Chapter 17

LOCATING AND TRACKING DIGITAL OBJECTS IN THE CLOUD

Philip Trenwith and Hein Venter

Abstract One of the biggest stumbling blocks in a cloud forensic investigation is the inability to determine the physical location of a digital object in the cloud. In order to provide better accessibility to users, data in the cloud is not stored at a single location, but is spread over multiple data centers. This chapter proposes a model for providing data provenance and reporting on the provenance of a digital object in the cloud. It also examines how data provenance can be used to track where a digital object has been in the cloud and where the object can be found at any point in time. A special file type (wrapper) that encapsulates a digital object and its provenance data is proposed. The design helps preserve the integrity of provenance data and meets many of the requirements set for data provenance, including the support of cloud forensic investigations. The implementation requires each cloud service provider to maintain a central logging server that reports on the locations of wrapped objects in its domain.

Keywords: Cloud computing, cloud forensics, data provenance, central logging

1. Introduction

This chapter explores how data provenance can be stored in a forensically-ready manner and how provenance data can be used in cloud forensic investigations to trace where an object is located in the cloud and where the object has been. In a traditional digital forensic investigation, it is customary to seize a device suspected to be involved in illegal activities in order to examine it. However, this approach does not scale to the cloud for several reasons, the most important being that the physical locations of "devices" are often unknown due to the virtual nature of cloud environments. Data in the cloud is also spread across a range of hosts and data centers, which renders the task of identifying the physical

G. Peterson, S. Shenoi (Eds.): Advances in Digital Forensics XI, IFIP AICT 462, pp. 287–301, 2015.
DOI: 10.1007/978-3-319-24123-4_17

location of an object even more challenging. It is, therefore, necessary to investigate how digital forensic readiness and data provenance can provide a technique for acquiring potential evidence that is better suited to the architecture and operation of cloud computing environments.

The primary challenge in cloud forensic investigations is determining the locations of relevant data [15]. The ability to trace a digital object in the cloud is made possible by Locard's principle of exchange (as stated in [7]): "Whenever two objects come into contact with each other, each object is left with a trace of the other object."

An example of such a trace is seen in the registry of a machine running the Windows 7 operating system. When a flash drive is connected to the Windows 7 machine, a record is written to the registry. This record contains the hardware ID of the flash drive and the time it was inserted [1]. The record remains in the registry even after the flash drive has been removed.

The principle of exchange provides a means, in theory, of tracing an object in the cloud. However, collecting potential evidence from the cloud is a challenge because the approach is reactive. The challenge can potentially be addressed by establishing a proactive approach based on digital forensic readiness, in which an organization prepares in advance for the possibility that an event might occur that requires an investigation to be conducted.

The primary research question discussed in the chapter is: How can a digital object be tracked in the cloud and its history be recorded in order to obtain accurate location information during a cloud forensic investigation?

2. Background

This section discusses cloud computing, digital forensics and data provenance.

2.1 Cloud Computing

Cloud computing is a scaled-up, virtual environment for distributed computing that has become immensely popular in recent years. The U.S. National Institute of Standards and Technology (NIST) defines cloud computing as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction" [11]. The cloud provides three common service models. Infrastructure as a service (IaaS) offers computing and storage services to cloud users. Platform as a service (PaaS) offers platform

support for software development. Software as a service (SaaS) offers fully functional applications to cloud users. These service models can be deployed in one of four modes: (i) private cloud; (ii) public cloud; (iii) hybrid cloud; and (iv) community cloud [11]. The primary difference between cloud computing and traditional computing is that storage is distributed throughout the cloud and the stored data is not necessarily in the same location as the device being used to access the data. The picture seen by an end user is much the same as in traditional computing, but the picture for a digital forensic investigator is very different.

Virtualization makes it difficult to tell the physical location of a data object in the cloud. The locations of objects in the cloud are important because the objects have to be found and retrieved in order to present them (and their constituent data) as evidence. Therefore, it is very important to maintain the locations and histories of cloud objects.

Consider, for example, the European Union's Data Protection Directive [6], which stipulates that no sensitive data may leave the European Union. Implementing this directive makes it necessary to always know the locations of cloud objects containing sensitive data. Cloud service providers also duplicate and distribute data between multiple data centers to enhance availability, but this makes it difficult to assure that the data is (or is not) stored in a specific jurisdiction and that all copies of the data are removed if a deletion request is made. Pearson and Benameur [14] discuss many of the challenges associated with cloud services and securing the data that is generated and used by the services. Some of these challenges are considered later in this chapter in developing a solution for locating and tracking digital objects in the cloud.

2.2 Digital Forensics

Digital forensics is a formal process involving several steps that are executed on computing devices to answer questions related to investigations of computer crime and other incidents. Traditionally, if it is found that a device is needed for an investigation, the device is seized in order to examine it for evidence. This approach does not scale to cloud computing environments primarily because of their virtual nature. Indeed, seizing and examining a single device would almost certainly be insufficient to produce all the evidence related to an incident in a cloud computing environment. Reilly et al. [15] emphasize that obtaining access to physical devices in the cloud is one of the main stumbling blocks in cloud investigations. Gaining physical access to devices in the cloud may not always be possible but, if adequate logs are maintained and if

the information is made available to investigators when required, then the stumbling block could be overcome.

Traditional digital forensic investigations follow a reactive approach. An investigation is only initiated after an incident has occurred and the incident has been identified as one that requires further investigation. The reactive approach must be adapted to a proactive approach in order to perform successful cloud investigations. The proactive approach is better known as digital forensic readiness, which is defined by Rowlingson [16] as "the ability of an organization to maximize its potential to use digital evidence whilst minimizing the costs of an investigation." Digital forensic readiness requires an organization to take proactive measures to continuously collect and store data for a period of time to ensure that it is adequately prepared to conduct future investigations.

The intersection of digital forensics and cloud computing has created the new discipline of cloud forensics, which is defined by Ruan et al. [17] as "a cross discipline of cloud computing and digital forensics." Cloud forensics offers many challenges to investigators, including the inability to perform independent investigations because cloud service providers maintain full control of their computing environments and, thus, the sources of evidence. Barbara [3] states that the biggest challenges in cloud forensics are to determine the who, what, when, where, how and why of cloud-based activity. These challenges can potentially be addressed by the provision of data provenance – as Birk and Wegener [4] state, the history of a digital object, combined with a suitable authentication scheme, are crucial to cloud forensic investigations.

2.3 Data Provenance

Muniswamy-Reddy and Seltzer [13] define data provenance as the history of a digital object. Data provenance is valuable data in a digital forensic investigation because it reveals information such as when a digital object was modified and who accessed it. Some researchers [10, 13, 23] have investigated the use of data provenance in the cloud as a tool for supporting digital forensic investigations.

Lu et al. [10] discuss one of the challenges of cloud forensics known as anonymous authentication. Cloud computing offers anonymous authentication, which enables a user to log into a service using group authentication. In other words, a group of users may be granted access to specific objects and services in the cloud with the same access rights. While each user has unique credentials to access the service, there is no way to identify a specific user. This is because the credentials of the users in the group are related through mathematical inverses. Therefore,

the credentials are not stored by the system, only the inverse function is stored. The authentication scheme calculates the mathematical inverse of the provided credentials and compares it with the function value. Therefore, the credentials cannot be used to uniquely identify one individual from the other members of the group. This is advantageous to users but a challenge to forensic investigators. Lu et al. [10] argue that this is the reason why provenance is required in the cloud. They further state that the addition of provenance data that can be used to report on the history of a digital object would contribute to the wider acceptance of cloud computing by the general public. However, when implementing data provenance in a cloud computing environment, care must be taken not to infringe on the confidentiality and privacy of data owners.

3. Related Work

Muniswamy-Reddy et al. [12] have proposed a system that creates and maintains provenance-aware network storage. However, this system does not scale well to the cloud because it requires the network to be set up in a specific way to provide provenance data. The system is, thus, not compatible with all cloud environments.

Trenwith and Venter [23] have proposed a system that obtains provenance data from cloud service provider logs in network infrastructures (e.g., TCP/IP networks) to identify the physical locations of objects in the cloud. The underlying approach is leveraged in this work to provide data provenance for cloud objects while also addressing the challenges related to cloud forensics. One of the challenges, involving the provision of cryptographic proofs to verifying cloud data integrity, has been listed as a requirement for cloud forensics in [9, 18].

Trenwith and Venter [22] have developed a technique for proving the integrity of log files stored at a central log server. This is accomplished by computing a hash value when the logs are captured, encrypting the hash value with a secret key and storing the encrypted hash value along with the log. The integrity of the log file may be verified at any time by recomputing the hash value and then encrypting the hash value with a validation application that holds the secret key. The application then compares the newly encrypted value with the original encrypted value that was saved with the log file.

3.1 Storing Provenance Data

The provenance of a digital object can be stored in one of two ways. One is to embed provenance data in the digital object. The other is to store the provenance data separately in a second digital object [19].

- **Embedded Provenance Storage:** This technique modifies a digital object to incorporate provenance data within the object. The provenance data is typically stored in the file header. The main advantage of this technique is that it helps maintain the integrity of the provenance data because the data is stored with the object and can easily be verified. A disadvantage is that it is difficult to search the provenance data [19]. This technique is used by NASA's Flexible Image Transport System [8] and by the Spatial Data Transfer Standard [2] used by GIS systems to reference spatial data.

- **Separate Provenance Storage:** This technique stores provenance data separately from the object. The need to maintain a separate data object can be viewed as a disadvantage. However, Tan [21] suggests that centralized logging is vital to an efficient digital forensic strategy. Thus, with proper maintenance, the disadvantage of separate provenance storage can be turned into an advantage. Trenwith and Venter [23] have suggested the use of this technique with a central log server to store provenance data. The implementation is discussed later in this chapter.

3.2 Data Provenance Requirements

Lu et al. [10] identify three requirements for a provenance record:

R1: A provenance record must be unforgeable.

R2: A provenance record must be kept confidential.

R3: The integrity of a provenance record must be maintained by the system.

Trenwith and Venter [23] suggest an additional requirement that addresses the cloud forensics challenges faced by investigators. In particular, a provenance data should produce sufficient information to answer most of the questions asked by investigators during cloud forensic investigations. The new requirement is:

R4: A provenance record must answer the who, what, when, where and how of an event. These terms are defined as [23]:

- **R4.1:** Who: The identity of the process or user account associated with the modification.
- **R4.2:** What: The object that was modified.
- **R4.3:** When: The time of the modification.

R4.4: Where: The location of the object at the time of the modification.

R4.5: How: The hash values of the object before and after the modification.

4. Cloud-Based Provenance Model

The primary research question discussed in the chapter is: How can a digital object be tracked in the cloud and its history be recorded in order to obtain accurate location information during a cloud forensic investigation? Trenwith and Venter [23] have proposed the use of a central log server for storing the provenance data of digital objects. Although central logging provides advantages with regard to digital forensics, this approach is not without challenges. The principal challenge is the bandwidth required to ensure that the central server can continuously receive the provenance data of every object maintained by a cloud service provider. Therefore, this work investigates if a technique can be developed that provides the advantages of central storage while reducing the bandwidth requirements. This is accomplished by revisiting the embedding techniques used to store provenance data.

Trenwith and Venter [23] have made good arguments for storing provenance data separately from digital objects. Their goal was to use provenance data to identify and track object locations in the cloud. An examination of the requirements set for provenance data reveals that some of the requirements are easily met if provenance data is embedded within digital objects.

- **R1: A provenance record must be unforgeable:** Lu et al. [10] stipulate that it should not be possible to forge a provenance record. Because provenance records are produced by the system, it may be possible for a malicious attacker to forge a record that looks like a valid record. Therefore, instead of addressing the procedure that creates provenance records, it is necessary to see how provenance data is validated to prove its integrity. The approach is similar to that proposed in [22] and discussed above with regard to validating log files. If provenance data is embedded in a data object, the most recent provenance record can easily be validated against the data object. Taking this one step further, the system should store a second hash value to validate historical provenance data. Having the system encrypt the provenance data ensures that the non-forgeability requirement is met.

- **R2: A provenance record must be kept confidential:** This requirement can also be met using encryption. Encrypting provenance records and user data prevents cloud service provider employees as well as other parties from accessing the data, thus maintaining confidentiality. This concept is leveraged by Venters et al. [24] in their implementation of a secure provenance service.

- **R3: The integrity of a provenance record must be maintained by the system:** The integrity of a provenance record is more easily maintained if the record is embedded within the object. The integrity can easily be verified by checking the record against the data object. The integrity of historical provenance data can also be maintained by including historical data when computing the hash value of an object.

- **R4.1: Who (Identity of the process or user account associated with the modification):** This requirement can be met by recording the user account that was used to edit the object. If this is not possible, the identity of the process that modified the object should be recorded.

- **R4.2: What (Object that was modified):** When data provenance is embedded with the object, the issue of what was modified is irrelevant because provenance data is stored in the object that was modified.

- **R4.3: When (Time of the modification):** The time of the modification should be saved with the provenance data. To avoid confusion regarding timestamps, a cloud service provider should synchronize the time on all its servers [22].

- **R4.4: Where (Location of the object at the time of the modification):** A process running on a machine can easily record the machine name and the IP address assigned to the machine. However, this information is not enough to identify the location of a machine in the cloud. If the IP address is recorded along with the modification time, a cloud service provider can identify the location of an object by examining a log that stores the identity of the machine that was associated with the IP address at the time of interest. This approach assumes the worst case scenario where IP addresses are assigned dynamically. This is often not the case in the cloud, but adopting this approach ensures that dynamic IP address allocation is not a concern at a later stage.

This work leverages digital forensic readiness principles to help locate objects in the cloud. However, the difficulty is to identify the location of an object of interest when the data that identifies the object is stored within the object. One way to solve this problem is to maintain a central logging service for all objects being monitored.

Embedding provenance data within an object has the advantage that it is not necessary to transmit all the provenance data to a central server. This addresses the bandwidth issue identified in [23]. The approach only requires an identification tag to be transmitted to the central server along with the current location of the object. Hence, if additional provenance data has to be retrieved from the object, the current location of the object can be looked up at the central server after which the object can be retrieved from the cloud for further investigation. This is only possible if the user has not deleted the object from the cloud. If the object has been deleted, its provenance data is no longer available.

The problem posed by a deleted object can be addressed by a data deletion policy implemented by a cloud provider. For example, the provider could mark an object as deleted, but retain the object for a certain period of time, in case the object or its provenance data may constitute evidence in a future investigation.

- **R4.5: How (Hash values of the object before and after the modification):** Hash values are used to verify object integrity and to indicate when the object was modified. This work uses the SHA-256 hash algorithm. According to NIST [5], any hash algorithm stronger than 112-bits should be adequate until the year 2030.

5. Using a Central Logging Server

As discussed in the previous section, embedding provenance data in objects overcomes some of the disadvantages of current systems and helps meet most of the data provenance requirements. However, one problem that remains to be addressed is that only a few file types allow user-defined metadata to be appended to the file header. Therefore, the successful implementation of an embedded provenance system requires the creation of a file type that can be used to wrap a digital object and its provenance data in a single object. The next section discusses the design of a file wrapper that can embed provenance data with any digital object.

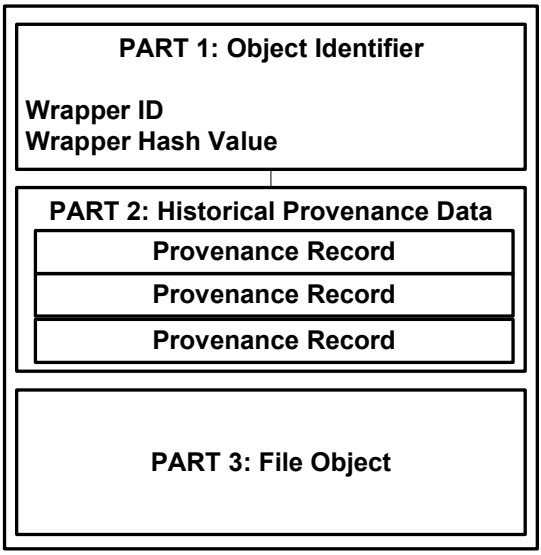

Figure 1. Wrapper design.

5.1 File Wrapper

This section describes the design of the wrapper used to embed a file and its provenance data in a single digital object. Figure 1 presents a schematic diagram of the file wrapper. The file wrapper has three components:

- **Object Identifier:** This component consists of the wrapper ID and a hash value. The wrapper ID is a unique key that is assigned to the object when it is constructed. The key, which is stored at the central server, is used to uniquely identify the object. This is necessary because there may be thousands of file objects wrapped with provenance data and it is necessary to know which wrapper contains a specific object. The hash value stored in the object identifier is computed by taking the hash value of the file object concatenated with its historical provenance data.
- **Historical Provenance Data:** This component is the provenance data of the object, which is covered by the requirements R4.1 through R4.5. The historical provenance data identifies when the object was constructed and includes information about subsequent modifications along with the time of each modification and the process or user account responsible for the modification. It also contains the hash value of the file object. Note that the historical provenance data includes not only the most recent modification,

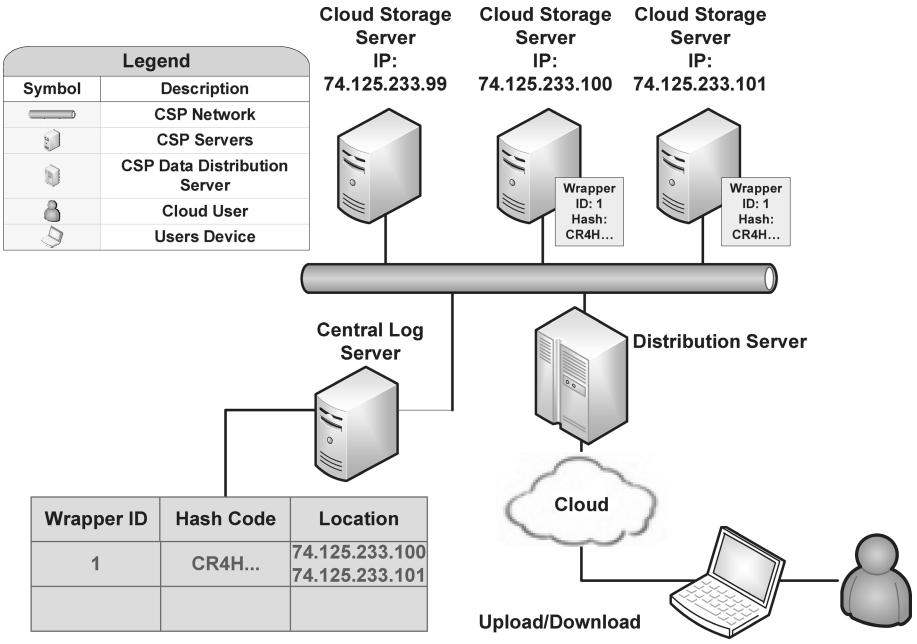

Figure 2. Wrapper implementation.

but also the provenance data associated with each modification and the file object hash value after each modification.

- **File Object:** This component is the user file; by design, it can be any type of file. A cloud service provider can decide what data is to be wrapped and how it is used. A cloud service provider may maintain provenance data for all the objects associated with its client accounts or it may maintain provenance data for a subset of objects associated with critical user data.

5.2 Tracking Wrapper Locations

Figure 2 shows a wrapper implementation by a cloud service provider such as Google Drive that could be used to track files uploaded by users. The cloud service provider implements the model by wrapping a file uploaded by a user before it is stored in the cloud. Before the wrapped file is stored in the cloud, the system registers the file and calculates a unique ID to identify and keep track of the wrapper. The hash value of the object is also computed and stored at the central server along with its location. The distribution server in Figure 2 distributes the object among its storage servers.

When a user logs in to access and modify the file, the object is unwrapped. When the user saves the file, the new provenance data is created and stored, and the file is wrapped. The wrapped object is required to connect to the central server to update its record:

- When the object is modified and the new hash value has been calculated, the central server records must be updated to reflect the latest hash value of the wrapped object.

- When the object is moved from one location to another, the location of the wrapped object that is stored at the central server must be updated.

If the cloud service provider maintains multiple copies of its data for backup purposes or to provide better data availability, all the locations of a wrapped object should be stored at the central server. The cloud service provider should ensure that changes made to an object are made concurrently to all copies of the object.

6. Discussion

Reilly et al. [15] note that the lack of knowledge of the physical locations of objects in the cloud is one of the main stumbling blocks in cloud forensics. The primary goal of this research has been to identify the physical locations of data in the cloud and, thereby, enhance cloud forensic investigations. The design helps preserve the integrity of provenance data and meets many of the requirements set for data provenance. In particular, the embedded data provenance system and the use of a central server enables digital forensic investigators to track the histories of digital objects and to easily identify their locations in the cloud if and when this becomes necessary.

Takabi et al. [20] emphasize that the owner of a data object should have full control over who accesses the data and what is done with the data. Encryption is the obvious solution to maintain data confidentiality, but this may introduce a performance penalty. However, the encryption technique is not intended to be employed by end users to encrypt their data. For this reason, it is suggested that user data be encrypted when it is uploaded to the cloud.

The technique also benefits the cloud service provider that controls the data. Since the data remains encrypted and is only decrypted when it is actually in use by the end user, there is no way for a malicious cloud provider employee to view, let alone compromise, the data. Thus, the user maintains full control over who can view or modify the data

because only individuals who are granted access to the data by the user may view or modify the data.

A potential problem is that provenance data can grow to become larger than the digital object it describes [19]. Data compression can be used to reduce the storage requirements. However, since there is no maximum limit on the size of the provenance data or the wrapper, the cloud service provider may wish to consider time and storage limits on the provenance data maintained with digital objects.

7. Conclusions

The proposed approach for locating and tracking digital objects in the cloud uses wrappers to encapsulate digital objects and their provenance data. The design helps preserve the integrity of provenance data and meets many of the requirements set for data provenance, including the support of cloud forensic investigations Also, it combines the advantages of embedded provenance storage with the strategic advantage of central logging.

Future research will address issues related to keeping user data inside specific jurisdictions and preventing data from crossing jurisdictional boundaries. Also, efforts will focus on developing a prototype and evaluating its performance.

8. Acknowledgement

This work was partially supported by the National Research Foundation of South Africa under Grant Nos. 88211, 89143 and TP13081227420.

References

[1] K. Alghafli, A. Jones and T. Martin, Forensic analysis of the Windows 7 registry, *Proceedings of the Eighth Australian Digital Forensics Conference*, 2010.

[2] P. Altheide, Spatial Data Transfer Standard, in *Encyclopedia of GIS*, S. Shekhar and H. Xiong (Eds.), Springer, New York, pp. 1087–1095, 2008.

[3] J. Barbara, Cloud computing: Another digital forensic challenge, *Digital Forensic Investigator News*, October 1, 2009.

[4] 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.

[5] B. Burr, NIST Hash Function Standards: Status and Plans, National Institute of Standards and Technology, Gaithersburg, Maryland (csrc.nist.gov/groups/SMA/ispab/documents/minutes/2005-12/B_Burr-Dec2005-ISPAB.pdf), 2005.

[6] F. Cate, The EU data protection directive, information privacy and the public interest, *Iowa Law Review*, vol. 80, pp. 431–443, 1995.

[7] Y. Guan, Digital forensics: Research challenges and open problems, tutorial presented at the *Thirteenth ACM Conference on Computer and Communications Security*, 2007.

[8] R. Hanisch, A. Farris, E. Greisen, W. Pence, B. Schlesinger, P. Teuben, R. Thompson and A. Warnock, Definition of the flexible image transport system, *Astronomy and Astrophysics*, vol. 376, pp. 359–380, 2001.

[9] A. Juels and B. Kaliski, PORs: Proofs of retrievability for large files, *Proceedings of the Fourteenth ACM Conference on Computer and Communications Security*, pp. 584–597, 2007.

[10] 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.

[11] P. Mell and T. Grance, The NIST Definition of Cloud Computing, Special Publication 800-145, National Institute of Standards and Technology, Gaithersburg, Maryland, 2011.

[12] K. Muniswamy-Reddy, U. Braun, D. Holland, P. Macko, D. MacLean, D. Margo, M. Seltzer and R. Smogor, Layering in provenance systems, *Proceedings of the Annual USENIX Technical Conference*, 2009.

[13] K. Muniswamy-Reddy and M. Seltzer, Provenance as first class cloud data, *ACM SIGOPS Operating Systems Review*, vol. 43(4), pp. 11–16, 2010.

[14] S. Pearson and A. Benameur, Privacy, security and trust issues arising from cloud computing, *Proceedings of the Second IEEE International Conference on Cloud Computing Technology and Science*, pp. 693–702, 2010.

[15] D. Reilly, C. Wren and T. Berry, Cloud computing: Forensic challenges for law enforcement, *Proceedings of the International Conference on Internet Technology and Secured Transactions*, 2010.

[16] R. Rowlingson, A ten step process for forensic readiness, *International Journal of Digital Evidence*, vol. 2(3), 2004.

[17] K. Ruan, J. Carthy, T. 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.

[18] Y. Shi, K. Zhang and Q. Li, A new data integrity verification mechanism for SaaS, *Proceedings of the International Conference on Web Information Systems and Mining*, pp. 236–243, 2010.

[19] Y. Simmhan, B. Plale and D. Gannon, A Survey of Data Provenance Techniques, Technical Report IUB-CS-TR618, Computer Science Department, Indiana University, Bloomington, Indiana, 2005.

[20] H. Takabi, J. Joshi and G. Ahn, Security and privacy challenges in cloud computing environments, *IEEE Security and Privacy*, vol. 8(6), pp. 24–31, 2010.

[21] J. Tan, Forensic readiness: Strategic thinking on incident response, presented at the *Second Annual CanSecWest Conference*, 2001.

[22] P. Trenwith and H. Venter, Digital forensic readiness in the cloud, *Proceedings of the Information Security for South Africa Conference*, 2013.

[23] P. Trenwith and H. Venter, A digital forensic model for providing better data provenance in the cloud, *Proceedings of the Information Security for South Africa Conference*, 2014.

[24] C. Venters, P. Townend, L. Lau, K. Djemame, V. Dimitrova, A. Marshall, J. Xu, C. Dibsdale, N. Taylor and J. Austin, J. McAvoy, M. Fletcher and S. Hobson, Provenance: Current directions and future challenges for service oriented computing, *Proceedings of the Sixth IEEE International Symposium on Service Oriented System Engineering*, pp. 262–267, 2011.