of the recommended solutions. In addition, we have identified the maturity of the solutions and identified the pros and cons of the suggested solutions, where applicable. The contributions from this work have been widely cited, enriched, or extended to other similar domains, including IoT forensics. This chapter concludes with a research impact summary of all such recent research work that used the content and contributions made in this work.

Chapter 4

Forensics Logging Framework for Cloud Computing

Logs, providing detailed events of actions on a time scale, have always been a prime forensic artifact. Any computing system, including cloud systems, produces diverse set of logs. The various logs include network logs, system logs, database logs, and application logs. In general, all such logs are stored in different physical locations and will be in a different format with no commonality. For example, network logs are generally placed in network devices, database logs in databases, and operating system logs in the system partition. Moreover, these logs are not easily accessible in the cloud, their locations are unknown, and often fail to provide any critical clues due to poor logging practices [81]. Therefore, such logs will have limited or no value in forensics since they seldom meet the specific needs of digital forensics investigations. For this reason, these logs are not commonly used in a digital investigation. Therefore, to support forensics, requires purpose built event logs, and a collection of event logs forms evidentiary forensics artifacts.

Event logs (or logs for brevity) are a systematic representation of the state of an object and the actions producing a change in status of the object, recorded on a uniform timeline, along with full client details. This particular process

is known as *forensics logging*, and that has become an integral part of cloud forensics. In context of forensics logging, each **object** refers to a 'digital file' or *executable*, **state** of an object means a *read, write* or *run*, and an **action** can be user *login/logout*, *upload*, *download*, or *executing* commands, etc. on a cloud platform.

In Chapter 3, specifically in Sections 3.4.1 and 3.4.4, we identified that the lack of a formal log framework is a significant issue in cloud forensics. Therefore, to advance the cloud forensics capabilities, there is a need to have a comprehensive log management system, which records and preserves a repository of the event history as a trustable source of evidence and that is the main focus of this chapter. This chapter, proposes and demonstrates a practical cloud forensics logging framework. The specific contributions of this chapter are:

1. It identifies and lists the forensics investigative requirements of the forensics practitioners;
2. It proposes a cloud forensics logging framework and architecture;
3. It designs, builds and demonstrates the architecture using ownCloud - an open-source cloud platform;
4. It validates the framework against the ACPO guidelines and relevant challenges listed in the NIST Cloud Computing Forensic Science Challenges (NISTIR 8006) publication;
5. It demonstrates the relevance of the framework by comparing it against other frameworks using the cloud forensics challenges listed in the NISTIR 8006 publication;
6. it establishes the framework's applicability for practical investigative purposes and contributes to advancing cloud forensics capabilities.

Our work was aimed at helping forensics examiners and law enforcement agencies establish confidence in log artifacts, by maintaining the evidence integrity and

straightforward interpretation of logs by presenting them in a user-friendly way. The framework is specifically designed to address the business needs of forensics practitioners. Our work also helps the investigators build a collective chain of evidence and the CSPs to provision forensics-enabled logging. To the best of our knowledge, this is the first attempt to model a forensics framework looking through the lens of an investigator. The major part of this chapter has been published in the author's journal article [146].

The rest of this chapter is organized as follows: Section 4.1 describes the problem, related work conducted in this field, and the motivation to carry out this specific research. Section 4.2 details the methodology. Section 4.3 describes the framework design, architecture, experimental environment used to conduct the experiments and gather test results. Analysis of the results and findings are described in Section 4.4. Section 4.5 provides a comparison of the framework with other similar frameworks and demonstrates the relevance. Section 4.6 summarizes the chapter, and Appendix C lists the detailed test results.

4.1 Motivation

As already noted, a forensics investigation is a post-crime activity, whether traditional forensics or digital forensics. In a cloud computing scenario, the forensics action starts with the evidence identification and collections process. Recall in chapter 3 we mentioned that evidence identification and collection in the cloud is very complicated and challenging. Primarily due to the transient nature, data volatility, multi-tenancy (causing privacy issues and co-mingling of different user's data), evidence data spread across ever-changing hosts and geographically distributed physical systems across jurisdictional boundaries [47, 58].

The cloud forensics investigators must have the evidence to ascertain the **what, when, where, how, why,** and **who** (i.e., 5W1H) elements of a crime to establish how a cyber adversary compromised the computer system(s). These elements help reconstruct the events that led to the incident and set the crime

scene. Therefore, cloud computing applications must have the capability to produce digitally admissible evidence [105]. Providing a trustable event history as an audit trail helps reconstruct the incident, assess the damage, and find the adversary.

As already noted, in Sections 3.4.4 and 3.4.4.1 we identified the need for a comprehensive audit trail logs and in fact some cloud systems provides such logs (for example AWS CloudTrail). These logs contain data of forensics value. However, such logs are not left around in a cloud environment once the perpetrator terminates their account. A perpetrator can create a cloud account and acquire computing resources that can be used to launch a cyber-attack. The perpetrator can then quickly terminate their account and disappear into the ether, leaving no traces. For CSPs, the cost of retaining the storage, logs and data of terminated users could be particularly substantial.

Compounding the problem of log retention, CSPs often implement propriety technology and log architecture. Therefore, prior knowledge of the log architecture is required to analyze and interpret the logs, making the investigation task much harder. Moreover, to access and collect the logs, CSPs' cooperation and willingness are essential, and in general, it is hard to come by [169]. Even if the CSPs cooperate with the investigators; there is no established process to verify that the CSPs are providing correct logs to the investigators either [63]. Given the situation, there is a solid need to have an architecture in which the evidence preservation mechanism is outside the control and knowledge of the CSUs. With the capability to persist the evidence in a format that is of forensics value, investigators should be able to collect the evidence when needed.

4.1.1 Forensics Logging: Relevance

Given that logging to support forensics is critical to solving digital crime, there has been significant research effort looking into the problem of recording and capturing trustworthy logs from multiple dimensions. We are citing here some of the

most relevant references and their contributions to cloud forensics logging. Marty [52] provided a guideline for cloud application logging, primarily for SaaS delivery model. Sang [110] described a log-based process for cloud forensics primarily for Software-as-a-Service (SaaS) and Platform-as-a-Service (PaaS) models. Zafarullah *et al.* [133] proposed a method for identifying and extracting log entries relevant to forensics from the Linux operating system and security logs. They proved their proposal in the Eucalyptus cloud environment and could produce fingerprints to reconstruct an event. Dykstra *et al.* [126] evaluated popular forensics data acquisition tools and proved that they could successfully return volatile and non-volatile data from the cloud, and examined various levels of trust required in the cloud. They further enhanced the work by developing digital forensics tools for OpenStack cloud platform, namely forensics Open Stack Tools (FROST) to collect logs from virtual disks, application logs, and firewall logs. FROST works at the cloud management plane requiring no trust of the guest machine [59]. Zawoad *et al.* [54] proposed a Secure-Logging-as-a-service that stores entire virtual machine logs and securely provides access to the logs for forensics purposes. They further expanded their work in which they presented a scheme for tamper-proof secure logging and proved that the integrity of the log could be ensured, even if the cloud actors such as the service provider, the user, and the investigator collude. The scheme provides that any violation of the integrity property can be detected during the verification process [63]. A layered cloud logging architecture was presented in the work of Patrascu *et al.* [53], including the way of monitoring activities in a cloud infrastructure. A substantial amount of research work has also been carried out in securing the logs, ensuring the integrity and trustworthiness of logs, secure transportation of logs, and enabling cloud infrastructure to provide secure logging as a service [52, 53, 63, 110, 110, 170]. However, none of the work looked from the angle of forensics practitioners and the information they wanted to capture in the logs. Therefore, we are taking it one step further by addressing this gap in this work.

Major CSPs provide various types and levels of logging, but primarily for security and compliance reasons; examples are AWS CloudTrail [114], Azure Activity Logs [124], and Google Stackdriver logs [171]. These logs can also be used as primary source of forensics evidence. In general, all such services are paid services. The customers have complete control to enable and configure the logging services. However, the criminals are not going to configure and allow the logging of their actions. Instead, they would be doing the opposite, trying to erase all the traces and evidence. Further, NISTIR 8006 publication on cloud forensics challenges mentions that *"an important source of forensics analysis is logs many of which may be available in cloud environment but may be hard to access or aggregate due to the segregation of duties among actors and lack of transparency of log data"* [36]. This substantiates the necessity of easily retrievable logs outside the accessibility of CSUs. Events are mandatorily logged and stored securely, with logs satisfying investigative forensics requirements. At the same time ensures that one user log does not co-mingle with other user's logs in a multi-tenant environment, thereby safeguarding the privacy of co-tenants.

4.1.2 Forensics Logging: Motivation

Though cloud security has matured significantly and many larger providers have many security accreditation and compliance, such as AWS compliance [76], Azure compliance [172], our study shows that they are not yet forensics mature, or forensics ready yet [47, 173]. Therefore, the motivational element for this work is promulgated from the following findings. There are lots of research outputs in the area of cloud forensics logging, which are outlined in Section 4.1.1. Still, none of them addressed forensics logging, looking at the needs of forensics practitioners. To solve a digital crime, the logs with evidence suitable for the investigators are critical and an absolute necessity. In this work, (i) we examine the logging requirements for forensics needs which the law enforcement wants such that it can provide better value and practical benefit to the investigator; (ii) the log solutions

provided by major CSPs are all user configured and CSUs have full access to the log artifacts, thereby rendering the log artifacts invalid or untrustworthy. Forensics logging should be different from other logs, and it should be completely outside of the preview and accessibility of the user community; (iii) the need to have a comprehensive logging with transparency, where different CSUs actions are not mingled in a multi-tenant platform and available in persistent storage; (iv) the logs to have a standard format, and more importantly (v) to contribute towards the advancement of forensics readiness of cloud platform.

4.2 Methodology

The methodology used for this project on a forensics logging framework for cloud computing is described as follows:

1. Derive and describe the forensics logging requirements.
2. Design and describe the framework architecture.
3. Describe the experimental setup and implement solution as per the framework design.
4. Describe the use cases used to test and validate the framework.
5. Test and validate the framework.
6. Capture test results and analyze the results.
7. Demonstrate suitability of the framework as a practical cloud forensics logging framework by
   (a) Analysing and establishing that the framework test results satisfies the requirements.
   (b) Doing a further analysis of the framework results against ACPO guidelines and NISTIR 8006 publication to establish that the framework can satisfy the investigative needs of the forensics practitioners.

8. Finally, compare the framework with similar other frameworks using the forensics challenges listed in the NISTIR 8006 as the key.

4.3 Framework Design

In this section, we introduce the design of the cloud forensics logging framework. The framework is termed as **CFLOG**. We step through items #1 through #5 listed in Section 4.2 on methodology describing every step in detail.

4.3.1 Cloud Forensics Logging Requirements

This Section proposes a set of cloud forensics logging requirements that aims to satisfy the needs of forensics practitioners. The requirements are derived from ISO/IEC 27037 standard [174], the ACPO guidelines [35], and two NIST publications, i.e., NISTIR 8006 [36] and NIST 800-144 [38].

International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC) forms the specialized system for worldwide standardization. They set the standards and provide guidelines. ISO/IEC 27037 Information technology – Security techniques - Guidelines for identification, collection, acquisition, and preservation of digital evidence is the standards document providing guidelines for specific activities in handling digital evidence during a digital investigation process. The document describes the digital evidence handling process and provides general requirements for digital evidence handling.

Recall that in Section 2.2.4, we mentioned that the ACPO guideline describes the digital investigative process and provides the guidance on how to conduct digital investigations. Forensics practitioners have used ACPO guidelines as the standard for digital investigations in many countries [95, 175]. Though the guidelines do not make any specific reference to cloud forensics as such, a study conducted regarding the applicability of ACPO guidelines to cloud forensics concluded that the ACPO guidelines could generally be applied to cloud forensics investigations

too [95, 122]. Therefore, ACPO guidelines are also a good source for drawing forensics requirements.

As already mentioned in Section 2.2.4, NIST Cloud Computing Forensic Science Challenges (NISTIR 8006) lists, defines, and describes the cloud forensics challenges. Knowing the challenges, one can draw requirements to address them. (It is to be noted here that only a draft version of this document was available during the development of this framework. This publication was developed by the NIST cloud forensics working group, and I was a team member of the working group till mid 2019. The framework was later refined using the current published version of the document).

NIST 800-144 provides an overview of the security and privacy challenges pertinent to the public cloud and guidelines on tackling them. Forensics data also need to conform to security and privacy regulations. Therefore this NIST publication is also used as an input for the forensics requirement gathering.

We used different but relevant sources to draw the requirements, but the ACPO guidelines and NISTIR 8006 provided more valuable requirements. Therefore, we additionally validated the framework against these two sources to demonstrate its applicability.

Recall that Figure 2.2 groups the cloud forensics into three dimensions, i.e., technical, organizational and legal. The forensics requirements span across all three dimensions. However, given that this work deals only with the technical dimension, the specific requirements related to the technical dimension and logging are considered here. The consolidated set of cloud forensics logging requirements are captured and summarized in Table 4.1.

Table 4.1: Cloud forensics logging requirements.

Id	Title	Description
R-1	Auditability	The ability to establish by an independent assessor to evaluate the activities performed, following appropriate documentation.
R-2	Repeatability	The ability to establish that the same test results are produced using the same measurement procedure and methods, using the same instruments/process and under the same condition, and can be repeated any time.
R-3	Reproducibility	The ability to establish that the same results can be reproduced using the same method/process, i.e., be able to reproduce the same incident by repeating the same process. In terms of logging this means that the same event records are logged, which can determine what and how the incident happened.
R-4	Justifiability	The ability to establish that the evidence can fully justify all actions which caused the incident.
R-5	Event trail	The ability to capture, create and record an event trail of the event history.
R-6	Traceability	The ability to track or map events along a timeline by connecting the dots in the chain of actions.
R-7	Provenance	The ability to clearly attribute the action to a doer.
R-8	Quality	The ability to answer *what* happened, *when* it happened, *who* did it, *how* it was done and from *where* elements of an incident.
R-9	Privacy	The ability to establish non-interference of one CSU's activity with others (i.e., clear separation of every CSU logs).
R-10	Trust	The ability to establish confidentiality and integrity in the logs, and they are true to the original.

Continued on next page

Id	Title	Description
R-11	Evidence collection	(i) The ability to collect the logs, when needed, preferably without affecting the cloud services provided to other customers. (ii) The ability to collect logs per user basis (i.e., selective evidence collection). (iii) The ability to collect logs across jurisdictional boundaries, if necessary.
R-12	Evidence correlation	The ability to correlate the logs across multiple CSPs.
R-13	Evidence segregation	The ability to segregate the logs across multiple CSUs in the log archive.
R-14	Evidence identification	The ability to find or locate the logs easily i.e., pre-defined log location.
R-15	Unified log format	The ability to have unified log format (i.e., interoperable log format) across multiple CSPs.
R-16	Persistence	The ability to persist the logs even after the CSU account has been terminated.
R-17	Relevance	The ability to demonstrate that the material acquired is relevant to the investigation, i.e., the logs should contain information of value in assisting the investigation of the incident, and there is a good reason for to acquire the logs.
R-18	Reliability	The ability to demonstrate that the logs are reliable.
R-19	Sufficiency	The ability to establish that the logs have sufficient enough information for the investigation to be carried out (Quantity).
R-20	Usability	The ability to represent the information supporting readability, understandability, and interpretability.

We propose a practical approach to support cloud forensics by presenting an event logging framework in the following sections. The framework has been designed to consider the requirements listed in the Table 4.1 and offer a flexible,

scalable, maintainable, and easily extensible architecture. We then demonstrate the model by implementing it on an ownCloud instance. Finally, we validate the model by correlating the model's outputs using scenario-based use cases and examining whether the model meets the requirements. We believe this work complements the work done so far in the same space.

4.3.2 The CFLOG Framework and Architecture

The design and architecture of the proposed framework for cloud forensics logging, termed CFLOG, are presented here. The main characteristics of the framework are:

- The CFLOG application creates an auditable trail of events that happened due to user actions along a uniform timeline. The trail of events is repeatable by following the same set of actions in the same environment, and the results are reproducible by following the same test scenario.

- The log captures detailed information regarding *what, when, who, where* and *how* (i.e., 5W1H)of an event. Any forensics investigation or security incident handling seeks to answer these six key questions or parameters of an incident [67, 176]. Therefore, recording the entire event history supporting forensics, increases the sufficiency and relevance of the logs.

- The CFLOG application generates logs that enable the time lining of events and records sufficient information to justify what caused the event. The events are time-stamped with Coordinated Universal Time (UTC). (Note: UTC is the primary time standard by which the world regulates clocks and time and is used in many internet standards [177]). As the CSUs and cloud environment can be in different time regions, recording events on a standard time scale helps to timeline events accurately.

- The CFLOG application runs at the Hypervisor level on the virtual stack of cloud environment, as depicted in Figure 4.2, i.e., outside the control

and accessibility of the CSUs. The figure represents a simplified version of cloud reference architecture defined by NIST [178]. In the architecture, the layers Hypervisor and below are controlled by the CSPs and not accessible to the CSUs, whereas those above Hypervisors are accessible to the CSUs. The CFLOG application is embedded at the Hypervisor level.

- The logs are generated mandatory. However, the CSU has no access to the logs, i.e., the logs are stored outside the CSUs preview and control. Hence, increasing trustworthiness, reliability, and confidence in the logs.

- The architecture proposes a uniform log structure. However, the architecture is very much scalable and extensible, which means that additional logging parameters or attributes can be added by making entries in the data structure. In addition, the architecture supports easy adaptation of specific forensics needs and compliance with standards, or jurisdictional requirements.

- The logs are stored in a pre-configured location in the cloud stack's CSP controlled area, making the evidence location identifiable.

- The event logs are stored in persistent storage. The lifetime of the logs is a configurable parameter. (The duration of storage is a jurisdictional matter. Typically, the logs are stored only for a few days and purged if no incidents have been noticed by then).

- The architecture proposes to create one or multiple uniquely identifiable log files per CSU, separating every CSU action from one another, stored in segregated logs. It makes it easier for Law Enforcement Agencies (LEA) to collect and retrieve the log data belonging to a specific user. If all the CSUs' actions were combined into one log file, that would create an enormous task of extracting the relevant parts of the log, removing the noise to produce the data of interest. Furthermore, in any court of law, modifications of the

logs will often be subject to questionable doubts and significantly weaken the case.

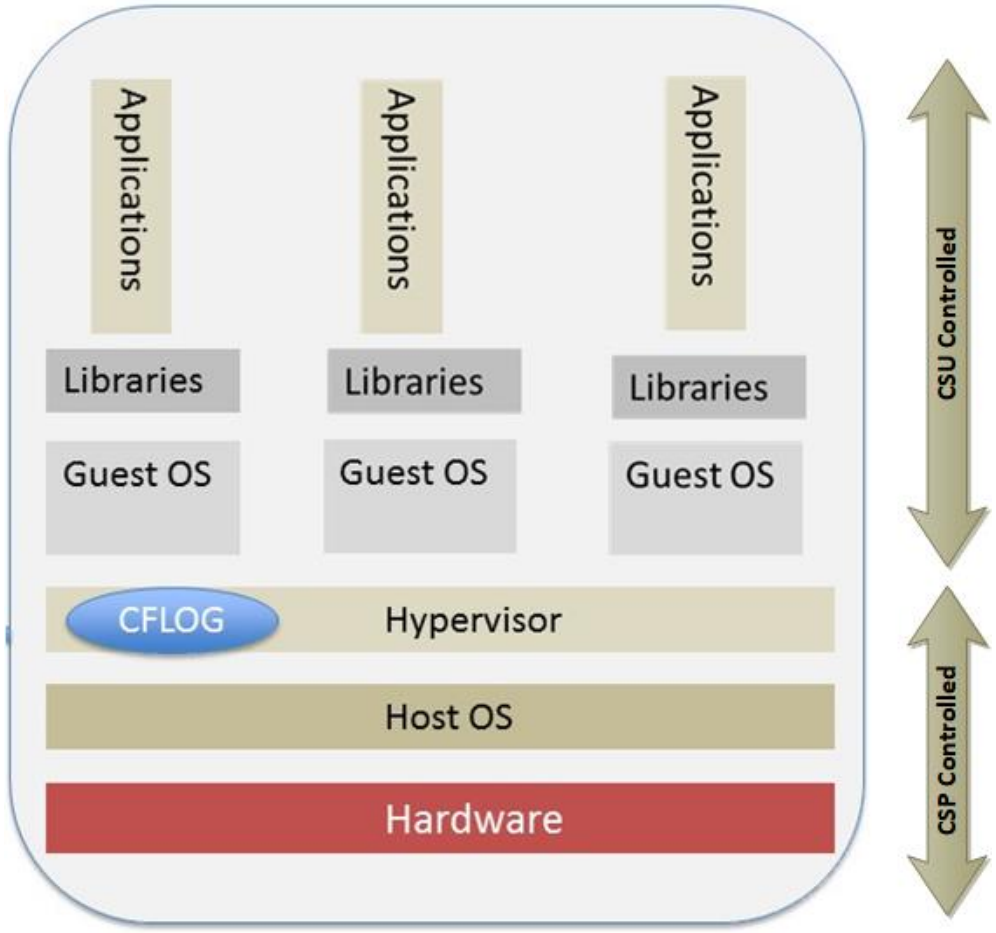


Figure 4.1: Cloud forensics log (CFLOG) stack

4.3.2.1 The log structure

Evidence is stored as log data. The structure of the log file can be represented as a set of Log File Per User ($LFPU$), i.e.,

$$LF = \{LFPU_x, LFPU_y, \cdots LFPU_n\} \tag{4.1}$$

where $LFPU_x$ is the Log File corresponding to CSUx.

Typically in an organization, there can be one root user account owner and multiple users under the root account using the same cloud platform. In such cases the Log File (LF) consists a set of Log File Per User for every root user account and sub-user account combination. As an example, for a given root user r and sub-users i, j.....n, the LF consists of

$$LF = \{LFPU_{r,i}, LFPU_{r,j}, \cdots LFPU_{r,n}\} \tag{4.2}$$

Where the root user id r can be $r1....rn$, if there are multiple root accounts. LFPU consists of a set of log files created per user at different point in time. For a given root user r and sub-user i

$$\{LFPU_{r,i}\} = \{LFPU_{r,i1}, LFPU_{r,i2}, \cdots LFPU_{r,in}\} \tag{4.3}$$

Where $LFPU_{r,ik}$, is the k^{th} file belonging to root user r, sub-user i

Each LFPU contains multiple Log Entries (LE). For $LFPU_{r,i1}$, the LE consists of:

$$\{LFPU_{r,i1}\} = \{LE_{r,1}, LE_{r,2}, \cdots LE_{r,n}\} \tag{4.4}$$

Each log entry LE contains a set of parameters required for the forensics investigation. The parameter set for a log entry LE consists of:

$$\begin{aligned} LE = [&UTC\_timestamp, user, source\_ip, source\_port, destination\_ip, \\ &local\_time, proto, [file\_param1, folder, file\_name, size, ...], \\ &location, user\_action, additional\_parm]. \end{aligned} \tag{4.5}$$

The parameters are described in Table 4.2. The LFPU names are created by using the combination of root id, CSU id, and file creation date-time. The structure allows having multiple log files per user, identified by the unique file names. Given the CSU id, the LFPU for the CSU can be easily identified and located in the CFLOG log archive. The LFPU facilitates evidence segregation in a multi-

tenancy environment. Note that evidence segregation has been identified as one of the key challenges in cloud forensics [4, 36]. The LFPU are stored in persistent storage. When a log file is closed, a hash of the file is created and held simultaneously, and the file attributes are changed to *read* only. The log records are in Java Script Object Notation (JSON) format. The log files are in Java Script Object Notation (JSON) format. JSON is an open standard data interchange format that uses human-readable text to store and transmit data objects. JSON represents data as a key-value pair. For investigative purposes the generated logs i.e., LF can be transferred from CSP to the LEA using tamper-proof secure protocol proposed by Zawoad *et al.*[63] when needed.

The parameters in LE_i are configurable. Any entry can be added or removed depending upon specific forensics needs, making the framework scalable. Table 4.2 lists commonly used parameters and their meanings.

Table 4.2: CFLOG - Log entry parameters and description

Parameter	**Description**
UTC_timestamp	timestamp in UTC *(when)*
user	CSU id or CSU name *(who)*
source_ip	ip address of the source device *(where)*
src_port	the port to which the source device is connected *(how)*
destination_ip	destination ip address of the target host *(the target)*
local_time	local time of the source device *(when)*
proto	protocol used for communication *(how)*
file_param1	contains an array of file parameters, *(data supporting CSU actions)*
folder	folder name of the file
file_name	file name *(objects or action parameters)*
size	file size
location	geographic location (user location *(where)*)
user_action	the action carried out *(what, how)*
additional_parm	additional info supporting the user actions (optional)

Sample log output is shown in Figure 4.2, where each parameter is represented by a key-value pair. The key-value pair representing the data makes the file easy to parse, understand and interpret.

```
{“Records”:[
{
   "UTC_Timestamp": "yyy-mm-dd hh:mm:ss",
   “user”:  “User name”,
   “src_ip”: “xxx.xxx.xxx.xxx”,
   “src_port”: “nnnnn”,
   “destination_ip”:  “yyy.yyy.yyy.yyyy”,
   “local_time”: “yyyyy-mm-dd hh:mm:ss”,
   “proto”: “protocol”,
   “file_params1”: {
         “folder”: ”folder name”,
         “file_name”: “file-name”,
         “size”: “file size”,
      },
   “file_params2”: { .....},
   “location”: “geographic location”,
   “action”: “user action”
},
{.....
  .....},
{.....
   .....},
 ]
}
```

Figure 4.2: CFLOG structure

4.3.3 System Details and Experimental Environment

To build the cloud platform, we used ownCloud[1] - an open-source product that can be used to create, configure and test cloud applications. Many researchers have used the platform to run simulations and validate their concepts and theory, including in the forensics space. To cite a few examples, Martini and Choo [140]

[1]https://owncloud.org

conducted the study on cloud storage forensics using ownCloud, Alex and Kishore [61] validated their cloud forensics framework using ownCloud. Rahman and Choo [24] used ownCloud platform for their work on Integrating digital forensics practices in cloud incident handling. Therefore, ownCloud is a proven platform for conducting such a study.

The cloud environment has been created using ownCloud (version 8.0), MySQL (version 5.6.17), and Appache web server (version 2.4.9), with PHP (version 5.5.12) running on 64 bit WampServer stack. The virtual environment is running on a host computer, with Intel (R) Core I7, 3.40 GHz CPU, and 8 GB RAM, with 500 GB hard drive, with 64 bit Windows 7 (Enterprise Edition) as the host operating system (OS). The host OS was later re-created with Windows 10. Figure 4.3 illustrates the experimental set up. Cloud forensics log application (CFLOG) is written using PHP and deployed as part of ownCloud platform. The CFLOG captures CSU actions, with parameters and values, which is then written to persistent storage as per the format explained in Figure 4.2.

The log files, i.e., LFs are delivered to a specified location once every configured time interval (e.g., once per hour) in compressed format. We examined the size of the files to ascertain possible overhead for providing such services. On average, the record size was around 400 bytes with a minimal set of parameters logged (as shown in the result data sets listed in Appendix C, specifically for the use case shown in Figure C.3. Repeating the experiments with a detailed set of parameters logged produced about 1200 bytes of data per activity record. Since the file was compressed, the storage size decreased approximately by ten-fold, e.g., in our environment that generated around 40,000 records, it consumed about 4.8MB of storage. Therefore the storage cost is insignificant. Applying life cycle rules (i.e., retention period to the logs) and purging the older files can reduce the storage even further.

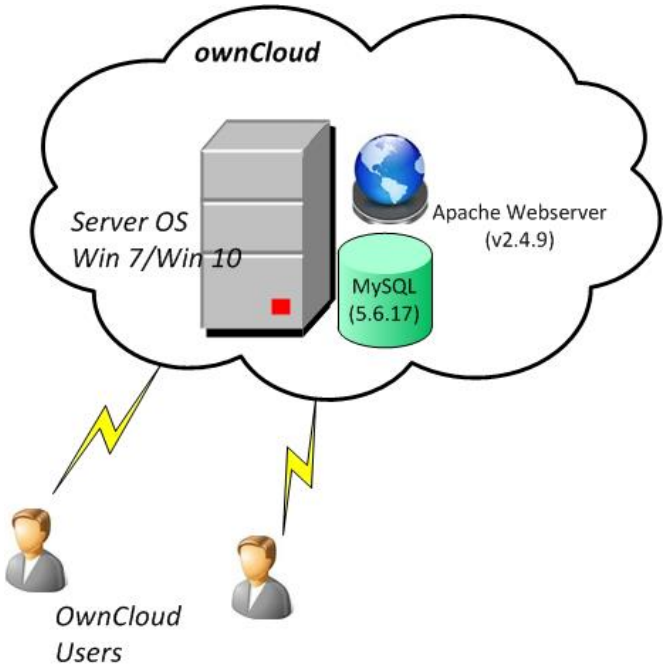


Figure 4.3: CFLOG framework experimental environment system configuration.

4.3.4 Validation of the Framework

In this section, we validate the proposed cloud forensics framework as per the description in Section 4.2. We are presenting here two sample use cases used to validate the CFLOG framework. However, we used more complex use cases and large data sets and repeated the tests.

4.3.4.1 Use cases

Use Case 1: Cloud storage is commonly used to store a large amount of data, including confidential personal information like social security or identification card numbers. Moreover, businesses often use the cloud to keep their clients' data, including credit card details to enable faster transactions, usability, and online access from anywhere. Hence cloud is an attractive target for data thieves. However, for security engineers and law enforcement agencies, the very nature of cloud computing makes it very difficult to secure and enforce the law and

catch the wrongdoers. In this particular use case, we assume that a malicious actor, referred to as *(user_hacker)* register for a CSU account (possibly with an assumed name), legitimately procure reasonably immense computing power and services in the cloud. Using the cloud resources, the *user_hacker* then executes malicious programs, such as password cracking, conducting identity theft and harvesting critical information of other CSUs. Collected the harvested data in a file. Subsequently, the results are shipped out. The *user_hacker* vanished after deleting his account with the cloud and ultimately terminating the services. When the cloud user account gets terminated, the storage allocated on cloud infrastructure will be quickly reallocated to other CSUs, enabling the criminals to disappear with no trace left. For Use Case 1, we consider the following test scenario: (Note: Steps 2 to 9 can be a repetitive action)

1. Register and create CSU account for *user_hacker*.
2. Login to the cloud as *user_hacker*.
3. Acquire cloud computing resources (computing power, memory, and storage).
4. Create folder and storage locations necessary to run the program.
5. Upload malicious programs and files.
6. Execute the malicious programs and scripts.
7. Collect the resulting output file.
8. Download the output files.
9. Log out of the session.
10. Finally terminates the CSU account of *user_hacker*.

Use Case 2: In this Use Case, we assume that a malicious actor *(User_Evil)* trades child pornography over the internet. He used to collect and keep a large

set of pornography image files on his laptop. Later *User_Evil* found out that a cloud computing environment is the best way to distribute the files with no traces of actions left behind. Subsequently, *User_Evil* created a cloud account, acquired the necessary storage space, and uploaded the image files to the cloud storage. The files were then wiped out from the local machine using special tools to erase any possible traces. *User_Evil* used the cloud services as a storage and transmission center and quickly terminated the account once the job had been done.

For the Use Case 2, we consider the following test scenario:

1. *User_Evil* Creates a CSU account.
2. *User_Evil* Logs in/logs out like a normal user and uses the cloud resources as often as required.
3. *User_Evil* Upload illegal image files to the cloud.
4. The image files are made available to the potential buyers by sharing them. The buyers can then download the files as they wish.
5. *User_Evil* Deletes the images from the cloud storage once the job is completed to erase the traces and possibility of detection.
6. Subsequently *User_Evil* terminates the CSU account.

4.4 Results and Analysis

We conducted extensive tests using the purpose built ownCloud experimental environment and collected results. In this section, we present, and analyze the results. Some of the results of the experiments are presented in Appendix - C, related to the use cases described in Section 4.3.4.1. Figure C.1 shows the result of an admin user login and creating (or deleting) CSU accounts, which is an equivalent action of CSUs creating account for themselves (or terminating their

own account) in a public cloud environment. Figure C.2 provides the result of *Use Case 1*. Figure C.3 provides the result of *Use Case 2*.

In this section, we analyze the results. First, we examine the use case test results listed in Appendix C against the forensics requirements listed in Section 4.3.1. Table 4.3 summarizes the analysis. The analytical summary cross-references to the cloud forensics logging requirements listed in the Table 4.1, to establish that the requirements have been met. Then, we provide further analysis against ACPO guidelines in Section 4.4.2, and finally with NISTIR 8006 in Section 4.4.3. This exercise aims to demonstrate the suitability of the framework for cloud forensics investigations.

4.4.1 Analysis of results against requirements

Table 4.3: Cloud forensics logging: Analysis of results.

Analysis	**Requirement Id**
All events are captured and recorded, and a chain of log entries are created, with full details of user actions and user details. The chain of log entries forms an auditable trail of event history. The entries are in reverse chronological order, i.e., the last event on the top of the file.	R-1, R-5
The events are repeatable, and the results are reproducible by following the same set of processes and test conditions.	R-2, R-3
Recording UTC helps time lining events or connect the events on a standard scale, regardless of the local time of the CSU or the actor. Reverse chronological order of the log entries helps identify the incidence first and then trace backward, as typically done in an investigation.	R-4, R-6, R-20
The parameters in every log entry record provide complete insight into 5W1H elements of an action, providing a rich source of information required for digital investigations.	R-4, R-5, R-8, R-17, R-19, R-20

Continued on next page

Analysis	Requirement Id
Recording user geo-location, source IP address, and user id, helps establish the provenance factor and location.	R-4, R-7, R-8
The LFPU design concept helps to separate the logs (evidence segregation) and also to acquire the log entries per CSU.	R-9, R-11, R-13, R-17
The CFLOG architecture, specifically the log structure and storage methods described in Section 4.3.2 increases the trust and confidence in the logs.	R-10, R-18
Having a well-defined and unified log structure, capturing all needed parameters helps to correlate the evidence, provided that CSPs adopt the framework.	R-12, R-15
The pre-defined location and persistence of the logs support evidence identification and keeping the logs in a multi-tenant, volatile cloud environment.	R-14, R-16
The extensibility feature of the logging framework allows further adding more parameters, and to enhance data sufficiency.	R-19
The log representation as a key-value pair in JSON format increases the usability helps to interpret the data efficiently	R-20

4.4.2 Analysis of results against ACPO guidelines

Recall that in Section 4.3.1, we noted the acceptability of the ACPO guidelines by the forensics practitioners and justified the use of the guidelines as one of the prime sources to gather cloud forensics logging requirements. Therefore, by validating this framework against the ACPO guidelines, the digital forensics community should be better placed to contextualize the arguments presented herein.

The ACPO guidelines is driven by its four core principles. The applicability of this framework is analyzed in the context of the core principles.

a) **Data integrity (Principle 1):** The ACPO Principle 1 states that:

No action taken by law enforcement agencies, person employed within those agencies or their agents should change data which may subseqently be relied upon in court [35].

The challenge presented by the first ACPO principle in a cloud investigation is to ensure the *integrity* of the data and the *chain of custody*. This framework addresses integrity aspects from multiple angles and hence the trustability of the evidence. First, the forensics logging application is part of the Hypervisor in the virtual machine stack. Second, no control or access to CSUs to the application or the logs, and event logging is mandatory. Third, the possibility for cross-validating the log hash sum against the initially computed hash sum. Finally, the logs are separated per CSU at the source. Commonly, in a public cloud platform, the event logging and storage is a CSU configurable service provided by CSP, meaning those logs are under the control of CSU, hence challenging to trust. In this model, the logs are in the CSP-controlled area of the cloud stack. We argue that this is a better proposition as CSPs can be audited, and the CSP provided services are better protected against cyber-attacks. Generally, CSPs are certified against various security and governance standards.

Chain of custody is the capability to document the entire digital evidence timeline chronologically, showing the seizure, custody, control, transfer, analysis, and disposition of the digital evidence [73], and it is a procedural factor. Therefore, it is beyond the remit of this study.

b) **Competency (Principle 2):** The second principle of ACPO guidelines states that:

In circumstances where a person finds it necessary to access original data held on a computer or on storage media, that person must be competent to do so and be able to give evidence explaining the relevance and the implications of their actions [35, 95].

Competency in this context refers to skills and knowledge of the inves-

tigative authority required to handle, investigate and present the evidence findings [95]. In our proposed framework, the architecture of the framework and the descriptive logging of evidence information as described in Section 4.3.2 and further explained in Table 4.2 and Figure 4.2, does not demand any technical expertise in the field of cloud computing, thus supporting the ACPO Principle 2.

c) **Audit Trail (Principle 3):** Principle 3 of the ACPO guidelines states that:

An audit trail or other record of all processes applied to computer based electronic evidence should be created and preserved. An independent third party should be able to examine those processes and achieve the same result. [35, 95].

The CFLOG framework supports Principle 3 as it records and stores all the CSU activities sequentially as a chain of events in a set of log files as a mandatory task. Our use cases and experiments demonstrated that repeating the process produces the same result. The flexibility and scalability feature of the framework allows for adding more parameters to the log structure if the requirement or law changes and more information needs to be captured. Sub-clause 3.5 of the guidelines recommend that mobile phone data collection is preferable to collect the call logs from the communication service provider rather than requesting the forensics examination of the mobile phones. Extrapolating the same principle to cloud forensics, we can say that it is preferable to collect the logs from the CSPs, rather than seizing the client end-point devices. This log framework enables the CSPs to retrieve logs associated with CSUs, and present them for any auditable purpose.

Principle 4 of the ACPO guideline deals with the organizational and procedural element of the investigation, and therefore, it is not within the preview of this study. However, further arguments in support of our framework con-

cerning the guidelines are presented below.

d) **Provenance**: Provenance is the science of associating a piece of evidence to a suspect or potential criminal. It is imperative in any investigative forensics, let alone in digital forensics. For example, criminal investigators try to capture fingerprints or blood traces from a crime scene to establish provenance. Digital evidence has no difference either. In this regard, quoting Section 5.10.3 of the ACPO guidelines states:
Establishing the provenance of digital evidence is another key task of the forensics practitioner, who must use their knowledge and skills to identify not just that the evidence exists but also how it came to be there.....It is the responsibility of the practitioner to carry out analysis to identify provenance where necessary, to mitigate the risk of their findings being misinterpreted.
This framework supports the provenance requirements too. As described in the CFLOG architecture described in Section 4.3.2 each log entry captures the key parameters, such as source CSU id, ip address, date, time, action, and the geographic location. All of them help to establish the provenance, and helps to attribute an action to the doer.

e) **Evidence interpretation**: Section 5.10.5 of the ACPO guidelines caution that:
the practitioners to be careful while stepping out of their knowledge boundary and suggests seeking the expertise of additional specialists when necessary.
The log data is presented logically and structured as a key-value pair, making the framework self-explanatory and straightforward. Therefore analyzing, interpreting, and reporting the facts collected from the logs is relatively easy, poses the least risk of the findings being misinterpreted, and does not require expertise in cloud computing or digital forensics.

f) **Evidence acquisition**: The framework further helps to support Section 4.3.2 of the guidelines that deals with evidence seizure which states that:

...The person in charge of the search must have reasonable ground to remove property and there must be justifiable reasons for doing so....

The framework mandates creating logs per CSU basis, i.e., $LFPUs$. Hence, the logs are not co-mingled. Therefore, the files do not require any processing while making the logs available to the investigators. Further, as per the framework, the log files are stored in a pre-configured location, making evidence identification and acquisition a seamless task. Also, the log parameters in our proposed CFLOG structure help the investigators arrive at sound and justifiable reasoning regarding further actions related to more evidence acquisition. For example, if the CFLOG output indicates that a program is being executed, that automates sending spam or phishing emails. Then, it is justifiable evidence to acquire the mail server logs.

g) **Inter-operability**: Lack of interoperability, including the non-existence of standard format for logging, between CSPs has been cited as one of the significant cloud forensics challenges by previous researchers and one of the central cloud forensics challenges identified by NIST [36]. The lack of interoperability between CSPs further complicates factors such as evidence correlation too. The logging framework that we proposed in CFLOG helps to address this problem. Suppose the service providers use the proposed format and principles of logging. In that case, it will simplify collating and unifying logs from different providers making it much easier to correlate the evidence.

4.4.3 Analysis of results against NISTIR 8006

Recall that in Section 4.3.1 we mentioned that NISTIR 8006 document identifies and summarizes the cloud computing forensics science challenges and is also used as a prime source draw the CFLOG framework requirements. This section further validates the framework against those forensics challenges related to logs. More detailed validation against the NIST forensics challenges are listed in Table

4.4. By doing so, we believe that the suitability of the framework as a practical forensics logging framework will be further strengthened.

a) **Deletion of objects in the cloud:** Attributing deleted data to a specific user in cloud based storage, and recovery of deleted data is one of the most discussed challenges. In the cloud systems, once a file is deleted the metadata is also deleted. Once the metadata is deleted no one can know who owned the file [131]. Though the logging services provided by CSPs do log file deletion event. However, the logging services are a CSU configurable item, which cannot be expected from a malicious actor. In any case, the logs will be lost once the client terminates the account entirely. In our proposed CFLOG framework, all such activities are logged mandatory and persisted, thereby providing a vital link to the forensics investigation.

b) **Log format unification:** As different CSPs have different architectures, they often differ in the event logging format. Suppose the providers accept the proposed uniform structure can significantly reduce the overhead when the logs are collated and filtered looking for forensics threads. In addition, following a uniform logging structure would also help to ease another challenges, i.e., *Interoperability issues among providers* in the logging space and evidence correlation across multiple CSPs.

c) **Time line analysis of logs:** Having time recorded in UTC format, as specified in the Section 4.3.2 helps to connect the chain of events and data correlation along a uniform timeline.

d) **Detection of the malicious act:** Attacks on the computer systems are carried out in incremental steps. Each step exploits a minor vulnerability and can quickly go unnoticed until the attacker penetrates the cloud and a significant system compromise happens. By logging every activity at Hypervisor stack sequentially, malicious actors cannot reach the logs, i.e., the LFs are protected. Analyzing it systematically and routinely would help

find any suspicious activity earlier, leading to the activation of incidence response triage.

e) **Evidence segregation and Selective data acquisition:** Traditionally, logs collect all the CSUs' activity over a period of time in one (or set of) file(s). Like in AWS CloudTrail [141], or Azure activity logs [124] all the users within a domain account are collected in a set of log files. When investigators want to trace the activity related to a specific person requires filtering and processing of the log files to extract the evidence of interest. Our model recommends having log files per user ($LFPU$) hence partially resolving the evidence segregation issue during the evidence acquisition phase. The framework also helps reduce the overall data set that the investigator is interested in, narrows down to the richest sources of information and perform selective data acquisition related to a user or set of users.

f) **Locating evidence and e-discovery:** Evidence collection is often infeasible in the cloud, as the specific location of evidence is unknown. The proposed log framework can write the log output to a predefined location, which can be made known only to the investigator on demand, further easing the e-discovery process and the dependency on cloud service providers.

g) **Service level agreement (SLA):** Lack of forensics related terms in SLA has been cited as cloud forensics challenge [4, 36]. Our log framework proposes a mandatory recording of user actions, despite the existence of forensics friendly SLA.

4.5 Comparison Against other Frameworks

This Section compares our proposed CFLOG framework, described in Section 4.3, against three other frameworks, i.e., ForFW [61], TamFor [72], and BCFL [58], in terms of their compliance to NIST cloud forensics challenges [36]. To see

progression of works in CFLOG framework, we have selected ForFw to represent an earlier work than ours, and TamFor and BCFL for more recent works. Briefly, The frameworks used for comparison are (i) ForFw: forensics framework for cloud computing presented by Alex *et al.* [61], (ii) TamFor: A tamper-proof cloud forensics framework presented by F Ye *et al.* [72] and (iii) BCFL: An approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem presented by Asuson-David *et al.* [58]. The ForFw highlights the concept of using a centralized forensics server and a forensics layer called forensics monitoring plane (FMP) running outside cloud infrastructure for collecting and storing events happening within the cloud. Investigators can access the forensics server as and when required without the need for CSP support [61]. The main contribution of TamForen is the proposal for a new distributed tamper-proof framework, which can be applied to untrusted cloud environments, i.e., where neither the CSPs nor CSUs are trusted [72]. The significance of the BCFL framework is the use of Blockchain distributed ledger technology to ensure trustworthiness, integrity, authenticity, and non-repudiation of log evidence in the cloud [72]. The novelty of the CFLOG framework is the business-centric approach, and the practical application for cloud forensics investigations. We used the NIST cloud forensics challenges listed in the NISTIR 8006 as the key for comparison. The NIST document lists 65 challenges that spans across all dimensions i.e. technical, organizational, and legal (explained in Figure 2.2) of cloud forensics. However, for this comparative study, we are using only those challenges categorized as technical. The comparative study is summarized in the Table 4.4.

Table 4.4: Framework comparison summary.

NIST Challenge	ForFw[61]	TamFor[72]	BCFL[58]	CFLOG[146]	Comments
FC-01: Deletion in the cloud. Recovering data deleted from the cloud and attributing that data to a specific user.	× Evidence of deletion is captured and saved in forensics server.	× Evidence of deletion is captured, if it is logged by client application.	× Allows recovering evidence deleted from cloud.	× Evidence of deletion is captured mandatory.	None provides a solution to this challenge, but the CFLOG captures the evidence of deletion and persists. CFLOG allows attributing the deletion action to the doer, and also records the previous owner of the file.
FC-02: Recovering overwritten data by another tenant in a shared virtual environment.	×	×	×	×	
FC-03: Evidence correlation across CSPs (lack of interoperability).	×	×	×	✓	CFLOG supports evidence correlation provided that CSPs adopts the log format specified in the framework.
FC-04:Reconstructing virtual storage from physical disk.	×	×	×	×	

Continued on next page

NIST Challenge	ForFw[61]	TamFor[72]	BCFL[58]	CFLOG[146]	Comments
FC-05:Timestamp synchronization. The ability to map record of events along a timeline and to recreate the crime scene.	×	×	✓ Blockchain timestamps every transaction and is part of transaction header.	✓ Data logged with UTC timestamp, irrespective of geographic location of end points.	In BCFL Blockchain maintains immutable time stamp, but did not specify the timestamp is local or universal. In CFLOG UTC time stamping allows evidence timelining along a uniform scale.
FC-06:Log format unification. No uniform log structure, different CSPs uses diverse or proprietary log structure.	×	×	×	✓	CFLOG helps to resolve interoperability issues among CSPs in logging space, by providing a uniform forensics logging structure.
FC-07:Use of metadata. To have access to persistent metadata files as needed.	✓ Captures events and associated metadata.	×	×	✓	CFLOG's forensics rich parameter set includes metadata associated with all user actions. CFLOG offers scalability and flexibility, easier to add additional parameters.

Continued on next page

NIST Challenge	ForFw[61]	TamFor[72]	BCFL[58]	CFLOG[146]	Comments
FC-08:Log capture. Capture and timeline analysis of logs.	✓	✓	✓	✓	All the frameworks captures log data, but only CFLOG provides enough parameters to do timeline analysis.
FC-09: No interoperability among CSPs. Lack of interoperable forensics techniques and tools. [2]	×	×	×	✓ Adaptable forensics log architecture.	CFLOG supports interoperability in forensics logging and evidence representation.
FC-11: No single point of failure for criminals. Lack of information for investigators to link all accounts of the same user on different clouds.	×	×	×	×	CFLOG's geographic location can partially helps to link user events.
FC-12: Detection of the malicious act. Attacks are typically carried out through sequence of incremental steps.	✓	×	×	✓	ForFw and CFLOG allows continuous monitoring and trigger alerts for suspicious activities.
FC-13: The cloud offers low cost computing power and resources to criminals.	×	×	×	×	

Continued on next page

[2]FC-10, FC-20:Considered obsolete and removed by NIST

NIST Challenge	ForFw[61]	TamFor[72]	BCFL[58]	CFLOG[146]	Comments
FC-14: Real-time investigation intelligence processes not possible.	×	×	×	×	
FC-15: Malicious code may circumvent VM isolation.	×	×	×	×	
FC-16: Configuration errors in cloud management portals, can be used by an unauthorized user to exploit.	×	×	×	×	
FC-17: Geo-location unknowns can impact the chain of custody, finding/ locating evidence.	×	×	×	✓	CFLOG captures the geo-location of the system where the crime event has been initiated.
FC-18:Lack of transparency in cloud's operational details (architecture and implementation) triggers lack of trust and difficulties of auditing.	✓ Architecture transparency in forensics data storage and access.	×	✓ Uses known Blockchain technology.	✓ Provides transparent forensics log architecture.	None provides transparency in cloud architecture. The ForFw exposes data storage and access, BCFL uses known distributed ledger technology, and CFLOG provides full forensics logging architecture transparency.

Continued on next page

NIST Challenge	ForFw[61]	TamFor[72]	BCFL[58]	CFLOG[146]	Comments
FC-19:Criminals can hide in the cloud and maintain small cells of operation with no one knowing the identity of others.	×	×	×	×	
FC-21: Segregation of potential evidence in a multi-tenant system.	×	×	×	✓	A key feature of CFLOG is the evidence separation at source per tenant, which enables the easy acquisition of evidence of one tenant without breaching the confidentiality of others.
FC-22: Due to elastic nature of cloud, system boundaries are often difficult to define.	×	×	×	×	
FC-23: Secure provenance, i.e.,establishing the chronology of ownership, custody, or location of data.	×	✓ Monitors the provenance data generation, transmission, and storage.	×	✓	CFLOG captures the 'who' attribute and chronologically records the owner of an action, custody, and location of the data.

Continued on next page

NIST Challenge	ForFw[61]	TamFor[72]	BCFL[58]	CFLOG[146]	Comments
FC-25:Decreased access and control of data or no knowledge about the physical location of forensics data. [3]	✓	×	×	✓	ForFw and CFLOG exposes evidence data location and makes it available for investigators.
FC-26:Chain of dependencies (or inter-dependencies) in multiple cloud systems.	×	×	×	×	
FC-27:Locating evidence in a large and changing system, making e-discovery difficult.	✓	×	×	✓	Both ForFw and CFLOG logs the evidence to a known location, easing e-discovery.
FC-28:Data location. Uncertainties in in dealing with transparency in the cloud and distribution boundaries for retrieval.	×	×	×	✓	CFLOG partially satisfies this challenge as it specifies the geo-location of the incident.
FC-29:Imaging, isolating, and collecting data.	×	×	×	×	
FC-30:Data available for a limited time (volatility).	×	×	×	×	None persists the source data, but ForFw and CFLOG persists the evidence logs.

Continued on next page

[3]FC-24:Chain of custody is more procedural, therefore not included

NIST Challenge	ForFw[61]	TamFor[72]	BCFL[58]	CFLOG[146]	Comments
FC-31:Identifying storage media evidence may be found.	×	×	×	×	
FC-32:Evidence identification. Sources/traces of evidence are generated differently compared to non-cloud.	✓	×	×	✓ Generates forensically valid data.	ForFw and CFLOG clearly specifies the location of evidence data.
FC-33:Dynamic Storage.	×	×	×	×	
FC-34:Live forensics.	×	×	×	×	
FC-35:Resource abstraction.	×	×	×	×	
FC-36:Application details are unavailable.	×	×	×	×	
FC-37:Additional evidence collection is often infeasible.	×	×	×	×	
FC-38:Imaging all evidence in the cloud is impractical.	×	×	×	×	
FC-39:Selective data acquisition - some prior knowledge to reduce the overall dataset.	×	×	×	✓	CFLOG supports selective data acquisition per user basis, helping to reduce the overall data set.
FC-40:Cryptographic key management.	×	×	×	×	

Continued on next page

NIST Challenge	ForFw[61]	TamFor[72]	BCFL[58]	CFLOG[146]	Comments
FC-41:Ambiguous trust boundaries between users can cause questionable data integrity.	×	×	×	×	
FC-42:Data integrity and evidence preservation. Shared responsibility in the cloud, chances for failure are higher.	✓	✓	✓	✓	All the four frameworks supports data integrity and evidence preservation, but uses different methodologies.
FC-43:Root of trust: Multiple layers of abstraction in cloud, the trustworthiness of data is dependent on the cumulative trust of all the layers.	×	×	×	×	
FC-44:Competence and trustworthiness of the CSPs as an effective, immediate first responder. [4]	×	×	×	✓	Only CFLOG generated data does not require any technical competence to analyse and interpret.

[4]Challenges FC-45 to FC-65 does not belong to technical dimension

4.6 Chapter Summary

Logs detailing the CSU actions and events are an essential part of digital forensics, and are even more critical for conducting digital forensics in cloud computing. Computing systems, in general, produce numerous logs. Collection and analysis of the logs is an enormous and challenging task, especially in a cloud computing environment. CSPs are the custodian of the data assets, and it is challenging and often impossible for investigators to seize the evidence from a multi-tenant cloud computing platform. Jurisdictional location of evidence data and complexities in evidence data access poses an additional challenge. Therefore, often investigators have to depend upon CSPs for evidence acquisition. Analysis of the logs is similar to finding a needle in a haystack, requiring specialist tools and expertise to filter out the noise, extract the relevant data, and connect the chain of events. Any modifications to the original log data can undermine the legal validity of the evidence too. Also, the logs are not very useful if they don't contain information that satisfies the legal needs and is valid in the law's eyes.

Much research work has been published in cloud logging, primarily in security, data transporting and maintaining integrity and confidentiality of logs. Unfortunately, none of them addressed the needs of forensics investigators. In this chapter, we analyzed the requirements of the forensics practitioners and proposed a logging framework addressing the practitioners' needs from a business angle. We used the ACPO guidelines, NIST documents, and relevant ISO/IEC standards as the primary source for assessing business needs, and governance rules. We designed and built the application, validated the framework on our ownCloud platform.

Further validation of the framework has been carried out using two use case examples and demonstrating the results validated against the requirements. In addition, we further validated the framework against ACPO guidelines and applicable challenges listed in NISTIR 8006 publication on cloud computing forensics

challenges. Thus, we established the validity and suitability of the framework with adequate reasoning and test case demonstrations. We also found out that the extra services that need to be provisioned by CSPs to accommodate the framework in their environment does not add any significant overhead to the CSPs.

To highlight the significant and vital contributions of our proposed forensics logging framework, we presented a comparative study with three other recent frameworks, using the NIST identified challenges as the key. The study shows that the CFLOG framework addresses the maximum number of challenges, emphasizing the relevance, and relative merit of our work.

In summary, the framework enables the following forensics activities:

- To re-create the events.
- To trace the chain of events and build a corroborative evidence set.
- To easily attribute an action to the doer.
- To clearly identify and separate the logs per CSU.
- To retain the logs, even after the CSU account has been terminated.
- To acquire the logs without affecting other consumers.
- To establish trust and confidence in logs to a significant level.
- To easily interpret the logs.

Therefore, we believe that the framework helps the forensics examiners establish confidence in the log artifacts, enable user-friendly evidence presentation and straightforward interpretation. In addition, it supports building a collective chain of evidence and further helps the CSPs to provision forensics-enabled logging.

Chapter 5

A Method to Assess Forensics Readiness of Cloud Computing

As highlighted in Chapter 1, digital forensics includes the technology, processes, and methods which is carried out as a post-crime activity to identify the culprit responsible for the crime. For every action that has been carried out on a computing platform, an event trace is left behind. When an event can be of value for a digital investigation, it is considered a piece of digital evidence or simply referred to as evidence. Evidence describes 5W1H elements of a crime incident. As already mentioned in Chapter 2, a collection of evidence contributing directly to a digital investigation is often referred to as forensics artifacts, or simply as artifacts. The forensics activity, therefore, requires credible evidence. Event logs are typically the primary source of evidence. Further, recall that in Chapter 4 we emphasized that the logs detailing user activities are a valuable and critical source of digital evidence. A comprehensive log management system serves different investigative and forensics purposes, mainly [179, 180]:

1) to use as a source of evidence in court.

2) to assist in reconstructing an attack.

3) to identify the relationships between events.

4) to detect abnormal system behaviors or user activity.

5) to study how the system was compromised.

Many research works carried out in this space established the importance of logging of events to support (a) cloud forensics, (b) security of evidence data (c) to ensure integrity and trust in the evidence, and (d) secure transportation of evidence [53, 63, 110]. However, none of the studies deals with the current level of forensics compliance in cloud computing.

This chapter presents a research study conducted on the forensics compliance of log services provided by three major cloud providers: Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). AWS, Azure, and GCP provide CloudTrail logs, Azure Activity logs, and Stackdriver logs, respectively, as their prime log services. Our study aims to look at the **quality, quantity, and availability** of the log data, which can be used as an evidence by directly contributing to the digital forensics investigations. We then find out the gap between what is required *(i.e., the requirements)* for an investigation and what is available in the logs *i.e., the gap*. We used live accounts with the CSPs, subscribed to their services, conducted experiments, and collected results and data. We analyzed the results to determine the fitness or suitability of the log data for digital forensics needs. The degree of fitness determines the forensics compliance, i.e., the more the evidence available in the log data meets the forensics needs, the better the compliance.

Our work in this chapter aims to help forensics investigators and cloud customers to understand the level of forensics compliance that the CSPs offer. The novelty and main contributions of this work are as follows.

1. It presents a systematic methodology to assess the cloud forensics readiness;

2. It methodically evaluates the cloud forensics readiness of three major CSPs, viz., AWS, Azure, and GCP;

3. It systematically presents the forensics compliance of the CSPs;

4. It describes where to find the evidence and how to extract the evidence from the log archives;

5. It identifies and quantifies the gaps in forensics compliance;

6. It provides a comprehensive analysis of the state of forensics readiness and maturity levels of the respective cloud computing platform, based on the providers' prime log services.

For correctness, all our specific findings have been cross-checked with the respective service provider.

It is to be noted that a significant part of the experimental study was conducted between April 2018 and February 2019. Therefore, the conclusions are based on the results of our study during the particular study period. However, following the systematic methodology explained in Section 5.2, the tests could be repeated any time to assess compliance at a given time. Major parts of this chapter are available in the author's accepted journal article [173].

The layout of this chapter is as follows: Section 5.1 related work conducted in this field and the motivation to do this research project. Section 5.3 details the methodology. Section 5.2 describes the problem and provides mathematical modelling of the problem. The case study and details of the use cases are described in Section 5.4. Section 5.5 further elaborates on the forensics requirements. Sections 5.6, 5.7 and 5.8 describe the configuration, use case scenario testing, test results and compliance summary of AWS, Azure and GCP platforms respectively. Section 5.9 summarizes the contributions of the chapter. Appendix A lists the forensics artifacts identified in the logs of AWS, Azure and GCP and maturity level of the CSPs. Appendix B provides a snapshot preview of the resulting logs produced by the CSPs.

5.1 Motivation

Digital evidence can be either *persistent*, i.e., the evidence is available in persistent storage, or *non-persistent*, i.e., the evidence is in volatile memory that has an only minimal lifetime. Thus, the evidence in the non-persistent storage must be collected using live forensics methods. One way to meet the forensics requirement in cloud computing is to have robust and trustworthy evidence logging and to persist the evidence logs. To this end, many researchers have established the importance and suggested various solutions to have some form of an audit trail as a valuable source of digital evidence [55, 63, 181] . Also, researchers have highlighted the importance of the quality, quantity, availability, and trustability of the evidence [181–183].

Reproducibility of evidence and event reconstruction by an independent authority is an important aspect of any digital forensics investigation [184]. This is more so in a cloud computing scenario, because of the cloud's dynamic nature and multi-tenancy support. Moreover, digital forensics in the cloud requires a proactive approach, expecting any future litigation in advance [70].

A logging system will not be comprehensive without addressing the needs of the practitioners. To this extend, Pichan *et al.* [146] detailed a business oriented logging frame work by addressing the evidence recording needs from a practitioner's point of view. Providing tamper resistant audit trails are essential for forensics. Researchers have also proposed creating tamper proof evidence logs based on Blockchain distributed ledger technology [58, 185]. Further, a formal method for event reconstruction using process algebra enabling digital forensics investigation has been laid out by other researchers recently [184].

Other works also have been done in the field of evidence logging addressing the forensics needs, such as (i) a secure and tamper proof logging system by storing the entire virtual machine logs and making them available on a secure platform to the investigators [54, 63]; (ii) a method for identifying and extracting

forensics related log entries primarily from Linux environments [133]; and (iii) a novel way of monitoring activity in cloud environments, by presenting a layered cloud logging architecture [53] are few important contributions to note.

Many researchers have put forward various cloud computing logging architectures to capture and preserve the logs as forensics evidence [53, 63, 110]. While some of the previous works mentioned that the cloud forensics is not mature yet, neither CSPs nor CSUs have put forward an applicable architecture to make the cloud computing forensics ready [4, 47, 118]. Regardless it is critical to generate, store and provide legally acceptable logs as evidence to support cloud forensics and to create a time line of events [105, 180].

Recall that in Section 4.1 we described some related work on the forensics logging space. However, none of the work addresses the forensic readiness or compliance of the CSPs. AWS, Azure, and GCP hold most of the market share in the public cloud space, and acquiring paid or un-paid (free tier) services from them is effortless. One can expect that anyone with malicious intent would acquire computing services and carry out malicious activities. The malicious activity must be traceable. Therefore, this study aims to determine the forensics readiness of the cloud platforms, i.e., where are we now, concerning cloud forensics maturity. We used the log services and the evidence available in the logs to evaluate forensics readiness and compliance to meet this objective.

5.2 Problem Description and Modeling

Note that in Section 4.3.1 we detailed the cloud forensics logging requirements. This Section model the problem using a crime use case and digital forensics requirements.

5.2.1 Problem Description

Digital investigation, conducted to prove a digital crime and to bring the culprits to justice, must follow a set of pre-defined principles, established processes, and

guidelines [35, 122]. In general, the requirements describe the properties of the digital evidence, such that the crime execution process is evident and the crime can be investigated. It is to be noted that every user action creates one or more events. In general, these events are recorded in the event logs. Events that satisfy one or more forensics requirements form the evidence. Therefore, a digital investigative problem can be specified as (i) being able to identify and locate the evidence data (ii) being able to collect the evidence in its original form (iii) being able to establish the immutability and trust factor in the evidence (iv) being able to timeline the evidence and produce a narrative history of events. In the following parts of this chapter, we demonstrate whether the evidence of a crime incident can address the above problem using use case scenarios.

5.2.2 Problem Model

Let C_i denote a crime use case i, and $R_i = \{r_{i,1}, r_{i,2}, ...r_{i,|Ri|}\}$ denote a set of $|R_i|$ forensics requirements for use case C_i. More specifically, $r_{i,k}$, is the k^{th} requirement for C_i. Let CSP_x be a cloud service provider x, and $S_x = \{S_{x,i}, S_{x,j}, ...S_{x,l}\}$ denote a set of digital forensics service type or service level provided by CSP_x, for $1 \leq l \leq |S_x|$. Note that some CSP provides only one service type, i.e., $|S_x| = 1$, while some others, may provide more than one type, i.e., $|S_x| > 1$. We use $F_{x,l,i}$ to denote a set of forensics artifacts that can be generated from $S_{x,l}$ for case C_i.

To execute a use case scenario requires a set of actions to be performed, generating events. For example, a user action of accessing a file $(i.e., object)$ from a cloud store and downloading it results in generating an event where the state of an object changes from 'Read' to 'Copy'. Thus, for each case i, we have $C_i = \{C_{i,a}, C_{i,b}....C_{i,n}\}$, where $a, b, ..., n$ are user actions which produces events. Evidence of such events contribute to the forensics artifacts which can be used to prove a digital crime and connect the actions to a doer.

We aim to asses if one can produce artefact $F_{x,l,i}$ from service $S_{x,l}$ for case C_i that satisfies the forensics requirement R_i. We assess whether each $C_{i,a}$ produces

one or more evidence(s). Therefore, we have $F_{x,l,i} = \{F_{x,l,i1}, F_{x,l,i2}....F_{x,l,in}\}$, where $F_{x,l,i1}$ is a forensics artefact generated for an action $C_{i,a}$ which can satisfy one or more requirements. We use $F_{x,l,i} \subset R_i$ to indicate that the artifacts generated by the service type l of CSP_x are not fully forensics compliant or satisfies the forensics requirement for the use case i. For this case, we aim to identify all the insufficient artifacts, required to satisfy a set of requirements, producing the *gap*. On the other hand, $F_{x,l,i} \supseteq R_i$ indicates that the artifacts generated for the service l fully meets the requirement R_i (i.e., there is no *gap*). This project also aims to identify and group the common or core set of requirements among different use cases. For example, we use $R_i \cap R_j$ to denote common requirements of use cases C_i and C_j. Note that when $R_i \cap R_j = \{\}$, there is no common requirement between use cases C_i and C_j. However, we envisage that $R_i \cap R_j \neq \{\}$.

5.3 Methodology

We applied the following methodology for this work.

1. Describe the case study and details of its use case(s).

2. List and describe the set of forensic requirements, including use case-specific requirements.

3. Source the required cloud computing resources and services from public cloud providers. For this study, we sourced services from AWS, Azure, and GCP.

4. For each CSP (e.g: CSP_x), identify and describe its provided service type and level (i.e., $S_{x,l}$).

5. For a given $S_{x,l}$, produce $F_{x,l,i}$ for use case C_i. Conduct experiments and perform validation tests using the use case scenarios for this step.

6. For each $F_{x,l,i}$, analyze it against its corresponding forensics requirement R_i, i.e., $F_{x,l,i} \subset R_i$, or $F_{x,l,i} \supset R_i$, or $F_{x,l,i} = R_i$. If $F_{x,l,i} \supseteq R_i$ then we consider the evidence available in the log meets the forensics requirements.

7. Assess the gap when $F_{x,l,i} \subset R_i$, and quantify the gap. The gap has been quantified (in a scale of 1 to 10) in terms of *high* (H), *medium* (M) and *low* (L) as explained below:

    - *High* (a value from 8 to 10): The gap is considered *High* when a given or a set of forensics requirement(s) are not at all met by the evidence available in the log(s) and requires significant time and effort to make the log(s) compliant.
    - *Medium* (a value from 4 to 7): The gap is considered *Medium* when a given or a set of forensics requirement(s) are partially met by the evidence available in the log(s) but requires reasonable time and effort to build the missing evidence and make it forensics compliant
    - *Low* (a value from 1 to 3) The gap is considered *Low* when a given or a set of forensics requirement(s) are partially met by the evidence available in the log(s), but requires only limited effort to make the log(s) compliant

5.4 Case Study

This Section describes a crime use case. The use case scenarios and the requirements drive the testing process, and eventually, we trace and find out the forensics artifacts left in the logs after every process step.

5.4.1 Use Case: Data Leakage (insider attack)

We used the following use case to measure the forensics compliance of logs provided by the CSPs. *This use case has been adapted from NIST Computer foren-*

sics Tool Testing Program Current Data Sets [186]. We used insider use cases primarily for multiple reasons. First, given that our objective is to find out the cloud forensic readiness of the public cloud platform, an insider with a cloud user account and service access credentials can conduct experiments and collect artifacts more efficiently. Secondly, insider attack is a well-known security issue in the outsourced cloud services [38]. (For example, a malicious insider launched a denial of service attack against an IaaS cloud platform. The attack was carried out by spinning up VM instances iteratively, growing and consuming resources exponentially, finally bringing down all services [38]). So, by simulating an insider attack, one can find out the depth and breadth of forensic artifacts left in the cloud. Thirdly, though with good intentions, hacking public cloud services is against the law.

Scenario overview: "Mr. Informant" was working as a middle level manager of the technology development division at a famous international company OOO which developed state-of-the-art technologies and gadgets. Mr. Informant joined the organization as a Senior Design specialist and moved up the ladder. The organization OOO uses cloud services for all of its IT needs, including storage, database services, and e-mail. The cloud-computing environment has been built and configured as per the organizational requirements and policy. Mr. Informant has the necessary access to the application, data and storage, including the access to search, read, copy, edit and delete objects, but not the access rights to modify the access control list or permission settings.

One day, Mr. Informant received an offer from "Spy Conspirator" to leak sensitive information related to the newest technology being developed by his team at OOO. Actually, Ms. Conspirator is an employee of a rival company, and Mr. Informant decided to accept her offer for an undisclosed reward and began establishing a detailed leakage plan.

Mr. Informant made a deliberate effort to hide the leakage plan. He did so, using his credentials and whatever log trails he could delete was deleted. Mr.

Informant and Ms. Conspirator agreed to use cloud based e-mail services, like a normal business relationship, for part of the data leakage. To avoid suspicion Mr. Informant started delivering the confidential files partially via cloud mail services and partially downloading files to a local removable storage. Eventually, Mr. Informant tried to take his storage devices away, but he and his devices were detected at the security checkpoint of the company, and he was suspected of leaking the company data.

Subsequently, the organization executive calls for a forensics investigation, upon realizing that highly confidential data, including intellectual property has been leaked.

The flowchart depicted in Figure 5.1 illustrates the use case workflow described in Section 5.4.1. We considered ten actions, starting from user sign-in to the system and finishing with a user sign-out action. Typical actions done to steal the data are (i) object search - to find out the data of interest (ii) object operations - to copy the data of value, or to delete the log traces (iii) storage service operations - to export the data from storage buckets, and similar operations with SQL/NoSQL databases as well with other steps described in the flow chart. Though the actions are shown sequentially, it can be done in any order and repeatedly. After every action, we check whether the action is recorded in the event log. If it is recorded, the compliance level is checked by assessing whether the evidence data in the log fully or partially meets the requirements. If the data meets the requirements, we consider it fully compliant, and if it meets only partially the requirements, we consider it only partially compliant. We then proceed to assess the gap. If there are no evidence data in the logs, or the events are not at all logged, then it is treated as non-compliant.

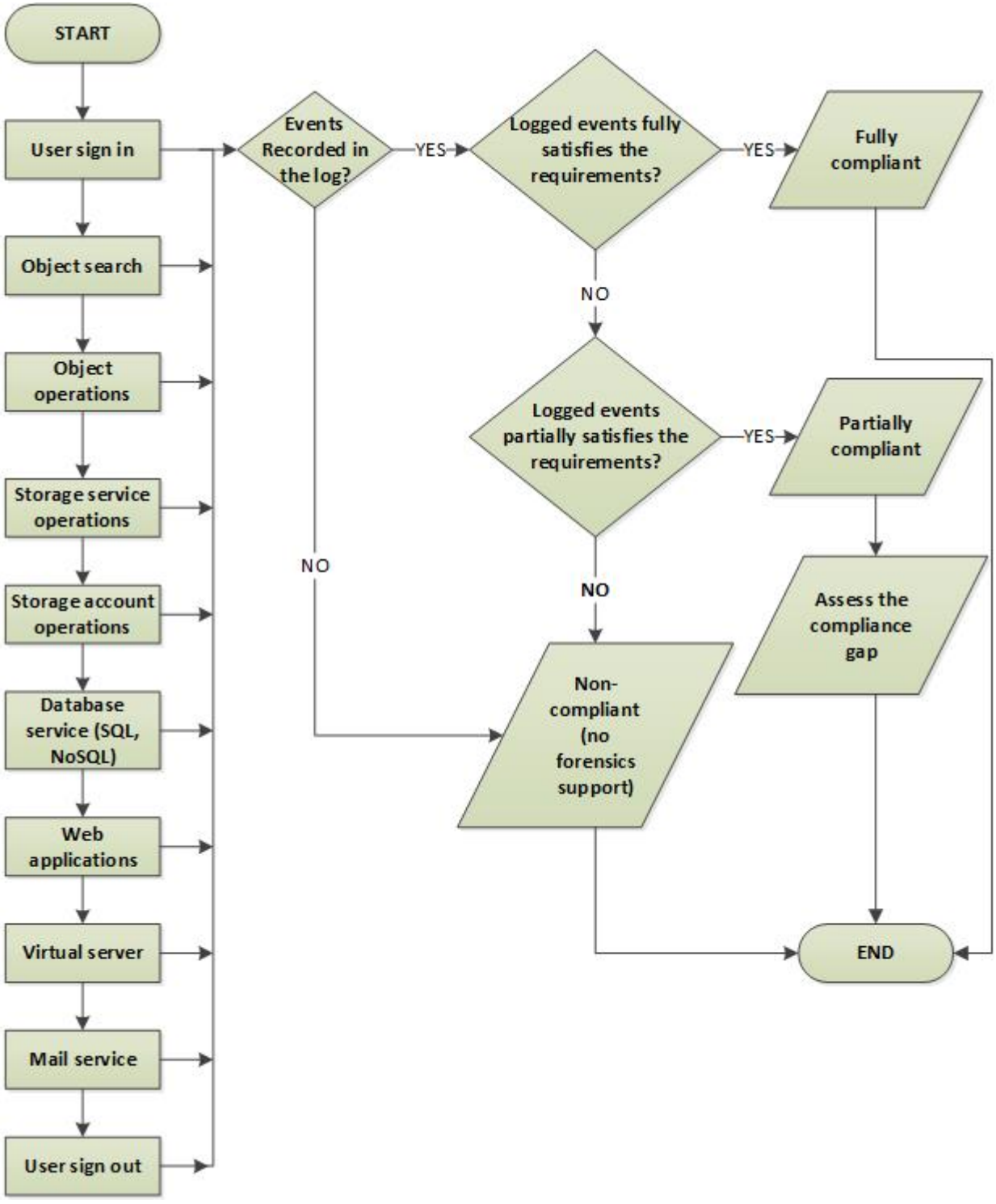


Figure 5.1: Use case execution flow

5.4.2 Assumptions

We subscribed and configured all the necessary cloud services to carry out the experiments as per the use case described in Section 5.4.1 The case study assumes the following:

- The insider has just enough access to steal data that is of significance.
- The Logs analyzed for evidence tracking, and identification is the primary log services, i.e., AWS CloudTrail logs, Azure Activity logs, and Google Stackdriver logs. These are the prime logging services provided by Amazon Web Services (AWS), Microsoft, and Google Cloud Platform (GCP), respectively.
- Only the products or services which are part of the cloud service offerings by the three CSPs are used for the study. In general, no third-party products or add-on services form part of this study. However, if a third-party product has been used, it is identified and specified accordingly.

5.5 Forensics Requirements

The requirements for digital forensics remain the same, irrespective of the computer system environment using which the crime has been conducted. As already noted any forensics investigation seeks to answer six key fundamental questions, referred as 5W1H of an incident [67]. Therefore, we are looking for evidence to answer these key questions in the logs. Recall that in Section 4.3.1, we have identified and described the forensics logging requirements. Here, we are consolidating and elaborating the requirements in two groups:

1. Core Requirements. These are the same requirements listed in Table 4.1. We refer them core requirements here as they are independent of the platform or use cases and address common investigative needs.

2. Use case specific requirements. These are the set of requirements addressing the scenario explained in the use case in Section 5.4.1 to support forensics. We derive these requirements in Section 5.5.1.

5.5.1 Use Case Specific Requirements

Note that in Section 5.5 we mentioned the fundamental questions that any forensics investigation seeks to answer. Next, we derive the use case specific requirements to answer these essential investigative needs in light of the use case described in Section 5.4.1. The requirements are listed in the Table 5.1. (Note: The term Object refers to a file or folder which may hold a piece of data of forensic interest)

Table 5.1: Use case specific requirements.

Id	Title	Description
UCR-1	User audit	The ability to establish when the CSUs are active in the system, i.e., a record of user login/log out sessions.
UCR-2	Object search	The ability to find out evidence of actions such as searching thru objects, storage locations, and logs.
UCR-3	Object operation	The ability to find out evidence of actions such as copy, download, update, delete or create objects or similar activities.
UCR-4	Storage service operations	The ability to find out evidence related to storage service operations such as data export, data import, bulk copy, file purging, etc.
UCR-5	Storage account operations	The ability to find out evidence related to storage account operations such as storage area creation, deletion, updating, or storage account creation and deletion.
UCR-6	Database services	The ability to find out evidence of all database actions, both in SQL and No-SQL databases.

Continued on next page

Id	Title	Description
UCR-7	Applications	The ability to trace events related to application execution, including web applications.
UCR-8	Virtual server	The ability to find out events related to VM operations, such as VM instance creation, deletion, start, stop VM, etc.
UCR-9	Mail services	The ability to trace all events related to in-built mail service usage.

5.6 Amazon Web Services (AWS)

In this section we are evaluating AWS as the first CSP_x candidate and their CloudTrail logging service i.e., $S_{x,l}$. In AWS, the service CloudTrail provides a logging feature. CloudTrail records actions taken by a user, role, or an AWS service as events. The record includes those actions performed using AWS console, Command Line Interface (CLI), and AWS SDKs and logs of all the API calls performed on the AWS account. CloudTrail provides visibility into *'who or what took which action, the resources acted upon, the date and time, the source, and the event happened'* [187]. CloudTrail logs are JSON formatted files stored in AWS Simple Storage Service (S3) buckets. Some of the events are visible on the CloudTrail dashboard and under the event history tab. However, AWS does not display all the events in the CloudTrail dashboard or event history tab, even though the events are recorded in the CloudTrail logs. Therefore, specific programming scripts must be developed to filter and extract particular events of interest from the log repository.

5.6.1 AWS System Configuration

We sourced and configured the following AWS services for this study.

a) **S3 Configuration:** AWS S3 is the storage service. Customers can create 'bucket' to store their data. To simulate the use case scenarios, we created

two buckets to store

1. the CloudTrail logs, namely 'myreseachlogbucket'
2. the confidential data of the organization OOO, namely 'confidential-datafiles'.

The confidential data bucket has been configured to encrypt the data in storage using AWS server-side (AES 256) encryption. Additionally, we enabled the recording of digest of the log files in the CloudTrail-Digest folder for file integrity checks.

b) **CloudTrail configuration:** AWS allows configuring the CloudTrail in multiple ways. We used the most common approach, i.e., to receive log files from all regions and put the log files in one AWS S3 bucket, irrespective of the location of many CSUs, under the main account. Note that a "region" is the location the user chooses while creating an AWS account, where the data and applications are hosted. AWS supports many regions across the globe. The region acts as the endpoint of users' web service requests. AWS claims that the data stored in a bucket in one region, always remains in the same region. CloudTrail only logs management events by default, but we configured it to log both management and data events. Management events provide insight into management operations performed on resources, whereas data events provide full details of the object-level activity. We developed specific AWS scripts to search and filter the logs, and to find out specific events of forensics value.

c) **CloudTrail digest:** The digest file records the hashes and digital signatures of the log files, thereby providing a powerful mechanism to validate the integrity of the log files.

d) **CloudWatch:** CloudTrail, as such, does not provide any capability to search and filter the logs and visualize the results. In addition, the log

file sizes are enormous and contain a vast number of files. Therefore, we used CloudWatch as a monitoring tool to look for events of interest in the CloudTrail logs. CloudWatch also allows configuring alarms. We configured CloudWatch to create alarms in the event of significant activity (e.g. downloading file(s)) from the "confidentialdatafile" bucket or root log-in attempts. CloudWatch dashboard visualizes the events and sends the notification by e-mail.

e) **EC2 instance:** We created VM instances, otherwise known as Elastic Cloud Computing (EC2) in AWS terminology. OOO used EC2 instances to run some of its business applications and store data files.

f) **WorkMail:** AWS WorkMail is a business email and calendar service. We used the WorkMail service to mimic the user action of Mr. Informant sending part of the data via email and to capture the log artifacts that WorkMail produces.

g) **DynamoDB:** DynamoDB is a NoSQL database service provided by AWS, and OOO uses DynamoDB to store critical information assets. We configured DynamoDB to simulate the use case scenarios.

h) **WebApp**: AWS Elastic Beanstalk provides an environment for creating and deploying web applications. We created and deployed a web application using elastic beanstalk environment with MySQL as the database. The purpose was to find forensics threads in the CloudTrail logs when the application gets executed.

i) **Relational database service (RDS):** AWS provides RDS using various database engines. We configured RDS using MySQL database engine. We also created and configured databases and instances to store design data and various OOO's flagship product parameters.

j) **Users:** We created identity and access management (IAM) users under one

AWS main account and configured a few users with limited access rights as per the organizational policy, including an "Admin" account with broad privileges and an “i_informant” account, which belongs to “Mr. Informant”, with limited rights.

5.6.2 AWS Experimental Validation

We executed different scenarios of Data leakage (*Use Case i*) as described in Section 5.4.1 simulating the actions of *Mr. Informant* stealing the data. Being forensics investigation is a post crime activity, we walked thru as an investigator, looking for forensics artifacts in the AWS ($i.e., CSP_x$) CloudTrail ($i.e., S_{x,l}$) logs. The primary task is to find the answers to the six key 5W1H questions, and to find evidence satisfying the requirements listed in Section 5.5. Since CloudTrail logs is a collection of the events and actions of all the users, we used special script to extract the logs of activity happening at the "confidentialdatafiles" bucket. For example, the following script checks for the object-level activity of putting, deleting, and getting any objects from the container named "confidentialdatafiles" by anyone.

{(($.eventName = ”PutObject”) || ($.eventName = "DeleteObjects") || ($.eventName = ”GetObject”)) &&
($.requestParameters.bucketName = "confidentialdatafiles")}

The result of the execution with details pointing to the key forensics parameters in AWS CloudTrail log is listed in Figure B.1 of Appendix B. The sample log shows that a user "Admin" deposited an object by name "Product Design Specification.pdf" to bucket "confidentialdatafiles". Other relevant parameters such as event date and time (in UTC), the source IP address, the user responsible for the action, etc., are also recorded. As another example, the following scripts extract all the activities done by the user *i_informant* from the CloudTrail logs.

$\{(\$.userIdentity.type = "IAMUser") \ \&\& \ (\$.userIdentity.userName = "i\_informant")\}$

5.6.3 AWS Log Analysis

Following the use case scenario tests, we collected the log data, analyzed its conformance and compliance with the forensics needs, and verified whether those actions are traceable in the log. If the actions are registered in the CloudTrail logs, then the logs are analyzed for forensics compliance. When the actions are not traceable in the log, or if the information is only partially available, we identified that as a non-compliant gap. Then the magnitude of the gap is assessed and quantified. The summary of our findings is detailed in the following paragraphs.

User audit $(C_{i,1}), (C_{i,2})$

User audit events include *sign-in* $(C_{i,1})$ and *sign-out* $(C_{i,2})$ actions. We found that though sign-in events are registered in the CloudTrail, none of the sign-out events are recorded. This gives a perception that the user has been active in the system, whereas in reality it may not be the case. To establish authenticity and assurance in the findings there has to be clear evidence that the culprit was active in the system when the data leakage occured. The absence of sign-out events causes to create a void in the chain of event traceability and hence to fail some of the use case specific requirements listed Table 5.1 specifically UCR-1 (User audit).

Object search $(C_{i,3})$

One can expect that the culprit will do a file and folder search operation to find out the objects of interest first. Therefore, we investigated the traces related to the 'search' operation, such as directory search, file search, etc. Upon analysis, we found that all the events and evidence pertaining to object search are in the CloudTrail logs.

Object operation ($C_{i,4}$)

Object operation events include the events generated from user actions like delete, read, copy, download, put or create object actions. We found that all events related to object read, copy, and download are recorded with full details, providing full forensics traceability, but not for the delete operation. We found that for ***object delete*** operation, different parameters are logged based on how the object delete action has been performed. We found that the object name is not recorded when AWS console command is used for deleting. However, the object name is recorded if the delete operation is carried out using the command line interface (CLI). Figure 5.2 shows two instances of the log when the delete operation is performed using console and CLI. Given that our use case assumes that Mr. Informant is tech-savvy, he might erase the activity traces by deleting the log files from the CloudTrail S3 bucket using AWS console. Doing so will erase the logs, thereby erasing the evidence of his activity.

Figure 5.2: Object delete snapshot from CloudTrail log

The deletion of the log file itself will be registered in later logs, marked as an event, "DeleteObjects", without the file name. Therefore, the investigator would not be able to identify which file has been deleted, making the traceability of the event actions challenging to prove and hence, not being able to satisfy some items listed in the core requirement in Table 4.1 specifically items R-4 (Justifiability) and R-6 (Traceability) and use case specific requirements listed in the Table 5.1, specifically item UCR-3 (Object operation).

Storage service operations: ($C_{i,5}$)

We performed all the use case operations related to storage service operations and particularly data exports from S3 storage buckets. Our experiments demonstrated that all the events, along with all the required forensics attributes, were recorded in the CloudTrail logs. Hence, satisfying the requirements.

Storage account operations ($C_{i,6}$)

While stealing the data, the culprit would do everything to hide their tracks. One possibility is to avoid using corporate storage area, which typically will be monitored more rigorously. One way to do it is by creating a private space to hold the data temporarily and funnel it through the holding area, and finally deleting the storage space. We simulated the scenario, created a temp storage area in the S3 bucket under a personal user account, and transferred the file through it. We found that, all the events related to such actions were traceable in the log.

Database service (NoSQL) ($C_{i,7}$)

Database service events include the events recorded in the logs, as a result of the actions carried out on the SQL and NoSQL databases. Given that the organization OOO uses databases to store some of the confidential data, we looked for the artifacts captured in CloudTrail for DynamoDB events. DynamoDB is NoSQL database service provided by AWS. We found that DynamoDB is well integrated with CloudTrail logs and all the table level activity events, such as *CreateTable, DeleteTable, ListTables, CreateBackup*, etc., are well captured in the CloudTrail log. However, we found that the log did not capture events related atomic activities like *putItem, getItem, updateItem, delteItem* etc. When Mr. Informant copies the records from DynamoDB (a *getItem* event), there are no traces left of the particular action anywhere. The fact that some but not all the DynamoDB events are registered in the CloudTrail log makes that the evidence available in the logs only partially satisfies the forensics requirements.

Database service (RDS) ($C_{i,8}$)

The organization OOO also uses AWS relational database service (RDS) to store product information and design parameters. We created an RDS instance using MySQL DB engine. Then we looked into the events that are logged in CloudTrail logs for typical database activities that a culprit would typically do, such as creating a DB snapshot, copying or transferring the snapshot elsewhere, and later deleting the snapshot. We found that RDS events and database actions such as *Create, Copy, Delete DB snapshot, Start, Stop DB instances* are all logged in CloudTrails with full forensics info [188]. We only looked at the database activities for this study rather than database record-level activities. The only exception we found is that for "sharing DB snapshot" (simulating a scenario where the culprit shared a DB snapshot with an external user). We found that no events related to *ShareDBSnapshot* and *MigrateDBSnapshot* actions in the CloudTrail logs, obviously producing a minor gap.

Web app ($C_{i,9}$)

Web applications are typical on a cloud platform. We assume that the OOO organization uses web applications and produces results upon execution. We assume that the results are also of importance to data thieves. We created a web application on AWS elastic beanstalk environment with SQL as the back end to simulate the situation. Then, we ran the application on a browser. The objective was to find out the events related to application execution in the CloudTrail logs. We found no events related to web app execution in the CloudTrail logs. In contrast, actions at the Beanstalk environment level like *CreateEnvironment, UpdateEnvironment, TerminateEnvironment* etc. are all registered in the CloudTrail [189]. One can assume that this is an expected behaviour, since *start*, *stop*, and *running* web apps are internal to the EC2 instances and will not be exported to a log service. For example, *start* and *stop* web app events are logged in "/var/log/eb-

activity.log" on that particular EC2 instance. However, the un-availability of app execution events in the CloudTrail logs creates a real deficiency from a forensics perspective.

Virtual server ($C_{i,10}$)

We created an AWS EC2 instance and examined the EC2 events related to the application execution, data access, and manual access of the data files in the EC2 instance. Anything that happens inside an EC2 instance is an operating system event and is internal to the EC2 instance, and they are in the respective EC2 system logs. Those events are not available in CloudTrail. The CSUs have to write inbuilt functions to pull the records and feed them into CloudWatch or similar tools for visualization while maintaining integrity and confidentiality.

Mail service ($C_{i,11}$)

WorkMail is the email service provided by AWS. We created the scenario of creating, sending, and receiving emails with and without attachments and then examined the artifacts captured in CloudTrail for WorkMail actions. We found that none of the WorkMail actions are recorded in the CloudTrail logs. Only the admin level activities (e.g., create or delete mail user accounts) are recorded in CloudTrail. AWS recommends using mail journaling to track emails. However, mail journaling is expensive and unlikely to be a practical solution for users. Therefore, the absence of any traces related to e-mail actions in the CloudTrail logs causes to fail some of the key forensics core requirement listed in the Table 4.1 (in particular requirements R-4 (Justifiability), R-6 (Traceability) and R-7 (Provenance) and use case-specific requirements listed in the Table 5.1 (in particular requirement UCR-9 (Mail services)).

5.6.3.1 AWS results summary

We analyzed further the CloudTrail log compliance with core requirements. The findings are summarized in Table 5.2. More detailed analysis with compliancy matrix are listed in Table A.1 of Appendix A. We used the following notations to categorize the compliance level.

✓ - fully compliant.

⊗ - partially compliant.

× - not compliant.

Table 5.2: AWS CloudTrail: Analysis of results

AWS CloudTrail Log Analysis	**Compliance**
CloudTrail produces an auditable, repeatable, and reproducible set of event logs, capturing all events. Repeating the same test/procedure logs the same/similar event trails.	R-1,R-2, R-3, R-5 (✓)
CloudTrail fails to capture and log entirely justifiable events. (For example, object name in delete object operation, user sign-out, and WorkMail events not recorded).	R-4 (⊗)
CloudTrail records events with UTC timestamp, hence easier to map the events on a timeline. However, it fails to record some of the events or parameters (e.g.: WorkMail events) required for comprehensive traceability.	R-6 (⊗)
Full provenance details are captured in every event log, including all the details to meet the quality requirements.	R-7, R-8 (✓)
The logs are co-mingled collection of all the events of all users using the same organizational account. Therefore, purpose-built scripts are to be executed on the log archive to separate and extract the event logs per user. Fails to meet the privacy and evidence segregation.	R-9, R-13 (×)
The logs and their respective digest files can be stored for any period using AWS's extra services. Also, access to log files can be controlled thru AWS access control policies.	R-10 (✓)

Continued on next page

AWS CloudTrail Log Analysis	Compliance
The log collection is possible across any jurisdictional boundaries, but not the selective event logs collection per user.	R-11 (⊗)
CloudTrail log format and event representation are unique and challenging to correlate with other CSP logs.	R-12, R-15 (×)
The log storage location is pre-configurable; hence, evidence identification is seamless.	R-14 (✓)
The logs nor their storage persists after the CSU account is terminated. The logs can be deleted by an authorized user anytime.	R-16 (×)
CloudTrail architecture and the AWS governance rules applied to the logs make them reliable.	R-18 (✓)
CloudTrail captures a good number of relevant forensic artifacts with enough details.	R-17, R-19 (✓)
The data representation supports readability, is easily interpretable, and is understandable.	R-20 (✓)

In summary, we found that the CloudTrail logs of AWS (Service S_x) satisfy a good number of forensics requirements but fail to meet fully all the stated forensics requirements, which can adversely impact the ability to track the actions by connecting the chain of events and finally to trace the crime to the doer.

5.7 Azure

In this section we are evaluating forensics compliance of Microsoft Azure (CSP_y), Activity log (S_y). Azure services are provisioned thru 'Azure subscriptions'. The Azure Activity log provides insight into events in a given user's Azure subscription. Using the Activity log one can determine the action taken on a computing resource in a subscription [124]. However, our study found that the Activity log does not capture all the events generated during the use case execution. Therefore, we used additional Azure logging services for doing a full comprehensive study. The additional logging services used are:

1. Azure Active Directory (AD) audit logs. The AD audit log provides trace-

ability into user sign-in, sign-out and attempted sign-in events. Also, Azure AD records user group activities, and enterprise application audit events.

2. Azure storage explorer event logs. Storage explorer log provides the file object-level activities of the objects in the storage.

3. Azure diagnostic logs. Diagnostic logs provide insight into diagnostic events.

Therefore, the log services can be considered as a set of services i.e., $S_{y,l} = \{S_{y,l,1}, S_{y,l,2}, ...S_{y,l,n}\}$, where $S_{y,l,k}$ represents the k^{th} logging service in the set $S_{y,l}$. All these logs were analyzed to collect forensics artifacts. The term 'Azure logs' are used to refer to all the above logs combined. We looked into these logs in detail and also considered all other available log resources for evidence collection. The log includes Windows event logs (which record Windows operating system events), Diagnostic logs (which record diagnostic events), and Office 365 audit logs (which record Office 365 mail events).

5.7.1 Azure System Configuration

We acquired and configured Azure services to conduct the experimental validation by executing the use case scenarios. First, we acquired an Azure subscription and created a resource group to group all the resources used in this evaluation, then we created and configured the required services. The following paragraphs briefly describe various Azure services used for this case study.

a) **Azure Active Directory (AD):** An Azure AD group (namely "APRESEARHORG") has been created with roles, permission, users, and enterprise applications assigned in the group. Specifically, the user "$i\_informant$" (which belongs to Mr. Informant) has been created and given a "Contributor" role, providing only limited but necessary access to the corporate owned resources.

b) **Storage account configuration:** Azure provides various types of storage, such as blobs, files, tables etc. Therefore, we created storage of different

types to carry out the experiments. We also created a container (namely "oooconfidentialdata") and stored the confidential objects in the container simulating the setup and environment used by the organization OOO.

c) **Azure storage explorer:** Azure storage explorer is a desktop application used for managing data in a storage account(s) in Azure Subscription(s). The storage explorer keeps the audit log of the activities at the storage object level in the $logs directory. We installed the Azure storage explorer application to capture and record storage level events.

d) **Azure Cosmos DB:** Cosmos DB is a NoSQL database service of Azure. We created and configured the Cosmos DB. Subsequently, created databases and collections within the Cosmos DB instance to simulate the conditions that OOO uses it as one of their information storage. The storage is accessible over the portal or using the Application Programming Interface (API).

e) **Diagnostic logs:** Configured the Azure Diagnostic log instance to capture the diagnostic events and stored that in a storage container and also streamed the logs into a log analysis platform.

f) **Application services:** We created web app services connected to SQL and Cosmos DB data repositories, simulating typical organizational web applications.

g) **Office 365 outlook:** Office 365 outlook is a SaaS cloud service provided by Microsoft, and it is not part of the Azure platform. However, since our case study assumes that part of the data leakage is done using an 'email service', we configured the Office 365 outlook component also to simulate the scenario.

h) **MailJet email service:** MailJet service is an email service available on the Azure platform, but it is a third party product. MailJet is primarily

to send out bulk emails to multiple contacts. However, to make our study comprehensive, we also looked into the artifacts generated in logs when the MailJet service is used for data transfer.

5.7.2 Azure Experimental Validation

We executed the same Data leakage use case, described in Section 5.4.1 on the Azure platform, simulating all the actions that $Mr.Informant$ would do to steal the data. We collected the Azure (i.e., CSP_y) logs (i.e., S_y), then analyzed the logs in detail, looking for forensics artifacts, with the key intention of answering the requirements stated in Section 5.5. Figure B.2 in Appendix B shows a snapshot preview of the Azure Activity log as an example. The example shows the activity of a normal user trying to create a role assignment to have the 'write' permissions enabled for the user on a resource belonging to an Azure subscription. The action fails, since the user does not have permission to do so. In the following Section 5.7.3 we are describing the results of our findings.

5.7.3 Azure Log Analysis

We analyzed the log data for its conformance and compliance with the forensics needs, similar to the analysis we did on AWS CloudTrail logs. Table A.2 in Appendix A provides the details of our analysis. We use the following notations to identify the logs viz., $l1$ for Azure Active Directory logs, $l2$ for Azure Activity logs, $l3$ for Azure Storage Explorer logs etc. forming the set of logging services $S_{y,l}$ where $l = \{l1, l2, l3...ln\}$. The result of our findings on Azure log events can be summarized as follows.

User audit $(C_{i,1}), (C_{i,2})$

We found that though sign-in $(C_{i,1})$ by CSUs are logged in the Azure AD logs, none of the sign-out events are recorded in any Azure logs, similar to AWS handling of user sign-out events $(C_{i,2})$. The non-recording of user sign-out events

could give a wrong perception that the user has been active in the system forever. As forensics always looks for credible evidence, the absence of sign-out events decreases the justifiability factor.

Object search ($C_{i,3}$)

Search events generated as a result of activities like a directory listing, object searching, etc. are recorded in the Storage Account Activity logs (i.e., service S_{y2}), with complete forensics artifacts. In addition, the events are also listed in Azure storage explorer activity event logs, i.e., service S_{y3}.

Object operation ($C_{i,4}$)

We looked into the events generated and available in the logs for typical operations like *read, put, copy, download* and *delete* operations performed on objects stored in a storage container. We found that all related events are logged in the Azure storage explorer activity logs in a particular folder $logs. The logs are plain text files that require a good understanding of the storage explorer log format to decipher and interpret the log data. The supporting forensic information of significance such as **'what', 'when', 'where'** and **'how'** are also available. But the user-id of the actor (i.e., **'who'** is not recorded in the logs, though client IP address, the requester account names, the storage account name, the interface used to perform the action, i.e., **'how'** and the results of the action are logged. The client IP address only provides the info to ascertain **'from where'** the action has been originated. We tested a scenario where the user *'i_informant'* has created a snapshot of a set of blob objects holding critical information assets, downloaded the snapshots, and then deleted the snapshots and subsequently deleted the logs too. We could extract the traces of all such actions from the logs in the $log folder. The record of deletion of the log files was available in the subsequent logs, but none recorded **'who'** (i.e., the user-id) did it. Azure comments that the logs do not include the user-id information for security reasons [190]. The 'actor' re-

sponsible for the event is a critical piece of forensics evidence. Without that info available in the log, it is pretty impossible to link the chain of events to a doer, therefore, challenging to prove the provenance. Though the Azure logs satisfied an almost complete set of required evidence, the logs failed to meet some of the most critical forensics requirements stated in Section 5.5, in particular items# R-7 (Provenance) and R-8 (Traceability).

Storage service operations ($C_{i,5}$)

We tested for the scenario where the user *'i_informant'* exported specific data stored in 'Azure table' storage, using Azure storage explorer. We found no record associated with the 'export' operation in the $log directory. Azure says that the 'export' operation is an application operation, so any storage account logs will not be recorded. But the 'export' operations request data from the storage account that will create a *QueryEntities* operation, which is registered in $log [190]. But the same event (i.e., QueryEntities) is also recorded for other operations which request data from the storage account. Therefore, it is tough to ascertain the fact that the user *'i_informant'* has in fact, exported data from storage. We also found no in-built Azure services available to automatically archive the logs files from $log folder along with their hashes for later use as evidence, and it is a serious inadequacy as far as forensics is concerned. Azure "AzCopy" command is one way to export the log data files. Therefore, to support forensics, one must develop specific programs for periodically archiving the logs and generating the hash.

Storage account operations ($C_{i,6}$)

We tested for another scenario in which the culprit is using a temp storage account and a data holding area to funnel the data out. One can avoid using organization-owned shared storage account, by utilizing a temp storage account,. The scenario steps are (i) creating a temporary storage account and data storage

space, (ii) copying the file(s) across to the temp storage, (iii) downloading the files, and finally deleting the temp storage account. We found the entire evidence of storage level actions (i.e., creating, updating, deleting) in the Azure activity logs and the object level actions on the $log folder accessible thru the Azure storage explorer. But, it is to be noted that the evidence is collected from different logs, from different locations, and the logs are of different formats.

Database service (NoSQL)($C_{i,7}$)

Cosmos DB is the NoSQL service, and it is used for storing large amounts of data. We assume that OOO also uses it as a data storage service. Upon examining the artifacts left in the log for Cosmos DB database events like *creating, deleting, updating, copying* documents, we found that the Cosmos DB events are scattered in two logs. They are the Azure Activity logs and Diagnostic logs. In the Cosmos DB context, the Activity log provides data about the operations on a resource from the control plane, such as *create, delete a container, list database* etc., but not any actions at the object level. Whatever is logged in the activity logs, it is logged with full details, such as operation name (**what**), timestamp (**when**), caller e-mail address and IP address (**who**) resource (**where**), methods of performing the operation (**how**). However, the list of operations recorded is only a subset of what is required from the forensics point of view. Therefore, we have to use also the Azure Diagnostic logs to collect details of actions at the object level, such as the operations done at the database document level. We found that the Diagnostic logs provide only very minimal info. For example, the 'user id' is stripped off, the client ip address is that of the internal virtual network. It does not correspond to the actual client IP (i.e., incorrect **who** info) [191]. Operation at a database collection level (e.g., deleting a Cosmos DB collection) activities are recorded in the log, but without the name of the actual object deleted (i.e., no **what** info). Object activities at the atomic level such as *update*, *delete* operations performed on the Cosmos DB documents are not registered in the logs either. It

is easy for an insider to cause a denial of service attack by deleting some/all of the documents with no activity traces. We combined both the logs, i.e., the Activity logs and Diagnostic logs, and tried creating a timeline of events. However, it is tough and challenging to associate an action to a doer with certainty since much of the critical information to connect the chain of events are missing.

Azure does not have an in-built feature for exporting or copying data from Cosmos DB. Exporting (or importing) data require specific scripts or purpose-built applications. To simulate the condition that the user *'i_ informant'* has funneled the data out by exporting Cosmos DB records, we used one of the Azure recommended tools (Studio 3T) and exported the data from Cosmos DB. But there were no traces of such actions in any logs. Therefore, investigators would need direct support from CSPs to find out that such activities have been done.

In an attempt to wipe out whatever evidence has been left around, the culprit would often delete the log file itself. The customers usually store the log files in long-term storage. We configured the diagnostic logs to be streamed to a storage container. We then simulated the situation where the user *'i_ informant'* deleted the diagnostic log files from the storage container and examined the traces of that action in any of the logs. Upon analysis, we found evidence of the corresponding activity, i.e., the deletion of the log file itself, in the Storage explorer logs.

Database service (RDS) ($C_{i,8}$)
We simulated the actions of stealing data or making the data unavailable by directly accessing the Azure SQL databases from the portal, such as *export*, *copy*, or *delete* database. We found the events related to all such actions, and they are all recorded in the activity log with full details, meeting forensics criteria.

Web app ($C_{i,9}$)
We built a web application using the ASP.NET framework and Visual Studio.

The web app uses Azure MS SQL and Cosmos DB as the data store. Integrated the application with Azure AD for user authorization, i.e., only Azure AD authorized users' can execute the application. We assume that the app's execution displays results that are of much value to the informant. We then analyzed the logs for traces. We found the events in the Azure AD sign-in logs, with full application execution details (i.e., 5W1H). However, the evidence is stored in yet another log, in .csv format. It is to be noted that, if the application is not integrated to use Azure AD credentials, no events related to the app execution are registered in any Azure logs.

Virtual server ($C_{i,10}$)

We examined the Virtual server events by creating a Windows virtual machine in Azure with an attached disk volume. We executed various commands typical to that of an insider culprit who wants to harvest data. We found that all the Azure level activities, such as *start, stop, run* command etc., are all recorded in the windows event logs. However, any action carried out at the server level by remotely connecting to the server itself is not part of the Azure events, which is expected. However, activities conducted on the disk volume, like creating a snapshot and exporting the disk data are all registered in the Activity logs.

Mail service ($C_{i,11}$)

As our case study assumes that the data leakage partially has happened via email service, we looked into the events recorded in the Office 365 outlook that is of forensics value. It is to be mentioned that Office 365 outlook is not part of the Azure stack. Microsoft provisions Office 365 as a separate SaaS service. Nevertheless, we decided to analyze the Office 365 Outlook events to get comprehensive coverage. None of the events generated by Office 365 are part of Azure logs, as expected. But Office 365 Outlook provides its complete Audit logs as a part of its security and compliance suit [192]. If the mailbox auditing is enabled, most of

the actions are logged in the mailbox audit logs [192], but events related to creating, sending, or receiving messages are not part of the audit logs. Such events are crucial to determining, *who, what* and *when* sent out the messages and to *whom* in a forensics world. User sign-in events to the mailbox are recorded, but not the sign-out events. To investigate the sent message events, one needs to use eDiscovery tools to find *whom*, *when*, and *emailsubject* as long as the message is present in the mailbox. In a scenario where the culprit, *'i_informant'* has sent the mail with data and permanently deleted the mail, the eDiscovery tools will not be able to fetch the details, leaving the organizations to go for rather expensive options like mail journaling or archiving to retain the messages.

Mail service (MailJet) ($C_{i,12}$)

MailJet is a third party cloud-based email service primarily used to send bulk marketing emails. Azure supports installing and using MailJet on its platform. To achieve comprehensive coverage of our study, we also looked into the MailJet events. We tested the scenario where the user *'i_informant'* used MailJet as a means of data transfer. We could extract the evidence of *i_informant's* activity from MailJet portal. The portal provided the **Date time, From, To and Subject** information, which are all critical forensics data.

	C	D	E	F	G
1	From	messageid	subject	status	To
2	i_informant@apresearchdomains.org	4.42E+16	sending data as per plan - Part 1	sent	09469082@student.curtin.edu.au
3	i_informant@apresearchdomains.org	4.50E+16	sending data as per plan - Part 2	sent	09469082@student.curtin.edu.au

H	I	J	K	L	M	N	O
unsub	bounce	blocked	click	spam	open	queued	sent
f	f	f	f	f	f	f	t
f	f	f	f	f	f	f	t

Figure 5.3: MailJet log snapshot

However, None of the MailJet, mail actions are captured in any Azure logs, which is also expected since MailJet is an external service. The Figure 5.3 depicts a snapshot of MailJet's own log.

Hiding of alerts on information assets

We also looked for evidence in the log available for activities like hiding alerts or alarms, which one can expect a culprit to do as the first activity normally, even though it is not part of the use case. Azure provides the capabilities to create alerts (i.e., trigger alarms) on actions carried out on the resources (for example, downloading blobs). Alerts help to initiate incident response and act as a warning sign. Therefore, simulating the scenario which culprits will typically do, we conducted tests in which the user *'i_informant'* disabled or deleted the 'alert' on file access, such that no alert message will be sent out. We then looked for the traces of such actions in the Activity log and found that for 'disabling' the alert event is marked as 'Create or Update metric Alert' in the log. However, we found that the same operation name is also used for 'update' operations. To find out whether the operation has indeed disabling of an alert, one have to dig deep into the log, and it is marked under 'requestbody' where the value corresponding to parameter "enabled" is "false", (correspondingly the value is "true" for enabling the alert). A snapshot from the corresponding parts of the Azure log is shown below:

{
. . .
"operationName":{
"value":"“Microsoft.insights/metricAlerts/write”",
"“localizedValue”": "“Create or update metric alert”", }
. . .
“properties” :{
“statusCode”: “OK”,
“serviceRequestId”: “8e358cf5-657204c73-8dc2-a698812a306”
“requestbody”:
”{. :\"description\”:\"alert \":\”,\enabled\”:false\. } }

Delete operation is marked as ""“Delete metric alert”"" as the operation name with no “requestbody”, as shown in the log extract below.

```
{
. . .
"operationName":{
"value":""“Microsoft.insights/metricAlerts/delete”"",
""“localizedValue”": ""“Delete metric alert”"", }
. . .
}
```

Though the attempt to hide alert is recorded in the log, the usage of the same operation name for different activities and the inconsistency in the data representation makes it challenging to decipher the information and analyze it.

5.7.3.1 Azure results summary

We analyzed further the Azure log compliance with core requirements. The findings are summarized in Table 5.3. More detailed analysis with compliance matrix and the specific logs in which the evidence can be traced are listed in Table A.2 of Appendix A. We used the same notations to describe the compliance level as described in Section 5.6.3.1.

Table 5.3: Azure logs: Analysis of results

Azure Log Analysis	**Compliance**
Azure logs produces an auditable, repeatable, and reproducible set of event logs, capturing all events. Repeating the same test/procedure logs the same/similar event trails.	R-1,R-2, R-3, R-5 (✓)
Azure logs fails to capture and record entirely justifiable events. (For example, user sign-out events not logged, insufficient data logged for database service events).	R-4 (⊗)

Continued on next page

Azure Log Analysis	Compliance
Azure records events with UTC timestamp. However, the events are spread across in many log files.	R-6 (✓)
Mostly, full provenance details are captured in every event log, including sufficient details to address the quality requirements.	R-7, R-8 (✓)
Azure logs are co-mingled collection of all the events of all the users using the same Azure subscription. Therefore, purpose-built scripts are required to separate and extract the event logs per user. Fails to meet the privacy and evidence segregation.	R-9, R-13 (×)
Azure performs transparent encryption of the data at rest in the storage by default and this feature cannot be disabled. However, Azure does not provide any in-built feature to create the digest (e.g: MD5 hash) of the log files when the files are streamed into a storage location. Having the facility to produce the digest of the log files is an essential element for integrity validation.	R-10 (⊗)
It is possible to go beyond one's own jurisdictional boundaries and download the logs from storage provided they all belong to the same Azure subscription, but to collect the logs per user within a subscription requires additional log extraction tools.	R-11 (⊗)
Azure log files and the events are spread across many logging services, such as Activity logs, Diagnostic logs, AD logs, and Storage explorer logs. The logs are of different formats too. Activity logs are of JSON format. Storage explorer logs are ';' delimited text files. AD sign-in information are .csv format. The logging services and storage locations are spread across Azure infrastructure. There is no consistency in log format within Azure itself, let alone among CSPs.	R-12, R-15 (×)
The log data are spread out in multiple logs at multiple storage locations, producing much complexity for evidence identification. Investigators must know **where** to look, **what** to look for.	R-14 (×)
The logs nor their storage persists after the CSU account is terminated. The logs can be deleted by an authorized user anytime.	R-16 (×)
Azure provided access control and governance rules applied to the logs make them reliable.	R-18 (✓)

Continued on next page

Azure Log Analysis	Compliance
Azure logs captures a good number of relevant forensics artifacts with enough details.	R-17, R-19 (✓)
There is no commonality in the data representation among Azure logs, and complex to understand and interpret. Investigators would require a good understanding of Azure architecture and log format to analyze the log data.	R-20 (×)

In summary, Azure log services register a good set of events happening in an Azure subscription. Activities outside of the subscription are not logged. The logs satisfy a good number of forensics requirements. Still, they fail to fully meet all the stated requirements, adversely impacting the ability to track the actions by connecting the chain of events and tracing the crime to the doer. It is to be noted that the different log services, different log formats, and varying log storage locations cause additional complexities in identifying, collecting, and interpreting the evidence. In general, it is assumed that forensics examiners may not be technically competent or have the knowledge of different cloud logging architectures, requiring the services of subject experts or CSPs.

5.8 Google Cloud Platform (GCP)

For studying Google cloud forensics compliance of the logs, we used Stackdriver logging service as their prime logging service as the Service $S_{z,l}$ provided by Google Cloud Platform CSP_z. Stackdriver logging provides a centralized and comprehensive logging service, and we used the events available in the Stackdriver logs for forensics compliance evaluation. [171].

5.8.1 GCP System Configuration

We configured GCP environment with users, groups, and resources to carry out the experiment and collect the forensics log artifacts, as briefly described below.

a) *User accounts:* We created Google accounts, owner account, and a few users account, including the user account for Mr. Informant as *"informant.iam"*, and then configured the user accounts with data read and write access.

b) *Storage accounts:* Google provides storage solutions for any type of workload, such as those requiring high performance object storage for frequent access and low frequency access storage for backup disaster recovery purposes. For the purpose of our use case tests we created frequently accessed regional storage ($ooo\_confidential\_data$) where typical OOO organizational data are stored and a low access storage for exporting the logs ($ooo\_log\_exports$). We also created instances of Bigtable (big data storage service) and Firestore (NoSQL) storage services. Firestore native mode configuration has been used, which allows the database to be organized into documents and collections.

c) *Relational Database:* To find out the artifacts available in the relational database services of GCP, we created and configured SQL instances using MySQL second generation as the database and connected to the Google app engine.

d) *Other:* Other GCP services like Compute Engine and Kubernetes Engine are configured with the appropriate server, database, and storage required to carry out the tests.

5.8.2 GCP Experimental Validation

We executed the same Data Leakage use case described in Section 5.4.1 on the GCP platform and collected the Stackdriver log data. Then, analyzed the logs looking for forensics evidence, with the key intention of answering the requirements stated in Section 5.5. Figure B.3 in Appendix B shows a snapshot of the Stackdriver log, corresponding to an event where the actor $Informant$ has downloaded a file from a GCP storage. Since the log files generate too many entries,

we used custom-built scripts to extract and display the records of interest. For example, the following script shows all user activities of $informant.iam$ on the resource type gcs_bucket (i.e., storage bucket), between the specified time slots.

---

resource.type="gcs_bucket" AND
protoPayload.authenticationInfo.principalEmail =
"informant.iam@gmail.com" AND timestamp >= "2018-10-15T08:45:00Z" AND
timestamp <= "2018-10-15T09:45:00Z"

5.8.3 GCP Log Analysis

We analyzed the Stackdriver log after every step of the use case scenario execution, for its conformance and compliance with the forensics needs. Table A.3 in Appendix A provides the details of our analysis. Unlike Azure, we could find all the log artifacts in one place, i.e., in the Stackdriver logs. The results of our findings are summarized as follows.

User audit $(C_{i,1}), (C_{i,2})$
Google cloud does not record any user login attempts to the cloud platform, neither sign-in $(C_{i,1})$, nor sign-out $(C_{i,2})$. Normal failed log-in attempts, such as using the wrong password, are also not recorded. We could find only the records of the failed request to the resources when IAM policies restrict the user accounts. In Google, user's authentication is managed by Google accounts, so GCP has no record of any user log-in/log-out details. This will create a major void if we depend only on the GCP Stackdriver logs from a forensics perspective.

Object search $(C_{i,3})$***, Object operations*** $(C_{i,4})$
We looked into the events generated and available in the logs for typical operations like *search, read, copy, download* and *delete* operations performed on objects

stored in a GCP storage container. It is logical to assume that these are the typical operations any culprit would do to steal the data or carry out a denial of service attack. We found that all the related events are logged in the Stackdriver logs with full details, satisfying the essential forensics requirements.

Storage service operations $(C_{i,5})$ ***and Storage account operations*** $(C_{i,6})$
One can expect that the data thief would create some temporary storage to hold the data of interest before being shipped out. To simulate the scenario, the actor $Informant$ first signed in to the GCP environment and created a temp storage bucket to hold the data. Then copied the data of interest to the temp storage using cloud shell commands (*gsutil cp* commands). Later, transferred the data from the temp storage to a local, removable media. Finally deleted the temp storage bucket and tried deleting the logs. We found the evidence related to all actions, with complete supporting information in the log, for different modes of data transfer services. We also found that GCP does not allow deletion of the audit logs, even by the project owner [193]. Audit logs are kept for a specified length of time before being deleted. This feature enhances the availability and trustworthiness of the logs. We further expanded our tests by performing the same operations using the file ''Transfer" feature provided by GCP. (i.e., data export). Though the actual execution of the transfer script is not recorded in the log, all the file transfer events are registered in the log.

Data storage service (Bigtable) $(C_{i,7})$
Bigtable is a highly scalable, compressed, and high-performing big data storage service of GCP, storing petabytes of data. In this case study, we assume that the Bigtable service is used for some of the organization's future product research using time series data analysis. The outputs of some of the studies are published in the Bigtable instance. Therefore, we looked into the typical activities, and the forensics evidence available in the Stackdriver logs for operations on the Bitgable

instance. We found all the admin activities logged (e.g: create/delete instance or table) but not the data access on the Bigtable data [193]. Though this creates a void in forensics traceability, logging all data access actions would generate a massive set of log entries.

Database service NoSQL (Firestore) ($C_{i,8}$)

Firestore is the newest NoSQL document database service provided by GCP. We assume that the organization uses NoSQL database to store critical company artifacts. Performing and analyzing the Stackdriver logs for actions performed on the Firestore data store, we found that none of the object level activities, such as creating document collection, adding/deleting documents, adding/updating fields, etc., are registered in the Stackdriver logs. Google recommends the CSUs use Stackdriver REST API to directly log Firestore events, i.e., requiring intervention by CSUs to make their applications forensics compliant. We found that data export activity exporting data from the Firestore database performed by cloud shell commands is fully registered in the Stackdriver logs.

Database service (RDS) ($C_{i,9}$)

We created SQL database using MySQL database engine within GCP. We assume that the database holds the master data of the organisation. We carried out various test scenarios for the database using the GCP console and cloud shell commands. We observed that the Stackdriver provided a complete log of all the actions. Database export is the main method of funneling the data, and the database export action has been logged with full event details. Figure B.4 in Appendix B shows an extract of the database export operation log.

Web app ($C_{i,10}$)

We built a Google App Engine application and integrated it with MySQL instance. We assume that the application execution lists vital information of inter-

est to the data thief. Analyzing the Stackdriver logs for traces of the application execution, we found partial evidence of the app execution, listed as a "GET" operation, answering **when, how, where, what** but not the ***who*** aspect, i.e., all information except the client's user id or ip-address. However, the Stackdriver provides interfaces for users to write application logs with more info, moving the responsibility to the user.

Virtual server Kubernetes engine ($C_{i,11}$)

We created Google compute engine web server instance with Mongo DB as the database engine and a user facing web client app and carried out all the tests. We also created a Google Kubernetes Engine (GKE) instance app and hosted it on a web app. Kubernetes engine is one of the latest Google innovative services allowing to deploy large scale applications in a containerized environment for seamless operation, auto-scaling with high availability [194]. We looked at the default logging of events corresponding to the execution of the web apps running on Google compute engine or Kubernetes Engine and found no specific events in the log. By installing Stackdriver logging agent on the Virtual Machine instance we could stream the Compute Engine events to the Stackdriver logs. We also found that by default GKE application events are not visible to the Stackdriver either, and hence those events are not logged. (To monitor the Kubernetes engine generated events, Google recommends customizing Stackdriver logs with Fluentd. Fluentd is an open source cross platform data collector). Therefore, to fully record evidence for forensics purposes, CSUs have to configure and maintain the GCP provided products and services with the necessary custom code.

Mail service

Google does not have any in-built mail service as a part of the Google cloud platform. Therefore, we could not test the data transmission scenarios using a GCP e-mail service.

5.8.3.1 GCP results summary

We analyzed further the GCP Stackdriver log compliance with core requirements. The findings are summarized in Table 5.4. More detailed analysis with compliance matrix are listed in Table A.3 of Appendix A. We used the same notations to describe the compliance level as described in Section 5.6.3.1.

Table 5.4: GCP Stackdriver logs: Analysis of results

GCP Stackdriver Log Analysis	Compliance
Stackdriver produces an auditable, repeatable, and reproducible set of logs, by capturing all events. Repeating the same test/procedure logs the same/similar event trails.	R-1,R-2, R-3, R-5 (✓)
Stackdriver fails to capture and record entirely justifiable events. (For example, user sign-in and sign-out events not logged, insufficient data logged for web app events).	R-4 (⊗)
Stackdriver records events with UTC timestamp, thereby making it possible to map the events on a timeline	R-6 (✓)
Mostly, full provenance details are captured in every event log. However, in cases like web app execution events; the provenance element is missing.	R-7, (⊗)
Stackdriver records events with sufficient details, satisfying the quality requirements.	R-8 (✓)
Stackdriver logs are stored per-project, clearly separating one project log from another. Within the project, activities can be filtered per user, supporting privacy and evidence segregation.	R-9, R-13 (✓)
When the Stackdriver log files are exported to the storage bucket, their corresponding hashes are computed and stored as a part of metadata. These hashes can be viewed using "gsutil" commands. The hashes can be used to validate and check the log files' integrity. Having the digest of the log files generated by the cloud platform ensures integrity and increases the trust factor.	R-10 (✓)

Continued on next page

GCP Stackdriver Log Analysis	Compliance
Stackdriver logs can be accessed regardless of jurisdictional boundaries, provided that access is provided and enabled thru GCP Identity and Access Management (IAM) access provisioning mechanism. Additionally, GCP allows the log collection per project or user basis.	R-11 (✓)
Stackdriver logs provides a central repository of all the events happening in GCP, and also allows the ingestion of custom log data from any source. The logs are of uniform (JSON) format. Having the log data in the same format and available in a central store facilitates easy forensics data identification, acquisition and correlation.	R-11, R-12, R-14, R-15 (✓)
The logs nor their storage persists after the CSU's, GCP account is terminated. However, the logs can be exported to long term storage in cloud or custom location.	R-16 (⊗)
GCP provided IAM access control and governance rules applied to the logs make them reliable.	R-18 (✓)
Stackdriver captures a good number of relevant forensics artifacts and records with enough details.	R-17, R-19 (✓)
Stackdriver data representation is simple to understand and interpret and follows a common structure, increasing usability.	R-20 (✓)

In summary, we found that the Stackdriver log service is a single source of repository of events and registers almost all activities happening in a GCP project, satisfying most of the forensics requirements.

5.9 Chapter Summary

The dependency on the log files for analyzing a post-crime activity performed on a cloud computing platform is one of the essential requirements for cloud forensics, especially in an era where cloud computing has acquired significant market share and continues to grow rapidly. However, collecting, analyzing, and finding the evidence from the logs are challenging. Many research works have been carried

out in cloud forensics, primarily addressing the issues, challenges, and solution proposals. There have also been significant works in the area of cloud logging in support of forensics and security.

In this project, we took cloud forensics to the next level by studying the current state of forensics readiness of the three major cloud platforms (AWS, Azure, and Google), following a well defined methodology. We used their respective logs as a service, validating them against a set of forensics requirements and using use case scenarios to test. The evaluation has been done by looking at the events registered in the logs and then assessing the value of the events as a forensic artifact. Such that the artifacts can be used to prove a digital crime. Being able to timeline the events is a critical factor in any digital investigation. All the CSPs we evaluated in this study record the event time in UTC helps in investigations, though the actual timelining of events remains an investigation task.

Further, we found that all the three log services we evaluated in this study support most forensics requirements and provided a good picture of where to find the evidence and what to expect. However, there are non-compliances. Below we are highlighting the important ones.

1. The ability to attribute an action with 100% certainty to a doer based on evidence collected from the log; we identified the gaps in meeting the requirements as listed in Tables A.1, A.2, and A.3 of Appendix A.

2. The ability to use CSPs' in-built capability to generate the hash of the logs to evaluate the integrity and trust factor; we found that Azure does not provide the capability of providing hash of the logs when the logs are exported to storage, at the time of this study.

3. The logs are mixed records of events produced by all users using the same environment. Therefore, purpose-built scripts are required to extract the records per user, which in turn changes the truth of source. The only exception is the GCP Stackdriver logs, where the logs are stored per project.

4. We also argue that the seamless, factual, and easy log interpretation is more difficult with Azure, because the evidence is scattered in different logs, different storage locations, and with inconsistent log formats as described in Section 5.7.3.

5. Mostly, every CSP's log conforms to a structured format, though CSPs have no log format consistency among themselves. In addition, we also found that different Azure logs use different formats and are stored in different storage buckets, making it harder for evidence identification, collection, and correlation.

6. None of the CSPs logs we used in this study persists after the user account has been terminated. Therefore, CSUs have to take additional actions to store the logs elsewhere before account termination.

7. No clear evidence of shipping the data out using the respective mail services in the respective logs provided the CSPs.

In summary, our research provided a comprehensive view of the cloud forensics maturity levels of the three major cloud platforms assessing the evidence available in the prime logging services of AWS, Azure, and GCP. The study concludes that the CSPs we looked at provide a good depth of support for cloud forensics. However, none of them are fully compliant and the level of compliance varies according to the CSP. Our assessment and research would help CSUs and forensics investigators to make well-informed judgments while resourcing services or using them in digital investigations. Since cloud services are a moving platform, using the methodology explained in this study would help to assess the forensics compliance of cloud platforms at any given time.

Chapter 6

Forensics Logging Framework for IoT

Recall that in Section 2.3 we provided an in-depth introduction of IoT. Further, we mentioned that IoT technology is becoming the next modern approach where boundaries between real and digital domains are progressively eliminated by consistently changing every physical device to a smart device designed to provide value-adding services through intelligent automation [25, 195]. In Section 2.3.1 we also highlighted that IoT is very much susceptible to security exploitation and cyber-attacks and the importance of IoT forensics. However, IoT technology is not yet mature for digital forensics. This chapter presents an IoT forensics framework for enabling digital forensics in IoT, using an interconnected Cloud-IoT computing platform. The proposed framework would facilitate proactive evidence collection and logging in anticipating of security incidents, supporting efficient and effective digital investigations. More importantly, our framework would also help study how an attack has been carried out, which would contribute to the discovery of IoT vulnerabilities. The specific contributions of this work can be summarized as:

1. It provides a comprehensive summary of IoT security attack vectors and IoT forensics challenges, and then derives and defines the IoT forensics

requirements.

2. It proposes an IoT event logging and data collection framework, specifying the log architecture and the parameters to log, addressing the requirements.

3. It proposes the log collection design stack for building forensics capable IoT systems.

Relevant parts of this chapter have been published in the author's conference article [196].

The layout of this chapter is as follows: Section 6.1 describes the need for IoT forensics and the motivation to do this research project. Section 6.2 details the methodology used in this study and Section 6.3 describes the IoT forensics, including forensics components and challenges. Section 6.4 describes and models the problem. Section 6.5 defines and specifies IoT forensics requirements. Section 6.6 describes the framework architecture and Section 6.7 proposes the solution design stack. Finally, Section 6.8 summarizes the contributions of the chapter.

6.1 Motivation

The motivation to undertake this project promulgated from the fact that there is serious lack of forensics capabilities in IoT space. The diverse use of IoT technology has created the proliferation of IoT devices with no conformance or strict governance on the architecture and communication protocols [25, 83]. No universally applicable IoT architecture currently exists either [197]. The lack of governance and standards also contributes to security vulnerabilities in IoT [23, 82]. To complicate the digital investigation further, there are reported cases where the intruders, instead of directly targeting the IoT, it is used as a weapon to launch an attack elsewhere [25]. Such cases also enforce the need for IoT to have robust security architecture and, equally important to have strong forensics capabilities. Investigating and analyzing the root cause of a cyber-attack on IoT

helps in the forensics investigations and helps to design better and more secure IoT systems. A few of the arguments supporting IoT forensics can be summarized as:

- **Extensive attack surface:** The large-scale deployment of IoT and wide usage has been creating an ever-increasing attack surface [25, 84]. IoT devices with public interfaces are exposed to greater risk levels because they could bring malware by exploiting it as an entry point to the private network from the public space.

- **New cyber physical environments:** The widespread deployment of IoT has also created a new cyber physical environment. For example, the IoT can be used to launch distributed denial of service (DDoS) attacks, causing a large-scale disruption of service. With billions of devices connected, the ferocity of such attacks will be very difficult to contain and respond to the incident quickly and effectively, especially in a scenario where more than one-quarter of the attacks will involve compromised disparate devices [29].

- **Life threatening security risks:** Since IoT technology has become an essential part of the medical industry and safety systems, any potential compromise of the systems can result in life-threatening consequences. For example, (i) cases have been reported where certain pacemaker models were vulnerable to hacking [25] and (ii) compromise of programmed IoT based fire safety lock system, fails to shut off the gas supply in the event of a fire [198].

- **Smart cities:** IoT-based technologies have been widely used to optimize city infrastructure, transport networks, and utilities, which rely on sensitive real-time data generated by IoT. Compromising such IoT could result in a dangerous, chaotic situation.

The above reasoning substantiates why IoT forensics is needed and the motivation for this project. Also, the data collected and shared by ubiquitous sensors present

an abundance of potential digital evidence by virtue of their numbers, variety, and coverage in many application areas [199]. Furthermore, the digital artifacts found in the IoT ecosystem can be used to support or refute investigation hypotheses and, subsequently any claims made by parties involved in a litigation [199], but there is a lack of formal methods to capture and save the artifacts. Moreover, conventional digital forensic tools and procedures do not fit the IoT environment. A few reasons to cite are: (i) the mobility (e.g., IoT is widely used in moving platforms) (ii) highly distributed and decentralized nature of IoT devices and services, (iii) the sheer volume of the devices, which can generate a massive amount of potential data [88], (iv) physical size, usually very tiny and (v) IoT lacks standard architecture and communication protocols.

We observed that there is no structured approach or well-defined framework to format, collect, and store the forensics artifacts from IoT pro-actively, such that the artifacts will be of value for an investigation, and that is the main focus of this research project.

6.2 Methodology

We applied the following methodology to conduct this research study.

1. Describe the IoT forensics components.
2. Describe IoT attack taxonomy and the possibilities for exploiting the IoT.
3. Identify and describe the IoT forensics challenges.
4. Describe and model the problem.
5. Derive the IoT forensic requirements. The requirements are derived using steps 2 and 3 inputs and investigative needs. We consider that the investigative needs are the same as that of the cloud forensics described in Section 4.1

6. Design and describe the proposed solution framework and architecture and

7. Provide the solution design and verification of the framework.

6.3 Internet of Things Forensics

As IoT applications are mainly used to control critical systems and infrastructure, safeguarding the smart devices against possible sabotage is one of the paramount objectives. So does the importance of having the capability to find out instantly in case of possible sabotage through a smart incident handling process. Therefore, the IoT must be designed and built with a security-first approach to thwart any exploitation and forensics-second to support post-incidence forensics investigation.

Recall that in Section 2.3.1, we outlined the IoT forensics, and related work in the same field. Further, we mentioned that IoT forensics is an extension of digital forensics applied to the IoT domain, with the same purpose and intent, i.e., to find out the modus operandi of the cyber attack and the culprit responsible. This section further elaborates on the relevant research outputs related to our study.

6.3.1 IoT Forensics Components

The IoT devices are seldom used alone and there is an amalgamation of enabling technologies behind it, such as networks and data storage. Therefore, IoT forensics also need to consider the enabling technologies. Zawoad *et al.* [26] defined IoT forensics as a combination of three digital forensics schemes: (i) Device level forensics, (ii) Network forensics, and (iii) Cloud forensics. Figure 6.1 illustrates the IoT forensics components.

1) **Device level forensics:** Involves identifying and collecting data from the IoT memory. Since the IoT devices have limited memory, data volatility is a significant issue. Therefore, device forensics is time-critical.

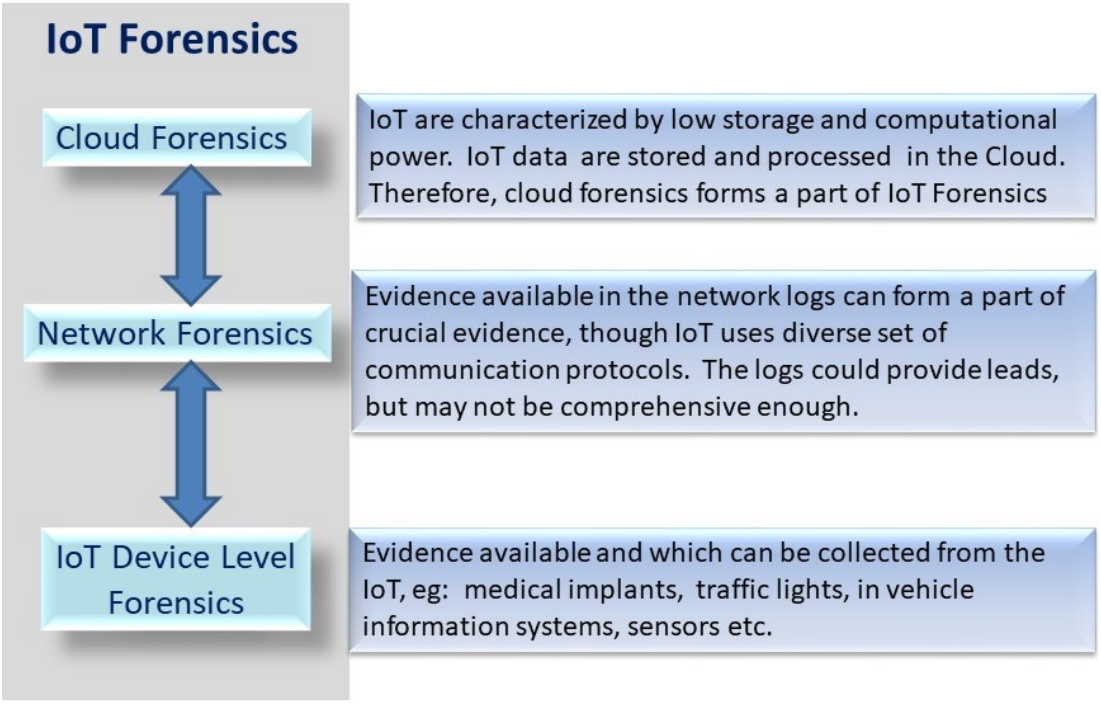


Figure 6.1: IoT forensics components

2) **Network forensics:** Networks are the backbone of any IoT infrastructure. Network forensics includes the forensics of the network logs or data stored on the network devices. Network logs would potentially identify the source of an attack or pronounce the legitimacy of an activity or otherwise.

3) **Cloud forensics:** Most of the IoT generated data will be stored or processed in the cloud. Hence, most of the vital clues of IoT activities can be found in the cloud. Therefore, cloud forensics is a major part of IoT forensics.

The primary contribution of our work is focused on enhancing the device level forensics capabilities thru smart event logging.

Li *et al.* [84] provide a different dimension to IoT forensics by first classifying the IoT-related crimes into three groups: **IoT as a target, IoT as a tool**, and **IoT as a witness**.

1) **IoT as a target:** Refers to the cyber attacks where vulnerabilities in IoT devices are exploited. As already mentioned, IoT devices are characterized by limited storage, computational power and power supply resources. Therefore, installing and running data security solutions on such devices is almost impractical, which makes the devices an easy target for cyber attacks.

2) **IoT as a tool:** IoT device becomes a tool when a compromised IoT has been used to facilitate or carry out malicious attacks elsewhere.

3) **IoT as a witness:** IoT devices become a witness when the evidence data is stored in the IoT, or stored elsewhere (e.g., cloud storage) but originated from the IoT, potentially contributing to proving or exonerating a crime. For example, the video footage of a crime scene, collected from a video monitoring IoT.

The framework proposed in this work has been designed to capture all the events, regardless of how the IoT devices have been used in a crime.

6.3.2 IoT Attack Taxonomy

Understanding the IoT attack scenarios is a valuable input to the IoT security architecture design and serves as an input to draw the forensics requirements. The large-scale deployment, wide usage of IoT technology, and many proprietary architecture and protocols have considerably widened the footprints for IoT-based attacks. Further, as noted earlier, it is not practical to implement robust security protocols in IoT. IoT is often deployed in unattended and remote locations too. All of these makes IoT very much susceptible to attacks. Abdul-Ghani *et al.* [200] and Deogirikar *et al.* [201] have described in detail the IoT attack model. Stoyanova *et al.* [25] later consolidated and classified the attacks. We have summarized and presented the most common IoT attack model in Figure 6.2. The main four vectors of attacks are (i) IoT hardware-based attacks, including physical dam-

age and power outage, (ii) attacks exploiting IoT software and applications, (iii) attacks on the IoT data, and (iv) attacks on the IoT communication protocols.

6.3.3 IoT Forensics Challenges

A comprehensive understanding of IoT forensics challenges is crucial to derive the IoT forensics requirements and thus model the framework. Therefore, this section analyzes the relevant and related research works, derives the forensics challenges, and categorizes them per the digital forensics phases.

Recall that in Chapter 2 we outlined digital forensics, cloud forensics, and IoT forensics and noted that IoT forensics is an extension of digital forensics applied to the IoT space. Further, in Chapter 3 we described the cloud forensics challenges, issues, and solutions. Some of the cloud forensics challenges are also applicable to IoT space, but the context varies. For example, data volatility in the cloud is due to the ephemeral nature of the cloud platforms, while in IoT, it is caused due to IoT capacity constraints. In the cloud, the main issue of evidence identification and acquisition is due to the mobility of data across virtualized platforms, while in IoT is caused due to the location and physical mobility of IoT.

Though there are similarities between cloud and IoT forensics, IoT poses additional and unique forensics problems, which many researchers have highlighted. Alenezi *et al.* [23] presented a review of IoT forensics challenges, IoT forensics frameworks and highlighted the need for organizational forensic readiness. MacDermott and Qi Shi [202] extended the IoT investigative methods to the Internet of Anything (IoA). As anything and everything soon will be "online", and highlighted the need for developing digital forensics standards that can be used for overall IoT and IoA security and forensics capability improvements. Servida and Casey [203] demonstrated extracting evidence traces from a few selected devices. They identified two main challenges in the forensics investigation of IoT devices (i) network traffic: as an increasing amount of traffic is encrypted. Though encryption is beneficial for security but limits the collection of interesting traces.

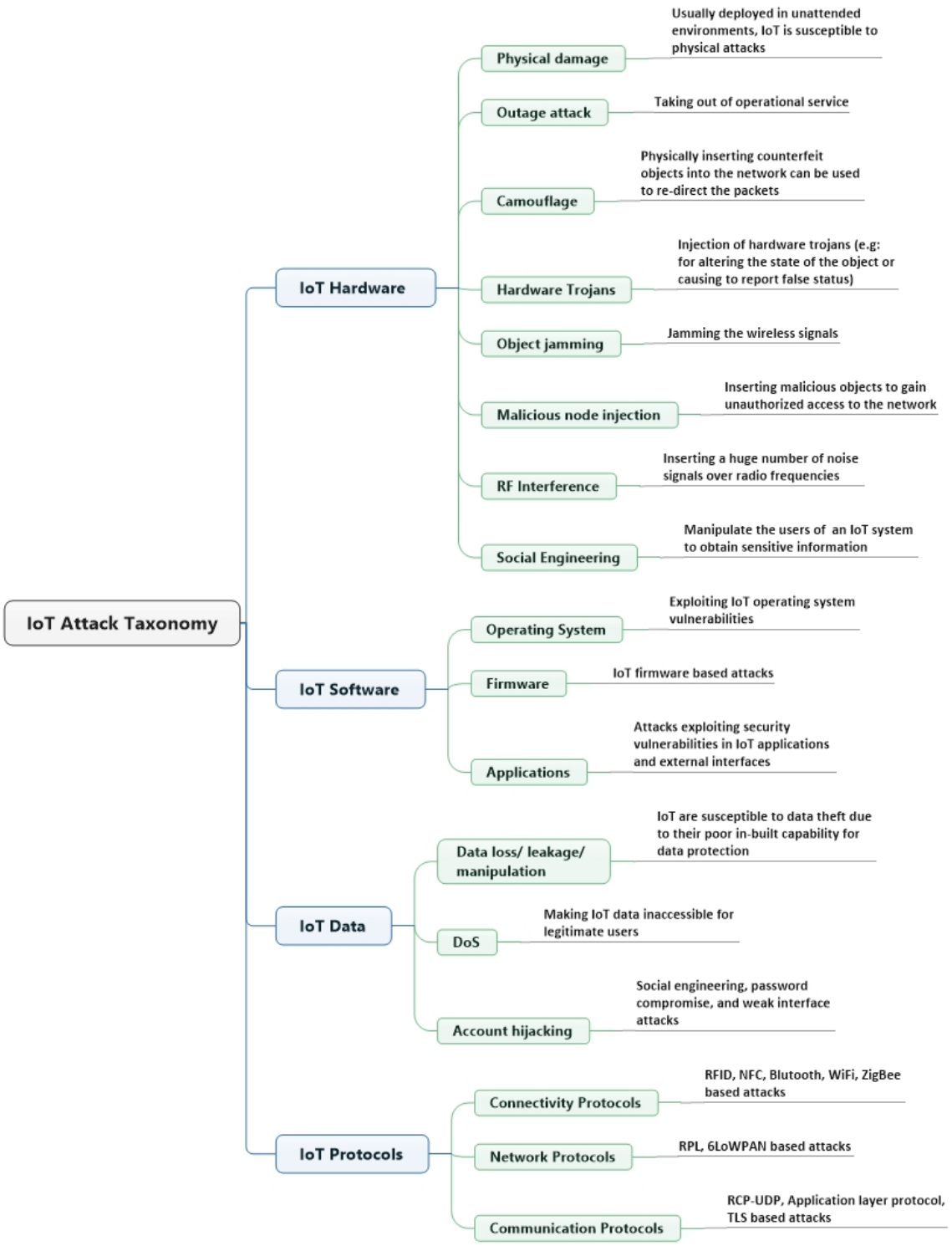


Figure 6.2: IoT attack taxonomy

(ii) devices level traces: the traces present on the physical device were limited to configuration settings, and the data of forensic value had limited persistence due to the limited storage. Ibrahim *et al.* [204] highlighted the identification of malicious IoT devices for "search & seizure" remains a critical issue. They proposed a formal method for IoT identification by fingerprinting the devices and their respective states so that it can guide the search & seizure. Oriwoh *et al.* [82] while highlighting the unique nature of IoT forensics, presented two approaches for IoT forensics (i) a zone-based investigative approach, where the zones are (a) internal network of IoT (b) middle zone (gateways, and boundary services) and (c) external network, and (ii) next best triage model, guiding 'where to look for' and the next best source of relevant evidence. Yaqoob *et al.* [205] explained novel factors affecting digital forensics in IoT and presented a taxonomy for IoT forensics. Stoyanova *et al.* [25] provided a comprehensive survey report on IoT forensics, challenges, approaches, and open issues. Analyzing and studying these research projects, we derived the IoT forensic challenges, and categorized them according to the different digital forensics phases. (Recall that digital forensic phases are described in Section 3.4). The challenges are described in Figure 6.3.

6.4 Problem Description and Modeling

This section describes and models the IoT forensics problem, leading to the design of the IoT forensics logging framework. The IoT forensics challenges outlined in Figure 6.3 describe the IoT forensics problems. We are using these problems to derive the IoT forensics requirements. Therefore, modeling the problem to address the requirements, in turn, addresses the challenges. The model mandates the creation of event record(s) satisfying the requirements. The characteristics of the event records are described below.

Let $\mathcal{IOT}$ represent an IoT network that contains a set of IoT devices. We denote $\mathcal{IOT}_i$ as the i^{th} IoT device, for $i = 1, 2, ..., n$. Therefore, an IoT network

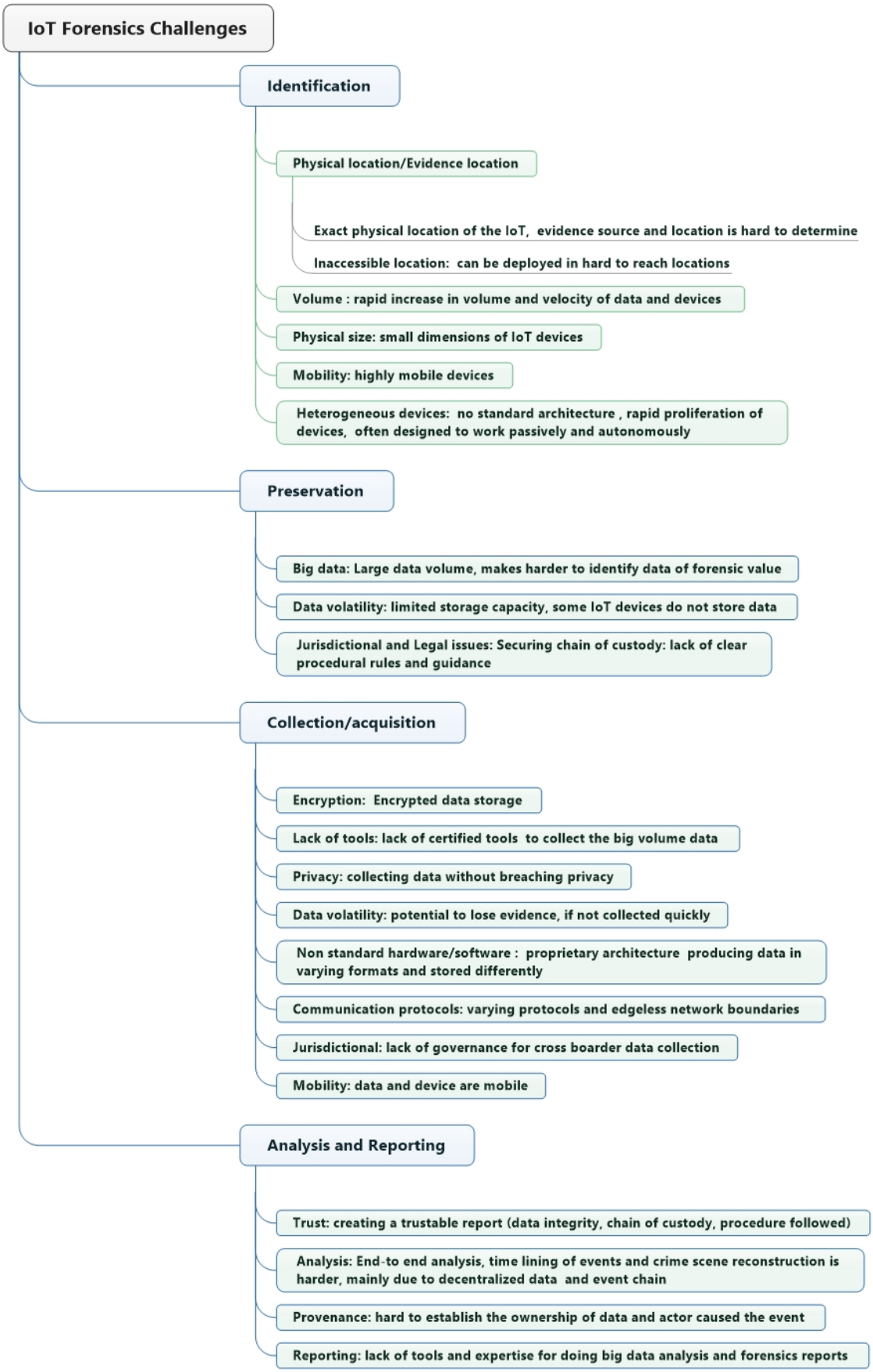


Figure 6.3: IoT forensics challenges

with n devices is denoted as

$$\mathcal{IOT} = \{\mathcal{IOT}_1, \mathcal{IOT}_2, \cdots \mathcal{IOT}_n\} \tag{6.1}$$

Each device provides one or more services. Let S_x denote a set of services provided by a device $x \in \mathcal{IOT}$. The set of l services by device x is represented as

$$S_x = \{S_{x,1}, S_{x,2}, \cdots S_{x,l}\} \tag{6.2}$$

For example, consider an $\mathcal{IOT}_x$ that provides a set of $l = 4$ services for patient health monitoring. Thus, we can have (i) $S_x,_1$: to monitor health parameters in real-time, (ii) $S_x,_2$: to transmit the values to the connected display device, (iii) $S_x,_3$: to raise alarms if the parameters fall outside of the acceptable values, and (iv) $S_x,_4$: to kick starts a control device to bring the values within the acceptable limit. We assume that an each IoT device provides at least one service.

When $\mathcal{IOT}_x$ executes service $S_{x,i}$, it generates one or more log records. We denote the log records for service i by LR_i. Therefore, LR_i can be represented as a set of one or more log records, i.e.,

$$LR_i = \{LR_{i,1}\,, LR_{i,2}\,.....LR_{i,n}\,\}. \tag{6.3}$$

Each log record i.e., $LR_{i,1}$ consists of a set of p parameters, which is defined as

$$LR_{i,1} = \{LP_{i,1,1}, LP_{i,1,2}.....LP_{i,1,p}\}. \tag{6.4}$$

The parameters are designed to satisfy the IoT forensic requirements. The requirements are defined in Section 6.5 and log parameters are defined in Section 6.6.1, specifically in Table 6.2. One log record produced during execution of an IoT service (i.e., $S_x,_i$) can meet one or more forensics requirements. Hence, a cumulative collection of log records forms evidence and supports forensics investigation.

6.5 The Framework: IoT Forensics Requirements

Recall that in Section 5.5 we mentioned that any forensic investigation aims to solve the crime by seeking answers to the main 5W1H cardinal questions. These cardinal questions applies to IoT forensics also [25, 30, 82, 91]. The goals and objectives of the IoT forensics investigation can be solved by finding answers to these cardinal questions. Further, it is worthwhile to cite the work of Atlam *et al.* [195], where the authors mentioned that cloud and IoT are two comparatively challenging technologies and a combination of two drastically change the future of the internet services, and cloud computing has resolved most of the IoT issues. On the same ground here, we are extending the log based solution applied in cloud forensics to IoT forensics as well. Note that we described extensively the log based approach for cloud forensics in Section 2.2.3, particularly in Table 2.1. The fundamental principle of answering 5W1H cardinal questions was used in the analysis of major cloud computing platforms' digital forensic readiness also [173]. Therefore, parallels can be drawn between cloud forensics and IoT forensics. Note that we provided a snapshot preview of cloud and IoT forensics in Section 2.3.1. Hence, the IoT forensic requirements can be derived by seeking answers to these cardinal questions in the context of IoT forensic challenges and attack vectors, described in Section 6.3, and taking relevant inputs from the cloud forensic logging requirements. The solution proposed in this work includes (i) defining the IoT forensic requirements (ii) designing the IoT systems for forensic readiness by embedding the forensic parameters in the log record, and (iii) transmitting the IoT log data to a persistent or long term storage in the cloud. Note that cloud forensics is one component of the IoT forensics as noted Section 6.3.1. Recall that in Section 4.3.1 we listed the cloud forensics logging requirements. Further, in Section 5.5 we referred to these requirements as core requirements as they are platform independent and address common digital investigative needs. Therefore, the same requirements are also valid for IoT forensics logging, but applied

to an IoT context. In addition, there are IoT specific requirements that are defined separately as a part of this project. Therefore, the IoT forensics logging requirements consists of core requirements listed in Table 4.1 and IoT specific requirements, listed Table 6.1.

Table 6.1: IoT specific forensics logging requirements.

Id	Title	Description
FR-1	Identification	The ability to uniquely identify a given device.
FR-2	Location	The ability to physically locate the device at any given time.
FR-3	Big data	The ability to capture and store the voluminous data generated by IoT.
FR-4	Data volatility	The ability to persist IoT generated data as long as necessary.
FR-5	Log format	The ability to record events in a uniform structure, irrespective of the device architecture.
FR-6	Data transmission	The ability to transmit data between IoT and cloud securely.
FR-7	Privacy	The ability to protect people's data which includes the ability to store evidence data per device and to collect the evidence data from the archive per device.
FR-8	DoS	The ability to detect when a device has been taken out of service.
FR-9	IoT Data	The ability to detect any access to IoT (including sign-in/sign-out), IoT data access or update (including configuration data updates).
FR-10	IoT Software	The ability to detect any abnormal behaviour of the IoT application(s).
FR-11	Compliance	The ability to satisfy applicable compliance rules.
FR-12	Configuration data	The ability to periodically report IoT configuration data baseline.

6.6 The Framework: Architecture

The key objective of the framework architecture is the capability to address the problem stated in Section 6.4. The proposed framework uses an integrated Cloud-IoT technology referred as as **CloudIoT**[206] to achieve this objective. Cloud and IoT are two complementary technologies and an integrated cloud-IoT infrastructure benefits from each other. The complementary features of the **CloudIoT** platform can be summarized as:

1. IoT is pervasive technology where things (i.e., real objects) can be anywhere. While the cloud is ubiquitous technology where resources are available from everywhere.

2. IoT has limited computational processing capabilities and limited or no storage. While the cloud has virtually unlimited computational and storage capabilities.

3. IoT uses the internet as a point of convergence. In contrast, the cloud uses the internet for service delivery.

4. IoT is a source of big data. In comparison, the cloud is a means to store and process big data.

An integrated cloud-IoT architecture (i.e., the **CloudIoT**) similar to the one proposed by Atlam *et al.* [195] enables the storage and analysis of the IoT generated forensics data and resolves some of the IoT forensics challenges (such as Bigdata, data volatility, evidence acquisition, etc.) outlined in Figure 6.3. The cloudIoT paradigm have been used in many IoT applications and industries [206]. In this work, we further extended the CloudIoT platform to support digital forensics in IoT and form the core architecture of our proposed framework. The high level architecture is shown in Figure 6.4. Different actions carried out by IoT generates events, i.e., the *event source*. The events are captured by the next layer, i.e.,

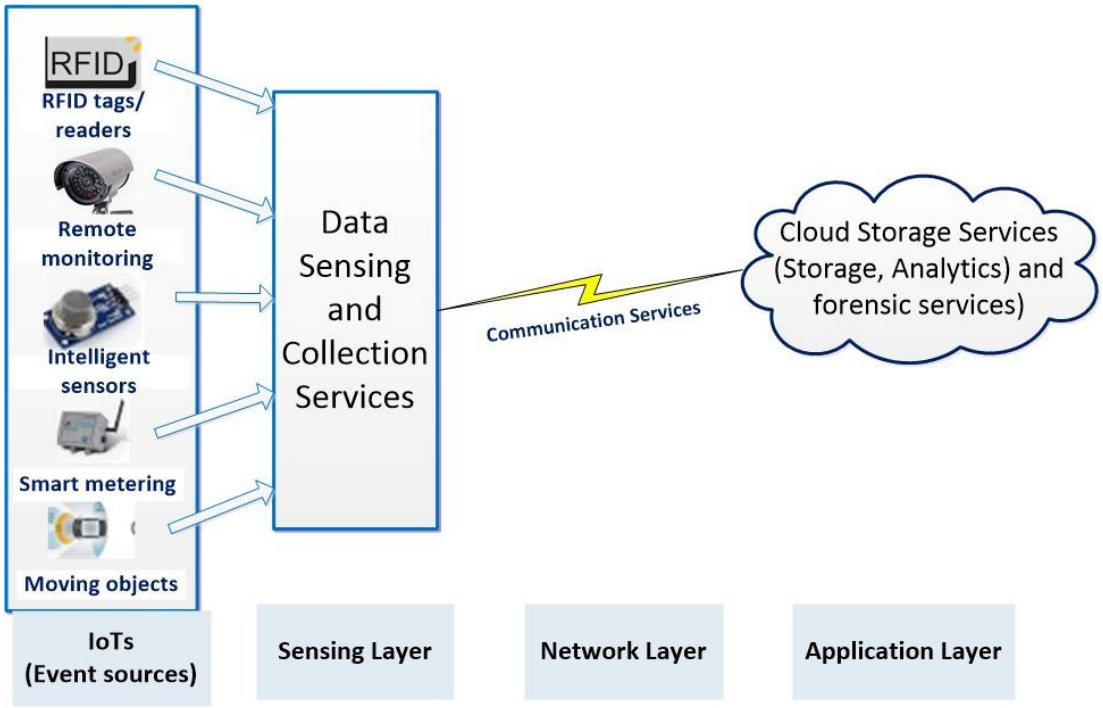


Figure 6.4: CloudIoT integrated architecture

Sensing layer. The Sensing Layer consists of the protocol stack and application services for event sensing, primary data aggregation, and validation. Here, we assume that several IoT devices are often required to provide a service. For example, a home automation system typically would have video monitoring devices, intrusion sensors, motion detectors, smart metering etc. Our model proposes that each action that IoT does as a part of its service provisioning should generate log records, embedded with forensic parameters satisfying forensic requirements. The Sensing Layer aggregates the data from those devices and transmits it to the cloud storage services.

The Application Layer consists of cloud-based storage and analytical services for forensic data analysis. The architecture proposes to use secure storage services (e.g: AWS CloudTrail). The cloud secure storage services provide the capacity to store big-data volume and enforce confidentiality and integrity.

6.6.1 The Log Parameters

To make the IoT systems forensic ready, they have to be designed and built with forensics capabilities in mind. This section describes the IoT log architecture that has been designed considering the IoT forensics needs. Recall that in Section 6.4 we explained that the log record consists of a set of log parameters. The parameters address the investigative needs. In the following paragraphs, we explain what parameters (i.e., LP_k) to transmit, when to transmit, and how to transmit to the cloud storage.

6.6.1.1 What to transmit

Table 6.2 describes what are the parameters to be generated and transmitted as a part of the log record.

Table 6.2: IoT forensics logging parameters.

Parameters	Description
Unique IoT ID	Unique ID (for example, IPv6 address or MAC address) used for fingerprinting and the physical identification of the IoT device. The ID is an important element to find out the source of the evidence and to trace the provenance.
Geo-location	Geo-location where the IoT device is physically located at a given time.
Timestamp	Timestamp logged in UTC is to identify **when** the event happened and to create time lining of events.
Application ID/Session ID	Application ID identifies the producer of the event. The application ID can be the ID of the embedded application running within the device. Session ID help to identify and to trace the event across multiple sessions of an application.
User Id	User Id of the logged-in user identifies the 'actor' responsible for the event. The combination of User Id and application id/session id helps to establish the provenance, i.e., to pin down the **who** factor.

Continued on next page

Parameters	Description
Impact	Impact level of the incidence. To prioritize how quickly the data to be collected.
Event Data	Provides description of the event. Typical event data field contents are described in Section 6.6.1.2

6.6.1.2 When to transmit

When to transmit a log record by IoT to the cloud services largely depends upon the business purpose that the IoT has been designed to support. However, other factors also affect the periodicity of the log transmission. In general, they are (i) operational events, (ii) security events, (iii) configuration data update events, (iv) compliance events, and (v) IoT data access events. These events are described in the following paragraphs, and they form the event data portion of the log record.

1. **Operational data events:** Operational data consists of the data generated during the regular operation of IoT devices. (Note that in Section 6.4 we provided an example of a normal functions of a medical IoT.) The following operational event data are captured as a minimum set.

    - *Error values:* IoT can generate 'error' as a result of regular operation, in case of unexpected events, errors generated by IoT applications, or as a result of an external command send to it. The severity of the error determines the critical conditions of business that the IoT is in-charge. From an investigation perspective an error log would help to find out any abnormal situation.

    - *Start, stop, re-start events:* Each of these events can be the result of a regular action (such as re-starting after a configuration data update), or as an indication of a severe problem. For example, an IoT device's unusual re-starting could indicate that the device could be compromised.

- *State change events:* Whenever there is a change of state in any of the values of the pre-defined parameters that the IoT has been deployed to monitor or control. ('Change of state' can be defined as a change in any parameters the IoT is monitoring. E.g., changing the on/off state of an air ventilator). Recording the change of state would help to find any abnormal behavior of the IoT or IoT software based attacks.

2. **Security events:** IoT security challenges, threats and vulnerabilities have been widely discussed and listed by researchers [78, 197, 207]. The main challenges for IoT security are from the heterogeneous nature and large scale foot print of the IoT devices. It is essential to transmit and log IoT security events such as:

   - *Authentication and authorization: Authentication* is the process of identifying an object or user. Passwords are the most commonly used method for user authentication [208]. Two-factor authentication using a combination of *what we have* such as tokens, or *what we are* such as bio-metric identification are used in addition to *what we know* such as passwords for enhanced security. *Authorization* is the process of identifying whether an entity (a user or an object) can access a resource, such as read/write data or execute programs. Authentication is a prerequisite for authorization. In machine-to-machine communication, cryptographic keys are commonly used as tokens for user authentication, and authorization [208]. In an IoT world, any user or device attempting to authenticate into IoT, accessing resources, changing passwords, revoking authorization parameters, configuration data updates, privileged account access, and activities executed by privileged accounts are all to be logged immediately. But the privileged account should **not be allowed** to change certain security protocols designed into the IoT, such as stopping the transmission of log data, changing the data security protocols (e.g.: removing the encryption

settings), or changing the parameter settings of the log data. Such actions can lead to detrimental security risks. Any attempts to perform such security violations must be logged immediately.

- *Software:* Normally, applications deployed in IoT always remain in execution mode, and it is one of the primary sources of events [205, 209]. Regular application events are not much value for forensics, but forensics being a post-incident activity, it is necessary to collect the data pro-actively. To this end, this model proposes to capture software start/stop and reset events, change of state events, and periodic software status signals as a minimum set. The periodicity of the software status signal depends upon the specific business purpose of the IoT. Recording such events would help to detect any IoT software abnormality [91, 205].
- *Periodic object identification:* IoT should be designed to transmit heartbeat signals identifying itself and its status at periodic intervals. The status signal is required to determine that an attacker has not silenced the IoT and taken it out of action [210]. Some business purposes might require that IoT to remain in passive mode for a long time and become active only when an activity is detected. This model insists on transmitting the heartbeat signals even during the passive mode.
- *Physical attack:* Since the IoT can be deployed anywhere, they are very much susceptible for physical attacks. Events related to possible physical attacks, including an attempt to remove or tamper the IoT or disconnect the power supply, should instantly trigger a security event transmission [91, 210].

3. **Configuration data update:** Configuration data are the software applications, hardware, or operational specific parameter data, stored inside IoT. Configuration data often determines the set limits and threshold val-

ues. For example, a glucose monitoring device would have minimum and maximum blood glucose levels stored as a part of configuration data. Any glucose level measurement outside the set values are treated as abnormal. Any updates or changes to the configuration data must be captured and logged with full details since the IoT behavior can be easily manipulated by changing the configuration data. Configuration data also includes:

- *New versions:* Deployment of new applications, firmware/operating systems, or new versions of the same.
- *Inventory:* IoT must be designed to periodically report an inventory list of all files (specifically executables and files with suspicious formats) in the device. Comparing what 'the IoT has' with 'what it should have' would signal malicious virus injections or unwanted file deposits [205]

4. **Compliance events:** Industrial compliance and regulatory rules can demand to record specific events. Often IoT is widely deployed in critical infrastructures (like power, water) or hospitals. All such industries are highly regulated and controlled by a compliance regime. The IoT deployed in a specific sector has to comply with the respective compliance laws, often requiring an audit trail of the IoT events. The frequency of transmission of event data and the data structure depends upon the industry compliance regulations and specific use cases. Compliance events also include re-certification events, described below:

   - *Re-certification events:* Re-certification of IoT and its software means certifying again that the IoT functions as designed. Periodic re certification ensures confidence in the device and its outputs. Re-certification of IoT is even more critical in an industrial control system. Otherwise, it could result in disastrous consequences. As in the case of 'Stuxnet' virus attack on industrial control systems, which was a turning point

in the history of cyber security. The details of the Stuxnet attack scenario and the damage done have been widely studied [211–213]. The studies show that malware such as Stuxnet can affect IoT controlling critical physical infrastructures, which implies that threats might extend to real lives [213]. In this particular instance, the malware completely damaged the centrifuges in a nuclear facility by spinning the centrifuges at a speed that they are not designed for and drastically changing the frequency of the spinning rate while reporting 'all good' status to the control system. The operators who were managing the facility, all that they saw on their dashboard was that everything was 'OK', whereas the physical systems were malfunctioning and breaking down. The malware was clever enough to re-play the unsuspicious pre-recorded input, instead of reporting the real values [211]. Therefore, the lessons learned from the Stuxnet incident re-enforces the need to periodically re-certify or re-calibrate the IoT and record the logs of such activities. The frequency of re-certification depends on factors such as the business purpose of IoT, regulatory compliance rules etc. From a forensics perspective, the IoT device re-certification data provides information regarding the recent 'good state' of the IoT.

5. **IoT Data:** Any updates or attempts to update or access the data stored in IoT must be captured and logged. Data events include read, write, update, and delete. Such events are of high significance and value for digital investigation.

6.6.1.3 How to transmit

IoT communication protocols lack standardization, and they uses numerous communication protocols. The IoT communication protocols and their comparative analysis has been summarized in the work of Al-Warawi *et al.* [83]. Regardless of the protocol used used for log transmission, the protocol should address and

satisfy the following criteria to support IoT forensics.

1. *Compressed data:* Data compression helps to optimise data transfer bandwidth.

2. *Secure transmission:* End-to-end encrypted data transfer to preserve confidentiality and integrity and hence, to establish the trust factor. IoT protocols 6LoWPAN, ZigBee, BLE, NFC, Z-Wave use the Advanced Encryption Standard (AES) [83].

3. *Normalized log data:* Normalized log data, such as data recorded as a key-value pair. For example, JSON formatted log records are easy to parse and decode and helps in event reconstruction, and digital investigations [82]. Key-value pair formatted data also helps the seamless and factual interpretation of the logs. Recall that in the Figure 4.2 we provided a snapshot of JSON formatted normalized log data.

4. *Storage:* The **CloudIoT** architecture described in Section 6.6 to capture and store the log data, enables to address the Big data and Data volatility issues. The architecture uses the cloud secure data storage services. To safeguard privacy, the log files are separated and unique IoT id, thus avoiding co-mingling of data and protecting privacy. (Recall that the principles explained in Section 4.1.1 shall be used to separate the files per IoT.)

Since the IoT event logs are stored in the cloud storage, the cloud forensic challenges, issues and solutions, specifically related to evidence data preservation and acquisition described in Chapter 3 are applicable to the IoT forensics too.

6.6.2 The Framework: Verification

This section verifies that the framework design meets the requirement specifications. The verification has been done by cross-referencing the framework's log

data parameters with the specifications. Table 6.3 illustrates the verification details and further strengthens the comprehensiveness of the proposed framework.

Table 6.3: Framework verification.

Log parameter	Requirements addressed	Comments
Unique IoT ID	FR-1 (Identification), R-7 (Provenance)	
Geo-location	FR-2 (Location)	
Timestamp	R-6 (Traceability)	Combination of unique ID, geo-location, and timestamp helps to determine the position of an IoT at a given time.
User Id	R-7 (Provenance), FR-9 (IoT Data)	
Application ID/Session ID	R-7 (Provenance)	An application can produce the event with no user signed in. if so, the provenance can be traced to the owner of the application.
Impact	FR-4 (Data volatility) , R-16 (Persistence)	Determines the priority of data collection to minimize any data loss.
Event data	R-1 (Auditability), R-4 (Justifiability), R-8 (Quality), R-17 (Relevance), R-18 (Reliability), R-19 (Sufficiency), R-5 (Event trail)	Event data along with other parameters contain full, comprehensive, and reliable information and, it forms an auditable record.
Operational data events	R-2 (Repeatability), R-3 (Reproducibility), FR-8 (DoS), FR-10 (IoT Software)	The framework clearly defines the event data sets to transmit and store. Hence, performing the same operational procedures produces the same/similar results.

Continued on next page

Log parameter	Requirements addressed	Comments
Security events	FR-9 (IoT Data), FR-10 (IoT Software), FR-8 (DoS)	Provides entire security event history leading to the incident.
IoT Data events	FR-9 (IoT Data)	Provides IoT data access/update events.
Configuration data	FR-12 (Configuration data)	Provides configuration data updates and installed inventory list.
Compliance events	FR-11 (IoT Compliance)	Records domain specific compliance logs
Secure transmission	FR-6 (Data transmission), R-10 (Trust)	
Normalized log data	FR-5 (Log format), R-20 (Usability), R-12 (Evidence correlation)	Normalized log structure enhances the usability and helps to correlate the evidence collected from diverse set of IoT.
Storage	FR-3 (Big data), FR-4 (Data volatility), FR-7 (Privacy), R-10 (Trust), R-13 (Evidence segregation), R-14 (Evidence identification), R-11 (Evidence collection), R-5 (Event trail)	The CloudIoT storage architecture resolves many IoT forensics issues.

6.7 The Framework: Design Stack

The proposed framework's design stack is shown in Figure 6.5. The design can be used to build software systems for the IoT forensics log architecture described in Section 6.4. The components of the design stack and their functions are described below.

1. **IoT layer:** The lowest layer of the design stack is the physical deployment of IoT devices, each one of them programmed to meet one or more business

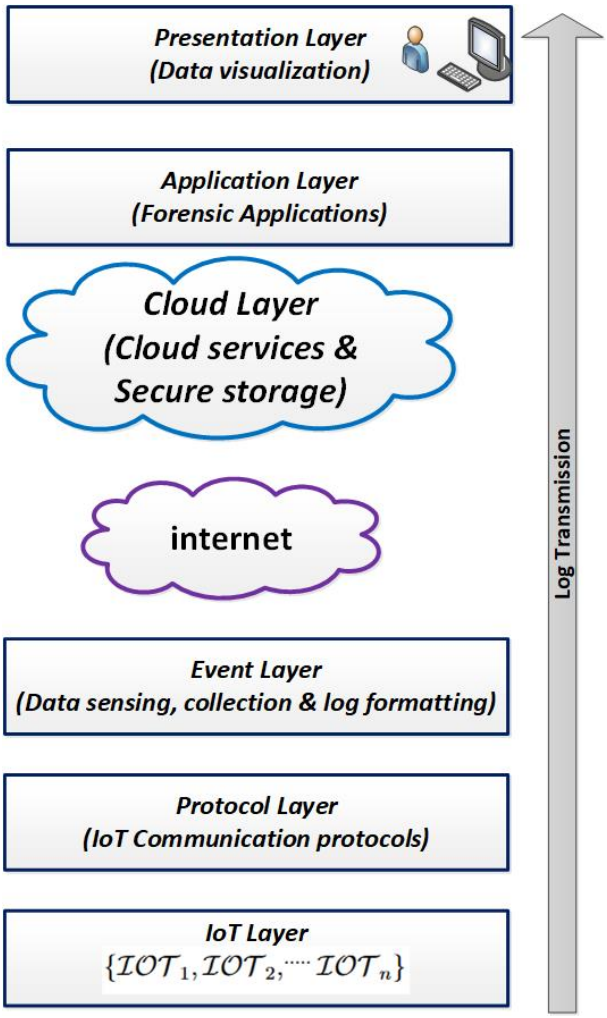


Figure 6.5: IoT log collection design stack

intend, and they are the event source.

2. **Protocol layer:** The Protocol layer comprises the IoT secure protocols (such as 6LoWPAN, ZigBee, BLE, NFC, Z-Wave, RFID, Bluetooth) used for carrying the data in/out of the IoT.

3. **Event layer:** The Event layer consists of applications responsible for capturing the events generated by IoT, formats the event data into a key-value pair (JSON structure) as described in the log architecture in Section 6.4. Further, the event layer encapsulates the data into a protocol stack and then transmits the data to cloud storage using secure internet protocols.

4. **Cloud layer:** The Cloud layer hosts the applications to stream the incoming IoT log data into a secure cloud storage bucket. In addition, the Cloud layer also enforces the confidentiality and integrity of the data, which can be achieved by utilizing the data storage security services provided by major cloud providers like Amazon AWS, Azure storage [114, 124, 141].

5. **Application layer:** The Application layer is the software stack consisting of forensic tools, applications, and scripts required to extract the relevant evidence from the cloud storage and align them along a timeline. Plus, the supporting tools to make forensic analysis easier for an investigator.

6. **Presentation layer:** The Presentation layer is the human interface part of the system, providing capabilities for the data manipulation and visualization of relevant forensics information.

6.8 Chapter Summary

The technology of IoT is a fast, modern approach that allows billions of intelligent devices to be interconnected and exchange information. The technology is being realized now at a rapid rate [82]. However, the large number, heterogeneous nature, size, location and rapid mobility of the IoT objects, and numerous communication protocols that IoT is using adds to the complexity of the IoT networks, as well as exponentially increases the forensics challenges [78, 82, 83, 197, 207].

Many elements must be designed and integrated into the IoT architecture for secure IoT systems, including incidence response and post-incidence forensics support. The fundamental objective of forensics investigation is to attribute an action to a doer and be able to prove that. This work presented an IoT logging framework and log collection stack, which can be designed into the IoT architecture to support digital forensics in IoT. First, we listed the IoT forensic challenges and defined the IoT forensic requirements. Subsequently, the requirements are used to design the framework, describe the method, and define the log param-

eters systematically. Further, the architecture described **when** to include these parameters in the logs. Later, we also verified the architecture, demonstrating the framework's suitability as a practical IoT forensics framework. The framework's architecture using CloudIoT integrated environment offers many benefits, overcoming the main limitations of IoT, such as low computing power, and storage while being a big data source. Having all the log data available in the cloud store helps easy acquisition of the event data from anywhere. Further, cloud storage enables data aggregation and long-term storage of the evidence data and can make the evidence data available for investigative purposes anytime. We believe that this model would help understand IoT forensics challenges and design and manufacture forensics capable IoT.

Chapter 7

Conclusion and Future Work

Rapid technological advancements in cloud computing and the Internet of Things have pushed the frontiers of digital forensics and exacerbated many technical, organizational and legal challenges of digital forensics. Nevertheless, cloud computing platforms also bring unique opportunities that can significantly advance the efficacy and speed of forensics investigation, such as the cost-effective computing power and storage resources, which can be used for quicker and voluminous data analysis. However, for the technologies to fully amalgamate for societal use, people's data and information assets must be secure in a digital space. Plus, the technologies must have the features and capabilities to support digital investigations.

The research presented in this thesis focuses on building and enhancing the digital forensics capabilities on the cloud and IoT platforms. We examined various issues and challenges that the cloud and IoT technologies pose to conduct digital forensics and proposed suitable forensics investigation models or frameworks. In the following paragraphs, we are highlighting the main contributions of this research project and future research opportunities.

Chapter 3 examined the cloud forensics process, challenges, and issues commonly faced in a digital investigation. After analyzing the most recent research

outputs, we presented the issues using the different phases of the digital investigation process as the base and proposed solutions to the challenges. The information presented in this chapter is expected to help the audience better understand the forensics investigations problems in the cloud. Briefly, the study concludes that (i) the cloud forensics issues are multi-dimensional, (ii) the digital investigation solutions are not yet mature, and some of them are very complex and resource intensive, and (iii) the traditional digital forensics process and methods are not suitable to the cloud environment. Future research opportunities include (i) developing specialized forensics technology tools specifically for the cloud, (ii) exploiting the elasticity characteristics of the cloud environment to store and process vast volumes of forensics data, i.e., big data analysis and (iii) extending such methods to provision "forensics-as-service". In addition, we also noticed that there is a strong need for forensically sound governance and processes to provide clear guidance for the cloud forensics practitioners, such as (i) guidelines specifically addressing cloud investigations, (ii) processes for trading services among CSPs, (iii) guidelines addressing access to the encryption keys to investigators without compromising privacy.

Chapter 4 formulated and presented a cloud forensics logging framework. Though there were many frameworks proposed earlier, we found that none of them addressed forensics practitioners' business needs and requirements, a well structured format and, data representation so that the information can be easily decoded and interpreted. Here, we tried to address it. The use of machine learning techniques for forensics data analysis, such as the correlation of logs and an automated approach for drawing temporal analysis of evidence, offers significant future research scope.

Chapter 5 assessed the cloud forensics readiness and maturity levels of three major cloud service providers (AWS, Azure and Google), following a systematic methodology and use case scenarios. Our work would help the cloud consumers and investigators to know in advance what to expect in terms of forensics compli-

ance. The study concluded that, in general, the cloud platforms are only partially forensics compliant, per February 2019. Generally, the lifetime of the evidence logs are determined by the user. Further, in some cases, the logs, are scattered in multiple locations and have varying formats with no uniformity in data representation. Therefore, the investigators need to have prior knowledge of 'what' to look for, 'where' to look, and 'how' to extract the evidence. Custom scripts to extract the evidence per user from the log archive obviously result in modifying the original data, which would make the integrity questionable. Since the logs are huge and voluminous, filtering out the noise or being able to collect the "needles from the haystack", is an absolute necessity. Future research work includes (i) developing evidence extraction techniques while preserving the integrity, (ii) developing evidence preservation techniques beyond the lifetime of the user, (iii) applying the proposed methodology to study the forensics readiness in the present context, and (iv) extending the case study with more complex use case scenarios.

Chapter 6 proposed an IoT forensics framework designed to capture the forensics artifacts using an integrated Cloud-IoT platform. IoT devices are most vulnerable to exploitation, and pose formidable challenges to forensics investigation. The challenges are mainly because of the IoTs' physical size, capacity limitations, location, mobility, and it is a source of big data. Lack of governance and standards to formalize IoT architecture is another major issue. The IoT forensics domain is in an infancy stage and there is substantial research potential. Few to mention are (i) defining and developing IoT specific forensics methods and processes, (ii) developing tools and technology for IoT based evidence collection, and (ii) defining standards and guidelines to formalize IoT architecture.

Appendices

Appendix A: Forensics Artifacts Identified in the CSP's Logs

Table A.1: Results and analysis of AWS CloudTrail logs.

AWS (CSP_x) CloudTrail ($S_{x,l}$) Forensics Artifacts				
Use Case Events (C_i)	**Evidence Traceability**	**Artifacts Generated ($F_{x,l,i}$)**	**Compliancy ($F_{x,l,i} \subset R_i$) or ($F_{x,l,i} \supseteq R_i$)**	**Compliance Gap**
User audit (Sign-in) events ($C_{i,1}$)	Yes	Sign-in events (success and failures) recorded, except root sign-in failures	($\otimes$) ($F_{x,l,i1}$) $\subset$ (R_{i1})	Low (2)
User audit (Sign-out) events ($C_{i,2}$)	No	Log out events are not recorded for all users (both root and IAM users)	($\times$) ($F_{x,l,i2}$) $\subset$ (R_{i2})	Medium (5)
Object search events ($C_{i,3}$)	Yes	Directory search events are listed	($\checkmark$) ($F_{x,l,i3}$) $\supseteq$ (R_{i3})	
Object operation events ($C_{i,4}$)	Partial	Object operational events are logged, but for object deletion done using different methods has different artifacts logged.	($\otimes$) ($F_{x,l,i4}$) $\subset$ (R_{i4})	Medium (6)

Continued on next page

Use Case Events (C_i)	**Evidence Traceability**	**Artifacts Generated ($F_{x,l,i}$)**	**Compliancy ($F_{x,l,i} \subset R_i$) or ($F_{x,l,i} \supseteq R_i$)**	**Compliance Gap**
Storage service operation events ($C_{i,5}$)	Yes	Object export events are logged similar to copy or download actions	(✓) $(F_{x,l,i5}) \supseteq (R_{i5})$	
Storage account operation events ($C_{i,6}$)	Yes	All events are logged.	(✓) $(F_{x,l,i6}) \supseteq (R_{i6})$	
Database (DynamoDB) events ($C_{i,7}$)	Partial	All DynamoDB table level activities are logged. But no events related to activities within a table are logged	($\otimes$) $(F_{x,l,i7}) \subset (R_{i7})$	Medium (7)
Database (RDS) events ($C_{i,8}$)	Yes	All DB events are logged	(✓) $(F_{x,l,i8}) \supseteq (R_{i8})$	
WebApp events ($C_{i,9}$)	No	Web app execution events are not logged n CloudTrail	($\times$) $(F_{x,l,9}) \subset (R_{i9})$	High (8)
Virtual server (EC2) events ($C_{i,10}$)	No	None of the EC2 events related to application access or manual access of the data is available in CloudTrail	($\times$) $(F_{x,l,10}) \subset (R_{10})$	High (9)
WorkMail events ($C_{i,11}$)	No	No events of forensics value, such as mail send,receive, date and time, recipients address, attachments (if any) etc. are not recorded	($\times$) $(F_{x,l,11}) \subset (R_{i11})$	High (10)

Table A.2: Results and analysis of Azure activity logs

MS Azure (CSP_y) logs ($S_{y,l}$) Forensics Artifacts				
Use Case Events (C_i)	**Evidence traceability**	**Artifacts generated ($F_{y,l,i}$)**	**Compliancy ($F_{y,l,i} \subset R_i$) or ($F_{y,l,i} \supseteq R_i$)**	**Compliance Gap**
User audit (Sign-in) events ($C_{i,1}$)	Yes (Azure Active Directory log ($S_{y,l1}$))	Sign-in events are logged including the failed sign-in attempts.	(✓)	
User audit (Sign-out) events ($C_{i,2}$)	No	Sign out events are not recorded in any logs	(×) $(F_{y,l,i2}) \subset (R_{i2})$	Medium (5)
Object search events ($C_{i,3}$)	Yes (Azure Activity log ($S_{y,l2}$))	Directory search events are listed	(✓) $(F_{y,l,i3}) \supseteq (R_{i3})$	
Object operation events ($C_{i,4}$)	Partial, (Azure Storage Explorer log ($\$log$)($S_{y,l3}$))	Almost all the information of forensics value are logged, except the user id of the actor.	(⊗) $(F_{y,l,i4}) \subset (R_{i4})$	Low (2)
Storage service operation events ($C_{i,5}$)	No	No events of forensics value recorded.	(×) $(F_{y,l,i5}) \subset (R_{i5})$	High (10)
Storage account operation events ($C_{i,6}$)	Yes (Activity logs ($S_{y,l2}$) and Storage Explorer logs ($S_{y,l3}$))	All events are logged	(✓) $(F_{y,l,i6}) \supseteq (R_{i6})$	
Database service (CosmosDB) events ($C_{i,7}$)	Partial	Events are logged in multiple places, i.e., Activity Logs and Diagnostic Logs	(⊗) $(F_{y,l,i7}) \subset (R_{i7})$	High (9)

Continued on next page

Use Case Events (C_i)	**Evidence traceability**	**Artifacts generated ($F_{y,l,i}$)**	**Compliancy ($F_{y,l,i} \subset R_i$) or ($F_{y,l,i} \supseteq R_i$)**	**Compliance Gap**
Database service (RDS) events ($C_{i,8}$)	Yes (Activity Log ($S_{y,l2}$))	All DB events are logged	(✓) $(F_{y,l,i8}) \supseteq (R_{i8})$	
WebApp events ($C_{i,9}$)	Yes (Azure AD logs ($S_{y,l1}$))	The events related to web app execution are logged in Azure AD logs	(✓) $(F_{y,l,i9}) \supseteq (R_{i9})$	
Virtual server events ($C_{i,10}$)	Yes (Azure Activity logs ($S_{y,l2}$))	All the Azure level activity events are logged in the Activity logs	(✓) $(F_{y,l,i10}) \supseteq (R_{i10})$	
Mail service (Office 365) events ($C_{i,11}$)	Partial (Office 365 Outlook audit logs ($S_{y,l4}$))	creating, sending, or receiving messages are not logged	(⊗) $(F_{y,l,i11}) \subset (R_{i11})$	High (9)
Mail service (MailJet) events ($C_{i,12}$)	Yes (MailJet statistics logs ($S_{y,l5}$))	Send mail details are logged in MailJet apps own logs	(✓) $(F_{y,l,i12}) \supseteq (R_{i12})$	

Table A.3: Results and analysis of Google Stackdriver logs

GCP Stackdriver (CSP_z) logs ($S_{z,l}$) Forensics Artifacts				
Use Case Events (C_i)	**Evidence traceability**	**Artifacts generated ($F_{z,l,i}$)**	**Compliancy ($F_{z,l,i} \subset R_i$) or ($F_{z,l,i} \supseteq R_i$)**	**Compliance Gap**
User audit (Sign-in) events ($C_{i,1}$)	No	Sign-in events are not logged	($\times$) $(F_{z,l,i1}) \subset (R_{i1})$	Medium (5)
User audit (Sign-out) events ($C_{i,2}$)	No	Sign out events are not recorded	($\times$) $(F_{z,l,i2}) \subset (R_{i2})$	Medium (5)
Object search events ($C_{i,3}$)	Yes	Directory search events are listed, (but not for contents inside files)	($\checkmark$) $(F_{z,l,i3}) \supseteq (R_{i3})$	
Object operation events ($C_{i,4}$)	Yes	All the information of forensics value are logged.	($\checkmark$) $(F_{z,l,i4}) \supseteq (R_{i4})$	
Storage service operation events ($C_{i,5}$)	Yes	All related events are logged	($\checkmark$) $(F_{z,l,i5}) \supseteq (R_{i5})$	
Storage account operation events ($C_{i,6}$)	Yes	All events are logged	($\checkmark$) $(F_{z,l,i6}) \supseteq (R_{i6})$	
Database service (Bigtable) events ($C_{i,7}$)	Partial	All admin level activities are logged but not data access events	($\otimes$) $(F_{z,l,i7}) \subset (R_{i7})$	Low (3)
Database service (Firestore) events ($C_{i,8}$)	Yes	All Firestore events are logged	($\checkmark$) $(F_{z,l,i8}) \supseteq (R_{i8})$	
Database service (RDS) events ($C_{i,9}$)	Yes	SQL operation events are logged	($\checkmark$) $(F_{z,l,i9}) \supseteq (R_{i9})$	
Web app events ($C_{i,10}$)	Partial	Partial info is available in the log	($\otimes$) $(F_{z,l,i10}) \subset (R_{i10})$	Low (2)
Virtual server (Compute or Kubernetes Engine) events ($C_{i,11}$)	No	No events related to the web app execution are logged	($\times$) $(F_{z,l,i11}) \subset (R_{i11})$	High (10)

Appendix B: Log Snapshots with Forensics Artifacts

Figure B.1: AWS CloudTrail log snapshot

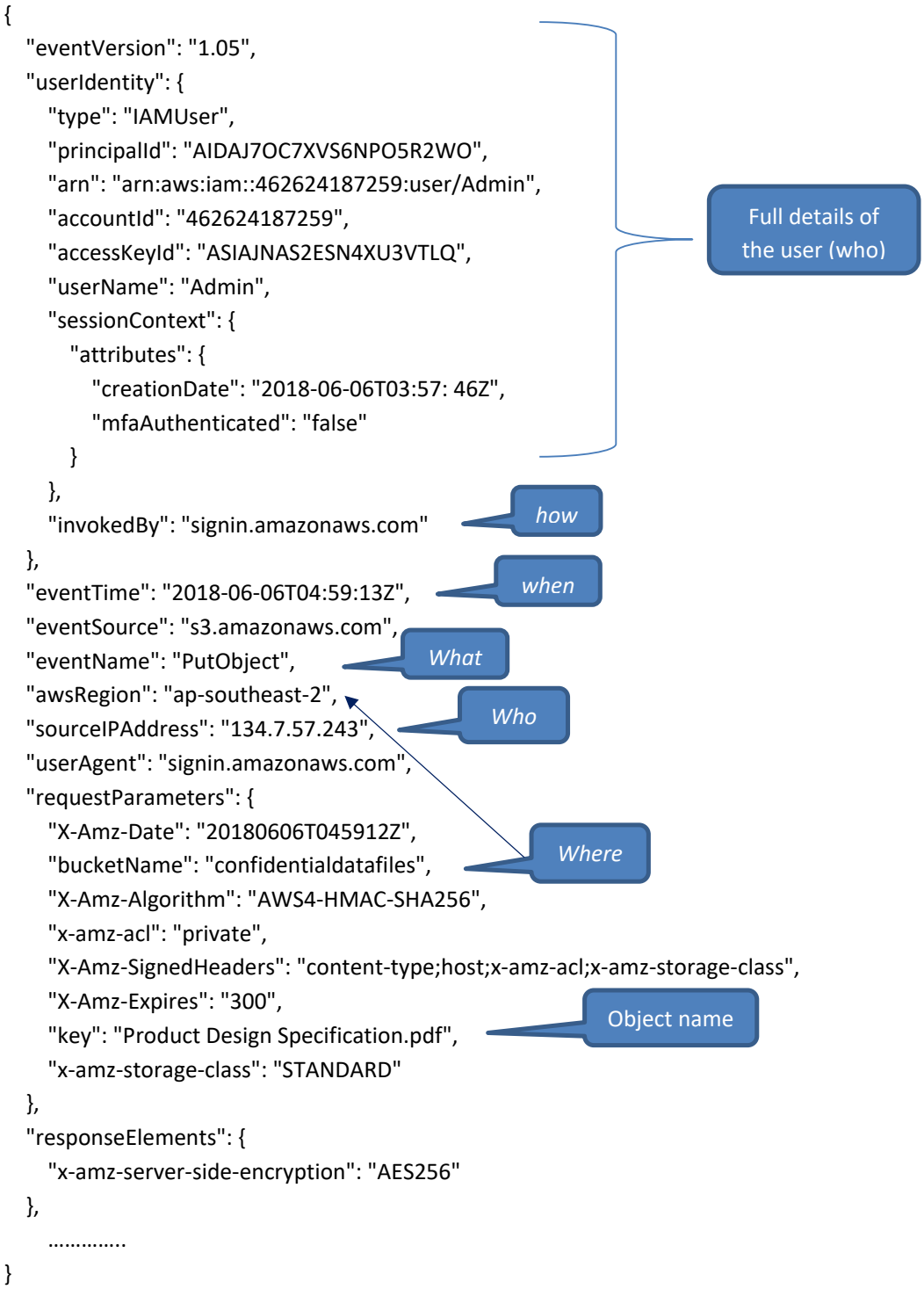


Figure B.2: Azure activity log snapshot

Figure B.3: GCP log snapshot

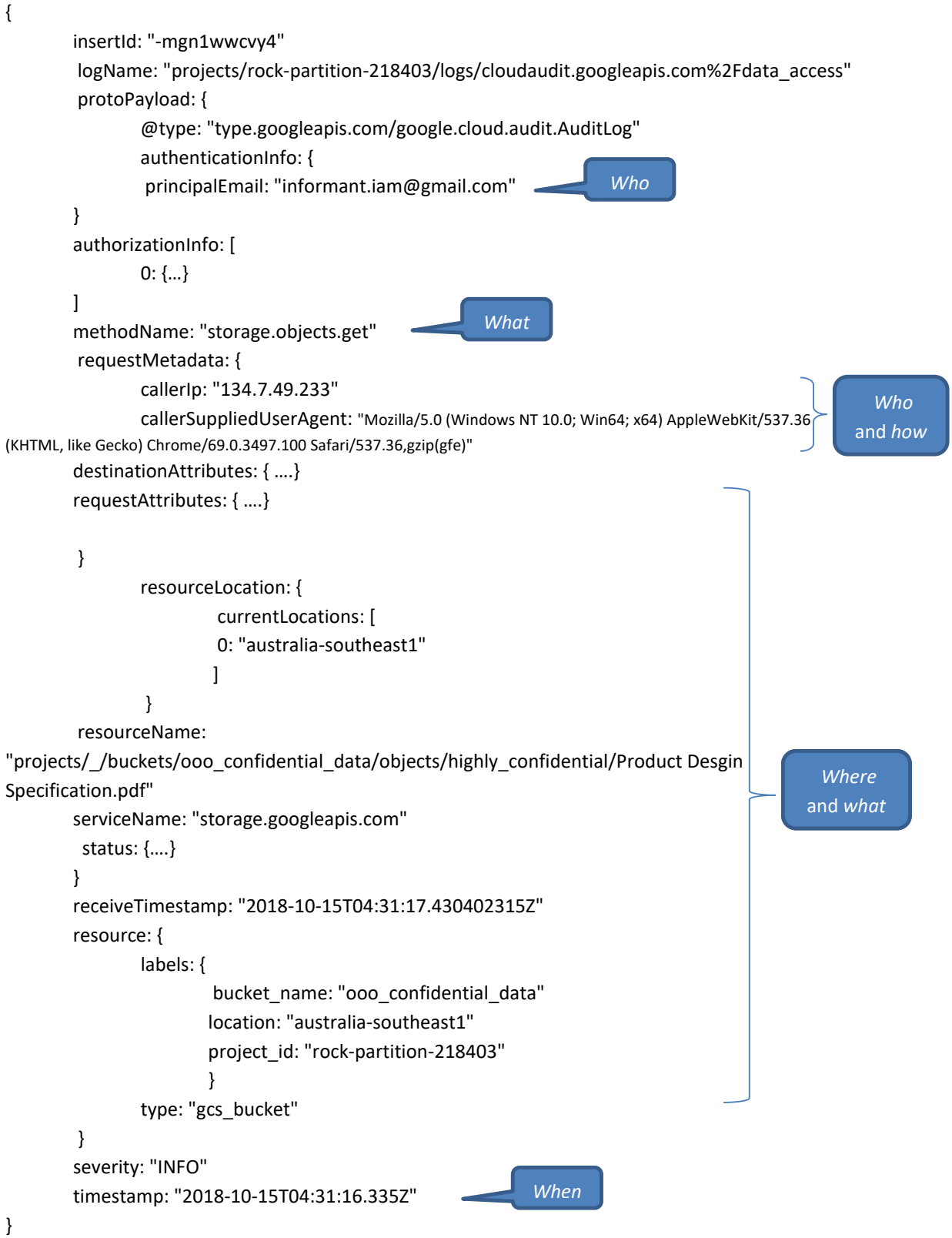
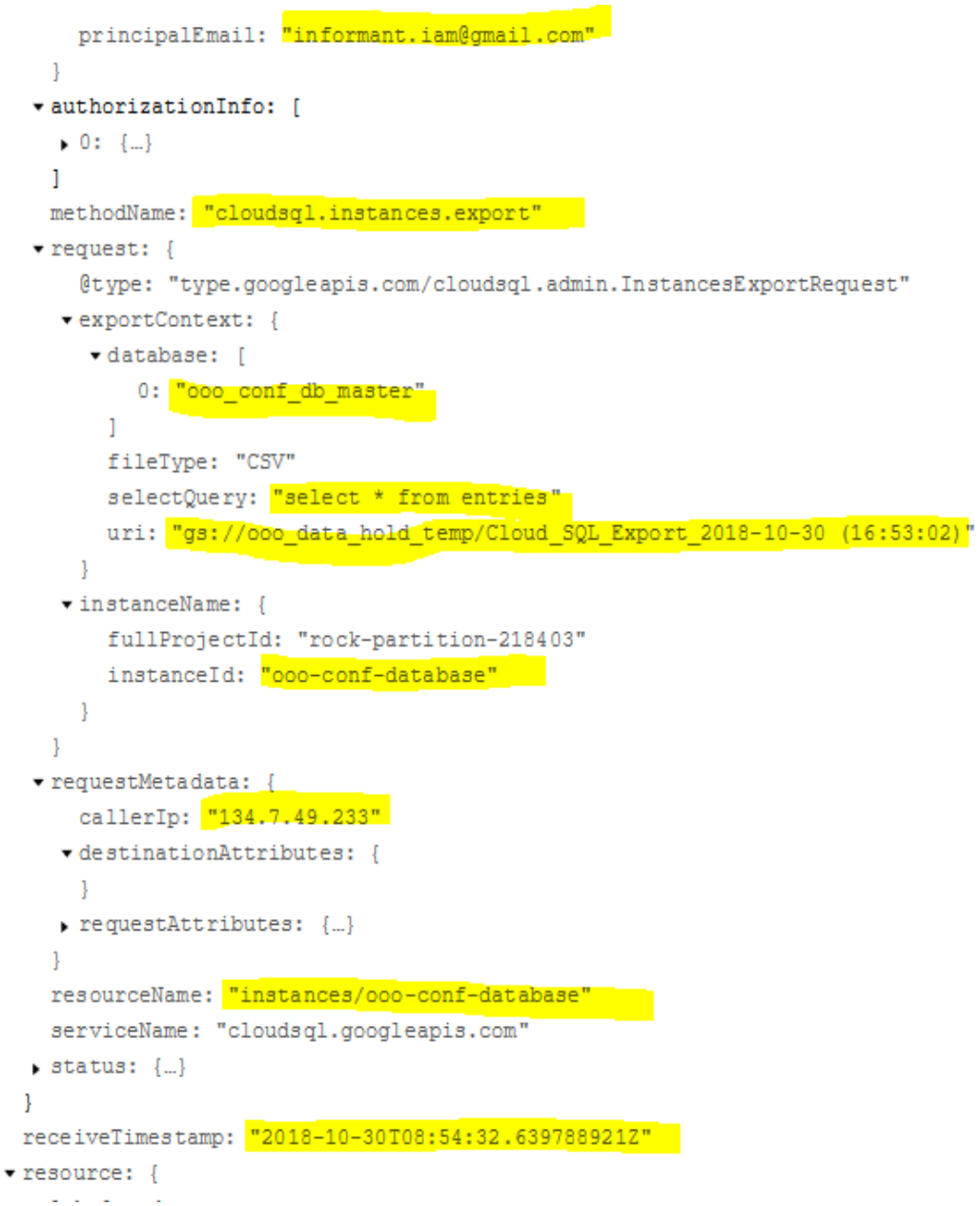


Figure B.4: Stackdriver database export snap short



Appendix C: Results of Cloud Forensics Logging

Figure C.1: Logs of administering CSU accounts


```
{"Records": [
        {
        "UTC_timestamp":"2016-03-3103:20:49",
        "user":"admin",
        "src_ip":"134.7.56.1",
        "src_port":"54510",
        "dest_ip": "134.7.57.9",
        "local_time":"2016-03-3111:20:49",
        "proto":"http",
        "file_params1":{
                "folder":"",
                "file_name":"",
                "size":“”
                },
        "file_params2":null,
        “location”:
        “Perth\Australia\Lat:32.0062\Long:115.8944”
        "action": "User logout"
        },
        {
        "UTC_timestamp":"2016-03-3108:20:18",
        "user":"admin",
        "src_ip":"134.7.56.1",
        "src_port":"54510",
        "dest_ip": "134.7.57.9",
        "local_time":"2016-03-3116:20:18",
        "proto":"http",
        "file_params1":{
                "folder":"",
                "file_name":  "user_hacker",
                "size":“”
                },
        "file_params2":null,
        “location”:
        “Perth\Australia\Lat:32.0062\Long:115.8944”
        "action": "user deleted"
        },
        {
        "UTC_timestamp":"2016-03-3103:18:18",
        "user":"admin",
        "src_ip":"134.7.56.1",
        "src_port":"54472",
        "dest_ip":“134.7.57.9",
        "local_time":"2016-03-3111:18:18",
        "proto":"http",
        "file_params1":{
                "folder":".",
                "file_name":  "user_hacker",
                "size":“”
                },
        "file_params2":null,
        “location”:
        “Perth\Australia\Lat:32.0062\Long:115.8944”
        "action": "New user created"
        },
        {
        "UTC_timestamp":"2016-03-2908:18:14",
        "user":"admin",
        "src_ip":"134.7.56.1",
        "src_port":"55083",  "
        "dest_ip": "134.7.57.9",
        "local_time":"2016-03-2916:18:14",
        "proto":"http",
        "file_params1":{
                "folder":"",
                "file_name":"",
                "size":“”
        },
        "file_params2":null,
        “location”:
        “Perth\Australia\Lat:32.0062\Long:115.8944”
        "action": "User login"
        },
        {
        "UTC_timestamp":"2016-03-2908:18:14",
        "user":"admin",
        "src_ip":"134.7.56.1",
        "src_port":"55083",
        "dest_ip": "134.7.57.9",
        "local_time":"2016-03-2916:18:14",
        "proto":"http",
        "file_params1":{
                "folder":"",
                "file_name":"",
                "size":“”
                },
        "file_params2":null,
        “location”:
        “Perth\Australia\Lat:32.0062\Long:115.8944”
        "action": "User login attempt"
        },
        {…….
        },
    ]
}
```

Figure C.2: Results of use case 1


```
{ "Records": [
        {
        "UTC_timestamp": "2016-03-31 07:38:45",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "61212",
        "dest_ip": "134.7.57.9",
        "local_time": "2016-03-31 15:38:45",
        "proto": "http",
         "file_params1": {
                  "folder": "",
                  "file_name": "",
                  "size": “”
                  },
        "file_params2": null,
        “location”:
        “Perth\Australia\Lat:32.0062\Long:115.8944”
        "action": "User logout"
        },
        {
        "UTC_timestamp": "2016-03-31 07:10:03",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "61093",
        "dest_ip": "134.7.57.9",
        "local_time": "2016-03-31 15:10:03",
        "proto": "http",
        "file_params1": {
                  "folder": "\/Documents\/App Storage",
                  "file_name":
                  "sample_pwd_cracker_output.xml",
                  "size": "568000"
                “location”:
                “Perth\Australia\Lat:32.0062\Long:115.8944”
                  },
                "file_params2": null,
                “location”:
                “Perth\Australia\Lat:32.0062\Long:115.8944”
                  "action": "File downloaded"
        },
        {
        "UTC_timestamp": "2016-03-31 07:07:57",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "61093",
        "dest_ip": "134.7.57.9",
        "local_time": "2016-03-31 15:07:57",
        "proto": "http",
        "file_params1": {
                "folder": "\/Documents\/App Storage",
                "file_name":
                "sample_pwd_cracker_output.xml",
                "size": "568000"
                  },
                  "file_params2": null,
                “location”:
                “Perth\Australia\Lat:32.0062\Long:115.8944”
        },
                        "action": "File written"
        {
        "UTC_timestamp": "2016-03-31 06:46:22",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "56156",
        "dest_ip": "134.7.57.9",
        "local_time": "2016-03-31 14:46:22",
        "proto": "http",
        "file_params1": {
                "folder": "\/Documents\/App Storage",
                "file_name":    "sample_pwd_cracker.exe",
                "size": "248000"
                  },
        "file_params2": null,
        “location”:
        “Perth\Australia\Lat:32.0062\Long:115.8944”
        "action": "File run"
        },
        {
        "UTC_timestamp": "2016-03-31 06:42:41",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "56100",
        "dest_ip": "134.7.57.9",
        "local_time": "2016-03-31 14:42:41",
        "proto": "http",
        "file_params1": {
                "folder": "\/Documents\/App Storage",
                "file_name":    "sample_pwd_cracker.exe",
                "size": "248000"
                  },
        "file_params2": null,
        “location”:
        “Perth\Australia\Lat:32.0062\Long:115.8944”
        "action": "File written"
        },
        {
        "UTC_timestamp": "2016-03-31 03:20:46",
        "user": "user_hacker",
        "src_ip": "134.7.56.1",
        "src_port": "54529",
        "dest_ip": "134.7.57.9",
        "local_time": "2016-03-31 11:20:46",
        "proto": "http",
        "file_params1": {
                  "folder": "",
                  "file_name": "",
                  "size": “”
                  },
        "file_params2": null,
        “location”:
        “Perth\Australia\Lat:32.0062\Long:115.8944”
         "action": "User login"
        },
   ]
}
```

Figure C.3: Results of use case 2


```
{"Records":[
    {
    "UTC_timestamp":"2016-06-08  14:04:24",
    "user":"User_Evil",
    "src_ip":"134.7.49.113",
    "src_port":"49294",
    "dest_ip":"134.7.57.9",
    "local_time":"2016-06-0822:04:24",
    "proto":"http",
    "file_params1":{
                "folder":"",
                "file_name":"",
                "size":“”
                 },
          "file_params2":null,
           “location”:
          “Perth\Australia\Lat:32.0062\Long:1
          15.8944”,
          "action":"User logout"
      },
      {
     "UTC_timestamp":"2016-06-08  14:04:15",
     "user":"User_Evil",
      "src_ip":"134.7.49.113",
     "src_port":"49293",
     "dest_ip":"134.7.57.9",
     "local_time":"2016-06-0822:04:15",
     "proto":"http",  "file_params1":{
                 "folder":""\/Documents\/Test_Site",
                 "file_name":"Desert.jpg",  "size":
                 “295752”
                 },
       "file_params2":null,
       “location”:
       “Perth\Australia\Lat:32.0062\Long:11
       5.8944”
       "action":"File shared"
     },
     {
     "UTC_timestamp":"2016-06-08  14:04:03",
     "user":"User_Evil",
      "src_ip":"134.7.49.113",
     "src_port":"49293",
      "dest_ip":"134.7.57.9",
     "local_time":"2016-06-0822:04:03",
     "proto":"http",
     "file_params1":{
                 "folder":"\/Documents\/Test_Site",
                 "file_name":"Penguins.jpg",
                 "size":"1387646"
                 },
     "file_params2":null,
      “location”:
     “Perth\Australia\Lat:32.0062\Long:115.
     8944”
     "action":"File shared"
     },
     {
     "UTC_timestamp":"2016-06-0814:01:34",
     "user":"User_Evil",
     "src_ip":"134.7.49.113",
     "src_port":"49278",
     "dest_ip":134.7.57.9",
     "local_time":"2016-06-0822:01:34",
     "proto":"http",
     "file_params1":{
           "folder":"\/Documents\/Test_Site"
           "file_name":"Penguins.jpg",
           "size":"1387646"
           },
       "file_params2":null,
       “location”:
       “Perth\Australia\Lat:32.0062\Long:11
       5.8944”
       "action":"File written"
       },
       {
       "UTC_timestamp":"2016-06-0814:00:50",
       "user":"User_Evil",
       "src_ip":"134.7.49.113",
       "src_port":"49272",
        "dest_ip":"134.7.57.9",
       "local_time":"2016-06-0822:00:50",
       "proto":"http",
       "file_params1":{
           "folder":"\/Documents\/Test_Site"
           "file_name":"Desert.jpg",   "size":"295752"
           },
     "file_params2":null,
     “location”:
     “Perth\Australia\Lat:32.0062\Long:115.
     8944”,
     "action":"File written"
     },
     {
     "UTC_timestamp":"2016-06-0813:59:13",
     "user":"User_Evil",
     "src_ip":"134.7.49.113",
     "src_port":"49237",
     "dest_ip":"134.7.57.9"
     "local_time":"2016-06-0821:59:13",
     "proto":"http",
     "file_params1":{
           "folder":"",
           "file_name":"",
           "size":“”
            },
     "file_params2":null,
     “location”:
     “Perth\Australia\Lat:32.0062\Long:115.
     8944”
     "action":"User login"
   }
   {……..
   },
  ]
}
```

Appendix D: Copyright Information

The copyright agreements for published papers allow authors to re-use their material in derivative works. Thus, copyright permission is not required. The following copyright information was obtained from IEEE, Elsevier and Inderscience publishers in which the author has published.

IEEE COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

A Logging Model for Enabling Digital Forensics in IoT, in an Inter-connected IoT-Cloud Eco-systems
Ameer Pichan, Mihai Lazarescu and Sie Teng Soh
2020 Fourth World Conference on Smart Trends in Systems Security and Sustainablity (WorldS4)

COPYRIGHT TRANSFER

The undersigned hereby assigns to The Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to: (a) the Work, including any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work; and (b) any associated written or multimedia components or other enhancements accompanying the Work.

GENERAL TERMS

1. The undersigned represents that he/she has the power and authority to make and execute this form.
2. The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
3. The undersigned agrees that publication with IEEE is subject to the policies and procedures of the IEEE PSPB Operations Manual.
4. In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall be null and void. In this case, IEEE will retain a copy of the manuscript for internal administrative/record-keeping purposes.
5. For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.
6. The author hereby warrants that the Work and Presentation (collectively, the "Materials") are original and that he/she is the author of the Materials. To the extent the Materials incorporate text passages, figures, data or other material from the works of others, the author has obtained any necessary permissions. Where necessary, the author has obtained all third party permissions and consents to grant the license above and has provided copies of such permissions and consents to IEEE

You have indicated that you DO wish to have video/audio recordings made of your conference presentation under terms and conditions set forth in "Consent and Release."

CONSENT AND RELEASE

1. In the event the author makes a presentation based upon the Work at a conference hosted or sponsored in whole or in part by the IEEE, the author, in consideration for his/her participation in the conference, hereby grants the IEEE the unlimited, worldwide, irrevocable permission to use, distribute, publish, license, exhibit, record, digitize, broadcast, reproduce and archive, in any format or medium, whether now known or hereafter developed: (a) his/her presentation and comments at the conference; (b) any written materials or multimedia files used in connection with his/her presentation; and (c) any recorded interviews of him/her (collectively, the "Presentation"). The permission granted includes the transcription and reproduction of the Presentation for inclusion in products sold or distributed by IEEE and live or recorded broadcast of the Presentation during or after the conference.
2. In connection with the permission granted in Section 1, the author hereby grants IEEE the unlimited, worldwide, irrevocable right to use his/her name, picture, likeness, voice and biographical information as part of the advertisement, distribution and sale of products incorporating the Work or Presentation, and releases IEEE from any claim based on right of privacy or publicity.

BY TYPING IN YOUR FULL NAME BELOW AND CLICKING THE SUBMIT BUTTON, YOU CERTIFY THAT SUCH ACTION CONSTITUTES YOUR ELECTRONIC SIGNATURE TO THIS FORM IN ACCORDANCE WITH UNITED STATES LAW, WHICH AUTHORIZES ELECTRONIC SIGNATURE BY AUTHENTICATED REQUEST FROM A USER OVER THE INTERNET AS A VALID SUBSTITUTE FOR A WRITTEN SIGNATURE.

Ameer Pichan	08-08-2020
Signature	**Date (dd-mm-yyyy)**

Information for Authors

AUTHOR RESPONSIBILITIES

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements as stated in section 8.2.1 of the IEEE PSPB Operations Manual, including provisions covering originality, authorship, author responsibilities and author misconduct. More information on IEEE's publishing policies may be found at http://www.ieee.org/publications_standards/publications/rights/authorrightsresponsibilities.html Authors are advised especially of IEEE PSPB Operations Manual section 8.2.1.B12: "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it." Authors are also advised of IEEE PSPB Operations Manual section 8.1.1B: "Statements and opinions given in work published by the IEEE are the expression of the authors."

RETAINED RIGHTS/TERMS AND CONDITIONS

- Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.
- Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.
- Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use.The IEEE Intellectual Property Rights office must handle all such third-party requests.
- Authors whose work was performed under a grant from a government funding agency are free to fulfill any deposit mandates from that funding agency.

AUTHOR ONLINE USE

- **Personal Servers**. Authors and/or their employers shall have the right to post the accepted version of IEEE-copyrighted articles on their own personal servers or the servers of their institutions or employers without permission from IEEE, provided that the posted version includes a prominently displayed IEEE copyright notice and, when published, a full citation to the original IEEE publication, including a link to the article abstract in IEEE Xplore. Authors shall not post the final, published versions of their papers.
- **Classroom or Internal Training Use.** An author is expressly permitted to post any portion of the accepted version of his/her own IEEE-copyrighted articles on the author's personal web site or the servers of the author's institution or company in connection with the author's teaching, training, or work responsibilities, provided that the appropriate copyright, credit, and reuse notices appear prominently with the posted material. Examples of permitted uses are lecture materials, course packs, e-reserves, conference presentations, or in-house training courses.
- **Electronic Preprints.** Before submitting an article to an IEEE publication, authors frequently post their manuscripts to their own web site, their employer's site, or to another server that invites constructive comment from colleagues. Upon submission of an article to IEEE, an author is required to transfer copyright in the article to IEEE, and the author must update any previously posted version of the article with a prominently displayed IEEE copyright notice. Upon publication of an article by the IEEE, the author must replace any previously posted electronic versions of the article with either (1) the full citation to the

Article:	**Towards a Practical Cloud Forensics Logging Framework**
Corresponding author:	**Mr Ameer Pichan**
E-mail address:	**ameer.pichan@postgrad.curtin.edu.au**
Journal:	**Journal of Information Security and Applications**
Our reference	**JISA2235**
PII:	**S2214-2126(17)30520-3**
DOI:	**10.1016/j.jisa.2018.07.008**

the Preprint on any website or repository at any time;
the Accepted Manuscript on certain websites and usually after an embargo period;
the Published Journal Article only privately on certain websites, unless otherwise agreed by Elsevier Ltd.

In the case of the Accepted Manuscript and the Published Journal Article the Author Rights exclude Commercial Use (unless expressly agreed in writing by Elsevier Ltd), other than use by the author in a subsequent compilation of the author's works or to extend the Article to book length form or re-use by the author of portions or excerpts in other works (with full acknowledgment of the original publication of the Article).

AUTHOR REPRESENTATIONS / ETHICS AND DISCLOSURE / SANCTIONS

I affirm the Author Representations noted below, and confirm that I have reviewed and complied with the relevant Instructions to Authors, Ethics in Publishing policy, Declarations of Interest disclosure and information for authors from countries affected by sanctions (Iran, Cuba, Sudan, Burma, Syria, or Crimea). Please note that some journals may require that all co-authors sign and submit Declarations of Interest disclosure forms. I am also aware of the publisher's policies with respect to retractions and withdrawal (https://www.elsevier.com/about/our-business/policies/article-withdrawal).
For further information see the publishing ethics page at https://www.elsevier.com/about/our-business/policies/publishing-ethics and the journal home page. For further information on sanctions, see https://www.elsevier.com/about/our-business/policies/trade-sanctions

Author representations

The Article I have submitted to the journal for review is original, has been written by the stated authors and has not been previously
The Article was not submitted for review to another journal while under review by this journal and will not be submitted to any other
The Article and the Supplemental Materials do not infringe any copyright, violate any other intellectual property, privacy or other righ
entity, or contain any libellous or other unlawful matter.
I have obtained written permission from copyright owners for any excerpts from copyrighted works that are included and have credit
Article or the Supplemental Materials.
Except as expressly set out in this Journal Publishing Agreement, the Article is not subject to any prior rights or licenses and, if my o
institution has a policy that might restrict my ability to grant the rights required by this Journal Publishing Agreement (taking into ac
permitted hereunder, including Internal Institutional Use), a written waiver of that policy has been obtained.
If I and/or any of my co-authors reside in Iran, Cuba, Sudan, Burma, Syria, or Crimea, the Article has been prepared in a personal, a
capacity and not as an official representative or otherwise on behalf of the relevant government or institution.
If I am using any personal details or images of patients, research subjects or other individuals, I have obtained all consents required
complied with the publisher's policies relating to the use of such images or personal information. See https://www.elsevier.com/abou
business/policies/patient-consent for further information.
Any software contained in the Supplemental Materials is free from viruses, contaminants or worms.
If the Article or any of the Supplemental Materials were prepared jointly with other authors, I have informed the co-author(s) of the
Publishing Agreement and that I am signing on their behalf as their agent, and I am authorized to do so.

GOVERNING LAW AND JURISDICTION

This Agreement will be governed by and construed in accordance with the laws of the country or state of Elsevier Ltd ("the Governing State"), without regard to conflict of law principles, and the parties irrevocably consent to the exclusive jurisdiction of the courts of the Governing State.

For information on the publisher's copyright and access policies, please see http://www.elsevier.com/copyright.
For more information about the definitions relating to this agreement click here.

☑ **I have read and agree to the terms of the Journal Publishing Agreement.**

31st July 2018 T-copyright-v22/2017

INDERSCIENCE PUBLISHERS
www.inderscience.com
PUBLISHERS OF DISTINGUISHED ACADEMIC, SCIENTIFIC AND PROFESSIONAL JOURNALS

Author Copyright Agreement

Inderscience Enterprises Ltd, trading as Inderscience Publishers, of World Trade Center Building II, 29 Route de Pre-Bois, Case Postale 856, CH-1215 Genève 15, Switzerland ("Inderscience")

If your article has been accepted for publication, each author must sign a copyright agreement form after reading the Explanatory Notes below and either *(for online submissions)* follow the online instructions or *(for email submissions)* send the signed forms, in electronic format, to the Editor (or other recipient as advised by the Editor of the specific journal), together with the final version of the article.

So that we can ensure both the widest dissemination and protection of material published in Inderscience's journals, we ask authors to assign copyright in their articles, including abstracts, to Inderscience. This enables us to ensure copyright protection against infringement, and to disseminate your article, and our journals, as widely as possible.

1. In consideration of the undertaking set out in paragraph 2, and upon acceptance by Inderscience for publication in the Journal, [insert the full names of all authors, reflecting the name order given in the article]

   Ameer Pichan, Mihai Lazarescu, Sie Teng Soh

   hereafter 'the Author' hereby assigns and transfers to Inderscience, the copyright in and to [insert article title]

   Case Study On Major Cloud Platforms Digital Forensics Readiness - Are We There Yet?

   hereafter 'the Article' by the Author to be published in [insert journal title]

   Int. J. of Cloud Computing

   hereafter ('the Journal'). This assignment provides Inderscience the sole right and responsibility to publish the Article, including the right to sub-license publishing or distribution rights to the Article as may be appropriate, in both printed and electronic form; the Article may be published in printed, online, CD-ROM, microfiche or in other media formats.

2. In consideration of this assignment, Inderscience hereby undertakes to prepare and publish the Article named in paragraph 1 in the Journal, subject only to its right to refuse publication as provided in paragraph 5 or if there are other reasonable grounds; in such case Inderscience reverts and assigns to the Author any and all copyright and other rights in the Article otherwise assigned to it under this Agreement.

3. The Editor of the Journal and Inderscience are empowered to make such editorial changes as may be necessary to make the Article suitable for publication. Every effort will be made to consult the Author if substantive changes are required.

4. The Author hereby asserts his/her moral rights under the UK Copyright Designs and Patents Act 1988 to be identified as the Author of the Article.

5. The Author warrants that the Article is the Author's original work, it has not been published before either in full or in part and is not currently under consideration for publication elsewhere; and that the Article contains no libellous or unlawful statements and that it in no way infringes the rights of others, nor it is in breach of any English law, and that the Author, as the owner of the copyright, is entitled to make this assignment. If the Author is the Corresponding Author*, the Author warrants that where s/he enters into any correspondence about or agrees to any changes to the Article s/he is authorised to act on behalf of any co-authors in doing so and has provided full and accurate information relating to them where required on the understanding that no further changes can be made after signature of this Agreement.

* The Author designated in the published Article as the individual to contact in the event of an enquiry about a manuscript. The Corresponding Author normally is responsible for correcting page proofs and working with the production editor.

Ameer Pichan

Digitally signed by Ameer Pichan
DN: cn=Ameer Pichan, o=IAEA, ou=IAEA, email=ameerp57@gmail.com, c=AT
Date: 2020.01.16 23:15:19 +01'00'

Signed by the Author .....

Date. 16 Jan 2020

INDERSCIENCE PUBLISHERS
www.inderscience.com
PUBLISHERS OF DISTINGUISHED ACADEMIC, SCIENTIFIC AND PROFESSIONAL JOURNALS

Author Copyright Agreement: Explanatory Notes

Inderscience's policy is to acquire copyright for all contributions, for the following reasons:

a. ownership of copyright by a central body helps to ensure maximum international protection against infringement and/or plagiarism;
b. requests for permission to reproduce articles in books, course packs, electronic reserve or for library loan can be handled centrally, relieving authors of a time-consuming administrative burden;
c. the demand for research literature in electronic form can be met efficiently, with proper safeguards for authors, editors and journal owners.

There are opportunities to reach institutions (e.g. companies, schools and public libraries) and individual readers that are unlikely to subscribe to the printed Journal. Inderscience works with other organisations to publish its journals online, and to deliver copies of individual articles. It has registered the Journal with the Copyright Licensing Agency, which offers centralised licensing arrangements for digital copying and photocopying around the world. Income received from all of these sources is used to further the interests of the Journal.

Once accepted for publication, your Article will be published in the Journal, and will be stored and distributed electronically, in order to meet increasing library and faculty demand and to deliver it as part of the Journal, as an individual article or as part of a larger collection of articles to meet the specific requirements of a particular market. By signing this Author Copyright Agreement and assigning copyright you agree to Inderscience making such arrangements.

It may be that the Author is not able to make the assignment solely by him- or herself:

a. If it is appropriate, the Author's employer may sign this agreement. The employer may reserve the right to use the Article for internal or promotional purposes (by indicating on this agreement) and reserve all rights other than copyright.
b. If the Author is a UK Government employee, the Government will grant a non-exclusive licence to publish the Article in the Journal in any medium or form provided that Crown Copyright and user rights (including patent rights) are reserved. This also applies to other Commonwealth countries.
c. If the Author is a US Government employee and the work was done in that capacity, the assignment applies only to the extent allowed by US law.

Under the UK's Copyright Design and Patents Act 1988, the Author has the moral right to be identified as the Author wherever the Article is published, and to object to its derogatory treatment or distortion. Inderscience encourages assertion of this right; it represents best publishing practice and is an important safeguard for all authors. Paragraph 4 above asserts the Author's moral rights, as required by the Act.

Authors can use their Article for non-commercial purposes after publication in these ways:

- Posting the *Author's Original** on the Author's personal or departmental web pages and/or institutional repositories and/or subject repositories without embargo and sharing it as much as desired. For open [freely available] repositories, if the manuscript was funded by either RCUK or the Wellcome Trust, use the CC-BY-NC: Creative Commons Attribution-NoDerivs 4.0. Otherwise, follow the licensing restrictions applied to all material copyrighted by Inderscience;
- Posting the *Accepted Manuscript**
  - Internally sharing the Accepted Manuscript within their research collaboration groups only, at any point after publication
  - Posting the Accepted Manuscript on institutional repositories and/or subject repositories, subject to an embargo of 12 months after publication (Green Open Access)
  - Posting the Accepted Manuscript on academic social networks or social media subject to an embargo of 24 months after publication (Green Open Access)

Note for authors of articles funded by Research Councils UK (RCUK) and Wellcome Trust and other governmental organisations: If you are required to deposit your accepted manuscript into your institutional repository within 90 days of acceptance and our embargo period is longer than that permitted by your funder, please choose Gold Open Access. If this is not possible for you, please speak to your institution about applying for an exception to HEFCE's Research Excellence Framework policy.

- Posting the *Version of Record** to a subject-based repository such as PubMed Central *only* in cases where a funding agency providing the grant for the research on which the Article is based requires this of the Author, upon condition that it shall not be accessible until after six months from Inderscience's publication date. The PDF of the VoR should not be posted anywhere else unless it has been published as Open Access. This also applies to any Author who has published with Inderscience in the past;
- Using the article in further research and in courses that the Author is teaching;
- Incorporating the article content in other works by the Author.

In all cases, acknowledgement in the form of a full citation must be given to the journal as the original source of publication, together with a link to the journal webpage and/or DOI as soon as they are available.

*Versions of a paper defined as

- **Author's Original** = Author's manuscript prior to peer review [often called a 'preprint']
- **Accepted Manuscript** = Accepted version of author's manuscript, accepted for publication, i.e. post-review, pre-typesetting. *We recommend retaining this version for future posting.*
- **Version of Record** = Publisher's version of finished article

Inderscience, as Publisher, reserves the right to refuse to publish your Article where its publication creates – or it reasonably believes may create – legal liability, or where circumstances come to light that were not known to the Editor, including prior publication of the whole or part of the Article, conflict of interest, manifest error, etc. Inderscience is the ultimate custodian of academic quality and integrity, and will ensure that this will be done only in exceptional circumstances and on reasonable grounds. In such circumstances the Article will be returned to the Author together with all rights in it.

Thank you for reading these notes. If you require more detailed information, go to Inderscience's web site. This assignment will enable Inderscience to ensure that the Article will reach the optimum readership.

Inderscience Enterprises Ltd, trading as Inderscience Publishers, of World Trade Center Building II, 29 Route de Pre-Bois, Case Postale 856, CH-1215 Genève 15, Switzerland ("Inderscience")

Bibliography

[1] ReportLinker, "Cloud Computing Market - Global Forecast to 2026," *[online] https://www.reportlinker.com/p05749258/?utm_source=GNW [Accessed: 20.01.2022]*, 2021.

[2] Fortune Business Insights, "Green Technology and Sustainability Global Market Analysis, Insights And Forecast, 2017 –2028," *[online] https://www.fortunebusinessinsights.com/cloud-computing-market-102697, [Accessed: 20.09.2021]*, 2021.

[3] P. Purnaye and V. Kulkarni, "A comprehensive study of cloud forensics," *Archives of Computational Methods in Engineering*, vol. 29, no. 1, pp. 33–46, 2022.

[4] K. Ruan, J. Carthy, T. Kechadi, and M. Crosbie, "Cloud forensics," in *IFIP International Conference on Digital Forensics*. Springer, 2011, pp. 35–46.

[5] Info security, "Infosecurity, Strategy, Insight, Technology," *[online] https://www.infosecurity-magazine.com/news/cybercrime-costs-orgs-per-minute [Accessed: 20.01.2022]*, 2021.

[6] K. Marshall and A. Rea, "Legal challenges in cloud forensics," *AMCIS 2021 Proceedings*, 2021.

[7] K. Prasad, "Cyberterrorism: addressing the challenges for establishing an international legal framework," 2012.

[8] IBM Security, "Cost of data breach report 2021," Tech. Rep., 2021.

[9] E. A. Vincze, "Challenges in digital forensics," *Police Practice and Research*, vol. 17, no. 2, pp. 183–194, 2016.

[10] G. Palmer, "A road map for digital forensics research-report from the first digital forensics research workshop (dfrws)," *Utica, New York*, 2001.

[11] A. Ghosh, D. De, and K. Majumder, "A systematic review of log-based cloud forensics," *Inventive Computation and Information Technologies*, pp. 333–347, 2021.

[12] S. Simou, C. Kalloniatis, E. Kavakli, and S. Gritzalis, "Cloud forensics: identifying the major issues and challenges," in *International conference on advanced information systems engineering*. Springer, 2014, pp. 271–284.

[13] S. A. Ali, S. Memon, and F. Sahito, "Challenges and solutions in cloud forensics," in *Proceedings of the 2018 2nd International Conference on Cloud and Big Data Computing*, 2018, pp. 6–10.

[14] Gartner, "Cloud market share 2018," *[online] https://www.gartner.com/en/newsroom/press-releases/2018-08-01-gartner-says-worldwide-iaas-public-cloud-services-market-grew-30-percent-in-2017 [Accessed: 20.01.2022]*, 2018.

[15] Synergy, "Cloud market growth growth and segment leaders 2019," *[online] https://www.srgresearch.com/articles/half-yearly-review-shows-150-billion-spent-cloud-services-and-infrastructure [Accessed: 20.01.2022]*, 2019.

[16] The Register, "Cloud a three-player market dominated by AWS, Google, Microsoft," *[online]*

https://www.theregister.com/2022/05/02/cloud$_m$arket$_s$hare$_q$1$_2$022/[*Accessed* : 20.01.2022], 2022.

[17] H. Shanmuganathan and A. Mahendran, "Current trend of iot market and its security threats," in *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*. IEEE, 2021, pp. 1–9.

[18] V. P. Gupta, "Smart sensors and industrial iot (iiot): A driver of the growth of industry 4.0," in *Smart Sensors for Industrial Internet of Things*. Springer, 2021, pp. 37–49.

[19] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on iot security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82 721–82 743, 2019.

[20] P. M. Chanal and M. S. Kakkasageri, "Security and privacy in iot: a survey," *Wireless Personal Communications*, vol. 115, no. 2, pp. 1667–1693, 2020.

[21] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying iot security: an exhaustive survey on iot vulnerabilities and a first empirical look on internet-scale iot exploitations," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2702–2733, 2019.

[22] A. Hameed and A. Alomary, "Security issues in iot: a survey," in *2019 International conference on innovation and intelligence for informatics, computing, and technologies (3ICT)*. IEEE, 2019, pp. 1–5.

[23] A. Alenezi, H. Atlam, R. Alsagri, M. Alassafi, and G. Wills, "Iot forensics: a state-of-the-art review, challenges and future directions," in *The 4th International Conference on Complexity, Future Information Systems and Risk (COMPLEXIS 2019)*, 2019, pp. 106–115.

[24] N. H. Ab Rahman, K.-K. R. Choo *et al.*, "Integrating digital forensic practices in cloud incident handling: A conceptual cloud incident handling model," in *The cloud security ecosystem : technical, legal, business and management issues*. Syngress Publishing, 2015, pp. 383–400.

[25] M. Stoyanova, Y. Nikoloudakis, S. Panagiotakis, E. Pallis, and E. K. Markakis, "A survey on the internet of things (iot) forensics: Challenges, approaches and open issues," *IEEE Communications Surveys & Tutorials*, 2020.

[26] S. Zawoad and R. Hasan, "Faiot: Towards building a forensics aware eco system for the internet of things," in *2015 IEEE International Conference on Services Computing*. IEEE, 2015, pp. 279–284.

[27] V. R. Kebande and I. Ray, "A generic digital forensic investigation framework for internet of things (iot)," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*. IEEE, 2016, pp. 356–362.

[28] V. R. Kebande, N. M. Karie, A. Michael, S. Malapane, I. Kigwana, H. Venter, and R. D. Wario, "Towards an integrated digital forensic investigation framework for an iot-based ecosystem," in *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*. IEEE, 2018, pp. 93–98.

[29] D. Quick and K.-K. R. Choo, "Iot device forensics and data reduction," *IEEE Access*, vol. 6, pp. 47 566–47 574, 2018.

[30] K. S. Rahman, M. Bishop, and A. Holt, "Internet of things mobility forensics," *de Proceedings of the 2016 Information Security Research and Education (INSuRE)*, 2016.

[31] M. M. Hossain, R. Hasan, and S. Zawoad, "Trust-iov: A trustworthy forensic investigation framework for the internet of vehicles (iov)." in *ICIOT*, 2017, pp. 25–32.

[32] C. Meffert, D. Clark, I. Baggili, and F. Breitinger, "Forensic state acquisition from internet of things (fsaiot) a general framework and practical approach for iot forensics through iot device state acquisition," in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, 2017, pp. 1–11.

[33] L. Babun, A. K. Sikder, A. Acar, and A. S. Uluagac, "Iotdots: A digital forensics framework for smart environments," *arXiv preprint arXiv:1809.00745*, 2018.

[34] G. L. Palmer, "A road map for digital forensics research-report from the first digital forensics research workshop (dfrws)(technical report dtr-t001-01 final)," *Air Force Research Laboratory, Rome Research Site, Utica*, pp. 1–48, 2001.

[35] J. Williams, "Acpo good practice guide for digital evidence," *Metropolitan Police Service, Association of chief police officers, GB*, 2012.

[36] M. Herman, M. Iorga, A. M. Salim, R. H. Jackson, M. R. Hurst, R. Leo, R. Lee, N. M. Landreville, A. K. Mishra, Y. Wang *et al.*, "Nist cloud computing forensic science challenges (nistir 8006)," *https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8006.pdf*, 2020.

[37] A. Pichan, M. Lazarescu, and S. T. Soh, "Cloud forensics: Technical challenges, solutions and comparative analysis," *Digital investigation*, vol. 13, pp. 38–57, 2015.

[38] W. A. Jansen, T. Grance *et al.*, "Guidelines on security and privacy in public cloud computing," *NIST Special Publication 800-144*, 2011.

[39] P. Mell, T. Grance *et al.*, "The nist definition of cloud computing," *NIST Special Publication 800-145*, 2011.

[40] vmware documentation, "vmware," *[online] https://www.vmware.com/topics/glossary/content/hypervisor/[Accessed: 08.08.2021]*, 2021.

[41] R. K. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "Trustcloud: A framework for accountability and trust in cloud computing," in *2011 IEEE World Congress on Services*. IEEE, 2011, pp. 584–588.

[42] K. Kent, S. Chevalier, T. Grance, and H. Dang, "Guide to integrating forensic techniques into incident response," *NIST Special Publication 800-86*, vol. 10, no. 14, 2006.

[43] M. Damshenas, A. Dehghantanha, R. Mahmoud, and S. bin Shamsuddin, "Forensics investigation challenges in cloud computing environments," in *Proceedings Title: 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*. IEEE, 2012, pp. 190–194.

[44] H. Guo, B. Jin, and T. Shang, "Forensic investigations in cloud environments," in *2012 International Conference on Computer Science and Information Processing (CSIP)*. IEEE, 2012, pp. 248–251.

[45] B. Hay, K. Nance, and M. Bishop, "Storm clouds rising: security challenges for iaas cloud computing," in *2011 44th Hawaii International Conference on System Sciences*. IEEE, 2011, pp. 1–7.

[46] D. Birk and C. Wegener, "Technical issues of forensic investigations in cloud computing environments," in *2011 Sixth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*. IEEE, 2011, pp. 1–10.

[47] D. Reilly, C. Wren, and T. Berry, "Cloud computing: Pros and cons for computer forensic investigations," *International Journal Multimedia and Image Processing (IJMIP)*, vol. 1, no. 1, pp. 26–34, 2011.

[48] S. D. Wolthusen, "Overcast: Forensic discovery in cloud environments," in *2009 Fifth International Conference on IT Security Incident Management and IT Forensics*. IEEE, 2009, pp. 3–9.

[49] B. Manral, G. Somani, K.-K. R. Choo, M. Conti, and M. S. Gaur, "A systematic survey on cloud forensics challenges, solutions, and future directions," *ACM Computing Surveys (CSUR)*, vol. 52, no. 6, pp. 1–38, 2019.

[50] D. Catteddu, G. Hogben *et al.*, "Cloud computing risk assessment," *European Network and Information Security Agency (ENISA)*, pp. 583–592, 2009.

[51] A. Alenezi, R. K. Hussein, R. J. Walters, and G. B. Wills, "A framework for cloud forensic readiness in organizations," in *2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. IEEE, 2017, pp. 199–204.

[52] R. Marty, "Cloud application logging for forensics," in *proceedings of the 2011 ACM Symposium on Applied Computing*, 2011, pp. 178–184.

[53] A. Patrascu and V.-V. Patriciu, "Logging system for cloud computing forensic environments," *Journal of Control Engineering and Applied Informatics*, vol. 16, no. 1, pp. 80–88, 2014.

[54] S. Zawoad, A. K. Dutta, and R. Hasan, "Seclaas: secure logging-as-a-service for cloud forensics," in *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security*, 2013, pp. 219–230.

[55] S. Rane and A. Dixit, "Blockslaas: Blockchain assisted secure logging-as-a-service for cloud forensics," in *International Conference on Security & Privacy*. Springer, 2019, pp. 77–88.

[56] B. K. Raju, N. B. Gosala, and G. Geethakumari, "Closer: Applying aggregation for effective event reconstruction of cloud service logs," in *Proceedings*

of the 11th International Conference on Ubiquitous Information Management and Communication, 2017, pp. 1–8.

[57] M. N. A. Khan and S. Ullah, "A log aggregation forensic analysis framework for cloud computing environments," *Computer Fraud & Security*, vol. 2017, no. 7, pp. 11–16, 2017.

[58] K. Awuson-David, T. Al-Hadhrami, M. Alazab, N. Shah, and A. Shalaginov, "Bcfl logging: An approach to acquire and preserve admissible digital forensics evidence in cloud ecosystem," *Future Generation Computer Systems*, 2021.

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

[60] S. Alqahtany, N. Clarke, S. Furnell, and C. Reich, "A forensic acquisition and analysis system for iaas," *Cluster Computing*, vol. 19, no. 1, pp. 439–453, 2016.

[61] M. E. Alex and R. Kishore, "Forensics framework for cloud computing," *Computers & Electrical Engineering*, vol. 60, pp. 193–205, 2017.

[62] V. R. Kebande and H. S. Venter, "On digital forensic readiness in the cloud using a distributed agent-based solution: issues and challenges," *Australian Journal of Forensic Sciences*, vol. 50, no. 2, pp. 209–238, 2018.

[63] S. Zawoad, A. K. Dutta, and R. Hasan, "Towards building forensics enabled cloud through secure logging-as-a-service," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 148–162, 2015.

[64] V. R. Kebande and H. Venter, "A cloud forensic readiness model using a botnet as a service," in *The international conference on digital security and*

forensics (DigitalSec2014). Ostrava: The Society of Digital Information and Wireless Communication, 2014, pp. 23–32.

[65] V. R. Kebande and H. S. Venter, "Novel digital forensic readiness technique in the cloud environment," *Australian Journal of Forensic Sciences*, vol. 50, no. 5, pp. 552–591, 2018.

[66] L. De Marco, M.-T. Kechadi, and F. Ferrucci, "Cloud forensic readiness: Foundations," in *International Conference on Digital Forensics and Cyber Crime*. Springer, 2013, pp. 237–244.

[67] N. H. Ab Rahman, W. B. Glisson, Y. Yang, and K.-K. R. Choo, "Forensic-by-design framework for cyber-physical cloud systems," *IEEE Cloud Computing*, vol. 3, no. 1, pp. 50–59, 2016.

[68] V. Kebande and H. Venter, "A functional architecture for cloud forensic readiness large-scale potential digital evidence analysis," in *European Conference on Cyber Warfare and Security*. Academic Conferences International Limited, 2015, p. 373.

[69] Z. Qi, C. Xiang, R. Ma, J. Li, H. Guan, and D. S. Wei, "Forenvisor: A tool for acquiring and preserving reliable data in cloud live forensics," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 443–456, 2016.

[70] P. M. Trenwith and H. S. Venter, "Digital forensic readiness in the cloud," in *2013 Information Security for South Africa*. IEEE, 2013, pp. 1–5.

[71] G. Sibiya, T. Fogwill, H. S. Venter, and S. Ngobeni, "Digital forensic readiness in a cloud environment," in *2013 Africon*. IEEE, 2013, pp. 1–5.

[72] F. Ye, Y. Zheng, X. Fu, B. Luo, X. Du, and M. Guizani, "Tamforen: A tamper-proof cloud forensic framework," *Transactions on Emerging Telecommunications Technologies*, p. e4178, 2020.

[73] K. Ruan and J. Carthy, "Cloud forensic maturity model," in *International Conference on Digital Forensics and Cyber Crime*. Springer, 2012, pp. 22–41.

[74] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure provenance: the essential of bread and butter of data forensics in cloud computing," in *Proceedings of the 5th acm symposium on information, computer and communications security*, 2010, pp. 282–292.

[75] O. Q. Zhang, M. Kirchberg, R. K. Ko, and B. S. Lee, "How to track your data: The case for cloud computing provenance," in *2011 IEEE Third International Conference on Cloud Computing Technology and Science*. IEEE, 2011, pp. 446–453.

[76] AWS, "Amazon web services overview of security processes," Tech. Rep., 2020.

[77] K. Gulati, R. S. K. Boddu, D. Kapila, S. L. Bangare, N. Chandnani, and G. Saravanan, "A review paper on wireless sensor network techniques in internet of things (iot)," *Materials Today: Proceedings*, 2021.

[78] K. Zhao and L. Ge, "A survey on the internet of things security," in *2013 Ninth international conference on computational intelligence and security*. IEEE, 2013, pp. 663–667.

[79] M. Chernyshev, Z. Baig, O. Bello, and S. Zeadally, "Internet of things (iot): Research, simulators, and testbeds," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1637–1647, 2017.

[80] S. Chakrabarty and D. W. Engels, "A secure iot architecture for smart cities," in *2016 13th IEEE annual consumer communications & networking conference (CCNC)*. IEEE, 2016, pp. 812–813.

[81] A. Ghosh, K. Majumder, and D. De, "A systematic review of digital, cloud and iot forensics," *The" Essence" of Network Security: An End-to-End Panorama*, pp. 31–74, 2021.

[82] E. Oriwoh, D. Jazani, G. Epiphaniou, and P. Sant, "Internet of things forensics: Challenges and approaches," in *9th IEEE International Conference on Collaborative computing: networking, Applications and Worksharing*. IEEE, 2013, pp. 608–615.

[83] S. Al-Sarawi, M. Anbar, K. Alieyan, and M. Alzubaidi, "Internet of things (iot) communication protocols," in *2017 8th International conference on information technology (ICIT)*. IEEE, 2017, pp. 685–690.

[84] S. Li, K.-K. R. Choo, Q. Sun, W. J. Buchanan, and J. Cao, "Iot forensics: Amazon echo as a use case," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 6487–6497, 2019.

[85] M. Harbawi and A. Varol, "An improved digital evidence acquisition model for the internet of things forensic i: A theoretical framework," in *2017 5th International Symposium on Digital Forensic and Security (ISDFS)*. IEEE, 2017, pp. 1–6.

[86] H. F. Atlam, A. Alenezi, M. O. Alassafi, and G. Wills, "Blockchain with internet of things: Benefits, challenges, and future directions," *International Journal of Intelligent Systems and Applications*, vol. 10, no. 6, pp. 40–48, 2018.

[87] D.-P. Le, H. Meng, L. Su, S. L. Yeo, and V. Thing, "Biff: A blockchain-based iot forensics framework with identity privacy," in *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE, 2018, pp. 2372–2377.

[88] M. Hossain, Y. Karim, and R. Hasan, "Fif-iot: A forensic investigation framework for iot using a public digital ledger," in *2018 IEEE International Congress on Internet of Things (ICIOT)*. IEEE, 2018, pp. 33–40.

[89] S. Brotsis, N. Kolokotronis, K. Limniotis, S. Shiaeles, D. Kavallieros, E. Bellini, and C. Pavué, "Blockchain solutions for forensic evidence preservation in iot environments," in *2019 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2019, pp. 110–114.

[90] J. H. Ryu, P. K. Sharma, J. H. Jo, and J. H. Park, "A blockchain-based decentralized efficient investigation framework for iot digital forensics," *The Journal of Supercomputing*, vol. 75, no. 8, pp. 4372–4387, 2019.

[91] J. Hou, Y. Li, J. Yu, and W. Shi, "A survey on digital forensics in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 1–15, 2019.

[92] M. James and P. Szewczyk, "Jurisdictional issues in cloud forensics," 2017.

[93] R. Montasari, R. Hill, S. Parkinson, P. Peltola, A. Hosseinian-Far, and A. Daneshkhah, "Digital forensics: Challenges and opportunities for future studies," *International Journal of Organizational and Collective Intelligence*, vol. 10, no. 2, 2019.

[94] S. Zawoad and R. Hasan, "Cloud forensics: a meta-study of challenges, approaches, and open problems," *arXiv preprint arXiv:1302.6312*, 2013.

[95] H. S. Lallie and L. Pimlott, "Applying the acpo principles in public cloud forensic investigations," *Journal of Digital Forensics, Security and Law*, vol. 7, no. 1, p. 5, 2012.

[96] R. McKemmish, *What is forensic computing?* Australian Institute of Criminology Canberra, 1999.

[97] M. D. Kohn, M. M. Eloff, and J. H. Eloff, "Integrated digital forensic process model," *Computers & Security*, vol. 38, pp. 103–115, 2013.

[98] D. Quick and K.-K. R. Choo, "Digital droplets: Microsoft skydrive forensic data remnants," *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1378–1394, 2013.

[99] B. Martini and K.-K. R. Choo, "An integrated conceptual digital forensic framework for cloud computing," *Digital Investigation*, vol. 9, no. 2, pp. 71–80, 2012.

[100] A. N. Moussa, N. Ithnin, N. Almolhis, and A. Zainal, "A consumer-oriented cloud forensic process model," in *2019 IEEE 10th Control and System Graduate Research Colloquium (ICSGRC)*. IEEE, 2019, pp. 219–224.

[101] R. Montasari, R. Hill, V. Carpenter, and A. Hosseinian-Far, "The standardised digital forensic investigation process model (sdfipm)," in *Blockchain and Clinical Trial*. Springer, 2019, pp. 169–209.

[102] M. A. Saleh, S. H. Othman, A. Al-Dhaqm, and M. A. Al-Khasawneh, "Common investigation process model for internet of things forensics," in *2021 2nd International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*. IEEE, 2021, pp. 84–89.

[103] B. Martini and R. Choo, "Distributed filesystem forensics: Xtreemfs as a case study," *Digital Investigation*, vol. 11, no. 4, pp. 295–313, 2014.

[104] G. Grispos, T. Storer, and W. B. Glisson, "Calm before the storm: The challenges of cloud computing in digital forensics," *International Journal of Digital Crime and Forensics (IJDCF)*, vol. 4, no. 2, pp. 28–48, 2012.

[105] M. Taylor, J. Haggerty, D. Gresty, and D. Lamb, "Forensic investigation of cloud computing systems," *Network Security*, vol. 2011, no. 3, pp. 4–10, 2011.

[106] I. Techniques, "Iso/iec 27037: 2012 information technology security techniques guidelines for identification collection acquisition and preservation of digital evidence," *ISO/IEC-The standard was published in October*, 2012.

[107] D. Brezinski and T. Killalea, "Guidelines for evidence collection and archiving," *RFC3227, February*, 2002.

[108] M. Alhamad, T. Dillon, and E. Chang, "Conceptual sla framework for cloud computing," in *4th IEEE International Conference on Digital Ecosystems and Technologies*. IEEE, 2010, pp. 606–610.

[109] K. Ruan, J. James, J. Carthy, and T. Kechadi, "Key terms for service level agreements to support cloud forensics," in *IFIP International Conference on Digital Forensics*. Springer, 2012, pp. 201–212.

[110] T. Sang, "A log based approach to make digital forensics easier on cloud computing," in *2013 Third International Conference on Intelligent System Design and Engineering Applications*. IEEE, 2013, pp. 91–94.

[111] D. Birk and M. Panico, "Mapping the forensic standard iso/iec 27037 to cloud computing," *Cloud Security Alliance*, pp. 1–31, 2013.

[112] B. R. Kandukuri, A. Rakshit *et al.*, "Cloud security issues," in *2009 IEEE International Conference on Services Computing*. IEEE, 2009, pp. 517–520.

[113] D. Quick, B. Martini, and R. Choo, *Cloud storage forensics*. Syngress, 2013.

[114] "AWS", "Security at scale: Logging in aws," *https://d1.awsstatic.com/whitepapers/compliance/AWS_Security_at_Scale_Logging_in_AWS_Whitepaper.pdf?did=wp_card&trk=wp_card*, 2015.

[115] A. Haeberlen, "A case for the accountable cloud," *ACM SIGOPS Operating Systems Review*, vol. 44, no. 2, pp. 52–57, 2010.

[116] K. Ruan, J. Carthy, T. Kechadi, and I. Baggili, "Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results," *Digital Investigation*, vol. 10, no. 1, pp. 34–43, 2013.

[117] S. Biggs and S. Vidalis, "Cloud computing: The impact on digital forensic investigations," in *2009 International Conference for Internet Technology and Secured Transactions,(ICITST)*. IEEE, 2009, pp. 1–6.

[118] R. Mogull, J. Arlen, F. Gilbert, A. Lane, D. Mortman, G. Peterson, and M. Rothman, "Security guidance for critical areas of focus in cloud computing, v4. 0," *Cloud Security Alliance*, 2021.

[119] C.-H. Lin, C.-Y. Lee, and T.-W. Wu, "A cloud-aided rsa signature scheme for sealing and storing the digital evidences in computer forensics," *International journal of security and its Applications*, vol. 6, no. 2, pp. 241–244, 2012.

[120] W. Delport, M. Köhn, and M. S. Olivier, "Isolating a cloud instance for a digital forensic investigation." in *ISSA*, 2011.

[121] C. Greamo and A. Ghosh, "Sandboxing and virtualization: Modern tools for combating malware," *IEEE Security & Privacy*, vol. 9, no. 2, pp. 79–82, 2011.

[122] R. Adams, "The emergence of cloud storage and the need for a new digital forensic process model," in *Cybercrime and cloud forensics: Applications for investigation processes*. IGI Global, 2013, pp. 79–104.

[123] T. G. Shipley and C. CFE, "Collecting legally defensible online evidence," 2007.

[124] Azure, "Azure Logging and Auditing," *[online] https://docs.microsoft.com/en-us/azure/security/azure-log-audit[Accessed: 08.08.2018]*, 2019a.

[125] M. Taylor, J. Haggerty, D. Gresty, and R. Hegarty, "Digital evidence in cloud computing systems," *Computer law & security review*, vol. 26, no. 3, pp. 304–308, 2010.

[126] J. Dykstra and A. T. Sherman, "Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques," *Digital Investigation*, vol. 9, pp. S90–S98, 2012.

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

[128] D. Liu, J. Lee, J. Jang, S. Nepal, and J. Zic, "A cloud architecture of virtual trusted platform modules," in *2010 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*. IEEE, 2010, pp. 804–811.

[129] F. J. Krautheim, D. S. Phatak, and A. T. Sherman, "Introducing the trusted virtual environment module: a new mechanism for rooting trust in cloud computing," in *International Conference on Trust and Trustworthy Computing*. Springer, 2010, pp. 211–227.

[130] N. Santos, K. P. Gummadi, and R. Rodrigues, "Towards trusted cloud computing." *HotCloud*, vol. 9, no. 9, p. 3, 2009.

[131] A. F. Alsadhan and M. A. Alhussein, "Deleted data attribution in cloud computing platforms," in *2018 1st International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 2018, pp. 1–6.

[132] C. Federici, "Cloud data imager: A unified answer to remote acquisition of cloud storage areas," *Digital Investigation*, vol. 11, no. 1, pp. 30–42, 2014.

[133] Zafarullah, F. Anwar, and Z. Anwar, "Digital forensics for eucalyptus," in *2011 Frontiers of Information Technology*. IEEE, 2011, pp. 110–116.

[134] J. Dykstra and A. T. Sherman, "Understanding issues in cloud forensics: two hypothetical case studies," *UMBC Computer Science and Electrical Engineering Department*, 2011.

[135] K. M. Khan and Q. Malluhi, "Establishing trust in cloud computing," *IT professional*, vol. 12, no. 5, pp. 20–27, 2010.

[136] F. Daryabar, A. Dehghantanha, N. I. Udzir, S. bin Shamsuddin, F. Norouzizadeh *et al.*, "A survey about impacts of cloud computing on digital forensics," *International Journal of Cyber-Security and Digital Forensics*, vol. 2, no. 2, pp. 77–95, 2013.

[137] S. Alsouri, T. Feller, S. Malipatlolla, and S. Katzenbeisser, "Hardware-based security for virtual trusted platform modules," *arXiv preprint arXiv:1308.1539*, 2013.

[138] E. G. Hogben and M. Dekker, "A guide to monitoring of security service levels in cloud contracts," *ENISA, Tech Rep TR-2012-04-02*, 2012.

[139] Google, "Google's approach to it security, a google white paper," *http://www.google.com/enterprise/apps/business/resources/docs/security-whitepaper.html*, 2014.

[140] B. Martini and K.-K. R. Choo, "Cloud storage forensics: owncloud as a case study," *Digital Investigation*, vol. 10, no. 4, pp. 287–299, 2013.

[141] AWS, "Aws cloudtrail user guide version 1.0," *https://docs.aws.amazon.com/awscloudtrail/latest/userguide/awscloudtrail-ug.pdf#cloudtrail-user-guide*, 2020.

[142] S. Bhatt, P. K. Manadhata, and L. Zomlot, "The operational role of security information and event management systems," *IEEE security & Privacy*, vol. 12, no. 5, pp. 35–41, 2014.

[143] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, pp. 680–698, 2018.

[144] S. Lins, S. Schneider, and A. Sunyaev, "Trust is good, control is better: Creating secure clouds by continuous auditing," *IEEE Transactions on Cloud Computing*, vol. 6, no. 3, pp. 890–903, 2016.

[145] S. Lins, S. Schneider, J. Szefer, S. Ibraheem, and A. Sunyaev, "Designing monitoring systems for continuous certification of cloud services: deriving meta-requirements and design guidelines," *Communications of the Association for Information Systems*, vol. 44, no. 1, p. 25, 2019.

[146] A. Pichan, M. Lazarescu, and S. T. Soh, "Towards a practical cloud forensics logging framework," *Journal of information security and applications*, vol. 42, pp. 18–28, 2018.

[147] A. Alenezi, N. H. N. Zulkipli, H. F. Atlam, R. J. Walters, and G. B. Wills, "The impact of cloud forensic readiness on security." in *CLOSER*, 2017, pp. 511–517.

[148] A. Alenezi, H. F. Atlam, and G. B. Wills, "Experts reviews of a cloud forensic readiness framework for organizations," *Journal of Cloud Computing*, vol. 8, no. 1, pp. 1–14, 2019.

[149] A. N. Moussa, N. Ithnin, and A. Zainal, "Cfaas: bilaterally agreed evidence collection," *Journal of Cloud Computing*, vol. 7, no. 1, pp. 1–19, 2018.

[150] R. Kumar and R. Goyal, "Assurance of data security and privacy in the cloud: A three-dimensional perspective," *Software Quality Professional*, vol. 21, no. 2, pp. 7–26, 2019.

[151] R. Montasari and R. Hill, "Next-generation digital forensics: Challenges and future paradigms," in *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*. IEEE, 2019, pp. 205–212.

[152] Y. Khan and S. Varma, "Development and design strategies of evidence collection framework in cloud environment," *Social Networking and Com-*

putational Intelligence; Springer: Berlin/Heidelberg, Germany, pp. 27–37, 2020.

[153] O. Akter, A. Akther, M. A. Uddin, and M. M. Islam, "Cloud forensics: Challenges and blockchain based solutions," *International Journal of Modern Education and Computer Science*, vol. 10, no. 8, pp. 1–12, 2020.

[154] V. R. Silvarajoo, S. Y. Lim, and P. Daud, "Digital evidence case management tool for collaborative digital forensics investigation," in *2021 3rd International Cyber Resilience Conference (CRC)*. IEEE, 2021, pp. 1–4.

[155] A. S. Zalazar, L. Ballejos, and S. Rodriguez, "Cloud dimensions for requirements specification," in *Requirements Engineering for Service and Cloud Computing*. Springer, 2017, pp. 23–43.

[156] R. Battistoni, R. Di Pietro, and F. Lombardi, "Cure—towards enforcing a reliable timeline for cloud forensics: Model, architecture, and experiments," *Computer Communications*, vol. 91, pp. 29–43, 2016.

[157] A. Akilal and M.-T. Kechadi, "An improved forensic-by-design framework for cloud computing with systems engineering standard compliance," *Forensic Science International: Digital Investigation*, vol. 40, p. 301315, 2022.

[158] W. Yassin, M. F. Abdollah, R. Ahmad, Z. Yunos, and A. Ariffin, "Cloud forensic challenges and recommendations: A review," *OIC-CERT Journal of Cyber Security*, vol. 2, no. 1, pp. 19–29, 2020.

[159] R. Montasari, R. Hill, S. Parkinson, P. Peltola, A. Hosseinian-Far, and A. Daneshkhah, "Digital forensics: challenges and opportunities for future studies," *International Journal of Organizational and Collective Intelligence (IJOCI)*, vol. 10, no. 2, pp. 37–53, 2020.

[160] B. Yankson and A. Davis, "Analysis of the current state of cloud forensics: The evolving nature of digital forensics," in *2019 IEEE/ACS 16th Inter-*

national Conference on Computer Systems and Applications (AICCSA). IEEE, 2019, pp. 1–8.

[161] F. Casino, T. K. Dasaklis, G. Spathoulas, M. Anagnostopoulos, A. Ghosal, I. Borocz, A. Solanas, M. Conti, and C. Patsakis, "Research trends, challenges, and emerging topics in digital forensics: A review of reviews," *IEEE Access*, 2022.

[162] M. J. Islam, M. Mahin, A. Khatun, S. Roy, S. Kabir, and B. C. Debnath, "A comprehensive data security and forensic investigation framework for cloud-iot ecosystem," *GUB Journal of Science and Engineering*, vol. 4, 2019.

[163] T. Wu, F. Breitinger, and I. Baggili, "Iot ignorance is digital forensics research bliss: a survey to understand iot forensics definitions, challenges and future research directions," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019, pp. 1–15.

[164] H. Jahankhani and J. Ibarra, "Digital forensic investigation for the internet of medical things (iomt)," *Forensic Leg. Investig. Sci.*, vol. 5, no. 2, p. 029, 2019.

[165] C. Perry, "Challenges of traditional forensic methods in a cloud environment," Ph.D. dissertation, Utica College, 2019.

[166] F. N. A. Sackey, "Strategies to manage cloud computing operational costs," Ph.D. dissertation, Walden University, 2018.

[167] J. Jacob, "Comparing security concerns in the adoption of public cloud computing based on region of residence," Ph.D. dissertation, Northcentral University, 2016.

[168] M. I. Alghamdi, "Digital forensics in cyber security—recent trends, threats, and opportunities," in *Cybersecurity Threats with New Perspectives*. IntechOpen, 2021.

[169] B. Raju, B. Moharil, and G. Geethakumari, "FaaSeC: enabling forensics-as-a-service for cloud computing systems," in *Proceedings of the 9th International Conference on Utility and Cloud Computing*. ACM, 2016, pp. 220–227.

[170] J. E. Holt, "Logcrypt: forward security and public verification for secure audit logs," in *ACM international conference proceeding series*, vol. 167, 2006, pp. 203–211.

[171] Google Cloud Platform, "Stackdriver Logging," *https://cloud.google.com/logging/, [online] https://cloud.google.com/storage/docs/[Accessed: 07.11.2018]*, 2019.

[172] C. Yuen and S. Vidich, "Microsoft Azure Compliance Offerings," *[online] https://aka.ms/AzureCompliance [Accessed: 26.10.2021]*, 2021.

[173] A. Pichan, M. Lazarescu, and S. T. Soh, "A case study on major cloud platforms digital forensics readiness - are we there yet?" *International Journal of Cloud Computing, https://www.inderscience.com/info/ingeneral/forthcoming.php?jcode=ijcc*, "in press".

[174] ISO, "ISO/IEC 27037:2012 information technology - security techniques - guidelines for identification, collection, acquisition and preservation of digital evidence," *ISO Publication*, vol. ICS 35.040, 2016.

[175] S. Mason and D. Seng, *Electronic evidence*. University of London Press, 2017.

[176] N. H. Ab Rahman, N. D. W. Cahyani, and K.-K. R. Choo, "Cloud incident handling and forensic-by-design: cloud storage as a case study," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 14, p. e3868, 2017.

[177] G. Panfilo and F. Arias, "The coordinated universal time (utc)," *Metrologia*, vol. 56, no. 4, p. 042001, 2019.

[178] R. B. Bohn, J. Messina, F. Liu, J. Tong, and J. Mao, "Nist cloud computing reference architecture," in *Services (SERVICES), 2011 IEEE World Congress on*. IEEE, 2011, pp. 594–596.

[179] H. Studiawan, F. Sohel, and C. Payne, "A survey on forensic investigation of operating system logs," *Digital Investigation*, vol. 29 (Jun), pp. 1–20, 2019.

[180] J. Farina, M. Scanlon, N.-A. Le-Khac, and M.-T. Kechadi, "Overview of the forensic investigation of cloud services," in *2015 10th International Conference on Availability, Reliability and Security*. IEEE, 2015, pp. 556–565.

[181] M. M. Ahsan, A. W. A. Wahab, M. Y. I. Idris, S. Khan, E. Bachura, and K.-K. R. Choo, "Class: cloud log assuring soundness and secrecy scheme for cloud forensics," *IEEE Transactions on Sustainable Computing*, 2018.

[182] C. Grajeda, F. Breitinger, and I. Baggili, "Availability of datasets for digital forensics–and what is missing," *Digital Investigation*, vol. 22, no. Supplement, pp. S94–S105, 2017.

[183] B. Boeck, D. Huemer, and A. M. Tjoa, "Towards more trustable log files for digital forensics by means of "trusted computing"," in *2010 24th IEEE International Conference on Advanced Information Networking and Applications*. IEEE, 2010, pp. 1020–1027.

[184] S. Soltani and S. A. H. Seno, "A formal model for event reconstruction in digital forensic investigation," *Digital Investigation*, vol. 30 (Sep), pp. 148–160, 2019.

[185] M. Neovius, J. Karlsson, M. Westerlund, and G. Pulkkis, "Providing

tamper-resistant audit trails for cloud forensics with distributed ledger based solutions," *CLOUD COMPUTING 2018*, pp. 19–24, 2018.

[186] NIST, "NIST CFReDS Current Data Sets," *[online] https://www.nist.gov/node/1303796/cfreds-current-data-sets [Accessed: 20.05.2018]*, 2017.

[187] AWS, "AWS CloudTrail," *[online] https://aws.amazon.com/cloudtrail/ [Accessed: 15.04.2018]*, 2018.

[188] AWS, "AWS Documentation," *[online] https://docs.aws.amazon.com [Accessed: 10.05.2018]*, 2019a.

[189] AWS, "AWS Elastic Beanstalk Documentation," *[online] https://docs.aws.amazon.com/elasticbeanstalk/latest/api/API_ Operations.html [Accessed: 16.06.2018]*, 2019b.

[190] Azure, "Azure Storage Analytics Log Format," *[online] https://docs.microsoft.com/en-us/rest/api/storageservices/storage-analytics [Accessed: 04.09.2018]*, 2019b.

[191] Azure, "Diagnostic logging in Azure Cosmos DB," *[online] https://docs.microsoft.com/en-us/azure/cosmos-db/logging, https://docs.microsoft.com/en-us/azure/cosmos-db/logging [Accessed: 12.09.2018]*, 2019c.

[192] Azure, "Search the audit log in the Office 365 Security & Compliance Center," *[online] https://docs.microsoft.com/en-us/office365/securitycompliance/search-the-audit-log-in-security-and-compliance#step-2-view-the-search-results, https://docs.microsoft.com/en-us/office365/securitycompliance/enable-mailbox-auditing [Accessed: 04.10.2018]*, 2019d.

[193] Google Cloud Platform, "Google Cloud Audit Logging," *[online] https://cloud.google.com/logging/docs/audit/ [Accessed: 18.11.2018]*, 2019.

[194] ——, "Kubernetes Engine," *[online] https://cloud.google.com/kubernetes-engine/[Accessed: 10.01.2019]*, 2019.

[195] H. F. Atlam, A. Alenezi, A. Alharthi, R. J. Walters, and G. B. Wills, "Integration of cloud computing with internet of things: challenges and open issues," in *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 2017, pp. 670–675.

[196] A. Pichan, M. Lazarescu, and S. T. Soh, "A logging model for enabling digital forensics in iot, in an inter-connected iot, cloud eco-systems," in *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*. IEEE, 2020, pp. 478–483.

[197] F. A. Alaba, M. Othman, I. A. T. Hashem, and F. Alotaibi, "Internet of things security: A survey," *Journal of Network and Computer Applications*, vol. 88, pp. 10–28, 2017.

[198] N. Akatyev and J. I. James, "Evidence identification in iot networks based on threat assessment," *Future Generation Computer Systems*, vol. 93, pp. 814–821, 2019.

[199] M. Chernyshev, S. Zeadally, Z. Baig, and A. Woodward, "Internet of things forensics: the need, process models, and open issues," *IT professional*, vol. 20, no. 3, pp. 40–49, 2018.

[200] H. A. Abdul-Ghani, D. Konstantas, and M. Mahyoub, "A comprehensive iot attacks survey based on a building-blocked reference model," *IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, pp. 355–373, 2018.

[201] J. Deogirikar and A. Vidhate, "Security attacks in iot: A survey," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2017, pp. 32–37.

[202] A. MacDermott, T. Baker, and Q. Shi, "Iot forensics: Challenges for the ioa era," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*. IEEE, 2018, pp. 1–5.

[203] F. Servida and E. Casey, "Iot forensic challenges and opportunities for digital traces," *Digital Investigation*, vol. 28, pp. S22–S29, 2019.

[204] M. Ibrahim, M. B. Jasser, M. T. Abdullah, and A. Abdullah, "Formalization in digital forensic triage for identification of malicious iot devices," *International Journal of Engineering and Advanced Technology (IJEAT)*.

[205] I. Yaqoob, I. A. T. Hashem, A. Ahmed, S. A. Kazmi, and C. S. Hong, "Internet of things forensics: Recent advances, taxonomy, requirements, and open challenges," *Future Generation Computer Systems*, vol. 92, pp. 265–275, 2019.

[206] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future generation computer systems*, vol. 56, pp. 684–700, 2016.

[207] Z.-K. Zhang, M. C. Y. Cho, C.-W. Wang, C.-W. Hsu, C.-K. Chen, and S. Shieh, "Iot security: ongoing challenges and research opportunities," in *2014 IEEE 7th international conference on service-oriented computing and applications*. IEEE, 2014, pp. 230–234.

[208] H. Kim and E. A. Lee, "Authentication and authorization for the internet of things," *IT Professional*, vol. 19, no. 5, pp. 27–33, 2017.

[209] R. C. Motta, K. M. de Oliveira, and G. H. Travassos, "A framework to support the engineering of internet of things software systems," in *Proceedings*

of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems, 2019, pp. 1–6.

[210] P. Lutta, M. Sedky, M. Hassan, U. Jayawickrama, and B. B. Bastaki, "The complexity of internet of things forensics: A state-of-the-art review," *Forensic Science International: Digital Investigation*, vol. 38, p. 301210, 2021.

[211] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[212] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp., Security Response*, vol. 5, no. 6, p. 29, 2011.

[213] T. Chen and S. Abu-Nimeh, "Lessons from stuxnet," *Computer*, vol. 44, no. 4, pp. 91–93, 2011.