Received: 5 February 2017 | Revised: 17 April 2017 | Accepted: 10 July 2017

DOI: 10.1002/cpe.4277

WILEY

SPECIAL ISSUE PAPER

CloudMe forensics: A case of big data forensic investigation

Yee-Yang Teing[1,2] | Ali Dehghantanha[2] | Kim-Kwang Raymond Choo[3]

[1] Department of Computer Science, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, Serdang, Selangor 43400, Malaysia

[2] The School of Computing, Science & Engineering, Newton Building, University of Salford, Salford, Greater Manchester M5 4WT, UK

[3] Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249-0631, USA

Correspondence
Kim-Kwang Raymond Choo, Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249-0631, USA.
Email: raymond.choo@fulbrightmail.org

Summary

The significant increase in the volume, variety, and velocity of data complicates cloud forensic efforts, and such (big) evidential data will, at some point, become too (computationally) expensive to be fully identified, collected, and analysed in a timely manner. Thus, it is important for digital forensic practitioners to have an up-to-date knowledge of relevant data artefacts that could be forensically recovered from the cloud product under investigation. In this paper, CloudMe, a popular cloud storage service, is studied. The types and locations of the artefacts relating to the installation and uninstallation of CloudMe client application, logging in and out, and file synchronization events from the computer desktop and mobile clients are described. Findings from this research will also help inform future development of tools and techniques (e.g., data mining techniques) for cloud-enabled big data endpoint forensics investigation.

KEYWORDS

Big data forensics, cloud forensics, CloudMe forensics, mobile forensics

1 | INTRODUCTION

With continuing advances in broadband and pervasive media devices (e.g., smartphones and tablets), it is not uncommon to find storage media in consumer devices containing terabytes (TB) of data. The Federal Bureau of Investigation's 15 Regional Computer Forensic Laboratories, for example, reported that the average amount of data they processed in 2014 is 22.10 times the amount of data 10 years ago, up from 22 to 5060 TB.[1,2] The increase in storage capacity has a direct impact on cloud forensics and operational investigations; hence, it is inevitable that big data solutions will become an integral part of cloud forensics.[3]

Due to the nature of cloud-enabled big data storage solutions, identification of forensic artefacts from the cloud hosting environment may be analogous to "finding a needle in a haystack."[4] The data could be segregated across multiple servers via virtualization.[5] Due to the lack of physical access to the cloud hosting environment, forensic examiners may need to rely on the cloud service provider (CSP) for preservation of evidence at a lower level of abstraction. This may, however, not be viable due to service level agreements between a CSP and its users.[6-14] Even if the location of the data could be identified, traditional practices and approaches to computer forensic investigation are unlikely to be adequate.[9] For example, existing digital forensic practices generally require a bit-by-bit copy of an entire storage media,[15-17] which is unrealistic and expensive on a large-scale dataset.[12] It has been demonstrated that it could take more than 9 hours to merely acquire 30 GB of data from an infrastructure as a service cloud environment.[18,19] Hence, the time required to acquire a significantly larger dataset could be considerably longer. These challenges are compounded in cross-jurisdictional investigations, which could prohibit the transfer of evidential data due to the lack of cross-nation legislative agreements in place.[16,20-26] Therefore, it is unsurprising that forensic analysis of cloud service endpoints (e.g., Android and iOS devices used to access cloud services, via either an application or a browser) remains an area of research interest.[23,27-33]

CloudMe (previously known as "iCloud") is a software as a service cloud model currently owned and operated by Xcerion.[34] The free version of CloudMe offers up to 19 GB storage space (with referral programme), and its premium version offers up to 500 GB storage space for individual users and 5 TB for business users.[35] CloudMe users may share contents with each other, as well as other public users, through email, text-messaging, Facebook, and Google sharing. There are three modes of sharing in CloudMe, namely: WebShare, WebShare+, and Collaborate. WebShare only permits one-way sharing, where the recipients are not allowed to make changes to the shared folder. WebShare+ allows users to upload files/folders only, while collaborative sharing allows the recipients to add, edit, or delete the content, even without the use of CloudMe client application.[36] The service can be accessed using the web user interface as an Internet file system or the client applications, which are available for Microsoft Windows, Linux, Mac OSX, Android, iOS, Google TV, Samsung Smart TV, Western Digital TV, Windows Storage Servers, Novell's

Dynamic File Services Suite, Novosoft Handy Backup, and others. CloudMe is also compatible with third-party software and Internet services, enabling file compression, encryption, document viewing, video and music streaming, and so on. through the web or client applications.[36]

In this paper, we seek to identify, collect, preserve, and analyse residual artefacts after using CloudMe cloud storage service on a range of endpoint devices, as these devices are typically available to forensic investigators. Evidence recovered from these endpoint devices could also be used to inform further investigation in a big data environment. Similar to the approaches of Quick and Choo,[31-33,37] we seek to determine the following in this research:

1. What residual artefacts remain on the storage media (hard drive and physical memory) after a user has used CloudMe desktop client application and web application?
2. Where are such data remnants located on a Windows, Ubuntu, and Mac OS client device?
3. What CloudMe residual artefacts remain on the internal memory, and where are such data remnants located on an Android and iOS client device?

We will briefly describe related work and the experimental set-up in the next two sections. In Section 4, we present the traces from the storage media and physical memory dumps of the desktop clients. Section 5 presents the findings from mobile clients and network traffic. We conclude the paper and outline potential future research areas in Section 6.

2 | RELATED WORK

The National Institute of Standard and Technology (NIST) defines cloud computing as

> [a] model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.[38]

The key aspects are to provide on-demand self-service, broad network access, resource pooling, rapid edacity, and measured services. There are three cloud computing service models,[38] namely, software as a service, platform as a service, and infrastructure as a service. NIST[38] also defined four deployment models, namely, public, private, community, and hybrid clouds. The public cloud is owned and operated by a provider organization. Users can subscribe to the service, say for a fee based on the storage or bandwidth usage. On the other hand, the private cloud is tailored to a single organization's needs. The cloud infrastructure that is administered by organizations sharing common concerns (e.g., mission, security requirements, policy, and compliance considerations) is called community cloud (e.g., San Antonio cloud for the community of San Antonio residents).

Cloud computing is not without its own unique forensics challenges.[39] Jurisdiction differences, loss of data control, physical inaccessibility of evidences, multi-tenancy, and lack of tools for large scale distributed and virtualized systems are often cited as key cloud forensic challenges.[40-43] Other related challenges include diverse range and types of digital media storage, decentralization, and utilization of anti-forensic and encryption techniques.[42,44,45] For example, Fahdi et al[46] found that the top three cloud forensic challenges according to digital forensic practitioners are volume of data, legal aspect, and time, while the top three challenges raised by digital forensic researchers are time, volume of data, and automation of forensic analysis.

In the review of the 2011 Australian Federal Government's Cybercrime Bill amendment on mutual legal assistance requests, Hooper et al[24] concluded that laws amendment in a single jurisdiction is unlikely to be adequate in addressing multi-jurisdiction investigation issues, such as in cloud computing environments. Martini and Choo,[7] Taylor et al,[47] and Daryabar et al[9] echoed the need for harmonizing relevant legislation across jurisdictions, although realistically it is challenging due to the different judicial and legal systems internationally. Our dependence on CSP in getting access to the evidential data compounds the challenges in cloud forensic investigation,[42,43,48,49] and researchers such as Farina et al[50] and Damshenas et al[3,11] have suggested using clearly defined service level agreements between CPSs and users to mitigate some of these challenges.

Martini and Choo[51] proposed the first cloud forensic investigation framework, which was derived based upon the frameworks of McKemmish[52] and NIST.[49] The framework was used to investigate ownCloud,[53] Amazon EC2,[18] VMWare,[54] and XtreemFS.[55] Quick et al[23] extended and validated the four-stage framework using SkyDrive, Dropbox, Google Drive, and ownCloud. Chung et al[56] proposed a methodology for cloud investigation on Windows, Mac OSX, iOS, and Android devices. Scanlon et al[57] outlined a methodology for remote acquisition of evidences from decentralized file synchronization networks and utilized it to investigate BitTorrent Sync.[58] In another study, Teing et al[27] proposed a methodology to investigate the BitTorrent Sync application (version 2.0) or any third-party and Original Equipment Manufacturer applications. Do et al[59] proposed an adversary model for digital forensics and demonstrated how such an adversary model can be used to investigate mobile devices (e.g., Android smartwatch and apps—Do et al[60]). Ab Rahman et al[61] proposed a conceptual forensic-by-design framework to integrate forensics tools and best practices in the design and development of cloud systems.

Marty[62] and Shields et al[63] proposed a proactive application-level logging mechanism designed to log information of forensics interest. However, Zawoad and Hasan[64] argued that the proposed solutions may not be viable in real-world scenarios. Dykstra and Sherman,[65] Gebhardt and Reiser,[66] Quick et al,[23] and Martini and Choo[54] presented methods and prototype implementations to support the (remote) collection of evidential

materials using application programming interfaces (API). Quick and Choo[37] and Teing et al[29] studied the integrity of data downloaded from the web and desktop clients of Dropbox, Google Drive, Skydrive, and Symform and determined that the act of downloading files from client applications does not breach the evidence integrity (e.g., no change in the hash values), despite changes in file creation/modification time.

In addition to remote collection of evidences, researchers also studied the potential of on-device collection of cloud artefacts such as from Evernote,[56] Amazon S3,[56] Dropbox,[33,56] Google Drive,[31,56] Microsoft Skydrive,[32] Amazon Cloud Drive,[67] BitTorrent Sync,[27,68] SugarSync,[69] Ubuntu One,[30] huBic,[70] Mega,[71] Syncany,[28] SpiderOak, JustCloud, pCloud,[72] and different mobile cloud apps.[20,73] Quick and Choo[31-33] also determined that data erasing tools such as Eraser and CCleaner may not completely remove the data remnants from Dropbox, Google Drive, and Microsoft SkyDrive.

In the context of CloudMe forensics, Dehghantanha and Dargahi[74] examined the artefacts from the Windows, Android, and iOS applications and recovered synced files, sync logs, databases of file synchronization caches, and user configuration files from the application folders. Similar findings on the CloudMe Windows application were reported by Amirullah et al.[75] Dehghantanha and Dargahi[74] also recovered copies of the files downloaded using the web application, alongside metadata such as file sizes and last accessed timestamps from the web browser caches. At the time of this research, we are not aware of any published research that focuses on forensic examinations of CloudMe on non-Windows desktop clients (e.g., Linux and Mac OS clients) thus the focus of this research.

3 | RESEARCH METHODOLOGY

We adopted the methodology in our previous research,[16,26-29,31-33] where we first established the test environments for the desktop and mobile clients. In this first step, we created a forensic workstation with the tools outlined in Table 1. We then created the desktop environment in the second step, which comprised three virtual machines (VMs) with the following configurations:

- Windows 8.1 Professional (Service Pack 1, 64-bit, build 9600) with 2 GB RAM and 20 GB hard drive.
- Ubuntu 14.04.1 LTS with 1 GB RAM and 20 GB hard disk.
- Mac OS X Mavericks 10.9.5 with 1 GB RAM and 60 GB hard drive.

The VMs were hosted using VMware Fusion Professional version 7.0.0 (2103067) on a Macbook Pro running Mac OS X Mavericks 10.9.5, with a 2.6 GHz Intel Core i7 processor and 16 GB of RAM. As explained by Quick and Choo,[31-33] it would have been time-consuming to replicate the experimental environment set-up on a physical workstation. The client mobile devices comprised a factory restored iPhone 4 running iOS 7.1.2 and an HTC One X running Android KitKat 4.4.4, which were jailbroken/rooted with "Pangu8 Version 1.1" and "Odin3 Version 185" (respectively) to enable root access to the user's partition. In the third step, we created a set of sample files for the file transfer experiments, which consisted of

TABLE 1 Tools prepared for Syncany forensics

Tools	Usage
FTK Imager version 3.2.0.0	To create a forensic image of the .VMDK files.
dd version 1.3.4-1	To produce a bit-for-bit image of the .VMEM files.
emf_decrypter.py	To decrypt the forensic image of iPhone.
Autopsy 3.1.1	To produce directory listings for the forensic images as well as extracting files and analyzing the windows registry, swap file/partition, and unallocated space from the forensic images.
HxD version 1.7.7.0	To conduct keyword searches in the forensic images.
Volatility 2.4	To extract the running processes and network record from the physical memory dumps, as well as dumping files from the memory space of the CloudMe client applications (i.e., using the "pslist," "netstat"/"netscan," and "memdump" functions).
SQLite Browser version 3.4.0	To view the contents of SQLite database files.
Photorec 7.0	To data carve the forensic images.
File juicer 4.45	To extract files from files.
BrowsingHistoryView v.1.60	To analyze the web browsing history.
Nirsoft Web Browser Passview v1.58	To recover the credential details stored in web browsers.
Nirsoft cache viewer, ChromeCacheView 1.56, MozillaCacheView 1.62, IECacheView 1.53	To analyze the web browsing caches.
Thumbcacheviewer version 1.0.2.7	To examine the Windows thumbnail cache.
Windows Event Viewer version 1.0	To view the Windows event logs.
Console version 10.10 (543)	To view log files.
Windows File Analyser 2.6.0.0	To analyze the Windows prefetch and link files.
NTFS Log Tracker	To parse and analyze the $LogFile, $MFT, and $UsnJrnl New Technology File System (NTFS) files.
Plist Explorer v1.0	To examine the contents of the Apple Property List (PLIST) files.

copies of the 3111th email message of the Berkeley Enron email dataset (downloaded from http://bailando.sims.berkeley.edu/enron_email.html) that were saved in .rtf, .txt, .docx, .jpg (print screen), .zip, and .pdf formats. This provides a basis for replicating the experiment in future.

Similar to previous studies,[29,32,69,76] the fourth step of the methodology involved conducting a predefined set of experiments such as installation and uninstallation of the CloudMe client applications as well as uploading, downloading, viewing, deleting, unsyncing, sharing, and inactivating sync files/folders to simulate various real-world scenarios of using the CloudMe desktop, mobile, and web applications. The web application was accessed using the Google Chrome client for Windows version 51.0.2704.103m. Before each experiment, we made a base snapshot of each VM workstation to serve as the control case. Similar to our previous research,[31-33] we created a snapshot of the VM workstations after each experiment, prior to taking a copy of the virtual memory and disk file (after system′s shutdown) in bit-stream (dd) and Encase Evidence (E01) formats, respectively. This is to prevent the memory/image acquisition tools from altering the data in the storage media and physical memory. As for the mobile clients, we made a binary image of the file system using "dd" over SSH/ADB Shell.

We collected data relevant to the CloudMe investigation in the fifth step, prior to analysing the data in the sixth step. In the former, we extracted data that matched the terms "cloudme," "xcerion," and "Enron3111" from the forensic images for analysis using the tools of relevance in the latter. These included SQLite database files,[77] PLIST files, prefetch files, event logs, shortcuts, thumbnail cache, $MFT, $LogFile, $UsnJrnl, and web browser files (e.g., in *%AppData%\Local\Google*, *%AppData%\Local\Microsoft\Windows\WebCache*, *%AppData%\Roaming\Mozilla*, *%AppData%\Local\Microsoft\Windows\Temporary Files\index.dat*). The artefacts from the physical memory dump were collected using Volatility, Photorec file carver, and HxD Hex Editor; network traffic using Wireshark and NetMiner. For all the data collected, both MD5 and SHA1 hash values were calculated and subsequently verified. All experiments were repeated thrice (at different dates) to ensure consistency of findings.

4 | ANALYSIS OF CLOUDME DESKTOP CLIENTS

The installation of the CloudMe desktop clients created the data directory at *%AppData%\Local\CloudMe*, */home/<User Profile>/.local/share/CloudMe*, */Users/<User Profile>/Library/Application Support/CloudMe* on the Windows, Ubuntu, and Mac OS desktop clients. The sync (download) folders were located in the OS′ Documents directory, such as *%Users%\<User Profile>\Documents*, */home/<User Profile>/Documents*, */User/<User Profile>/Documents/* on the Windows, Ubuntu, and Mac OS clients by default. When the sync folders with the option "When delete folder in the cloud and all its content is selected." in the client applications were deleted, we observed that the sync folders remained locally but were removed completely from the server. In all scenarios, the data and download directories remained after uninstallation of the client applications.

4.1 | Cache.db database

The file synchronization metadata and cloud transaction records could be predominantly located in the */%CloudMe%/cache.db* database (in the data directory). The tables of forensic interest are "user_table," "syncfolder_table," "syncfolder_folder_table," and "syncfolder_document_table." The user_table holds the property information of users who had logged in from the desktop client applications; syncfolder_table maintains a list of metadata associated with the sync folder(s) added by or download to the local device; syncfolder_folder_table keeps track of the tree structure for the sync folder(s); and syncfolder_document_table records the metadata associated with the synced files in the sync folder(s). Table columns of forensic interest are presented in Table 2.

To construct a meaningful file synchronization timeline, we threaded the data fields in the four tables to provide the information such as the following: Which are the synced files? Where are the locations? Who is the owner of the files? What time were the files created? What is the last sync time? Figure 1 shows the SQL query used to parse Cache.db and produce synchronization history shown in Figure 2.

4.2 | CloudMe registry, Sync.conf, and com.CloudMe.Sync.plist files

An examination of the Windows registry revealed the username for the currently logged in user and the device name in *HKEY_USERS\<SID>\Software\CloudMe\Sync\startup\me* and *HKEY_USERS\<SID>\Software\CloudMe\Sync\<Username>_xClientId* (respectively). The username can be a useful identifying information for the cache.db database′s remnants, i.e., locating copies of the "user_table" data in physical memory dumps. The client ID is a unique 32-character alphanumeric string used to identify a CloudMe device, which can be used to correlate residual evidences.

In Ubuntu client, both username and clientID were located in the */home/<User Profile>/.config/CloudMe/Sync.conf* file, by looking at values for attributes "me" (of the "startup" property) and "_xClientId" (of the "Username" property), respectively. In the Mac OSX client, Username and ClientID were located in the "startup.me" and "<Username>.xClientId" properties of the */Users/<User Profile>/Library/Preferences/com.CloudMe.Sync.plist* file.

4.3 | CloudMe log files

Log files play a vital role in an incident investigation.[13] The CloudMe log files are located in the "logs" subdirectory of the application directory and named as [Year-Month-Day].txt. Although the log file only recorded application errors, it was possible to identify the file synchronization time alongside the sync path from the log entries such as "*2016-03-15 14:52:02: CloudMeUnthreaded: Request error: "/Users/alice/Documents/UbuntuShareFolder/UbuntuSubFolder/UbuntuSubFolder/Enron3111.docx*" | *"Error downloading https://os.cloudme.com/v1/users/12886417622/*

TABLE 2 Tables and table columns of forensic interests from cache.db

Table	Table Column	Relevance
user_table	user_id	A unique numerical user ID for the user(s) logged in from the local device. This ID could assist a practitioner in correlating any user-specific data that might have been obtained from other sources of evidence.
	username	Username provided by the user during registration.
	devicename	Device name provided by the user during registration.
	created	Holds the addition time of the user account(s) in datetime format.
syncfolder_table	owner	Owner's ID which correlates with the "user_ID" table column of the "user_table" table.
	name	Folder name.
	local_path	Local directory path.
	cloud_path	Server's directory path.
	folder_id	A unique numeric folder ID for the sync folder(s).
	created	Folder creation date in datetime format
	last_run	Last sync time in datetime format.
	inactivated	Folder has been inactivated; "true" if yes, "false" if no.
	encrypted	Folder has been encrypted; "true" if yes, "false" if no.
Syncfolder_folder_table	name	Folder name which correlates with the "name" table column of the "syncfolder_table" table.
	root_folder_id	Folder ID for the root sync folder, which correlates with the "folder_id" table column of the "syncfolder_table" table.
	folder_id	Folder ID for the sync folder(s), including the folder ID for the subfolder(s).
	child_folder_id	A unique numeric folder ID for the subfolder(s) associated with the sync folder(s). The root folder retains its original folder ID unchanged.
	creation_date	Folder creation time in datetime format.
	deleted	Folder has been deleted; *NULL* if not deleted.
	owner	Owner's ID for the sync folder(s), which correlates with the "user_ID" table column of "user_table" table.
syncfolder_document_table	owner	Owner's ID for the sync folder(s), which correlates with the "user_ID" table column of "user_table" table.
	name	Folder name.
	root_folder_id	Folder ID for the root sync folder.
	folder_id	Folder ID for the sync folder(s), including the folder ID for the subfolder(s), which correlates with the "child_folder_ID" table column of the "syncfolder_folder_table" table.
	document_id	A unique numeric document ID for the sync file(s).
	size	File size.
	modified_date	Last modified date in datetime format.
	checksum	MD5 checksum for the modified document.
	main_checksum	MD5 checksum for the original document.

favorites/112112/webshare/UbuntuSubFolder/UbuntuSubFolder/Enron3111.docx - server replied: Not Found" Error number: 203," "2016-03-15 14:56:30: onSyncRequestFailed: "WindowsSubFolder/WindowsSubFolder/Enron3111.pdf" | Type: "Uploading" | Error: "7"," "2016-03-15 14:56:30: SYNC_FILE_NOT_FOUND—SYNC_FOLDER_NOT_FOUND: (0) "WindowsSubFolder/WindowsSubFolder/Enron3111.pdf" :" and "2016-03-15

```
SELECT
d.username AS 'Owner Name',
a.folder_id AS 'Sync Folder ID',
a.document_id AS 'Sync File ID',
a.name AS 'Sync File Name',
c.local_path AS 'Sync Folder Path',
a.size AS 'File Size',
a.modified_date AS 'Sync File Last Modified Date',
c.created AS 'Folder Creation Time',
c.last_run AS 'Folder Last Sync Time',
b.deleted AS 'Folder is Deleted',
c.inactivated AS 'Folder is inactivated',
c.encrypted AS 'Folder is encrypted'
FROM syncfolder_document_table a
INNER JOIN syncfolder_folder_table b ON a.folder_id=b.child_folder_id
INNER JOIN  syncfolder_table c ON c.folder_id=a.root_folder_id
INNER JOIN user_table d ON d.user_id=a.owner;
```

FIGURE 1 The SQL query used to parse the synchronization history from Cache.db

	Owner Name	Sync Folder ID	Sync File ID	Sync File Name	Sync Folder Path	File Size	ync File Last Modified Dat	Folder Creation Time	Folder Last Sync Time	Folder is Deleted	Folder is inactivated	Folder is encrypted
24	adamthomson	56295856959613 6	4457417804	Enron3111.jpg	C:/Users/anonymous/ Documents/MacSyncFolder	287937	2016-03-16 12:25:07	2016-03-15 22:06:55	2016-03-16 04:41:40	NULL	false	false
25	adamthomson	56295856959613 6	4457417805	Enron3111.pdf	C:/Users/anonymous/ Documents/MacSyncFolder	31747	2016-03-16 12:25:10	2016-03-15 22:06:55	2016-03-16 04:41:40	NULL	false	false
26	adamthomson	56295856959613 6	4457417806	Enron3111.rtf	C:/Users/anonymous/ Documents/MacSyncFolder	43360	2016-03-16 12:25:13	2016-03-15 22:06:55	2016-03-16 04:41:40	NULL	false	false
27	adamthomson	56295856959613 6	4457417807	Enron3111.txt	C:/Users/anonymous/ Documents/MacSyncFolder	2734	2016-03-16 12:25:13	2016-03-15 22:06:55	2016-03-16 04:41:40	NULL	false	false
28	adamthomson	56295856959613 6	4457417808	Enron3111.zip	C:/Users/anonymous/ Documents/MacSyncFolder	30967	2016-03-16 12:25:20	2016-03-15 22:06:55	2016-03-16 04:41:40	NULL	false	false

FIGURE 2 An excerpt of the output of the SQLite query from Cache.db

14:51:52: addRemoveLocalFolder:Fail: "/home/suspectpc/UbuntuSyncFolder/UbuntuSubFolder"". We could also recover the login time alongside the logged in username from the log entry "*2016-03-15 13:48:22: Logged in as: "adamthomson"*".

4.4 | Web browser artefacts

Web browsing activities history is a critical source of evidence.[29,31-33,53] Our analysis of the web browsing history found unique identifying URLs associated with the user actions. For example, when accessing a sync folder in the CloudMe web application, we observed following URLs:

- *https://www.cloudme.com/en#files:/Documents/<Folder name>*,

https://www.cloudme.com/en#files:/f:<Folder ID>,

- *https://www.cloudme.com/en#sync:/f: <Folder ID>*,

https://www.cloudme.com/en#sync:/<Folder ID>, and

- *https://www.cloudme.com/en#sync:/f:<Folder ID>, <Folder name>*.

  Accessing or downloaded a sync file produced following URL:

- *https://www.cloudme.com/v1/documents/<Folder ID>/<Document ID>/1/<Filename>*.

  When we accessed the folders shared with other users, we observed the following URL:

- *https://www.cloudme.com/en#webshares:/<Folder name>*.

  Accessing the folder shared by other users produced the URL:

- *https://www.cloudme.com/en#following:/<Folder name>*

  The download URL for the shared file could be discerned from:

- *https://www.cloudme.com/v1/documents/<Folder ID>/<Document ID>/1/<Filename>?dl=<Filename>*.

  The web client's logout action generated from the following URL:

- *https://www.cloudme.com/en?r=1458192365602&logout=1*.

Rebuilding the web browsing caches produced the root directory for the web application at *www.cloudme.com/v1*. In particular, within the */%v1%/folders* directory, we recovered a list of metadata files for the sync folders accessed by the user, which be differentiated by the folder IDs. Figure 3 illustrates that the files maintained the folder trees associated with the sync folders; each folder creates a "folder" property to house the

```
<fs:folder id='562958569591836' xmlns:xlink='http://www.w3.org/1999/xlink' xmlns:fs='http://xcerion.com/folders.xsd'>
    <fs:folder id='562958569603280' name='cloudme investigation'>
        <fs:tag type='update' value='718585' group='webshare' propagated='true' />
    </fs:folder>
    ...
</fs:folder>
```

FIGURE 3 An excerpt of the metadata file for a CloudMe sync folder

folder ID and name, and a "tag" subtag to store the folder sharing information such as the webshare ID and folder sharing type, i.e., in the "value" and "group" attributes.

A search for the filenames of the sample files recovered files viewed on the web application in cache at */%v1%/documents/<Folder ID>/<Document ID>/1/*. We also recovered thumbnails for the viewed files in */%v1%/documents/<Folder ID>/<Document ID>/<Thumbnail ID>*. Notice that the */%v1%/documents* directory will always contain at least one folder, i.e., holding the metadata files associated with the sync devices at */%v1%/documents/<Folder ID>/<Document ID for device-specific metadata file>\1*. Figure 4 shows the recovered device name and client ID from the "dName" and "clientId" attributes of the "sync" property in the metadata file. Each sync folder creates a "syncfolder" subtag to define the folder name, directory path, folder ID, last sync time, and information about whether the sync folder has been synchronized and if it is a favourite folder in the "name," "path," "folderId," "lastSync," "hasSynchronized," and "favoriteFolder" attributes, respectively.

Another directory of interest with the "v1" directory is the user-specific */%v1%/users/<User ID>* directory, which maintains a list of OpenSearch[71,78] description documents containing a wealth of folder metadata of forensic interest about the sync folders.[72] For example, the */%v1%/users/<User ID>/favorites/extended=true&order=favoritename&count=1000&offset=0&_=1458191.xml* document holds the OpenSearch description for the favourite folders. The metadata of interest recovered from this document include the folder IDs, folder names, folder sharing passwords, webshare IDs, and usernames and user IDs for the favourite folders in the "folder_id," name, "password," "webShareId," "sharingUserId," "sharingUserName" attributes of the sync folder/file-specific "favorite" subtags of the "favorites" property (see Figure 5). The */%v1%/users/<User ID>/webshares/order=name&desc=false&count=1000&offset=0&resources=true&_=145.xml* document defines the OpenSearch properties of the shared folders/files, such as the update time, creation time, passwords, creators' IDs, webshare IDs in the "updated," "created," password, "userId," and "id" attributes of the sync folder/file-specific "webshare" subtags of the "webshares" property, while the folder name and ID could be discerned from the "name" and "id" properties of the "folder" subtag (see Figure 6). Further details of the folder/file sharing could be located in the */%v1%/users/<User ID>/lifestream* document, such as the senders' user ID, senders' group ID, senders' username, receivers' user ID, receivers' group ID, receiver's username, favourite IDs (for favourite folders), and whether the sharing has been seen in the "senderId," "senderGroupId," "senderName," "receiverId," "receiverGroupId," "receiverName," "parentFolder," and "seen" attributes in the "event" subtags of the "lifestream" property.

4.5 | Physical memory analysis

Physical memory analysis is widely recognized for its ability to provide caches of cloud computing usage, which would otherwise be lost without passive monitoring, such as network socket information, encryption keys, and in-memory database.[28,79,80] For all investigated client applications, analysis of the physical memory dumps using the "pslist" function of Volatility recovered the process name, process identifier, parent process identifiers, and process initiation time, consistent with the findings reported in Yang et al.[16] The CloudMe process could be differentiated using the process names "CloudMe.exe," "cloudme-sync" and "CloudMe" on the Windows, Ubuntu, and Mac OS clients, respectively.

Undertaking data carving of the memory image of the CloudMe process determined that the files of forensic interest such as Sync.config, CloudMe logs, and cache.db could be recovered intact. A search for terms such as "startup.me" and "xClientId" could enable future identification

FIGURE 4 An excerpt of the metadata file for a CloudMe device

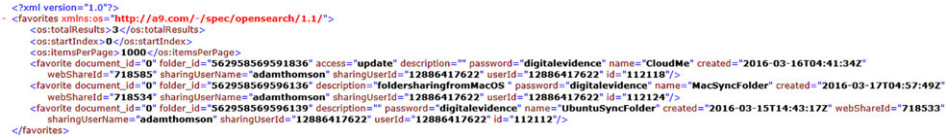

FIGURE 5 The content of the *extended=true&order=favoritename&count=1000&offset=0&_=1458191.xml* document

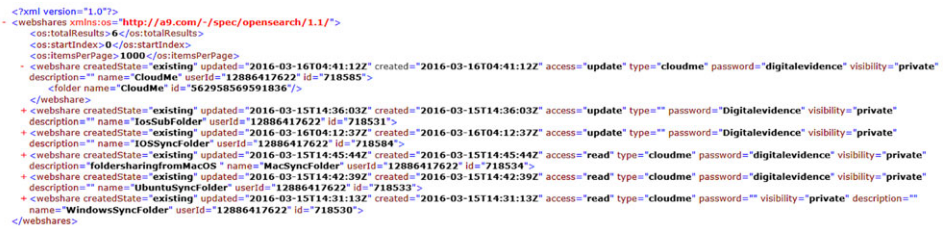

FIGURE 6 The content of the *order=name&desc=false&count=1000&offset=0&resources=true&_=145.xml* document

of the Sync.config file, while the terms "error downloading" and "Type: "Uploading" could be used to correlate the log entries of interest. For the cache database, a search for the username for the user located the data[73] of the "user_table," which holds the user ID in the row ID variant field of the cell header section[73] in hex format. Once the user ID is identified, a practitioner may locate the file offsets contained between the cell data section of the "syncfolder_document_table," "syncfolder_folder_table," and "syncfolder_table" tables and work backwards to read the header field type varints[73] to recover the remaining data fields. When CloudMe was accessed using the web client, we recovered copies of the OpenSearch description documents containing the folder sharing passwords from the web browser's memory space unencrypted. The password could be easily located using the keyword "password=."

5 | ANALYSIS OF CLOUDME MOBILE CLIENTS

The pervasive use of smartphone devices caused a demand for mobile forensics.[81-84] In the context of cloud forensics, mobile devices can provide large amounts of relevant information such as copies of downloaded files, unencrypted credential information, sync logs, databases of file synchronization caches, and user configuration files.[27,28,85] Our examinations of the CloudMe mobile clients determined that the data directory is located in */private/var/mobile/Applications/<Universally Unique Identifier (UUID) for the CloudMe iOS app>/* and */data/data/com.excerion.android* on the iOS and Android clients. Although the mobile clients did not keep a copy of the sync folders from the user's account (like the desktop clients), it was possible to recover copies of the viewed files from *%<Universally Unique Identifier (UUID) for the CloudMe iOS app>%/Documents/persistentCache/* and */storage/sdcard0/Android/data/com.xcerion.android/cache/files/Downloads/* of the iOS and Android clients by default.

5.1 | com.xcerion.icloud.iphone.plist and user_data.xml files

A closer examination of the files in the directory listings located the username and password in plaintext in the "username" and "password" properties of the *%<Universally Unique Identifier (UUID) for the CloudMe iOS app>%/Library/Preferences/com.xcerion.icloud.iphone.plist* and *%com.excerion.android%/shared_prefs/user_data.xml* files. The former also held the last upload time in datetime format in the "<username_LastUploadTime>" property.

5.2 | db.sdb database

Analysis of the Android client revealed the cache database at */storage/sdcard0/Android/data/com.xcerion.android/cache/db.sdb*. The tables of interest with the cache database are "files" and "folders". The "files" table maintains a list of metadata of the sync files viewed by the user, while the "folders" table holds the metadata of the sync folders associated with the user's account. Table 3 shows the table fields of interest from the db.sdb database. We also proposed using a SQL query to thread the table fields of interest from the tables to present the records in a forensically friendly format as shown in Figure 7.

5.3 | Cache.db database

Further examination of the iOS client recovered copies of the responses for the web API queries in the *%<Universally Unique Identifier (UUID) for the CloudMe iOS app>%/Library/Caches/com.xcerion.icloud.iphone/nsurlcache/Cache.db* database. Specifically, we located the cached items in the

TABLE 3 Table fields of forensic interest from the db.sdb database

Table	Table Column	Relevance
files	_id	A unique numerical user ID used to identify a CloudMe sync file.
	name	Filename for the sync file.
	folder_id	Folder ID for the folder housing the sync file.
	size	File size for the sync file.
	href	URL to the sync file.
	published	Sync file addition time in datetime format.
	updated	Last updated time of sync file in datetime format.
	owner	Owner's name of the sync file.
	Mime	Multipurpose Internet Mail Extensions (MIME) format of the sync file.
folders	Owner	Owner's name of the sync folder.
	Folder_id	A unique numerical user ID used to identify a CloudMe sync folder.
	Name	Folder's name.
	Parent	Folder's name for the parent folder.
	Is_root	Whether the sync folder is a root folder?
	Path	Original directory path for the sync folder.

```
SELECT
a.owner AS 'Owner',
a.name AS 'Filename',
a.size AS 'File Size',
b.name AS 'Folder Name',
a.href AS 'URL',
a.published AS 'Published Time',
a.updated AS 'Last Updated Time',
a.mime AS 'File Type',
b.path AS 'Origin'
FROM files a
INNER JOIN folders b ON a.folder_id=b.folder_id
```

	Owner	Filename	File Size	Folder Name	URL	Published Time	Last Updated Time	File Type	Origin
1	adamthomson	Enron3111.jpg	689402	AndroidSyncFolder	https://os.cloudme.com/v1/documents/56295856959614 5/4457368187/1	2016-03-15T14:28:27Z	2016-03-15T14:28:35Z	image/jpeg	xios://Documents/CloudMe/AndroidSyncFolder/
2	adamthomson	Enron3111.docx	78080	AndroidSyncFolder	https://os.cloudme.com/v1/documents/56295856959614 5/4457368325/1	2016-03-15T14:29:24Z	2016-03-15T14:29:24Z	application/vnd.openxmlformats-officed...	xios://Documents/CloudMe/AndroidSyncFolder/
3	adamthomson	cloudme investigation.zip	8939743	cloudme investigation	https://os.cloudme.com/v1/documents/56295856960328 0/4457426501/1	2016-03-16T11:53:52Z	2016-03-16T11:53:52Z	application/zip	xios://Documents/CloudMe/cloudme investigation/

FIGURE 7 The SQL query used to parse the file view history from the db.sdb database and the output

TABLE 4 Locations of files of forensic interest from CloudMe

Content	Directory Paths
Database	• Cache.db database in *%AppData%\Local\CloudMe*, */home/<User Profile>/.local/share/CloudMe/*, */Users/<User Profile>/Library/Application Support/CloudMe/*, and *%<Universally Unique Identifier (UUID) for the CloudMe iOS app>%/Library/Caches/com.xcerion.icloud.iphone/nsurlcache/*of the Windows, Ubuntu, Mac OS, and iOS clients. • */storage/sdcard0/Android/data/com.xcerion.android/cache/db.sdb* on the Android client.
Log files	• [Year-Month-Day].txt in *%AppData%\Local\CloudMe\logs*, */home/<User Profile>/.local/share/CloudMe/logs/*, */Users/<User Profile>/Library/Application Support/CloudMe/logs/*, and of the Windows, Ubuntu, and Mac OS clients.
Default download directory	• *%Users%\[User Profile]\Documents*, */home/[User Profile]/Documents/*, */User/[User Profile]/Documents/*, *%<Universally Unique Identifier (UUID) for the CloudMe iOS app>%/Documents\persistentCache*, and */storage/sdcard0/Android/data/com.xcerion.android/cache/files/Downloads/* on the Windows, Ubuntu, Mac OS, iOS, and Android clients.
Configuration files	• *HKEY_USERS\<SID>\Software\CloudMe registry key*, */home/<User Profile>/.config/CloudMe/Sync.conf*, */Users/<User Profile>/Library/Preferences/com.CloudMe.Sync.plist*, *%<Universally Unique Identifier (UUID) for the CloudMe iOS app>%/Library/Preferences/com.xcerion.icloud.iphone.plist*, and *%com.excerion.android%/shared_prefs/user_data.xml* files on the Windows, Ubuntu, Mac OS, iOS, and Android clients.
Web caches	• www.cloudme.com/v1 directory of the web application.

"receiver_data" table column of the cfurl_cache_receiver_data table in Binary Large OBject (BLOB) including metadata files and OpenSearch documents for the sync folders. Within the cfurl_cache_response table, we located the corresponding URLs and timestamps in datetime format, in the "request_key" and "time_stamp" table columns, respectively. By threading the data fields using the SQL query "*SELECT cfurl_cache_receiver_data.receiver_data, cfurl_cache_response.request_key, cfurl_cache_response.time_stamp FROM cfurl_cache_receiver_data, cfurl_cache_response WHERE cfurl_cache_receiver_data.entry_ID=cfurl_cache_response.entry_ID*", it was possible to correlate the cached items with the URLs and timestamps.

6 | CONCLUSION AND FUTURE WORK

In this paper, we studied the types and locations of CloudMe residual artefacts on desktop and mobile client devices running Windows 8.1, Ubuntu 14.04.1 LTS, Mac OS X Mavericks 10.9.5, iOS 7.1.2, and Android KitKat 4.4.4. Our research included installing the client applications as well as uploading, downloading, deleting, sharing, and activating/inactivating the sync folders/files using the client and web applications.

Our findings suggested that a forensic practitioner investigating CloudMe cloud application should pay attention to the cache database, web caches, and log and configuration files, as highlighted in Table 4. Unlike the findings of our previous forensic analysis of BitTorrent Sync[27] and Symform,[29] we determined that CloudMe client application did not create any identifying information (e.g., configuration file and cache folder) in the sync folders. Hence, a practitioner is unlikely to be able to identify the sync directories from the directory listing. This also suggests that the cache database is a critical source of evidence for the synchronization metadata and cloud transaction records and should not be overlooked.

Analysis of the mobile clients determined that the findings were not as conclusive as compared with the desktop clients, as only the viewed files could be forensically recovered. This indicated that the iOS and Android mobile clients are merely a user interface for the web application. Our examination of the web browsing activities identified unique URLs that facilitate the identification of user actions on the web application, such as login, logout, and accessing and downloading sync files/folders. Although the application layer was fully encrypted (using HTTPS), we were able to recover the root directory for the web application from the web browser's caches unencrypted. This includes viewed files and metadata files and OpenSearch documents for the sync files/folders that contain the timestamp information and sharing passwords for the sync folders/files.

However, a practitioner should note that the availability of the cached items depends on the API requests made to the web application; hence, the artefacts may not be consistent on different occasions.

Our analysis of the physical memory captures revealed that the memory dumps may provide the ability to link artefacts from the static data analysis, such as applications cache, logs, configuration files, and other files of forensic interest. The artefacts also include the folder sharing password from the web cache in plaintext, but not the login password. This suggested that a practitioner can only obtain the login password from the mobile clients, using WebBrowserPassView when manually saved in the web browsers, through an offline brute-force technique, or directly from the user. As we had previous noted, data in physical memory may be overwritten on low memory and system's shut down.[16] Therefore, obtaining the memory snapshot as quickly as possible increases the likelihood of preserving the artefacts.

Our planned research in the future includes extending this study to other popular and contemporary cloud storage services. This will allow us to contribute to a better understanding, in terms of breadth and depth, of the big data artefacts that could be forensically recovered from different cloud deployment models. These findings will also lay the foundation for the development of data reduction techniques (e.g., data mining and intelligence analysis) for these technologies.[86,87]

ORCID

Kim-Kwang Raymond Choo http://orcid.org/0000-0001-9208-5336

REFERENCES

1. U.S. Department of Justice and Federal Bureau of Investigation, Regional computer forensics laboratory annual report for fiscal year 2014. U.S. Department of Justice, 2014.
2. Quick D, Choo K-KR. Data reduction and data mining framework for digital forensic evidence: Storage, intelligence, review, and archive. *Trends Issues Crime Crim Justice*. 2014;480:1-11.
3. Damshenas M, Dehghantanha A, Mahmoud R, bin Shamsuddin S. Forensics investigation challenges in cloud computing environments. In *Proceedings of 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, 2012;190-194.
4. Watson S, Dehghantanha A. Digital forensics: The missing piece of the internet of things promise. *Comput Fraud Secur*. 2016;2016(6):5-8.
5. Daryabar F, Dehghantanha A, Norouzi F, Mahmoodi F. Analysis of virtual honeynet and VLAN-based virtual networks. In *Proceedings of 2011 International Symposium on Humanities, Science and Engineering Research*, 2011;73-77.
6. Choo K-KR. Cloud computing: challenges and future directions. *Trends Issues Crime Crim Justice*. 2010;400:1-6.
7. Martini B, Choo K-KR. Cloud forensic technical challenges and solutions: A snapshot. *IEEE Cloud Comput*, 2014;1(4):20-25.
8. Choo K-KR. Organised crime groups in cyberspace: A typology. *Trends Organ Crime*. 2008;11(3):270-295.
9. Daryabar F, Dehghantanha A. A review on impacts of cloud computing and digital forensics. *Int J Cyber-Secur Digit Forensics IJCSDF*. 2014;3(4):183-199.
10. Daryabar F, Dehghantanha A, Udzir NI, Sani NF, Binti M, bin Shamsuddin S. A review on impacts of cloud computing and digital forensics. *Int J Cyber-Secur Digit Forensics IJCSDF*. 2013;2(2):77-94.
11. Damshenas M, Dehghantanha A, Mahmoud R. A survey on digital forensics trends. *Int J Cyber-Secur Digit Forensics*. 2014;3(4):209-235.
12. Nepal S, Ranjan R, Choo KKR. Trustworthy processing of healthcare big data in hybrid clouds. *IEEE Cloud Comput*. 2015;2(2):78-84.
13. Ab Rahman NH, Choo K-KR. A survey of information security incident handling in the cloud. *Comput Secur*. 2015;49(C):45-59.
14. Dehghantanha A, Franke K. Privacy-respecting digital investigation. In *Proceedings of 2014 Twelfth Annual International Conference on Privacy, Security and Trust (PST)*. 2014;129-138.
15. Dezfouli FN, Dehghantanha A, Eterovic-Soric B, Choo K-KR. Investigating social networking applications on smartphones detecting Facebook, Twitter, LinkedIn and Google+ artefacts on Android and iOS platforms. *Aust J Forensic Sci*. 2016;48(4):469-488.
16. Yang TY, Dehghantanha A, Choo K-KR, Muda Z. Windows instant messaging app forensics: Facebook and Skype as case studies. *PLoS One*. 2016;11(3):e0150300.
17. Quick D, Choo K-KR. Big forensic data reduction: Digital forensic images and electronic evidence. *Clust Comput*. 2016;19(2):1-18.
18. Thethi N, Keane A. Digital forensics investigations in the cloud. In: *Proceedings of 2014 IEEE International Advance Computing Conference (IACC)*. 2014;1475-1480.
19. Dykstra J, Sherman AT. Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust, and techniques. *Digit Investig*. 2012;9(Supplement):S90-S98.
20. Martini B, Do Q, Choo K-KR. Chapter 15 - Mobile Cloud Forensics: An Analysis of Seven Popular Android Apps. In: *The Cloud Security Ecosystem*. Boston: Syngress; 2015:309-345.
21. Najvadi Y, Dehghantanha A. Forensic investigation of social media and instant messaging services in Firefox OS: Facebook, Twitter, Google+, Telegram, OpenWapp, and Line as case studies. In: *Contemporary Digital Forensic Investigations Of Cloud And Mobile Applications*. Elsevier; 2016.
22. Mohd Najwadi Y, Dehghantanha A. Network Traffic Forensics on Firefox Mobile Os: Facebook, Twitter and Telegram as Case Studies. In: *Contemporary Digital Forensic Investigations of Cloud and Mobile Applications*. Elsevier; 2016.
23. Quick D, Martini B, Choo R. *Cloud Storage Forensics*. 1st ed. Syngress; 2013.
24. Hooper C, Martini B, Choo K-KR. Cloud computing and its implications for cybercrime investigations in Australia. *Comput Law Secur Rev*. 2013;29(2):152-163.
25. National Institute of Standards and Technology (NIST). *NIST Cloud Computing Forensic Science Challenges*. National Institute of Standards and Technology; 2014.

26. Yee Yang T, Dehghantanha A, Zaiton M. Investigating America Online instant messaging application : Data remnants on Windows 8.1 client machine. In: Choo K-KR, Dehghantanha A, eds. *Contemporary Digital Forensic Investigations Of Cloud And Mobile Applications*. Elsevier; 2016.
27. Teing Y-Y, Dehghantanha A, Choo K-KR, Yang LT. Forensic investigation of P2P cloud storage services and backbone for IoT networks: Bittorrent sync as a case study. *Comput Electr Eng*. 2017;58:350-363.
28. Teing Y-Y, Dehghantanha A, Choo K-KR, Muda Z, Abdullah MT, Chai W-C. A Closer Look at Syncany Windows and Ubuntu Clients′ Residual Artefacts. In: Wang G, Ray I, Calero JMA, Thampi SM, eds. *Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer International Publishing; 2016:342-357.
29. Teing Y-Y, Dehghantanha A, Choo K-KR, Dargahi T, Conti M. Forensic investigation of cooperative storage cloud service: Symform as a case study. *J Forensic Sci*. 2017;62(3):641-654.
30. Shariati M, Dehghantanha A, Martini B, Choo K-KR. Chapter 19 - Ubuntu One Investigation: Detecting Evidences on Client Machines. In: *The Cloud Security Ecosystem*. Boston: Syngress; 2015:429-446.
31. Quick D, Choo K-KR. Google drive: Forensic analysis of data remnants. *J Netw Comput Appl*. 2014;40:179-193.
32. Quick D, Choo K-KR. Digital droplets: Microsoft SkyDrive forensic data remnants. *Future Gener Comput Syst*. 2013;29(6):1378-1394.
33. Quick D, Choo K-KR. Dropbox analysis: Data remnants on user machines. *Digit Investig*. 2013;10(1):3-18.
34. CloudMe AB. Interview with founder and CEO. 2016. https://www.cloudme.com/newsletter/2015-01_en.html; Accessed 26-May-2016
35. CloudMe AB. Pricing. 2016. https://www.cloudme.com/en/pricing; Accessed 26-May-2016
36. CloudMe AB. Tutorials on CloudMe. 2016. https://www.cloudme.com/en/tutorials; Accessed 26-May-2016
37. Quick D, Choo K-KR. Forensic collection of cloud storage data: Does the act of collection result in changes to the data or its metadata? *Digit Investig*. 2013;10(3):266-277.
38. Mell P, Grance T. The NIST definition of cloud computing. 2011.
39. Choo K-KR, Dehghantanha A. *Contemporary Digital Forensic Investigations of Cloud and Mobile Applications*. Syngress; 2016.
40. Ruan K, Carthy J, Kechadi T, Baggili I. Cloud forensics definitions and critical criteria for cloud forensic capability: An overview of survey results. *Digit Investig*. 2013;10(1):34-43.
41. Ruan K, Baggili I, Carthy J, Kechadi T. *Survey on Cloud Forensics and Critical Criteria for Cloud Forensic Capability: A Preliminary Analysis*, Electr. Comput. Eng. Comput. Sci. Fac. Publ., 2011.
42. Simou S, Kalloniatis C, Kavakli E, Gritzalis S. Cloud Forensics: Identifying the Major Issues and Challenges. In: *Advanced Information Systems Engineering*. Springer International Publishing; 2014:271-284.
43. Pichan A, Lazarescu M, Soh ST. Cloud forensics: Technical challenges, solutions and comparative analysis. *Digit Investig*. 2015;13:38-57.
44. Birk D, Wegener C. Technical issues of forensic investigations in cloud computing environments. In: *2011 IEEE Sixth International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE)*; 2011;1-10.
45. Sibiya G, Venter HS, Fogwill T. Digital forensics in the cloud: The state of the art, in *Proceedings of IST-Africa Conference, 2015*, 2015;1-9.
46. Fahdi MA, Clarke NL, Furnell SM. Challenges to digital forensics: A survey of researchers amp; practitioners attitudes and opinions, in *2013 Information Security for South Africa*, 2013;1-8.
47. Taylor M, Haggerty J, Gresty D, Almond P, Berry T. Forensic investigation of social networking applications. *Netw Secur*. 2014;2014(11):9-16.
48. Wilkinson S. ACPO good practice guide for digital evidence. 7Safe, 2012.
49. Kent K, Chevalier S, Grance T. Guide to integrating forensic techniques into incident. 2006.
50. Farina J, Scanlon M, Le-Khac NA, Kechadi MT. Overview of the forensic investigation of cloud services, in *2015 10th International Conference on Availability, Reliability and Security (ARES)*, 2015;556-565.
51. Martini B, Choo K-KR. An integrated conceptual digital forensic framework for cloud computing. *Digit Investig*. 2012;9(2):71-80.
52. McKemmish R. What is forensic computing. Australian Institute of Criminology, 1999.
53. Martini B, Choo K-KR. Cloud storage forensics: Owncloud as a case study. *Digit Investig*. 2013;10(4):287-299.
54. Martini B, Choo K-KR. Remote programmatic vCloud forensics: A six-step collection process and a proof of concept, in *Proceedings of 13th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2014)*, 2014;935-942.
55. Martini B, Choo K-KR. Distributed filesystem forensics: Xtreemfs as a case study. *Digit Investig*. 2014;11(4):295-313.
56. Chung H, Park J, Lee S, Kang C. Digital forensic investigation of cloud storage services. *Digit Investig*. 2012;9(2):81-95.
57. Scanlon M, Farina J, Kechadi M-T. BitTorrent sync: Network investigation methodology, in *Proceedings of 2014 9th International Conference on Availability, Reliability and Security*, 2014;21-29.
58. Scanlon M, Farina J, Khac NAL, Kechadi T. Leveraging decentralization to extend the digital evidence acquisition window: Case study on BitTorrent sync, *ArXiv14098486 Cs*, 2014;1-14.
59. Do Q, Martini B, Choo K-KR. A forensically sound adversary model for mobile devices. *PLoS One*. 2015;10(9):e0138449.
60. Do Q, Martini B, Choo K-KR. Is the data on your wearable device secure? An android wear smartwatch case study. *Softw Pract Exp*. 2017;47(3):391-403.
61. Ab Rahman NH, Cahyani NDW, Choo K-KR. Cloud Incident Handling and Forensic-By-Design: Cloud Storage as a Case Study. *Concurr Comput Pract Exp*. 2017;29(14):e3868.
62. Marty R. Cloud application logging for forensics. In: *Proceedings of the 2011 ACM Symposium on Applied Computing*. New York, NY, USA; 2011;178-184.
63. Shields C, Frieder O, Maloof M. A system for the proactive, continuous, and efficient collection of digital forensic evidence. *Digit Investig*. 2011;8(Supplement):S3-S13.
64. Zawoad S, Hasan R. Cloud forensics: A meta-study of challenges, approaches, and open problems. *ArXiv13026312 Cs*, 2013.

65. Dykstra J, Sherman AT. Design and implementation of frost: Digital forensic tools for the Openstack cloud computing platform. *Digit Investig*. 2013;10: S87-S95.
66. Gebhardt T, Reiser HP. Network Forensics for Cloud Computing. In: Dowling J, Taïani F, eds. *Distributed Applications and Interoperable Systems*. Springer: Berlin Heidelberg; 2013:29-42.
67. Hale JS. Amazon cloud drive forensic analysis. *Digit Investig*. 2013;10(3):259-265.
68. Farina J, Scanlon M, Kechadi M-T. BitTorrent sync: First impressions and digital forensic implications. *Digit Investig*. 2014;11(Supplement 1):S77-S86.
69. Shariati M, Dehghantanha A, Choo K-KR. SugarSync forensic analysis. *Aust J Forensic Sci*. 2016;48(1):95-117.
70. Blakeley B, Cooney C, Dehghantanha A, Aspin R. Cloud storage forensic: HubiC as a case-study, in *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2015;536-541.
71. Daryabar F, Dehghantanha A, Choo K-KR. Cloud storage forensics: MEGA as a case study. *Aust J Forensic Sci*. 2016;1-14.
72. Mohtasebi H, Dehghantanha A, Choo R. Cloud storage forensics: Analysis of data remnants on SpiderOak, JustCloud, and pCloud. In: *Contemporary Digital Forensic Investigations Of Cloud And Mobile Applications*, Elsevier; 2016.
73. Daryabar F, Dehghantanha A, Eterovic-Soric B, Choo K-KR. Forensic investigation of Onedrive, Box, Google Drive and Dropbox applications on Android and iOS devices. *Aust J Forensic Sci*. 2016;48(6):615-642.
74. Dehghantanha A, Dargahi T. Chapter 14 - Residual Cloud Forensics: CloudMe and 360Yunpan as Case Studies. In: *Contemporary Digital Forensic Investigations of Cloud and Mobile Applications*. Syngress; 2017:247-283.
75. Amirullah A, Riadi I, Luthfi A. Forensics analysis from cloud storage client application on proprietary operating system. *Int J Comput Appl*. 2016;143(1): 1-7.
76. Cahyani NDW, Martini B, Choo K-KR, Al-Azhar AMN. Forensic Data Acquisition from Cloud-of-Things Devices: Windows Smartphones as a Case Study. *Concurr Comput Pract Exp*. 2017;29(14):e3855.
77. SQLite, Database file format, 2016. https://www.sqlite.org/fileformat.html. Accessed: 10-Nov-2016.
78. A9.com Inc., OpenSearch 1.1 draft 5, 2016. http://www.opensearch.org/Specifications/OpenSearch/1.1#The_.22OpenSearchDescription.22_element. Accessed 26-May-2016.
79. Dezfouli FN, Dehghantanha A, Mahmoud R, Sani NFBM, bin Shamsuddin S. Volatile memory acquisition using backup for forensic investigation. In: *Proceedings of 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, 2012;186-189.
80. Canlar ES, Conti M, Crispo B, Di Pietro R. Windows mobile LiveSD forensics. *J Netw Comput Appl*. 2013;36(2):677-684.
81. Damshenas M, Dehghantanha A, Choo K-KR, Mahmud R. M0Droid: an android behavioral-based malware detection model. *J Inf Priv Secur*. 2015;11(3):141-157.
82. Tassone C, Martini B, Choo K-KR, Slay J. Mobile device forensics: A snapshot. *Trends Issues Crime Crim Justice*. 2013;(460):1-7.
83. Azfar A, Choo K-KR, Liu L. An Android Communication App Forensic Taxonomy. *J Forensic Sci*. 2016;61(5):1337-1350.
84. Azfar A, Choo K-KR, Liu L. Android mobile Voip apps: A survey and examination of their security and privacy. *Electron Commer Res*. 2016;16(1):73-111.
85. Teing YY, Ali D, Choo K, Abdullah MT, Muda Z. Greening cloud-enabled big data storage forensics: Syncany as a case study. *IEEE Trans Sustain Comput*. 2017;PP(99):1-1.
86. Quick D, Choo K-KR. Impacts of increasing volume of digital forensic data: A survey and future research challenges. *Digit Investig*. 2014;11(4):273-294.
87. Dehghantanha A. Mining the social web: Data mining Facebook, Twitter, LinkedIn, Google+, Github, and more. *J Inf Priv Secur*. 2015;11(2):137-138.

How to cite this article: Teing Y-Y, Dehghantanha A, Choo K-KR. CloudMe forensics: A case of big data forensic investigation. *Concurrency Computat Pract Exper*. 2018;30:e4277. https://doi.org/10.1002/cpe.4277