Greening Cloud-Enabled Big Data Storage Forensics: Syncany as a Case Study

Yee-Yang Teing, Ali Dehghantanha, *Senior Member, IEEE*, Kim-Kwang Raymond Choo, *Senior Member, IEEE*, Mohd Taufik Abdullah, and Zaiton Muda

Abstract—The pervasive nature of cloud-enabled big data storage solutions introduces new challenges in the identification, collection, analysis, preservation and archiving of digital evidences. Investigation of such complex platforms to locate and recover traces of criminal activities is a time-consuming process. Hence, cyber forensics researchers are moving towards streamlining the investigation process by locating and documenting residual artefacts (evidences) of forensic value of users' activities on cloud-enabled big data platforms in order to reduce the investigation time and resources involved in a real-world investigation. In this paper, we seek to determine the data remnants of forensic value from Syncany private cloud storage service, a popular storage engine for big data platforms. We demonstrate the types and the locations of the artefacts that can be forensically recovered. Findings from this research contribute to an in-depth understanding of cloud-enabled big data storage forensics, which can result in reduced time and resources spent in real-world investigations involving Syncany-based cloud platforms.

Index Terms— Green forensics, big data forensics, cloud forensics, Syncany forensics

1 Introduction

Big data and cloud computing are two information and communications technology (ICT) trends in recent years. For example, a 2013 survey by the International Data Corporation (IDC) predicted that the cloud computing spending will exceed USD 107 billion and will drive 17% of the IT product expenditures by 2017 [1]. Gartner [2] also predicted that the cloud computing market will be worth more than USD 1.1 trillion by 2017 as more big data platforms and big data analytical solutions are deployed over cloud storage services.

There are, however, situations where public clouds are not suitable to host the big data platforms (e.g. inadequate privacy protection for data). Public cloud users are often billed for the resources consumed, including the storage input output (I/O), CPUs and memory, and the cost can become prohibitively expensive in storing, accessing and analysing high velocity, variety, veracity, and volume data. There were also concerns about Cloud Service providers (CSPs) being able to infer or profile users based on the hosted/stored data [3],[4],[5]. Thus, it is unsurprising that cloud and big data security and privacy are current research focus [6],[7],[8],[9],[10],[11],[12],[13],[14]. Data owners in industries such as healthcare and banking are also subject to exacting regulatory requirements, which restrict the outsourcing of the data for storage and analysis. The concerns are compounded in the recent leakage of the PRISM and MUSCULAR surveillance programs that allegedly enable government and law enforcement agencies to tap into CSPs' data centres without a search warrant [15]. Therefore, in-house private dedicated cloud storage services have become an ideal solution for big data platforms.

The capability to collect evidence from the cloud has real-world implications for both criminal investigations and civil litigations [16]. For example, due to the nature of cloud-enabled big data storage solutions (e.g. data physically stored in distributed servers), identification of residual evidences may be a 'finding a needle in a haystack' exercise [17],[18],[19],[20]. Even if the data could be located, traditional evidence collection tools, techniques and practices are unlikely to be adequate [21]. 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 [22],[23],[24],[25]. Overcoming these investigation challenges demands significant time and resources of an investigation team [26].

In this paper, we extend our previous work [27][1] seeking to identify potential artefacts that remain on the client devices and cloud servers involving the use of Syncany as a private cloud storage solution supporting big-data platforms. We attempt to answer the following questions:

- *Yee-Yang Teing is with the Department of Computer Science, Faculty of Computer Science and Technology, Universiti Putra Malaysia, Serdang, 43400 Selangor, Malaysia, as well as the School of Computing, Science and Engineering, University of Salford, Salford, Greater Manchester M5 4WT, UK (e-mail: teingyeeyang@gmail.com).*
- *Ali Dehghantanha is with the School of Computing, Science and Engineering, University of Salford, Salford, Greater Manchester M5 4WT, UK (e-mail: A. Dehghantanha@salford.ac.uk).*
- *Kim-Kwang Raymond Choo is with the Department of Information Systems and Cyber Security, University of Texas at San Antonio, San Antonio, TX 78249-0631, USA (e-mail: raymond.choo@fulbrightmail.org).*
- *Zaiton Muda and Mohd Taufik Abdullah are with the Department of Computer Science, Faculty of Computer Science and Technology, Universiti Putra Malaysia, Serdang, 43400 Selangor, Malaysia (e-mail: zaitonm@upm.edu.my and taufik@upm.edu.my).*

[1] This paper is an extended version of an earlier published conference paper [27], with more than 50% new content.

1. What residual artefacts can be recovered from the hard disk and memory after using the Syncany desktop clients, and the locations of the data remnants on a Windows 8.1, Ubuntu 14.04.1 LTS (Ubuntu), and Mac OS X Mavericks 10.9.5 (Mac OS) client device?
2. What data of forensic interest can be recovered from a cloud server hosting Syncany private cloud storage service and the location of the data remnants on an Ubuntu 14.04.1 LTS server?
3. What data of forensic interest can be collected from the network traffic communications between the Syncany clients and servers?

We describe related works and Syncany private cloud service in the next two sections, respectively. Section 4 outlines our research methodology and experiment environment setup. In Sections 5 and 6, we discuss findings from the analysis of Syncany clients and servers, respectively. Section 7 summarises the research findings. Finally, in Section 8, we conclude the paper and outline potential future research areas.

2 Related Work

Due to the underlying legal challenges and complexity involved in cross-jurisdictional cloud forensic investigations, Marty [28] and Shields et al. [29] proposed a proactive application-level logging mechanism to log information of forensic interest which can also be used in incident response. Forensic researchers such as Dykstra and Sherman [30], Gebhardt and Reiser [31], Quick et al. [24], and Martini and Choo [32] have also presented frameworks and prototype implementations to support collection of evidential materials using Application Programming Interfaces (API) from a variety of cloud storage platforms. Although data collection using APIs could potentially reduce interactions with the CSPs, it could be limited by the APIs' feature sets [32].

In our earlier work [33],[34], we studied the integrity of data downloaded from the web and desktop clients of Dropbox, Google Drive, Skydrive, and Symform, and identified that the act of downloading data from client applications does not tamper the contents of the files, despite changes in the file creation/modification timestamps. Other cloud applications such as Evernote [35], Amazon S3 [35], Dropbox [35], [36], Google Drive [35], [37], Microsoft Skydrive [38], Amazon Cloud Drive [39], BitTorrent Sync [40], SugarSync [41], Ubuntu One [42], huBic [43], Syncany [27], Symform [34], as well as mobile cloud apps [45], [46] have also been examined. These forensics studies located artefacts of client applications from the user settings and application data resided on the media storage through keyword search. Quick and Choo [36],[37],[38] also identified that data erasing tools such as Eraser and CCleaner do not completely remove the data remnants from Dropbox, Google Drive, and Microsoft SkyDrive.

The first cloud forensic framework was proposed by Martini and Choo [47], which was derived based upon the frameworks of McKemmish [48] and NIST [49]. The framework was used to investigate ownCloud [50], Amazon EC2 [51], VMWare [32], and XtreemFS [52]. Quick et al. [24] further extended and validated the four-stage framework using SkyDrive, Dropbox, Google Drive, and ownCloud. Chung et al. [35] proposed a methodology for cloud investigation on Windows, Mac OS, iOS, and Android devices. The methodology was then used to investigate Amazon S3, Google Docs, and Evernote. Scanlon et al. [53] outlined a methodology for remote acquisition of evidence from decentralised file synchronisation network which utilised to investigate BitTorrent Sync [40], [54]. In another study, we proposed a methodology for investigating the newer BitTorrent Sync application (version 2.0) or any third party or Original Equipment Manufacturer (OEM) applications [55]. Do et al. proposed an adversary model for digital forensics, and they demonstrated how such an adversary model can be used to investigate mobile devices and apps [56] and Android smartwatches [57]. Ab Rahman et al. [58], on the other hand, proposed a conceptual forensic-by-design framework to integrate forensics tools and best practices in the development of cloud systems. Tzuen et.al. [26], discussed about opportunities and challenges in streamlining digital investigation by referring to research that identified and documented residual artefacts.

The majority of existing published work in the field of cloud forensics have focused on client investigations on public cloud infrastructures. This paper is one of few studies which focus on both client and server forensics. Such a study may assist in reducing the carbon footprints of investigation cases involving private cloud storage solutions.

3 Background

Syncany is a popular open source and cross-platform cloud storage solution written in Java and supports data storage on different backend media (e.g., FTP, Box.net, WebDAV and SFTP, Amazon S3, Google Storage, IMAP, Local, Picasa and Rackspace Cloud Files). Hence, it does not require a server-side software [59]. In Syncany, the data synchronisation is facilitated by the sync link or repo[2] URL. This is similar to the concept of BitTorrent Sync, with the difference being that Syncany uses a central storage infrastructure. Syncany encrypts (by default) the sync files locally using 128-bit AES/GCM encryption algorithm before uploading the sync files to the central (offsite) storage; hence, only clients in possession of the password can access the repositories. The data model consists of the following [59]:

- Versioning: Syncany captures different versions of a file and keeps track of the changes using metadata such as date, time, size and checksums. There are three primary versioning concepts, which are database versions, file histories, and file versions. A database version represents the point in time when the file tree is captured. Each database ver-

[2] In Syncany, a repository (commonly known as a repo) refers to a dumb storage i.e., a folder that stores the packaged data and metadata about a sync folder in the backend system [59].

sion contains a list of file histories, representing the identity of a file. Each file history contains a collection of file versions, representing the incarnations of a file.

- Deduplication/Chunking: Syncany uses data deduplication technique to break individual files into small chunks on the client. The chunks are represented in data blobs (each about 8-32 KB in size), which are identified by their checksums.
- Multichunking: Individual chunks are grouped into multichunks, compressed and encrypted before being uploaded to the offsite storage.

Fig. 1 shows an example of logical data model of a Syncany repository. The entities are stored locally in the form of plain-old-Java-object (POJOs) in the org.syncany.database package and tables are stored in the local HSQLDB-based database in .XML format [60].

Syncany uses a command line interface by default, but the users can manually install a plugin which supports Graphical User Interface (GUI). Syncany commands of forensic relevance are as follows [61]:

- *sy init*: To initialise the repository for a new sync folder. It creates a sync-folder-specific config.xml and repo file. The former holds the local configuration information while the latter contains the chunking/crypto details required to initialise the remote repository. This command also generates the sync link in two formats: a commonly used encrypted link structured such as *syncany://storage/1/<master-salt>/<encrypted-config>*, where both <master-salt> and <encrypted-config> are base58 encoded; a plaintext link structured such as *syncany://storage/1/not-encrypted/<plaintext-config>*, where the <plaintext-config> is a base58-encoded representation of the storage/connection config.
- *sy connect*: To connect to an existing repository using the sync link or manually using the repo URL. This command is similar to *sy init*, with the difference being that it downloads the repo files from a remote storage.
- *sy status*: Lists changes made to the local sync files by comparing the local file tree (e.g., last modified dates and file sizes) with the local database.
- *sy up*: Detects changes in the local sync directory (using the 'sy status' command), indexes new files and uploads changes to the remote repository (using the 'sy up' command). File changes are packaged into new multichunks and uploaded to an offsite storage, alongside the delta metadata database.
- *sy ls-remote*: Queries the remote storage and lists the client database versions that have not yet been downloaded/processed.
- *sy down*: Detects file changes made by other clients (as identified by the 'sy ls-remote' command). The command first downloads the metadata of relevance. Then, it evaluates which multichunks are changed or required. Finally, it downloads and arranges the multichunks according to the vector clocks, if necessary.

4 Research Methodology

Similar to our previous approaches [27], [34], [36],[37],[38], [55], our test environments consist of four (4) VMware Workstations (VMs), one (1) for server and three (3) for desktop clients – see TABLE 1. The VMs were hosted using VMware Fusion Professional version 7.0.0 (2103067) on a Macbook Pro (Late 2012) running Mac OS X Mavericks 10.9.5, with a 2.6GHz Intel Core i7 processor and 16GB of RAM. The 3111th email message of the Berkeley Enron email dataset (downloaded from *http://bailando.sims.berkeley.edu/enron_email.html*) were used to create a set of sample files and saved in .RTF, .TXT, .DOCX, .JPG (print screen), .ZIP, and .PDF formats, providing a basis for replication of the experiments in future. The set of sample files were placed in a new directory on the clients' workstations, before being uploaded to the servers, and subsequently downloaded to the corresponding client devices. It is noteworthy that the Syncany application does not require an ac-

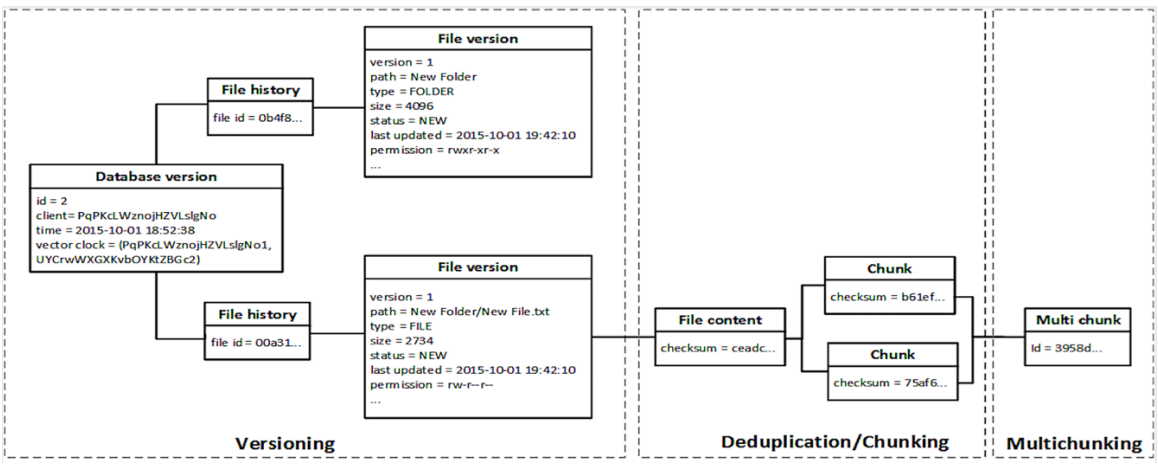

Fig. 1. Logical data model of a Syncany repository (Adapted from [59]).

TABLE 1
SYSTEM CONFIGURATIONS FOR SYNCANY FORENSICS

Server configurations	Client configurations
Ubuntu Server Operating system : Ubuntu 14.04.1 LTS Virtual memory size: 1GB Virtual storage size: 20GB Web application: Apache 2 Storage type: WebDAV Database server: HSQL Database Engine 2.3.0 IP address/URL: http://172.16.38.180/webdav	**Windows Client** Operating system: Windows 8.1 Professional (Service Pack 2, 64-bit, build 9600). Virtual memory size: 2GB Virtual storage size: 20GB Client application: Syncany 0.4.6-alpha Plugins installed: WebDAV and GUI
	Ubuntu Client Operating system: Ubuntu 14.04.1 LTS Virtual memory size: 1GB Virtual storage size: 20GB Client application: Syncany 0.4.6-alpha Plugins installed: WebDAV and GUI
	Mac OS Client Operating system: Mac OS X Mavericks 10.9.5 Virtual memory size: 1GB Virtual storage size: 60GB Client application: Syncany 0.4.6-alpha Plugins installed: WebDAV and GUI

count, and hence no user account was created.

In our experiments, we conducted a predefined set of activities to simulate various real world scenarios of using Syncany private cloud storage service. These included installation and uninstallation of the client applications as well as accessing, uploading, downloading, viewing, deleting, and unsyncing the sync files. A snapshot was undertaken of the VM workstations prior to and after each experiment. Wireshark was deployed on the host machine to capture the communication network traffic. For the purpose of this research, we used WebDAV as the carrying protocol to enable the study of the network's inner workings. Other forensic tools prepared for the experiments are presented in TABLE 2. The experiments were repeated thrice (at different dates) to ensure consistency of findings.

The reliability of digital evidences rests upon the ability of the practitioners to adhere to generally accepted principles, standards, guidelines, procedures, and best practices [49], [62]. In this research, the cloud forensics framework of Martini and Choo [47] was used to guide our investigations:

1. *Evidence source identification and preservation.* We identified the physical hardware of interest, which contained the virtual disk (VMDK) and memory (VMEM) files in each VM folder. Then, we created a forensic copy of the VMDK and VMEM files in E01 container and raw image file (dd) formats respectively. For all the forensic images created, we calculated the MD5 and SHA1 hash values for each original copy and subsequently verified for each working copy.

2. *Collection.* We extracted test data that matched the search terms 'syncany' and 'enron' in the hard disk images. These included SQLite databases, Apple Property List (PLIST) files, event logs, shortcuts, thumbnail caches, and Windows' Prefetch, $MFT, $LogFile and $UsnJrnl files. Data from the physical memory dumps were collected using Volatility, Photorec file carver, and HxD Hex Editor; network traffic were captured using Wireshark and Netminer. Similar to the first stage, for all the extracted files, we calculated the MD5 and SHA1 hash values for each original copy and subsequently verified for each working copy.

3. *Examination and analysis.* This phase is concerned with the assessment and extraction of the evidential information from the collected data. We seek to recover the following artefacts of evidential value from the client devices:

- **Sync and file management metadata –** The property information generated by the Syncany client to facilitate the synchronisation process with the hardware hosting the cloud environment. These include the node/peer IDs, usernames, device names, folder/file IDs, share keys (if any), and other property information that can facilitate file-to-device/file-to-user mappings.
- **Authentication and encryption metadata –** Artefacts describing the encryption algorithm, public/private keys or certificates, encryption salt, initialization vectors (IVs), password hashes, login URL, and (possibly) plaintext credentials that could provide the opportunity to login to the user's online account or undertake offline brute-force attacks to obtain the login password.
- **Cloud transaction logs –** Records of user actions made to the cloud instance, such as installation and uninstallation of client applications, user account creation, logging in and off the web/client application, as well as file upload, download, modification, and deletion. The records are often accompanied by the directory paths and timestamp information which are useful for timeline analysis.
- **Data storage –** Actual storage or temporary holding place (e.g., the application generated sync/archive folders, OS and web browser's thumbnail caches, trash folders, and unallocated space) that may be relied upon for recovering the synced files in case the cloud environment is not accessible or the actual copies of sync files have been deleted.
- **Network captures –** Network captures provide copies of network packets transmitted between the

TABLE 2
TOOLS PREPARED FOR SYNCANY FORENSICS

Tool	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.
Autopsy 3.1.1	To produce directory listings for the forensic images as well as extracting files and analysing 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 and dumping files from the memory space of the Syncany client applications (i.e., using the 'pslist', 'netstat'/'netscan', and 'memdump' functions, respectively).
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.
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 analyse the Windows prefetch and link files.
NTFS Log Tracker	To parse and analyse the $LogFile, $MFT, and $UsnJrnl New Technology File System (NTFS) files.
SQL Workbench/J Build 118.	To view the contents of Hyper SQL Database (HSQLDB).

client and the hardware hosting the cloud environment, which may contain digital fingerprints of cloud transaction events including the logical/physical locations of the correspondents/cloud hosting environment, timestamps of file synchronisation processes, and other network signatures. Network traffic analysis is particularly pertinent when the network traffic are unencrypted as this may provide the opportunity to recover the data exchanged between computers in the network, such as synced files and credential information [63],[64],[65].

- **Volatile memory dumps** – Cloud usage always leaves traces in the physical memory, which could turn out to be information of value in criminal investigations. As demonstrated by the findings from [64], [66], it may be necessary to collect volatile memory dumps as a potential source of residual artefacts contributing to an extended investigation of above mentioned items [67].

We seek to recover the following artefacts of evidential value from the server:

- **Administrative and file management metadata** – The server application generated instances that provide copies of the sync and file management as well as authentication and encryption metadata of the clients.
- **Cloud logging and authentication data:** Records of cloud transaction and authentication events/requests made by the clients to the cloud hosting instance, which are often accompanied with identifying information such as node IDs, logical addresses, session IDs, and timestamps that will facilitate provenance determination.
- **Data repositories** - The data uploaded by the users to the server, which may assist a practitioner in recovering the synced files from the cloud hosting environment.

4. *Reporting and presentation.* This final phase relates to the preparation and presentation of the information resulting from the analysis phase. As observed by Cahyani et al. [68],[69],[70], it is important for forensics examiners to articulate and explain complex and technical forensic terminologies to the judiciary and juries in order for them to understand "how these crimes were committed, what digital evidence is and where it may exist, and how the process of digital evidence collection was undertaken by the forensic investigators" [71].

5 EVIDENTIAL ANALYSIS OF THE SYNCANY DESKTOP CLIENTS

An inspection of the directory listings determined that artefacts of the Syncany client could be predominantly located in the (hidden) '.syncany' directory specific to the sync folders as well as user-specific 'Syncany' directory in *%Users%\<User Profile>\AppData\Roaming*, */home/<User Profile>/.config/*, and */Users/<User Profile>/.config/* on the Windows, Ubuntu, and Mac OSX clients (respectively). The former stores folder-specific configurations and caches while the latter manages the user-specific configurations. Both the '.syncany' and 'syncany' directories remained after uninstallation of the client applications [59].

5.1 Sync and File Management Metadata

Analysis of the user-specific 'Syncany' directory located the last used Process Identification (PID) for the daemon process in the */%syncany%/daemon.pid* file. The PID could be useful information for file-to-process mapping i.e., dumping the in-memory data storage and synced files from the memory space of the Syncany process using the 'memdump' function of Volatility. Within the */%syncany%/daemon.xml* file we located a list of directory paths for the sync folders and information about whether the sync folders were enabled in the 'path' and 'enabled' attributes of the 'folders' property. The folder metadata is meant for the daemon process to identify the sync directories [59].

Examination of the *%.syncany%/config.xml* file revealed the 20-character machine and display names, in the 'ma-

chineName' and 'displayName' properties (respectively). The machine name is the random local machine name used to identify a client associated with a repository, while the display name is the human readable user name for given local machine [61]. The file also recorded the type of backend storage in use in the 'type' attribute of the 'connection' property. Knowing the type of backend storage in use may assist a practitioner to extrapolate the potential artefacts and search terms in an investigation involving the use of the Syncany cloud storage. For example, if the WebDAV protocol is used, the web server logs can be a potential source of relevant information in the investigation. The use of the WebDAV protocol may also suggest the potential to include the HTTP request/response method names as part of the evidence search, such as 'PROPFIND'[3].

Further details of the sync and file management metadata could be located in the 'FILEVERSION_FULL' table of the *%.syncany%/db/local.db* database. Specifically, we recovered the unique identifications (ID), directory paths, file sizes, SHA1 checksums, and permission information (in POSIX format) for the sync folders and files specific to a repository in the 'FILEHISTORY_ID', 'PATH', 'SIZE', 'FILECONTENT_CHECKSUM', and 'POSIXPERMS' table fields. The file checksum information could enable identification of a synced file outside the sync folder, as been moved or deleted by the user.

5.2 Authentication and Encryption Metadata

Within the 'connection' property of the *%/.syncany%/config.xml* file, we located the URL to the repository as well as the username and encrypted password for the backend storage in the 'url', 'username', and 'password' attributes. The encrypted password could be potentially decrypted using the masterkey and masterkey salt stored in the 'configEncryptionKey' attribute of the 'userConfig' property in the */%syncany%/config.xml* file [61]. Having access to the backend storage could potentially enable a practitioner to recover other data that might have been hosted on the web server.

Upon further examination of the *%/.syncany%/config.xml* file, we located the masterkey and masterkey salt used to derive the AES-128 encryption key for the encrypted files in the repository, in the 'key' and 'salt' attributes of the 'masterKey' property [61]. The latter is similar to that stored in the *%.syncany%/master* file. Although it is computionally infeasible to reverse the masterkey to derive the user password due to the implementation of the PBKDF2 HMAC-SHA1 (with 1 million iterations) hashing algorithm, the masterkey could be useful when seeking to decrypt the encrypted files in the repository. This is especially pertinent when the synced files were deleted as the practitioner could potentially recover caches of the (encrypted) databases and multichunks of synced files uploaded to the server from the hidden *%.syncany%/cache* directory, which is undeleted during uninstallation of the client applications and often unnoticed by the user.

With the masterkey and encrypted file in place, along with the random salt available in the 20th to 31st byte of the file header (of the encrypted file), the practitioner could work on deriving the encryption key using the HKDF key derivation function (with SHA256). The encryption key, when derived and combined with the random initialization vector available in the 32nd to 43th byte of the file header (of the encrypted file), could then be used to decrypt the ciphertext (from byte 76th of the encrypted file) using the default AES-128 (GCM mode) encryption algorithm [61].

5.3 Cloud Transaction History

Upon inspections of the directory listings, we identified the sync folders' initialisation/addition timestamps from the creation times of the '.syncany' sub-directory.

From the 'FILEVERSION_FULL' table of the *%.syncany%/db/local.db* database, we observed that records of the last modified and updated timestamps could be located for the sync directories/files (file tree) associated with the repository in the 'LASTMODIFIED' and 'UPDATED' table fields, respectively. Sync directories/files that were modified could be differentiated from the 'VERSION' table column given the value of more than 1. Additionally, the table holds the directory/file modification status in the 'STATUS' table field, indicating whether a sync directory/file is newly added (with the value 'NEW'), modified ('CHANGED;), or deleted ('DELETED'). We could also determine the machine names for the clients that made changes to the sync data (e.g., add and remove sync folders or files) as well as the database local timestamps associated with the changes in the 'DATABASEVERSION_CLIENT' and 'DATABASEVERSION_LOCALTIME' table fields, respectively (see Fig. 2).

Examination of the */%syncany%/logs/syncany.log* revealed the client application's access timestamps as well as file creation, modification, and deletion timestamps of the sync folders/files. The log file also provided the directory paths for the sync folders/files alongside the corresponding property information as detailed in TABLE 3.

Further examination of the Windows client determined that the prefetch files could be differentiated from 'SYNCANY-LATEST-X86_64-2.EXE-FDCA2B01.pf'. We recovered information of forensic interest, such as the number of times the application has been loaded and the last run time from the prefetch files. We also located shortcuts to the loader files *%Program Files (x86)%\Syncany\bin\launcher.vbs* and *%Program Files (x86)%\Syncany\unins000.exe* at *%ProgramData%\Microsoft\Windows\Start Menu\Programs\Syncany\Syncany.lnk* and *%ProgramData%\Microsoft\Windows\Start Menu\Programs\Syncany\Uninstall Syncany.lnk* (respectively), providing potential for alternative methods for identifying the application installation and last run timestamps.

[3] In WebDAB, the 'PROPFIND' method is used to retrieve the properties from a web resource [72].

DATABASEVERSION_STATUS	DATABASEVERSION_LOCALTIME	DATABASEVERSION_CLIENT	DATABASEVERSION_VECTORCLOCK_SERIALIZED	FILEHISTORY_ID	VERSION	DATABASEVERSION_ID	PATH	TYPE	STATU
MASTER	2015-10-01 18:52:38	UYCrwWXGXKvbOYKtZBGc	(UYCrwWXGXKvbOYKtZBGc1)	8509cfe7c5b1f82e305c0aaade57c8f0745de5c6	1	0	.syignore	FILE	NEW
MASTER	2015-10-01 19:26:14	UYCrwWXGXKvbOYKtZBGc	(UYCrwWXGXKvbOYKtZBGc2)	3d247e3ddc8f73c49c98a3a9abd89897dc0a8179	1	1	Enron3111.pdf	FILE	NEW
MASTER	2015-10-01 19:26:14	UYCrwWXGXKvbOYKtZBGc	(UYCrwWXGXKvbOYKtZBGc2)	bab75c4cb7537d5a7641869834913123eed4304c	1	1	Enron3111.rtf	FILE	NEW
MASTER	2015-10-01 19:26:14	UYCrwWXGXKvbOYKtZBGc	(UYCrwWXGXKvbOYKtZBGc2)	9bec78b6de24d0df15e7469e72b77cdbf163bd55	1	1	Enron3111.txt	FILE	NEW
MASTER	2015-10-01 19:26:14	UYCrwWXGXKvbOYKtZBGc	(UYCrwWXGXKvbOYKtZBGc2)	c3bd62f81eda2db41d6beb294ccfbb36f452f99c	1	1	Enron3111.jpg	FILE	NEW
MASTER	2015-10-01 19:26:14	UYCrwWXGXKvbOYKtZBGc	(UYCrwWXGXKvbOYKtZBGc2)	ee439cb5636 17c8a9d7de9c8c533bb015a946d1a	1	1	Enron3111.docx	FILE	NEW
MASTER	2015-10-01 19:26:14	UYCrwWXGXKvbOYKtZBGc	(UYCrwWXGXKvbOYKtZBGc2)	3a947cff54c9e3b748f9cb4f65752b16d4c637e8	1	1	Enron3111.zip	FILE	NEW
MASTER	2015-10-01 19:42:10	PqPKcLWznojHZVLslgNo	(PqPKcLWznojHZVLslgNo1,UYCrwWXGXKvbOYKtZBGc2)	2ab118932cfbf75a27f0657eefec8f1eb2b737b8	1	2	WindowsToUbuntu/Enron3111.pdf	FILE	NEW
MASTER	2015-10-01 19:42:10	PqPKcLWznojHZVLslgNo	(PqPKcLWznojHZVLslgNo1,UYCrwWXGXKvbOYKtZBGc2)	00a31c436450b3a71e58ad12c28f9b4bebcc2c45	1	2	WindowsToUbuntu/Enron3111.txt	FILE	NEW
MASTER	2015-10-01 19:42:10	PqPKcLWznojHZVLslgNo	(PqPKcLWznojHZVLslgNo1,UYCrwWXGXKvbOYKtZBGc2)	10f26510ae4b8f5c3a6fac86c499eca2266a914f	1	2	WindowsToUbuntu/Enron3111.zip	FILE	NEW
MASTER	2015-10-01 19:42:10	PqPKcLWznojHZVLslgNo	(PqPKcLWznojHZVLslgNo1,UYCrwWXGXKvbOYKtZBGc2)	0fa188da9eb4c6aaa4d495d8ad9e20d972ff54fb	1	2	WindowsToUbuntu/Enron3111.docx	FILE	NEW
MASTER	2015-10-01 19:42:10	PqPKcLWznojHZVLslgNo	(PqPKcLWznojHZVLslgNo1,UYCrwWXGXKvbOYKtZBGc2)	588275ee75c158cc3a1071207d17e8285281d7f	1	2	WindowsToUbuntu/Enron3111.rtf	FILE	NEW
MASTER	2015-10-01 19:42:10	PqPKcLWznojHZVLslgNo	(PqPKcLWznojHZVLslgNo1,UYCrwWXGXKvbOYKtZBGc2)	248eaab9dc360ee0a2ac601c1afad9fec6551842	1	2	WindowsToUbuntu/Enron3111.jpg	FILE	NEW
MASTER	2015-10-01 19:42:10	PqPKcLWznojHZVLslgNo	(PqPKcLWznojHZVLslgNo1,UYCrwWXGXKvbOYKtZBGc2)	0b4f85e8b1435574e71332e7d023931219e37d4f	1	2	WindowsToUbuntu	FOLDER	NEW
MASTER	2015-10-01 19:48:32	UYCrwWXGXKvbOYKtZBGc	(PqPKcLWznojHZVLslgNo1,UYCrwWXGXKvbOYKtZBGc3)	10f26510ae4b8f5c3a6fac86c499eca2266a914f	2	3	WindowsToUbuntu/Enron3111.zip	FILE	DELETE
MASTER	2015-10-01 19:48:32	UYCrwWXGXKvbOYKtZBGc	(PqPKcLWznojHZVLslgNo1,UYCrwWXGXKvbOYKtZBGc3)	00a31c436450b3a71e58ad12c28f9b4bebcc2c45	2	3	WindowsToUbuntu/Enron3111.txt	FILE	DELETE
MASTER	2015-10-01 19:48:32	UYCrwWXGXKvbOYKtZBGc	(PqPKcLWznojHZVLslgNo1,UYCrwWXGXKvbOYKtZBGc3)	588275ee75c158cc3a1071207d17e8285281d7f	2	3	WindowsToUbuntu/Enron3111.rtf	FILE	DELETE

Fig. 2. Portion of the 'FILEVERSION_FULL' table recovered in our research.

TABLE 3
ENTRIES OF PARTICULAR INTEREST IN SYNCANY.LOG

Relevance	Examples of log entries
Assists a practitioner in the identification of the username for the backend storage.	• 1-10-15 18:52:20.274 | PluginSettingsP | main | INFO : Setting field 'username' with value 'syncanyserver'
Assists a practitioner in identifying the repository initiation time for a sync folder as well as the directory path and repository URL.	• 1-10-15 18:52:27.196 | ManagementReque | main | SEVE : Executing InitOperation for folder /home/suspectpc/SyncanyUbuntuClient ... • 1-10-15 18:52:34.807 | WebdavTransferM | IntRq/SyncanyU | INFO : WebDAV: Uploading local file /home/suspectpc/SyncanyUbuntuClient/.syncany/master to http://172.16.38.180/webdav/UbuntuRepo/master ...
Assists a practitioner in identifying the time when a sync folder is connected to an existing repository, including the directory path.	• 1-10-15 19:32:38.646 | ManagementReque | main | SEVE : Executing ConnectOperation for folder /home/suspectpc/SyncanyWindowsDownloadToUbuntu .
May provide a practitioner with the addition, modified and updated timesstamps for a sync file as well as the property information such as filename and modification counter (similarly to the records in the 'FILEVERSION_FULL' table of the *%.syncany%/db/local.db* database).	• 1-10-15 19:26:14.015 | Indexer | Thread-68 | INFO : * Added file version: FileVersion [version=1, path=Enron3111.zip, type=FILE, status=NEW, size=30967, lastModified=Sat Dec 13 08:35:00 PST 2014, linkTarget=null, checksum=75a666ba87fef0f8425a71edcd621d0a4367aa47, updated=Thu Oct 01 19:26:14 PDT 2015, posixPermissions=rw-r--r--, dosAttributes=--a-]
May provide a practitioner with the addition, modified and updated timestamps for a sync folder as well as the property information such as filename and modification counter (similarly to the records in the 'FILEVERSION_FULL' table of the *%.syncany%/db/local.db* database).	• 1-10-15 19:42:18.751 | FileSystemActio | NotifyThread | INFO : with winning version : FileVersion [version=1, path=WindowsToUbuntu, type=FOLDER, status=NEW, size=4096, lastModified=Mon Sep 28 21:40:44 PDT 2015, linkTarget=null, checksum=null, updated=Thu Oct 01 19:42:10 PDT 2015, posixPermissions=rwxr-xr-x, dosAttributes=----]
Enables a practitioner in determining the deletion time of a sync folder.	• 1-10-15 20:14:17.091 | AppIndicatorTra | PySTDIN | INFO : Python Input Stream: Removing folder '/home/UbuntuPc/SyncanyUbuntuClient' ...
By searching for the 'file' tag, a practitioner can identify the filenames associated with a sync folder.	• 1-10-15 19:26:17.032 | AppIndicatorTra | Timer-0 | INFO : Sending message: <updateRecentChangesGuiInternalEvent> <recentChanges> <file>/home/suspectpc/SyncanyUbuntuClient/Enron3111.zip</file> <file>/home/suspectpc/SyncanyUbuntuClient/Enron3111.txt</file> <file>/home/suspectpc/SyncanyUbuntuClient/Enron3111.rtf</file> … </recentChanges> </updateRecentChangesGuiInternalEvent>

On the Ubuntu client, we could discern the installation timstamp from the Syslog entry "*2015-10-01 18:32:39 install syncany:all <none> 0.4.6.alpha*" in the dpkg log and "*Oct 1 17:45:41 ubuntu AptDaemon.Worker: INFO: Installing local package file: /home/UbuntuPc/Desktop/syncany-latest.deb*". Analysis of the Zeitgeist database (*/.local/share/zeistgeist/activity.sqlite*) revealed the current directory paths for the sync folders/files in the 'subj_url' and 'subj_current_uri' table fields of the 'event_view' table, as well as original directory paths for the sync folders/files in the 'subj_origin_uri' table field. This suggested that if a sync (download) file has been moved from the sync folder, there will still be records remaining in the Zeitgeist database to enable identification of the sync files. The Zeitgeist database also held the types of actions made to the folders/files of relevance (e.g., Access, Create, Move, Leave, and Delete), alongside the timestamps associated with the actions; each action created a record in the 'interpretation' table field. Analysis of the Gnome's */home/<User Profile>/.local/share/recently-used.xbel* log located caches of sync directory paths accessed by the client application, which included the last added, modified, and

Fig. 4. Portion of the recently-used.xbel log for the Syncany Ubuntu client application.

"<?xml version="1.0" encoding="utf-8"?>
<D:multistatus xmlns:D="DAV:" xmlns:ns0="DAV:">
<D:response xmlns:lp1="DAV:" xmlns:lp2="http://apache.org/dav/props/">
<D:href>/webdav/MacRepo/actions/</D:href>
<D:propstat>
<D:prop>
<lp1:resourcetype><D:collection/></lp1:resourcetype>
<lp1:creationdate>2015-09-29T15:47:30Z</lp1:creationdate>
<lp1:getlastmodified>Tue, 29 Sep 2015 15:47:30 GMT</lp1:getlastmodified>
<lp1:getetag>"1000-520e4bb450e3a"</lp1:getetag>
…".

Fig. 3. An excerpt of the response body of the PROFIND HTTP method for the Syncany repository.

visited timestamps in the 'bookmark' property. Fig. 4 shows that the directory paths as well as last added, modified, and added timestamps could be differentiated from the 'href', 'visited', 'modified', and 'added' entries in the 'bookmark' property. The figure also illustrates that the Gnome's log records the number of times the client application has accessed a sync folder in the 'count' entry of the 'bookmark:applications' property.

5.4 Data Storage

The sync directories could be discerned from directories containing the hidden '.syncany' directory. Analysis of the thumbnail cache located thumbnail images for the synced files in *%AppData%\Local\Microsoft\Windows\Explorer* and */home/<User Profile>/.cache/thumbnails/large/* of the Windows and Ubuntu clients (respectively). The thumbnail cache could be differentiated from the thumbnail header property 'jtEXtThumb::URI file:' that referenced the directory path for the sync folder in the latter. The files deleted (locally and remotely) could be recovered from the unallocated space/partition.

5.5 Network Analysis

Our findings from network analysis revealed that the IP address of the Syncany server using the WebDAV protocol as the backend storage could be differentiated from the network packets of the PROPFIND HTTP method "*PROPFIND /webdav/<Repo Name>/actions/ HTTP/1.1*". Analysis of the network payload of the PROPFIND HTTP method determined that the last creation and modified times of the repositories could be recovered from the 'creationdate' and 'getlastmodified' entries of the 'prop' property (see Fig. 3).

Further examination of the network traffic recovered the master key salt of the repositories from the HTTP methods "*PUT /webdav/<Repo Name>/master HTTP/1.1*" and "*GET /webdav/<Repo Name>/master HTTP/1.1*". Since the master key salt is only created (once) during the repository initialisation or connection, the timestamp of which could reflect the sync folder creation or addition time.

Undertaking data carving of the network traffic produced the repositories uploaded to or downloaded from the backend storage intact. However, the repositories were fully encrypted by default, and viewing the data in the repositories requires access to the client application and user password.

5.6 Memory Analysis

The findings from the 'pslist' function of Volatility showed that the process name was masqueraded with 'java.exe'. However, we could differentiate the PID from the last used PID recorded in the daemon.pid and syncany.log files. Other process information was also recovered such as the process identifier (PID), parent process identifier (PPID), as well as process initiation and termination times. Examinations of the memory dumps using the 'netscan' or 'netstat' function of Volatility recovered

the network socket records associated with the Syncany processes such as the server and clients' IP addresses, port numbers, and socket states, which could be useful supporting information for network analysis.

Undertaking data carving of the memory space of the Syncany daemon process recovered files of forensic interest, such as config.xml, syncany.log and local.db database, intact. The presence of the data remnants in plain text also suggested that it is possible to recover the texts from the files by searching for the file entries of relevance (see Sections 5.1-5.3 for details). We also found that the *%.syncany%/config.xml* file could be potentially carved using the header and footer values of "3C 63 6F 6E 66 69 67 3E…3C 2F 63 6F 6E 6E 65 63 74 69 6F 6E 3E 0A 3C 2F 63 6F 6E 66 69 67 3E" in hex format or "<config>...</connection>.</config>" in non-Unicode string format, but such capability might not be present in a future release of the software. As for the records from the 'FILEVERSION_FULL' table of the local.db database, a search for the machine name (identified from the *%.syncany%/config.xml* file) could enable future searches of the raw cell data in the physical memory dump.

6 Evidential Analysis of the Syncany Ubuntu Server

From the examination of the Ubuntu server, we determined that the repositories hosted using the WebDAV protocol could be located in */var/www/webdav/<Repo Name>/*. Within the repositories, we found copies of the databases and multichunks of synced files uploaded by the clients in the 'databases' and 'multichunks' directories, but the data were fully encrypted. At the time of this research, it appears that there is no method outside the Syncany client available to reconstruct the data in the repository. Hence, accessing the sync data is only possible with the client application. As the Syncany storage cloud service is server independent, we determined that it is also possible to reconstruct the sync data from the repositories outside the server environment. This involves the following steps:

1. Make a logical copy of a repository on the backend storage in a forensically sound manner.
2. Install the Syncany client application on a third-party workstation.
3. Extract copy of the repository from the forensic image to the forensic workstation.
4. Run the Syncany application on the third-party workstation, and set up a sync folder from the repository using the "sy connect" command or choosing the "Add an existing Syncany folder" option on the GUI. It should be noted that the process may require the user password to be entered before the sync folder can be created.

An inspection of the Apache access log */var/log/apache2/access.log* showed that the log entries of the HTTP requests made by the Syncany clients to the WebDAV backend storage were in the format "*<Client's IP address>* - *<Username for the backend storage>* *[The time when a request was received]* "*<Request line>*" *<Status code>* *<Size of the object uploaded or downloaded by the client to the backend storage>* " –" "*<User agent>*"". Since the the sync files were uploaded in multichunks and the filenames were not visible, we could only estimate the timestamps of cloud transaction events in relation to a repository based on the requests made to the 'multichunks' directory of the repository. Having said that, we were able to determine the initialisation time for a repository based on the HTTP request for the masterkey salt "*172.16.38.132 - syncanyserver [01/Oct/2015:18:52:34 -0700] "PUT /webdav/UbuntuRepo/master HTTP/1.1" 201 480 "-" "Sardine/UNAVAILABLE"*" i.e., only available during initialisation or connection of a repository.

7 Reporting and Presentation

A timeline of the data from the various sources was outlined to demonstrate the cloud transactions from the clients and the server (see TABLE 4). To fully present the complete timeline of the analysis, we would need to use a visualisation software (we refer interested reader to [44],[73] for a recent review of forensic visualization literature, and a proof of concept for a three-stage forensic data storage and visualization life cycle). Therefore, we only present the timeline for the Enron3111.txt sample file from the Ubuntu client.

8 Concluding Remarks

In this paper, we presented an investigation approach for Syncany, a cloud-enabled big data private cloud storage service, with the aim of reducing the time and resources required in a real-world investigation. We demonstrated that artefacts of the file synchronisation, file management, authentication, and encryption metadata could be recovered from the folder-specific '.syncany' directory on the desktop clients. In practice, the presence of the folder-specific '.syncany' directory could also prove useful for the identification of the sync directories from the directory listing, as well as other file system transaction logs such as shortcuts, event logs, $LogFile, $MFT, $UsnJrnl, registry ('RecentDocs', 'UserAssist', 'Run', and 'ComDig32' etc.), Zeitgeist and recently-used.xbel logs, and thumbnail cache.

Our examination of the physical memory indicated that the memory dumps is a potential alternative for recovering the application caches, logs, and other authentication and encryption metadata available in the configuration and log files in plain text. However, the user password was not located on both the storage media and memory dumps, suggesting that a practitioner can only obtain the user password via an offline brute-force/dictionary attack or directly from the user. Nevertheless, a practitioner must keep in mind that memory changes frequently according to user activities and will be wiped as soon as the system is shut down, and hence should be undertaken as quickly as possible to increase the likelihood of preserving the artefacts.

Analysis of the network traffic produced unique identifiable information such as URL references to the reposito-

TABLE 4
TIMELINE FOR SYNCANY FORENSICS

Source	Key Date	Event Type	Comment
Autopsy file List	2015-10-01 01:14:41	Created	Created directory */home/UbuntuPc/SyncanyUbuntuClient*.
Autopsy file list	2015-10-01 10:26:10	Created	Created file */home/UbuntuPc/SyncanyUbuntuClient/Enron3111.txt*.
Dpkg log	2015-10-01 18:32:39	Created	Installed Syncany Ubuntu client application version 0.4.6.alpha (i.e., "*2015-10-01 18:32:39 install syncany:all <none> 0.4.6.alpha*").
syncany.log	2015-10-01 18:34:12	Last run	Last run Syncany client application (i.e., *1-10-15 18:34:12.064 | DaemonOperation | Thread-5 | INFO : Starting daemon operation with action RUN ....*)
syncany.log	2015-10-01 18:52:27	Initialised	Initialised the repository for sync directory */home/suspectpc/SyncanyUbuntuClient* (i.e., "*1-10-15 18:52:27.196 | ManagementReque | main | SEVE : Executing InitOperation for folder /home/suspectpc/SyncanyUbuntuClient…*").
Autopsy file list	2015-10-01 18:52:28	Initialised	Initialised the repository for sync directory */home/UbuntuPc/SyncanyUbuntuClient* (i.e., created */home/UbuntuPc/SyncanyUbuntuClient/.syncany*).
local.db	2015-10-01 19:26:14	Created	Locally added sync file */home/UbuntuPc/SyncanyUbuntuClient/Enron3111.txt*.
syncany.log	2015-10-01 19:26:14	Added	Added sync file */home/UbuntuPc/SyncanyUbuntuClient/Enron3111.txt (i.e., "1-10-15 19:26:14.004 | Indexer | Thread-68 | INFO : * Added file version: FileVersion [version=1, path=Enron3111.txt, type=FILE, status=NEW, size=2734, lastModified=Sat Dec 13 08:33:11 PST 2014, linkTarget=null, checksum=ceadc4b7d47af4125a68e03ec3141cb3fde407ff, updated=Thu Oct 01 19:26:14 PDT 2015, posixPermissions=rw-r--r--, dosAttributes=--a-]*").
local.db	2015-10-01 19:42:10	Created	Node 'WindowsPc' (Machine Name='PqPKcLWznojHZVLslgNo') uploaded sync folder */home/UbuntuPc/SyncanyUbuntuClient/WindowsToUbuntu*.
local.db	2015-10-01 19:42:10	Created	Node 'WindowsPc' (Machine Name='PqPKcLWznojHZVLslgNo') uploaded sync file */home/UbuntuPc/SyncanyUbuntuClient/WindowsToUbuntu/Enron3111.txt*.
Autopsy file list	2015-10-01 19:42:18	Created	Created sync directory */home/UbuntuPc/SyncanyUbuntuClient/WindowsToUbuntu*.
syncany.log	2015-10-01 19:42:18	Added	Added sync folder */home/UbuntuPc/SyncanyUbuntuClient/WindowsToUbuntu* (i.e., "*1-10-15 19:42:18.751 | FileSystemActio | NotifyThread | INFO : with winning version : FileVersion [version=1, path=WindowsToUbuntu, type=FOLDER, status=NEW, size=4096, lastModified=Mon Sep 28 21:40:44 PDT 2015, linkTarget=null, checksum=null, updated=Thu Oct 01 19:42:10 PDT 2015, posixPermissions=rwxr-xr-x, dosAttributes=----]*").
syncany.log	2015-10-01 19:42:18	Added	Added sync file */home/UbuntuPc/SyncanyUbuntuClient/WindowsToUbuntu/Enron3111.txt* (i.e., "*1-10-15 19:42:18.848 | FileSystemActio | NotifyThread | INFO : + NewFileSystemAction [file1=null, file2=FileVersion [version=1, path=WindowsToUbuntu/Enron3111.txt, type=FILE, status=NEW, size=2734, lastModified=Sat Dec 13 08:33:11 PST 2014, linkTarget=null, checksum=ceadc4b7d47af4125a68e03ec3141cb3fde407ff, updated=Thu Oct 01 19:42:10 PDT 2015, posixPermissions=rw-r--r--, dosAttributes=--a-]]*").
Autopsy file List	2015-10-01 19:42:18	Created	Created sync file */home/UbuntuPc/SyncanyUbuntuClient/WindowsToUbuntu/Enron3111.txt*.
Autopsy file list; Trashinfo	2015-10-01 19:48:29	Deleted	Deleted sync file */home/UbuntuPc/SyncanyUbuntuClient/WindowsToUbuntu/Enron3111.txt*.
local.db	2015-10-01 19:48:32	Deleted	Deleted sync file */home/UbuntuPc/SyncanyUbuntuClient/WindowsToUbuntu/Enron3111.txt*.

ries and HTTP requests associated with the cloud transaction events. It was also possible to recover copies of the repositories uploaded to and downloaded from the backend storage intact, but the data were not human readable even with the use of HTTP protocol. This is, perhaps, not surprising as the data are fully encrypted locally before being uploaded to the backend storage [61]. However, we remark that the artefacts may vary according to network protocol implementations i.e., if the HTTPS protocol is used, then the HTTP requests will be fully encrypted.

As expected, examination of the Syncany server demonstrated that the repositories were fully encrypted on the backend storage, and it appeared that accessing the sync data is only possible with the client application. The ability to reconstruct the repository outside the cloud hosting environment suggested that a practitioner can undertake logical acquisition of the repositories in the event where physical acquisition is not feasible. Our research also demonstrated that the web application log could be a source of information on the logical locations of the clients, and hence should be collected whenever practical. A summary of artefacts of interest with the Syncany private cloud storage service is presented in TABLE 5.

Collectively, our research highlighted the challenges that could arise during forensic investigations of cloud-enabled big data solutions. While the use of deduplication/chunking and encryption technologies can benefit users by providing an efficient and secure means for managing big data, evidence collection and analysis may necessitate intercepting and collecting of password and utilisation of other vendor-specific applications. This can be subject to potential abuse by cyber criminals seeking to hide their tracks. Without the cooperation of the user (suspect), forensics endeavours may end up an exercise in futility. Therefore, we suggest vendors implement a forensically friendly logging mechanism (e.g., providing information about who accesses the data, what data has been accessed, from where did the user access the data, and when did the user access the data) that supports the collection of the raw log data outside the encrypted da-

TABLE 5
SUMMARY OF FINDINGS FROM THE SYNCANY PRIVATE CLOUD STORAGE SERVICE

Artefact Category	Sources of Information
Sync and file management metadata	• Folder-specific *%.syncany%/config.xml* file and *%.syncany%/db/local.db* database • User-specific */%syncany%/logs/syncany.log*, */%syncany%/daemon.xml*, and */%syncany%/daemon.pid* files
Authentication and encryption metadata	• Folder-specific *%.syncany%/config.xml* file • User-specific */%syncany%/config.xml* file • Encrypted files in */%.syncany%/cache/* • Encrypted files in */%<Repo Name>/databases/* and */%<Repo Name>/multichunks/* of the backend storage
Cloud transaction history	• Folder-specific *%.syncany%/db/local.db* database • SYNCANY-LATEST-X86_64-2.EXE-FDCA2B01.pf, Syncany.lnk, and Uninstall Syncany.lnk on the Windows client • The Zeitgeist, Gnome' recently-used.xbel, system, and dpkg logs of the Ubuntu client • */var/log/apache2/access.log* on the Ubuntu server
Data storage	• Directories containing the '.syncany' directory • Copies of the thumbnail images for synced files in the thumbnail cache • Unallocated space/partitions for deleted files
Network analysis	• IP addresses of the clients and cloud hosting instance • The repository initialisation and addition timestamps • Masterkey salts of the repositories • Copies of the encrypted repositories uploaded to or download from the backend storage intact
Memory analysis	• Process name could be differentiated from 'java'exe' • Copies of the *%.syncany%/db/local.db*, */%syncany%/logs/syncany.log*, and */%syncany%/daemon.xml* files in plain text

tasets by default. In a recent work, for example, Ab Rahman et al. [58] highlighted the importance of forensic-by-design and presented a conceptual forensic-by-design framework.

Future work would include extending this study to other private cloud storage services such as Seafile to have an up-to-date and comprehensive forensic understanding and facilitate in greening investigation of big data platforms.

REFERENCES

[1] International Data Corporation (IDC), "Worldwide and Regional Public IT Cloud Services 2013–2017 Forecast," 2014. [Online]. Available: http://www.idc.com/getdoc.jsp?containerId=242464. [Accessed: 20-Nov-2015].

[2] Gartner, "Forecast: IT Services, 2011-2017, 4Q13 Update," 2013. [Online]. Available: https://www.gartner.com/doc/2637515/forecast-it-services--q. [Accessed: 26-May-2016].

[3] K. K. R. Choo and R. Sarre, "Balancing Privacy with Legitimate Surveillance and Lawful Data Access," *IEEE Cloud Comput.*, vol. 2, no. 4, pp. 8–13, Jul. 2015.

[4] S. Nepal, R. Ranjan, and K. K. R. Choo, "Trustworthy Processing of Healthcare Big Data in Hybrid Clouds," *IEEE Cloud Comput.*, vol. 2, no. 2, pp. 78–84, 2015.

[5] A. Dehghantanha and K. Franke, "Privacy-Respecting Digital Investigation," in *Proceedings of 2014 Twelfth Annual International Conference on Privacy, Security and Trust (PST)*, 2014, pp. 129–138.

[6] K.-K. R. Choo, "Cloud Computing: Challenges and Future Directions," *Trends Issues Crime Crim. Justice*, vol. 460, pp. 1–7, 2010.

[7] K.-K. R. Choo, "Organised Crime Groups in Cyberspace: A Typology," *Trends Organ. Crime*, vol. 11, no. 3, pp. 270–295, 2008.

[8] N. H. Ab Rahman and K.-K. R. Choo, "A Survey of Information Security Incident Handling in the Cloud," *Comput. Secur.*, vol. 49, no. C, pp. 45–69, 2015.

[9] L. Li, R. Lu, K. K. R. Choo, A. Datta, and J. Shao, "Privacy-Preserving-Outsourced Association Rule Mining on Vertically Partitioned Databases," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 8, pp. 1847–1861, 2016.

[10] R. Jiang, R. Lu, and K.-K. R. Choo, "Achieving High Performance and Privacy-Preserving Query over Encrypted Multidimensional Big Metering Data," *Future Gener. Comput. Syst*, 2017.

[11] X. Liu, R. Choo, R. Deng, R. Lu, and J. Weng, "Efficient and Privacy-Preserving Outsourced Calculation of Rational Numbers," *IEEE Trans. Dependable Secure Comput.*, 2017.

[12] Y. Hu, J. Yan, and K.-K. R. Choo, "PEDAL: A Dynamic Analysis Tool for Efficient Tool for Efficient Concurrency Bug Reproduction in Big Data Environment," *Clust. Comput.*, vol. 19, no. 1, pp. 153–166, 2016.

[13] L. Zhao, L. Chen, R. Ranjan, K.-K. R. Choo, and J. He, "Geographical Information System Parallelization for Spatial Big Data Processing: A Review," *Clust. Comput.*, vol. 19, no. 1, pp. 139–152, 2015.

[14] Y. Yang, H. Zhu, H. Lu, J. Weng, Y. Zhang, and K.-K. R. Choo, "Cloud Based Data Sharing with Fine-Grained Proxy Re-Encrypted," *Pervasive Mob. Comput.*, vol. 28, pp. 122–134, 2016.

[15] B. G. Soltani Ashkan and A. Peterson, "How We Know the NSA Had Access to Internal Google and Yahoo Cloud Data," *The Washington Post*, 04-Nov-2013.

[16] M. Damshenas, A. Dehghantanha, R. Mahmoud, and S. bin Shamsuddin, "Forensics Investigation Challenges in Cloud Computing Environments," in *Proceedings of 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, 2012, pp. 190–194.

[17] J. M. Cauthen, "Executing Search Warrants in the Cloud," *FBI*, 10-Jul-2014. [Online]. Available: https://leb.fbi.gov/2014/october/executing-search-warrants-in-the-cloud. [Accessed: 26-May-2016].

[18] D. Quick and K.-K. R. Choo, "Big Forensic Data Reduction: Digital Forensic Images and Electronic Evidence," *Clust. Comput.*, vol. 19, no. 2, pp. 1–18, 2016.

[19] D. Quick and K.-K. R. Choo, "Impacts of Increasing Volume of Digital Forensic Data: A Survey and Future Research Challenges," *Digit. Investig.*, vol. 11, no. 4, pp. 273–294, 2014.

[20] D. Quick and K.-K. R. Choo, "Data Reduction and Data Mining Framework for Digital Forensic Evidence: Storage, Intelligence, Review, and Archive.," *Trends Issues Crime Crim. Justice*, vol. 480, pp. 1–11, 2014.

[21] F. Daryabar, A. Dehghantanha, N. I. Udzir, N. F. binti M. Sani, and S. bin Shamsuddin, "A Review on Impacts Of Cloud Computing and Digital Forensics," *Int. J. Cyber-Secur. Digit. Forensics*, vol. 2, no. 2, pp. 77–94, 2013.

[22] C. Hooper, B. Martini, and K.-K. R. Choo, "Cloud Computing and Its Implications for Cybercrime Investigations in Australia," *Comput. Law Secur. Rev.*, vol. 29, no. 2, pp. 152–163, 2013.

[23] National Institute of Standards and Technology (NIST), "NIST Cloud Computing Forensic Science Challenges." National Institute of Standards and Technology, 2014.

[24] D. Quick, B. Martini, and K.-K. R. Choo, *Cloud Storage Forensics*, 1st ed. Syngress, 2013.

[25] B. Martini and K.-K. R. Choo, "Cloud Forensic Technical Challenges and Solutions: A Snapshot," *IEEE Cloud Comput.*, vol. 1, no. 4, pp. 20–25, 2014.

[26] Y. TzeTzuen, A. Dehghantanha, A. Seddon, and S. H. Mohtasebi, "Greening Digital Forensics: Opportunities and Challenges," in *Signal Processing and Information Technology*, 2011, pp. 114–119.

[27] Y.-Y. Teing, A. Dehghantanha, K.-K. R. Choo, Z. Muda, M. T. Abdullah, and W.-C. Chai, "A Closer Look at Syncany Windows and Ubuntu Clients' Residual Artefacts," in *Security, Privacy and Anonymity in Computation, Communication and Storage*, G. Wang, I. Ray, J. M. A. Calero, and S. M. Thampi, Eds. Springer International Publishing, 2016, pp. 342–357.

[28] R. Marty, "Cloud Application Logging for Forensics," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, New York, NY, USA, 2011, pp. 178–184.

[29] C. Shields, O. Frieder, and M. Maloof, "A System for the Proactive, Continuous, and Efficient Collection of Digital Forensic Evidence," *Digit. Investig.*, vol. 8, Supplement, pp. S3–S13, 2011.

[30] J. Dykstra and A. T. Sherman, "Design and Implementation of Frost: Digital Forensic Tools for the Openstack Cloud Computing Platform," *Digit. Investig.*, vol. 10, pp. S87–S95, 2013.

[31] T. Gebhardt and H. P. Reiser, "Network Forensics for Cloud Computing," in *Distributed Applications and Interoperable Systems*, J. Dowling and F. Taïani, Eds. Springer Berlin Heidelberg, 2013, pp. 29–42.

[32] B. Martini and K.-K. R. Choo, "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, pp. 935–942.

[33] D. Quick and K.-K. R. Choo, "Forensic Collection of Cloud Storage Data: Does the Act of Collection Result in Changes to the Data or Its Metadata?," *Digit. Investig.*, vol. 10, no. 3, pp. 266–277, 2013.

[34] Y.-Y. Teing, A. Dehghantanha, K.-K. R. Choo, T. Dargahi, and M. Conti, "Forensic Investigation of Cooperative Storage Cloud Service: Symform as a Case Study," *J. Forensic Sci.*, 2017.

[35] H. Chung, J. Park, S. Lee, and C. Kang, "Digital Forensic Investigation of Cloud Storage Services," *Digit. Investig.*, vol. 9, no. 2, pp. 81–95, 2012.

[36] D. Quick and K.-K. R. Choo, "Dropbox Analysis: Data Remnants on User Machines," *Digit. Investig.*, vol. 10, no. 1, pp. 3–18, 2013.

[37] D. Quick and K.-K. R. Choo, "Google Drive: Forensic Analysis of Data Remnants," *J Netw Comput Appl*, vol. 40, pp. 179–193, 2014.

[38] D. Quick and K.-K. R. Choo, "Digital Droplets: Microsoft SkyDrive Forensic Data Remnants," *Future Gener. Comput. Syst.*, vol. 29, no. 6, pp. 1378–1394, 2013.

[39] J. S. Hale, "Amazon Cloud Drive Forensic Analysis," *Digit. Investig.*, vol. 10, no. 3, pp. 259–265, 2013.

[40] J. Farina, M. Scanlon, and M.-T. Kechadi, "BitTorrent Sync: First Impressions and Digital Forensic Implications," *Digit. Investig.*, vol. 11, Supplement 1, pp. S77–S86, 2014.

[41] M. Shariati, A. Dehghantanha, and K.-K. R. Choo, "SugarSync Forensic Analysis," *Aust. J. Forensic Sci.*, vol. 48, no. 1, pp. 1–23, 2015.

[42] M. Shariati, A. Dehghantanha, B. Martini, and K.-K. R. Choo, "Chapter 19 - Ubuntu One Investigation: Detecting Evidences on Client Machines," in *The Cloud Security Ecosystem*, Boston: Syngress, 2015, pp. 429–446.

[43] B. Blakeley, C. Cooney, A. Dehghantanha, and R. Aspin, "Cloud Storage Forensic: hubiC as a Case-Study," in *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2015, pp. 536–541.

[44] C. F. R. Tassone, B. Martini, and K.-K. R. Choo, "Visualizing Digital Forensic Datasets: A Proof of Concept," *J Forensic Sci*, 2017.

[45] B. Martini, Q. Do, and K.-K. R. Choo, "Chapter 15 - Mobile Cloud Forensics: An Analysis of Seven Popular Android Apps," in *The Cloud Security Ecosystem*, Boston: Syngress, 2015, pp. 309–345.

[46] F. Daryabar, A. Dehghantanha, B. Eterovic-Soric, and K.-K. R. Choo, "Forensic Investigation of Onedrive, Box, Google Drive and Dropbox Applications on Android and iOS Devices," *Aust. J. Forensic Sci.*, vol. 48, no. 6, pp. 1–28, 2016.

[47] B. Martini and K.-K. R. Choo, "An Integrated Conceptual Digital Forensic Framework for Cloud Computing," *Digit. Investig.*, vol. 9, no. 2, pp. 71–80, 2012.

[48] R. McKemmish, "What is Forensic Computing." Australian Institute of Criminology, Jun-1999.

[49] K. Kent, S. Chevalier, and T. Grance, "Guide to Integrating Forensic Techniques into Incident." 2006.

[50] B. Martini and K.-K. R. Choo, "Cloud Storage Forensics: Owncloud as a Case Study," *Digit. Investig.*, vol. 10, no. 4, pp. 287–299, 2013.

[51] N. Thethi and A. Keane, "Digital Forensics Investigations in the Cloud," in *Proceedings of 2014 IEEE International Advance Computing Conference (IACC)*, 2014, pp. 1475–1480.

[52] B. Martini and K.-K. R. Choo, "Distributed Filesystem Foren-

sics: Xtreemfs as a Case Study," *Digit. Investig.*, vol. 11, no. 4, pp. 295–313, 2014.

[53] M. Scanlon, J. Farina, and M.-T. Kechadi, "BitTorrent Sync: Network Investigation Methodology," in *Proceedings of 9th International Conference on Availability, Reliability and Security (ARES 2014)*, 2014, pp. 21-29.

[54] M. Scanlon, J. Farina, N. A. L. Khac, and T. Kechadi, "Leveraging Decentralization to Extend the Digital Evidence Acquisition Window: Case Study on BitTorrent Sync," *ArXiv14098486 Cs*, pp. 1–14, Sep. 2014.

[55] Y.-Y. Teing, A. Dehghantanha, K.-K. R. Choo, and L. T. Yang, "Forensic Investigation of P2P Cloud Storage Services and Backbone for IoT Networks: Bittorrent Sync as a Case Study," *Comput. Electr. Eng.*, 2017.

[56] Q. Do, B. Martini, and K.-K. R. Choo, "A Forensically Sound Adversary Model for Mobile Devices," *PLoS ONE*, vol. 10, no. 9, p. e0138449, 2015.

[57] Q. Do, B. Martini, and K.-K. R. Choo, "Is the data on your wearable device secure? An Android Wear smartwatch case study," *Softw. Pract. Exp.*, 2017.

[58] N. H. Ab Rahman, N. D. W. Cahyani, and K.-K. R. Choo, "Cloud Incident Handling and Forensic-By-Design: Cloud Storage as a Case Study," *Concurr. Comput. Pract. Exp.*, 2017.

[59] P. C. Heckel, "Syncany Explained: Idea, Progress, Development and Future (Part 1)," *Philipp's Tech Blog*, 18-Oct-2013. [Online]. Available: https://blog.heckel.xyz/2013/10/18/syncany-explained-idea-progress-development-future/. [Accessed: 26-May-2016].

[60] P. C. Heckel, "Deep Into the Code of Syncany - Command Line Client, Application Flow and Data Model (Part 2)," *Philipp's Tech Blog*, 14-Feb-2013. [Online]. Available: https://blog.heckel.xyz/2014/02/14/deep-into-the-code-of-syncany-cli-application-flow-and-data-model/. [Accessed: 26-May-2016].

[61] Syncany, "Syncany User Guide." [Online]. Available: https://syncany.readthedocs.io/en/latest/. [Accessed: 26-May-2016].

[62] S. Wilkinson, "ACPO Good Practice Guide for Digital Evidence." Association of Chief Police Officers (ACPO), 2012.

[63] Y. Mohd Najwadi and A. Dehghantanha, "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.

[64] K.-K. R. Choo and A. Dehghantanha, *Contemporary Digital Forensic Investigations of Cloud and Mobile Applications*. Boston: Syngress, 2017.

[65] A. Azfar, K.-K. R. Choo, and L. Liu, "Android Mobile Voip Apps: A Survey and Examination of Their Security and Privacy," *Electron. Commer. Res.*, vol. 16, no. 1, pp. 73–111, 2016.

[66] F. N. Dezfouli, A. Dehghantanha, R. Mahmoud, N. F. B. M. Sani, and S. bin Shamsuddin, "Volatile Memory Acquisition Using Backup for Forensic Investigation," in *Proceedings of 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, 2012, pp. 186–189.

[67] T. Y. Yang, A. Dehghantanha, K.-K. R. Choo, and Z. Muda, "Windows Instant Messaging App Forensics: Facebook and Skype as Case Studies," *PLOS ONE*, vol. 11, no. 3, p. e0150300, 2016.

[68] N. D. W. Cahyani, B. Martini, and K.-K. R. Choo, "Effectiveness of Multimedia Presentations in Improving Understanding of Technical Terminologies and Concepts: A Pilot Study," *Aust. J. Forensic Sci.*, vol. 49, no. 1, pp. 106–122, 2017.

[69] N. D. W. Cahyani, B. Martini, and K.-K. R. Choo, "Do Multimedia Presentations Enhance Judiciary's Technical Understanding of Digital Forensic Concepts? An Indonesian Case Study," in *Proceedings of 49th Annual Hawaii International Conference on System Sciences (HICSS 2016)*, Hawaii, 2016, pp. 5617–5626.

[70] N. D. W. Cahyani, B. Martini, K.-K. R. Choo, and A. M. N. Al-Azhar, "Forensic Data Acquisition from Cloud-of-Things Devices: Windows Smartphones as a Case Study," *Concurr. Comput. Pract. Exp.*, Jan. 2017.

[71] N. D. W. Cahyani, B. Martini, and K. K. R. Choo, "Using Multimedia Presentations to Enhance the Judiciary's Technical Understanding of Digital Forensic Concepts: An Indonesian Case Study," in *26th Australasian Conference on Information Systems (ACIS 2015)*, 2015, pp. 1–9. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/1606/1606.01425.pdf [Accessed: 02-Feb-2017].

[72] webdav.org, "HTTP Extensions for Distributed Authoring -- WEBDAV," 2016. [Online]. Available: http://www.webdav.org/specs/rfc2518.html. [Accessed: 02-Feb-2017].

[73] C. F. R. Tassone, B. Martini, and K.-K. R. Choo, "Chapter 11 - Forensic Visualization: Survey and Future Research Directions," in *Contemporary Digital Forensic Investigations of Cloud and Mobile Applications*, Boston: Syngress, pp. 163–184, 2017.

Yee-Yang Teing is a research fellow at the Putra University of Malaysia, and holds a Bachelor of Computer Forensics (First Class Honours). He is a Certified Ethical Hacker (CEH), Computer Hacking Forensic Investigator (CHFI) and Certified Security Analyst (ECSA). His research interests include cybercrime investigations, malware analysis and network security.

Dr. Ali Dehghan Tanha is a Marie-Curie International Incoming Fellow in Cyber Forensics, a fellow of the UK Higher Education Academy (HEA) and an IEEE Sr. member. He has served for many years in a variety of research and industrial positions. Other than Ph.D in Cyber Security he holds several professional certificates such as GXPN, GREM, GCFA, CISM, and CISSP

Kim-Kwang Raymond Choo (SM'15) received the Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio. He has served as the Special Issue Guest Editor of ACM Transactions on Embedded Computing Systems (2017; DOI: 10.1145/3015662), ACM Transactions on Internet Technology (2016; DOI: 10.1145/3013520), Digital Investigation (2016; DOI: 10.1016/j.diin.2016.08.003), Future Generation Com-

puter Systems (2016; DOI: 10.1016/j.future.2016.04.017), IEEE Cloud (2015; DOI: 10.1109/MCC.2015.84), IEEE Network (2016; DOI: 10.1109/MNET.2016.7764272) Journal of Computer and System Sciences (2017; DOI: 10.1016/j.jcss.2016.09.001), Multimedia Tools and Applications (2017; DOI: 10.1007/s11042-016-4081-z), Pervasive and Mobile Computing (2016; DOI: 10.1016/j.pmcj.2016.10.003), etc. His research has been cited in reports published by Australian Government agencies, Australian Government House of Representatives, United Nations Congress, United Nations Office on Drugs and Crime, International Telecommunication Union, UK Home Office, US CRS Report for Congress, US National Institute of Standards and Technology, etc. He is the recipient of various awards including ESORICS 2015 Best Paper Award, Winning Team of the Germany's University of Erlangen-Nuremberg Digital Forensics Research Challenge 2015, and 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008. He is also a Fellow of the Australian Computer Society.

Mohd Taufik Abdullah obtained a PhD from University Putra Malaysia, Malaysia. He is a senior lecturer from Department of Computer Science and also member of Information Security Research Group, Universiti Putra Malaysia, Malaysia and Digital Forensics Investigation Research Laboratory, University College Dublin, Dublin, Ireland. He has served for many years in a variety of research and industrial positions. Other than PhD. in Security in Computing he holds many professional certificates such as Digital Etiquette Certification, ECSS, ENSA, CEH V8, and CHFI V8 His research interest is software engineering, security in computing and digital forensics.

Zaiton Muda received the B.Sc and M.Sc degrees in Computer Science from Universiti Kebangsaan Malaysia in 1984 and 1989 respectively. She joined Universiti Putra Malaysia in 1984 as a tutor in Computer Science. She is a senior lecturer in Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. She is the coordinator of External Education Unit, Faculty of Computer Science and Information Technology, Universiti Putra Malaysia. Her research interests include computer security and parallel computing.