Chapter 16

A TRUSTWORTHY CLOUD FORENSICS ENVIRONMENT

Shams Zawoad and Ragib Hasan

Abstract The rapid migration from traditional computing and storage models to cloud computing environments has made it necessary to support reliable forensic investigations in the cloud. However, current cloud computing environments often lack support for forensic investigations and the trustworthiness of evidence is often questionable because of the possibility of collusion between dishonest cloud providers, users and forensic investigators. This chapter presents a forensics-enabled cloud environment that supports trustworthy forensics in cloud environments. The forensic environment is designed on top of the OpenStack open-source cloud operating system. The environment enables cloud service providers to provide trusted digital forensics support to customers and forensic investigators, and enables customers to establish their own forensics-friendly infrastructures without making significant financial investments.

Keywords: Cloud computing, cloud forensics, trustworthy environment

1. Introduction

Consumers around the world avail of cloud computing services when they access Gmail, Google Calendar, Dropbox, Microsoft Office Live, or run Amazon Elastic Compute Cloud (EC2) instances. According to Gartner [8], consumers will store more than one third of their digital content in the cloud by 2016. A recent Market Research Media study [16] states that the global cloud computing market is expected to grow at a 30% compounded annual growth rate, reaching $270 billion in 2020.

However, the highly-scalable computing and storage resources offered by the cloud can be targeted by machines within the cloud [9, 13] or abused to store and distribute illegal images [22]. As a result, investigators are increasingly called upon to conduct digital forensic examina-

G. Peterson, S. Shenoi (Eds.): Advances in Digital Forensics XI, IFIP AICT 462, pp. 271–285, 2015.
DOI: 10.1007/978-3-319-24123-4_16

tions of cloud environments. This particular branch of digital forensics is known as cloud forensics.

Many of the traditional digital forensic assumptions do not transfer to cloud forensics. One of the major hurdles is that neither users nor investigators have physical access to the cloud. Moreover, each cloud server contains files belonging to multiple users, raising issues of privacy and cross-contamination. Even with a subpoena, it is not feasible to seize servers from a cloud service provider without violating the privacy of many other users. The trustworthiness of the collected evidence is also questionable because, aside from the cloud service provider's word, no consistent approach is available to determine the integrity of the evidence.

To control costs, cloud providers do not generally support persistent storage of terminated virtual machines (VMs). Hence, data residing in a cloud virtual machine is unavailable after it has terminated. This significantly hinders forensic investigations of illegal activities involving a virtual machine that has already terminated. Additionally, cloud providers or forensic investigators can collude with malicious users to hide traces of illegal activities or frame innocent users. For these reasons, it is imperative to provide reliable digital forensics support in cloud computing environments.

Several researchers have proposed solutions to address the challenges of cloud forensics [3, 7, 20, 23]. However, the solutions do not provide a complete cloud computing architecture that supports the extraction and preservation of trustworthy evidence. Moreover, the solutions do not consider the possibility of malicious cloud stakeholders and collusions between cloud stakeholders.

To support trustworthy forensics in cloud environments, it is necessary to collect and securely preserve logs, data attribution and provenance information, and timestamps. The required evidence should be made available to users, forensic investigators and legal authorities.

This chapter proposes a forensics-friendly cloud architecture – FECloud, which is designed on top of the OpenStack architecture and meets the major cloud forensics requirements discussed above. FECloud introduces five new components into the OpenStack architecture: (i) Logger (Themis); (ii) Data Possession Manager (Metis); (iii) Timestamp Manager (Chronos); (iv) Provenance Manager (Clio); and (v) Proof Publisher (Brizo). New functions are added to the OpenStack block storage (Cinder) and compute node (Nova) to communicate with the new components. The OpenStack dashboard (Horizon) and identity manager (Keystone) are augmented to provide user interfaces and authenticate the new OpenStack components. Finally, a forensics-enabled image for

virtual machines is available to provide digital forensic features. The FECloud environment enables cloud service providers to provide trusted digital forensics support to customers and forensic investigators. Also, it enables customers to establish their own forensics-friendly cloud infrastructures without making significant financial investments.

2. Related Work

McCormick et al. [17] have emphasized that digital evidence is not the counterpart of statements provided by humans, which should ideally be tested by cross-examination. Instead, the admissibility of digital evidence should be determined based on the reliability of the system and the processes that generated the evidence. Thus, the collection and preservation of trustworthy evidence are priorities in digital forensic investigations as well as in cloud forensic investigations.

Recognizing that isolation helps protect evidence from contamination, Delport et al. [6] have focused on isolating virtual machine instances to mitigate the multi-tenancy issue. Hay and Nance [12] have shown that if a virtual machine instance is compromised by installing a rootkit to hide the traces of malicious activity, it is still possible to identify the malicious activity by performing virtual machine introspection. To make the activity logs available to customers and forensic investigators, Birk and Wegener [3] have proposed that cloud providers only expose read-only APIs. Zawoad et al. [23] have proposed the concept of secure logging-as-a-service as a means to store virtual machine activities, which ensures the integrity and confidentiality of logs despite the possibility of malicious cloud providers and forensic investigators. Thorpe and Ray [20] have developed a log auditor for cloud environments that detects temporal inconsistencies in virtual machine timelines.

Dykstra and Sherman [7] have recently implemented FROST, a forensic data collection tool for OpenStack. Cloud users and forensic investigators can use FROST to acquire images of the virtual disks associated with user virtual machines and validate the integrity of the images using cryptographic checksums. It is also possible to collect logs of all API requests made to a cloud provider as well as OpenStack firewall logs for virtual machines. Data provenance in cloud computing is a relatively new research area that was first proposed by Muniswamy-Reddy et al. [18]; the researchers also developed a solution for gathering provenance data from Xen hypervisors [15]. More recently, Lu et al. [14] have introduced the concept of secure provenance in cloud environments; they proposed a trusted third party based scheme for secure cloud provenance that ensures data confidentiality, unforgeability and full anonymity of signa-

tures, and full traceability from a signature. Schmidt [19] has proposed a legal hold framework in cloud environments; however, the approach does not consider the trustworthy management of litigation holds to protect evidence from dishonest cloud providers, users or forensic investigators.

3. Desired Properties

Five properties are required for a trustworthy forensics environment to meet the unique characteristics of cloud systems:

- **Trustworthy Log Management:** It is often the case that experienced attackers tamper with system logs to hide their traces [1]. An adversary who hosts a botnet server, spam email server or phishing website in a cloud virtual machine can remove the traces of malicious activity by modifying the logs. Hence, a forensics-enabled cloud should acquire all activity logs from the virtual machines and store them in persistent storage while ensuring their integrity and confidentiality.

- **Proof of Data Possession:** Preserving proof of data possession is important to prove the presence of a specific file in a storage system at a certain time and to ensure the preservation of a litigation hold. In order to capture evidence that a suspect accessed a file in cloud storage at a certain time, a forensics-enabled cloud should preserve the proof of data possession. This also meets the requirements of a litigation hold, which is a notice to an organization to preserve all the electronically stored information (ESI) relevant to a lawsuit for a stipulated time period [21, 22].

- **Secure Timestamps:** Timestamps associated with digital evidence can be crucial to convict or acquit a suspect [5]. An attacker could change the system clock on a virtual machine before launching an attack and later reset it to the correct time as an anti-forensic measure. Tampering with the system clocks of a cloud host and virtual machine produces a set of events that are temporally coherent but different from the actual event times. Any event timeline generated from the system clocks of a host and guest virtual machines can, therefore, have an integrity problem. In order to guarantee trustworthy timelines in forensic investigations, it is necessary to ensure that all cloud system clocks have not been tampered with.

- **Secure Provenance:** Provenance is the history of an object, which includes its origins and use. Secure provenance helps a dig-

ital forensic investigator maintain proper chain of custody. However, because files and their access histories are under the control of cloud service providers, it is possible that the provenance records could be modified intentionally or unintentionally. An attacker who can access provenance records may obtain valuable and sensitive information about data stored in the cloud. A secure provenance scheme [10] must be implemented to protect provenance records from attacks on integrity and confidentiality.

- **Evidence Availability:** The physical inaccessibility of evidence residing in the cloud is always a challenge. A cloud service provider can support evidence acquisition by offering a secure software interface to authorized entities. Using the interface, customers and forensic investigators can collect network, process and database logs, as well as other digital evidence and the provenance records of the evidence.

4. Challenges

Achieving trustworthy forensic properties in current cloud infrastructures is challenging for several reasons:

- **Collusion:** Cloud customers and forensic investigators have limited control over evidence stored in the cloud. Despite the availability of state-of-the-art frameworks for collecting evidence from cloud systems, customers and investigators have no other option than to trust cloud service providers because they cannot verify the completeness and integrity of the evidence recovered from cloud systems.

  For example, an employee of a cloud provider could collude with a malicious user to hide important evidence or fabricate evidence that points to the innocence of the malicious user. A malicious cloud provider employee could also provide incomplete or modified logs, remove documents and their traces, maintain false timestamps and tamper with provenance information. A forensic investigator could also intentionally alter the evidence before presenting it in court. In a traditional computer system, only the user and investigator can collude. The three-way collusion in a cloud environment increases the attack surface and makes cloud forensics more challenging.

- **Volatile Data:** Data residing in a virtual machine is volatile, meaning that no data is preserved after the virtual machine is terminated. The volatile data includes documents, network logs,

operating system logs and registry logs. An entity who terminates a virtual machine after conducting malicious activities can cause vital evidence to be lost.

Cloud service providers can constantly monitor running virtual machines and store the volatile data in persistent storage in order to provide logs or proofs of data possession when needed. However, the routine preservation of the data of terminated virtual machines would overwhelm the storage resources of cloud providers. Therefore, it is necessary to find effective ways to preserve logs, data possession histories and provenance records.

- **Multi-Tenancy:** In cloud environments, multiple virtual machines share the same physical infrastructure (e.g., data belonging to multiple customers is co-located). A suspect could claim that the evidence collected from a cloud environment is associated with other cloud tenants, not the suspect. In this case, the forensic investigator has to prove that the evidence is actually associated with the suspect. In contrast, the owner of a traditional computing system is solely responsible for the electronically stored information in the computing system. Additionally, when conducting a forensic investigation of a cloud environment, it is imperative to protect the privacy of other cloud tenants.

5. FECloud Architecture

The FECloud forensics-enabled cloud architecture provides forensic investigators with the means to obtain and preserve cloud-based evidence in a secure manner. FECloud enhances the OpenStack cloud operating system by incorporating five components:

- **Logger (Themis):** This component collects logs from virtual machines, Cinder and Nova compute nodes and preserves them in a secure manner.
- **Data Possession (DP) Manager (Metis):** This component collects trace evidence about data possession from Cinder and stores the records in a secure manner.
- **Timestamp Manager (Chronos):** This component handles the timestamp verification cycles between the virtual machines, Nova compute node and itself, and preserves the verification information in a secure manner.
- **Provenance Manager (Clio):** This component collects various provenance records (data, application and state) from the virtual

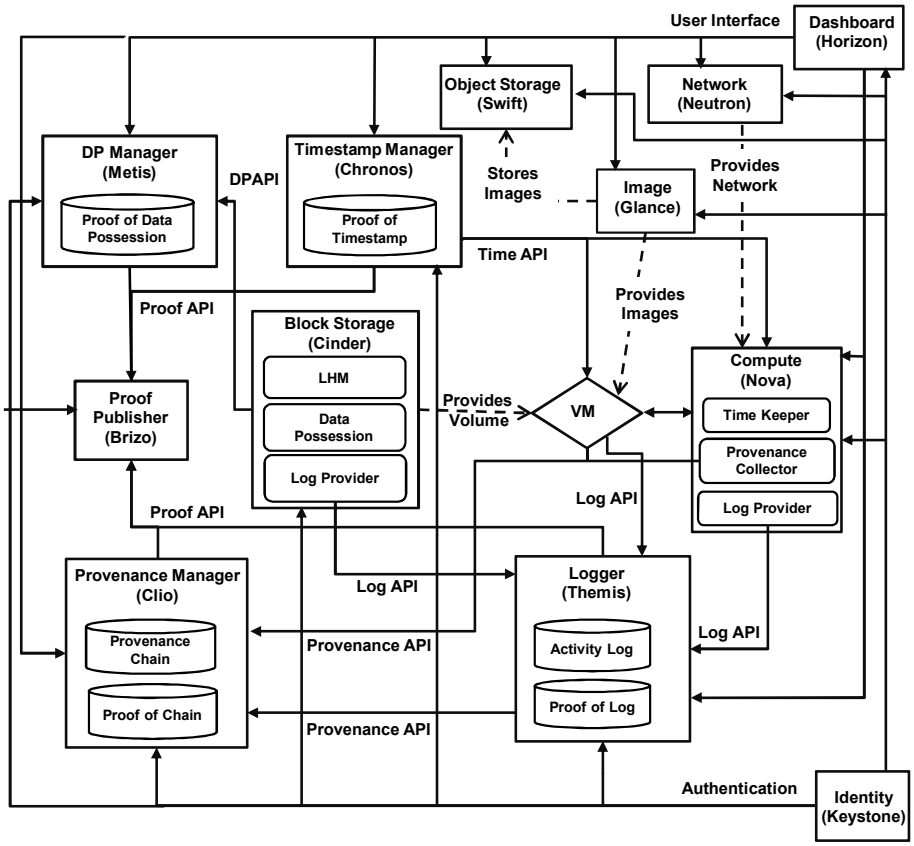

Figure 1. FECloud architecture.

machines, Nova compute node and Themis and securely creates and preserves the provenance chain.

- **Proof Publisher (Brizo):** This component distributes evidence traces so that any evidence alteration by cloud service providers, users or investigators can be detected.

Figure 1 presents the FECloud architecture. FECloud incorporates augmentations to the OpenStack dashboard (Horizon) and identity manager (Keystone) to support a user interface and the authentication of the new components. A forensics-enabled image for virtual machines that provides various forensic features is also available.

5.1 Logger (Themis)

The Logger (Themis) communicates with the OpenStack compute node (Nova), block storage (Cinder) and the running virtual machines

to collect all possible activity logs. To facilitate communications with Themis, a new Log Provider module is added to the Nova and Cinder nodes of OpenStack and virtual machine images.

The Log Provider module of the Nova compute node monitors the network activity and processor usage of running virtual machines and sends the logs to Themis. Logs that cannot be gathered from Nova (e.g., operating system logs) are collected directly from the virtual machines. The Log Provider module of Cinder sends logs of block storage usage to Themis. Logs from different entities are sent to Themis by the log API exposed by Themis.

When Themis receives a log entry via the log API, it stores the log in persistent storage in order to retain the log after the virtual machine has terminated. When a virtual machine is active, Themis tracks the data that belongs to the virtual machine and associates the data with the virtual machine user, ensuring that the data of multiple virtual machine users are not co-mingled. Data confidentiality is protected from unauthorized entities using public key encryption.

To prevent collusion between cloud service providers, forensic investigators and cloud users, Themis creates hashes of the logs using an accumulator data structure such as a one-way accumulator [2] or Bloom filter [4]. The computed hashes are compared with stored hash values to verify the integrity of the logs.

5.2 Data Possession Manager (Metis)

The Data Possession Manager (Metis) collects information about data possession from Cinder and stores the data possession records in a data possession database. Cinder is augmented with a Data Possession module to communicate with this new component.

A naive way to preserve data possession records is to store them in persistent storage. However, this increases the storage cost significantly. A more efficient way to preserve the records is to use an accumulator data structure [2, 4]. The accumulator also enables Metis to preserve data possession records without revealing the original data. Specifically, the membership checking method of the accumulator checks the data possession record of a suspect to verify if a document of interest belongs to the suspect without having to examine the document content.

Data possession records can also be used to identify violations of litigation holds. In such a scenario, a litigant presents all the documents that are under a litigation hold to the court. The verification method creates data possession information of the documents provided by the litigant. It then compares the generated data possession information

with the data possession records collected from the cloud. Document deletions are detected when the generated data possession information does not match the stored data possession records.

5.3 Timestamp Manager (Chronos)

Since it is not possible to prevent a virtual machine owner from changing the system time of a guest virtual machine or prevent a malicious system administrator from changing the system time of the host, a verification protocol is implemented using the Timestamp Manager (Chronos) to reveal clock changes. This secure timestamp verification protocol involves three entities, the Nova compute node, a running virtual machine and Chronos, in which each entity verifies the timestamp of the other two entities to detect timestamp alterations. Traces pertaining to the verification phase are stored securely using a hash-chain scheme in a timestamp database. Before beginning the verification cycle, the virtual machine and Chronos determine the round trip times with the Nova compute node. The validity of a requestor's timestamp depends on the current timestamp of the verifier and the round trip time values. The timestamp of one requestor is attested by the other two entities and each attestation is subsequently certified by an entity other than the verifier and requestor. A new Time Keeper module is incorporated in the Nova compute node to handle timestamp verification. Public key encryption and signature generation are used in all communications to preserve confidentiality and integrity.

This new feature enables a forensic investigator to present timestamp verification information along with the evidence collected from the cloud. Because the timestamp verification information is preserved using a hash-chain scheme, malicious entities cannot change the system time without breaking the verification chain.

5.4 Provenance Manager (Clio)

The Provenance Manager (Clio) extracts provenance records related to data, application and virtual machine state from the log database as well as from the provenance layer of the Nova compute node and the running virtual machines. Since the Logger (Themis) collects logs of data modifications from block storage, Clio collects the necessary log records to build the data provenance from Themis via the provenance API. Provenance records for the Virtual File System and applications running within virtual machines are directly collected from the virtual machines using the same API. Finally, provenance records for establish-

ing the system level provenance of the Nova compute node are collected from the provenance layer of Nova.

After collecting the various provenance records, Clio applies secure provenance chaining [11] to preserve the integrity of the provenance records. The secure provenance information is stored in the provenance chain database. To ensure that a malicious cloud service provider cannot modify the chain, the head of the provenance chain is stored in the proof of chain database after certain time periods.

5.5 Proof Publisher (Brizo)

The Proof Publisher (Brizo) periodically publishes the records of logs, data possession, timestamp verification and provenance chain on the web. Making these records publicly available prevents cloud service providers and investigators from altering or fabricating evidence because any manipulated evidence would not exist in the published record.

Information published by Brizo can be made available by an RSS feed to protect it from manipulation by cloud service providers. A trust model can also be established by engaging multiple cloud service providers in the publication process. Whenever one cloud service provider publishes a record, it is shared with the other cloud service providers. Therefore, a record can be considered to be valid as long as more than 50% of the cloud service providers are honest.

5.6 Evidence Access Interface (Horizon)

The OpenStack dashboard (Horizon) has been augmented to provide computerized access to cloud-based electronically stored information. Thus, physical access to cloud infrastructures is not necessary to acquire logs and data possession and provenance information. Four new modules are incorporated in Horizon to provide user interfaces for Metis, Chronos, Clio and Themis, with one module dedicated to each component. These modules enable users and investigators to collect activity logs, provenance, proof of data possession and proof of timestamp information in a secure and reliable manner.

5.7 Forensics-Enabled Image

The Nova compute node and Cinder do not provide all the evidence relevant to incidents in cloud environments. Indeed, without introducing new capabilities for virtual machines, it would not be possible to develop a complete forensics-enabled cloud environment.

Figure 2 shows a proposed forensics-enabled virtual machine image. A virtual machine launched using this image would be able to support

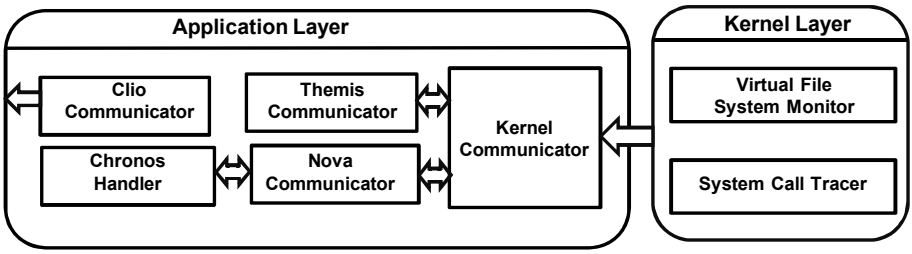

Figure 2. Forensics-enabled virtual machine image.

all the required forensic features. Some of the modules would be inside the kernel while others would be in the application layer. The following are the names and functionalities of the seven modules:

- **Virtual File System (VFS) Monitor:** This module is placed inside the kernel to trace virtual file system operations, which are important for constructing data provenance records for virtual machines.
- **System Call Tracer:** This module is placed inside the kernel to track all system calls. System call information can reveal the activities of cloud users and is important for establishing application and state provenance.
- **Kernel Communicator:** This module resides inside the application layer and acts as a bridge between the kernel and application layer. The module collects information from the Virtual File System Monitor and the System Call Tracer module in the kernel and feeds the information to other modules in the application layer.
- **Chronos Handler:** This application layer module participates in the timestamp verification step to verify the timestamps of the Nova compute node and Chronos, and also to have its own timestamp verified by the other two entities.
- **Themis Communicator:** This module collects the system call information and virtual file system activities from the Kernel Communicator and sends the logs to Themis using the log API.
- **Clio Communicator:** This module sends provenance records of applications, virtual file system and virtual machine state to Clio using the provenance API. The provenance records are collected from the Virtual File System Monitor and System Call Tracer via the Kernel Communicator module.

- **Nova Communicator:** This module is required for communications between the Nova compute node and the virtual machines. The Nova Communicator is also required during the timestamp verification phase during which the Nova compute node and a virtual machine mutually verify their timestamps.

5.8 Preliminary Results

The proof-of-concept FECloud implementation incorporates cryptographic frameworks for Metis [24] and Themis [23]. A Bloom-filter-based data possession scheme is available for storage-as-a-service cloud environments [24]. Experiments revealed that a FECloud user has 0.13% to 3.73% overhead in terms of time to upload files based on the file size and security properties. This overhead decreases as the file size increases and becomes almost constant when the file size is greater than 6 MB. The storage overhead on the cloud service provider side is also low. Regardless of the file size, approximately 1,262 bytes are required to preserve the data possession records for 1,000 files.

Themis uses the secure logging scheme from [23], which incorporates a Bloom filter and RSA accumulator to achieve the desired security properties. The design has $O(n)$ time and space complexity for log insertion and storage. The verification algorithm requires a constant amount of time to verify logs using both the accumulator schemes.

Current work related to FECloud involves securing the system time of the Nova compute node and the virtual machines with the assistance of Chronos. Initial experiments indicate that the timestamp verification cycle between the three entities can be executed every 60 seconds while introducing less than 1% system overhead on each entity. Another experiment, which involved running the verification protocol between 20 virtual machines, a Nova compute node and Chronos for 24 hours with a verification frequency of 60 seconds, recorded that the system was 99.98% stable.

6. Conclusions

At this time, forensic investigators are dependent on cloud service providers to identify and extract evidence from cloud computing environments. Unfortunately, under these circumstances, there is no way to verify if a cloud service provider has transmitted all the evidence that is relevant to a case and that the integrity of the evidence has been maintained. As a result, a forensic investigator has no way of knowing if the evidence is complete and valid. The FECloud architecture described in this chapter supports trustworthy forensics in cloud computing envi-

ronments. Designed on top of the OpenStack open-source cloud operating system, FECloud enables cloud service providers to provide trusted forensics support to customers and investigators, and enables customers to establish their own forensics-friendly infrastructures without making significant financial investments.

Future research will focus on the design of an efficient and secure cloud provenance scheme for Clio and the complete integration of all the proposed components within OpenStack. Following this, the overhead and stability of individual components and the integrated system will be evaluated using OpenStack benchmarking tools to establish the feasibility of using FECloud in real-world cloud environments.

Acknowledgement

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

References

[1] M. Bellare and B. Yee, Forward Integrity for Secure Audit Logs, Technical Report CS98-580, Department of Computer Science and Engineering, University of California at San Diego, San Diego, California, 1997.

[2] J. Benaloh and M. de Mare, One-way accumulators: A decentralized alternative to digital signatures, *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques*, pp. 274–285, 1994.

[3] D. Birk and C. Wegener, Technical issues of forensic investigations in cloud computing environments, *Proceedings of the Sixth IEEE International Workshop on Systematic Approaches to Digital Forensic Engineering*, 2011.

[4] B. Bloom, Space/time trade-offs in hash coding with allowable errors, *Communications of the ACM*, vol. 13(7), pp. 422–426, 1970.

[5] E. Casey, Error, uncertainty and loss in digital evidence, *International Journal of Digital Evidence*, vol. 1(2), 2002.

[6] W. Delport, M. Kohn and M. Olivier, Isolating a cloud instance for a digital forensic investigation, *Proceedings of the Information Security for South Africa Conference*, 2011.

[7] J. Dykstra and A. Sherman, Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform, *Digital Investigation*, vol. 10(S), pp. S87–S95, 2013.

[8] Gartner, Gartner says that consumers will store more than a third of their digital content in the cloud by 2016, Stamford, Connecticut (www.gartner.com/newsroom/id/2060215), June 25, 2012.

[9] D. Goodwin, Amazon cloud hosts nasty banking Trojan, *The Register*, July 29, 2011.

[10] R. Hasan, R. Sion and M. Winslett, Preventing history forgery with secure provenance, *ACM Transactions on Storage*, vol. 5(4), article no. 12, 2009.

[11] R. Hasan, R. Sion and M. Winslett, The case of the fake Picasso: Preventing history forgery with secure provenance, *Proceedings of the Seventh USENIX Conference on File and Storage Technologies*, pp. 1–14, 2009.

[12] B. Hay and K. Nance, Forensics examination of volatile system data using virtual introspection, *ACM SIGOPS Operating Systems Review*, vol. 42(3), pp. 74–82, 2008.

[13] Infosecurity Magazine, DDoS-ers launch attacks from Amazon EC2, July 30, 2014.

[14] 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.

[15] P. Macko and M. Chiarini, Collecting provenance via the Xen hypervisor, *Proceedings of the Third Workshop on the Theory and Practice of Provenance*, article no. 23, 2011.

[16] Market Research Media, Global cloud computing market forecast 2015-2020, San Francisco, California (www.marketresearchmedia.com/?p=839), January 8, 2014.

[17] C. McCormick, C. Tilford and J. Strong (Eds.), *McCormick on Evidence*, West, St. Paul, Minnesota, 1992.

[18] K. Muniswamy-Reddy, P. Macko and M. Seltzer, Making a cloud provenance-aware, *Proceedings of the First Workshop on the Theory and Practice of Provenance*, article no. 12, 2009.

[19] O. Schmidt, Managing a Legal Hold on Cloud Documents, U.S. Patent Application 20140012767, 2014.

[20] S. Thorpe and I. Ray, Detecting temporal inconsistency in virtual machine activity timelines, *Journal of Information Assurance and Security*, vol. 7(1), pp. 24–31, 2012.

[21] United States District Court (Southern District of Ohio, Eastern Division), Robert A, Brown et al. v. Tellermate Holdings Ltd. et al., Case No. 2: 11-cv-1122, 2014.

[22] United States District Court (Southern District of Texas, Houston Division), Quantlab Technologies Ltd. and Quantlab Financial LLC v. Godlevsky et al., Civil Action No. 4:09-CV-4039, 2014.

[23] S. Zawoad, A. Dutta and R. Hasan, SecLaaS: Secure logging-as-a-service for cloud forensics, *Proceedings of the Eighth ACM SIGSAC Symposium on Information, Computer and Communications Security*, pp. 219–230, 2013.

[24] S. Zawoad and R. Hasan, Towards building proofs of past data possession in cloud forensics, *ASE Science Journal*, vol. 1(4), pp. 195–207, 2012.