University of Amsterdam

MSc Security and Network Engineering

Research Project 1

Microsoft Office Upload Center Cache Files in Forensic Investigations

Kotaiba Alachkar
kotaiba.alachkar@os3.nl

Rick van Gorp
rick.vangorp@os3.nl

February 21, 2018

Supervisor:
Yonne de Bruijn
Fox-IT

Abstract

Since Microsoft Office 2010, Microsoft Office Upload Center generates cache files when uploading files from Microsoft Office to Microsoft OneDrive. Currently, there is only speculation on the internet about what information those cache files hold. This leads to the research question: *In what way do the cache files produced by Microsoft Office Upload Center contribute to a forensic investigation?* In our research we focus on Microsoft Office 2016. Based on our own generated dataset containing FSF-, FSD- and CentralTable.accdb-files, we derived the global structure of the FSD-file, the more detailed structure of the FSF-file and the content of *CentralTable.accdb*. The purpose of the FSD-file is holding an Office document. *CentralTable.accdb* holds the metadata corresponding to this document. The FSF-file is the connecting file and links the metadata in *CentralTable.accdb* to the document in the FSD-file. Using our parser for *CentralTable.accdb* we determined the existence of a row history in table *MasterFile* for *CentralTable.accdb*. The parser also retrieves all metadata corresponding to the Office document, such as modified dates, author, remote location and e-mail. We determined ways to recover documents from the FSD-files automatically and manually. We concluded the availability of this data results in information, which can be used in a forensic investigation.

1 Introduction

Since Microsoft Office 2010, Microsoft Office Upload Center was implemented as a feature in Office suites [10]. This feature added the possibility to directly upload documents to SharePoint, OneDrive or any other file-hosting cloud platform that integrates with Microsoft Office. Currently, there is speculation on the internet about what information the cache files produced by Microsoft Office Upload Center hold. Brent Muir states in his presentation and on his LinkedIn blog that Microsoft Office documents are stored in the FSD cache files [13][14]. A user on Microsoft Technet mentions that the cache files contain the original documents, integrity verification related data and metadata about the documents [3][17]. However, there is no research that verifies these findings nor methods are published to retrieve the data. According to Microsoft and WindowsCentral, Microsoft Office has approximately 1.2 billion users worldwide [11] [5]. This might imply there is a higher probability Microsoft Office cache files will appear in a forensic investigation. The fact that little is known about the forensic value of those cache files and what data can be retrieved from them results in the research question: *In what way do the cache files produced by Microsoft Office Upload Center contribute to a forensic investigation?* In this research we will focus on Microsoft Office 2016.

2 Related work

At present, there are many researches for video and image forensics, which have enriched forensic investigations and made large numbers of achievements [7]. However, the study on digital forensics of electronic documents is the opposite less, especially on Microsoft Office cache files. After intensive research on related works, we found one research paper about digital forensics of Microsoft Office 2007–2013 documents to prevent covert communication. This research was done by a group of researchers from Nanjing University of Information Science & Technology, Nanjing, China [7]. They proposed nine forensic methods and an integrated forensic tool for the Office Open Extensible Markup Language (OOXML) format documents on the basis of researching the potential information hiding methods. Their proposed forensic methods and tool covered three categories; document structure, document content and document format . In 2015 an Australian technology company called Nuix identified, in their presentation about cloud hosted data in digital forensics, the table structures and parts of their contents in *CentralTable.accdb*. They described a general use of every table and described the way dates are stored [16].

3 Analysis methods

In order to perform analysis on a dataset, an environment was set up and a dataset of FSF-, FSD- and CentralTable.accdb-files was generated. These files can appear in five different states: queued, uploading, paused, failed and finished.

3.1 Environment set-up

We have setup the environment on one virtual machine running Windows 7 with the latest updates installed. This version of Windows was specifically chosen because statistics from NetMarketshare and StatCounter show that it's the most popular

operating system globally, with 45.07% market share [15] [18]. The most popular operating system was chosen, because the probability is higher the generated dataset covers the greatest amount of forensic investigations involving Microsoft Office cache files. The virtual machine runs Microsoft Office 2016 and has two users, which are used to generate the dataset. *Figure 1* shows the analysis environment in stack form.

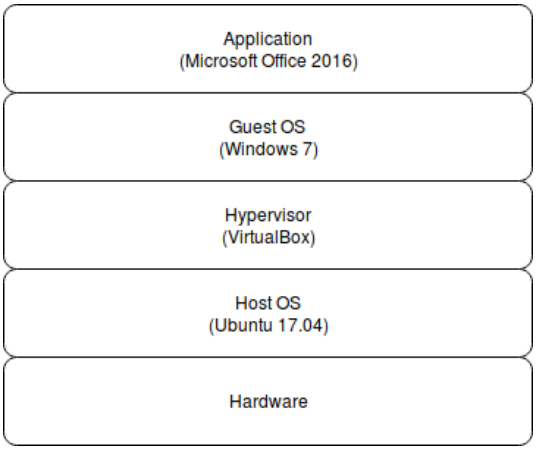

Figure 1: Environment setup in the form of a stack

Table 1 shows more specific characteristics of the application and operating system set-up.

Software	**Characteristic**
Microsoft Windows 7	Build 7601 Professional 64-bits edition
Microsoft Office 2016	Version 16.0.8827.2131 64-bits edition

Table 1: Application and Operating System characteristics of the virtual machine

We have created two Outlook accounts for access to Microsoft OneDrive. Those are listed in *Table 2*. The destination folder on OneDrive will be labeled *user1* and will contain the source documents generated as described in *Section 3.2*.

3.2 Dataset generation

The dataset consists of FSD-, FSF- and CentralTable.accdb-files generated by Microsoft Upload Center 2016. From a description of Upload Center by Microsoft is derived that these cache files can be in a queued, uploading, paused, failed or finished state [10]. The state diagram is shown in *Figure 2*.

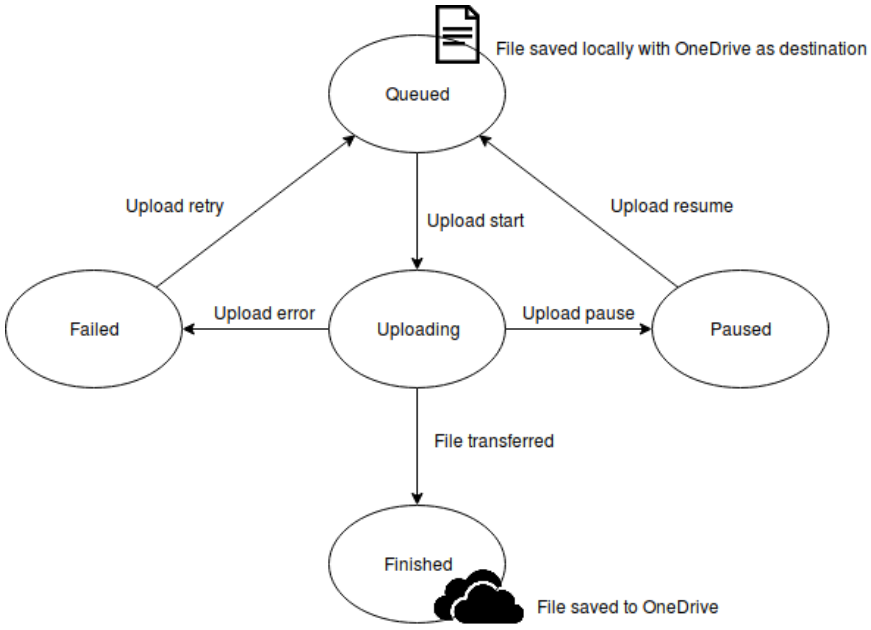

Figure 2: State diagram of the cache files in the upload process to OneDrive

For every user on the virtual machine the cache files are stored in a separate directory and grouped by state. The label of the directory was set to the corresponding state name selected from *Figure 2*. The states are reproduced by using the navigation in Microsoft Upload Center's Graphical User Interface (GUI). The queued state is reproduced clicking the pause button before submitting the document to OneDrive, then submit the document. The uploading state is reproduced by clicking the resume button. The paused state is reproduced by clicking the pause button during the uploading process. The finished state is reproduced by finishing the uploading process; a complete document is saved to OneDrive. To reach the paused state, the bandwidth is limited temporarily to *10Kbps* for the Virtual Machine to prevent the upload from finishing before the paused state can be reproduced. The failed state is reproduced by first uploading all document files to the *user1*-folder on OneDrive. When a user uploads a document with the same filename to the same Microsoft account and *user1*-folder, Microsoft Upload Center will show a failed state until the user overwrites the destination file.

In order to perform statistical analysis and determine differences between files, the dataset is generated under different circumstances. We use two users per virtual machine with a different timezone set. The characteristics of the environment per user are shown in *Table* 2.

Username	**Timezone**	**Upload to account**
User1	GMT+1	randkrp1@outlook.com
User2	GMT+2	randkrp2@outlook.com

Table 2: Characteristics of the environment per user

The input documents used to generate the dataset are Word docx-files, PowerPoint pptx-files and Excel xlsx-files [6]. For docx-files the document's content is either empty, contains one line of text or has a large size. For pptx-files the document's content is either empty, contains one slide with text or has multiple slides with text resulting in a large file size. For xlsx-files the document's content is either empty, contains one worksheet with text or has multiple worksheets filled with text resulting in a large file size. The sizes of the large source documents are listed in *Table 3*.

Document type	**Size (bytes)**
Word (.docx)	5,660,169
PowerPoint (.pptx)	2,067,228
Excel (.xlsx)	1,242,685

Table 3: Sizes of large source documents in the dataset

All cache files in the dataset exist with or without a PNG-image. The PNG-image is called Panama and is provided by the Wikimedia Foundation [20]. The generated dataset is published on GitHub [1].

The PNG-format was chosen, because statistics from W3Techs show the PNG-format is the most used format on websites [19]. Although the statistics do not directly apply to Office documents, the statistics give an indication on commonly used image file formats worldwide.

3.3 Desk research

In order to find information about file formats desk research will be performed. The information about file formats will be used for statistical analysis. We will look for existing knowledge about the file formats of the FSD-, FSF- or CentralTable.accdb-file. This also includes information about files that are embedded in one of the files.

3.4 Performing analysis

The generated dataset will be used to perform analysis on the structure of the cache files. One of the analysis methods is looking for fields, where the data describes the structure of sections in the cache files. This includes fields that indicate lengths of data sections, offsets of data sections, known file headers, number of files in a data section and checksums. The second method is to use the generated dataset and compare files on binary level using known conditions. In this method we will look at repetitive data sections, structures and whether data is available or not. The known conditions used are listed in *Table 2* and described in *Section 3.2*.

For the statistical analysis we will write a parser in Python that reads the Microsoft Access 2016 database format. The parser interprets tables and their contents and exports this data to a comma-separated value file (CSV-file). The analysis performed on the CSV-file consists of checking differences in row count per table per state and the availability of data in a table per column per state. This is done to determine what content of the Microsoft Access database CentralTable.accdb is available under what circumstances. To determine the differences in row count per table per state, the mean of the row count of all tables per state in the dataset will be determined and presented in a histogram. We will look for patterns in changes made to fields or row count per state that impact the availability of information.

4 Results

The generated dataset and statistical results from our analysis are published on GitHub [1]. The results show the differences in the file formats, what data is available in the files that is of forensic value and the implication of the availability of this data.

4.1 File descriptions

The comparison on byte level of cache files in the dataset, finding patterns in the structure of the cache files and desk research resulted in the file descriptions for the FSD-file and the FSF-file. For both files, the purpose of several sections are still unknown. The Australian technology company Nuix stated that *CentralTable.accdb* is a database file in the Microsoft Access Database format [16]. Execution of the dataset generation procedure showed the cache files are located in: `\Users\<USERNAME>\AppData\Local\Microsoft\Office\16.0\OfficeFileCache` for Microsoft Office 2016.

4.1.1 FSD-file

According to our dataset, the size of an FSD-file differs depending on the size of a source document. The size of an FSD-file will be larger when a large input document is used and smaller when a small input document is used. This implies that an FSD-file hold contents of a source document. Examples of file sizes of source documents compared to the file sizes of output FSD-files are shown in *Table 4*.

Size source document (bytes)	Size FSD-file (bytes)	State FSD-file
11,762	65,536	Queued
11,762	131,072	All, except queued
11,869	65,536	Queued
11,869	131,072	All, except queued
1,163,631	1,245,184	Queued
1,163,631	2,424,832	All, except queued
5,660,169	5,767,168	All, except failed

Table 4: Examples of differences between file sizes of input documents and FSD-files per state

Table 4 shows that for FSD-files where the source documents have different sizes, the size of FSD-files can be equal. This indicates that fixed sections or padding is used in the file. The comparison of all FSD-files in the dataset has led to information about the headers used for every FSD-file. In Microsoft Office 2016 the same section header indicators are used for every FSD-file. The first part of the file is shown in

Table 5. The purpose of the data in this part is currently unknown and has a fixed size of 8176 bytes.

Magic Header (16 bytes) - 0C 83 D2 91 AE 1B D4 4D AA 65 46 79 FB DA DD 7A
Data and additional 0-padding (8176 bytes)
Total Size: 8192 bytes (Fixed)

Table 5: The FSD-file format part 1

Table 6 shows the structure of the section after the first file part shown in *Table 5*. This section can occur multiple times depending on a multiplication by unknown variable β. The section starts with header `C4 F4 F7 F5 B1 7A 56 A4` and is divided in two sections. We will label this header as header A. The main section contains data with an unknown purpose and subsection 1 contains the data with more headers. Subsection 2 is identified in all FSD-files by header `4B BA 33 82 C3 15 C2 8B`. Depending on the data length set, this second section can also contain subsections. These sections start with `CF AA 69 49` and end with `79 05`. Those headers will be referred to as header K and header I respectively.

Main Section	
Header A (8 bytes) - C4 F4 F7 F5 B1 7A 56 A4	
Subsection 1	
Data about second section (Data1) (Unknown bytes)	
Subsection 2	
Header K (8 bytes) - 4B BA 33 82 C3 15 C2 8B	Data (Unknown bytes)
Optional Subsection of Section 2 (* α times, based on unknown α)	
Optional header I (8 bytes) - CF AA 69 49	Optional Data (Unknown bytes)
Optional end of header I (2 bytes) - 79 05	
Total Size: 8 bytes + Length Section 1 + Length Section 2	

Table 6: The FSD-file format part 2

Subsection 2 from *Table 6* could contain a document as headers related to ZIP-files were found in some of these sections. The input documents are created in Word, PowerPoint or Excel and are stored in the OOXML format. Microsoft describes on its website it uses the OOXML format for documents and store the OOXML-files in

a ZIP-archive [12]. The description of the ZIP-file headers is included in the ZIP-file specification and is published by PKWare Incorporated on October 1st 2014 [8]. The generated dataset shows that documents are separated per PK-section (headers in ZIP-files) or, in the case of large input documents, at random locations. The separated sections are placed in a header *I*. In case documents contain images, the images are separated into *I*-headers. Data recovery from FSD-files is described in *Section 4.2.2*.

4.1.2 CentralTable.accdb

According to the slidedeck of Australian Technology company Nuix and our dataset, the *CentralTable.accdb* file is a Microsoft Access database that contains information about all Office files in its five states that are used with OneDrive [16]. *CentralTable.accdb* holds metadata that provides information about the document and its authors. The file consists of the following tables:

1. ***MasterFile***: Contains metadata and holds details of files in OneDrive.
2. ***CacheProperties***: Details number of files still to upload & supported file types.
3. ***IncomingEvents***: Holds details of files stored locally.
4. ***OutgoingEvents***: Details ID & paths to files being uploaded to OneDrive.
5. ***ServerTarget***: Holds URL of OneDrive.

Tables 7, 8 and 9 list example entities of *MasterFile*, *CacheProperties*, and *ServerTarget* tables respectively. Metadata such as the last modified date, remote location of the document, author and state of the upload are stored in table *MasterFile*.

Entity Name	**Description**
FileEntryFileID	Unique GUID pointing to FSF-filename
DisplayUrl	Target Cloud platform URL where the file is uploaded to
UserName	File owner username
UserEmailAddress	User email account
DocumentUserTypedUrl	Direct URL to document where the file is uploaded to
DocumentLastModifiedTimeOnServer	File last modification time
DocumentLastModifiedBy	File last modified by which user
DocumentLastUploadTime	File last upload time
DocumentLastSuccessfulUploadTime	File last successfully upload time

Table 7: CentralTable.accdb: MasterFile table entities

Entity Name	**Description**
CacheID	File given cache GUID

Table 8: CentralTable.accdb: CacheProperties table entities

Entity Name	Description
ServerTargetID	Unique GUID of targeted server
ServerTargetAlternateUrl1	URL of targeted cloud platform (e.g. OneDrive, SharePoint, etc.)

Table 9: CentralTable.accdb: ServerTarget table entities

This information can be retrieved using Microsoft Access or using a parser that reads the file according to the file format. One available parser for Microsoft Access file formats is `mdbtools` and is created and published on GitHub by Brian Burns [4]. File "HACKING" in his repository describes the file format used for Microsoft Access 97, 2000 and 2002 databases.

Although the database can be read using Microsoft Access, fields where the column has a name related to time or date are unreadable. This is caused by the fact that Microsoft Upload Center stores these values in *binary* format instead of *datetime* format. This is derived from the output of `mdb-schema` from `mdbtools`, which shows all columns and their types in all tables. The *datetime* format is based on the *filetime* format, which is used by Microsoft [9]. The fact that Microsoft Upload Center stores the values in *binary* format is also described in a presentation of an Australian technology company called Nuix [16]. In our GitHub repository, we show a script that we used to parse CentralTable.accdb and shows the date/time data in human readable format [1].

4.1.3 FSF-file

The FSF-files from the dataset have been compared and we found that a large part of the file matches for every FSF-file. The file format of the FSF-file is shown in *Table 10*.

Header (20 bytes)	
α: Data Length (1 byte)	*Data*: Filename & Terminator(*0x05*) (α bytes)
Total Size: 21 bytes + Data Length	

Table 10: The FSF-file format with size indication

All FSF-files contain a reference to the FSD-file that is linked to it. The only reference available is the filename of the FSD-file. In our dataset, the hexadecimal length of the filename (*5D*) and the header is the same for all FSF-files. In scenarios where the FSF-file is unavailable and CentralTable.accdb and an FSD-file exist, the FSF-file can be created using the reference to the FSF-file in CentralTable.accdb and the filename of the FSD-file. The reference to the FSF-file is located in table *MasterFile* under column *FileEntryFileID* of *CentralTable.accdb*. In case an FSF-file has to be created and table *MasterFile* contains multiple rows, it is not derivable to what FSD-file a row corresponds, unless the name of the document is known. In that case the name of the document can be connected to the name of the document in the remote location metadata. However, when multiple entries exist containing the same document name, it is still not derivable what FSD-file corresponds to the entry

in *MasterFile*. If this is not derivable, then additional metadata can not be related to a document.

4.2 Availability of information

According to the differences between files in our dataset, it depends on the state the files are in whether information is available or not. The approach to retrieve data from FSD-files depends on the size of the source document and whether it contains images or not.

4.2.1 CentralTable

The statistical analysis of *CentralTable* results in information about the available data per state. According to our comparisons, rows were only added to table *MasterFile* when transitioning to different states. The mean of all *MasterFile* tables in the dataset containing the amount of rows per state is shown in *Figure 3*. The remaining tables contain the same amount of rows in different states.

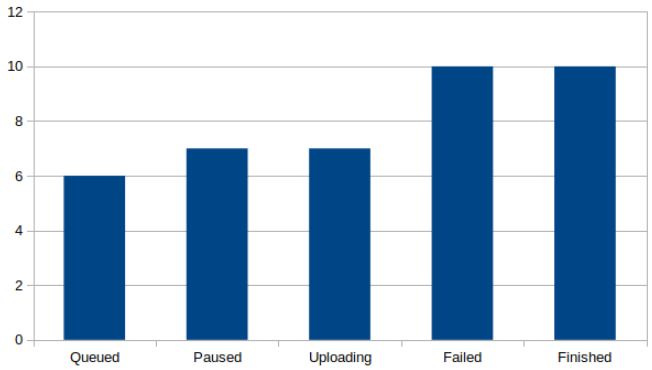

Figure 3: Mean amount of rows in *MasterFile* tables per state from all CentralTable.accdb-files

Figure 3 shows a pattern where rows are added when the uploading process reaches a new state. According to *Figure 3*, the amount of rows increases in table *MasterFile* when the transition from queued to uploading and the transition from uploading to failed occurs. During the state transitions the data in the tables change.

Generic changes during state transitions
During the state transitions, tables *MasterFile and CacheProperties* change the value in column *ColumnRevisionID*. This is the revision number of the column which can be used to derive in which order actions regarding specific files have occurred. According to our dataset, only those two tables change their values in rows while changing states.

***MasterFile*-table changes during state transitions**
Our dataset shows that during the transitions between the states, most changes are made in the MasterFile table. In the queued state there is no information available yet whether the file is saved to the server, what the last upload time was to the server and whether there were any errors during the upload or not. These values are

set in columns *FFileSavedToServer*, *DataLastUploadTime* and *DataCurrentUpload-Error* respectively. In the uploading state, the value *FFileSavedToServer* is set to 0, meaning that the file has not yet been saved to the server. In this state the values in *DataLastUploadTime* and *DataCurrentUploadError* have been set. The failed, paused and finished state contain extra values when being compared to the queued state. One of those values is available in the column *DocumentLastModifiedBy*, which contains information about the author of the file. Also during the paused and failed state the columns *DataFNewDataAvailable* and *DataFNewDataAvailable_AltContext1* have their values set to -1. In the finished state column *FFileSavedToServer* registers that the file is saved on the server by setting its value to -1. *DataFPendingupload* is also set to 0, such that Microsoft Upload Center does not attempt to upload the file to OneDrive. Also the columns *DataFNewDataAvailable* and *DataFNewDataAvailable_AltContext* have value 0 in during this state. There is no data available for upload or download.

CacheProperties-table changes during state transitions
The state transitions also have influence on table *CacheProperties*. The values in column *FPendingUploadVersion* change inconsistently. Using our dataset we could not derive an obvious pattern from it. Column *FglobalSyncEnabled* shows that it is set to 0 in the queued and paused state and that it is set to -1 in the uploading, failed and finished state. However, on some records it shows 0 for the failed state. Therefore also no obvious pattern could be derived from it.

4.2.2 Document recovery from cache files

The FSD-files contain the source document that was uploaded to Microsoft OneDrive. Data can be recovered from the FSD-files using three methods: Manually carve the data, use Microsoft Office to open the document and automatically carve the data. Manual and automatic carving of the document can be done by removing content unrelated to the ZIP-archive. In *Section 4.1.1* we describe that the sections in *Table 6* contain ZIP-archive data. Information about the location of the start and ending *header I* in combination with the location of the start of the ZIP-archive header results in the possibility to extract full small-sized documents without images. If the ZIP-archive header is fixed according to the ZIP-archive specification, parts of larger documents can be retrieved. This includes headers, footers, the document's content, the author, last modified date and references to image names. In our repository on GitHub we have included a script that exports (parts of) documents from FSD-files [1]. Another method of recovering documents from FSD-files is to use Microsoft Office. In case the FSF-, FSD- and CentralTable.accdb-file are available, the files are to be placed in the *OfficeCacheFiles* folder. If the value of column *FFileSavedToServer* is set to 0, the document can be opened through Microsoft Upload Center. In case only the FSD-file is available, the FSF-file and CentralTable.accdb have to be generated. In *Section 4.1.3* we describe the structure of the FSF-file. In our repository on GitHub we have included a script that generates the FSF-file based on the FSD-file's name [1]. Microsoft Office 2016 can be used to upload a file to OneDrive and generate CentralTable.accdb with records for one file. The columns *FFileSavedToServer* and *FileEntryFileID* should be set to 0 and to the GUID in the FSF-file's name respectively.

4.3 Retrieved data implication

When conducting a forensic investigation on a system, an investigator needs to reconstruct as many events and actions that took place on the system as are necessary to draw decisive conclusions. In our research, the retrieved data is divided into two parts, the recovered documents and metadata about these documents. Combining these two parts could assist in answering the questions: who, what, when, how, where and why.

Deriving the complete document with its original attributes from the FSD-file assists an investigation in many circumstances, where it can classify as additional evidence within the case. Furthermore, metadata from *CentralTable.accdb* can be used to provide the investigator with information on the files that are being investigated. It implicates the date and time of creation/modification, file's author, publisher's email address, the number of times the file has been modified, in addition to the targeted file-hosting cloud platform URL that is used to save the file. Fahad Alanzi and Andrew Jones describe in their report *The Value of MetaData in Digital Forensics* that in legal cases, the identification of relevant digital evidence is crucial for supporting the case and verification and examination of forms of existing legal arguments [2].

5 Discussion

The *CentralTable* parser exports rows that are not directly visible when using Microsoft Access. This can includes deleted rows or old records of rows. Although the information is relevant to forensic research, it may influence the results shown in *Figure 3* as possibly in some database copies the row could be unrecoverable. To prevent this in our research, we have generated a dataset ourselves that contains no corrupt CentralTable.accdb-files. Furthermore, *CentralTable* parser provides the capability to check whether the *CentralTable.accdb* has been altered or not. This results will observe any change(s) in the table entries, which indicates adjustment in the metadata regarding the document. However, any modification in the *CentralTable.accdb* tables entries will prevents the parser from exporting deleted/old records of rows.

The results regarding the FSD-files are limited to a global description of the file. This should be extended to a more detailed description, so that it will be possible to extract full large sized documents and documents containing images. There were indications that the FSD-file also contains references towards the remote location of the file, but this could not be researched properly due to the limited knowledge on the structure of the FSD-file. In the FSD-file the reference to the remote location is prefixed by a byte or short indicating the size of the remote location in the file. This part is located in the first header *A* at the section starting with header *K*.

In some cases, the information that can be retrieved from the Microsoft Office Cache files can not be used as evidence in a legal case as they may have been altered. We have presented a parser script to look at the history of rows in table *MasterFile* of the CentralTable. As the history is removed when records are altered in Microsoft Access, the parser reveals information about whether the table has been altered with manually. However, depending on the case and technical skills of a person the data still might be altered.

6 Conclusion

This research has shown what data can be retrieved from Microsoft Office cache files in its five states. The file descriptions show that the FSD-file is used to store the document, the FSF-file is used as a connecting point between the FSD-file and CentralTable.accdb and CentralTable.accdb is used to store all metadata regarding the upload process in Microsoft Upload Center.

Based on the file descriptions in *Section 4.1*, we have shown which data can be retrieved from CentralTable.accdb and how (parts of) documents can be retrieved from FSD-files. This results in a large amount of data and metadata, which can be used as an additional evidence in a forensic investigation.

Our main research question is: "*In what way do the cache files produced by Microsoft Office Upload Center contribute to a forensic investigation?*". Since the original document can be (partially) recovered and the corresponding metadata, such as data identifying the author, the remote location of the saved file and which time a file was uploaded and modified, is available in CentralTable.accdb, the cache files could have value in a forensic investigation. This is, however, limited to whether all cache files are available or not. If only the FSD-file is available, a document can be retrieved from it, but the additional metadata, identifying the person whom has uploaded the file and the destination of the file are not available. The retrieved document does contain metadata such as the creator of the file and the last modified date. Depending on what a forensic investigator wants to prove, this still can serve as useful information in a case.

7 Future work

To restore a complete document from the FSD-file using a script, more research into the FSD-file format is required. Currently, data such as offsets and data lengths are missing. These would indicate at what position in header *I* the data starts. This way, also data not containing ZIP-archive headers can be recovered, resulting in the recovery of a complete document.

This research is limited to Microsoft Office 2016 on Windows 7 while using OneDrive. In the future it could be expanded to include various Microsoft Office versions: 2010, 2013 and 365. Next to desktop versions the research should include mobile versions. The research should also be expanded to include more Operating Systems and file-hosting cloud platforms as the resulting FSD-files might have a different format or different metadata is available.

References

[1] Kotaiba Alachkar and Rick van Gorp. *Research: Office Cache Files*. URL: https://github.com/rickvg/office-cachefiles (visited on 02/01/2018).

[2] F. Alanazi and A. Jones. "The Value of Metadata in Digital Forensics". In: *2015 European Intelligence and Security Informatics Conference*. Sept. 2015, pp. 182–182. DOI: 10.1109/EISIC.2015.26. URL: http://ieeexplore.ieee.org.proxy.uba.uva.nl:2048/document/7379751/?reload=true.

[3] Adam Brown. *Office 365 OneDrive app - OfficeFileCache extremely large*. URL: https://www.experts-exchange.com/questions/28400526/Office-365-OneDrive-app-OfficeFileCache-extremely-large.html (visited on 12/28/2017).

[4] Brian Burns. *MDB Tools - Read Access databases on *nix*. URL: https://github.com/brianb/mdbtools (visited on 01/29/2018).

[5] John Callaham. *There are now 1.2 billion Office users and 60 million Office 365 commercial customers*. Mar. 2016. URL: https://www.windowscentral.com/there-are-now-12-billion-office-users-60-million-office-365-commercial-customers (visited on 01/17/2018).

[6] Microsoft Corporation. "Microsoft Office File Format Documentation Introduction". In: Revision 3.0. Aug. 2017, pp. 10, 16, 19. URL: http://interoperability.blob.core.windows.net/files/MS-OFFDI/[MS-OFFDI].pdf (visited on 01/15/2018).

[7] Z. Fu, X. Sun, and J. Xi. "Digital forensics of Microsoft Office 2007-2013 documents to prevent covert communication". In: *Journal of Communications and Networks* 17.5 (Oct. 2015), pp. 525–533. ISSN: 1229-2370. DOI: 10.1109/JCN.2015.000091.

[8] PKWare Inc. "APPNOTE.TXT - .ZIP File Format Specification". In: Oct. 2014. URL: https://pkware.cachefly.net/webdocs/casestudies/APPNOTE.TXT (visited on 02/01/2018).

[9] Microsoft. "FILETIME structure". In: *Windows Dev Center*. URL: https://msdn.microsoft.com/en-us/library/windows/desktop/ms724284(v=vs.85).aspx (visited on 01/19/2018).

[10] Microsoft. *Microsoft Office Upload Center*. 2018. URL: https://support.office.com/en-US/article/microsoft-office-upload-center-f08161d9-ab64-4486-af69-7cd30b34df71 (visited on 01/11/2018).

[11] Microsoft. "More than 1.2 billion people use Microsoft Office". In: URL: https://news.microsoft.com/bythenumbers/planet-office (visited on 01/17/2018).

[12] Microsoft. "Open XML Formats and file name extensions". In: URL: https://support.office.com/en-gb/article/open-xml-formats-and-file-name-extensions-5200d93c-3449-4380-8e11-31ef14555b18?omkt=en-GB&ui=en-US&rs=en-GB&ad=GB (visited on 02/01/2018).

[13] Brent Muir. *Windows 10 Forensics*. July 2015. URL: https://www.linkedin.com/pulse/windows-10-forensics-brent-muir (visited on 01/10/2018).

[14] Brent Muir. *Windows 10 Forensics: OS Evidentiary Artefacts*. July 2015. URL: https://www.slideshare.net/bsmuir/windows-10-forensics-os-evidentiary-artefacts (visited on 01/10/2018).

[15] NetMarketShare. *Operating System Share by Version*. URL: https://www.netmarketshare.com/operating-system-market-share.aspx?options=%7B%22filter%22%3A%7B%22%24and%22%3A%5B%7B%22deviceType%22%3A%7B%22%24in%22%3A%5B%22Desktop%2Flaptop%22%5D%7D%7D%5D%7D%2C%22dateLabel%22%3A%22Trend%22%2C%22attributes%22%3A%22share%22%2C%22group%22%3A%22platformVersion%22%2C%22sort%22%3A%7B%22share%22%3A-1%7D%2C%22id%22%3A%22platformsDesktopVersions%22%2C%22dateInterval%22%3A%22Monthly%22%2C%22dateStart%22%3A%222017-01%22%2C%22dateEnd%22%3A%222017-12%22%2C%22segments%22%3A%22-1000%22%7D (visited on 01/08/2018).

[16] Nuix. "Techno Security Forensics Investigations Conference". In: Cloud Hosted Data in Digital Forensics. Myrtle Beach, South Carolina, June 2014, pp. 20–21. URL: http://info.nuix.com/rs/nuix/images/cloud%20hosted%20data%20in%20digital%20forensics.pdf (visited on 01/21/2018).

[17] Microsoft Social Technet user Skye Legon. *Sharepoint workspace 2010, restore document in OfficeFileCache*. URL: https://social.technet.microsoft.com/Forums/en-US/dd36bf5e-187c-4a81-8aad-ae06c047f1f0/sharepoint-workspace-2010-restore-document-in-officefilecache?forum=officeitproprevious (visited on 12/28/2017).

[18] StatCounter. *Desktop Windows Version Market Share Worldwide*. URL: http://gs.statcounter.com/os-version-market-share/windows/desktop/worldwide (visited on 01/08/2018).

[19] W3Techs. "Usage of PNG for websites". In: Feb. 2018. URL: https://w3techs.com/technologies/details/im-png/all/all (visited on 01/15/2018).

[20] Derekitou via Wikimedia Commons. *Panama*. CC BY-SA 4.0. URL: https://commons.wikimedia.org/wiki/File:Panama_city.png (visited on 01/15/2018).