DOI: 10.17323/1998-0663.2019.4.39.48

Simulation of artefact detection in Viber and Telegram instant messengers in Windows operating systems

Alexander I. Borodin [a]
E-mail: aib-2004@yandex.ru

Roman R. Veynberg [a]
E-mail: veynberg@gmail.com

Dmitry V. Pisarev [b]
E-mail: d.pisarev@warwick.ac.uk

Oleg V. Litvishko [a]
E-mail: Litvishko.OV@rea.ru

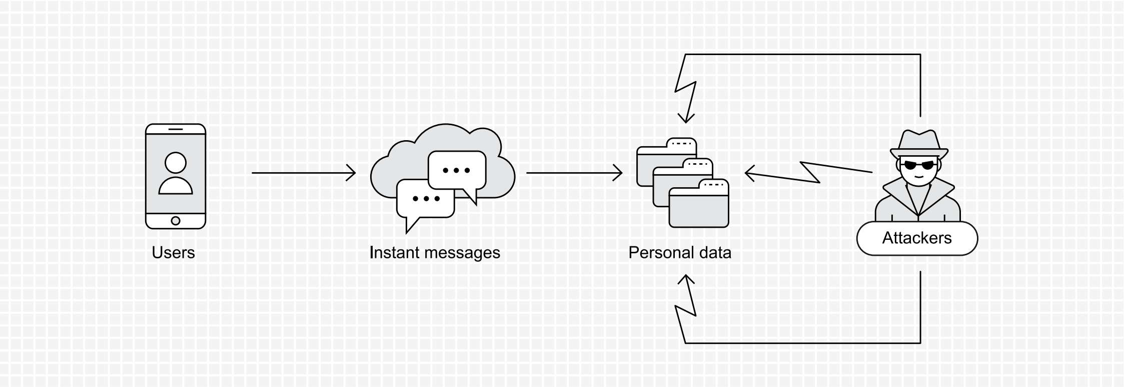
[a] Plekhanov Russian University of Economics
Address: 36, Stremyanny Lane, Moscow 117997, Russia

[b] University of Warwick
Address: Coventry CV4 7AL, United Kingdom

Abstract

Messengers are popular today on mobile devices and traditional computers. Starting as a small text messaging service, they have turned into effective communication channels for both private and corporate users, becoming more than just an SMS replacement. Users entrust to them a huge amount of information, such as a time-based map of activity, photos and other personal data. Messengers changed the way communication is done; they reduce the distance to the user and along with social networks become tools for fraud, spam or blackmail and terrorism. In this regard, it is vital to study instant messengers from a forensic point of view. This research explores and compares two popular messengers: Viber and Telegram, which is rapidly gaining popularity in the criminal world and the darknet as secure message tools. The main purpose of the research is to investigate and analyze potential artefacts remaining during the installation and use of instant messengers, as well as after their uninstallation. The authors have done several experiments to investigate the artefacts in different environments and provide clear explanation of the results. The experiments showed that even though Telegram is considered to be one of the most secure instant messengers, important and useful material on a hard drive and registry remain after complete uninstallation of the application. Exploring Viber artefacts showed up information that helps to restore the whole history of a communication. Moreover, the study confirmed that artefacts are still accessible in Windows after removal of the application.

Graphical abstract



Key words: instant messenger; computer science; simulation; exploit; artefact; Viber; Telegram.

Citation: Borodin A.I., Veynberg R.R., Pisarev D.V., Litvishko O.V. (2019) Simulation of artefact detection in Viber and Telegram instant messengers in Windows operating systems. *Business Informatics*, vol. 13, no 4, pp. 39–48. DOI: 10.17323/1998-0663.2019.4.39.48

Introduction

In recent years, instant messaging (IM) applications have gained in popularity because they are free of charge and easy to use. Nowadays IM is one of the most convenient ways to text messages, share files and videos, as well as make audio and visual calls. According to the research [1], worldwide IM user accounts are expected to grow to over 3.8 billion by year-end 2019.

The growing popularity of instant messengers relates also to various criminal activities such as fraud and terrorism [2]. They attract criminals by the opportunity the afford to simplify communication with victims or accomplices, as well as the availability of end-to-end encryption and other ways to secure or illuminate information that might be required by the authorities during an investigation.

However, despite the increased level of encryption and security, IM applications for Windows OS can provide to a potential researcher a lot of useful material. The artefacts can show information about the last date of launch, an SSID of the wireless network connected to the PC, outgoing connections, geolocations and other helpful information.

This research performs a forensic examination of popular Viber and Telegram applications by looking at the artefacts produced by IM applications. The interest in instant messengers grew with their popularity and IM applications have became the subject of various digital forensic studies.

Grispos et al. [3] tested user behavior from residual data in cloud-based synchronized applications. Communication between an attacker and victim were simulated, such as a file transfer and dialogs. The results of the study showed that artefacts remaining in the registry can link the criminal and the victim, such as traces of the file transfer between users and registry entries related to contact details. Moreover, fragments of the conversation can be recovered from memory dump.

Grispos et al. [3] also analysed residual data, simulating the conversation and file transfer between a suspect and a victim. Fragments of the conversation were found within the Windows 7 swap file, but the study of the mobile device did not provide much useful information. A broad list of artefacts that can be useful for forensic researchers such as references to the URLs and last access times was presented in the conclusion.

Cheng et al. [4] tested Windows Live Messenger installed on Windows 7. The results suggest that remaining artefacts allow one to restore the whole picture of a communication. Moreover, a user must be very competent to hide them.

Levendoski et al. [5] released information about the Yahoo messenger. Windows Vista and Windows 7 operating systems were used as platforms and comparisons conducted between OS artefacts remained after de-installation. The research showed that the structure of changes in the Windows 7 registry was modified inconsiderably compared to Windows XP.

Social network messengers have received attention from researchers because of their increased popularity. Al Mutawa et al. [6] studied Facebook chat based on web technology as a source of potential evidence for investigations. This article gives detailed information about possible artefacts, but their location depends on the browser and encoding. The study outlined a method for investigating Arabic string artefacts, but searching and converting them to readable view can take a lot of time to complete. However, the study is only limited to web-based Facebook chat.

Yasin and Abulaish [7] studied the Digsby IM aggregator to retrieve user sessions for use in investigations, despite attempts to hide information from a researcher. Results showed that they were similar to traditional IM applications. Despite the relatively recent date of the study, the messenger is not developing and supporting.

Karpisek et al. [8] studied an opportunity to decrypt traffic during WhatsApp communications and retrieve the details of calls. The current study presented a new approach to decryption of the information and found that calls can be decrypted. However, end-to-end encryption was changed by WhatsApp in 2016 and made the proposed method irrelevant.

The focus in research has recently shifted to social networks and cross-platform messengers.

Majeed et al. [9] studied three different applications: Facebook, Viber and Skype on the Windows 10 platform and the possibility to find artefacts. The result of the research showed that many artefacts are stored in one folder \AppData\Local\Packages\ for all the third-party applications. Moreover, for all applications they found artefacts saved as plain text files. The most important forensically relevant finding was common artefacts remaining for tested applications.

Dehghantanha et al. [10] studied Facebook and Skype messengers. The results indicated that artefacts could be recovered from a PC because of use of the Windows Store. IM applications installed using Windows Store leave elements valuable or critical to an investigation on the hard drive, in memory dumps and network captures.

To the best of the authors’ knowledge, the number of studies that are focused on comparison of secure messengers such as Telegram and another widely used IM application such Viber in the Windows environment is limited. Telegram was investigated by Cahyani et al. [11] and Carvey and Hull [12] as a tool for terrorist-related activities. Results of the study can be of great value for forensic analysts, but the research was strictly limited on mobile devices only. As a result, it is necessary to fill the gap and study artefacts that Telegram application leaves in the Windows environment compare to Viber — another well-known messenger.

1. Methods

This section gives information about tests that were provided with Viber and Telegram messengers. The experiment was performed on Windows 10 installed in a virtual machine environment. For the research we created: a windows user with administrative rights (“user_a”) and two new IM accounts (one for Viber and one for Telegram). Each of the IM apps was installed on a Windows installation. Interactions during the experiment were made using the author's personal account.

The artefacts were investigated through a series of research controlled experiments. All configuration changes were selected equally for both messengers. The detailed outline of the scenario and environment will be provided below.

1.1. Experimental environment

This study is based on the artefacts produced by two IM applications: Viber 6.9.6 and Telegram 1.1.23. The experiment was implemented on the following hardware platform: HP Z620 Workstation, CPU – Intel(R) Xeon(R) 2x E5-2660 2.20GHz, 16 Gb DIMM DDR3 (1866 MHz), 2TB Hard Drive using Ubuntu v.16.04.6 as the software operating system (OS).

Oracle Virtual Box (5.1.30 r118389 Qt5.6.3) containing Windows 10 Education (64bit, build 15063) was chosen as a platform for the experiment. The virtual workstation was configured with 4 GB RAM and 20 GB HDD space. Use of the virtual machine helped to make a considerable amount of snapshots and revert to a restore point quickly. As a result, this approach leaves researchers room for errors.

Registry and file data were collected using Regshot Portable v.1.9.0 which allow one to make a registry snapshot before and after a user activity and compare results.

The open source tool SQLite DB Browser v3.10.1-win64 was used for exploring details of databases. It helps to search, analyze and edit data and metadata in *.db files.

RegRipper v2.8 was used as a tool that helps to indicate user activity through analysis of the NTUSER.DAT file. The file provides very useful information (including key LastWrite times and data derived from binary and string values), indications of user actions. RegRipper userassist.pl plugin handles a translation UserAssist key which includes a 64-bit time stamp as well as a counter (referred to as a "run count") that appears to indicate how many times the user has interacted with the shell in the manner in which these values would be created or modified.

All software applications were installed with default setting and removed using standard Windows uninstaller.

1.2. Experiment procedure

The first step of the experiment was virtual machine creation, using Virtual Box containing Windows 10. The system was installed with default configuration and windows update service was disabled on the workstation for decreasing the number of artefacts not related to the experiment. Finally, we installed forensic tools and created a snapshot by Virtual Box. The snapshot was used as the "starting point" of the research for each IM application.

The second step was the IM app installation to collect and compare registry and file data using Regshot. The snapshots were performed on each IM application listed below in chronological order.

1. Immediately prior to installation of IM application;

2. Immediately after installation of IM application;

3. Before and after changing configurations such as:

- ✦ switch language to German;
- ✦ disable all automatic media downloading;
- ✦ enable auto startup;
- ✦ change default background;
- ✦ deactivation/logout from the IM application;

4. Immediately prior to removal of the IM application;

5. Immediately after removal of the IM application.

A communication between an attacker and a victim was emulated by sending a simple image file. A registry snapshot was made before and after the activity.

Local databases of the messengers were stored after the aforementioned activities for further research by SQLite DB Browse.

During the configuration changing experiments, many values of the registry and files were changed and modified. Table 1 contains the most significant changes for each type of operation.

All reports were stored in a plan text file and isolated for further investigation.

The final step of the research was analyzing reports and datasets. The search of required registry values was carried out by the standard application regedit.exe. Databases of IM applications were investigated using SQLite DB Browser for data and potential artefacts and messages stored on the computer. Files containing messages were transferred and we attempted to open them without access to the owner's account.

The users' and application activities were examined through file analysis NTUSER.DAT file by RegRipper v.2.8 application.

The experiment was repeated twice in order to be sure of consistent results.

2. Results

All reports and datasets were examined in this section. The finding for each application is provided below. Further details of registry keys and paths are listed in *Table 1.*

2.1. Telegram artefacts left after installation, file structure and database

The researcher was able to find the full path to the related IM application, installation date, version and user login who installed the application as highlighted by key (*Table 1*, no 1).

During the installation, folders were created that contain the database and file structure for the Telegram application.

Files of the IM can be found in the folder: \AppData\Roaming\Telegram Desktop\. The database of the Telegram is presented in the folder: %\tdata\D877F783D5D3EF8C. However, the database is stored as separated and encrypted files and not human-readable. Attempts to open the content of the database on using another Telegram account or read with SQLite DB were unsuccessful because the files are encrypted.

It is interesting to note that image or video files saved by a user during the communication can be found in the folder unencrypted and readable in the following folder: %UserName%\Downloads\Telegram Desktop.

During the installation, several folders and registry keys (*Table 1*, no 2) were created for interaction with the AI assistant Cortana.

2.2. Telegram configuration artefacts

Further evidence shows that language configuration added and modified the following file: \AppData\Roaming\Telegram Desktop \ tdata\settings0.

Recent changes are written to the log file AppData\Roaming\Telegram Desktop \log.txt. Unfortunately, the file is updating every time the application was restarted.

The following key (*Table 1*, no 3) was constantly modified after disabling the automatic download. Changing the startup mode of the application can be traced by detecting the following key: AppData\Roaming\Microsoft\Windows \StartMenu\Programs\Startup\ Telegram.lnk.

The application applies settings by modifying each dialog file in the folder after changing background: \tdata\D877F783D5D3EF8C\. However, deactivating the application has removed all message files from IM database and created the folder: tdata\D877F783D5D3EF8C1.

The last launch of Telegram can be found in the following registry key (*Table 1*, no 4). The value of the key LastAccessedTime is stored in hexadecimal or binary format and it is necessary to use a converter to translate them into a readable form. Keys value LoggedOnSAMUser

Table 1.

Registry and file information

No	Description
1	[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S–1–5–21–92284784–4191497677–2105538262–1001\Products\47A4A0DF1FC991646A19B825E007A0D6\ InstallProperties] “InstallLocation”=”C:\\Users\\user_a\\AppData\\Roaming\\Telegram Desktop\\” “InstallDate”=”20171017”
2	HKU\S–1–5–21–92284784–4191497677–2105538262–1001\Software\Microsoft\Windows\CurrentVersion\Search\Microsoft.Windows.Cortana_cw5n1h2txyewy\AppsConstraintIndex\LatestConstraintIndexFolder: “C:\Users\user_a\AppData\Local\Packages\Microsoft.Windows.Cortana_cw5n1h2txyewy\LocalState\ConstraintIndex\Apps_{8adcf8d1–d1f5–43e9–805d–af5466e37b69}”
3	HKU\S–1–5–21–92284784–4191497677–2105538262–1001\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD–ACE2–4F4F–9178–9926F41749EA}\Count\HRZR_PGYFRFFVBA
4	[HKCU\Software\Microsoft\Windows\CurrentVersion\Search\RecentApps\{DC6BD851–959F–45DA–BD7B–87FD4EBF9648}] “AppId”=”C:\\Users\\user_a\\AppData\\Roaming\\Telegram Desktop\\Telegram.exe” “LastAccessedTime”=hex(b):20,b5,3a,31,bc,55,d3,01 “LaunchCount”=dword:00000015
5	[HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Authentication\LogonUI\SessionData\1] “LoggedOnSAMUser”=”test_pc\\user_a” “LoggedOnUser”=” test_pc\\user_a”
6	[HU\S–1–5–21–92284784–4191497677–2105538262–1001\ Software\ Classes \ tg] “URL Protocol”=”” @=URL:Telegram Link
7	[HKU\S–1–5–21–92284784–4191497677–2105538262–1001\Software\Classes\tdesktop.tg\DefaultIcon] @=”\”C:\\Users\\user_a\\AppData\\Roaming\\Telegram Desktop\\Telegram.exe,1\””
8	[HKU\S–1–5–21–92284784–4191497677–2105538262–1001\Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store] “C:\\Users\\user_a\\AppData\\Roaming\\Telegram Desktop\\unins000.exe”=hex:53,
9	HU\S–1–5–21–92284784–4191497677–2105538262–1001\ Software\ Classes \tdesktop.tg\DefaultIcon] @=”\”C:\\Users\\user_a\\AppData\\Roaming\\Telegram Desktop\\Telegram.exe,1\””
10	HLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Installer\UserData\S–1–5–21–92284784–4191497677–2105538262–1001\Products\47A4A0DF1FC991646A19B825E007A0D6\InstallProperties “InstallDate”=”20171027” “DisplayVersion”=”6.9.6.16” “DisplayName”=”Viber”
11	HKU\S–1–5–21–92284784–4191497677–2105538262–1001\Software\Microsoft\Windows\CurrentVersion\Search\Microsoft.Windows.Cortana_cw5n1h2txyewy\AppsConstraintIndex\LatestConstraintIndexFolder: «C:\Users\user_a\AppData\Local\Packages\Microsoft. Windows.Cortana_cw5n1h2txyewy\LocalState\ConstraintIndex\Apps_{87f4a862–0157–4db6–927a–464474baefcd}»
12	HKU\S–1–5–21–92284784–4191497677–2105538262–1001\Software\Microsoft\Windows\CurrentVersion\Explorer\SessionInfo\1\ApplicationViewManagement\W32:00000000001104F8
13	HKU\S–1–5–21–92284784–4191497677–2105538262–1001\Software\ Microsoft\Windows\CurrentVersion\Run\Viber: “”C:\Users\user_a\AppData\Local\Viber\Viber.exe” StartMinimized”
14	%User%\AppData\Roaming\ViberPC\%phone№%\Backgrounds\3\10000403.jpg
15	[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Uninstall\{cbbefdcb–c7ee–4854–a1bc–c96d22b9d367}] “DisplayVersion”=”6.9.6.16” “Publisher”=”Viber Media Inc.”
16	[HKEY_CLASSES_ROOT\Local Settings\Software\Microsoft\Windows\CurrentVersion\AppModel\SystemAppData\Microsoft.Windows.Photos_8wekyb3d8bbwe\PersistedStorageItemTable\ManagedByApp\{1653CDC0–15E2–4885–A58A–E21C803F0BAA}] “Metadata”=”C:\\Users\\user_a\\AppData\\Roaming\\ViberPC\\447718905468\\Thumbnails\\thumb–c06ce8612230f51f80144f7077213b68.png” “LastUpdatedTime”=hex:04,a2,9e,1d,92,4d,d3,01
17	[HKEY_CLASSES_ROOT\Local Settings\Software\Microsoft\Windows\Shell\MuiCache] “C:\\Users\\user_a\\AppData\\Local\\Viber\\Viber.exe.FriendlyAppName”=”Viber”
18	[HKCU\Software\Microsoft\Windows\CurrentVersion\Search\RecentApps\{33D886D6–91BA–419C–A151–C9D0D31EEE34}] “LastAccessedTime”=hex(b):e0,b8,bb,3d,43,50,d3,01 “AppId”=”C:\\Users\\user_a\\AppData\\Local\\Viber\\Viber.exe”
19	Uninstall: Software\Microsoft\Windows\CurrentVersion\Uninstall Fri Oct 27 14:48:48 2017 (UTC) Viber v.6.9.6.16

and LoggedOnUser in the following registry key (*Table 1*, no 5) help to understand the details of application users.

2.3. Artefacts remaining after removal of Telegram

Despite difficulties with reading files containing messages and their removal after deactivating or uninstalling the application, many artefacts remain in the registry that researchers can find for understanding the directory structure and menu link as it is presented in the keys (Table 1, no 6, 7 and 8).

Some keys (*Table 1*, no 9) provide complete information about the installation path of the program despite removal.

It is interesting to note that a huge amount of useful information can be extracted from the NTUSER.DAT file. For example "most recently used" list, or "MRU report" from RegRipper shows the time of the last Telegram database record.

2.4. Viber artefacts left after installation, file structure and database

Examination of the registry determined that the following registry key (*Table 1*, no 10) was created by Windows specifying installation date and Viber version. The installer created different changes in the file structure and registry during the installation process, for example, interaction keys for AI Cortana (*Table 1*, no 11).

✦ The following folders contain most files of the Viber application:

✦ Database: %user%\AppData\Roaming\ViberPC\;

✦ Application: %user%\AppData\Local\Viber\;

QML caching: %user%\AppData\Local\Viber Media S.a r.l.

A folder named as a user phone number that contained the main database viber.db was created after installation and activation of the Viber.

The database is unencrypted and most messages and information are readable through the SQLite DB Browser. Nevertheless, messages are presented in an unstructured form, but the contacts table gives full information about names and phone numbers.

The messages can be opened in a user-friendly form by simply replacing the viber.db file on the PC with the installed Viber application. In this case, there is no way to respond and receive messages on behalf of the owner of the database, but a researcher has full access to the messages history.

2.5. Viber configuration changes artefacts

Changes of the language settings application modify the following file: %user%\AppData\Roaming\ViberPC\%phone% \QmlWebCache\data8\7\1tt95mf7.d.

Further evidence shows that all automatic media downloads have been disabled. This can be seen in the presence of a new registry key (Table 1, no 12). This value (*Table 1*, no 13) shows that a startup mode has been changed for the Viber application.

Changes of the default background for the application can be traced by adding a new file presented in *Table 1*, no 14. All content was deleted in the database folder \ViberPC\%phone№% after deactivation of the Viber account. However, the database file config.db containing settings of the application was available in the folder. The researcher can retrieve information about the phone number and previous IM account from the "Accounts" table of the config.db file using SQLite DB Browser.

2.6. Artefacts remaining after removal of Viber

The application left the key (*Table 1*, no 15) in the registry that provides information about de-installation of the program from Windows. The artefacts remaining in the registry allow us to restore a folder structure, location and history of file transfers via messenger (*Table 1*, no 16).

The HKEY_CLASSES_ROOT\viber branch record values were added by Viber installation and are still available in the system after removal. These keys (*Table 1*, no 17 and 18) specifying the path, last accessed time to the application remain in the registry after the application has been uninstalled. Moreover, Windows created a record in the NTUSER.DAT file that indicated the date and time when the IM application was uninstalled (*Table 1*, no 19).

3. Discussion

This study investigated Windows 10 for a location of Telegram and Viber artefacts. The results indicated that use of the messenger applications leaves registry artefacts which contain material that might be useful for investigation.

Even though Telegram is considered to be one of the most secure instant messengers, this study shows that useful material such as time-based artefacts and traces of user application on a hard drive and registry have remained.

Exploring Viber artefacts showed that the researcher is able to find very interesting information that helps to restore the whole history of a communication. Moreover, the study confirmed that artefacts are still available in Windows after removal of the application. Experts can unveil information about a user who installed the software and the account which used it.

In the future, research will include exploring system processes of the IM applications in Windows 10 for further deep forensic analysis of the IM behavior and cooperation with other system applications and software.

Conclusion

Messengers are popular today on mobile devices and traditional computers. Starting as a small text messaging service, they have turned into effective communication channels for both private and corporate users, becoming more than just an SMS replacement.

Users entrust to them a huge amount of information, such as a time-based map of activity, photos and other personal data. Messengers have changed the way communication is done; they reduce the distance to the user and along with social networks become tools for fraud, spam or blackmail and terrorism.

In this regard, it is vital to study IM from a forensic point of view. This research explores and compares two popular messengers: Viber and Telegram, which is rapidly gaining in popularity in the criminal world and the darknet as secure message tools. The main purpose of the research is to investigate and analyze potential artefacts remaining during the installation and use of instant messengers, as well as after their uninstallation.

The authors have done several experiments to investigate the artefacts in different environments, with clear explanation of the results. The experiments showed that even though Telegram is considered to be one of the most secure instant messengers, important useful material on a hard drive and registry have remained after complete uninstallation of the application.

Exploring Viber artefacts showed up information that helps to restore the whole history of communication. Moreover, the study confirmed that artefacts are still available in Windows after removal of the application. ■

Acknowledgments

This research was funded within the internal research of the Plekhanov Russian University of Economics under the name "Development of methods to forecast prices for financial instruments based on neural networks."

References

1. The Radicati Group (2015) *Instant messaging market*, 2015–2019. Available at: http://www.radicati.com/wp/wp-content/uploads/2015/02/Instant-Messaging-Market-2014-2018-Executive-Summary.pdf (accessed 25 September 2018).
2. Roberts J.J. (2017) *Here are the most popular apps for secure messages*. Available at: http://fortune.com/2017/01/17/most-popular-secure-apps/ (accessed 27 September 2018).
3. Grispos G., Glisson W.B., Pardue H., Dickson M. (2014) Identifying user behavior from residual data in cloud-based synchronized apps. Proceedings of the *Conference on Information Systems Applied Research (CONISAR 2014), Baltimore, MD, USA, 6–9 November 2014*, no 3310. Available at: http://proc.conisar.org/2014/pdf/3310.pdf (accessed 27 September 2018).
4. Cheng L., van Dongen B.F., van der Aalst W.M.P. (2019) Scalable discovery of hybrid process models in a cloud computing environment. *IEEE Transactions on Services Computing* (Early Access Article). DOI: 10.1109/TSC.2019.2906203.
5. Levendoski M., Datar T., Rogers M. (2014) Yahoo! Messenger forensics on Windows Vista and Windows 7. *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 88, pp. 172–179. DOI: 10.1007/978-3-642-35515-8_14.
6. Al Mutawa N., Al Awadhi I., Baggili I., Marrington A. (2011) Forensic artefacts of Facebook's instant messaging service. Proceedings of the *6th International Conference for Internet Technology and Secured Transactions (ICITST 2011), Abu Dhabi, United Arab Emirates, 11–14 December 2011*, pp. 771–776.
7. Yasin M., Abulaish M. (2014) DigLA – A Digsby log analysis tool to identify forensic artefacts, *Digital Investigation*, vol. 9, no 3–4, pp. 222–234. DOI: 10.1016/j.diin.2012.11.003.
8. Karpisek F., Baggili I., Breitinger F. (2015) WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages. *Digital Investigation*, vol. 15, pp. 110–118. DOI: 10.1016/j.diin.2015.09.002.
9. Majeed A., Zia H., Imran R., Saleem S. (2015) Forensic analysis of three social media apps in Windows 10. Proceedings of the *2015 12th International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies (HONET), Islamabad, Pakistan, 21–23 December 2015*, pp. 1–5. DOI: 10.1109/HONET.2015.7395419.
10. Dehghantanha A., Choo K.-K.R., Muda Z. (2016) Windows instant messaging app forensics: Facebook and Skype as case studies. *PloS One*, vol. 11, no 3, pp. e0150300. DOI: 10.1371/journal.pone.0150300.
11. Cahyani N.D.W., Ab Rahman N.H., Glisson W.B., Choo K.-K.R. (2017) The role of mobile forensics in terrorism investigations involving the use of cloud storage service and communication apps. *Mobile Networks and Applications*, vol. 22, no 2, pp. 240–254. DOI: 10.1007/s11036-016-0791-8.

12 Carvey H., Hull D. (2014) *Windows registry forensics*. Elsevier. DOI: 10.1016/C2009-0-63856-3.

About the authors

Alexander I. Borodin

Dr. Sci. (Econ.);

Professor, Department of Financial Management, Plekhanov Russian University of Economics,
36, Stremyanny Lane, Moscow 117997, Russia;

E-mail: aib-2004@yandex.ru

ORCID: 0000-0002-2872-1008

Roman R. Veynberg

Cand. Sci. (Econ.);

Associate Professor, Department of Informatics, Plekhanov Russian University of Economics,
36, Stremyanny Lane, Moscow 117997, Russia;

E-mail: veynberg@gmail.com

ORCID: 0000-0001-8021-5738

Dmitry V. Pisarev

Master of Science in Cyber Security and Management;

University of Warwick, Coventry CV4 7AL, United Kingdom;

E-mail: d.pisarev@warwick.ac.uk

Oleg V. Litvishko

Cand. Sci. (Econ.);

Associate Professor, Department of Financial Management, Plekhanov Russian University of Economics,
36, Stremyanny Lane, Moscow 117997, Russia;

E-mail: Litvishko.OV@rea.ru