

Chapter 13

ISOLATING INSTANCES IN CLOUD FORENSICS

Waldo Delport and Martin Olivier

Abstract The isolation of a computing environment is an integral part of a digital forensic process. Isolation helps prevent evidence contamination and possible tampering. This paper focuses on the process of isolating instances in cloud computing systems. Several conditions are specified to serve as a guide for the successful isolation of cloud instances. Also, the complications that can arise during a cloud forensic investigation are discussed.

Keywords: Cloud forensics, instances, isolation

1. Introduction

Cloud computing enables service providers to provide virtual resources to users over large networks in a flexible and scalable manner [8]. However, the massive distribution and virtualization of resources in multi-user and multi-jurisdictional environments create significant challenges for digital forensic investigations involving cloud systems. Key challenges are to locate the suspect instances, isolate the “crime scene,” prevent evidence contamination and extract the evidence in a forensically sound manner.

In a cloud investigation, an instance must be isolated when it becomes apparent that it was involved in the incident under investigation or may contain evidence pertaining to the incident. Isolation helps preserve the integrity of the evidence collected from the instance. However, one of the problems with preserving integrity is that one instance may share storage with multiple instances and the data may not be in a fixed location in the cloud [2]. Another complexity is that other instances on the same cloud node may belong to other users; a forensic investigation should not impact the availability and privacy of these users’ resources [6].

This paper focuses on the process of isolating instances in cloud systems. Several conditions are specified for the successful isolation of cloud instances. Also, the complications that can arise during a cloud forensic investigation are discussed.

2. Cloud Computing

Cloud computing is a relatively old term that has been adopted widely over the last few years [21]. For the purposes of this paper, we define cloud computing as a distributed computing architecture that provides flexible, cost-effective, on-demand resources to users over a network using a virtualization technology to create virtual resources.

There are three types of service models in cloud computing [3]:

- **Infrastructure as a Service (IaaS):** This service model provides a cloud user with a virtual computer that is typically accessed over the Internet [2]. This virtual computer, which is referred to as an instance, is configured and maintained by the user. Normally, an instance can be accessed from anywhere in the world, depending on the security configuration. The instance can be a small instance that is used by a single user to store file backups, or it could be a large instance that operates as a server for a corporate website and database. The user only pays the provider for services rendered. If the user requirements change in terms of computational power or storage space, it is an easy process to reconfigure the instance to accommodate the new requirements. Likewise, the task of starting up a new instance is a trivial matter. In the IaaS model, the service provider typically maintains the confidentiality, integrity and availability of cloud instances at the hardware level. The user is responsible for maintaining security properties at a higher level (e.g., operating system and files) [11].
- **Platform as a Service (PaaS):** This service model provides a cloud user with a platform that is maintained by the cloud service provider [3]. The platform is an instance created with a specific focus by the service provider. Alternatively, the service provider may create a default platform for a web server, which the user can subsequently configure as desired. In the PaaS model, the service provider may also provide tools to build upon the platform.
- **Software as a Service (SaaS):** This service model makes software available through the cloud. Applications and their data are viewed as cloud resources [15]. The user pays the service provider for access to an application that can be customized as desired. In

the SaaS model, the user has no concerns related to the underlying hardware and software.

There are four deployment models in cloud computing [13]:

- **Public Cloud:** A public cloud is owned by a service provider that sells the cloud resources to other companies and the public. The service provider is responsible for managing the cloud.
- **Private Cloud:** A private cloud is owned and used by a single company. The cloud infrastructure can be located on company premises or elsewhere. The owner company or a contracted company is responsible for maintaining the cloud.
- **Community Cloud:** A community cloud is owned and used by multiple companies, which form a community with a shared interest. The cloud infrastructure can be located on the premises of one of the companies or it may be located elsewhere. One or more of the owner companies or a contracted company are responsible for maintaining the cloud.
- **Hybrid Cloud:** A hybrid cloud is a combination of at least two of the other three models. However, each individual cloud model persists as a separate entity in the hybrid cloud. Various technologies are used to bond the entities together and perform load balancing.

Cloud computing provides significant value to small and medium enterprises [18]. These enterprises often operate under a survivalist mentality [9], primarily because they have limited capital and a highly focused knowledge base. Cloud computing enables these enterprises to access resources without significant capital outlays and hardware set-up and maintenance costs. Moreover, the cloud infrastructures can be provisioned as necessary as the enterprises grow.

Cloud computing has become a billion dollar industry [19]. Many of the largest IT companies, notably Google, Microsoft, IBM and Amazon [2, 11], have made massive investments in cloud computing. These companies are employing various techniques to ensure that users can have provisioned cloud resources with the desired levels of confidentiality, integrity and availability.

3. Digital Forensic Process

A well-defined forensic process must be followed to obtain admissible evidence. Cohen [7] proposes a model for digital forensic examinations that consists of seven phases: (i) identification; (ii) collection; (iii)

transportation; (iv) storage; (v) examination and traces; (vi) presentation; and (vii) destruction. The examination and traces phase has four sub-phases: (i) analysis; (ii) interpretation; (iii) attribution; and (iv) reconstruction [7].

Cohen [7] emphasizes that documentation is a continuous process that needs to occur during all the phases. Good documentation can help preserve the integrity of evidence. The documentation, at the very least, should include the name of the item of evidence and the location where it was obtained. The documentation should also cover the processes involved in identifying, retrieving, storing and transporting the evidence (including chain of custody processes).

Several other digital forensic processes have been specified. Most of them incorporate the phases identified by Cohen [7]. One of the most prominent is the digital forensic process specified by the National Institute of Justice [20]. The process comprises four phases: (i) collection; (ii) examination; (iii) analysis; and (iv) reporting. While the two processes incorporate the same set of underlying phases, the process advocated by Cohen, which is subdivided into more phases, enables a more systematic flow of events.

The normal computer forensic process uses static or “dead” analysis [4]. In dead analysis, the computer is turned off as soon as possible, the storage media are imaged and the images are analyzed. The other approach is “live” analysis, where the computer is kept on and evidence is gathered from the running computer. The main disadvantage of dead analysis is that information in buffers and RAM can be lost. The problem with live analysis is that the evidence can be unintentionally destroyed or modified [4].

Cloud computing adds considerable complexity to a digital forensic investigation. As stated above, a major problem is that a cloud instance that is the focus of an investigation may contain data belonging to multiple users [2]. Thus, isolation of data is an important issue in cloud forensics.

4. Isolation

The term “isolate” is defined as “the state of being identified and then separated from others” [16]. Isolation is an important requirement in a forensic process because it helps protect possible evidence from contamination and loss of continuity [22]. A traditional crime scene is often cordoned off and divided into separate parts to provide isolation. These parts can only be accessed by authorized persons in an authorized

manner. A log is maintained of the locations and activities of all the persons who were present at the crime scene.

Isolation is also employed in the digital forensics realm to protect possible evidence from contamination and tampering. A seized cell phone is placed inside a Faraday bag to prevent the phone from communicating with the outside world [10]. In hard drive forensics, a write blocker is employed to ensure write-free reads [12].

After considerable research, we identified several conditions that must be met in order for a cloud instance to be successfully isolated:

- **Location:** The physical location of the instance is known.
- **Incoming Blocking:** The instance is blocked from receiving communications from the outside world.
- **Outgoing Blocking:** The instance is blocked from sending communications to the outside world.
- **Collection:** Evidence from the instance can be gathered.
- **Non-Contamination:** Evidence from the instance is not contaminated by the isolation process.
- **Separation:** Information unrelated to the incident is not part of the isolation process.

In order to know the physical location of an instance, its location in the cloud must be known. This requires locating the node on which the instance resides.

The instance must be protected from other instances and other external sources. Since all interactions with a cloud instance are via the network, network connections must be disabled to protect against evidence contamination and tampering. The instance is also blocked from sending messages over the network.

It must be possible to collect all possible evidence from the instance after it has been isolated on the node. The possible evidence includes running programs, data in the swap space and data on the hard drive.

The isolation process is designed to protect the evidence. The isolation process has failed if the evidence on the instance is contaminated. Note that there is no reason to perform the isolation process if it does not protect the evidence.

Data from multiple instances may reside on a node. To ensure confidentiality, the isolation must be performed so that all unrelated information is excluded by the isolation process while all related information is isolated.

5. Isolation in a Cloud

The cloud deployment and service models have an impact on isolation. This section focuses on isolation within each deployment and service model. The emphasis is on private and public deployment models; community and hybrid community clouds can be shown to fit into either public or private clouds.

Cloud deployment models differ in where the data is located and who owns the data. The location and ownership of the data are important considerations in performing the isolation process.

In the case of a private cloud, all the data in the cloud belongs to the company that owns and uses the cloud. Concerns about confidentiality are low if this company itself performs a digital forensic investigation. On the other hand, if an external company performs the investigation, then only the data related to the incident must be accessed; the other data should remain private. Therefore, when the owner company itself conducts an investigation of its private cloud, only the first five isolation conditions must be satisfied; the sixth condition, separation, is not important.

However, all six conditions, including separation, should be satisfied when an external company performs an investigation of a private cloud.

In a public cloud, the data belongs to different users who may be located anywhere in the world. This introduces a jurisdictional problem, because the data in the cloud may be under different legal systems. The service provider is responsible for ensuring confidentiality in a public cloud, but a forensic investigation would be conducted by the provider or by an external entity. In either case, separation is required to protect data belonging to other users. This means that all six isolation conditions must be satisfied.

The three cloud service models separate the data into different layers within a cloud. Each layer holds separate conditions for isolation.

In an IaaS cloud, each instance (virtual machine) in the cloud could belong to a different user. Since each instance is potentially unique, the entire instance should be isolated. Another reason is that each instance is an unknown computer that could destroy evidence residing in the instance itself or contaminate evidence residing in other instances.

In a PaaS cloud, the platform that underlies each instance is known. The instances differ in their installed software and stored data. In order to isolate the instance, all applications and data running on the instance should be isolated. Because the underlying platform is known, the expected behavior of the instance is known and can be controlled.

In the case of a SaaS cloud, instances only differ in their application configurations and stored data, which may constitute evidence. Therefore, only these portions of an instance need to be isolated. The task is simplified by the fact that the expected behavior of the instance is known and can be controlled.

6. Distributed Instance System

The previous section discussed the isolation of single instances; this section focuses on isolation in a cloud environment containing multiple suspect instances.

Availability is a necessity in a cloud computing environment [13]. A distributed instance system is a popular cloud environment that is designed to provide availability. In such a system, multiple cloud instances are combined to create a single logical resource. The combination of resources enhances availability. When one of the instances malfunctions, it is discarded, and a new instance is launched to take its place. This means that the instances in a distributed instance system are dispensable.

A unique implementation is a multi-node server farm constructed using an IaaS cloud [5]. In a web server farm, a website is split over two or more nodes. Users interacting with the website only see a single server. The server farm routes requests from users to the various nodes. Distribution technologies are used to enable this service. The distribution enhances the quality of service of the website. Also, there is no single point of failure; when a node fails, the router simply stops sending requests to the node.

In an IaaS cloud implementation, a distributed instance system comprises several nodes. This helps provide continuous availability. The failure of a node has little to no effect on the overall system.

7. Locating an Instance

In order to locate an instance, the node on which the instance is running needs to be identified. One method is to use the cloud management software, which may provide the functionality to locate instances. Another method is subnetting, where a subnetwork is created for each node. The subnetwork is then used to trace an instance.

We conducted an experiment that used subnetting to locate instances. The Nimbula Director [14], a cloud operating system that provides IaaS in a private network, was used in the experiment. An instance was launched on the cloud, and the goal was to locate the node on which the instance resides.

```
nimbulaadmin@0-27-e-c-fc-c4:~
File Edit View Search Terminal Help

tapa515130 Link encap:Ethernet HWaddr A2:6A:EF:40:6A:7A
inet addr:10.128.0.9 Bcast:0.0.0.0 Mask:255.255.255.255
inet6 addr: fe80::a06a:eff:fe40:6a7a/64 Scope:Link
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:360 errors:0 dropped:0 overruns:0 frame:0
TX packets:173 errors:0 dropped:11 overruns:0 carrier:0
collisions:0 txqueuelen:500
RX bytes:37473 (36.5 KiB) TX bytes:19726 (19.2 KiB)

[nimbulaadmin@0-27-e-c-fc-c4 ~]$
```

Figure 1. Output of the `ifconfig` command on a node.

Three pieces of the information were available: (i) IP address of the instance; (ii) subnetwork architecture; and (iii) wire address. The IP address of 10.128.0.10 was provided by Nimbula when the instance was started. In the experiment, a subnetwork (subnet) was created by Nimbula for each instance group in the cloud. The wire address corresponded to the network connection address, the lowest IP address in the network [1].

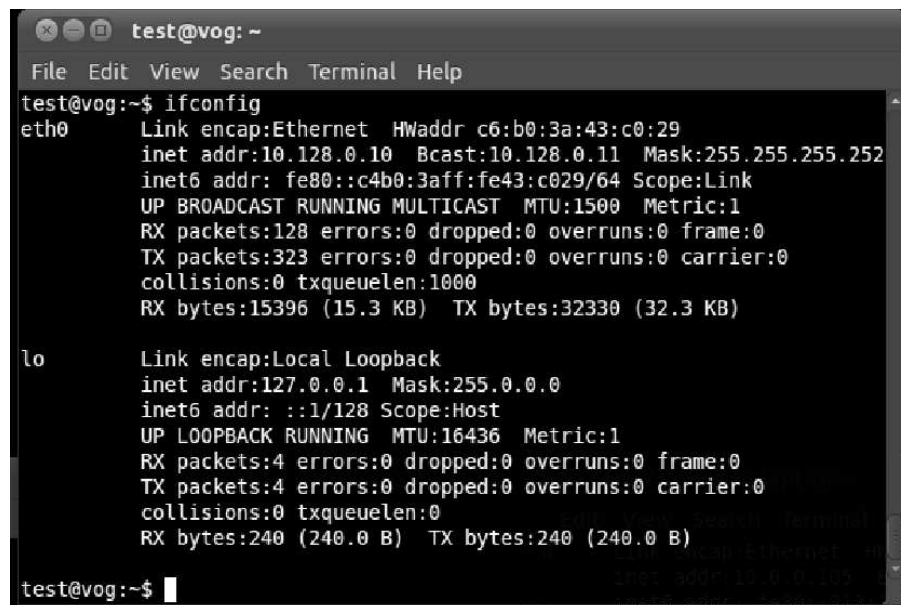
To locate instances using this information, it is necessary to analyze network information corresponding to the nodes and instances. First, the `ifconfig` command was run on each node.

Figure 1 contains a portion of the `ifconfig` output corresponding to a node in the cloud. The figure shows the IP address of one of the network interfaces on the node.

Figure 2 contains the `ifconfig` output corresponding to an instance running in the cloud. The figure shows the network interface IP address and other network attributes.

In Figure 2, the subnet mask for the instance is 255.255.255.252, i.e., 11111111.11111111.11111111.11111100 [1]. The mask uses the first 30 bits as the network ID. The remaining two bits can be used as the address. This means that three IP addresses are available in the private network of the instance.

According to Figure 2, the instance IP address is 10.128.0.10, and the broadcast address of the network of the instance is 10.128.0.11. Therefore, it can be inferred that, because three addresses are available and two of the addresses have been used, the wire address corresponds to 10.128.0.9. In particular, the wire address, which corresponds to



```

test@vog: ~
File Edit View Search Terminal Help
test@vog:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr c6:b0:3a:43:c0:29
          inet addr:10.128.0.10  Bcast:10.128.0.11  Mask:255.255.255.252
          inet6 addr: fe80::c4b0:3aff:fe43:c029/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:128 errors:0 dropped:0 overruns:0 frame:0
          TX packets:323 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15396 (15.3 KB)  TX bytes:32330 (32.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:240 (240.0 B)  TX bytes:240 (240.0 B)

test@vog:~$ █

```

Figure 2. Output of the `ifconfig` command on an instance.

the first address of the subnet, is calculated by setting all the address bits to zero and adding one to the address [1].

According to Figure 1, the IP address of one of the devices of the node is 10.128.0.9. This corresponds to the wire address of the instance. It can be inferred that the instance is located on the node with a device that has the same address as the wire address of an instance. Therefore, to locate an instance, it is necessary to find a node with a network device that has an IP address that is one lower than the IP address of the instance.

Note that the instance is located without accessing the instance. Since the instance is not aware of the search, this technique helps protect the evidence from possible contamination.

8. Blocking Communications

Most cloud communications occur through a network [13]. Multiple methods exist to block network communications. One option is to bring down the network. Another option is to use a device such as a firewall to block network traffic.

A network on a Linux computer can be brought down using the command `ifconfig eth0 down`. We executed the command `ifconfig tap8d6cb50 down` on a node with a network interface to an instance to

stop the instance from sending and receiving data. (Attempting to establish an SSH session to the instance fails because no network traffic is allowed to the instance.) This method also stops the instance from communicating with the outside world. Note that this approach is not ideal because all communications are stopped; in fact, the cloud operating system and the investigators would not have access to the instance.

A firewall that blocks network traffic can be positioned on the node or in the cloud. When the firewall is used on the node, the communications involving a specific network device must be stopped; this network device is the virtual connection to the instance. The other option is to use a single large firewall in the cloud to control the network. In this case, the IP address of the instance is given to the firewall to block all communications to and from the instance. A central firewall is easier to configure and maintain. Also, the blocked traffic can be logged and subsequently provided to the investigator.

The cloud operating system can also be used to block communications to and from a specific instance. The operating system could also provide information about the nature of the communications and the other instances involved in the communications.

The methods used to block communications generally block all network traffic. In live investigations, custom software could be used to selectively block communications. For example, web traffic could be blocked as it could be irrelevant while SSH connections could be enabled. However, all open connections should be monitored to avoid unauthorized use.

9. Gathering Evidence

Obviously, it is important to gather all the possible evidence and ensure that it is not contaminated. Instances must be protected from contamination when performing an isolation process. When locating an instance, there is no direct communication or interaction with the instance. Nevertheless, communications should be blocked in a manner that minimizes interference with instances. The precautions taken in the previous steps can help protect instances from contamination.

The process of gathering possible evidence from instances that have been isolated is outside the scope of this paper. Interested readers are referred to [17] for details about evidence gathering in cloud systems.

10. Separation

In order for the crime scene to be “clean,” the node must contain only the suspect instance. This is accomplished by moving the instance to

a clean node or by moving the other instances from the node. Moving other instances has the advantage that the targeted instance would be unaware of the moving process.

Instances can be moved in a variety of ways. The simplest and most effective technique is to use the cloud operating system.

11. Isolation of Cooperating Instances

In a distributed instance system, evidence could be spread over multiple instances. One option is to gather evidence from all the instances. Another is to gather evidence from one instance. The third option constitutes the middle ground between the first two options.

Obviously, attempting to gather evidence from all the instances requires considerable time to locate all the instances and isolate them. During this time, evidence in the suspect instances could be contaminated. Another issue is the specific order in which the instances must be isolated. The isolation process should be performed in a manner that does not raise suspicion among the targeted instances.

Gathering evidence from just one instance in the cloud is the other extreme. The time required is much less, but the other suspect instances are left untouched in the investigation. If the distributed instance system is configured so that the instances mirror each other, then focusing on one instance may be adequate. But it is inappropriate if the system is fully distributed and each instance only contains a small subset of the evidence. However, the advantage is that the other suspect instances can be kept unaware of the isolation of the targeted instance. The loss of an instance is expected in a distributed instance system, and the other suspect instances may ignore the loss and continue to operate as normal.

The remaining option, which is a middle ground between the two previous options, is to collect evidence from a subset of instances. The number of instances to be isolated and the amount of resources to be dedicated depend on the investigation.

As in other digital forensic investigations, live or dead forensic techniques can be used to examine cloud systems [4]. However, two other types of techniques can be used in cloud forensics. One is a “half-dead” technique where the instance is shut down but the cloud node is running; in this case, the node is trusted and is used to gather evidence. The other is a “resurrected” technique, where the node is shut down but the instance has been restarted and is running in a new controlled environment.

A combination of live and dead techniques, which we call “community live,” can be used in a forensic investigation of a distributed instance

Table 1. Analyzed network traffic.

Address A	Address B	Packets	Bytes	Bytes A → B	Bytes A ← B	Duration
10.128.0.22	10.128.0.26	1,046	81,006	44,051	36,955	303.5644
10.128.0.26	10.128.0.34	1,100	85,200	38,900	46,300	294.145
10.128.0.26	10.128.0.30	1,102	85,332	38,900	46,432	292.0971
10.128.0.18	10.128.0.26	1,122	86,904	47,226	39,678	298.7616

cloud system. This involves isolating some instances and monitoring others. The instances that are isolated are subsets of the suspect instances, and are examined using dead techniques. Live techniques are used to monitor the network traffic, running processes and RAMs of other instances. Thus, a total of five types of forensic techniques can be used on a distributed instance cloud system.

Once again, the right balance must be struck between the various techniques. If too few instances are isolated, there may not be enough dead forensic evidence. On the other hand, if too many instances are isolated, some suspect instances might become aware of an ongoing forensic investigation and attempt to contaminate the evidence.

Table 1 shows the results of an experiment involving five cloud instances that were configured to work together. The instances sent network traffic to a controlling instance, which responded appropriately. The captured traffic was retrieved from the node on which one of the instances was located. The IP address of this instance was 10.128.0.26. The results demonstrate that network traffic can be used to detect instances that were working together. For example, the second row in Table 1 shows that instance 10.128.0.22 sent 44,051 bytes to instance 10.128.0.26 and instance 10.128.0.26 sent 36,955 bytes back to instance 10.128.0.22. Analysis of the traffic reveals that the instances 10.128.0.18, 10.128.0.22, 10.128.0.30 and 10.128.0.34 were working together.

12. Conclusions

The isolation of instances in a cloud computing environment is critical to conducting forensically sound digital investigations. The six isolation conditions specified for cloud instances provide valuable guidance to forensic investigators. The focus on distributed instance systems is also important. These cloud configurations, which incorporate multiple instances in a single logical resource to ensure availability, introduce additional challenges with regard to instance isolation. The solution is to

use specialized half-dead, resurrected and community live techniques, in addition to the standard live and dead forensic techniques.

Our future work will fully explore the concept of isolation in diverse cloud environments, with the goal of specifying cloud forensic methodologies that are both sound and efficient.

References

- [1] D. Barrett and T. King, *Computer Networking Illuminated*, Jones and Bartlett, Sudbury, Massachusetts, 2005.
- [2] S. Biggs and S. Vidalis, Cloud computing: The impact on digital forensic investigations, *Proceedings of the International Conference on Internet Technology and Secured Transactions*, pp. 1–6, 2009.
- [3] C. Binnig, D. Kossmann, T. Kraska and S. Loesing, How is the weather tomorrow? Towards a benchmark for the cloud, *Proceedings of the Second International Workshop on Testing Database Systems*, 2009.
- [4] M. Caloyannides, N. Memon and W. Venema, Digital forensics, *IEEE Security and Privacy*, vol. 7(2), pp. 16–17, 2009.
- [5] E. Casalicchio and S. Tucci, Static and dynamic scheduling algorithms for scalable web server farms, *Proceedings of the Ninth Euro-micro Workshop on Parallel and Distributed Processing*, pp. 369–376, 2001.
- [6] R. Chow, P. Golle, M. Jakobsson, E. Shi, J. Staddon, R. Masuoka and J. Molina, Controlling data in the cloud: Outsourcing computation without outsourcing control, *Proceedings of the ACM Workshop on Cloud Computing Security*, pp. 85–90, 2009.
- [7] F. Cohen, *Digital Forensic Evidence Examination*, ASP Press, Livermore, California, 2010.
- [8] I. Foster, Y. Zhao, I. Raicu and S. Lu, Cloud computing and grid computing 360-degree compared, *Proceedings of the Grid Computing Environments Workshop*, 2008.
- [9] M. Khosa (Ed.), *Infrastructure Mandate for Change 1994-1999*, Human Sciences Research Council, Pretoria, South Africa, 2001.
- [10] N. Lim and A. Khoo, Forensics of computers and handheld devices: Identical or fraternal twins? *Communications of the ACM*, vol. 52(6), pp. 132–135, 2009.
- [11] R. Lu, X. Lin, X. Liang and X. Shen, Secure provenance: The essential of bread and butter of data forensics in cloud computing, *Proceedings of the Fifth ACM Symposium on Information, Computer and Communications Security*, pp. 282–292, 2010.

- [12] J. Lyle, A strategy for testing hardware write block devices, *Digital Investigation*, vol. 3(S), pp. S3–S9, 2006.
- [13] P. Mell and T. Grance, The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology, NIST Special Publication 800-145, National Institute of Standards and Technology, Gaithersburg, Maryland, 2011.
- [14] Nimbus, Nimbus Director, Mountain View, California (nimbus.com/product).
- [15] Nitu, Configurability in SaaS (software as a service) applications, *Proceedings of the Second India Software Engineering Conference*, pp. 19–26, 2009.
- [16] Oxford University Press, Oxford Dictionaries, Oxford, United Kingdom (oxforddictionaries.com), 2012.
- [17] D. Ras and M. Olivier, Finding droplets in the cloud, in *Advances in Digital Forensics VIII*, G. Peterson and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 169–185, 2012.
- [18] G. Reese, *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud*, O’Reilly, Sebastopol, California, 2009.
- [19] K. Ruan, J. Carthy, M. Kechadi and M. Crosbie, Cloud forensics, in *Advances in Digital Forensics VII*, G. Peterson and S. Shenoi (Eds.), Springer, Heidelberg, Germany, pp. 35–46, 2011.
- [20] Technical Working Group for Electronic Crime Scene Investigation, Electronic Crime Scene Investigation: A Guide for First Responders, NIJ Guide, NCJ 187736, U.S. Department of Justice, Washington, DC, 2001.
- [21] M. Vouk, Cloud computing – Issues, research and implementations, *Proceedings of the Thirtieth International Conference on Information Technology Interfaces*, pp. 31–40, 2008.
- [22] P. White (Ed.), *Crime Scene to Court: The Essentials of Forensic Science*, Royal Society of Chemistry, Cambridge, United Kingdom, 2010.