Analysis of Cloud Digital Evidence

Irfan Ahmed
University of New Orleans, USA

Vassil Roussev
University of New Orleans, USA

ABSTRACT

Cloud forensics analyzes digital evidence obtained from cloud computing environments. In the cloud, the traditional forensic focus of acquiring and analyzing snapshots of digital artifacts will becomes significantly less relevant as many of the artifacts are ephemeral, while the history of state changes become pervasive. This, over the medium-to-long term, the task of forensically reconstructing prior events will become easier over time as a growing fraction of the relevant information will be *explicitly* recorded. However, in the short term, the shift to cloud computing necessitates a paradigm change for forensic acquisition and processing tools from primarily artifact-centric to primarily log-centric. As the discussion will show, existing tools are ill-equipped to deal with the situation and we urgently need a new generation of cloud-focused forensic tools.

Keywords

Cloud forensics, Logging, Digital forensics, Cloud computing, IaaS, PaaS, SaaS

1 Introduction

Analysis of digital evidence acquired from cloud computing deployments, which we refer to as *cloud evidence* analysis, is in its very early stages of development. It is still in its exploration and experimentation phase where new, ad-hoc solutions are developed on a per-case basis; efforts are made to map problems in the cloud domain to prior solutions; and, most of all, ideas for the future are put forward. In other words, the state of knowledge is quite immature and that is well illustrated by the steady stream of recommendations—primarily from academia—on what *should* be done by providers and clients to make cloud forensics easier and better (for the existing toolset).

The goal of this chapter is to present a broad framework for reasoning about cloud forensic architectures and scenarios, and for the type of evidence they provide. We use this framework to classify and summarize the current state of knowledge, as well as to identify the blank spots and likely future direction of cloud forensic research and development.

1.1 Cloud forensics as a reactive technology

Since our discussion is primarily focused on the technical aspects of analyzing cloud evidence (and not on legal concerns), we adopt the following technical definition of digital forensics:

> *Digital forensics is the process of reconstructing the relevant sequence of events that have led to the currently observable state of a target IT system or (digital) artifacts*.
>
> The notion of *relevance* is inherently case-specific, and a big part of forensic analyst's expertise is the ability to identify case-relevant evidence. Frequently, a critical component of the forensic analysis is the causal *attribution* of event sequence to specific human actors of the system (such as users and administrators). When used in legal proceedings, the *provenance*, *reliability*, and *integrity* of the data used as evidence is of primary importance.

In other words, we view all efforts to perform system, or artifact analysis after the fact as a form of forensics. This includes common activities, such as incident response and internal investigations, which almost never result in any legal actions. On balance, only a tiny fraction of forensic analyses make it to the courtroom as formal evidence.

Digital forensics is fundamentally reactive in nature—we cannot investigate systems and artifacts that do not exist; we cannot have *best practices* before an experimental period during which different technical approaches are tried, (court-)tested, and validated. This means that there is always a lag between the introduction of a piece of information technology and the time an adequate corresponding forensic capability is in place. The evolution of the IT infrastructure is driven by economics and technology; forensics merely identifies and follows the digital breadcrumbs left behind.

It follows that forensic *research* is also inherently reactive and should focus primarily on understanding and adapting to the predominant IT landscape, as opposed to trying to shape it. Throughout the rest of this chapter, we will make the case that the cloud presents a new type of challenge for digital forensics, and that it requires an entirely different toolset, as the existing one becomes quickly and increasingly inadequate.

Less than ten years have elapsed since the introduction in 2006 of public cloud services by *Amazon* under the *Amazon Web Services* (*AWS*) brand. As of 2015, according to RightScale's *State of the Cloud Report* (RightScale, 2015), cloud adoption has become ubiquitous: 93% of businesses are at least experimenting with cloud deployments, with 82% adopting a *hybrid* strategy, which combines the use of multiple providers (usually in a public-private configuration). However, much of the technology transition is still ahead as 68% of enterprises have less than 20% of their application portfolio running in a cloud setup. Similarly, *Gartner* (Gartner, 2014) predicts another 2-5 years will be needed before cloud computing reaches the *"plateau of productivity"*, which marks mass mainstream adoption and widespread productivity gains.

Unsurprisingly, cloud forensics is still in its infancy despite dozens of articles in the literature over the last five years; there is a notable dearth of *practical* technical solutions on the analysis of cloud evidence. Thus, much of our discussion will necessarily tilt towards identifying new challenges and general approaches as opposed to summarizing existing experiences, of which there are few.

1.2 The new forensic landscape

Most cloud forensics discussions start with the false premise that, unless the current model of digital forensic processing is *directly* and faithfully reconstructed with respect to the cloud, then we are bound to lose all notions of completeness and integrity. The root of this misunderstanding is the use of traditional desktop-centric computational model that emerged in the 1990s as the point of reference. (This approach has been subsequently tweaked to work for successive generations of ever more mobile client devices.)

The key attribute of this model is that practically all computations take place on the device itself. Applications are monolithic, self-contained pieces of code that have immediate access to user input and consume it instantly with (almost) no trace left behind. Since a big part of forensics is attributing the observed state of the system to user-triggered events, we (forensic researchers and tool developers) have obsessively focused on two driving problems—discover every little piece of log/timestamp information, and extract every last bit of discarded data that applications and the OS leave behind either for performance reasons, or just plain sloppiness. Courtesy of countless hours spent on painstaking reverse engineering, we have become quite good at these tasks.

Indeed, our existing toolset is almost exclusively built to feast upon the leftovers of computations—an approach that is becoming more challenging even in traditional (non-cloud) cases. For example, *file carving* of acquired media (Richard, 2005) only exists because it is highly inefficient for the operating system to sanitize the media. However, for SSD devices, the opposite is true—they *need* to be prepared before reuse; the result—deleted data gets sanitized and there is little left to carve (King, 2011).

The very notion of low-level *physical* acquisition is reaching its expiration date even from a purely technological perspective—the upcoming 2015 generation of high-capacity HDD (8TB+) will use a *track shingling* technique and will have their very own ARM-based processor, which will be tasked with identifying hot and cold data and choosing appropriate physical representation for it. The HDD device will expose an object store interface (not unlike key-value databases) that will effectively make physical acquisition, in a traditional sense, impossible—legacy block level access will still supported but the block identifiers and physical layout will no longer be coupled as they were in prior generations of devices. By extension, the feasibility of most current data recovery efforts, such as file carving, will rapidly diminish.

Mass hardware disk encryption is another problem worth mentioning, as it is becoming increasingly *necessary* and *routine* in IT procedures. This is driven both by the fact that there is no observable

performance penalty, and the need to effectively sanitize ever larger and slower HDDs. The only practical solution to the latter is to always encrypt and dispose of the key when the disk needs to be reclaimed.

In sum, the whole concept of acquiring a *physical* image of the storage medium is increasingly technically infeasible and is progressively less relevant as interpreting the physical image requires understanding the (proprietary) internals of the device's data structures and algorithms. The inevitable conclusion is that forensic tools will have to increasingly rely on the logical view of the data presented by the device.

Logical evidence acquisition and processing will also be the norm in most cloud investigations, and it will be performed at an even higher level of abstraction via software-defined interfaces. Conceptually, the main difference between cloud computing and client-side computing is that *most* of the computation and, more importantly, the application logic executes on the server with the client becoming mostly a remote terminal for collecting user input (and environment information) and for displaying the results of the computation.

Another consequential trend is the way cloud-based software is developed and organized. Instead of one monolithic piece of code, the application logic is almost always decomposed into several layers and modules that interact with each other over well-defined service interfaces. Once the software components and their communication are formalized, it becomes quite easy to organize extensive logging of all aspects of the system. Indeed, it becomes *necessary* to have this information just to be able to test, debug, and monitor cloud-based applications and services. Eventually, this will end up helping forensics tremendously as important stages of computation are routinely logged, with user input being both the single most important source of events and the least demanding to store and process.

Returning to our driving problem of analyzing cloud artifacts, it should be clear that—*in principle*—our task of forensically reconstructing prior events should be becoming *easier* over time as a growing fraction of the relevant information is being *explicitly* recorded. As an example, consider the fact with modest reverse protocol engineering effort, Sommers (Sommers, 2014) was able to demonstrate that *Google Docs* timestamps and logs every single user keystroke. The *entire* history of the document can be replayed with a simple piece of *JavaScript* code; in fact, it appears that the log is the primary representation of the document itself, and the current (or prior) state of the document is computed on the fly by replaying (part of) the history. From a forensic perspective, this is a time travel machine and is practically everything we could ask for in terms of evidence collection with every user event recorded with microsecond accuracy.

1.3 Adapting to the new landscape

According to Ruan (Ruan, 2013), more forensic examiners see cloud computing as making forensics harder (46%) than easier (37%). However, the more revealing answers come from analyzing the reasoning behind these responses. The most frequent justifications for the *harder* answer fall into two categories: a) restricted access to data (due to lack of jurisdiction and/or cooperation from the cloud provider), and b) inability to physically control the process of evidence acquisition/recovery, and to apply currently-standard processing techniques. While data access is primarily a non-technical issue—to be solved by appropriate legal and contractual means—the natural inclination to attempt to apply legacy techniques will take some time and new tools to resolve.

All of the above concerns merely emphasize that we are in a transitional period during which requirements, procedures, and tools are still being fleshed out. From a technical perspective, the simplest way to appreciate why forensics professionals lack any *cloud-specific* tools is to consider the current dearth of tools that support the investigation of server deployments. Practically all integrated forensic

environments, both commercial and open source, focus on the client and the local execution model discussed earlier. However, a server-side inquiry is an exercise in tracking down and correlating the available logs—a task that, today, is accomplished largely in an ad-hoc manner using custom scripts and other "homegrown" solutions. Investigating a cloud application is not unlike performing a server side investigation; with a lot more log data and more complex data relationships.

From a research perspective, it may appear that log analysis does not present a very exciting prospect for the future. However, the real challenge will come from the sheer volume and variety of logs and the need to develop ever more intelligent tools to process and understand them. The long-term upside—and, likely, necessity—is that we can, for the first time, see a plausible path towards *substantially higher levels of automation* in forensics. At present, the process of extracting bits and pieces of artifacts and putting them together in a coherent story is too circuitous and vaguely defined for it to be further automated to a meaningful degree. Once log data becomes the primary source of evidence, it is conceivable that a substantial number of forensic questions could be formalized as data queries to be automatically answered, complete with formal statistical error estimates.

The transition from artifact-centric to log-centric forensics will take some time—artifact analysis methods will always be needed but they will be applied much more selectively. Nevertheless, it is a relatively safe prediction that, within a decade, that transition will fundamentally change forensic computing and the nature of investigative work.

1.4 Summary

Cloud computing presents a qualitatively new challenge for the existing set of forensic tools, which are focused on analyzing leftover artifacts on local client devices. Cloud services are inherently distributed and server-centric, which renders ineffective large parts of our present acquisition and analysis toolchain. In this transitional period, forensic analysts are struggling to bridge the gap using excess manual technical and legal effort to fit the new data sources into the existing pipeline.

Over the medium-to-long term, the field requires an entirely new set of cloud-native forensic capabilities that work in unison with cloud services. Once such tools are developed and established, this will open up the opportunity to define more formally legal and contractual requirements that providers can satisfy at a reasonable cost. As soon as technical and legal issues are worked out, we will face the unprecedented opportunity of automating and scaling up forensic analysis to a degree that is currently not possible.

The rest of this chapter is organized as follows: section 2 presents necessary background of cloud computing service models, followed by section 3 discussing the current approaches for each model. Section 4 presents potential approaches and solutions as a way forward to address forensics in cloud computing. Sections 5 and 6 present a detailed discussion and conclusion respectively.

2 Background

Cloud computing services are commonly classified into one of three canonical models—*software as a service* (SaaS), *platform as a service* (PaaS), and *infrastructure as a service* (IaaS)—and we use this split as a starting point in our discussion. We should note, however, that in reality the distinctions are often less clear cut, and a practical IT cloud solution (and a potential investigative target) can incorporate elements of all of these. As illustrated on Figure 1, it is useful to break down cloud-computing environments into a stack of layers (from lower to higher): *hardware* such as storage, and networking, *virtualization*

consisting of hypervisor allowing to install virtual machines, *operating system* installed on each virtual machine, *middleware* and *runtime* environment, and *application* and *data*.

In a private (cloud) deployment, the entire stack is hosted by the owner and the overall forensic picture is very similar to the problem of investigating a non-cloud IT target. Data ownership is clear as is the legal and procedural path to obtain it; indeed, the very use of the term *cloud* is mostly immaterial to forensics; therefore, we will not discuss this case any further.

In a public deployment, the SaaS/PaaS/IaaS classification becomes important as it indicates the ownership of data and service responsibilities. Figure 1 shows the typical ownership of layers by customer and service provider on different service models; Table 1 presents the examples of commercial products of the cloud service models.

In hybrid deployments, layer ownership can be split between the customer and the provider and/or across multiple providers. Further, it can change over time as, for example, the customer may handle the base load on owned infrastructure, but *burst* to the public cloud to handle peak demand, or system failures.

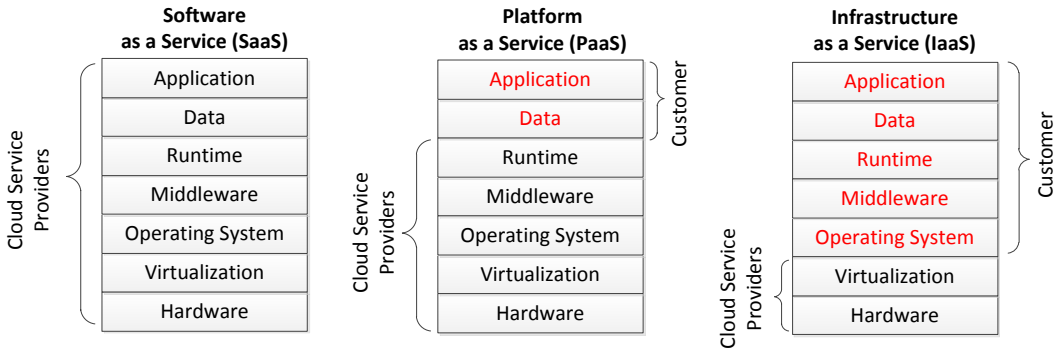

Figure 1 Layers of cloud computing environment owned by customer and cloud service provider on three service models: IaaS, PaaS, and SaaS (public cloud)

2.1 Software as a service (SaaS)

In this model, cloud service providers own all the layers including application layer that runs the software offered as a service to customers. In other words, customer has only indirect and incomplete control (if any) over the underlying operating infrastructure and application (in the form of policies). However, since cloud service provider (CSP) manages the infrastructure (including the application), the maintenance cost on customer side is substantially reduced. Google Gmail/Docs, Microsoft 365, Salesforce, Citrix GoToMeeting, Cisco WebEx are popular examples of SaaS, which run directly from web browser without downloading and installing any software. Their desktop and smartphone versions are also available to run on client machine. The applications have varying, but limited, presence on the client machine making the client an incomplete source of evidence; therefore, investigators would need access to server-side logs to paint a complete picture.

SaaS applications log extensively, especially when it comes to user-initiated events. For instance, Google Docs records every insert, update, and delete operation of characters performed by user along with the

timestamps, which makes it possible to identify specific changes made by different users in a document (Sommers, 2014). Clearly, such information is a treasure trove for a forensic analyst and is much more detailed and direct account of prior events than typically recoverable from a client device.

2.2 Platform as a service (PaaS)

In PaaS service model, customers develop their applications using software components built into middleware. Apprenda (Apprenda, 2014), and Google App Engine (GAE, 2014) are popular examples of PaaS, offering quick and cost-effective solution for developing, testing, and deploying customer applications. In this case, the cloud infrastructure hosts customer-developed applications and provides high-level services that simplify the development process. PaaS provides full control to customers on the application layer including interaction of applications with dependencies (such as databases, storage etc.), and enabling customers to perform extensive logging for forensics and security purposes.

2.3 Infrastructure as a service (IaaS)

In IaaS, the CSP is the party managing the virtual machines, however, this is done in direct response to customer requests. Customers then install operating system, and applications within the machine without any interference from the service providers. Amazon Web Service (AWS), Microsoft Azure, Google Compute Engine (GCE) are popular examples of IaaS. IaaS provides capabilities to take snapshots of the disk and physical memory of virtual machines, which has significant forensic value for quick acquisition of disk and memory. Since virtual machines have closely resemble physical machines, the traditional forensic tools for data acquisition and analysis can also be used inside the virtual machines as remote investigation of a physical machine is performed. Furthermore, virtual machine introspection provided by hypervisor enables cloud service providers to examine live memory and disk data, perform instant data acquisition and analysis. However, since the functionality is supported at hypervisor level, customers cannot take advantage of this functionality.

Table 1 Examples of some popular commercial products based on the cloud service models: Software as a service, Platform as a service, and Infrastructure as a service

Software as a service	Platform as a service	Infrastructure as a service
Google Gmail Microsoft 365 Salesforce Citrix GoToMeeting Cisco WebEx	Apprenda Google App Engine	Amazon Web Service Microsoft Azure Google Compute Engine

In summary, we can expect SaaS and PaaS investigations to a have high dependency on logs since disk and memory image acquisition is difficult to perform due to lack of control on middleware, operating system and lower layers. In IaaS, costumer has control on operating system and upper layers, which makes it possible to acquire disk and memory images, and perform traditional forensic investigation.

3 Current Approaches

The different architectural models shown on Figure 1 imply the need for a differentiated approach to building forensic tools for each one of them. An additional dimension to this challenge is that implementations of the same class of service—e.g., IaaS—can vary substantially across providers.

Moreover, providers themselves could be using and/or reselling other provider's services making the task of physically acquiring the source data impractically complicated, or even intractable.

Over time, we can expect the large (and still growing) number of cloud service types and implementation to naturally coalesce into a smaller set of de facto standards, which may eventually provide some needed visibility into the provider's operations. In the meantime, cloud forensic research is likely best served by focusing on the information available at the subscriber-provider boundary interface. The key observation is that providers *must* collect and retain substantial amounts of log information for accounting and operational purposes. For example, an IaaS provider must have detailed record of VM operation (launch, shutdown, CPU/network/disk usage), assignment of IP addresses, changes to firewall rules, long-term data storage requests, and so on. Such data should be readily available, segregated by subscriber and, therefore, readily obtainable via the legal process. However, invasive requests for physical media and information that cut to the core of a provider's operation are highly unlikely to succeed in the general case.

3.1 SaaS forensics

Cloud customers access SaaS applications such as Google Gmail, Google Docs, and Microsoft 365 through a web interface or desktop application from their personal computing devices such as laptop/desktop computers and smart phones. The applications maintain a detailed history/log of user inputs that is accessible to customers and has forensic value. Since the applications are accessed from a client machine, remnants of digital artifacts pertaining to user activities on the applications are usually present and can be forensically retrieved and analyzed from the hard disk of the machine.

3.1.1 Cloud-native application forensics

Perhaps the very first *cloud-native* tool with forensics applications is *DraftBack* (draftback.com): a browser extension created by the writer and programmer James Somers (Somers, 2014), which can replay the complete history of a *Google Docs* document. The primary intent of the code is to give writers the ability to look over their own shoulder and analyze how they write. Coincidentally, this is precisely what a forensic investigator would like to be able to do—rewind to any point in the life of a document, right to the very beginning.

In addition to providing the in-browser playback of all the editing actions—either in fast-forward, or real-time mode—*DraftBack* provides an analytical interface which maps the time of editing sessions to locations in the document (Figure 2). This can be used to narrow down the scope of inquiry for long-lived documents.

Somers' work, although not motivated by forensics, is probably the single best example of SaaS analysis that does *not* rely on trace data resident on the client—all results are produced solely by reverse engineering the web application's simple data protocol. Assuming that an investigator is in possession of valid user credentials (or such are provided by Google under legal order) the examination can be performed on the spot; any spot with a browser and an Internet connection.

The most profound forensic development here is that, as far as *Google Docs* is concerned, there is no such thing as deletion of data; every user editing action is recorded and timestamped. Indeed, even the investigator cannot spoil the evidence as any actions on the document will simply be added to the editing history with the appropriate timestamp. This setup is likely to be sufficient for most informal/internal scenarios, however, in order to make it to the court room, some additional tooling and procedure will need to be developed.

Clearly, the *acquisition* and long-term *preservation* of the evidence itself is yet to be addressed. The data component is the low-hanging fruit as the existing replay code can be modified to produce a log of the desired format. The challenging part is preserving the application logic that interprets the log; unlike client-side applications, a web app's code is split between the client and the server, and there is no practical way to acquire an archival copy of the execution environment as of a particular date. Since the data protocol is internal, there is no guarantee that a log acquired now could be replayed years later.

One possible solution is to produce a screencast video of the entire replayed user session. The downside here is that most documents are bigger than a single screen, so the video would have to be accompanied by periodic snapshots of the actual document (e.g., in PDF). Another approach would be to create a reference text editor playback application in which the logs could be replayed. This would require extra effort and faces questions of how closely the investigated application can be emulated by the reference one.

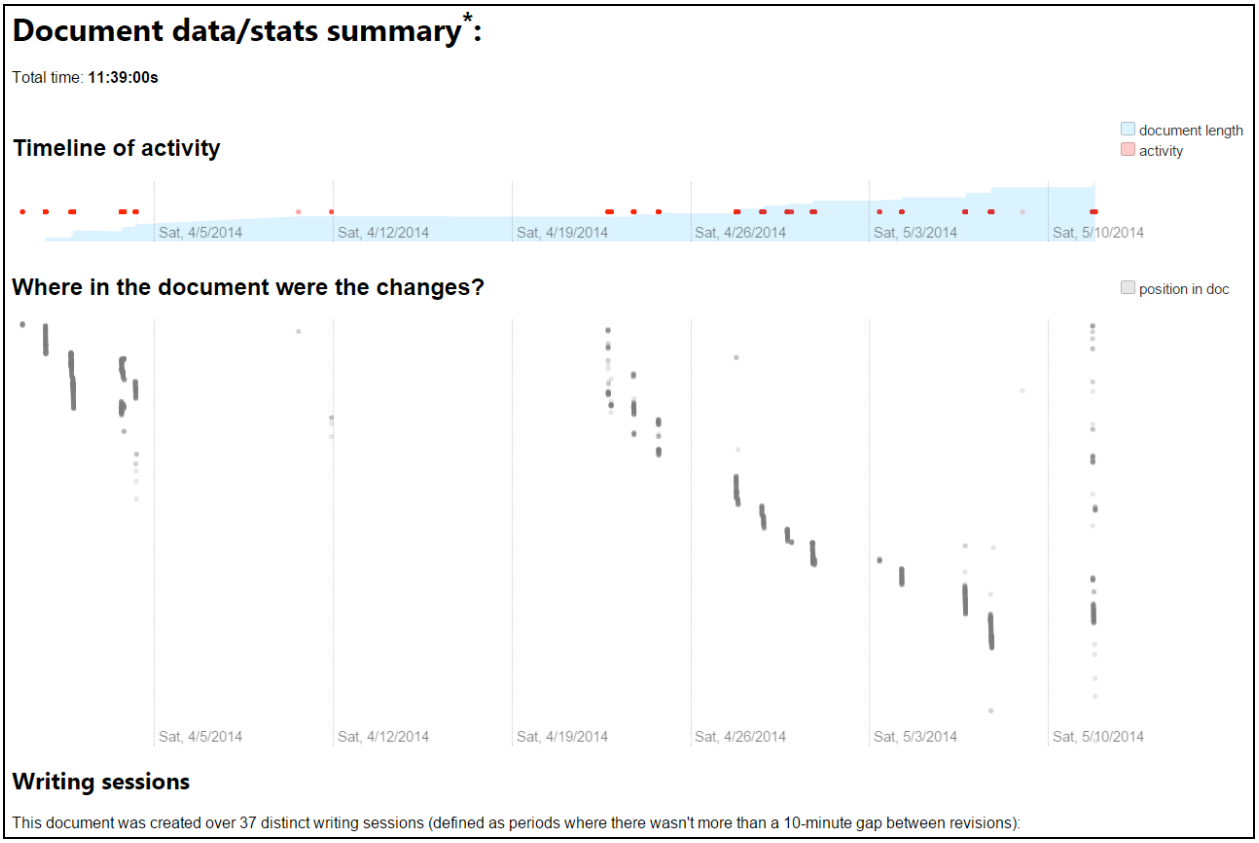

Figure 2 *DraftBack* analytical interface

More broadly, how common is fine grain user logging among cloud applications? As one would expect, a number of other editing applications from Google's *Apps for Work* suite—such as *Sheets*, *Slides*, and *Sites*—use a very similar model and provide scrupulously detailed history (*Slides* advertises "unlimited revision history"[1]). A deeper look reveals that the particular history available for a document depends on

[1] https://www.google.com/work/apps/business/products/slides/. Accessed: Dec 20, 2014

the age of the file and/or the size of the revisions as revisions may be merged to save storage space[2]. In other words, Google starts out with a detailed version of the history; over time, an active document's history may be trimmed and partially replaced with snapshots of its state. It appears that the process is primarily driven by technical and usability considerations, and not policy.

Microsoft's *Office 365* service also maintains detailed revisions based on edits in part because, like Google, it supports real-time collaboration. *Zoho* is another business suite of online apps that supports detailed history via a shared *track changes* feature similar to Microsoft Word's feature. Most cloud drive services offer generic file revision history; however, the available data is more limited as it is stored as simple snapshots.

3.1.2 Cloud drive forensics

Cloud users access their cloud storage through personal computing devices such as laptop/desktop computers, and smart phones. Chung et al. (Chung, 2012) suggest that the traces of services are present in client devices that can be helpful to investigate criminal cases, particularly when cloud service providers do not provide the cloud server logs to protect their client's privacy. Investigation on cloud storage identifies user activities from the subscription to the service till the end of using the service. Chung et al. analyze four cloud storage services (i.e. Amazon S3, Google Docs, Dropbox, and Evernote), and report that the services may create different artifacts depending on specific features of the services. Chung et al. proposed a process model for forensic investigation of cloud storage services. The model combines the collection and analysis of artifacts of cloud storage services from both personal computer and smart phones. The model suggests to acquiring volatile data from personal (Mac/Windows) computer (if possible), and then gathering data from the Internet history, log files, files, and directories. In Android phone, rooting is performed to gather data, and iTunes data is gathered from iPhone and the backup iTunes files from personal computer. The analysis checks for the traces of a cloud storage service exist in the collected data.

Hale (Hale, 2013) discusses the digital artifacts left behind after an Amazon Cloud Drive has been accessed or manipulated from a computer. Amazon's cloud storage service allows users to upload and download file from any location, without having a specific folder on local hard drive to sync with the cloud drive. User uses a desktop application or online web interface to transfer selected files or folders to/from cloud drive. The online interface is similar in appearance to Windows Explorer with *upload*, *download* and *delete* buttons to perform actions on files and folders. Desktop application also provides drag-and-drop facility. Hale analyzes the cloud drive by accessing and manipulating the drive's content via the desktop application, and online web interface. He found artifacts of the interface on web browser history, and cache files. The desktop application has artifacts on Windows registry, application installation files on default location, and a SQLite database used by the application to hold the transfer tasks (i.e. upload/download) while the task's status is pending.

Quick et al. (Quick, 2013) discuss the digital artifacts of Dropbox after a user has accessed and manipulated Dropbox content. Dropbox is a file hosting service enabling users to store and share files and folders, and is accessed through web browser or client software. Quick et al. use hash analysis and keyword searches to determine if Dropbox client software has been used. They determine Dropbox username from browser history (of Mozilla Firefox, Google Chrome, and Microsoft Internet Explorer),

[2] https://support.google.com/docs/answer/95902. Accessed: Dec 20, 2014

and the use of the Dropbox through several avenues such as directory listings, prefetch files, link files, thumbnails, registry, browser history, and memory captures.

Martini and Choo (Martini, 2013) discuss the digital artifacts of ownCloud – both server and client side. ownCloud is a file sync and share software configured and hosted on server and provides client software, and web interface to access files on the server. The authors recover the artifacts including Sync and file management metadata (such as logging, database and configuration data), cached files describing the files the user has stored on the client device and uploaded to the cloud environment or vise versa, and browser artifacts.

3.1.3 Building the new tools for SaaS forensics

As our review of existing work demonstrates, research and development efforts have been focused on the traditional approach of finding local artifacts on the client. This is an inherently limited and inefficient approach requiring substantial reverse engineering effort; the future of SaaS forensics lies in working *with* the web infrastructure the way web application do—through APIs.

Practically all user-facing cloud services strive to be platforms (i.e., they provide an API) in order to attract third-party developers who build apps and extensions enhancing the base product. The most relevant example is the wide availability of backup products, such as the ones provided by EMC's *Spanning*[3], which provide scheduled cloud-to-cloud backup service on multiple platforms—*Google Apps*, *Salesforce*, and *Office 365*. *Otixo* (http://otixo.com) is another service, which provides single sign-on and data transfer capability across more than 20 different SaaS platforms.

The latter provides a clear demonstration that the systematic acquisition of SaaS data can be readily accomplished via the provided APIs. At present, the amount of data in cloud storage is small relative to the size of local storage—free services only offer up to 15GB. However, as businesses and consumers get comfortable with the services, and more ready to pay for them, we can expect a fast and substantial increase in volume. For example, *Google Drive* currently offers up to 30TB at $10/month per terabyte.

This development is likely to blur the line between acquisition and analysis—as full acquisition becomes too burdensome and slow, one logical development would be to use the *search* interface offered by cloud service APIs to narrow down the scope of the acquisition. Procedurally, this aligns well with what already takes place during e-discovery procedures.

Looking further ahead, as cloud storage grows, it will be increasingly infeasible to acquire the data by downloading it over the Internet. This will bring a new impetus to the development of *forensics as a service* (FaaS), as the only practical means to perform the processing in a reasonable amount of time would be to collocate the data and the forensic computation in the same data center. Clearly, the practical use of FaaS is some years away and a non-trivial number of legal and regulatory issues would have to be addressed beforehand. Nonetheless, the fact that storage capacity growth constantly outpaces network bandwidth growth inexorably leads to the need to move the computation close to the data. Forensic computation will be no exception and procedures will have to be adjusted to account for technological realities.

---

[3] http://spanning.com/products/. Accessed: Dec 21, 2014

3.2 PaaS/IaaS forensics

PaaS packages middleware platforms on top of OS. Commercial products on PaaS are available such as Google App Engine (GAE, 2014). However, forensic research community did not pay much attention on PaaS, which is evident from lack of research papers on this topic. On the other hand, IaaS received attention because of the closest resemblance among service models to traditional computing infrastructure. It has virtual machines running contemporary operating systems and software, similar to physical machines. The conventional forensic tools for data acquisition and analysis such as WinDD, FTK, EnCase can be used on virtual machines. Furthermore IaaS offers unique features such as VM snapshotting for quick physical memory and HDD acquisition.

Dykstra et al. (Dykstra, 2012) evaluate effectiveness of EnCase, FTK, and three physical-memory acquisition tools (i.e. HBGary's Fastdump, Mandiant's Memoryze, and FTK Imager) in a cloud-computing environment. They remotely acquire geographically dispersed forensic evidence over Internet, and test the success at gathering evidence, the time to do so, and the trust required. They use a public cloud, EC2 from Amazon Web Services (AWS) for experiments and conclude that forensic tools are technically capable of remote data acquisition. They illustrate IaaS in six layers (from lower to higher): Network, Physical hardware, Host OS, Virtualization, Guest OS, and Guest application/data. Each layer (except the last) has to trust all its lower layers. Give the trust requirements, Dykstra et al. mention that technology alone is insufficient to produce trustworthy data and solve the cloud forensic acquisition problem. They recommend a management plane enabling consumers to manage and control virtual assets through an out of-band channel interfacing with the cloud infrastructure such as provided by Amazon Web Services a.k.a. AWS Management Console. The management plane interfaces with the provider's underlying filesystem and hypervisor, and is used to provision, start and stop virtual machines.

Dykstra (2013) developed a tool called *FROST*, which integrates forensic capabilities with OpenStack. FROST uses management plane through a website and application programming interface, and collects data from the host operating system level outside the guest virtual machines, assuming that the hardware, host operating system, hypervisor, and cloud employees are trusted. OpenStack creates a directory on the host operating system containing the virtual disk, ramdisk, and other host-specific files. FROST retrieves the files from the host and transforms virtual disk format to raw using the available utilities. For instance, QEMU provides utilities to convert QEMU QCOW2 images to raw format.

4 Proposed Comprehensive Approaches

At present, there are no dedicated deployable solutions that can handle cloud forensic tasks and evidence at the same level of comprehensiveness as current integrated forensic suites as already discussed. The key reasons for that are of procedural and technical nature. In many respects, evidence procedures have been—up until now—relatively easy to establish as the physical location and ownership of the hardware, software, and data were closely tied and readily accessible. In turn, this allowed much of legal evidence procedures to be directly translated to the digital world without having to rethink the rules. Cloud services break this model and will, eventually, require that legal procedures evolve to catch up with technology development. Technical challenges arise from the overall shift from client-centric to server-centric computations, which breaks many of the assumptions of traditional digital forensics and makes applying the currently prevalent processing workflow problematic.

Thus, efforts to address the challenges posed by cloud forensics in a general way have taken one of two approaches. Work coming from digital forensics researchers tends to heavily favor the procedural approach, which assumes that we need (primarily) new acquisition processes but the toolset and

investigative process will mostly remain the same. Often, it is assumed that new legal obligations will be placed on service providers and/or tenant.

The alternative is to look for new technical solutions; these need not be strictly forensics in origin, but may be addressing related problems in auditing, security and privacy. In essence, this would acknowledge the concept that cloud-based software works differently, and that we need a completely new toolset to perform effective and efficient forensic analysis on cloud systems.

4.1 Procedural expansion of existing forensic practices

Martini (2012) is a representative example of the approach embraced in the digital forensics literature—it seeks to identify the problems but has little in the way of technical detail. The focus is on minor refinements of the accepted procedural models of forensics, such as McKemmish (1999) and Kent (2006). Further, the authors prescribe a six-step process to get the final version of the new framework: *conceptual framework*, *explication interviews*, *technical experiments*, *framework refinement*, *validation interviews*, *finalized framework*.

It is entirely possible that digital forensic researchers and practitioners, as a community, will adopt such a deliberate, and time-consuming process to solve the problem; however, looking at the history of digital forensics, this seems highly unlikely. Experience tells us that the way new practices get established is much more improvised during disruptive technological transitions. Once enough experience, understanding, and acceptance of the new forensic techniques are gained, practices and procedures undergo revisions to account for the new development.

An instructive example in that regard is main memory forensics—as recently as ten years ago, best practices widely prescribed pulling the plug (literally) on any computer found running during search and seizure operations. This was not entirely unreasonable as dedicated memory forensics tools did not exist, so there was no extra evidence to be gained. Today, we have highly sophisticated tools to acquire and analyze memory images (Ligh, 2014) and they are the only key to solving many investigative scenarios. Accordingly, training and field manuals have been rewritten to point to the new state of knowledge.

To summarize, by their very nature, *standards lag technical development* and merely incorporate the successful innovations into the canon of best practices. In our view, it is critically important to understand that we *need* new technical solutions and no amount of procedural framework enhancement will bring them about. Technical advances are achieved by extensive and deliberate research and experimentation and often borrow and adapt methods from other fields.

4.2 API-centric acquisition and processing

In traditional forensic models, the investigator works with physical evidence carriers, such as storage media or integrated compute devices. Thus, it is possible to identify the computer performing the computations and the media that store (traces of) the processing, and to physically collect, preserve and analyze information content. Because of this, research has focused on discovering and acquiring every little piece of log and timestamp information, and extracting every last bit of discarded data that applications and the OS have left behind.

Conceptually, cloud computing breaks this model in two major ways. First, resources—CPU cycles, RAM, storage, etc.—are first pooled (e.g., RAID storage) and then allocated at a fine granularity. This

results in physical media potentially containing data owned by many users, and to data relevant to a single case being spread among numerous providers. Applying the conventional model creates a long list of procedural, legal, and technical problems that are unlikely to have an efficient solution in the general case. Second, both computations and storage contain a much more ephemeral record as virtual machine (VM) instances are created and destroyed with regularity and working storage gets sanitized.

As we discussed in the prior section, current work on cloud storage forensics has treated the problem as just another instance of application forensics. It applies basic differential analysis techniques to gain a basic understanding of the artifacts left on client devices by taking before and after snapshots of the target compute system, and deducing relevant cause and effect relationships. During an actual investigation, the analyst would be interpreting the state of the system based on these known relationships.

Unfortunately, there are several serious problems with this extension of existing client-side methods:

- *Completeness*. The reliance in client-side data can leave out critical case data. One example is older versions of the files, which most services provide; another one is cloud-only data, such as a *Google Docs* document, which literally has no serialized local representation other than a link. Some services, such as a variety of personal information management apps, live only in the browser so a flush of the cache would make them go away.
- *Reproducibility*. As cloud storage apps get updated on a regular basis and versions have a relatively short live span, it becomes harder to maintain the reproducibility of the analysis and make require frequent repetition of the original procedure.
- *Scalability*. As a continuation of the prior point, manual client-side analysis is burdensome and simply does not scale with the rapid growth of the variety of services and their versions.

We have performed some initial work using an alternative approach for the acquisition of evidence data from cloud storage providers; one that uses the official APIs provided by the services. Such an approach addresses most of the shortcomings above:

- *Correctness*. APIs are well-documented, official interfaces through which cloud apps on the client communicate with the service; they tend to change slowly and changes are clearly marked--only new features need to be incrementally incorporated into the acquisition tool.
- *Completeness/reproducibility*. It is easy to demonstrate completeness (based on the API specification) and reproducibility becomes straightforward.
- *Scalability*. There is no need to reverse-engineer the application logic. Web APIs tend to follow patterns, which makes it possible to adapt existing code to a new (similar) service with modest effort. It is often feasible to write an acquisition tool for a completely new service from scratch in a short time.

We have developed a proof-of-concept prototype called *kumodd*, which can perform complete (or partial) acquisition of a cloud storage account's data. It works with four popular services—*Dropbox*, *Box*, *Google Drive*, and Microsoft's *OneDrive*—and supports the acquisition of revisions and cloud-only documents. The prototype is written in *Python* and offers both command line and web-based user interfaces.

4.3 Audit-centric forensic services

The move to cloud computing raises a number of security, privacy, and audit problems among the parties involved—tenants, (multiple) cloud providers, and cloud brokers. The only practical means to address them is for all to have a trustworthy history—a *log*—of all the relevant events in the computation. Such logs are created on the boundary between clients and servers, as well as during the normal operation of services, typically in response to client requests.

For example, a tenant providing SaaS for medical professionals will need to convince itself and its auditors that it is complying with all relevant privacy and audit regulations. Since the computation executes on the provider's platform, the tenant needs the assurances of a third party, such as a trusted logging service.

In many cases, the same log information collected for other purposes can directly answer forensic queries, such as the history of user input over time. It is also likely to provide much greater detail than currently unavailable on the client. For example, *Google Cloud Storage* (Google Storage, 2015) maintains access logs in CSV (comma-separated values) format containing information about the access requests made on the cloud storage area allocated to user. Table 2 and 3 present the list of fields and their descriptions maintained in the log files on Google Cloud Storage, and Amazon Web Services.

Table 2: List of fields (with their data types and descriptions) used in Google Storage Access Log Format (Google Storage, 2015)

Field	**Type**	**Description**
time_micros	Integer	Time taken by a request to complete in microseconds
c_ip	String	The IP address of the client system from which the request is made
c_ip_type	Integer	The version of IP used i.e. either IPv4 or IPv6
cs_method	String	The HTTP method of the request from client to server
cs_uri	String	URI of the request
sc_status	Integer	HTTP status code sent from server to client
cs_bytes	Integer	Number of bytes sent in HTTP request message from client to server
sc_bytes	Integer	Number of bytes sent in HTTP response message from server to client
time_taken_micros	Integer	The time it took to serve the request by server in microseconds
cs_object	String	The object specified in the request
cs_operation	String	The Google cloud storage operation such as GET_Object.

Zavou et al. (Zavou, 2012) developed Cloudopsy – a visualization tool to address privacy concerns of a cloud customer about the third-party service/infrastructure providers handling customer data properly. The tool offers a user-friendly interface to the customers of cloud-hosted services to independently monitor the handling of sensitive data by third party. Cloudopsy's mechanism is divided into three main components: (1) the generation of the audit trails, (2) the transformation of the audit logs for efficient processing, and (3) the visualization of the resulting audit trails. Cloudopsy uses Circos visualization tool to generate graphs that enables users with no technical background to get a better understanding of the management of their data by third-part cloud services.

Pappas et al. (Pappas, 2013) proposed CloudFence, which is a data flow-tracking framework for cloud-based applications. CloudFence provides application programming interface (APIs) for integrating data flow tracking in cloud services marking sensitive user data to monitor the data propagation for protection. Pappas et al. implement a prototype of CloudFence using PIN without modifying applications. CloudFence can protect against information disclosure attacks with modest performance overhead.

Table 3: List of fields (and their descriptions) used in Amazon Web Services Server Access Log Format (AWS, 2015)

Field	**Description**
Time	The time at which the request was received
Remote IP	IP address of the requester
Requester	Canonical user id of the requester.
Request ID	The request ID is a string generated by Amazon S3 to uniquely identify each request
Operation	Such as SOAP.*operation*, and REST.*HTTP_method*
Request-URI	URI in HTTP request message
HTTP status	HTTP status code in response message
Error Code	Amazon S3 Error Code
Bytes Sent	Number of bytes sent in response message
Object Size	Total size of the object requested
Total Time	Number of milliseconds the request was in flight from the server's perspective
Turn-Around Time	Number of milliseconds that Amazon S3 spent processing your request

5 Discussion

Clearly, the discussed work in other relevant domains may not perfectly align with the typical needs of forensics. Nonetheless, it is representative of the *kind* of data that is likely to be available for forensic purposes. This can provide critical help in both building new analytical tools, and in defining *reasonable* additional data retention and provenance requirement for different cloud actors. In our view, one likely outcome is the emergence of *secure logging* services (SecLaaS) such as the ones proposed by Zawoad (2013).

Another bright spot for the future is the fact that logging is central to *all* aspects of the functioning of a cloud system and most of the forensic concerns related to investigating logs (Keyun (2013), Zawoad (2013)), such as cleanup, time synchronization, data correlation and timeline analysis, are quite generic. Therefore, a robust log management infrastructure will be available to the right forensic tools—Marty (2011).

Even as digital forensics practitioners struggle to come to terms with the current state of cloud computing, an even bigger wave of challenges is on the horizon. Indeed, the very definition of cloud forensics is likely to expand to include much of mobile device forensics (as more of the computation and logged data remain in the cloud) and the emerging concept of automated *machine-to-machine* interaction (a.k.a., IoT—*Internet of Things*). The latter is poised to dramatically escalate the amount of data generated as the current limiting factor—the human operator—is removed from the loop. In other words, the number of possible machine-to-machine interactions will no longer be tied to the number of humans and can readily grow at an exponential rate.

6 Conclusions

In this chapter, we argued that the acquisition and analysis of cloud artifacts is in its infancy, and that current generation tools are ill-suited to the task. Specifically, the continued focus on analyzing client-side artifacts is a direct extension of existing procedural approaches that, in the cloud context, fail to deliver on two critical forensic requirements: completeness and reproducibility. We have shown that SaaS cloud

services routinely provide versioning and use cloud-native artifacts, which demands a new API-centric approach to discovery, acquisition, and analysis.

Another major point of our analysis is that we are in the very early stages of a paradigm shift from artifact-centric to log-centric forensics. That is, the current focus on extracting snapshots in time of OS and application data structures (primarily out of the file system) will have wide applicability *only* to IaaS investigative scenarios. For PaaS/SaaS situations—which will be increasingly common—the natural approach is to work with the existing log data. In some cases, such as Google Docs, such data can provide a *complete* chronology of user edits since the creation of the document.

Finally, although data growth was identified as a primary concern for forensic tool design over a decade ago (Roussev, 2004), we are facing an even steeper curve as the IoT—built on automated machine-to-machine interaction—will escalate the amount of data available that (potentially) needs to be examined. This implies that the drive for ever higher levels of automated processing will no longer be just an issue of efficiency, but clear and present necessity. On the bright side, we note that logical (API-based) acquisition will *enable* higher levels of automated processing by eliminating tedious, low-level device acquisition and interpretation; the acquired data will have known structure and semantics, thereby eliminating much of the need for manual reverse engineering.

It is important to recognize that, in forensics, technical experimentation and development has always lead the way, with best practices and legal scrutiny following suit. At present, we are at the starting point of a major technology transition, which naturally leads to a moment of confusion, hesitation, and efforts to tweak old tools to a new purpose. There seems to some conviction that, more than ever, we need a multi-disciplinary effort to cloud forensics (NIST, 2014); that is, we should attempt to solve *all* problems—technical and non-technical—at the same time.

It is worth remembering that the current state of digital forensics is the result of some thirty years of research, development, and practice; all on client devices. We face a moment of disruption with the weight (and history) of computations shifting to the server environment. This is a substantial change, and 'big project' approaches almost never succeed in managing such an abrupt transition. What *does* succeed is small-scale technical experimentation, followed by tool development, and field use. The legal and procedural framework can only be meaningfully developed once a critical amount of experience is accumulated.

References

AWS (2015), Amazon Web Services (AWS) S3 Server Access Log Format, http://docs.aws.amazon.com/AmazonS3/latest/dev/LogFormat.html

Apprenda (2014), http://apprenda.com/?utm_source=library&utm_medium=post&utm_term=saaspaasiaas&utm_campaign=apprenda-com

Chung (2012). Chung, H., Park, J., Lee, S., & Kang C., "Digital forensic investigation of cloud storage services", *Journal of Digital Investigation*, Volume 9, Issue 2, November 2012, Pages 81-95, ISSN 1742-2876, http://dx.doi.org/10.1016/j.diin.2012.05.015.

Dykstra (2012). Dykstra, J., Sherman, A. T., "Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques", *Proceedings of the Twelfth Annual Digital Forensic Research Conference* (DFRWS), August 2012, Pages S90-S98, http://dx.doi.org/10.1016/j.diin.2012.05.001.

Dykstra (2013). Dykstra, J., & Sherman, A. T., "Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform", *Journal of Digital Investigation*, Volume 10, Supplement, August 2013, Pages S87-S95, ISSN 1742-2876, http://dx.doi.org/10.1016/j.diin.2013.06.010.

GAE (2014), Google App Engine, https://cloud.google.com/appengine/pricing

Gartner (2014). "Gartner's 2014 Hype Cycle of Emerging Technologies Maps", http://www.gartner.com/newsroom/id/2819918

Google Storage (2015), Google Storage access log format, https://cloud.google.com/storage/docs/access-logs

Hale (2013). Hale, J. S., "Amazon Cloud Drive forensic analysis", *Journal of Digital Investigation*, Volume 10, Issue 3, October 2013, Pages 259-265, http://dx.doi.org/10.1016/j.diin.2013.04.006.

Kent (2006). Kent K., Chevalier S., Grance T., & Dang H. "Guide to integrating forensic techniques into incident response. SP800–86. Gaithersburg: U.S. Department of Commerce; 2006.

King (2011). King, C., & Vidas, T.,"Empirical analysis of solid state disk data retention when used with contemporary operating systems." Journal of Digital Investigation 8 (August 2011), S111-S117. http://dx.doi.org10.1016/j.diin.2011.05.013.

Ligh, M. H., Case, A., Levy, J., & Walters, A., "The Art of Memory Forensics: Detecting Malware and Threats in Windows, Linux, and Mac Memory", Wiley, 2014. ISBN: 978-1118825099.

Martini (2012). Martini, B., & Choo, K. R.,"An integrated conceptual digital forensic framework for cloud computing," *Journal of Digital Investigation*, Volume 9, Issue 2, November 2012, Pages 71-80, ISSN 1742-2876, http://dx.doi.org/10.1016/j.diin.2012.07.001.

Martini (2013). Martini, B., & Choo, K. R., "Cloud storage forensics: *ownCloud* as a case study", *Journal of Digital Investigation*, Volume 10, Issue 4, December 2013, Pages 287-299, ISSN 1742-2876, http://dx.doi.org/10.1016/j.diin.2013.08.005.

Martini (2014). Martini, B., & Choo, K. R., "Distributed filesystem forensics: XtreemFS as a case study", *Journal of Digital Investigation*, Volume 11, Issue 4, December 2014, Pages 295-313, ISSN 1742-2876, http://dx.doi.org/10.1016/j.diin.2014.08.002.

Marty (2011). Marty, R., "Cloud application logging for forensics." In Proceedings of the 2011 ACM Symposium on Applied Computing (SAC '11). ACM, New York, NY, USA, 178-184. http://doi.acm.org/10.1145/1982185.1982226.

McKemmish (1999). McKemmish R., "What is forensic computing?" Trends & Issues in Crime and Criminal Justice 1999;118:1–6.

NIST (2014). NIST Cloud Computing Forensic Science Working Group, "NIST Cloud Computing Forensic Science Challenges" (draft NISTIR 8006), Jun 2014. http://csrc.nist.gov/publications/drafts/nistir-8006/draft_nistir_8006.pdf

Pappas (2013). Pappas, V., Kemerlis, V. P., Zavou, A., Polychronakis, M., & Keromytis, A. D., "CloudFence: Data Flow Tracking as a Cloud Service", RAID 2013, 411-431

Prakash (2014). Prakash, A., Venkataramani, E., Yin, H., & Lin, Z., "On the Trustworthiness of Memory Analysis--An Empirical Study from the Perspective of Binary Execution", IEEE Transactions on Dependable and Secure Computing (TDSC), 2014.

Quick (2013). Quick, D., Choo, K. R., "Dropbox analysis: Data remnants on user machines", *Journal of Digital Investigation*, Volume 10, Issue 1, June 2013, Pages 3-18, ISSN 1742-2876, http://dx.doi.org/10.1016/j.diin.2013.02.003.

Richard (2005). Richard, G., Roussev, V., "Scalpel: A Frugal, High-Performance File Carver". In Proceedings of the 2005 Digital Forensics Research Conference (DFRWS). Aug 2005, New Orleans, LA.

RightScale (2015). "RightScale 2015 State of the Cloud Report", http://assets.rightscale.com/uploads/pdfs/RightScale-2015-State-of-the-Cloud-Report.pdf

Roussev (2004). Roussev, V., Richard, G., "Breaking the Performance Wall: The Case for Distributed Digital Forensics". In Proceedings of the 2004 Digital Forensics Research Workshop (DFRWS). Aug 2004, Baltimore, MD.

Ruan (2011). Ruan, K., Carthy, J., Kechadi, T., & Crosbie, M., "Cloud forensics", In *Advances in Digital Forensics VII*, IFIP Advances in Information and Communication Technology Volume 361, 2011, pp 35-46, http://dx.doi.org/10.1007/978-3-642-24212-0_3.

Ruan (2013). Ruan, K., Carthy, J., Kechadi, T., & Baggili, I., "Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results", *Journal of Digital Investigation*, Volume 10, Issue 1, June 2013, Pages 34-43, http://dx.doi.org/10.1016/j.diin.2013.02.004.

Sommers (2014). Somers, J., “How I reverse engineered Google Docs to play back any document’s keystrokes”, http://features.jsomers.net/how-i-reverse-engineered-google-docs/. Accessed: Dec 20, 2014.

Zawoad (2013). Zawoad, S., & Hasan, R., “Cloud Forensics: A Meta-Study of Challenges, Approaches, and Open Problems,” arXiv:1302.6312.

Zavou (2013). Zavou, A., Pappas, V., Kemerlis, V. P., Polychronakis, M., Portokalidis, G., & Keromytis, A. D., “Cloudopsy: An Autopsy of Data Flows in the Cloud”. HCI (27) 2013: 366-375