RESEARCH: TECHNICAL REPORT

MemTri: Memory Forensics Triage Tool (Extended Abstract)

Antonis Michalas* and Rohan Murray†
Cyber Security Group, Department of Computer Science, University of Westminster, Cavendish Street, W1W 6UW London, UK

*Correspondence:
a.michalas@westminster.ac.uk
Full list of author information is available at the end of the article
†Equal contributor

Abstract

This work explores the development of MemTri. A memory forensics triage tool that can assess the likelihood of criminal activity in a memory image, based on evidence data artefacts generated by several applications. Fictitious illegal suspect activity scenarios were performed on virtual machines to generate 60 test memory images for input into MemTri. Four categories of applications (i.e. Internet Browsers, Instant Messengers, FTP Client and Document Processors) are examined for data artefacts located through the use of regular expressions. These identified data artefacts are then analysed using a Bayesian Network, to assess the likelihood that a seized memory image contained evidence of illegal activity. Currently, MemTri is under development and this paper introduces only the basic concept as well as the components that the application is built on. A complete description of MemTri coupled with extensive experimental results is expected to be published in the first semester of 2017.

Keywords: Digital Forensics; Triage; Cyber Crime; Digital Evidence; Random Access Memory

Digital Forensics; Triage; Cyber Crime; Digital Evidence; Random Access Memory

1 Introduction

With the current advances in digital forensics, it is becoming more common for law enforcement personnel to encounter digital devices as part of seized evidence to be examined. This list of digital devices include various machines with different architectures and specifications (e.g. desktops, laptops, mobile phones, tablets etc). The growing influx of seized digital devices has generated a backlog of court case evidence to be forensically examined [1]. A proposed solution for alleviating this evidence backlog is to develop triage execution tools that incorporate data mining techniques [2]. The main aim of such triage tools is to quickly assess whether a digital device contains relevant case evidence or not, and how much priority should be placed on fully analyzing the device.

Even though there are many crime classification triage tools for disk and mobile forensics, there is a clear lack of any such similar triage tool for memory forensics. The absence of such memory forensics tools is considered as an obstacle that prevents investigators from thoroughly analyzing digital devices. This is mainly due to the fact that various research has shown that memory can contain critical evidence such as internet browsing data, network traffic, malware, passwords, cryptographic

keys and decrypted content, some of which may never be stored to disk [3, 4]. A possible reason for the apparent low research in developing crime classification triage tools for memory forensics is due to the complexity in analyzing operating system (OS) memory structures, which is still a fairly adolescent area of research. The open-source tools Volatility [5] and Rekall [6] have aided in simplifying the analysis of such OS memory structures by incorporating the academic research done by various authors in reverse engineering these structures. In this paper, we leverage from the various research incorporated into the Volatility framework [5] and we propose MemTri – a memory triage application that analyzes OS memory structures. It was simply decided to utilize the Volatility framework for this project, due to it being the most widely utilized and tested memory analysis tool in the academic community. Another factor that may have contributed to the apparent research in developing crime classification triage tools for memory forensics, is due to the fact that acquiring memory requires careful planning and skill in order to collect a 'forensically sound' [7] memory image, which in-turn has led to the slow adoption of performing memory image acquisitions by law enforcement departments.

Another challenge in memory forensics is that, if the user terminates the application process used to perform an illegal activity then the freed virtual address space is often quickly overwritten by other activity within the operating system. However, Garfinkel et al. [8] showed that portions of unallocated memory can remain unchanged for up to 14 days – even when the system is actively being utilized. Therefore, since some data artefacts may not be overwritten in unallocated memory space by the OS, it is still possible to extract such data artefacts for memory analysis, similar to carving for files in a file system. MemTri is developed with two modes of operation, namely *normal* and *scan* mode, that gives valuable insights for the best methods to process evidence artefacts in a volatile memory environment.

MemTri offers a way to quantitatively measure the likelihood that a specific criminal offence was committed. The results are based on an extended analysis of test evidence data artefacts that were found in Random Access Memory (RAM), and help us to determine the priority that should be placed on fully examining a set of memory images. Towards this direction, we build MemTri on top of a Bayesian Network and the Volatility Framework. Furthermore, in order to successfully achieve our goal, we developed a certain set of algorithms through which we can assess the effectiveness of locating data artefacts in RAM, after the process that generated the artefact has terminated. This is of paramount importance since a forensics analysis is likely to be held after the termination of the application that used to create private information of the "corrupted" parties.

1.1 Organization

The rest of this position paper is organized as follows. In Section 2, we describe relevant related work regarding existing triage solutions in the field of digital forensics. In Section 3, we introduce the system model, as well as the preliminaries that MemTri is built on. Finally, in Section 4 we conclude the paper by providing a set of future directions.

2 Related Work

In this section we review the most important works that have been published in the field of digital forensics and we specifically focus on existing triage solutions.

According to [9], the definition of triage in regards to digital forensics is "a process in which digital evidence is ranked in terms of importance or priority". There have been proposed various methodologies for developing triage tools for the main branches of digital forensics, i.e. disk forensics, memory forensics, mobile phone forensics and network forensics.

Bogen et al. [10] developed Redeye – a disk document triage tool. Redeye, utilizes a corpus-based term weighting scheme (TF-ICF) and semi-supervised machine learning to triage identification of documents that relate to a specific case. The corpus-based term weighting scheme mainly assesses the similarity between documents based on the frequency of a word and its position in relation to other keywords. Document analysts are then able to identify documents that are most likely similar/related to certain key documents they have marked as relevant to an investigation. The system further monitors the tags and comments made by analysts in order to 'learn' which type of documents are of particular importance to an investigation. Moreover, Redeye successfully aided to significantly reduce the completion time of a forensic analysis. Even though Redeye focuses on a different field of forensics than MemTri, it demonstrates the ability of supervised machine learning techniques (similarly utilized by MemTri's Bayesian Network) to successfully triage tasks in an investigation.

Li et al. [11] developed a memory triage tool that uses fuzzy hashing to intuitively identify malware by detecting common pieces of malicious code found within a process. Authors, identified a limitation with the asymmetric distance computation of existing fuzzy hashing algorithms and assess four key insights, based on precision and recall, which can improve the fuzzy hashing algorithms' performance. The improvement of such fuzzy hashing algorithms aids investigators to more quickly and accurately determine whether a machine has been affected by malware before attempting a full investigation. MemTri's performance is similarly tested using such performance measures which can reveal key areas of triage-related improvements.

Walls et al. [12] developed DEC0DE, a mobile phone forensics triage tool. DEC0DE, uses block hash filtering (BHF), Viterbi's algorithm and Decision Tree inference. During BHF, similar byte streams between mobile phone models, which most likely will contain operating system data that is not relevant to the investigator, are removed. Therefore, the mobile data that remains after BHF completes is likely to be user' data such as call logs and address book information. This data is further processed using Viterbi's algorithm and Decision Tree inference to improve the recall and precision of the filtered data. Authors, highlighted that mobile phone forensics triage can help to gather key information upfront for use in suspect interviews, before the full analysis is performed which can take months to complete due to backlog of devices to be analyzed. Similarly, MemTri provides the digital investigator with a quick assessment of key evidence artefacts found in a memory image which can then be used as persuasive evidence in a suspect interview.

In [13], authors developed a network triage application that uses a client-server model in order to search multiple client machines for evidence. An automated network triage (ANT) server that hosts various services is used to configure and boot

PXE enabled clients. When the client machine boots, a batch script is simply ran to search for keywords, patterns and file hashes on the client machine's disk. This network forensics triage tool can essentially help to locate a machine within a network that was most likely involved in the crime being investigated and thus the identified machine can be seized/prioritised for further investigation. Without such a triage tools an investigator would have to analyze all the machines in the network individually which is impractical/time-consuming.

The aforementioned works by the various authors in the different fields of digital forensics shows that triage tools have proven to be a valuable solution to the 'data volume challenge' [2]. Generally, these triage solutions offer a quick way of narrowing down the devices to those that contain critical data before a full digital forensics analysis is performed. Similarly, our work contributes to the area of digital forensics triage tools with an emphasis on *memory forensics*.

3 System model and Preliminaries

In this section we describe the system model, as well as some terminology and basic concepts that will be used in the rest of the paper.

Suspect Machine (SM): For the needs of our work, four types of software applications, namely Internet Browser, Instant Messenger, Document Processor and FTP Client, are examined. Therefore, we created several Windows 7 virtual machine instances where we pre-installed the various software applications listed in Table 1. These applications are also referred to as the *'target applications'*. Each virtual machine is then shutdown and a copy of the virtual machine files is made. These copied files are referred to as the base virtual machine image which is used as the starting point for performing the suspect activity analysis.

Type	Application(s)
Internet Browser	Tor, Chrome
Instant Messenger	Wickr, Skype
Document Processor	Windows Notepad, Libre Writer
FTP Client	Filezilla

Table 1 List of applications installed by type

Evidence Search Engine (ESE): The Evidence Search Engine component is responsible for extracting evidence artefacts from the 'suspect' memory image and translating them into features that can be used by a bayesian network analyser. This approach was mainly inspired by [4, 14, 15] which showed that intuitive evidence artefacts can be retrieved by simply searching for ASCII/Unicode data patterns generated by specific applications. This regular expressions approach is also flexible in that it can locate evidence artefacts in a memory image regardless of the OS environment in which the artefacts were generated. Additionally, regular expressions can be executed fairly quickly to locate evidence within large datasets. This intuitiveness, flexibility and speed offered by regular expression evidence searching methods, are essential traits for the development of an effective digital forensics triage tool.

Bayesian Network Analyser (BNA): MemTri uses a Bayesian Network to analyze the evidence found by the ESE. An output rating is then produced that can be used to rank a set of suspect memory images, based on the likelihood level of criminal activity. We decided to build the Bayesian Network based on the model proposed in [16], since it is simple to interpret and has proven successful in correctly analyzing real-life criminal investigations. Comparative studies have also analyzed that Bayesian approaches to developing digital forensics triage tools, on average have the best accuracy performance [17] (88.5%) compared to other supervised machine learning (SML) techniques such as Support Vector Machines, Decision Trees and K-Nearest Neighbour. This combination of accuracy and ease of interpretation supported by Bayesian Network approaches, are favourable traits when seeking to triage a criminal investigation. Additionally, Bayesian Networks handles missing evidence most eloquently, since it is naturally incorporated into its design. Handling missing evidence is of paramount importance for successful forensics investigations since evidence can often be missing due to it being destroyed or not yet discovered.

Bayesian Network: In digital forensics triage, law enforcement personnel often has to make quick decisions based on evidence found on a crime scene. Thus, a soundly built Bayesian Network can efficiently aid in determining the best course of action to be taken based on the evidence found. The Bayesian Network model is an acyclic graph that encodes the conditional independence relationship of the graph nodes. Figure 1 illustrates a diagram of the Bayesian Network we used for the needs of our work. This Bayesian Network has been set up by using the Netica [18] software.

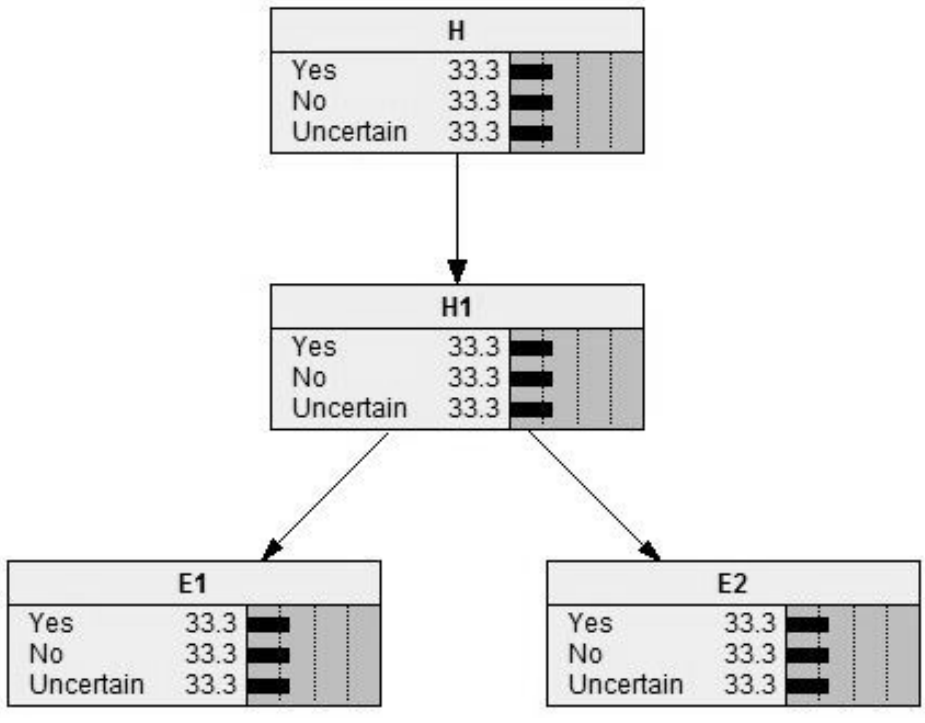

Figure 1 Bayesian Network Example

	H1		
H	**Yes**	**No**	**Uncertain**
Yes	60	35	5
No	35	60	5
Uncertain	5	5	90

Table 2 Likelihood Joint Probability for $P(H1 \mid H)$

	E1		
H1	**Yes**	**No**	**Uncertain**
Yes	85	15	0
No	15	85	0
Uncertain	0	0	100

Table 3 Likelihood Joint Probability for $P(E1 \mid H1)$

	E2		
H1	**Yes**	**No**	**Uncertain**
Yes	75	25	0
No	25	75	0
Uncertain	0	0	100

Table 4 Likelihood Joint Probability for $P(E2 \mid H1)$

This is the general structure of the Bayesian Network that is used throughout the development of MemTri. The top-most nodes prefixed with H are referred to as hypothesis nodes while the lowest level nodes prefixed with E are referred to as the evidence nodes. To make this example more intuitive the nodes have been assigned specific meanings as follows:

- H: The suspect employee's computer was used to send confidential company files to a third party using FTP.
- H1: An FTP connection was established between employee machine and a third party.
- E1: Network Logs show a TCP connection on port 21 between employee machine and a third party.
- E2: FTP "Transfer OK" response packet found between employee machine and a third party in router cache.

The probability values shown in Figure 1 is the Prior Probability values of the Bayesian Network. The following joint probability tables 2, 3, 4 represent the likelihood probability values that are associated with the given Bayesian Network.

These probability values are usually set based on the data gathered from experts in the field of the investigation. From the aforementioned tables we see that a node has three states 'Yes', 'No' or 'Uncertain'. An important point to note is that the probabilities in the Bayesian Network must add up to 100%.

Now, let us assume that an investigator wants to determine the probability that the suspect employee sent confidential files to a third party given that he has observed that there was a FTP 'Transfer OK' packet found. In other words, the investigator wants to determine $P(H = Y \mid E2 = Y)$. This hypothesis can be examined by performing Bayesian Inference. Statistically inferring a conclusion for this hypothesis can be useful in aiding the investigator to confidently decide whether the investigation is worth a certain dedication of resources.

Now, the nodes encountered from H to $E2$ are H, $H1$ and $E2$. There are also no additional parent nodes that has to be considered. Therefore, the joint probability equation for the portion of the Bayesian Network needed for inference is:

$\mathrm{P}(\mathrm{H} \cap H1 \cap E2) = P(H)\, P(H1 \mid H)\, P(E2 \mid H1)$

Applying the enumeration method for calculating Bayesian Inference, the equation that is needed to evaluate the investigator's request is:

$$\begin{aligned} P(H = Y \mid E2 = Y) &= \frac{\sum_{H1} P(H \cap H1 \cap E2)}{P(E2)} \\ &= \frac{0.333\,[0.45 + 0.0875 + 0]}{0.333} \\ &= 0.5375 \\ &\approx 0.538 \end{aligned}$$

Therefore, the probability that the employee sent the files to a third party given the FTP packet evidence found based on Bayesian Inference is 0.538. This can be seen visually in Figure 2.

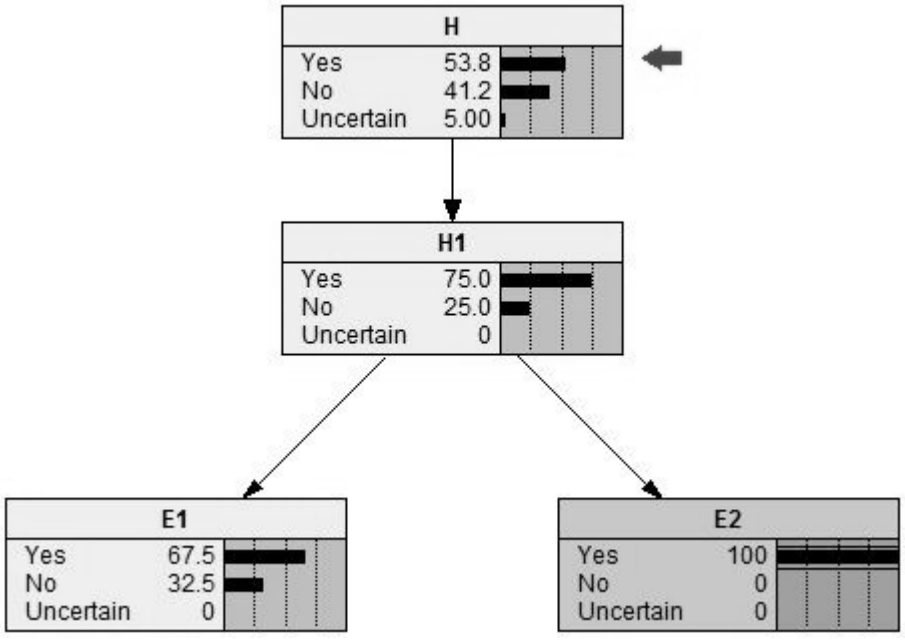

Figure 2 Bayesian Network with Observed Evidence Example

4 Conclusion and Future Work

Actions carried out by a suspect on a computer generates various forms of data artefacts in volatile main memory. In this paper, we presented MemTri. A Memory Forensics triage tool that identifies data artefacts in memory for certain Internet Browsers, Instant Messengers, Document Processors and FTP Client applications, using regular expressions. This work demonstrated that even after a targeted application process was terminated, some data artefacts could still be extracted from unallocated regions of memory.

The Bayesian Network developed in this work, encodes expert knowledge gathered from a designed digital forensics expert questionnaire, and successfully uses it to provide a probabilistic output rating that a memory image contains evidence of illegal firearms trading activity. Currently, MemTri is under development so we did not present any experimental results. However, in the final paper we plan to provide extensive experimental results regarding the overall performance and accuracy of MemTri.

We hope that this project would inspire further research into developing digital forensics triage tools, specifically geared at assessing criminal activity found in main memory. Some significant improvements have been identified to prepare MemTri for use in actual criminal investigations. Finally, this work only utilized a limited set of case-specific words to locate evidence. The next stage is to implement a Knowledge-based Natural Language Processing (NLP) system into MemTri's Evidence Search Engine which utilizes a domain-specific dictionary [19] (for example, a dictionary of illegal firearms related words). This upgrade will allow MemTri to effectively locate evidence in the context of any specified criminal investigation, thus making it practical for use in a real-life environment.

Finally, an interesting direction would be to incorporate MemTri into cloud-based services and also use it to investigate data collected through participatory sensing applications [20]. More precisely, our vision is to install MemTri on a Trusted Cloud Service provider [21, 22, 23, 24, 25] and give the option to users to run regular experiments in order to identify possible malicious behaviours. To do so, MemTri will have to develop an API that will be available via a Platform-as-a-Service infrastructure similar to the one described in [26] and [27]. By doing this, MemTri will be able to offer a reliable solution to many applications that today suffer from poor investigation of malicious behaviours. For example, the health sector that is gradually moving to the cloud will gain lot of benefits since personal health records are considered as sacrosanct [28, 29, 30] and needs to be properly protected. In addition to that, by moving MemTri with cloud-based services, we will be able to further enhance the accuracy of our tool by incorporating specific techniques [31, 32] where users' will be able to rate the veracity of the tool in an anonymous and privacy-preserving way [33, 34].

References

1. Hitchcock, B., Le-Khac, N.-A., Scanlon, M.: Tiered forensic methodology model for Digital Field Triage by non-digital evidence specialists. Digital Investigation **16**, 75–85 (2016). doi:10.1016/j.diin.2016.01.010
2. Quick, D., Choo, K.-K.R.: Impacts of increasing volume of digital forensic data: A survey and future research challenges. Digital Investigation **11**(4), 273–294 (2014). doi:10.1016/j.diin.2014.09.002
3. Hausknecht, K., Foit, D., Buric, J.: RAM data significance in digital forensics. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1372–1375. IEEE, ??? (2015). doi:10.1109/MIPRO.2015.7160488

4. Joseph, N., Sunny, S., Dija, S., Thomas, K.L.: Volatile Internet Evidence Extraction from Windows Systems. In: 2014 IEEE International Conference on Computational Intelligence and Computing Research, pp. 1–5. IEEE, ??? (2014). doi:10.1109/ICCIC.2014.7238452
5. The Volatility Foundation: Volatility 2.4 (Art of Memory Forensics) (2014). http://www.volatilityfoundation.org/ Accessed 15/08/2006
6. Sindelar, A., Moser, A., Stuettgen, J., Sanchez, J., Cohen, M., Misha, B.: Rekall Memory Forensic Framework (2016). http://www.rekall-forensic.com/ Accessed 2016-04-15
7. Vömel, S., Freiling, F.C.: Correctness, atomicity, and integrity: Defining criteria for forensically-sound memory acquisition. Digital Investigation **9**(2), 125–137 (2012). doi:10.1016/j.diin.2012.04.005
8. Garfinkel, T., Pfaff, B., Chow, J., Rosenblum, M.: Data lifetime is a systems problem. In: Proceedings of the 11th Workshop on ACM SIGOPS European Workshop: Beyond the PC - EW11, p. 10. ACM Press, New York, USA (2004). doi:10.1145/1133572.1133599
9. Rogers, M.K., Goldman, J., Mislan, R., Wedge, T., Debrota, S.: Computer Forensics Field Triage Process Model. In: Proceedings of the Conference on Digital Forensics, Security and Law, vol. 1, pp. 27–40 (2006)
10. Bogen, P.L., McKenzie, A., Gillen, R.: Redeye: A Digital Library for Forensic Document Triage. In: Proceedings of the 13th ACM/IEEE-CS Joint Conference on Digital Libraries - JCDL '13, p. 181. ACM Press, New York, USA (2013). doi:10.1145/2467696.2467716
11. Li, Y., Sundaramurthy, S.C., Bardas, A.G., Ou, X., Caragea, D., Hu, X., Jang, J.: Experimental Study of Fuzzy Hashing in Malware Clustering Analysis. In: 8th Workshop on Cyber Security Experimentation and Test (CSET 15) (2015). https://www.usenix.org/conference/cset15/workshop-program/presentation/li
12. Walls, R.J., Learned-Miller, E., Levine, B.N.: Forensic triage for mobile phones with DEC0DE. In: SEC'11 Proceedings of the 20th USENIX Conference on Security, p. 7. USENIX Association, ??? (2011)
13. Koopmans, M.B., James, J.I.: Automated network triage. Digital Investigation **10**(2), 129–137 (2013). doi:10.1016/j.diin.2013.03.002
14. Said, H., Al Mutawa, N., Al Awadhi, I., Guimaraes, M.: Forensic analysis of private browsing artifacts. In: 2011 International Conference on Innovations in Information Technology, pp. 197–202. IEEE, ??? (2011). doi:10.1109/INNOVATIONS.2011.5893816
15. Simon, M., Slay, J.: Recovery of Skype Application Activity Data from Physical Memory. In: 2010 International Conference on Availability, Reliability and Security, pp. 283–288. IEEE, ??? (2010). doi:10.1109/ARES.2010.73
16. Ray, I., Shenoi, S.: Reasoning about Evidence using Bayesian Networks. In: Advances in Digital Forensics IV, pp. 275–289. Springer, New York, USA (2008)
17. McClelland, D., Marturana, F.: A Digital Forensics Triage Methodology based on Feature Manipulation Techniques. In: 2014 IEEE International Conference on Communications Workshops (ICC), pp. 676–681. IEEE, ??? (2014). doi:10.1109/ICCW.2014.6881277
18. Norsys Software Corp.: Netica (2016). https://www.norsys.com/ Accessed 2016-06-09
19. Riloff, E.: Automatically constructing a dictionary for information extraction tasks. In: Proceedings of the Eleventh National Conference on Artificial Intelligence, pp. 811–816. AAAI Press, ??? (1993)
20. Michalas, A., Komninos, N.: The lord of the sense: A privacy preserving reputation system for participatory sensing applications. In: Computers and Communication (ISCC), 2014 IEEE Symposium, pp. 1–6 (2014). IEEE
21. Paladi, N., Gehrmann, C., Michalas, A.: Providing user security guarantees in public infrastructure clouds. IEEE Transactions on Cloud Computing **PP**(99), 1–1 (2016). doi:10.1109/TCC.2016.2525991
22. Paladi, N., Michalas, A., Gehrmann, C.: Domain based storage protection with secure access control for the cloud. In: Proceedings of the 2014 International Workshop on Security in Cloud Computing. ASIACCS '14. ACM, New York, NY, USA (2014)
23. Paladi, N., Michalas, A.: "One of our hosts in another country": Challenges of data geolocation in cloud storage. In: Wireless Communications, Vehicular Technology, Information Theory and Aerospace Electronic Systems (VITAE), 2014 4th International Conference On, pp. 1–6 (2014)
24. Michalas, A., Yigzaw, K.Y.: Locless: Do you really care your cloud files are? In: 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC), pp. 618–623 (2015)
25. Michalas, A., Bakopoulos, M.: Secgod google docs: Now i feel safer! In: 2012 International Conference for Internet Technology And Secured Transactions, pp. 589–595 (2012)
26. Verginadis, Y., Michalas, A., Gouvas, P., Schiefer, G., Hübsch, G., Paraskakis, I.: PaaSword: A Holistic Data Privacy and Security by Design Framework for Cloud Services. In: Proceedings of the 5th International Conference on Cloud Computing and Services Science, pp. 206–213 (2015). doi:10.5220/0005489302060213
27. Michalas, A.: Sharing in the rain: Secure and efficient data sharing for the cloud. In: 2016 International Conference for Internet Technology And Secured Transactions, pp. 589–595 (2016)
28. Michalas, A., Dowsley, R.: Towards trusted ehealth services in the cloud. In: 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC), pp. 618–623 (2015). doi:10.1109/UCC.2015.108
29. Yigzaw, K., Michalas, A., Bellika, J.: Secure and scalable statistical computation of questionnaire data in r. IEEE Access **PP**(99), 1–1 (2016). doi:10.1109/ACCESS.2016.2599851
30. Michalas, A., Paladi, N., Gehrmann, C.: Security aspects of e-health systems migration to the cloud. In: e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference On, pp. 212–218 (2014). IEEE
31. Dimitriou, T., Michalas, A.: Multi-party trust computation in decentralized environments. In: 2012 5th International Conference on New Technologies, Mobility and Security (NTMS), pp. 1–5 (2012). doi:10.1109/NTMS.2012.6208686
32. Dimitriou, T., Michalas, A.: Multi-party trust computation in decentralized environments in the presence of malicious adversaries. Ad Hoc Networks **15**, 53–66 (2014). doi:10.1016/j.adhoc.2013.04.013
33. Michalas, A., Bakopoulos, M., Komninos, N., Prasad, N.R.: Secure amp; trusted communication in emergency situations. In: Sarnoff Symposium (SARNOFF), 2012 35th IEEE, pp. 1–5 (2012). doi:10.1109/SARNOF.2012.6222751
34. Michalas, A., Oleshchuk, V.A., Komninos, N., Prasad, N.R.: Privacy-preserving scheme for mobile ad hoc

networks. In: Computers and Communications (ISCC), 2011 IEEE Symposium On, pp. 752–757 (2011). doi:10.1109/ISCC.2011.5983930