A Novel Methodology to Acquire Live Big Data Evidence from the Cloud

Aniello Castiglione, *Member, IEEE*, Giuseppe Cattaneo, Giancarlo De Maio, Alfredo De Santis, *Member, IEEE*, and Gianluca Roscigno

Abstract—In the last decade Digital Forensics has experienced several issues when dealing with network evidence. Collecting network evidence is difficult due to its volatility. In fact, such information may change over time, may be stored on a server out jurisdiction or geographically far from the crime scene. On the other hand, the explosion of the Cloud Computing as the implementation of the *Software as a Service* (SaaS) paradigm is pushing users toward remote data repositories such as Dropbox, Amazon Cloud Drive, Apple iCloud, Google Drive, Microsoft OneDrive. In this paper is proposed a novel methodology for the collection of network evidence. In particular, it is focused on the collection of information from online services, such as web pages, chats, documents, photos and videos. The methodology is suitable for both expert and non-expert analysts as it "drives" the user through the whole acquisition process. During the acquisition, the information received from the remote source is automatically collected. It includes not only network packets, but also any information produced by the client upon its interpretation (such as video and audio output). A trusted-third-party, acting as a *digital notary*, is introduced in order to certify both the acquired evidence (i.e., the information obtained from the remote service) and the acquisition process (i.e., all the activities performed by the analysts to retrieve it). A proof-of-concept prototype, called `LINEA`, has been implemented to perform an experimental evaluation of the methodology.

Index Terms—Digital Forensics; Network Forensics; Live Network Investigation; Certified Network Acquisitions; Big Data Forensics; Digital Investigations.

1 Introduction

The role of digital evidence in the context of forensic investigations is becoming more and more crucial. In the last years the scientific community has developed a large number of principles and methods for digital investigation. Nowadays most of the international laws and best-practices are based on those results [1], [2], [3], [4]. Recently, the development of technologies such as Cloud Computing and service-oriented infrastructures has radically changed the way of storing and transferring digital information. Common methods and techniques for traditional digital forensics (or storage media forensics) are based on the assumption that the device under investigation is physically available to the analyst [5], [6], [7], [8]. Clearly, these approaches cannot be applied when dealing with the aforementioned technologies, especially when in presence of Big Data [9], [10], [11], [12], [13], cloud storage forensics [14], [15], [16], [17], [18], [19] and so on [20], [21]. Moreover, the availability of high speed networks, the widespread of mobile devices and smartphones, the growing use of *Online Social Networks* (OSNs) [22] [23] lead to believe that more and more digital investigations will involve the Internet. In this context, the scientific community has developed an increasing interest towards acquisition and analysis of information flowing through a network involving Big Data [24], [25], [26]. This is the main goal of Network Forensics which intends to provide law enforcement authorities with robust remote forensics technologies to access such kind of evidence [27].

Live Network Evidence (LNE) consists of any kind of information that can only be accessed through a network. In [28] Nickel focuses on the acquisition of LNE from online services, such as websites, network repositories and OSNs. Typically all these services are delivered through the Internet according to an emerging model known as Cloud Computing. Some methods have been proposed in the past to discover and collect information from the Internet, but none of those produce *trustworthy* evidence. Evidence is trustworthy if its integrity, authenticity and origin (*where* and *when* the acquisition has been performed) can be proved in a strong way.

Nowadays, the common practice to produce a trustworthy LNE involves the participation of a "human" *Trusted-Third-Party* (TTP) in the acquisition process, such as a notary. In that case, the notary should certify with reports, photos and videos any information of interest obtained by the investigator from the inquired online service. Such trusted individual should be able to distinguish eventual errors or anomalies in the process, which requires a certain technical knowledge.

An alternative to the human TTP is the use of a forensic software able to automatically generate trustworthy LNE. Although some tools for the acquisition of evidence from online services have been recently proposed (e.g., WebCase [29], Hashbot [30] and *PNFEC* [31]), they do not provide sufficient requirements in order to guarantee confidentiality, integrity and authenticity of the collected evidence.

- *All the authors are with the Department of Computer Science, University of Salerno, Italy. E-mails: castiglione@ieee.org, cattaneo@unisa.it, g.demaio@gmail.com, ads@unisa.it, giroscigno@unisa.it*

Corresponding author: Aniello Castiglione, Department of Computer Science, University of Salerno, Via Giovanni Paolo II, 132 - I-84084 Fisciano (SA) - Italy (e-mail: castiglione@ieee.org)., Phone: +39089969594, Fax: +39089969821.

Manuscript received 01 September 2016; revised 15 December 2016; revised 16 February 2017; accepted 24 February 2017.
For information on obtaining reprints of this article, please send e-mail to: preprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TBDATA.2017.XXXXXXX

In this paper a novel methodology for automatic acquisition of evidence from online services is proposed. The methodology complies with the definition of *forensically-sound evidence* given by McKemmish [32]: "The application of a transparent digital forensic process that preserves the original meaning of the data for production in a Court of Law." We show a reliable and accurate forensic process that allows to produce trustworthy LNE, whose integrity, authenticity and origin can be verified at any time after the acquisition. The methodology has been implemented through a *network trusted service* embedded in the Cloud which presents to the remote investigators different *Operating Modes* (OMs). The first OM is based on a transparent HTTPS proxy, which is able to record any activities at network level (IP) during the access of an online service. The second OM makes use of an *agent* which, in addition to network-level data, is able to collect higher level information in a *What You See Is What You Get* (WYSIWYG) manner. In both cases, the acquisition and production of evidence is transparent to the investigator, whose task is just to *tell* the collector *where* is the information to be acquired. The third OM is meant for expert investigators and provides a low-level control of the acquisition process.

Afterwards, the collected evidence is digitally signed, timestamped and provided to the investigator. The collector produces a report containing high-level information, which can be accessed by non-technical parties (such as judges, juries, lawyers, etc.) without the assistance of technical consultants and/or advanced analysis tools.

Finally, the implementation and evaluation of a fully-fledged prototype to collect LNE, called *LIve Network Evidence Acquisition* (`LINEA` for short), is presented. It has been proven to be forensically-sound, i.e., the collected evidence is robust and its reliability can be verified at any time after the acquisition. The robustness property is enhanced by the correlation of information from multiple sources of evidence, while the reliability is provided by encrypting, timestamping and digitally-signing the result of the acquisition.

Organization of the paper: In Section 2 is treated in detail the LNE problem, focusing on juridical and technical issues. In Section 3 are discussed local and remote tools for digital investigations proposed to solve the technical issues related to the LNE. Our methodology for the LNE acquisition is presented in Section 4, while security and integrity issues are treated in Section 5. Subsequently, the implementation and evaluation of the `LINEA` prototype for LNE acquisition is presented in Section 6. Sections 7 and 8 present the future directions and the conclusion, respectively.

2 Live Network Evidence

The definition of LNE given so far [28] is very general. The purpose of this section is to clarify the objective of this work. In this sense, we need to give a more precise definition of LNE and to evaluate all the issues related to its acquisition and management.

2.1 New Definition of Live Network Evidence

Digital evidence is supposed to reside on a digital device. Traditional storage media forensics assume that such device is available to the investigator for the analysis. On the contrary, a LNE has been defined as any kind of information flowing through a network. Communications on computer networks are typically based on the client-server model. As a consequence, most of the relevant information on a network is maintained by servers, which can be requested by the clients through a particular protocol. This work focuses on the acquisition of LNE from online services. In this context, a LNE can be defined as a digital information that holds the following properties: (1) the system(s) containing the evidence is physically unaccessible; (2) the target information can be accessed through a network request.

Information generated by a remote server may undergo various modifications until it is experienced by the user. A different acquisition technique has to be adopted depending on the source used to observe the target information. The result of the acquisition may vary as well. In particular, at least four different sources can be identified: (1) the physical device containing the information (e.g., the server storage media), (2) an intermediate element of the network (e.g., a router or a proxy), (3) the client-side network interface and (4) the client-side application processing the information from the server. In the last case, the result of the acquisition is a higher-level information, such as the video rendering of a web page in case the client-side application is a browser. Figure 1 shows an overview of the different sources where LNE can be acquired.

A typical example of LNE may be an information contained in a web page. It is physically stored in a file on the web server, then it is transmitted over the network as a sequence of packets upon a client's request. On the client side, the network flow is reassembled, processed and rendered by the web browser. During this process the initial information (the one stored on the web server) may be transformed by the various elements along the communication path. Sometimes the information presented to the user may be totally different and unrelated to the original. This may happen, for example, in case of a *man-in-the-middle* attack, where the attacker replaces the server response before redirecting it to the client. This case may be detected if the analyst is able to capture the transient information from the different elements along the communication channel. It allows to correlate this information at different levels in order to increase both the accuracy and the reliability of the collected evidence.

2.2 Issues on Acquiring LNE

The scope of network forensics is very wide [33]. In particular, this work focuses on the problem of acquiring information (digital evidence) from the "Cloud World" (online services like a network repository, e.g., Dropbox), which can be used by a party in a legal case. This question is becoming more and more relevant due to the increasing number of crimes involving the Internet, such as online frauds, identity theft, copyright infringement, illegal material sales, libel or public injury produced by publishing false or private records. The only assumption of our methodology is that the online service under investigation is accessible through a network. This solution can be applied in case the physical device containing the target information is not available or is not under the direct control/jurisdiction of the investigator.

Before presenting our methodology, it is worth discussing the most common issues that may incur while acquiring LNE, which may be classified into juridical and technical issues. This work mostly focuses on giving a solution for the latter category, but also the juridical issues are introduced and reviewed.

2.2.1 Juridical Issues

The final goal of Digital Forensics is to gather information from digital devices which may have probative value in a legal proceed-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2683521, IEEE Transactions on Big Data

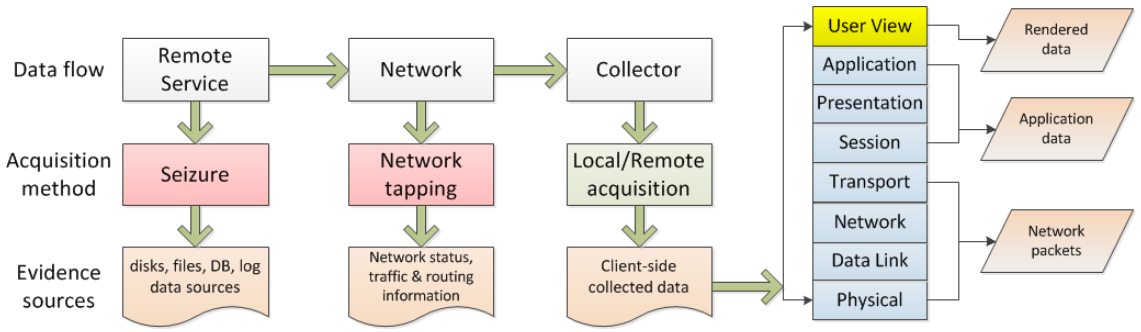


Figure 1: LNE acquisition overview.

ing. Most of the best-practices, techniques and tools for digital forensics currently viable in a Court of Law suppose that the device under investigation (e.g., the notebook or the smartphone of the accused person) is physically available to the analysts. On the contrary, LNE inherently suffers of "loss of location" and its use in Courts is not yet regulated by a common and robust legislative framework.

Nowadays experts in law from all the countries are aware of the strategic role played by digital evidence in legal proceedings and are granting more and more credits to it. Tasks concerning the digital forensics are often assigned to technical experts, who are in charge of demonstrating the integrity and the authenticity of the evidence presented to the Court. Cloud Computing allows malicious users to store their data in a virtual space physically located beyond the jurisdiction of their country of residence. In case the evidence is located on a remote host (e.g., a website) in the same country, the judge may order the seizure of the hardware containing the original information (e.g., the physical server). This approach suffers of two hard limits: a) information on the Internet is transient, which means that the content of the device may have changed at the time of the seizure; b) even if a judicial cooperation treaty has been signed, when the server hosting the online service is located in a foreign country, the seizure is often hard and time consuming. In some other cases the owner/producer of the evidence is unknown or undeclared. For instance, article 22 of the Budapest Convention on Cybercrime [34] asserts the crimes committed on ships or aircraft are subject to jurisdiction of the *flag State* (known as the *flag principle*). Applying this principle to the Cloud scenario crimes committed by means of online services should be under the jurisdiction of the state where the company hosting the service has been registered.

From a legal point of view, most of well-known assumption valid for traditional digital evidence do not apply to LNE. In facts it is difficult to answer to questions like: "What is a copy of a web page?", "Which documents should be produced so that the copy can be considered exactly equivalent to the original (at a given time)?", "How can a notary certify such an equivalence?". In the last decade, due to the lack of rules concerning those aspects, the Courts accepted printouts of web pages, photos, videos or transcripts as evidence [35]. Such a kind of evidence cannot be proven to be conform to the original data. In fact, even if the acquisition is supervised by an authoritative third-party like a notary, the observed value may vary depending on various condition of the network as well as the operating system and the browser used to perform the analysis. Moreover, it requires that the human third-party have enough technical skills in order to understand and document the methodology adopted by the investigator to access the inquired information. Another issue is that, in some cases, the correct acquisition of a LNE may depend on the timeliness. For example, just think about a fraudulent announce on a website which may last a few minutes. The time frame needed for the intervention of a human third-party may determine the total loss of that information.

Since the use of LNE is not currently ruled by neither precise laws nor best-practices, its acceptability in Courts is still a hot topic. Traditional digital evidence is acquired through a clear procedure, which enforces the use of write-blocking tools in order to prevents alteration of the original data. On the contrary, LNE is extremely volatile since online information may be modified in any moment by the author, by the administrator of the server, by malicious users, and so on. Managing LNE is extremely difficult, since it is not usually possible neither to rely on the availability of the original copy, nor to verify (*ex post*) the integrity of the acquired information.

2.2.2 Technical Issues

The structure of the Internet inherently complex includes the presence of elements like gateways or proxies makes hard to geographically localize the physical system providing an online service. This is even more difficult in case the inquired information is present on a peer-to-peer network or an anonymity network such as Tor [36]. Even in domestic investigations the seizure of a server is not always viable, for example, in case more mission-critical processes run on the same hardware of the service under investigation. In the Cloud scenario things may be even worse. In fact, service providers may use resources spanning all over the planet and the service endpoint may transparently be moved over different servers according to load balance rules or fault tolerance schemas. Data are often duplicated across different servers far from each other.

More issues are related to the technology adopted to implement the service. For example, the web is becoming more and more dynamic due to the introduction of HTML5 [37]. In such a context, an information present on a web-server is even more transient and a prompt timeliness may be required to capture a particular information. To better clarify this concept we distinguish two distinct sources of evidence:

- **Static services**: the information stored on the server is provided to the user without any further manipulation or, in other words, the information received by the client is the perfect copy of that stored on the physical system providing the online service. Examples of this kind of

information are files provided by a FTP servers, static web pages, PDF file, and so on. In such a case, the acquisition of a LNE is quite simple since different requests for the target information will always produce the same response (at least within a certain time frame).

- **Dynamic services**: the information provided by the service is dynamically computed on the fly before being sent to the client. For example, the server response may be the result of queries to a DBMS or other computations depending on parameters contained in the request. This is the case of web pages implemented by means of server-side scripting languages (PHP, ASP, etc.), web pages making calls to servlets or any other processing entity. In such a context, it is not possible to guarantee that two different requests for the target information produce the same response. In other words, the LNE may vary depending on several factors and the investigator should be able to properly tune the request in order to retrieve the desired information.

In the latter case the acquisition is much more complex since the evidence may be the result of user interactions with service interface. For example, a particular image on Facebook can only be gathered by manually browsing a photo gallery. Due to the high dynamicity of this online social network, it is not possible to automatically accomplish such a task. In fact, the actions necessary to retrieve the same picture may vary depending on the time (e.g., the user has modified the order of the photos in the album).

The information to be acquired not only depend on the response of the server, but also on the way it is processed by the client. For example, the rendering of a web page may depend on the browser, language, presence of specific plugins, as well as on physical characteristics of the workstation such as screen resolution. In order to cope with these situations, the investigator should be able to reproduce the exact series of steps needed to retrieve the information of interest.

3 The Acquisition of LNE

A number of tools for digital investigations have been proposed in the last years to try and solve the technical issues related to the LNE acquisition. These are generically called *Network Forensic Tools* (NFTs) [28]. For sake of clarity, we make a distinction between *local* and *remote* tools. The first category includes software operating on the investigator's workstation, while the second category comprises tools implemented as a third-party service, which can be accessed by the investigator through the network. In both cases, a human third-party can be employed in order to validate the operations performed by the investigator during the acquisition process.

3.1 Local and Remote LNE Acquisition Tools

Local tools, such as WebCase [29], are directly installed on the investigator's workstation and provide facilities to collect LNE. This kind of tools allow the investigator to acquire information at different abstraction levels: network traffic, application-level data and post-processing information (i.e., information experienced by the user as result of the application processing).

However, the use of a local acquisition tool has some drawbacks. First of all, both the human analyst and the equipment used for the acquisition should be fully trusted. In fact, a malicious investigator with full access to the acquisition environment could tamper the collected evidence, or the hardware/software equipment could have been compromised by a malware, with it invalidating the admissibility of the result. The employment of a human TTP does not solve the problem, since the investigation system could have been tampered before the beginning of acquisition process. Moreover, the notary should have enough technical knowledge to certify the work of the investigator.

With remote tools, such as Hashbot [30], the acquisition is performed by a TTP which acts on behalf of the investigator. The main advantage of this approach is that trustiness of the equipment used for the acquisition is no longer required, since the evidence is collected on the trusted remote host.

Remote acquisition tools proposed in the last years suffer from several limitations. First of all, they typically lack of non-repudiation and data-integrity solutions to protect the collected information, which means that the result of the investigation could be tampered by a malicious user/investigator. Moreover, even though the acquisition process is driven by the investigator, which instructs the online service on the information to acquire, it is typically limited to static resources (e.g., a static web page). This is a serious limitation in case the target information resides on a dynamic website. In fact, the server response could depend on specific client parameters (user-agent, operating system family and version, installed plugins, screen resolution, etc.) which, typically, cannot be tuned by the investigator. Even worse, the target information could be the result of client-side processing (e.g., JavaScript) or user interactions (e.g., a mouse click).

The limitations of the currently-available acquisition tools can be solved by the introduction of a *fully-automated service* providing a complete user-driven acquisition interface. Acting as a TTP, it is able to validate and certify both the acquired information and the operations performed by the investigator to produce those results.

Before presenting our novel methodology, it is worth enumerating and discussing in details the most common acquisition tools used today in digital investigations.

3.2 Local Acquisition Tools

A *packet analyzer* is a computer program or a piece of computer hardware able to intercept, filter and log traffic passing over a network segment. Some of the most common software tools belonging to this category are *tcpdump* [38] and *Wireshark* [39], both open-source, which are based on the *pcap* libraries [38], [40]. As data streams [41] flow across the network, the sniffer, can set its network adapter in promiscuous mode, and captures each packet present on the network segment.

Packet sniffers, such as Wireshark, are specialized in the analysis of specific network and/or application protocols. Based on the appropriate RFC or other specifications, they are able to reconstruct and decode the raw network stream in order to present the carried information in a human-readable form. Packet analyzers are largely used in network forensics to intercept communications over a network. Concerning the acquisition of LNE from remote services, packet analyzers suffer from some limitations. First of all, they should cope with end-to-end encryption protocols such as *Transport Layer Security* (TLS) and *Secure Sockets Layer* (SSL). In order to decode the encrypted communication [42] [43], the acquisition tool should be used locally on the collector workstation.

Another limitation is that the acquired data are typically complex, mostly when dealing with dynamic resources (dynamic web pages, multimedia, etc.) or proprietary communication protocols. Moreover, the interpretation of such data may depend on the specific environment (operating system, user agent, language, etc.), which makes it hard to reconstruct the original information.

WebCase [29] is a local acquisition tool able to collect data from online services. It records information at different abstraction levels, besides network-level information (packets, routing data, etc.), and it is also able to acquire post-processing information (audio and video output). WebCase suffers from the typical limitation of local acquisition tools. In particular, it requires full trust in both the acquisition environment and the human investigator.

The last version of WebCase allows to capture the contents of a web page also including parts for which the user is supposed to scroll down the windows to display them. This tool has some operational limitations. For instance, it works correctly on Microsoft Windows operating systems family, requiring Microsoft.NET Framework version 2.0 or later. In addition, WebCase is compatible with Internet Explorer 6, 7 and 8, but it does not allow to use other browsers.

FAW [**?**] *Forensic Acquisition of Websites* (FAW) is a web browser explicitely created to capture web pages for forensics purposes. It can acquire a web page or part of it, also including streaming videos, frames and tool-tips. FAW uses the capabilities of Wireshark to capture all of the traffic on all interfaces active network during the acquisition of the web pages. In addition, FAW calculates MD5 and SHA1 hashes of all acquired files, producing a detailed log of operations carried out with related time references. The function of acquisition's integrity check allows to verify if all the captured files have not been forged. Finally, the acquisition output can be automatically sent at a certificated mailbox with the purpose to temporally attest the acquired data.

3.3 Remote Acquisition Tools

Some free services available on the Internet allow to retrieve past data published on the web, event after it is no longer available. For example, *Internet Archive* [44] is a non-profit digital library with the stated mission of providing "universal access to all knowledge". It offers permanent storage and access to a wide collections of digitized materials, including websites, music, animated images, and nearly 3 millions of public domain books.

A similar service is *Page Saver* [45], which allows to capture and backup web pages (or visible portions of web pages) as images. The capturing process can be tuned by means of a variety of settings, which include image format and scale.

Although not originally meant for digital forensic purposes, these services can be used as remote acquisition tools based on a TTP. However, forensically-soundness of the collected evidence cannot be guaranteed, since no warranty is provided on the data stored on the servers. In addition, no cryptographic techniques are employed to enforce originality and integrity of the acquired information. A further limitation is that only static web pages could be retrieved.

Hashbot [30] is *"a forensic web tool to acquire and validate, over the time, the status of an single web page or web document"*. It provides a web interface that allows to specify the URL of the remote resource be acquired. The remote resource is subsequently obtained by means of a HTTP request, the server response is packed and provided to the investigator. The integrity of the acquired information is enforced by including an hash value of the received data.

The main limitation of this service is that only static resources can be acquired, since dynamic resources may require user interaction or client-side interpretation. The number of tunable parameters is limited to a small set of user agents. Moreover, it does not provide robust mechanisms, like digital signature, to prevent tampering of the downloaded package.

PNFEC [31] is a portable network forensic evidence collection device, built using inexpensive hardware and open-source software. It operates at the link layer and can be transparently inserted inline between a network node and the rest of a network. It allows to intercept all the traffic to/from a single network node. The device offers different operating modes, including an investigator mode meant for collection of evidence from remote services.

Unlike previous solutions, PNFEC provides various cryptographic techniques to preserve non-repudiation and confidentiality of the collected data. Since it is just an intermediate element of the network, this tool is not able to acquire data exchanged over a secure end-to-end channel such as SSL. Moreover, the low-level nature of the acquired data could make interpretation of the evidence very difficult.

4 A New Methodology for the LNE Acquisition

In this section we present a novel methodology to collect LNE. Although the methodology can be used to acquire information from a generic service on a common network, for sake of simplicity we focus on the acquisition of information from the World Wide Web (WWW), whose communication protocol is HTTP.

The methodology assumes the presence of a network service acting as a TTP, which is in charge of collecting information from an online Service (for brevity, *S*) on behalf of an Investigator (for brevity, *I*). The acquisition process is initialized by the investigator *I* by establishing a connection with TTP. The communication channel between *I* and TTP is protected (e.g., by means of an encrypted Virtual Private Network) in order to guarantee confidentiality and integrity of the exchanged information as well as privacy to the involved parties. The TTP provides a communication interface through which the investigator is able to *drive* the acquisition process.

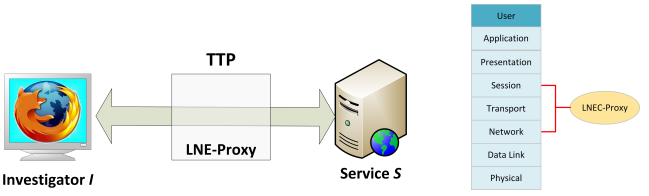
The methodology provides three different *Operating Modes* (OMs) with several features and communication interfaces. With first operating mode (for brevity, *OM1*), called LNE-Proxy, the investigator can use its local web browser as graphical interface, while all the traffic to/from the inquired service is tunneled through the TTP. In the second OM (for brevity, *OM2*), called LNE-Agent, the investigator can open a remote session with the TTP in order to use its graphical interface (browser) to access the target service The third OM (for brevity, *OM3*) is meant for savvy users, who can leverage all their experience to perform the acquisition. In that case, the TTP provides a fully-fledged virtual environment where the investigator can use the all the pre-installed tools or even download additional software from the Internet.

In all the three cases, the TTP is in charge of recording the traffic to/from the investigator *I* and TTP as well as the traffic to/from TTP and *S*. With OM2 and OM3, the TTP is also in charge of recording all the operations performed by the investigator. In

addition, in these latter cases, the TTP is able to record a wider set of information, which includes the rendering on screen of the data received from the target service (e.g., the rendering of a web page in the browser).

4.1 LNE-Proxy Operating Mode

In the first OM (OM1), that we called LNE-Proxy, the TTP acts as a HTTP/HTTPS proxy. The investigator can configure its local browser in order to use the TTP to acquire digital evidence. All the network traffic flowing through the TTP proxy is captured, signed and finally sent to the investigator *I*. An overview of the architecture implemented by LNE-Proxy is shown in Figure 2a. The TTP has access to information only from the basic network levels of the ISO/OSI model, as shown in Figure 2b.



(a) LNE-Proxy operating mode (b) OSI reference layers

Figure 2: LNE-Proxy architecture.

The network traffic captured by the TTP is saved in the standard *pcap* format, which is supported by most of network analysis tools. It is worth noting that network packets contain lot of low-level information which can be useful for the subsequent analysis phase (which is out of scope of this work).

For example, packet headers include source and destination IP addresses, which may be used to correctly identify all the parties involved in the communication, as well as to geo-locate the server under investigation. The packet checksum may be used to detect eventual errors in the communication, which might have altered the information obtained from the target service. Moreover, each packet header includes an accurate timestamp (with resolution of 1 microsecond) expressed as number of seconds since January 1, 1970 00:00:00 GMT. This information allows to discern correlations among content of packets and other events, assuming that the clock of the TTP is synchronized with a worldwide official time server (for example, by means of the Network Time Protocol protocol, for brevity, NTP).

Thanks to these low-level information, a packet stream can be considered a reliable source of evidence. The consistency of the stream can be checked by means of a packet analyzer, which can also provide an higher-level view of the traffic. For example, tools like *Xplico* [46] are able to decode interesting objects such as images, videos, web pages and so on.

In order to strengthen the result of the acquisition, the investigator could produce a video showing the rendering of the collected information on his display (e.g., by means of a camcorder). However, the methodology cannot guarantee the reliability this evidence, since the TTP does not have access to such a high-level information. The rendering on screen may depend on a series of parameters that are not under control of the TTP, such as browser software used by the investigator, display resolution, color intensity and so on.

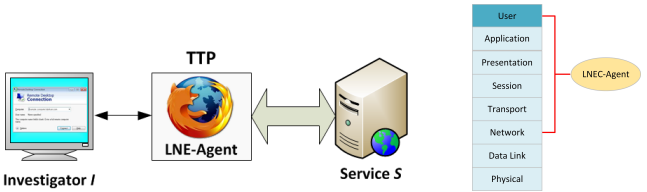
Whether the target service requires an encrypted connection (HTTPS), the TTP proxy is delegated to the negotiation of the session key with the server. Whenever a mutual authentication is required, the TTP should be equipped with a valid certificate accepted by the server. This should be provided by the investigator, for example, before initializing the acquisition session.

4.2 LNE-Agent Operating Mode

In the previous case all the requests to the service *S* are generated from the investigator's workstation, with them transparently flowing through the LNE-Proxy. In this case, the LNE-Agent plays an active role in the communication with the target server, since it is in charge of generating requests on behalf of the investigator. The general architecture of LNE-Agent is depicted in Figure 3a.

Also in this case the acquisition is driven by the investigator. The service uses a *Remote Desktop Protocol* (RDP) to provide the investigator with a remote control interface. In fact, in order to gather the target information, the investigator *I* remotely accesses the TTP server console. Then, he/she starts the browser and finally can visit the target website typing its URL and navigating through the pages under investigation. The audio and video stream experienced by the user during the acquisition process, referred to as *user view*, is simultaneously recorded by LNE-Agent along with the entire network traffic to/from service *S*.

For sake of simplicity and without loss of generality, we can assume that the investigator only requires a web browser to access the service *S*.



(a) LNE-Agent operating mode (b) OSI reference layers

Figure 3: LNE-Agent architecture.

With respect to OM1, the second OM (OM2), that we called LNE-Agent, gains access to a wider range of information sources, as summarized in Figure 3b, where they have been extended up to the *user* experience. According to the definition given by Nikkel [28], [47], the LNE-Agent OM2 enforces the concept of multi-layer acquisition, since it allows to collect evidence from a number of different and complementary sources.

In particular, at least four tightly coupled evidence sources can be identified:

1) *Transport-Layer and Application-Layer Packets*
   All the packets at transport level (TCP and UDP) and all the messages at application level (e.g., HTTP) exchanged with the service *S*.
2) *Global Network-State Information*
   Dynamic information about the current network status, such as routing information, network name resolution, network paths, domain-related information and so on.
3) *User View*
   The rendering of the data (i.e., what user sees) received

from the service as result of the client application processing, which includes the audio/video information experienced by the investigator through the remote desktop connection.

4) *Input Events*
A complete log of the commands used by the investigator to perform the acquisition (e.g., keyboard and mouse events), which may reveal information that are not visible on screen or are difficult to discern from the network traffic (e.g., encrypted passwords).

At the end of the acquisition session, a package file containing these four sources of evidence is generated, signed and timestamped by the TTP. During the analysis phase, the target evidence can be correlated by using the time references associated with the acquired information.

In order to improve security and privacy as well as packet filtering capability and process isolation, OM2 requires that the use of the TTP is exclusive, i.e., the TTP cannot be shared among different concurrent users. Therefore, each investigator should be provided with a dedicated (virtual) machine which is destroyed at the end of the acquisition session. Under this assumption, traffic acquisition and filtering is straightforward since each investigator uses a dedicated (virtual) adapter.

Each time a remote host is accessed, a set of network state information is collected. This includes information about the hostname resolution, the network path traversed to reach the host as well as information related to the domain name. This information can be subsequently used to prove the correctness of the acquisition or, conversely, to show an eventual corruption of the acquisition (e.g., due to external attacks such as DNS poisoning or IP address spoofing).

The LNE-Agent OM produces a digital record of the audio/video output experienced by the user during the investigation. It contains several high-level information such as mouse movements and the use of GUI components like buttons, text areas, scroll bars and so on. In this way, the investigator is provided with an immediate visual feedback of the acquisition. Moreover, this kind of information is also accessible by non-technical staff such as lawyers and judges, and can be used by the investigator to explain evidence during the trial.

The LNE-Agent service collects not only information from the target service but also all the operations performed by the investigator in order to access it (e.g., keystrokes and mouse movements). The acquisition procedure may be itself subject to investigation (e.g., by means of the counterpart in the trial), and can be used to expose erroneous or even malicious activities. Moreover, this source of evidence may contain information that is neither displayed on the screen nor easily identifiable in the network traffic. These can be, for example, passwords typed in a input box with the hidden attribute on, mouse clicks on hidden objects (e.g., in case of click-jacking attacks) and so on. In some cases, data might be intentionally hidden to the user, for example, by means of sophisticated steganographic techniques [48], [49]. For these reasons, the LNE-Agent OM requires that all the input events produced by the investigator are recorded. It can be accomplished, for example, by means of a mouse/keylogger.

4.3 Expert Investigator Operating Mode

The idea of the third operating mode (that we called OM3) is to provide the investigator with a trusted fully-fledged workstation. Like the previous case, the communication interface provided by the TTP is a remote desktop service. The main difference is that the investigator has access to a fully-fledged operating system, which may be equipped with a set of computer forensic tools (e.g., the entire DEFT suite [50]). The investigator could also download more software from the Internet and combine the use of multiple tools to investigate complex cases. Choosing the OM3, the investigator is free to adopt his own strategy for LNE acquisition.

Also in this case three information flows are collected and signed by the TTP: network traffic, network status and input events. In order to prevent tampering, the processes related to these activities should be executed with higher system permissions.

The main difference with the previous OMs is the absence of any system restrictions. While this OM provides more acquisition potentiality, this may be misleading since no guidance is provided to the investigator. Therefore, the use of the Expert Investigator Mode should be reserved to expert users.

4.4 A Comparison between the Operation Modes

The main advantage of the LNE-Proxy OM (OM1) is its simple design, which allows for a straightforward implementation and deployment. With respect to the LNE-Agent OM (OM2), the beginning of a new acquisition session does not require any complex pre-computations to prepare the TTP. In general, the OM2 is meant for situations where timelines is required.

Visual information is typically more accessible with respect to low-level data like network traffic records. For this reason, an investigator should always produce a visual report of the collected LNE. In the case of the OM1, the output of the acquisition process is the investigator's workstation. As a consequence, the audio/video stream can only be produced locally with all the disadvantages discussed in 3.1. In general, the OM1 cannot guarantee tight correlation between the audio/video stream acquired on the investigator's workstation and network data collected by the TTP.

On the contrary, in the case of the LNE-Agent OM, the audio/video stream is recorded by the TTP itself. As result this stream is perfectly coherent with the network traffic and the network status information, since the output data are a direct function of the data received from the network layer. In fact, when the browser receives a HTTP message, it interprets its content and the result is rendered on the display of the TTP. At the same time, the user view is forwarded to the investigator by means of the Remote Desktop Protocol. Clearly, with respect to the OM2, the audio/video record produced by the TTP is more robust.

In addition, the video record provided by the LNE-Agent OM allows to avoid any ambiguities in the interpretation of data received from the service *S*. It is particularly useful when the responses of the service *S* depend on some particular client characteristics, which typically happens when accessing dynamic web pages whose content may vary in function of browser version, system language, display resolution and so on.

Finally, the acquisition of *input events* improves the completeness and the robustness of the acquisition, since it allows to collect information which may be not available from other evidence sources.

5 Security Analysis of the Proposed Methodology

The collector environment should be equipped with effective security measures to avoid that a malicious user could tamper the

acquisition. In a more relaxed form, the investigator or any other third-party should be at least able to verify if the acquisition has been corrupted. In this section, some solutions to guarantee such a property are discussed.

5.1 Threats and Strength

The correctness of both LNE-Proxy and LNE-Agent OMs strongly depends on the integrity of the system involved in the acquisition process. In both cases, it is possible to identify two points of failure: the investigator's workstation and the TTP system. The TTP cannot be corrupted by definition. The integrity of all the software tools running on the TTP is assumed as well.

An attack to the investigator's workstation may have different impacts on the acquisition process depending on the adopted OM. In the case of LNE-Proxy, if the investigator's workstation is compromised the attacker might be able to tamper the requests to the service S. For example, the web browser used by the investigator could be redirected to a website different from that expected. As a consequence, the evidence collected by the TTP might be unpredictable. In order to mitigate this problem, a remote auditing and assessment method similar to those proposed in [51] and [52] could be used to verify that critical system components respond positively to sanity and consistency checks. Another approach is that the TTP could provide the investigator with an *hardened* browser to be downloaded before starting the acquisition. In any case, an analysis of the network state information collected by the TTP may expose such a kind of attack.

In the case of OM2 and OM3, the investigator's workstation plays a less relevant role. In fact, it can be considered as a *dumb* terminal. Assuming that an attacker gained access to the investigator's workstation, he could only alter the user view by tampering the multimedia elements (i.e., images, audio and video) received from the TTP) without affecting the LNE acquisition. Subsequently, it is easy for the investigator to discern that the information acquired by the TTP is different from that expected.

For all the OMs, the security of the TTP can be enhanced by leveraging the isolation principle guaranteed by the virtualization environment. In practice, it is possible to implement the TTP system as a *virtual machine* (VM) which is instantiated from scratch each time a new acquisition session is initiated. The virtual machine is destroyed immediately after the acquisition is terminated, but, anyway, a different policy can be also to keep the virtual machine after the end of the acquisition. Clearly, multiple TTP instances can be executed to concurrently serve multiple users. It is worth noting that also the communication channel between the investigator I and the TTP must be secured to avoid man-in-the-middle attacks. Therefore, all the OMs assume that a secure tunnel between the TTP and the investigator is established before starting the acquisition. The collection of network status information can be also considered as an enhancement of the TTP security. For each host accessed during the acquisition, information regarding the hostname-to-IP-address resolution (and vice-versa) as well as the network path traversed to reach the destination is collected. The analysis of this information may expose attacks aimed at redirecting the TTP to malicious hosts (e.g., DNS poisoning).

5.2 Data Integrity

Regardless of the specific OM, immediately after the investigator terminates the session with TTP, an archive file with a filename which uniquely identifies the acquisition session is created. It contains:

1) the set of collected evidence: D
2) the cryptographic hash value of these data set: $H(D)$
3) a digital signature of this hash value obtained with the private key TTP_{sk} of the TTP: $Sig_{TTP_{sk}}(H(D))$
4) a timestamp released by an official Time Stamping Authority: $TS(H(D))$.

This information allows the investigator to prove the integrity of the collected LNE, the time when the acquisition has been performed, the identity of the TTP and, as side effect, the procedure used by the investigator himself to perform the acquisition.

Since in many countries the digital signature has legal value, the TTP is supposed to seal the collected data by using a legally-compliant electronic signature algorithm and a digital certificate released by an official Certification Authority. It allows to verify the author (TTP) and the integrity of the package. In addition, a timestamp released by an official Time Stamping Authority is added to the package, which allows to verify the validity of the contained information. Thanks to these operations, no further manipulation of the acquired data is possible, exactly as it happens for traditional digital evidence such as that collect from write-locked memories.

6 The LINEA Prototype: Design and Implementation

The three operating modes presented in the previous sections share the same methodology, which is based on a TTP in charge of acquiring information from a remote service in order to produce trusted LNE. As a consequence, most of components used for the implementation are in common. Hypothetically, a single machine can implement all the three OMs.

In this section we present the tool LINEA a fully-fledged prototype implementing the methodology described in Section 4. The prototype has been developed in order to perform an experimental evaluation of the proposed methodology on real-word cases involving the acquisition of LNE. In particular, LINEA implements the LNE-Agent OM. Using the same architecture and tools, the prototype can be straightforwardly extended with the other OMs.

As shown in Figure 4, LINEA is composed by two main components: the *Master Website* (MS) and several instances (one for each connected user) of *CollectorVM*s. During an acquisition session the CollectorVM runs four concurrent applications: (1) a web browser, which is used by the investigator to access the network service S, (2) a desktop video recorder, used to capture the user view, (3) a packet sniffer to collect the network traffic and (4) a mouse/keylogger to capture the input events. The numbered arrows in Figure 4 represent the six steps to be performed by the investigator I in order to accomplish the acquisition of LNE from a target service S.

Firstly the MS provides a webpage to let the investigator I to set up his account (step 1 Registration phase). Optionally, in this step the user can upload a digital certificate with his own RSA public key which will be used by the TTP to encrypt the acquisition results when the session is closed. After the registration, a private area is reserved to the user where the collected data are temporary stored before the download. In this way the history of the acquisitions performed by the user with the resulting packages

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2683521, IEEE Transactions on Big Data

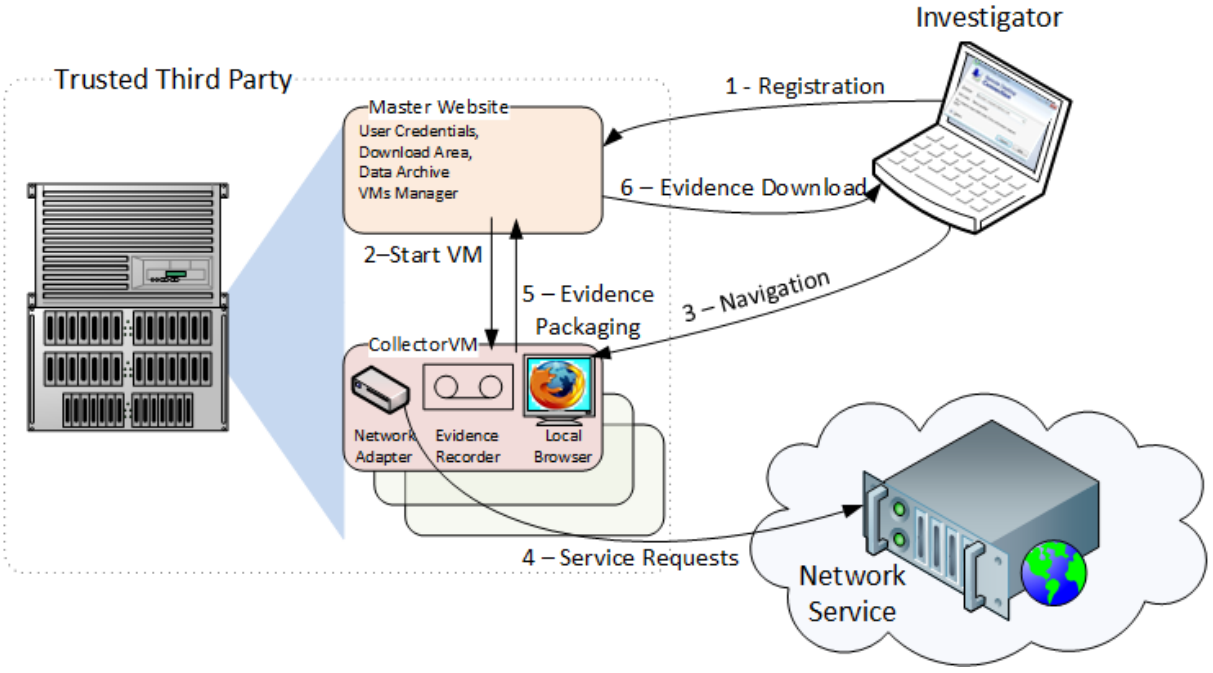


Figure 4: Architecture of the LINEA prototype.

produced by the TTP is recorded and made available for further accesses.

After the registration, the user logs into the system and can start an acquisition session (namely a new *case*). When a new case is started, a dedicated VM (CollectorVM) is created from a predefined template (step 2). The new machine has just one local account which has been set up with the same credentials chosen during the step 1. Therefore only the investigator I can log on this instance of CollectorVM. When the CollectorVM is ready, the investigator is notified with an email. By means of the Remote Desktop Protocol (RDP), the investigator can start a remote session on the VM just clicking on the hyperlink specified in the received email. In this way, he is automatically connected to the remote desktop interface of the CollectorVM and a login screen is presented (step 3).

After logging on the CollectorVM an initialization script automatically sets up the environment and starts the acquisition tools. The standard output and error channels are redirected to a log file which reports all the user activities. The remote display service is provided by means of `XRDP` [53], an open-source RDP server. The communication channel between the investigator's workstation and the CollectorVM is secured by means of a SSL like protocol with server side authentication. The initialization script starts also the screen recorder software which is able to record everything is shown on the screen, including mouse pointer and keyboard typing. In particular, the `recordMyDesktop` [54] tool has been chosen because of its reliability, flexibility (resolution and frame rate can be set accordingly to the required performance) and particularly because it produces files in open formats such as `theora` for video and `vorbis` for audio. The resulting `ogg` container can be read by any open-source players. It is worth noting that `recordMyDesktop`, in order to record all the screen output, takes the input directly from the video adapter memory. As result the video will be a faithful replica of what the investigator has seen during the acquisition session. Also the output audio stream is recorded by reading the memory buffer of the audio server (ALSA standard driver). Many parameters can be chosen depending on the overall system performance. Currently, the video is recorded with a resolution of $1,280 \times 1,024$, a color depth of 16 and a rate of 10 fps. Moreover, the `--on-the-fly-encoding` option of the recordMyDesktop command is used in order to preserve the acquired copy whenever a system crash would happen.

The initialization script also starts a packet sniffer and a daemon in charge of monitoring the network status. In particular, `tcpdump` is executed in background for the entire acquisition session and collects all the traffic flowing through the network adapter of the CollectorVM. The RDP traffic to/from the investigator's workstation is discarded. No other filtering is required since the virtual network adapter is not shared. The network status daemon is in charge of collecting information about the hosts that have been accessed by the investigator during the acquisition. In particular, it runs tools like `nslookup`, `traceroute` and `whois`. In order to cope with unexpected events such as connection lost, a background process is started to control two events: inactivity timeout and maximum session length. When a fixed treshold is reached the whole session is automatically closed.

At the end of the initialization script after the login into the CollectorVM, a full screen instance of a predefined browser is presented to the user. The browser Firefox [55] has been chosen for this experiment. It has been customized in order to operate in kiosk mode with a configuration which allows only Internet navigation. Any other actions such as file system access (`file://`), configuration menus and keyboard shortcuts have been disabled to prevent unexpected actions (like using the shell or deleting files). Even the options "Open in a new tab" and "Open in a new window" of the hyperlink contextual menu have been blocked in order to avoid data loss due to overlapping frames. This has been considered necessary in order to ensure that everything

received and rendered by the browser is also recorded in the video produced by the TTP. The internal browser is connected to a local HTTP/HTTPS proxy (a customized version of `squid` [56]), which generates the list of all the hosts that have been contacted. In particular, each time a new host is accessed, it is passed to the network status daemon in order to collect network information about it.

Once the session with the CollectorVM has been opened, the investigator *I* can navigate through any website (step 4), eventually providing credentials to access restricted websites (such as online social networks) and scrolling the browser window in order to seek information of interest. This process does not affect and does not depend on the technology used by the remote website. The behavior of the browser is exactly the same as the user would be using his local browser.

When the investigator closes the browser application, the `terminate.sh` script executes all the procedures needed to guarantee robustness to the acquired LNE. In particular, all the files generated by the various acquisition tools are saved in a compressed archive. Subsequently, the package is digitally signed and timestamped. If the user during the registration provided a digital certificate with a RSA public key, the whole package is encrypted with the RSA algorithm. At the end of this process an email is sent (step 5) to notify the user that the final package is ready to be downloaded from his private area (step 6). Of course the packaging process can be time and space consuming depending on the amount of the collected data. For example, a session lasting 15 minutes, in which the user accessed ten websites and viewed a 2 minutes video from YouTube, required about 5 minutes for the generation of a package sized 400 MB. The CollectorVM is completely destroyed as soon as the process terminates.

In order to enforce integrity and security, `LINEA` provides the investigator with a fresh CollectorVM instance each time a new acquisition session in initialized. This ensures that, in case the CollectorVM is corrupted or compromised during an acquisition session, the problem is not propagated. Using a dedicated VM also gives many advantages in term of reliability, robustness and scalability. In particular, the following features are guaranteed:

- **Isolation** In case two or more investigators concurrently use the `LINEA` service, the virtualization technology guarantees a better isolation with respect to a multiuser system. In fact, isolation among different VMs is more robust and reliable than isolation among processes provided within an operating system.
- **Packet filtering** acquisition of network traffic is straightforward and trustworthy. In fact, if a different VM is assigned to each user, then we are sure that the traffic flowing through its virtual network adapter is all and only that generated by that particular user.
- **Safeness** The use of a temporary VM improves the safeness of the acquisition. In fact, if a new CollectorVM is generated for each acquisition session, the system is not permanently exposed to threats from the Internet.
- **Scalability** Finally, the use of VM technology provides scalability to the service whereas many concurrent users could access the system.

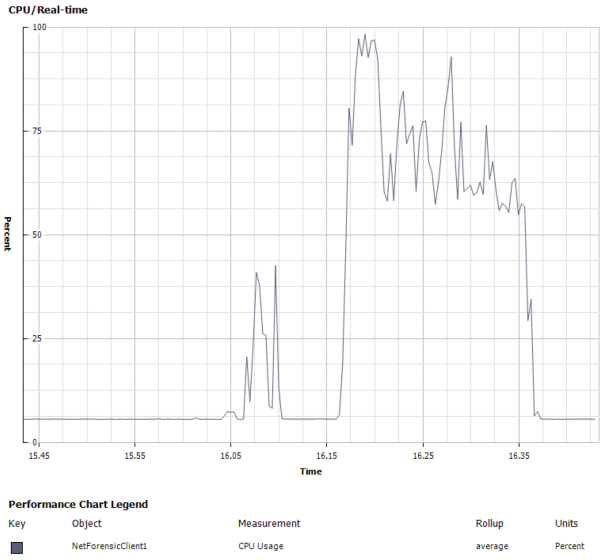


Figure 5: CPU used during the navigation.

6.1 Performance Evaluation

In order to state the effective quality, robustness and scalability of our methodology, the `LINEA` prototype has been extensively tested. An experimental evaluation of its performance has been reported in this section. In particular, the following six Key Performance Indicators (KPIs) has been considered: CPU usage, memory usage, network usage, disk usage, video quality and overall file size.

The following procedure has been followed for the experimentation. First, three categories of web services have been identified: (1) static websites with limited graphical elements (e.g., the website of the University of Salerno), (2) highly-dynamical websites with rich graphic content (e.g., Facebook), (3) rich multimedia resources (e.g., video streaming on YouTube). Three test cases have been prepared, each involving the acquisition of information from a mixed set of online services belonging to these categories. Then, these test cases have been executed by means of `LINEA` and the aforementioned KPIs have been actually measured.

In this section the following test case is considered. At 16:15, the investigator *I* visited YouTube and viewed a video lasting about 3 minutes. At 16:19, the user logged in the Facebook website and browsed his personal gallery containing about 30 pictures. After about 5 minutes, the user visited his personal diary and scrolled the page down in order to read old notifications. At 16:28, a chatting session on Facebook, lasting about 2 minutes, has been carried out. Finally, at 16:30, the user visited the website of the University of Salerno, browsing internal web pages for about 5 minutes.

A summary of the performance related to CPU usage, memory usage, network usage and disk usage are reported in Figures 5, 6, 7 and 8 respectively, where time information is reported on the abscissa axis.

All the experiments have been carried out on a Dell Power Edge R900 with 4 quad-core processors Intel Xeon X7350 (for a total of 16 cores) at 2.93 Ghz, with 32 GB of RAM and VMWare ESXi as virtualization software. Each CollectorVM activated by `LINEA` has been configured with 1 cpu and 2 GB of RAM.

In more details, the experiment produced the following results:

1) The CollectorVM consumed more then 80% of the CPU

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2017.2683521, IEEE Transactions on Big Data

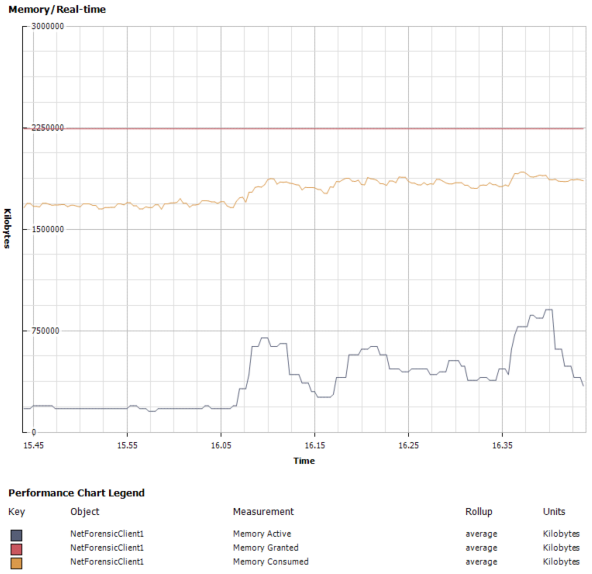


Figure 6: Memory used during the navigation.

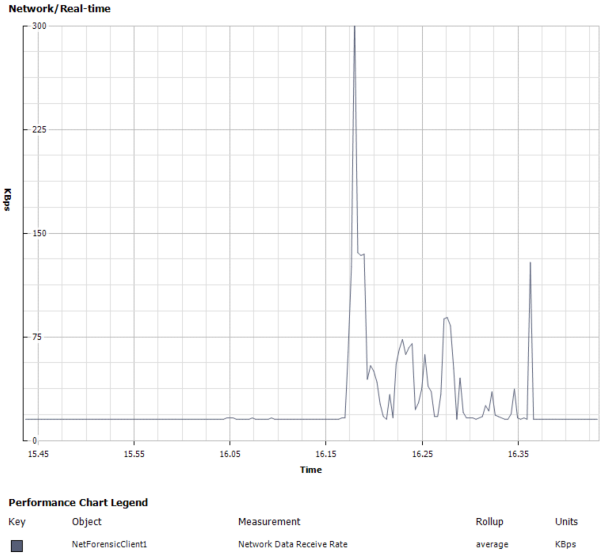


Figure 7: Network used during the navigation.

only whilst viewing the video on YouTube. During the other tasks, the amount of used CPU was always less the 60%, even during the compression and the encryption phases performed for the package generation. The CPU usage peak is mainly due to the screen recording and encoding.

2) Concerning the memory usage, all the processes never exceeded the 0.75 GB of active memory, while the overall allocated memory size was about 1.6 GB. Therefore, considering CPU and memory usage, it can be stated that up to 16 CollectorVMs could be concurrently executed on the test configuration.
3) The graphic related to the network traffic shows a peak in the bandwidth usage during the video streaming. During the other tasks the bandwidth usage is very limited. Even if not reported in the graphic, it is worth noting that the RDP used a huge amount of bandwidth (up to 7 Mbps) in order to update the remote display with the same frame rate of the video played on YouTube. However, as mentioned in the previous sections, the traffic between user and TTP is not included in the acquisition.
4) The disk R/W rate was very low for the entire duration of the test case. Only during the package finalization (compression, encryption and signature) a throughput of about 9 MB/sec has been necessary.

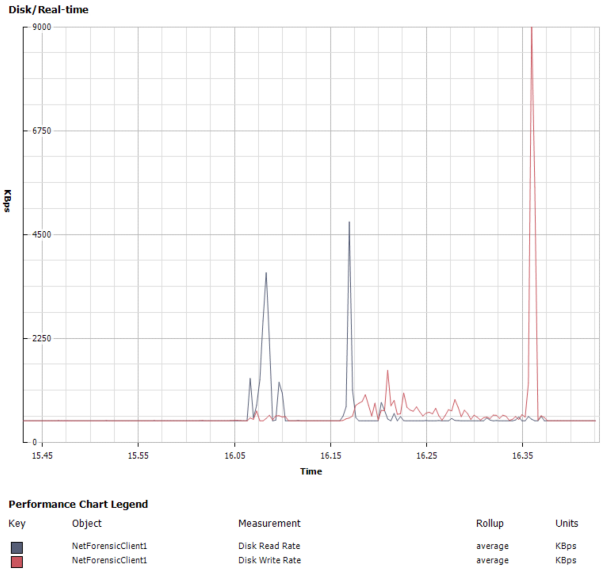


Figure 8: Disk throughput during the navigation.

The experiment produced a package of about 100 MB. The size turned out to be very large compared to other test cases, mainly due to the screen recording. In fact, the video encoder was not able to compress the video when using YouTube due to the fast frame rate. Table 1 summarizes the size of packages resulting from different experiments, consisting of a 2 minutes acquisition from different categories of services ranging from static websites to rich multimedia resource.

Table 1: Resulting package size for 2 minutes of acquisition

	Light Website	**Interactive Website: Facebook**	**Streaming: YouTube native res**	**Streaming: YouTube full screen**
.pcap file size	0.85 MB	15.20 MB	11.15 MB	11.15 MB
video file size	9.76 MB	37.47 MB	17.22 MB	17.87 MB
ratio	11.48	2.46	1.54	1.60

7 Future Directions

In this section some possible enhancement of the methodology presented so far are discussed. In particular, we concentrate on the access through proxy and the multiple identities.

Access through proxy: One of the limitations of the LINEA prototype is that the same public IP address is assigned to each CollectorVM. The maintainer of a web service could blacklist this address in order to divert the investigation. For example, a web server could be instructed to reply with a different resource if the originator of the request is blacklisted or if it is known a

priori. This technique is typically implemented by exploit kits [57] in order to thwart detection. This problem can be bypassed by providing the investigator with the capability of using a series of proxies in order to access the target service.

Multiple identities: The use of multiple identities for the acquisition of the same information can be useful in order to enhance the correlation property defined by Nikkel, as well as to determine if the response of the service under investigation — and therefore the availability of the target information — depends on specific characteristics of the client. In order to cope with these situations, a scenario in which multiple proxies are used by the TTP to access the target service could be considered.

A typical situation where the use of multiple identities may be useful is when a web server returns the content of a web page in different languages based on the geographical location of the client. In such a case, an evidence (e.g., an injury) could be only obtained if the IP address of the client belongs to a specific range. Another example is when the target information resides on an anonymizing network such as TOR. In such a case, the service containing the evidence is only accessible if the TTP is connected to the same anonymizing network (such as TOR).

The acquisition with multiple identities could be provided in a automatic way or in a manual way. In the first case, the TTP may programmatically change its identity (e.g., by switching proxy from a predefined list), while in the second case the investigator could manually customize the identity of the TTP at runtime (e.g., by changing the proxy settings). Moreover, the acquisition with multiple identities can be exclusive or concurrent. The second case is tricky, since the TTP should be able to multiplex a single request of the investigator into n simultaneous requests to be forwarded to different proxies. Moreover, each proxy should acquire the own user view, since it may vary depending on the specific response.

In general, the TTP should operate as a multiplexer, forwarding each user request to n ($n \geq 2$) cooperating proxies which are in charge of capturing the specific response of the target service S. It is easy to see that the bigger is the number of replicas, the higher is the robustness of the system and the service will continue to produce reliable results even if a number of $k \leq n/2$ proxies have been corrupted or attacked.

8 Conclusions

This paper presents a novel method for the acquisition of LNE from online services. It is based on a TTP which is in charge of collecting information on behalf of the investigator. The methodology provides three different operating modes depending on the technical skills of the investigator. The LNE acquisition is driven by the investigator by means of a remote control interface. In addition to the network flow, the user view (audio/video information generated by the processing of the data received from the service) is atomatically acquired. The methodology has been proven to be forensically-sound. In other words, the evidence collected by the TTP is robust and its reliability can be verified at any time after the acquisition. The robustness property is enhanced by the correlation of information from multiple sources of evidence, while the reliability is provided by encrypting, timestamping and digitally-signing the result of the acquisition.

As side effect, the proposed methodology may be exploited to forensically certify that a given resource (i.e., a web page or any kind of document) has been effectively published on a specific website (in a known time interval). This is a very common task for forensics investigators but in many circumstances many problems may arise during their activities.

Finally, a possible extension of the methodology has been proposed in order to enhance the overall robustness of the system. It makes use of multiple proxies in order to correlate the response of the target service to multiple originators. The methodology introduced in this paper can be used to uncover a recent kind of attack which is very difficult to detect and may undermine the trustworthiness of the digital evidence and the Digital Forensics world in general: the false digital alibi [58], [59], [60], [61], [62]. Using the proposed methodology, an investigator is able to better understand the state of a network where such a kind of attack may be perpetrated.

Acknowledgements

Authors would like to thank their friends from IISFA (International Information System Forensics Association) for their support, their valuable suggestions and useful discussions during the research phase. In particular their thanks go to Francesco Cajani (Deputy Public Prosecutor High Tech Crime Unit Court of Law in Milano, Italy) and Gerardo Costabile (President of IISFA Italian Chapter) for giving authors precious ideas that have been considered during the engineering of the proposed solution.

References

[1] ACPO Computer Crime Group, "Good practice guide for computer based evidence," Association of Chief Police Officers, Tech. Rep., 1999.

[2] NIST, "Disk imaging tool specification," Computer Forensics Tool Testing (CFTT) Project, Tech. Rep., 2001.

[3] Computer Crime and Intellectual Property Section, Criminal Division, "Searching and seizing computers and obtaining electronic evidence in criminal investigations," U.S. Department of Justice, Tech. Rep., 2002.

[4] National Institute of Justice (USA), "Digital Forensics Standards and Capacity Building," (Available from: http://nij.gov/topics/forensics/evidence/digital/standards/welcome.htm), [Accessed on 17 October 2012].

[5] W. Kruse and J. Heiser, *Computer Forensics: Incident Response Essentials*. Pearson Education, 2001. [Online]. Available: http://books.google.it/books?id=-qWa5Svv7BIC

[6] M. Sheetz, *Computer Forensics: An Essential Guide for Accountants, Lawyers, and Managers*. Wiley, 2007.

[7] D. Quick and K.-K. R. Choo, "Digital droplets: Microsoft SkyDrive forensic data remnants," *Future Generation Computer Systems*, vol. 29, no. 6, pp. 1378 – 1394, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X13000265

[8] ——, "Google Drive: Forensic analysis of data remnants," *Journal of Network and Computer Applications*, vol. 40, pp. 179 – 193, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1084804513002051

[9] L. Wang, S. Tasoulis, T. Roos, and J. Kangasharju, "Kvasir: Scalable Provision of Semantically Relevant Web Content on Big Data Framework," *IEEE Transactions on Big Data*, vol. PP, no. 99, pp. 1–1, 2016.

[10] W. Dai, L. Qiu, A. Wu, and M. Qiu, "Cloud Infrastructure Resource Allocation for Big Data Applications," *IEEE Transactions on Big Data*, vol. PP, no. 99, pp. 1–1, 2016.

[11] B. Blakeley, C. Cooney, A. Dehghantanha, and R. Aspin, "Cloud Storage Forensic: hubiC as a Case-Study," in *7th IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2015, Vancouver, BC, Canada, November 30 - Dec. 3, 2015*. IEEE Computer Society, 2015, pp. 536–541. [Online]. Available: http://dx.doi.org/10.1109/CloudCom.2015.24

[12] S. Nepal, R. Ranjan, and K. R. Choo, "Trustworthy Processing of Healthcare Big Data in Hybrid Clouds," *IEEE Cloud Computing*, vol. 2, no. 2, pp. 78–84, 2015. [Online]. Available: http://dx.doi.org/10.1109/MCC.2015.36

[13] L. Zhao, L. Chen, R. Ranjan, K. R. Choo, and J. He, "Geographical information system parallelization for spatial big data processing: a review," *Cluster Computing*, vol. 19, no. 1, pp. 139–152, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10586-015-0512-2

[14] D. Quick, .-a. Martini, Ben, a. Choo, Kim-Kwang Raymond, and EBSCOhost, *Cloud Storage Forensics*. Amsterdam Boston Elsevier/Syngress, 2014. [Online]. Available: http://site.ebrary.com/id/10810980

[15] N. H. Ab Rahman, N. D. W. Cahyani, and K.-K. R. Choo, "Cloud incident handling and forensic-by-design: cloud storage as a case study," *Concurrency and Computation: Practice and Experience*, pp. n/a–n/a, 2016. [Online]. Available: http://dx.doi.org/10.1002/cpe.3868

[16] N. D. W. Cahyani, B. Martini, K.-K. R. Choo, and A. M. N. Al-Azhar, "Forensic data acquisition from cloud-of-things devices: windows smartphones as a case study," *Concurrency and Computation: Practice and Experience*, pp. n/a–n/a, 2016. [Online]. Available: http://dx.doi.org/10.1002/cpe.3855

[17] B. Martini and K.-K. R. Choo, "Distributed filesystem forensics: XtreemFS as a case study," *Digital Investigation*, vol. 11, no. 4, pp. 295 – 313, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287614000942

[18] ——, "Cloud storage forensics: ownCloud as a case study," *Digital Investigation*, vol. 10, no. 4, pp. 287 – 299, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287613000911

[19] D. Quick and K.-K. R. Choo, "Forensic collection of cloud storage data: Does the act of collection result in changes to the data or its metadata?" *Digital Investigation*, vol. 10, no. 3, pp. 266 – 277, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287613000741

[20] B. Martini and K. K. R. Choo, "Remote Programmatic vCloud Forensics: A Six-Step Collection Process and a Proof of Concept," in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, Sept 2014, pp. 935–942.

[21] D. Quick and K.-K. R. Choo, "Impacts of increasing volume of digital forensic data: A survey and future research challenges," *Digital Investigation*, vol. 11, no. 4, pp. 273 – 294, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287614001066

[22] A. Castiglione, G. Cattaneo, and A. De Santis, "A forensic analysis of images on Online Social Networks," *3rd IEEE International Conference on Intelligent Networking and Collaborative Systems, INCoS 2011*, pp. 679–684, 2011.

[23] S. Jiang, X. Qian, T. Mei, and Y. Fu, "Personalized Travel Sequence Recommendation on Multi-Source Big Social Media," *IEEE Trans. Big Data*, vol. 2, no. 1, pp. 43–56, 2016. [Online]. Available: http://dx.doi.org/10.1109/TBDATA.2016.2541160

[24] M. Damshenas, A. Dehghantanha, R. Mahmoud, and S. bin Shamsuddin, "Forensics investigation challenges in cloud computing environments," in *2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic, CyberSec 2012, Kuala Lumpur, Malaysia, June 26-28, 2012*. IEEE, 2012, pp. 190–194. [Online]. Available: http://dx.doi.org/10.1109/CyberSec.2012.6246092

[25] D. Quick and K. R. Choo, "Big forensic data reduction: digital forensic images and electronic evidence," *Cluster Computing*, vol. 19, no. 2, pp. 723–740, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10586-016-0553-1

[26] C. J. D'Orazio, K. K. R. Choo, and L. T. Yang, "Data exfiltration from internet of things devices: ios devices as case studies," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2016.

[27] Q. Do, B. Martini, and K. K. R. Choo, "A Cloud-Focused Mobile Forensics Methodology," *IEEE Cloud Computing*, vol. 2, no. 4, pp. 60–65, July 2015.

[28] B. J. Nikkel, "Generalizing sources of live network evidence," *Digital Investigation*, vol. 2, no. 3, pp. 193 – 200, 2005.

[29] Vere Software, "WebCase 2.0 Features," (Available from: http://veresoftware.com/index.php/webcase_overview/webcase/WebCase-2-0-Features), 2016, [Accessed on 27 June 2016].

[30] G. Amato, "Hashbot by Digital-Security.IT," (Available from: https://www.hashbot.com/), 2016, [Accessed on 27 June 2016].

[31] B. J. Nikkel, "A portable network forensic evidence collector," *Digital Investigation*, vol. 3, no. 3, pp. 127 – 135, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1742287606001022

[32] R. McKemmish, "When is Digital Evidence Forensically Sound?" in *Advances in Digital Forensics IV*, ser. IFIP International Federation for Information Processing, I. Ray and S. Shenoi, Eds. Springer Boston, 2008, vol. 285, pp. 3–15, 10.1007/978-0-387-84927-0_1.

[33] S. Davidoff and J. Ham, *Network Forensics: Tracking Hackers Through Cyberspace*. Prentice Hall, 2012.

[34] Council of Europe, "Convention on cybercrime," (Available from: http://conventions.coe.int/Treaty/en/Treaties/Html/185.htm), 2001, [Accessed on 27 June 2016].

[35] VereSoftware, "WebCase WebLog," (Available from: http://veresoftware.com/blog/?page_id=269), 2014, [Accessed on 27 June 2016].

[36] The Tor Project, Inc., "Tor Project: Anonymity Online," (Available from: https://www.torproject.org/), 2016, [Accessed on 27 June 2016].

[37] W3C, "HTML5," (Available from: https://www.w3.org/TR/2014/REC-html5-20141028/), 2014, [Accessed on 15 June 2016].

[38] The Tcpdump Team, "Tcpdump/libpcap public repository," (Available from: http://www.tcpdump.org/), 2016, [Accessed on 27 June 2016].

[39] Wireshark Foundation, "Wireshark - go deep." (Available from: http://www.wireshark.org/), 2016, [Accessed on 27 June 2016].

[40] Riverbed Technology, "WinPcap - Home," (Available from: http://www.winpcap.org/), 2013, [Accessed on 27 June 2016].

[41] S. Liu, Q. Qu, L. Chen, and L. M. Ni, "SMC: A Practical Schema for Privacy-Preserved Data Sharing over Distributed Data Streams," *IEEE Trans. Big Data*, vol. 1, no. 2, pp. 68–81, 2015. [Online]. Available: http://dx.doi.org/10.1109/TBDATA.2015.2498156

[42] Z. Yan, W. Ding, X. Yu, H. Zhu, and R. H. Deng, "Deduplication on Encrypted Big Data in Cloud," *IEEE Trans. Big Data*, vol. 2, no. 2, pp. 138–150, 2016. [Online]. Available: http://dx.doi.org/10.1109/TBDATA.2016.2587659

[43] M. Conti, L. V. Mancini, R. Spolaor, and N. V. Verde, "Analyzing Android Encrypted Network Traffic to Identify User Actions," *IEEE Trans. Information Forensics and Security*, vol. 11, no. 1, pp. 114–125, 2016. [Online]. Available: http://dx.doi.org/10.1109/TIFS.2015.2478741

[44] Internet Archive, "Internet Archive: Digital Library of Free Books, Movies, Music & Wayback Machine," (Available from: http://archive.org), 2016, [Accessed on 27 June 2016].

[45] Pearl Crescent, "Page Saver," (Available from: http://pearlcrescent.com/products/pagesaver/), 2016, [Accessed on 27 June 2016].

[46] G. Costa and A. D. Franceschi, "Xplico - open source network forensic analysis tool (nfat)," (Available from: http://www.xplico.org/), 2013, [Accessed on 27 June 2016].

[47] B. J. Nikkel, "Improving evidence acquisition from live network sources," *Digital Investigation*, vol. 3, no. 2, pp. 89 – 96, 2006.

[48] A. Castiglione, B. D'Alessio, and A. De Santis, "Steganography and secure communication on online social networks and online photo sharing," *International Conference on Broadband and Wireless Computing, Communication and Applications, BWCCA 2011*, pp. 363–368, 2011.

[49] A. Castiglione, B. D'Alessio, A. De Santis, and F. Palmieri, "New steganographic techniques for the OOXML file format," *Lecture Notes in Computer Science*, vol. 6908 LNCS, pp. 344–358, 2011.

[50] DEFT Association, "DEFT Linux - Computer Forensics live CD," (Available from: http://www.deftlinux.net/), 2015, [Accessed on 15 June 2016].

[51] F. Palmieri and U. Fiore, "Audit-based access control in nomadic wireless environments," *Lecture Notes in Computer Science*, vol. 3982 LNCS, pp. 537–545, 2006.

[52] F. Palmieri, U. Fiore, and A. Castiglione, "Automatic security assessment for next generation wireless mobile networks," *Mobile Information Systems*, vol. 7, no. 3, pp. 217–239, 2011.

[53] Jay Sorg, "xrdp: An open source remote desktop protocol (rdp) server," (Available from: http://www.xrdp.org/), 2016, [Accessed on 27 June 2016].

[54] recordMyDesktop, "About recordMyDesktop," (Available from: http://recordmydesktop.sourceforge.net/about.php), 2016, [Accessed on 27 June 2016].

[55] Mozilla Foundation, "Mozilla Firefox Web Browser," (Available from: https://www.mozilla.org/en-US/firefox/products/), 2016, [Accessed on 27 June 2016].

[56] Squid Software Foundation, "Squid: Optimising Web Delivery," (Available from: http://www.squid-cache.org/), 2016, [Accessed on 27 June 2016].

[57] V. Kotov and F. Massacci, "Anatomy of exploit kits: Preliminary analysis of exploit kits as software artefacts," in *Proceedings of the 5th International Conference on Engineering Secure Software and Systems*, ser. ESSoS'13. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 181–196. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-36563-8_13

[58] A. De Santis, A. Castiglione, G. Cattaneo, G. De Maio, and M. Ianulardo, "Automated construction of a false digital alibi," *Lecture Notes in Computer Science*, vol. 6908 LNCS, pp. 359–373, 2011.

[59] A. Castiglione, G. Cattaneo, G. De Maio, A. De Santis, G. Costabile, and M. Epifani, "The forensic analysis of a false digital alibi," *6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2012*, pp. 114–121, 2012.

[60] A. Castiglione, G. Cattaneo, G. De Maio, and A. De Santis, "Automated production of predetermined digital evidence," *Access, IEEE*, vol. 1, pp. 216–231, 2013.

[61] A. Castiglione, G. Cattaneo, R. De Prisco, A. De Santis, and K. Yim, "How to forge a digital alibi on Mac OS X," *Lecture Notes in Computer Science*, vol. 7465 LNCS, pp. 430–444, 2012.

[62] P. Albano, A. Castiglione, G. Cattaneo, G. De Maio, and A. De Santis, "On the construction of a false digital alibi on the Android OS," *3rd IEEE International Conference on Intelligent Networking and Collaborative Systems, INCoS 2011*, pp. 685–690, 2011.

Aniello Castiglione (S'04-M'08) received the Ph.D. degree in Computer Science from the University of Salerno (Italy). Actually he is an adjunct professor at the University of Salerno (Italy) and at the University of Naples "Federico II" (Italy). He received the Italian national qualification as Associate Professor in Computer Science. He published more than 150 papers in international journals and conferences. He served in around 130 international conferences as Program Chair, General Chair and TPC member. One of his papers has been selected as "Featured Article" in the IEEE Cybersecurity initiative. He served as a Reviewer for several international journals and he is the Managing Editor of two ISI-ranked international journals. He acted as a Guest Editor in several journals and serves as Associate Editor in several editorial boards of international journals. His current research interests include Information Forensics, Digital Forensics, Security and Privacy on Cloud, Communication Networks, and Applied Cryptography. He is a member of several associations, including IEEE and ACM. He has been involved in forensic investigations, collaborating as a consultant with several law enforcement agencies.

Giuseppe Cattaneo received a degree in computer science from the Università di Salerno in 1983. Since 1986 he has been Research Associate with the Dipartimento di Informatica ed Applicazioni of the University of Salerno, where he is currently a Associate Professor. From 1987 to 1990, he has been a Visiting Researcher at Laboratoire d'Informatique Thèorique et Programmation (LITP), Universite Paris 6, Paris, France, working on a project aimed to the development of the first European Parallel Lisp Machine. Since 1993, he has been involved in research activities on experimental algorithm evaluation, algorithm engineering, system security and digital forensics. Since 2009, his research group, in collaboration with Italian Police (CNCPO) has been working on source camera identification applied on pictures coming from online social networks and generally on image forensics issues. In the last ten years, he has been the leader of the local team for 8 ICT projects co-funded by national large companies.

Giancarlo De Maio is a R&D engineer at Lastline, Inc. He got his PhD from the University of Salerno (Italy), after an internship at the University of California, Santa Barbara (USA). His professional experience include software design and development, automation, networking and information security. His main research interests are digital forensics, web security and mobile security.

Alfredo De Santis received a Degree in computer science (cum laude) from the the University of Salerno, Salerno, Italy, in 1983. Since 1984, he has been with the Dipartimento di Informatica ed Applicazioni, Università di Salerno. Since 1990, he has been a Professor of computer science. From November 1991 to October 1995 and from November 1998 to October 2001, he was the Chairman of the Dipartimento di Informatica ed Applicazioni, University of Salerno. From November 1996 to October 2003, he was the Chairman of the PhD Program in computer science with the University of Salerno. From September 1987 to February 1990, he was a Visiting Scientist at IBM T. J. Watson Research Center, Yorktown Heights, NY, USA. He was with the International Computer Science Institute (ICSI), Berkeley CA, USA, in 1994, as a Visiting Scientist. From November 2009 to October 2012, he was with the Board of Directors of Consortium GARR (the Italian Academic & Research Network). His current research interests include algorithms, data security, cryptography, information forensics, communication networks, information theory, and data compression.

Gianluca Roscigno received in 2010 the Bachelor's Degree cum laude in Computer Science from University of Salerno, and in 2012 the Master's degree cum laude in Computer Science from University of Salerno. He received in 2016 the Ph.D. in Computer Science at the University of Salerno under the supervision of the Prof. Giuseppe Cattaneo. Since 2016 he is Research Fellow at the Department of Computer Science of the University of Salerno, and he is a member of "Benchmarking & Algorithm Engineering" Laboratory of the Prof. Giuseppe Cattaneo. His research interests focus on digital forensics, image and video forensics, distributed computing, security, experimental analysis of algorithms, algorithm engineering and bioinformatics.