Abstract. The proliferation of smartphones introduces new opportunities in digital forensics. One of the reasons is that smartphones are usually equipped with sensors (e.g., accelerometer, proximity sensor, etc.), hardware which can be used to infer the user’s context. This context may be useful in a digital investigation, as it can aid in the rejection or acceptance of an alibi, or even reveal a suspect’s actions or activities. Nonetheless, sensor data are volatile, thus are not available in post-mortem analysis. Thus, the only way to timely acquire them, in case such a need arises during a digital investigation, is by software that collects them when they are generated by the suspect’s actions. In this paper we examine the feasibility of ad-hoc data acquisition from smartphone sensors by implementing a device agent for their collection in Android, as well as a protocol for their transfer. Then, we discuss our experience regarding the data collection of smartphone sensors, as well as legal and ethical issues that arise from their collection. Finally, we describe scenarios regarding the agent’s preparation and use in a digital investigation.

Keywords: Smartphones, Digital Forensics, Investigation Models, Sensor, Sensor Data, Information Gathering, Android.

1 Introduction

The proliferation of smartphones - devices with increasing popularity due to their processing and network capabilities, low cost, and dimensions - introduces new challenges to digital forensics. This holds true, as the heterogeneity and progressive evolution of smartphone platforms increase the complexity of a digital investigation, together with the skills and toolset required to respond to a crime or incident. For instance, the different hardware and software, which is offered by smartphone platforms to increase their market share, hinder a forensic examination. This happens since dedicated software has to be implemented and cables to be acquired per case. Moreover, as smartphone cables (data or power cables) tend to be available only as long as a smartphone model is in the market. More difficulties appear in the examination of old devices that are not supported anymore.

Along with the popularity of smartphones, the one of sensors (i.e., hardware that measures the user’s context) is also increasing in third-party software, such as games and navigation apps. In a previous paper we have argued that sensor data are a data source which can be collected for the purpose of digital forensics (Mylonas et al., 2012b). Sensor data may be useful in certain crimes in order to lawfully deduce the user’s context. This context may provide crucial information for law enforcement authorities in a digital investigation both for the detection and sanction and the prevention of crime. In the same time, this kind of data collection is often intrusive (Mylonas et al., 2012c) and may raise ethical and legal issues pertaining to the
secrecy of telecommunication and the privacy of the concerned persons. Due to the interference with these fundamental rights, such forms of surveillance and collection of the respective digital evidence may be implemented only in relation to crimes that are regarded by the law as “severe” or “national security threats”. A generalized and indiscriminate collection of probably useful evidence against individuals at random would perceive all citizens as potentially future wrongdoers. Such a model of routine intelligence gathering against random citizens could “blow up the cornerstones of the rule of law state” (Hoffmann-Riem, 2002).

Smartphone forensics is an emerging discipline which currently appears to be still in its infancy. Nonetheless, the consumerization of smartphones has triggered the forensic community to focus on these devices’ technical details as well as the collection of their data. Therefore, tools have been created which perform either physical (i.e. low level, bit to bit), or logical (i.e. data organized in logical objects, e.g. directories) data acquisition in: (a) Windows Mobile (Casey et al., 2010; Klaver, 2010; Rehault, 2010; Grispos, 2011), (b) Symbian (Mokhonoana and Olivier, 2007; Distefano and Me, 2008; Pooters, 2010; Thing and Chua, 2012; Thing and Tan, 2012), (c) iOS (Zdziarski, 2008; Bader and Baggili, 2010; Husain and Sridhar, 2010; Jung et al., 2011), and (d) Android (Lessard and Kessler, 2010; Thing et al., 2010; Vidas et al., 2011; Hoog, 2011; Mutawa et al., 2012; Park et al., 2012; Sylve, 2012).

Until now the practicality of sensor data collection has not been adequately studied in the forensics literature. This is because they are time-sensitive and volatile data and therefore not available for analysis during post-mortem forensics. In (Mylonas et al., 2012b), we proposed a scheme for the ad-hoc collection of smartphone data. In specific cases and under certain conditions, the law may allow collection of smartphone data for proactive use and for the purpose of crime prevention. Furthermore, proactive collection of a smartphone’s sensor data may take place in critical infrastructures for incident response and forensic readiness reasons - when the device is considered part of this infrastructure and the use of such intrusive control fully complies with the organization’s security policy, its culture and the existing legal framework. For instance, the security policy of a critical infrastructure may specify employee usage profiling - assuming the employee’s informed consent - in order to mitigate the insider threat via an insider prediction model (Kandias, et al., 2010).

The paper contributes the following:
(a) Development of an investigation frameworks taxonomy that can be applied in smartphone forensics.
(b) Study of the smartphone sensors as a potential data source for digital forensics.
(c) Identification of vulnerabilities in Android’s security model that enables the ad-hoc acquisition of sensors data.
(d) Implementation in the Android of the acquisition software and the transport protocol of the proactive investigation scheme that was proposed in (Mylonas et al., 2012b).
(e) Evaluation of the implementation in a laboratory environment and sharing of the experience regarding the collection of sensor data.
(f) In depth analysis of legal and ethical issues that arise from the ad-hoc collection of sensor data.

The paper is organized as follows. The next section includes taxonomy of the digital investigation frameworks that can be used in smartphone forensics. Section 3 elaborates on sensor data in the context of digital forensics. In section 4, we present our implementation for remote, ad-hoc sensor data acquisition from smartphones. Then, we summarize our experience regarding the data collection of smartphone sensors by our tool and describe a number of scenarios regarding its use and preparation (section 5). Section 6, includes the ethical and legal issues that arise from the ad-hoc data acquisition from smartphone sensors. Finally, section 7 includes a discussion of our approach and conclusions.

2 Digital forensic investigation frameworks

A Digital Forensic Investigation Framework (DFIF) describes procedures that take place before, during, and after the investigation of physical or digital crime scenes. These frame-
works refer to methodologies and techniques capable of collecting, analyzing and interpreting
digital evidence that is suitable for a forensic investigation. This investigation may be triggered
either from internal sources, e.g. in a corporate investigation, or from external ones, e.g.
the evidence will be presented to support a hypothesis in courts of law.

This section focuses on the DFIF that have been proposed, and are suitable, for use in cell
phone forensics, i.e. digital evidence is collected from a cell phone - either feature phone or
smartphone (Theoharidou et al., 2012). We organize their presentation, starting from abstract
DFIF to more device-specific frameworks, i.e. in: (a) abstract DFIF, (b) DFIF for feature pho-
nes, and c) DFIF for smartphones.

2.1 Abstract DFIF

Abstract investigation frameworks are widely used for the collection and analysis of digi-
tal evidence without focusing on the devices’ technological ‘nuts and bolts’. These frame-
works progressively build upon each other. Their procedures are commonly used as a basis
for the construction of subsequent frameworks that are more technologically-specific.

Reith et al. (2002) propose an abstract investigation framework that includes nine phases,
namely: identification, preparation, approach strategy, preservation, collection, examination,
analysis, presentation and returning evidence. The authors characterize their work, which ex-
tends the work from the Digital Forensics Research Working Group (DFRWS, 2001), as abst-
tract since it is independent of technology and crime type.

The investigation framework proposed in (Casey, 2004), includes seven steps, namely:
authorization and preparation, identification, documentation, collection and preservation, exa-
mination and analysis, reconstruction, and reporting. Documentation is an on-going process
that takes place in all of the framework’s phases as a basic property of the investigation and is
not considered as a standalone phase.

Carrier and Spafford (2003; 2004) propose a framework that is based on the process of in-
vestigating physical crime scenes. The framework aids the investigator to develop and test hy-
potheses, as well as discover insights about events that have occurred in the crime scene. This
framework includes five phases: readiness, deployment, physical crime scene investigation,
digital crime scene investigation, and presentation. Among them, the first phase is of interest
since it includes the necessary steps for quicker incident handling, i.e. investigation unit
training, collection and training on the necessary equipment. In the same perspective, referen-
ce to forensic readiness can be found in corporate systems that integrate mechanisms for secur-
ity incident handling (Tan, 2001; Weise and Powell, 2005).

The investigation framework that is proposed by Ciardhuáin (2004) is independent from
the organizational/technological context and aims to have an interdisciplinary nature, i.e. to be
applicable not only by technically savvy users. In this context, the framework can be applied
by first responders, investigators, IT executives, and auditors. The framework describes thirte-
en steps, namely: awareness, authorization, planning, notification, search and identification of
evidence, collection, transport, storage, examination, hypothesis, presentation, proof/defense,
and dissemination.

Beebe and Clark (2005) propose an investigation framework, which - in contrary to the a-
bove mentioned frameworks - is multi-tiered. The framework includes six high level phases:
preparation, incident response, data collection, data analysis, findings presentation, and inci-
dent closure. The authors propose evidence triage in which collected data are prioritized ac-
cording to their importance, hence reducing the volume of data that is send to the lab for exa-
mination.

In the same context, Rogers et al. (2006) elaborate on forensic triage in time sensitive in-
vestigations. Their investigation framework is tailored for on-site forensics, i.e. the investiga-
tion is conducted on the crime scene, without sending any device to a lab for an exhaustive a-
alysis. It includes the following six phases: planning, triage, usage/user profiles, chronology-
timeline, Internet activity, and case specific evidence.
Contrarily to the frameworks in (Ciardhuáin, 2004; Beebe and Clark, 2005), which are rather detailed, the framework in (Kohn et al., 2006) is more abstract. As a result, it is more adaptive to technology changes. The framework merges the investigation phases of frameworks that have preceded it in only three phases, namely: preparation, investigation, and presentation. The authors highlight that the investigator must document each step taken, as well as are aware of the legal requirements that are in effect in each investigation.

Ieong (2006) aims to provide a framework that is understood from all the participants in a digital investigation, irrespective of their technical skills (i.e. from information technologists to legal practitioners). The framework includes forensic procedures that are based on the Zachman’s framework (Zachman, 1987), as well as elaborates on the principles of reconnaissance, reliability, and relevancy. It defines different roles for each forensic investigation participant and compiles a key question list for them, namely: what (i.e. data), why (i.e. motivation), how (i.e. function), who, where, and when.

Abstract investigation frameworks often include in their initial phases guides for not trained personnel. The guides are followed when a field expert is not available. Two noteworthy first responder guides are: a) the UK Association of Chief Police Officers (ACPO) (Wilkinson and Haagman, 2007) and b) the US National Institute of Justice (NIJ) (Mukasey et al., 2008). ACPO describes basic principles of a digital investigation procedure, as well as refers to cases in which different processes may be required, such as pedophile incidents, or collection and preservation of evidence from Personal Digital Assistants (PDAs). On the other hand, the guidelines from NIJ include five steps: (a) gathering investigation tools and equipment, (b) securing and evaluating the scene by checking the device’s state and/or by executing preliminary interviews, (c) documenting the scene, (d) evidence collection, and (e) packaging, transportation and storage of potential digital evidence. Both guides include detailed instructions describing the desired behavior of first responders, e.g. what steps must be followed when the device is switched on with an enabled mobile carrier connection.

2.2 DFIF for feature phones

The framework in (Jansen and Ayers, 2007) focuses on the digital investigation of cellular phones. It describes actions taking place before, during, and after the investigation. Before the start of the investigation, a number of planning actions occur, i.e. the preparation and training on forensic tools, as well as determination of responsibilities of each participant in the investigation. Then, the investigation continues with the following phases: preservation, acquisition, examination, and analysis. Finally, the investigation findings are reported, which concludes the investigation.

Jansen and Ayers argue that their framework is also applicable in smartphones. However, as their work is now outdated, it cannot reflect the changes that occurred due to the rapid technological and commercial evolution, or address the challenges posed by the heterogeneity of smartphone platforms. For instance, its evidence source taxonomy is a subset of the current smartphone’s evidence source taxonomy (Mylonas et al., 2012b).

2.3 DFIF for smartphones

More recent frameworks focus more on the technical details of smartphones and their heterogeneous operating systems. In this context, the framework described in (Ramabhadran, 2009) is tailored for Windows Mobile. It includes twelve investigation phases, namely: preparation, securing the scene, survey and recognition, documenting the scene, communication shielding, volatile evidence collection, non-volatile evidence collection, preservation, examination, analysis, presentation and review. The authors refer to Windows Mobile compatible third-party software and vendor specific (proprietary) software and connectors (cables) that can be used for the evidence collection and examination phases.

Yu et al. (2009) propose a framework for Symbian, which is based on (Carrier and Spafford, 2003; Ramabhadran, 2009). The framework describes the following phases: a) preparati-
on and version identification, b) remote evidence acquisition, i.e. command-response protocols such as AT commands that perform logical data copy, c) internal evidence acquisition, d) analysis, and e) presentation and review. The authors also focus on some details in the Symbian security model that hinder digital investigation.

Dancer et al. (2013) propose a framework that includes five phases, namely: transportation, classification, analysis, interpretation, and retention. Their framework, which is based on an abstract model of smartphone devices, aims to be cross-platform, i.e. applicable to all smartphones regardless of their operating system.

2.4 Overview

The investigation frameworks that exist in the forensic literature often name similar or even identical procedures differently. Moreover, in some cases a phase in one framework may be included as an activity of another framework’s phase. Hence, since a digital investigation framework that fits all purposes can hardly be found in the literature, this complexity regarding the framework’s terminology may create confusion to a new digital investigator and increase the cost of her training. The above frameworks at least include activities concerning:

- (a) the investigation’s preparation (e.g. training on the available tools, reviewing of the case, etc.),
- (b) collection and preservation (e.g. in a faraday box) of the smartphone,
- (c) data acquisition,
- (d) data analysis, and
- (e) presentation of the investigation results, either to a court of law or in corporate audience.

Table 1 summarizes the availability or absence of the above mentioned activities in the investigation frameworks of this section, as well as their categorization into abstract DFIF, DFIF for feature phones and DFIF for smartphones.

<table>
<thead>
<tr>
<th>Framework</th>
<th>Type</th>
<th>Prp</th>
<th>Col</th>
<th>Prv</th>
<th>Acq</th>
<th>Als</th>
<th>Prs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Casey, 2000</td>
<td>A</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Reith et al., 2002</td>
<td>A</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Carrier et al., 2004</td>
<td>A</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ciardhuáín, 2004</td>
<td>A</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Beebe and Clark, 2005</td>
<td>A</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Kohn et al., 2005</td>
<td>A</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Rogers et al., 2006</td>
<td>A</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ieong, 2006</td>
<td>A</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>✓</td>
</tr>
<tr>
<td>Wilkinson and Haagman, 2007</td>
<td>A</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Mukasey et al., 2008</td>
<td>A</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>Jansen and Ayers, 2007</td>
<td>F</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Ramabhadran, 2009</td>
<td>S</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Yu et al., 2009</td>
<td>S</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Dancer et al., 2013</td>
<td>S</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

† **Prp** (Preparation), **Col** (Collection), **Prv** (Preservation), **Acq** (Acquisition), **Als** (Analysis), **Prs** (Presentation), **Type**{A (Abstract), F (Feature Phone), S (Smarphone)}.  

5
The majority of the investigation frameworks are tailored for ‘post-mortem’ forensics, i.e. the evidence is collected after a crime has taken place or an incident has occurred. Also, the analysis takes place after the collected devices have reached a forensic lab and/or a forensically sound copy has been created, which may take hours or even days after the incident/crime. This makes these investigation frameworks rather cumbersome, comparing to forensic triage techniques (Rogers et al., 2006; Walls et al., 2011). Also, this rationale of post-mortem analysis has influenced the available data acquisition methods, omitting the collection of time-sensitive and/or volatile data, such as sensor data. Nonetheless, as discussed in the next section, the ad-hoc collection of sensor data from smartphones provides context awareness during a digital investigation, which may aid the prevention of an incident/crime in proactive forensics (Mylonas et al., 2012b).

3 Sensor data in digital forensics

The majority of smartphones, as well as feature phones, include multimedia hardware such as a microphone and camera (often a pair of front and back cameras). Moreover, other hardware is getting more ubiquitous in smartphones, such as: (a) location sensor (i.e. Global Positioning System (GPS)), (b) motion sensors (i.e. accelerometer, gyroscope), and (c) environmental sensors (e.g. magnetometer, proximity, light, temperature, pressure). Spyware authors have already used smartphone sensors in order to acquire context awareness either for research reasons (e.g. in Mylonas, 2008; Niemelä, 2008; Xu et al., 2009; Schlegel et al., 2011), or for commercial ones, for instance spyware software used for family affairs (e.g. FlexiSPY2). In addition, sensors provide a data source that can be collected in relation to crime investigations in order to provide context awareness. This context awareness can be used, in accordance with the existing legal framework, in a proactive forensic investigation to construct a suspect’s activities, which aid the creation of a hypothesis, or rejection or acceptance of an alibi (Mylonas et al., 2012b).

Nonetheless, the majority of sensor data cannot be collected during a post-mortem investigation. This stems from their inherent characteristics; sensor data are volatile and are time-sensitive. Thus, time stamped evidence derived from sensor data can hardly be found on the device after a crime’s occurrence, unless they have been explicitly collected. Currently, only GPS data are collectable in post-mortem forensics. They are treated as metadata, which are either embedded in other files (e.g. geotagging), or cached by the operating system (Allan and Warden, 2011) or by other applications (e.g. Google maps, social media apps).

One may argue that investigators can infer context awareness by collecting data from a service/network provider, without accessing the suspect’s device. For instance, device location can be estimated from collected data from the carrier’s network via triangulation and device speed can be estimated from the distance between the cell ids that the device is visiting. However, on the one hand the accuracy of this estimation cannot be compared with the data that are generated from the device’s dedicated hardware, which, in general, gives more accurate measurements. On the other hand, sensors provide unique measurements from: (a) a single hardware and (b) combinations of measurements deriving from two or more sensor hardware - these combinations are commonly referred to as virtual/software sensors. For instance, for the former, the light sensor provides data that cannot be found from another source. For the latter, to the best of our knowledge, the only available method to collect the device’s orientation is from dedicated hardware, e.g. the accelerometer and the magnetometer.

In a digital investigation sensor hardware can provide direct evidence, such as location evidence from a GPS. They also provide indirect evidence, i.e. evidence that is inferred from the acquired data values. For instance, an investigator may use the light sensor to infer whether a suspect is indoors or outdoors, or to infer keystrokes from a nearby keyboard via the smartphone accelerometer (Marquardt et al., 2011). Some sensor data are easily comprehen-

sible (i.e. GPS coordinates, values from environmental and multimedia sensors), whereas the values from motion sensors are not. For the latter, existing sensing literature has elaborated on the classification of data from motion sensors with the aim to perform activity recognition (e.g. in (Choudhury et al., 2008; Liu et al., 2009; Lane et al., 2010; Kwapisz et al., 2011; Bujari et al., 2012)). While this discussion about sensor data analysis is not exhaustive, we consider that a forensic investigator can benefit from the existing literature if there is a need to infer gestures (Kern et al., 2003; Liu et al., 2009; Choe et al., 2010) as well as other activities, such as walking, running, etc. (Bao and Intille, 2004; Ravi et al., 2005; Lester et al., 2006; Long et al., 2009; Khan et al., 2010; Kwapisz et al., 2011; Bujari et al., 2012).

Finally, the combination of sensor data provides stronger indications about the smartphone’s context. For instance, combined data from the GPS, accelerometer and magnetometer provide a smartphone’s navigation (e.g. device orientation, speed, etc.). In addition, such data combinations often improve the collection effectiveness, e.g. the light sensor may be used as a condition of whether a camera caption will take place or not.

4 Themis

Sensor data are volatile; therefore, the only way to acquire them, in case such a need arises during a digital investigation, is by software that collects them when they are generated by the suspect’s actions. In this section we discuss the implementation details of Themis\(^3\), a system for remote and ad-hoc (sensor) data acquisition from smartphones.

4.1 Themis’ architecture

Themis implements key components (i.e. the collection agent and transfer protocol) from a proactive smartphone investigation scheme that is described in (Mylonas et al., 2012b). In this scheme an investigator has ad-hoc access to smartphone data, to combat crime that in effect legal framework considers as “serious crimes”, in the same manner as lawful interception is used today for combating them. We regard that the deployment of Themis in serious, as well as time-sensitive crimes (i.e. the investigation is sensitive to time constrains (Rogers et al., 2006)) increases the effectiveness of law enforcement by aiding the investigators to assess the context more accurately and to decide the necessary actions. In such cases (e.g. pedophilia, etc.), we consider that crime prevention has a much more positive impact on the protection of a citizen than a thorough analysis of a collected device after the crime has taken place.

When such a technology is used, specific legal and institutional guarantees and tools must be provided, so as to ensure its lawful use and prevent its abuse. In this context, we proposed in (Mylonas et al., 2012b) that the scheme’s cornerstone would be a specialized Authority, which is: (a) trusted to comply with the legal framework, and (b) independent with regard to the police investigators (see Fig. 1). This trusted Authority, referred hereinafter as a Law Enforcement Entity (LEE), hinders investigators and/or malicious individuals from misusing its data collection mechanism. Furthermore, LEE collects, controls, and stores the acquired data in accordance with the existing laws and regulations.

In Themis, data collection is controlled by a C&C server (hereinafter ‘workstation’), which is managed in the LEE premises. The workstation stores and protects the acquired data. Data are collected by the ‘Themis’ agent (hereinafter referred to as ‘agent’) that resides in the suspect’s smartphone. The data are transferred through a wireless secure channel that is set up between the agent and the workstation. Investigators have only read access to the acquired data via investigation sessions managed by the LEE (Mylonas et al., 2012b).

---

\(^3\) According to the Greek mythology, Themis was the Goddess of Justice, i.e. the personification of divine or natural law, order, and justice.
4.2 Implementation considerations

Themis uses a wireless channel for transferring collected data from the device to the workstation. To ensure the forensic soundness of transmitted data via a wireless channel, a custom protocol is required that includes mechanisms to provide:

(a) **Confidentiality.** The transmitted data must be encrypted so as to avoid their unauthorized disclosure (i.e. eavesdropping).

(b) **Integrity.** The transmitted data must be protected against intentional or unintentional unauthorized alteration/modification.

(c) **Data freshness.** The protocol must include appropriate mechanisms to avoid message replay attacks.

(d) **Data origin authentication.** The receiver must be able to verify that the collected data originate from the expected sender (i.e. protection from data fabrication).

(e) **Interruption avoidance.** The receiver must not be hindered from receiving data from the agent.

Moreover, the protocol must be cross-platform, without restricting the use of any technology, either in the workstation or in the agent’s implementation. The above mentioned properties (a)-(d) of a secure wireless channel ensure the authenticity of the collected evidence (Karyda and Mitrou, 2007). On the other hand, the agent - as any data acquisition tool - must adhere to the four ACPO principles (Wilkinson and Haagman, 2007), which are summarized in (Jansen and Ayers, 2007) as follows:

1. No actions performed by investigators should change data contained on digital devices or storage media that may subsequently be relied upon in court.
2. Individuals accessing original data must be competent to do so and have the ability to explain their actions.
3. An audit trail or other record of applied processes, suitable for replication of the results by an independent third-party, must be created and preserved, to accurately document each investigative step.
4. The person in-charge of the investigation has the overall responsibility to ensure that the above-mentioned procedures are followed in compliance with laws in force.

The first principle refers to ensuring the integrity of the device and its stored data, as well as the integrity of extracted data. It is the basis of traditional digital forensics, but in current forensics practices it progressively fades (e.g. ‘live’ forensics, forensic triage). This is because in many occasions it is infeasible to extract the storage component of the digital device (e.g. smartphone), without losing data or causing damage to it. This principle is also infeasible for sensor data, as they are volatile. Hence, the only way to acquire them is by using software on the device. Nonetheless, the agent’s installation must cause minimal alterations to the device, as well as to the sensor data sources. For the former, in the majority of smartphone platforms applications are sandboxed. For the latter, access to sensor data is performed via the plat-
forms’ API without impairing its sources’ integrity. Finally, regarding the integrity of extracted data the use of a secure channel in Themis, meeting the above requirements, ensures the integrity of the extracted data. The last three principles ensure the accountability of the digital investigation. In Themis, accountability is ensured by the LEE, which manages and controls access to smartphone data according to the scheme proposed in (Mylonas et al., 2012b). Investigators can query smartphone data only via the LEE’s infrastructure and according to the guarantees in force in the country, if and only if the investigation request can justify the need for an investigation session. The LEE stores the acquired data as well as investigation session logs for a period that is compliant with the existing laws and regulations. This documentation of investigation sessions from a trusted LEE enables a forensic analyst to prove that acquired data have not been devised or tampered with. Moreover, as sensor data can be deemed biometric (Mylonas et al., 2012b), the LEE must enable protection mechanisms to satisfy legal requirements or standards.

Finally, this paper focuses on sensor data acquisition and not on trust and functional issues regarding the LEE, albeit, we note that in a real deployment of Themis a pair of LEE may be used, in order to satisfy the principle of separation of duties and to ensure system availability.

4.3 Implementation platform

The Themis’ agent was implemented in the Android platform. Android was selected for the following reasons:

- It is currently the smartphone platform with the largest user space (64%, in the 2nd quarter of 2012), as well as the highest increasing rate in its user base (20.7% increase from the 2nd quarter of 2011) (Gupta et al., 2012).
- It is considerably portable, i.e. compatible with the hardware of several smartphones (e.g. Samsung, LG, etc.).
- Android is open source, hence the details of its security model are readily accessible and have been studied in the smartphone literature.
- It is extensible, i.e. a custom ROM can be created either from device manufacturers (e.g. for unique driver support) or by researchers (e.g. to add enhanced security or functionality to its core components).
- It allows the installation of applications from sources other than the official app repository (i.e. Google Play). Thus, there is no need to root or ‘jailbreak’ the device in order to install the agent to a device.
- Applications are sandboxed, i.e. their installation does not alter other applications or their files. Also, they can access only the resources that the sandbox’s permissions allow.
- Development and testing in Android is considerably aided by the existence of programming libraries and device emulators that are freely and readily accessible.

The agent’s implementation is based on Java, customized for the Android platform. Thus, our implementation may be used as a basis for the creation of agents in smartphone platforms with similar development languages - conforming to the details of the security models of each platform. In (Mylonas et al., 2011, 2012a) the user study revealed that the effort to develop a subset of the agent’s functionality (GPS collection and transfer) was comparable in smartphone platforms that had similar a) security model rationale and b) syntax in the supported programming language, i.e. Android, BlackBerry, Windows Mobile/Phone. Moreover, our agent is compatible with other Android devices, such as tablets.

4.4 Platform’s security considerations

This subsection summarizes our efforts to circumvent the protection mechanisms that are currently employed by Android security model to collect sensor data from the device.
4.4.1 Circumventing permissions

Android’s security model is permission-driven, i.e., apps are sandboxed and can only access a resource (e.g., data, app components) if they are granted the permission that is protecting it (Google, 2012). Specifically, Android users are delegated to grant access to resources that are protected with the dangerous protection level. Before an app’s installation the user is presented with a graphical interface, which lists its requested permissions. If the user is convinced to accept them and install the app, she cannot later withdraw any granted permissions at runtime. Nor will the app prompt the user again for access to these resources. The user may only uninstall the app from the device to stop it from accessing these resources. Moreover, Android uses an all-or-nothing approach concerning permission authorization. In this context, the user does not have the choice to selectively grant a subset of the permissions that an app requests. Hence, she must either accept all permission requests or skip the app’s installation.

Nonetheless, by examining Android’s security model and its API for smartphone sensors during Themis’ implementation, we found out that access to the majority of smartphone sensors is unprotected (Table 2). This holds true as they are not associated with any permission - not even one at normal protection level (Google, 2012). This is orthogonal to the documentation of Android’s security model, where access to every resource must be protected with permission (Google, 2012). In this context, this vulnerability allows any Android application to access the majority of the sensors without prompting the user for authorization. Therefore, the only effective method for a user to discover whether an app accesses sensor data is by static or dynamic app analysis.

As depicted in Table 2, we found that only three sensors are protected with permission. Specifically: (a) camera is protected with the permission `android.permission.CAMERA`, (b) microphone is protected with `android.permission.RECORD_AUDIO`, and (c) GPS is protected with the permission `android.permission.ACCESS_FINE_LOCATION`. The above are dangerous permissions, thus the user must authorize them during app installation. Nonetheless, we expect that average smartphone owners are more likely to accept them. This holds true since in practice smartphone users tend to ignore Android permissions during application installation (Felt et al., 2012b; Kelley et al., 2012; Mylonas et al., 2013) and are unable to comprehend Android permissions, as well as the risks that are related with them (Felt et al., 2012b; Kelley et al., 2012). Moreover, users in any platform are trained to click through interruptive and repetitive messages that appear while they are completing a task (Motiee et al., 2010). Android permissions appear in every app installation and interrupt users from installing and using a new app. This partially explains why they are often ignored.

Smartphone users who elaborate on Android permissions are likely to reject permission according to their perception about its impact on their security and privacy, as well as its relevance with the app’s functionality. For instance, a user is expected to grant access to a permission that allows pairing to Bluetooth devices (`android.permission.BLUETOOTH`) more easily than to one that changes her contact list (`android.permission.WRITE_CONTACTS`). In addition, a voice memo application is expected to be granted the request to access the microphone, whereas a game is not.

Table 2 includes a qualitative metric regarding the likelihood that the suspect rejects a permission protecting access to the device sensors and, due to Android’s all-or-nothing approach, to cancel its installation. This metric is based on a user study conducted by Felt et al. (2012a), regarding the perceived impact of smartphone risks deriving from smartphone permissions. Smartphone users regard the camera and microphone sensors intrusive and thus we regard that their rejection likelihood is high. On the contrary, users feel more comfortable to share their location - and this may stem from the popularity of location based services and, as a result, we assess its rejection likelihood as low.

A digital investigator may circumvent the protection offered by dangerous permissions and hence lower the possibility of app installation rejection by Android users who elaborate on permissions. This can be achieved in Android by deploying the agent as a pair of colluding

---

4 We have verified the existence of this vulnerability even in the latest releases of Android (i.e. v. 4.1.2).
apps which share their permissions and their intended functionality matches their permission requests. Android applications may share their permissions in an either intentional (i.e. colluding apps that break the permission system) or unintentional (i.e. confused deputy attack) manner (Dietz et al., 2011; Felt et al., 2011; Schlegel et al., 2011; Grace et al., 2012). In this context, the investigator must convince the suspect or criminal to install either two individual apps, or one app that, in turn, convinces the user to install the second app (e.g. with a pop-up ad).

Finally, the agent receives commands from the workstation either from the Internet or via SMS messages. In the first case, the `android.permission.INTERNET` is required, which has one of the lowest rejection likelihood among the permissions. This holds true as it is the most commonly requested permission for free benign applications (Zhou and Jiang; 2012). On the other hand, receiving SMS messages requires the permission `android.permission.RECEIVE_SMS`, which has considerable rejection likelihood\(^5\).

### Table 2: Sensor data collection details in Android

<table>
<thead>
<tr>
<th>#</th>
<th>Android Sensor</th>
<th>Access</th>
<th>Rejection likelihood</th>
<th>Spoofing</th>
<th>Unstable</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Accelerometer</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>2</td>
<td>Camera</td>
<td>user granted</td>
<td>High</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>GPS</td>
<td>user granted</td>
<td>Low</td>
<td>X</td>
<td>N/A</td>
</tr>
<tr>
<td>4</td>
<td>Gravity</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>5</td>
<td>Gyroscope</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>6</td>
<td>Humidity</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>7</td>
<td>Light</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>8</td>
<td>Linear Acceleration</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>9</td>
<td>Magnetometer</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>10</td>
<td>Microphone</td>
<td>user granted</td>
<td>High</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>11</td>
<td>Orientation</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>12</td>
<td>Pressure</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>13</td>
<td>Proximity</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>14</td>
<td>Rotation Vector</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>15</td>
<td>Temperature</td>
<td>unprotected</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

#### 4.4.2 Circumventing visual notifications

Amongst the protected sensors, the camera and GPS have additional level of protections to avoid their misuse. Both of them have visual notifications to alert a user that data are being collected, i.e. camera preview window and flash light, and GPS notification icon. Moreover, the camera has also a sound notification (i.e. shutter), which in some countries must be enabled even when the device is muted.

In our experiments we were able to circumvent the camera preview, as well as the shutter by abusing the API of the camera. We changed the properties of the preview window (i.e. Android Activity), e.g. set its width and height to zero and set null parameters in the relevant API to disable the shutter. It should be noted that these changes are only in effect when the agent is using the camera and other apps that use the camera are not affected. However, in our tests we found that the device drivers for the camera are implemented differently by each device’s manufacturer. Therefore, we had to abuse the API differently in each of the test devices. This makes the API misuse rather unstable and, hence, it must be tested in advance by the forensic investigator. Otherwise, the agent is more likely to fail. Finally, the flash light can be easily disabled by an API parameter, without misusing its API parameters.

\(^5\) The user study in (Felt, et al., 2012b) did not include the impact of malicious handling of incoming SMS messages.
To the best of our knowledge there is no effective method to hide the GPS notification, which is presented in the screen’s notification area, when: (a) an app is running in the background (i.e. it does not consume any space in the smartphone screen), (b) the device owner is using the device, and (c) and the app requests access to the GPS. Nonetheless, we consider that normal users will not notice this notification icon when the agent is collecting location data, as: (a) users are accustomed to using location based services and thus see the notification icon often, (b) the size of notification icon is rather small and as a result easy to ignore, and (c) the device owner may be simultaneously using another location based application (e.g. for navigation) and thus the presence of the icon may not seem suspicious. In addition, in Android an app can access the last known user’s location, thus the investigator can access semi-fresh location data, without the GPS symbol being shown to the user. The investigator can combine these location data with approximate location data from the mobile network provider and try to infer the suspect’s location.

Finally, by examining Android’s API we found out that Android’s security model does not provide any visual distinction between error messages that are being created by the operating system and the ones created by third-party apps. Therefore, a malicious app may exploit this spoofing messages are often used by malware to mislead smartphone users (Mylonas, 2008).

4.5 Agent’s installation

In specific circumstances regarding the ad-hoc collection of smartphone data - only and solely for law enforcement purposes and in accordance with the existing laws and regulations - the investigation team may use software that pre-exists in the device (Rose, 2012). For instance, the device provider may cooperate with law enforcement authorities by including such software in a custom ROM, this software may be part of incident response and thus be included in the security policy of a critical infrastructure. However, in general we expect that investigators who use Themis will come across the problem of software installation. In this subsection, we discuss three methods for the Themis’ agent installation in an Android smartphone, namely: physical access (i.e. manual installation), social engineering, and vulnerability exploitation.

4.5.1 Physical access

Manual installation of an app in Android is not time-consuming. Hence, if the device is left unattended, then the agent’s installation depends on whether the device is locked or not. Password protection can be circumvented by: a) entering the device in debugging mode via USB (Hoog, 2011), b) guessing the password, especially if a short one is used, and c) inferring a graphical password from smudges on the device’s screen (Aviv et al., 2010).

In a previous user study (Mylonas et al., 2013), we found that 64% of participants did not password protect their device. In this context, due to the portability and small size of smartphones, the likelihood of a temporary unauthorized access cannot be neglected.

Finally, manual installation may be used, apart from the scenarios with a temporary access to the smartphone (e.g. family affairs, corporate espionage, insider threat), in corporate environments for incident response purposes. In this scenario, administrators install the agent in an employee’s smartphone to enforce a security policy and/or to ensure forensic readiness.

4.5.2 Social engineering

During an installation with social engineering, the agent is installed by the device’s owner. The owner is lured into downloading the agent, which is masquerading as an app of her interest. This app can either be a repackaged version of an existing Android app, or a new app.

Prior to app installation, the device owner must be convinced to accept any access request the app makes. As discussed previously, due to Android’s all-or-nothing approach the user does not have the choice to selectively grant a subset of the permissions that an app requests.
Hence, she must either accept all permissions or skip the app’s installation. Nonetheless, smartphone security literature has revealed that users tend to ignore Android permissions during application installation (Felt et al., 2012b; Kelley et al., 2012; Mylonas et al., 2013) and our previous analysis showed that the majority of smartphone sensors are unprotected.

The agent’s installation via social engineering requires prior intelligence for the suspect or criminal. For instance, in the case that the investigation team has leads to a ‘severe’ crime, the investigators may use relevant social engineering context in order to lure him to install the agent. Also, colluding apps may be used in order to circumvent security savvy users who scrutinize permissions. Finally, this installation option is not appropriate when time-sensitive and ad-hoc access is needed without the device owner’s participation in the agent’s installation. In this case, an intrusive installation vector is required, i.e. vulnerability exploitation.

4.5.3 Vulnerability exploitation

This installation method involves the exploitation of a vulnerability, which leads to elevation of privileges. If the exploitation is successful, then the investigator has privileged access (i.e. with root privileges) in the smartphone. Therefore, she is able to install the agent in the device. The vulnerability may be present either in the operating system, or in an application (e.g. Web browser). Furthermore, the rooting process that users follow to unlock their devices often create vulnerabilities that are easily exploitable (Cluley, 2009).

Known vulnerabilities and exploits are often publicly available in databases such as Open Source Vulnerability Database6 (OSDVB) and Exploit Database7. Table 2 summarizes the exploits that are currently being used by Android malware in the wild (Zhou and Jiang, 2012).

<table>
<thead>
<tr>
<th>Exploit</th>
<th>Vulnerable Software</th>
<th>Exploit details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asroot</td>
<td>Kernel</td>
<td><a href="http://milw0rm.com/sploits/android-root-20090816.tar.gz">http://milw0rm.com/sploits/android-root-20090816.tar.gz</a></td>
</tr>
<tr>
<td>Exploid</td>
<td>init (Android &lt;= 2.2)</td>
<td><a href="http://c-skills.blogspot.com/2010/07/android-trickery.html">http://c-skills.blogspot.com/2010/07/android-trickery.html</a></td>
</tr>
<tr>
<td>Rage against the cage</td>
<td>adbd (Android &lt;= 2.2.1) and zygote (Android &lt;= 2.2.1)</td>
<td><a href="http://c-skills.blogspot.com/2010/08/droid2.html">http://c-skills.blogspot.com/2010/08/droid2.html</a></td>
</tr>
<tr>
<td>Zimperlich</td>
<td>adbd (Android &lt;= 2.2.1) and zygote (version &lt;= 2.2.1)</td>
<td><a href="http://c-skills.blogspot.com/2011/02/zimperlich-sources.html">http://c-skills.blogspot.com/2011/02/zimperlich-sources.html</a></td>
</tr>
<tr>
<td>KillingInThe-NameOf</td>
<td>ashmem (Android &lt;=2.2.1)</td>
<td><a href="http://c-skills.blogspot.com/2011/01/adb-trickery-again.html">http://c-skills.blogspot.com/2011/01/adb-trickery-again.html</a></td>
</tr>
<tr>
<td>GingerBreak</td>
<td>vold (Android &lt;=2.3.3)</td>
<td><a href="http://c-skills.blogspot.com/2011/04/yummy-yummy-gingerbreak.html">http://c-skills.blogspot.com/2011/04/yummy-yummy-gingerbreak.html</a></td>
</tr>
<tr>
<td>zergRush</td>
<td>libsysutils (Android &lt;=2.3.6)</td>
<td><a href="http://forum.xda-developers.com/showthread.php?t=1296916">http://forum.xda-developers.com/showthread.php?t=1296916</a></td>
</tr>
</tbody>
</table>

Similarly to penetration testing, the investigation team will have to perform information gathering to select an exploit that matches a known vulnerability of the suspect’s device (McClure et al., 2005). Otherwise, incorrect use of exploits, or use of exploits that are not thoroughly tested for their effectiveness, may reset the device or even brick it (i.e. render it useless).

This installation option is suitable in cases, where time-sensitive, ad-hoc acquisition of sensor data is required. It assumes that the investigation team has the expertise and the resour-

---

6 http://osvdb.org/
7 http://www.exploit-db.com/
ces in order to complete it. Also, it assumes that information gathering has been performed and the investigation team has prepared or tested the appropriate exploits, so as to cause minimal changes to the device.

4.6 Agent’s functionality

The agent’s functionality is depicted in Fig. 2. It includes agent’s activation, agent’s configuration and data acquisition.

After its installation, the agent is activated when one of the following conditions is met:

a) When the operating system finishes its boot procedure. This enables the agent to execute after the device restarts.

b) Each time the user interacts with the device, i.e. unlocks the device. This enables the collection of data that require the presence of device’s owner (e.g. voice, camera).

c) Each time the device reconnects to the Internet, irrespective of the network channel (i.e. mobile carrier’s network or Wi-Fi). This enables the agent to contact the forensic workstation in order to acquire a new acquisition configuration file and/or to transfer to it any pending data, i.e. collected data that have been stored locally.

d) When an SMS command arrives (Mylonas, 2008). This enables the agent to receive an acquisition configuration file when the device cannot connect to the Internet.

![Diagram of agent's functionality](image)

**Fig. 2. Agent’s functionality**

After the first three events the agent attempts to contact the forensic workstation and receive an acquisition configuration file. Then, the agent decrypts and parses the acquisition configuration file and starts the data acquisition. In data acquisition, Android’s sensor API is used and, thus, the collected evidence is reliable (Karyda and Mitrou, 2007) - assuming the integrity of the reported data from the sensors. The collected data are compressed, hashed and encrypted. If the agent is unable to connect to the forensic workstation, then the acquired data are stored in a local database. Otherwise, they are transferred to the forensic workstation.

4.7 Evidence transfer protocol

We implemented a custom protocol for the transfer of sensor data from the agent to the forensic workstation, which we refer to as Evidence Transfer Protocol (ETP). ETP is a Command and Control (C&C) protocol, where the workstation sets the acquisition options in an
acquisition configuration file and the agent gathers data from the smartphone sensors according to these options. Then, the data are transferred to the workstation via a secure channel.

As depicted in Fig 3, ETP includes three messages (Table 4):

(a) **Heartbeat.** The agent registers to the workstation its details with this message, i.e. current IP address and its id (i.e. Universally Unique Identifier - UUID). The first Heartbeat message, i.e. the one after the agent’s installation, includes the availability and details of sensor hardware in the device.

(b) **Configure.** This message includes the acquisition options, i.e. multiple entries that specify selection of sensor data sources (e.g. GPS, accelerometer, etc.) and, optionally, collection options (i.e. capture duration, sample rate, and capture intervals).

(c) **Acquire.** This message includes the collected data to the forensic workstation. It consists of multiple entries that contain raw data and their metadata (e.g. identifier of the sensor source, collection resolution and timestamp).

![Fig. 3. Overview of the evidence transfer protocol](image)

Our implementation aims to be simple (i.e. with low implementation complexity in order to be used and extended for academic purposes), as well as cross-platform. For the previous reasons, the protocol’s implementation is based on Hypertext Transfer Protocol (HTTP) for message exchange. However, the use of HTTP messages is not restrictive, thus any C&C protocol can be used, e.g. Internet Relay Chat (IRC). Moreover, we did not delve into any technologies that conceal data transfer, e.g. covert channels. We expect that such technologies will be used in a real deployment of Themis.

In our protocol implementation, the agent registers to the workstation with a HTTP POST message and then the workstation replies with an acquisition configuration file. Finally, the agent transfers data with a HTTP POST message. The messages’ syntax adheres to JSON\(^8\), a text format that is language independent, but easy to parse in any programming language. Table 4 summarizes the JSON attributes that ETP uses.

The implementation includes commonly used mechanisms that provide a secure channel, i.e.: all messages are encrypted (AES-256) and hashed (SHA-1) in a hash-then-encrypt manner. This provides data confidentiality and integrity, as well as data origin authentication. All messages that are exchanged are time-stamped, which provides data freshness. Moreover, every collected sensor data is time-stamped, which allows investigators to create a timeline of the events that happened.

In a real deployment of Themis we expect that the forensic workstation would incorporate mechanisms that: a) avoid message interruption (i.e. denial of service attacks), such as load balancing, server farms, etc., and b) provide key management and key exchange. These mechanisms are out of the scope of this paper.

**Table 4. ETP message exchange syntax**

<table>
<thead>
<tr>
<th>Message</th>
<th>Protocol descriptors</th>
<th>JSON included attributes(\d)</th>
</tr>
</thead>
</table>

\(^8\) [http://www.json.org/](http://www.json.org/)
**HEARTBEAT**

| Agent’s unique identifier, e.g. UUID. |
| Agent’s current IP address. |

**DATA_SRC_DETAILS**

A generic identifier of the data source. It is either a single identifier, or a vector containing sensors’ details in a name-value pair (Appendix C).

**TIMESTAMP**

Timestamp of the agent’s request.

**HASH**

Hash value for the message’s data.

**CONFIGURE**

A name-value (sensor_type: value) pair that selects a sensor data source, e.g. 8 (Proximity Sensor), 4 (Gyroscope).

**COLLECTION_OPTS**

A vector of name-value pairs (capture_duration: value, sample_rate: value, capture_interval: value) that specifies capture configuration. For instance, capture_duration: 20 sec.

**TIMESTAMP**

Timestamp of the agent’s configuration.

**HASH**

Hash value for the message’s data.

**ACQUIRE**

A generic identifier of the data source. It is either a single sensor identifier, or a vector containing the following sensor’s details in a name-value pair: (a) RESOLUTION: value, (b) NAME: value, and (c) TYPE: value.

**RAW_DATA**

Contains raw data values, either a single value or a vector of values.

**TIMESTAMP**

Timestamp of the data acquisition.

**HASH**

Hash value for the message’s data.

---

1 See Appendix A for the description of the JSON Attributes.

5 Experimental results

In this section we summarize our experience regarding data collection of smartphone sensors by Themis, as well as describe scenarios regarding its use and preparation.

5.1 Test environment

Before experimentation, we set-up a forensics test lab (Fig. 4). The workstation was implemented with a server running Linux (kernel 2.6.32-38), PHP (v. 5.3.8), and Apache server (v. 2.2.22). Initially, we installed and debugged the agent’s implementation in the emulator provided by Google. Then, we installed and tested our implementation in two Android smartphone devices: a) a LG Optimus L3 (LG-E400) with Android v. 2.3.6 and b) a Samsung Nexus S with Android v. 2.3.6.
Table 5 summarizes sensor availability in the two devices and the emulator. The sensors are either hardware-based or virtual ones. The latter mimic hardware sensors and their measurements are derived by combining measurements from one or more hardware sensors. Android does not impose any sensor configuration in the underlying hardware. Thus, it is common that different devices have different sensor configurations. In this context, Samsung Nexus S includes all sensors, except for the three environmental sensors that are not commonly found in smartphones (i.e. humidity, pressure, temperature). LG Optimus L3, apart from the above mentioned three sensors, does not include the front camera, light, and gyroscope sensors. The device emulator only supports the camera, microphone and GPS sensors. Third-party software exists that adds support to the emulator for the rest sensors, e.g. OpenIntents. However, to keep our tests realistic, we decided to carry out the collection of sensor data only from the real devices.

### Table 5: Sensor availability in the testing devices

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Type</th>
<th>Samsung Nexus S</th>
<th>LG Optimus L3</th>
<th>Google Emulator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>hardware</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Camera (Front)</td>
<td>hardware</td>
<td>X (X)</td>
<td>X (-)</td>
<td>emulated</td>
</tr>
<tr>
<td>GPS</td>
<td>hardware</td>
<td>X</td>
<td>X</td>
<td>emulated</td>
</tr>
<tr>
<td>Corrected Gyroscope</td>
<td>virtual</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Gravity</td>
<td>virtual</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>hardware</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Humidity</td>
<td>hardware</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Light</td>
<td>hardware</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Linear Acceleration</td>
<td>virtual</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Magnetometer</td>
<td>hardware</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Microphone</td>
<td>hardware</td>
<td>X</td>
<td>X</td>
<td>emulated</td>
</tr>
<tr>
<td>Orientation</td>
<td>virtual</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Pressure</td>
<td>hardware</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Proximity</td>
<td>hardware</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Rotation Vector</td>
<td>virtual</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Temperature</td>
<td>hardware</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

During our tests, both test devices were able to open network connections either via Wi-Fi or via the carrier’s network (e.g. UMTS (3G), HSDPA). We also tested scenarios where network connectivity was unavailable after parsing the acquisition configuration file. In these scenarios the collected data were stored locally and transmitted when network connectivity was restored.

### 5.2 Bandwidth and battery experiments

Effective data collection from smartphone sensors requires a proper configuration of the acquisition file. Otherwise, the device’s battery or internal memory may be exhausted, which may result in making the device owner suspicious. Therefore, we expect that before the deployment of Themis its owners would perform initial stress tests, similar to those discussed herein, in order to avoid such occasions. These tests can be part of the configuration and training on the tools, which the forensic team will use. The aforementioned typically take place in the initial phase of a digital investigation, i.e. Preparation phase.

In this context, we performed preliminary stress tests regarding the battery and bandwidth consumption of each sensor in the two test devices. While, these results are clearly not repre-
sentative (i.e. they must be tested in more devices), they show a clear indication of each sensor’s requirements.

Figures 5-9 summarize bandwidth requirements when the two devices were instructed to start data collection from all sensor and the devices were on a table (Appendix D refers to the details of these tests). We collected sensor data for 30sec in all available sampling rates of the sensor in Android\textsuperscript{10} (where applicable), i.e. normal, ui, game, and fastest.

The results regarding bandwidth consumption are comparable, but not similar, due to the different hardware configurations of the two devices. Specifically, our results indicate that bandwidth requirements grow exponentially as the sampling rate increases. This is the case with the bandwidth experiments on LG L3. As expected the multimedia sensors consume the largest bandwidth. Among the rest sensors the gyroscope and corrected gyroscope are the most ‘bandwidth hungry’ sensors. Our, experiments showed that these two sensors produce more events in the given time frame. On the other hand, the light, GPS, and proximity sensors consume the lowest bandwidth.

The sensor’s raw data are represented in Android with float array values. In ETP these values are inserted in JSON strings for their transmission, which introduces a bandwidth overhead (Fig. 10 presents these tests for the accelerometer; the rest sensors are skipped for readability reasons). Moreover, the multimedia sensors are encoded into Base64, which introduces additional overhead (33\%). This overhead is eliminated even with a simple compression algorithm (i.e. GZip), thus the final bandwidth requirements are comparable or less than the raw data requirements (Figures 7-8, 10).

![Figure 5: Bandwidth requirements for Nexus S (excluding multimedia sensors and GPS)](image)

Figure 6: Bandwidth requirements for LG Optimus L3 (excluding multimedia sensors and GPS)

Figure 7: Bandwidth requirements for Nexus S (only multimedia sensors)

Figure 8: Bandwidth requirements for LG Optimus L3 (only multimedia sensors)
Finally, we used Android’s power API to get an approximation of power consumption for a subset of the device’s sensors. These tests were software-based, thus their accuracy is unclear. Moreover, the API did not cover all available sensors (e.g. camera). These results are given in Figs. 11-12 (see Appendix D for details). In both tests we found that the use of the environmental sensors (i.e. light, proximity) and the accelerometer is cheap in terms of power consumption. For the latter, this is an interesting finding, since these sensors are well studied and applied in the literature to infer users activities (e.g. running, standing), or as side channel attacks (i.e. keylogging from a nearby PC keyboard).
5.3 Themis scenarios

In all scenarios, which are presented in the sequel, we assumed that the suspect is a smartphone user. Moreover, we assume that the device has an active Internet connection via the carrier’s network, as well as via Wi-Fi hotspots. Also, we assume that a digital investigation takes places for a crime that the legal and regulatory context considers severe. The scenarios that are presented do not aim to be exhaustive, but only indicative on the use of Themis in a digital investigation.

5.3.1 Social engineering installation

The installation method that is selected to satisfy the requirements and/or restrictions of a forensic investigation affects the risk signals (i.e. requests for permission authorization) that will be presented to the device owner during app installation. Hence, if the forensic team selects manual installation or vulnerability exploitation for the agent’s deployment, then the owner will not see any installation risk signals. This is because the app is installed without the device owner’s ‘participation’ either remotely or via physical access to the device.

On the other hand, when the agent is installed via social engineering the risk signals that will be presented depend on the available sensor data sources that the investigator selects to access. In this context, a different variant of the agent may be deployed, e.g. one without access to the microphone if this data source is not used in the investigation.

Herein we describe a scenario of the agent’s installation via social engineering. We assume that during a digital investigation the suspect is convinced to download and install a third-party Android app, including the agent’s functionality. Android allows the installation of apps that reside outside the official app repository (i.e. Google Play). Therefore, the installation of the app depends on whether the suspect will examine the apps permissions (Fig. 13). We assume that the investigation team has reasons to believe that the suspect is not security and technically savvy. Hence, the suspect is more likely to accept the app’s permission requests, according to the security discussion presented in section 4.4.
In this scenario we present the use of a spoofing error message, i.e. an error message that look exactly similar to system warnings (Fig. 14). Android’s security model does not provide any visual distinction between error messages that are being presented by the operating system and the ones created by third-party apps. We used the spoofed error message of Fig. 14b to mislead the suspect that the app is not installed. We also, ensured that the app does not appear in the device’s menu with a quick launch icon. Themis will only show up in the list of running services with the app name Google Sync. In Android, other background services execute with a similar app icon as well as similar names, e.g. Google Backup Transport, Google Contacts Sync, Google Partner Set up. In this context, the suspect is expected to ignore the message and continue with another task, assuming that the agent is not installed. This holds true as in a previous user study we found out that smartphone users do not adequately consider security issues during app installation (Mylonas, et al. 2012b).

5.3.2 Validation of suspect’s identity

In this scenario investigators must validate that the collected data are generated from actions belonging to the suspect. In general, a smartphone - in contrast to a computer - is a single-user device, therefore it is uncommon that the device is lent or shared. However, in some cases the suspect may try to trick investigators and create false alibi, e.g. by giving the device to another individual or by placing it in a vehicle (e.g. leaving it in a bus).

In such a case, investigators may benefit from Themis by instructing the device to capture the device holder’s face each time the device is accessed. The workstation’s acquisition configuration string is included in Snippet 1.
The above acquisition string instructs the device to get a snapshot from the front camera, every time the user unlocks the device. In our implementation, we also access the proximity sensor every time such a command is sent to the agent. This is to deduce whether the screen’s surface, and hence the front camera, is blocked with another object. For readability reasons we omit the presentation of the collected data in this subsection.

5.3.3 Validation of location data from the GPS

In this scenario we assume that the suspect has installed software in his device that protects his location privacy. This software tampers with data provided via the GPS (as in (Beresford et al., 2011; Hornyack et al., 2011; Zhou et al., 2011)) and returns static mocked values, namely (latitude=38.51618037248121, longitude=23.631649307180822). Moreover, we assume that the network provider is not cooperative. In this context, the only option to validate GPS data, if the investigator has reasons to believe that they have been tampered with, is to switch to an alternative location provider. Specifically, the investigator may use databases which map Wi-Fi MAC addresses or carrier cell identifiers to GPS coordinates (Android refers to these data as network location provider). These databases are used by location providers (e.g. Apple, Google, Sky-hook), so as to improve their location-based services. The investigation team can use these databases to cross validate GPS data, if there are suspicions that they are invalid.

The investigator notices that the suspect’s location remains unchanged for three hours when GPS data are collected from the device (see Fig. 15 for a timeline of events and Appendix D for the acquisition file as well as the collected data). Then, the investigator uses the acquisition configuration entry that is presented in snippet 2. As a result, Themis agent collects location data from the GPS, as well as the network location provider. By examining the collected data (Snippet 2), the investigator can unveil the presence of software which tampers with GPS data. This holds true, since on the one hand the GPS locations remain the same, but on the other location data from the network location provider change.

Finally, in case that in this scenario the mobile network provider is cooperative, the location data acquired from the device via the network location provider can be cross-validated.

---

Snippet 1: Data used for validation of suspect’s identity

Snippet 2: Data collected from the device

---

11 For the creation of the timeline we extended the open source project available in http://www.simile-widgets.org/timeline/
During a digital investigation, a cooperative provider may assist investigators by providing the suspects context. For instance, the carrier may provide approximation of the suspect’s whereabouts. Nonetheless, sensor data provide a superset of context awareness that is available to the carrier, as well as the data measurements are more accurate. For instance, the combination of measurements from the location sensor with environmental sensors (e.g. light, temperature) and multimedia sensors (i.e. microphone, camera), may provide an extended context of a suspect. This context may be crucial to protect future victims (e.g. protection of minors, etc.) in time-sensitive crimes (Rogers et al., 2006).

In this context, Snippet 3 shows a snapshot of our experiments. In this case, combined data from the light sensor, the GPS, and the microphone were collected to get the extended context of the test device. The acquisition configuration string instructs the device to capture data from the light sensor for five seconds every five minutes, as well as data from the GPS and microphone for thirty seconds every five minutes.

```
Collected data: [{"sensor_type":5,"capture_interval":300,"capture_duration":30},"location_provider":"gps","capture_interval":300,"capture_duration":30}]
```

### 5.3.4 Extended context awareness

Figure 15. Legitimate error message (from Android) and spoofed error message (from Themis)

**Snippet 2: Validation of location data**

![Fig. 15. Legitimate error message (from Android) and spoofed error message (from Themis)](image)

**Snippet 3**

**Extended context awareness data**

```
Acquisition configuration entry:
 [{"sensor_type":5,"capture_interval":300,"capture_duration":30},"location_provider":"gps","capture_interval":300,"capture_duration":30}]
```

```bash
430ece1cd4b10dc3b60038362577a0af3a9a87c4eb83a86ce4a0f2a8c83025abeb3383f7df2e0b0d0038362577adee
```
The initial steps of the analysis included plotting the GPS data in a map (Fig. 15). Then, we used the light sensor measurements to infer whether the suspect is indoors or outdoors. To do so we used a classifier (Appendix E). The first value from the light sensor (i.e., 10874.4) suggested that the device was outdoors (right point on the map). The second value from the light sensor (i.e., 10.0) shows that the device was indoors (left point on the map). By using the timestamps of the sensor data we produced a timeline of events (Appendix D), i.e. we inferred that the device’s holder entered a building. This was subsequently confirmed by the holder of the test device. This timeline of events can be useful to investigators in order to evaluate a suspect’s alibi.

![Fig. 16. Suspect’s context](image)

The analysis ended by examining audio data, which were collected from the microphone. For readability and space reasons we omit the binary data of the audio. In the first audio data, loud sounds from car traffic were collected, giving additional evidence that the device was outdoors. In the second, we collected a snapshot of a conversation between two persons (a male and a female). Data from the microphone may reveal the social context of the users, thus investigators may benefit from such data in cases of restrictive measures (e.g. abuse of minor, etc.).

6 Ethical and legal considerations

Ubiquitous computing and communicating have a Janus Face, i.e., if a smartphone combines the functionalities and characteristics of both a cell phone and a computer it enables, on the other side, ubiquitous surveillance. A mobile device with such capabilities as a smartphone is actually a portable data carrier that may provide (also self-) incriminating data for almost every class and type of crime (Bennett, 2012). By providing tools for accessing files and messages as well as data not presently stored on the device, a smartphone offers law enforcement authorities “a treasure trove of evidence” (Morrissey, 2010). By tracking and searching a smartphone, law enforcement authorities may gain not only hard evidence, but also information and insight into valuable information, concerning the character, the habits and the context of a (punishable) behavior.

Data acquired from smartphones and sensor applications which are intended to be used as evidence, should have all the attributes of “conventional” evidence (Karyda and Mitrou, 2007). Primarily they need to be admissible, i.e. comply with the legal principles and requirements in a judicial criminal procedure), irrefutable complete and authentic, i.e. it must be possible to positively tie evidentiary material to the incident. Last but not least, they must be reliable, i.e. collected in accordance with formal requirements to establish its reliability. It is a matter of fact that, not only forensics evidence is only “as valuable as the integrity of the method that the evidence was obtained” (Bennett, 2012), but the process of every digital forensic investigation is subject to considerable scrutiny of both the integrity of the evidence and the integrity of the investigation process. Its reliability has to be demonstrated in consistence with the regulatory framework applicable in the jurisdiction in question (Kuntze et al., 2012). Infringing legal rules with regard to collection, conservation, communication and presentation of data, would compromise the admissibility of the evidence. In this respect, specific attention
has to be paid to the compliance with the communicational and informational privacy requirements.

The use of forensic methods itself may constitute an intrusion of a person’s fundamental right to privacy (Commission of the European Communities, 2006). Tracking a smartphone user and using sensor applications to gain information may implicate in both the communications secrecy and the privacy of a person.

(Lawful) Interception refers to the surveillance of a communication between communication partners in line with the law. However, data gathered through smartphone sensors may be deemed as external communications data (the so-called traffic data). According to the European Court of Human Rights (Case Malone v. UK, Case Copland v. UK), such data fall under the protection of correspondence (and communication) as laid down in Art. 8 of the European Convention for the Protection of Human Rights and Fundamental Freedoms. In parallel, such data may also be regarded as personal (biometric) data, enabling the collection of sensitive information and even the profiling of the person concerned. Furthermore, installing such data collecting applications may affect the personality rights of the person under surveillance, as it may provide a revealing picture of the very core area of her private life. Both national constitutions and supranational texts (as the European Convention for Human Rights) allow restrictions on privacy and the freedom of communication in order to pursue legitimate public interests, such as national security, public safety, prevention and detection of crime. However, material and procedural rules have to ensure that collection and further processing of electronic evidence complies with the provisions guaranteeing privacy protection and communications’ secrecy.

The relevant legal framework varies - sometimes considerably - across the different jurisdictions. The differences between the European approaches, in comparison to the common law countries, concern not only the substantive law requirements but also the constitutional background, the legal context, as well as the legislative technique of the relevant provisions. In US the critical constitutional framework for informational and communicational privacy consists of the Fourth Amendment and its interpretation by the courts, mainly the U.S. Supreme Court. The Fourth Amendment affirms the right of the people to be secure in their persons, homes, papers and effects, against unreasonable searches and seizure (Kerr, 2005; Salgado 2005). In general, the use of forensic methods in the course of a criminal investigation is usually subject to relatively strict procedural controls and guarantees, such as a judicial warrant (Karyda and Mitrou; 2007).

Does it mean that it is necessary to have proper legal authority in order to perform a forensic investigation of cellular telephones and mobile handheld devices? Is the acquisition of data and evidence through smartphone sensors’ data comparable to a search of body or a search of premises or is it analogous with GPS tracking? The use of sensors may amount to a search, as it is capable to explore details about a person that would have previously been unknowable without psychical intrusion. Wiebke (2009) argues that if the data provided by such a device are remotely searched or are not presently stored within the confines of the device, it appears that the acquisition of evidence is comparable to a search of premises. It is worthy to mention that the German Federal Constitutional Court has placed strict limits upon the ability of law enforcement authorities to remotely access computers, PDAs and mobile phones. The Court has specified the rights to “informational self-determination” and “absolute protection of the core area of the private conduct of life” by lifting “security and integrity of information systems” as a fundamental right of the user (Togias, 2010; Wiebe, 2008). According to the theory this new right encompasses systems as PCs or smartphones as well as virtual hard drives or network-based application programs (Bäcker, 2009).

In the context of electronic evidence it is significant to distinguish between systems that are conceived and designed to serve a surveillance purpose and systems that are built and used with another purpose (e.g. traffic monitoring, personal use, etc.) (Coudert, et al., 2011). Especially in the latter case, the conditions of access of law enforcement authorities to the relevant information are of crucial importance. The German Federal Constitutional Court in a case concerning the use of automatic license plate recognition held that the retention of any
digital data that was not predestinated for a specific use was too indiscriminate as to violate the right to privacy and – without guarantees and limitations - may amount to “complete surveillance” (Rushin, 2011). In USA the Supreme Court abstained from its previous jurisprudence about the “suspect’s diminished expectation to privacy in an automobile on a public thoroughfare” (United States v. Knotts) and upheld in the case United States v. Jones (2012) the D.C. Circuit finding that an attachment of a GPS device to the defendant’s vehicle to monitor the vehicle’s movements, constituted a search under the Fourth Amendment and should be subject to a warrant (Reid, 2012). The D.C. Circuit pointed out the prolonged, ongoing monitoring of the suspect was much more intrusive, as it created an extensive log of a person’s behaviour. The same considerations are applicable to the case of monitoring through smartphone’s sensors.

The use and importance of digital evidence gathering is increasing (Rushin, 2011). This holds true due to the increasing smartphones use for every kind of communications (e.g. social networking, etc.) on the one side, as well as the technological feasibility and the decreased costs of easy and extensive data acquisition and retention on the other side. The ad-hoc data acquisition from smartphone sensors is proposed as a valuable method for combating serious crime and severe security threats both through detection and prevention.

Such methods in combination with cheap data storage capabilities provide incentives for organizations and law enforcement authorities to collect and retain data in anticipation of potential risks and future investigations. However, proactive data storage, where evidence gathering is dissociated of wrongdoing of specified suspects, has considerable and far-reaching impacts on fundamental rights. In this context, we should also consider the forensic readiness (Kuntze et al., 2012) of a device: the incorporation of design requirements aiming at ensuring availability and admissibility of data produced by the device has to comply with data protection and communication secrecy requirements.

Technological advances as well as the accessibility and the wide availability of data augments the expectations of law enforcement authorities, while lowering the privacy expectations of citizens, who sometimes may regard the trade-off between security (or even) convenience and loss of privacy worthwhile. The Carrier IQ software, which has been installed on more than 150 million smartphones and has the capacity to log web usage, to chronicle where and when, to which numbers calls and text messages were sent and received (Rose, 2012), indicates the pervasiveness of surveillance. In any case, users should be aware of the data that may be “produced” through the applications and sensors they use. Laws that authorize the State to interfere in informational and communicational privacy must meet the standards of accessibility and foreseeability inherent in the rule of law. Thus, citizens can be aware of all circumstances in which State authorities may conduct surveillance, so that they can regulate their behavior accordingly to avoid unwarranted intrusions (European Court of Human Rights Cases Malone v. U.K., Kruslin v. France).

Concluding, data acquisition through smartphone sensors should be used only in restricted cases and in relation to the investigation. They may be acquired in extreme cases of prevention of illegal behavior and acts that the law defines as “serious crimes” or “national security threats” (e.g. organized crime).

7 Discussion and conclusions

In this paper we proposed Themis, a generic tool for lawful remote ad-hoc acquisition of sensor data. During a data collection session with Themis sensor data are collected as soon as they are generated from an agent that is present on the suspect’s device. Then, the data are transferred via a secure channel, with mechanisms that ensure their forensic soundness. Themis uses mechanisms, which are often found in malware, e.g. in spyware, to collect suspect’s data. Due to the intrusiveness of the tool, it should only be used in extreme cases for detection and prevention of illegal behaviour and acts that the law defines as “serious crimes” or “national security threats”.
One may argue that the ad-hoc acquisition of smartphone data can be skipped since a subset of its data (Mylonas et al., 2012b) are available from the network provider (e.g. approximate location). Also, depending on the legal context and framework, it might be more convenient to acquire them from the network provider rather than the device. However, ad-hoc data acquisition from smartphone sensors is deemed as a valuable method for combating severe crime and security threats, both through detection and prevention. This holds true, since: a) the data acquisition from a smartphone may be used to validate evidence that have been collected from an provider that is not trusted, b) the smartphone may be the only available data source, since the provider may not be co-operative, c) data which are available from the provider often lack the accuracy of sensor data (e.g. location data that are collected from a provider are estimates of the real location and depend on the current cell that the device connects to), and d) some data are not available to the smartphone provider (e.g. light data).

Furthermore, we expect that the use of Themis would increase the effectiveness of law enforcement for crimes in which such intrusive data collection is allowed, especially for the time-sensitive ones (Rogers et al., 2006). This holds true as it can aid investigators to deduce the suspect’s context and to act appropriately (preventively). In such cases (e.g. pedophilia, etc.), crime incident prevention has a far more positive impact on the society, compared to a thorough analysis of collected data from a device that is performed after a crime has taken place.

Even though, Themis was tailored for proactive forensics, the tool’s functionality does preclude its use for post-mortem forensics. In fact, it can be extended to perform logical acquisition for the rest smartphone data sources (e.g. SMS messages, contact list, etc.) during the collection phase (either on crime scene or in the forensic lab) of digital investigation frameworks for smartphones. In this context, Themis can be useful when connectors (i.e. data/power cables) are not available - either for budget or compatibility reasons - for a device that was found in a crime scene.

Themis can be also used as an ‘anti-mugging’ solution, in the context of participatory sensing (Lane et al, 2010). In this case, an individual may install the agent and enable it with a static phrase, e.g. “help”. This phrase can be implemented to enable all smartphone sensors and upload them to servers operated by law enforcement. This software, on the one hand, may deter muggers from attacking individuals and, on the other hand, enable law enforcement to assess a case and thus more effectively manage police force. Finally, in corporate environment Themis can be used as a tool providing forensic readiness, or as part of an incident response system. In such a case, the use of Themis must comply with data protection and communication secrecy requirements. In both cases, different trust and operational requirements may arise, but fall outside the scope of this paper.

When such intrusive technology is used for the prevention of illegal behavior and acts that the law defines as “serious crimes” or “national security threats”, legal and institutional guarantees and tools must exist so as to prevent its abuse. In this context, we proposed in (Mylonas et al., 2012b) that the intrusive technology that we implemented must be managed by a specialized Authority, which is trusted to comply with the legal framework and independent with regard to the police investigators. This trusted Authority, herein referred to as Law Enforcement Entity (LEE), hinders the abuse of the data collection mechanism by either law enforcement (i.e. intelligence gathering for random citizens irrespective of wrongdoing, etc.), or malicious individuals (e.g. espionage, cyber stalking, etc.). In this context, the LEE is a single point of failure in the proposed architecture. This can easily be managed if two different IA - equally trusted - are used. This modified architecture also complies with the separation of duties principle and achieves an increased level of trust. Furthermore, the operation of LEE is costly, since collected data must be stored ensuring their forensic soundness and compliance with the existing laws and regulations. We consider that storage requirements, with respect to the volume of the collected data, may not be prohibitive. This is since Themis is not proposed for surveillance of all citizens, but for a few malicious individuals who are involved in extreme cases illegal of behavior and acts that the law defines as “serious crimes” or “national security threats”.
The agent’s implementation is tailored for Android and its security model. Specifically, our implementation is compatible with all Android devices with version up Gingerbread (i.e. ver. 2.3.*). This includes the 71% of the current active Android smartphones\(^\text{12}\). Moreover, some Android devices cannot update to a newer Android version either due to hardware incompatibility or due to the fact that the device manufacturer is not willing to provide the update. In this context, we expect that the implementation and security details that have been presented in this paper will remain valid for a considerable population of Android devices. However, these details might not be valid for other platforms, or in the newer versions of Android. In this case, the agent must be specifically tailored for their security model.

Our implementation uncovers vulnerabilities in Android which were previously unexplored. Among them the fact that most smartphone’s sensors are unprotected, i.e. breaking Android’s sandbox rationale, is deemed as a rather serious vulnerability. This holds true, as patching this vulnerability would create considerable inconvenience in the platform’s ecosystem, since all the apps that access the previously unprotected sensors must be updated. This suggests that developers must redevelop their apps and resubmit them in the app repositories (e.g. Google Play), which in turn must manage again each submission and often analyze it for malware. Also, users must update their apps otherwise they will fail each time they try to access sensor data. This however, apart from the cost in bandwidth and time, is often not a trivial task for novice users.

The effectiveness of data collection with Themis depends on the investigation team’s available budget and expertise. The tool must be properly configured, otherwise the suspect’s device resources may be consumed quickly (e.g. battery, device memory, etc.) and the suspect may become suspicious. In addition, the investigation team may need to collect the appropriate ‘intelligence’ regarding the suspect’s device. This intelligence will enable the correct selection of exploits that Themis uses in order to be installed or to function. The collection of this intelligence is an important part of penetration testing, but is out of this paper’s scope.

Moreover, the effectiveness of data collection depends on the security level provided by endpoint security that is present. Currently, endpoint security in smartphones has to operate under the same sandbox rules that apply to all apps. This fact restricts their capabilities and for an enhanced security level it requires the rooting of the smartphone, which albeit voids their warranty. In a previous work, we found out that smartphone users adopt endpoint security in a really poor manner (Mylonas et al., 2013). Even though in this work we did not elaborate on the evasion of smartphone endpoint security, Themis’ agent could use mechanisms typically used by malware to evade security controls, such as covert channels or dynamic command execution at runtime (Zhou and Jiang, 2012).

For smartphone data collection we assume that an active Internet connection is present. In practice, such a connection may not be present (i.e. not purchased from the suspect) or, if it is present, its bandwidth quota is not infinite. In this context, the consumption of the suspect’s available bandwidth quota may de-cloak the presence of a digital investigation. This can be avoided through assuming a cooperative network provider by using a ‘silent’ Internet channel, in the same manner that silent SMS messages\(^\text{13}\) are used in some digital investigations (i.e. the suspect is not notified about the communication taking place).

‘Traditional’ digital forensics literature recommends that changing the device state must be avoided, as it is not considered a sound action. This is because it may have a negative impact on the admissibility of the evidence on courts of law (Lessard and Kessler, 2009). However, this trend constantly fades in ‘modern’ digital forensics. This holds true as in some circumstances rooting a device is the only effective way for data acquisition, e.g. live memory dumps are used in order to check the presence of malware, or to circumvent disk encryption. For instance, the investigator may need to acquire the memory contents from a live computer where she is logged in as a simple user (i.e. not administrator), thus, the only option to fulfill her goal is to elevate her privileges.


In our work we change the device’s state by installing an app and in some occasions by exploiting vulnerabilities for the agent’s installation. In case the investigator uses rooting techniques in Themis, we argue that specific requirements must be met: a) this action has to be documented, b) this action must be performed only from a technically capable personnel, c) the exploitation vector must not infect the device with malware or open a backdoor, and d) the exploitation vector must be effective with the specific device (e.g. device model, operating system version, etc.), or else the device may be bricked. On the other hand, we argue that the documented installation of an app on the device does not affect its integrity. This is true, as apps in the majority of smartphone platforms are sandboxed, hence, the installation and execution of an app does not affect the files or execution of others. Moreover, in Themis data collection does not affect other apps and does not impair the integrity of the smartphone’s sensors.

A limitation of our work is that our software assumes the integrity of the reported data from the sensor sources. Albeit, software may exist which provides mock data to any app in the device that requests access to its sensors (Beresford et al., 2011; Hornyack et al., 2011; Zhou et al., 2011). This software is not currently popular to users and it often requires the installation of a custom version of Android. However, in case such software becomes more widespread, Themis can circumvent it by hiding deeper in Android’s kernel and directly accessing the sensor sources, i.e. use a functionality that is commonly used by rootkits in order to conceal their presence. Also, Themis can validate collected data from alternative sources, for instance location data from Wi-Fi MAC addresses (section 5.X)

Another limitation of our work is the validation that the collected data are generated by the suspect herself. Contrary to computers, smartphones are personalized devices that are normally used by only one individual. As a result, in a digital investigation the source of the collected smartphone data can be considered as a fait accompli. However, in some cases an investigator who collects ad-hoc sensor data may need to deduce whether the suspect actually uses her device hardware, or if he is tries to mock the investigation process (e.g. the device is left enabled in a vehicle in order to provide time or alibi to the suspect). This can be achieved with rather intrusive means or by non-intrusive ones. For the former, the investigator may try to infer if the suspect is using the smartphone by using the available technology, e.g. enabling the device’s microphone and/or camera (section 5), using the suspect’s gait as biometric for her identification (Mantyjarvi et al., 2005; Gafurov et al., 2006). For the latter, the investigator may simply attempt to contact the suspect through her device (i.e. place a phone call) and commence social engineering, in the same way attackers use social engineering for information gathering. In any case, the decision on whether such an action takes place or not is dependent on the nature of the crime and the expected expertise of the suspect (e.g. it may be skipped if the investigative team has leads that the suspect is indeed using her device). This decision is normally taken during the preparation phase of the investigation.

Our work lacks the experience and input from law enforcement. This cooperation, on the one hand, could enable the testing of our implementation practicality in more realistic scenarios of digital investigations. On the other hand, law enforcement may provide feedback or training data for the creation of new classifiers for suspect’s activities. For instance, classifiers could be built by getting accelerometer data from a police officer that fires a gun in a test lab. This classifier could be used to deduce when a subject is firing a gun, ensuring that either law enforcement does not misuse their powers during their service, or inferring if a suspect possesses a gun.

In the current implementation of Themis, we do not elaborate on the integrity of the agent. We rather assume that the agent is trusted since it is deployed from a trusted LEE. As a result, we assume that the agent does not insert false evidence, during data collection. In a corporate environment this assumption may not hold. Thus, the integrity of the agent must be tested before its operation (e.g. with remote attestation). However, this falls outside the paper’s scope.

For future work we plan to create taxonomy of sensor data and the available mechanisms from their analysis. We also plan to explore mechanisms that can be added in the agent in order to hinder the construction of false evidence and alibi.
Acknowledgements. This work has been co-funded by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Model (NSRF) - Research Funding Program: Heracleitus II (Investing in knowledge society through the European Social Fund). The authors would also like to express their appreciation to our colleagues, B. Tsoumas and N. Tsalis, for their valuable comments and suggestions.

References


Morrissey S. iOS forensic analysis for iPhone, iPad and iPod touch. New York: Apress; 2010


Mylonas A. Smartphone spying tools. Master’s thesis; Royal Holloway, University of London; 2008.


Tan J. Forensic readiness. Cambridge, USA: @ Stake 2001;
Thing VL, Ng KY, Chang EC. Live memory forensics of mobile phones. Digital Investigation 2010;7, Supplement(0):S74 –82.
CV of authors

Alexios Mylonas (amylonas@aueb.gr) is a Ph.D. candidate and a Researcher with the Information Security and Critical Infrastructure Protection Research Laboratory of the Dept. of Informatics, Athens University of Economics and Business, Greece. He received a B.Sc. in Informatics (Athens University of Economics and Business) and a M.Sc. in Information Security (Royal Holloway, University of London, UK). His current research interests include smartphone security, digital forensics, insider threat, and risk assessment.

Vasilis Meletiadis (meletiadisv@aueb.gr) holds a B.Sc. in Informatics and a M.Sc. in Information Systems, both from the Dept. of Informatics of Athens University of Economics and Business, Greece. His current research interests include ubiquitous computing, smartphone security, and digital forensics.

Lilian Mitrou (l.mitrou@aueb.gr) is an Assistant Professor of Information Law, and a Senior Member of the Information Security and Critical Infrastructure Protection Research Laboratory, with the Dept. of Informatics of Athens University of Economics & Business, Greece. She holds a B.Sc. in Law (University of Athens, Greece) and a Ph.D. in Data Protection (University of Frankfurt, Germany). She has served as Associate Commissioner of the Greek Data Protection Commission. Her professional experience includes senior consulting and researcher positions in a number of private and public institutions, on a national and international level. Her current research interests focus on privacy and data protection, e-Democracy and e-Government, and Internet Law.

Dimitris Gritzalis (dgrit@aueb.gr) is a Professor of ICT Security, the Director of the Information Security and Critical Infrastructure Protection Research Laboratory, and the Director of the M.Sc. Programme on Information Systems, with the Dept. of Informatics of Athens University of Economics and Business, Greece. He holds a B.Sc. (Mathematics, Univ. of Patras), a M.Sc. (Computer Science, City University of New York), and a Ph.D. (Critical Information Systems Security, Univ. of the Aegean). His current research interests focus on VoIP systems security, digital forensics, risk assessment, and critical infrastructure protection. Prof. Gritzalis has served as Associate Commissioner of the Greek Data Protection Commission and as the President of the Greek Computer Society. He is the Academic Editor of the Computers & Security journal (Elsevier).
Appendix A

Table A. 1: Description of JSON attributes that are used in the ETP in the CONFIGURE message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sensor_type</td>
<td>Sensor’s identifier (Android API specific)</td>
</tr>
<tr>
<td>location_provider</td>
<td>Location provider identifier (e.g. GPS, network)</td>
</tr>
<tr>
<td>media_provider</td>
<td>Multimedia sensor’s identifier (i.e. front/back camera, microphone)</td>
</tr>
<tr>
<td>capture_duration</td>
<td>Capturing time, measured in seconds</td>
</tr>
<tr>
<td>capture_interval</td>
<td>Time between two capturing actions, measured in seconds</td>
</tr>
</tbody>
</table>

Table A. 2: Description of JSON attributes that are used in the ETP in the ACQUIRE message

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference_timestamp</td>
<td>Unix timestamp of collection start (in nanoseconds)</td>
</tr>
<tr>
<td>event_timestamp</td>
<td>Time passed since smartphone’s uptime (in nanoseconds)</td>
</tr>
<tr>
<td>timestamp</td>
<td>Unix timestamp (in seconds)</td>
</tr>
<tr>
<td>is_last_known</td>
<td>Used for cached locations</td>
</tr>
</tbody>
</table>

Appendix B

Table B.1 summarizes the details pertinent to the sensor hardware in the two test devices, i.e. Nexus S and LG Optimus L3. These details must be taken into consideration when data acquisition is being configured (e.g. for the selection of transport channel), as well as in the manual analysis of the recovered data.

Table B. 1. Sensor data details pertinent to the two test devices

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Values</th>
<th>Range (Nexus S)</th>
<th>Range (LG)</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>X, Y, Z</td>
<td>19.6 (abs)</td>
<td>39.22(abs)</td>
<td>m/s²</td>
</tr>
<tr>
<td>Camera</td>
<td>raw</td>
<td>N/A</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>GPS</td>
<td>X,Y</td>
<td>[0.0-90.0],</td>
<td>[0.0-90.0],</td>
<td>rad</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[-180.0-180.0]</td>
<td>[-180.0-180.0]</td>
<td></td>
</tr>
<tr>
<td>Gravity</td>
<td>X, Y, Z</td>
<td>19.6 (abs)</td>
<td>39.22 (abs)</td>
<td>m/s²</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>X, Y, Z</td>
<td>34.9 (abs)</td>
<td>-</td>
<td>rad/s</td>
</tr>
<tr>
<td>Humidity</td>
<td>Single</td>
<td>-</td>
<td>-</td>
<td>%</td>
</tr>
<tr>
<td>Light</td>
<td>Single</td>
<td>[0.0 – 30.0x10⁷]</td>
<td>-</td>
<td>Lx</td>
</tr>
<tr>
<td>Linear Acceleration</td>
<td>X, Y, Z</td>
<td>19.6</td>
<td>39.22</td>
<td>m/s²</td>
</tr>
<tr>
<td>Magnetometer</td>
<td>X, Y, Z</td>
<td>2000.0</td>
<td>1600.0</td>
<td>µT</td>
</tr>
<tr>
<td>Microphone</td>
<td>raw</td>
<td>N/A</td>
<td>N/A</td>
<td>-</td>
</tr>
<tr>
<td>Orientation</td>
<td>X, Y, Z</td>
<td>[0.0-360.0]</td>
<td>[0.0-360.0]</td>
<td>rad</td>
</tr>
<tr>
<td>Pressure</td>
<td>Single</td>
<td>-</td>
<td>-</td>
<td>hPa/mbar</td>
</tr>
<tr>
<td>Proximity</td>
<td>Single</td>
<td>[0.0 – 5.0]</td>
<td>[0.0 – 1.0]</td>
<td>Cm</td>
</tr>
<tr>
<td>Rotation Vector</td>
<td>X, Y, Z</td>
<td>1.0 (abs)</td>
<td>1.0</td>
<td>rad</td>
</tr>
<tr>
<td>Temperature</td>
<td>Single</td>
<td>-</td>
<td>-</td>
<td>°C</td>
</tr>
</tbody>
</table>
Appendix C

An example of a vector containing sensor’s info in a name-value pair is the following:

1. RESOLUTION: 0.038300782
2. POWER: 0.03
3. "NAME":"Linear Acceleration Sensor"
4. "VERSION":1
5. "MINIMUM_DELAY":0
6. "MAXIMUM_RANGE":39.22
7. "TYPE":10
8. "VENDOR":"Google Inc."

Appendix D

Tables D.1-D.6 summarize the results of our tests regarding bandwidth consumption for each sensor in the test devices.

Table D.1: Bandwidth requirements for Nexus S (excluding multimedia sensors and GPS)

<table>
<thead>
<tr>
<th>Sensor/Rate (B/s)</th>
<th>NORMAL</th>
<th>UI</th>
<th>GAME</th>
<th>FASTEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>287,2</td>
<td>288,4</td>
<td>288,8</td>
<td>289,2</td>
</tr>
<tr>
<td>Corrected gyroscope</td>
<td>1274,8</td>
<td>1274,4</td>
<td>1274,4</td>
<td>1307,34</td>
</tr>
<tr>
<td>Gravity sensor</td>
<td>280,8</td>
<td>280</td>
<td>282</td>
<td>281,6</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>636,8</td>
<td>636,8</td>
<td>636,8</td>
<td>729,94</td>
</tr>
<tr>
<td>Light sensor</td>
<td>0,14</td>
<td>0,27</td>
<td>1,74</td>
<td>6,54</td>
</tr>
<tr>
<td>Linear acceleration</td>
<td>35,6</td>
<td>281,6</td>
<td>282,4</td>
<td>282,4</td>
</tr>
<tr>
<td>Magnetometer</td>
<td>48</td>
<td>60</td>
<td>296,4</td>
<td>295,6</td>
</tr>
<tr>
<td>Orientation sensor</td>
<td>35,6</td>
<td>282,4</td>
<td>281,6</td>
<td>282,4</td>
</tr>
<tr>
<td>Proximity sensor</td>
<td>0,14</td>
<td>0,14</td>
<td>0,14</td>
<td>0,14</td>
</tr>
<tr>
<td>Rotation vector</td>
<td>35,6</td>
<td>281,6</td>
<td>280,4</td>
<td>280,8</td>
</tr>
</tbody>
</table>

Table D.2: Bandwidth requirements for Nexus S (only multimedia sensors)

<table>
<thead>
<tr>
<th>Multimedia sensors on Nexus S</th>
<th>Raw Data</th>
<th>JSON</th>
<th>JSON GZipped</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microphone</td>
<td>1699,2</td>
<td>4713,47</td>
<td>1668,7</td>
</tr>
<tr>
<td>Video low (h264)</td>
<td>29478,54</td>
<td>82345,87</td>
<td>30602,07</td>
</tr>
<tr>
<td>Video high (720p, h264)</td>
<td>158121,1</td>
<td>427204,4</td>
<td>161070,54</td>
</tr>
</tbody>
</table>

Table D.3: Bandwidth requirements for LG Optimus L3 (excluding multimedia sensors and GPS)

<table>
<thead>
<tr>
<th>Sensor/Rate (B/s)</th>
<th>NORMAL</th>
<th>UI</th>
<th>GAME</th>
<th>FASTEST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>30</td>
<td>99,2</td>
<td>297,6</td>
<td>587,2</td>
</tr>
<tr>
<td>Gravity sensor</td>
<td>30</td>
<td>100</td>
<td>297,6</td>
<td>586,4</td>
</tr>
<tr>
<td>Linear acceleration</td>
<td>30</td>
<td>99,6</td>
<td>296,8</td>
<td>588,8</td>
</tr>
<tr>
<td>Magnetometer</td>
<td>30</td>
<td>100</td>
<td>297,6</td>
<td>297,2</td>
</tr>
<tr>
<td>Orientation sensor</td>
<td>30,4</td>
<td>98</td>
<td>298</td>
<td>297,2</td>
</tr>
<tr>
<td>Proximity sensor</td>
<td>0,14</td>
<td>0,14</td>
<td>0,14</td>
<td>0,14</td>
</tr>
<tr>
<td>Rotation vector</td>
<td>30</td>
<td>99,6</td>
<td>297,2</td>
<td>589,2</td>
</tr>
</tbody>
</table>
### Table D.4: Bandwidth requirements for LG Optimus L3 (only multimedia sensors)

<table>
<thead>
<tr>
<th>Multimedia sensors on LG L3</th>
<th>Raw Data</th>
<th>JSON</th>
<th>JSON GZipped</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microphone on LG L3</td>
<td>755,64</td>
<td>2095,74</td>
<td>675,6</td>
</tr>
<tr>
<td>Video low on LG L3</td>
<td>7387,57</td>
<td>20659,27</td>
<td>7636,1</td>
</tr>
</tbody>
</table>

### Table D.5: Bandwidth requirements for GPS (both devices)

<table>
<thead>
<tr>
<th>Location providers</th>
<th>Raw data</th>
<th>JSON</th>
<th>JSON GZipped</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS/Network Provider</td>
<td>53,34</td>
<td>441,6</td>
<td>197,34</td>
</tr>
</tbody>
</table>

### Table D.6: Bandwidth requirements for accelerometer (LG Optimus L3)

<table>
<thead>
<tr>
<th>Accelerometer on LG L3</th>
<th>Raw Data</th>
<th>JSON</th>
<th>JSON GZipped</th>
</tr>
</thead>
<tbody>
<tr>
<td>NORMAL</td>
<td>30</td>
<td>237,47</td>
<td>3,94</td>
</tr>
<tr>
<td>UI</td>
<td>99,2</td>
<td>1110,2</td>
<td>5,37</td>
</tr>
<tr>
<td>GAME</td>
<td>297,6</td>
<td>3275,67</td>
<td>7,94</td>
</tr>
<tr>
<td>FASTEST</td>
<td>587,2</td>
<td>6363,8</td>
<td>11,57</td>
</tr>
</tbody>
</table>

Tables D.7-D.8 summarize the results of our tests with battery consumption for each sensor in the test devices.

### Table D.7: Power requirements Nexus S

<table>
<thead>
<tr>
<th>Sensors</th>
<th>Power (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>0,23</td>
</tr>
<tr>
<td>Magnetometer</td>
<td>6,80</td>
</tr>
<tr>
<td>Light sensor</td>
<td>0,75</td>
</tr>
<tr>
<td>Proximity sensor</td>
<td>0,75</td>
</tr>
<tr>
<td>Gyroscope</td>
<td>6,10</td>
</tr>
<tr>
<td>Rotation vector</td>
<td>13,13</td>
</tr>
<tr>
<td>Gravity sensor</td>
<td>13,13</td>
</tr>
<tr>
<td>Linear acceleration</td>
<td>13,13</td>
</tr>
<tr>
<td>Orientation sensor</td>
<td>13,13</td>
</tr>
<tr>
<td>Corrected gyroscope</td>
<td>13,13</td>
</tr>
</tbody>
</table>

### Table D.8: Power requirements LG L3

<table>
<thead>
<tr>
<th>Sensors</th>
<th>Power (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accelerometer</td>
<td>0,03</td>
</tr>
<tr>
<td>Magnetometer</td>
<td>0,50</td>
</tr>
<tr>
<td>Proximity sensor</td>
<td>0,02</td>
</tr>
<tr>
<td>Gravity sensor</td>
<td>0,03</td>
</tr>
<tr>
<td>Linear acceleration</td>
<td>0,03</td>
</tr>
<tr>
<td>Orientation sensor</td>
<td>0,53</td>
</tr>
</tbody>
</table>
Snippet D.1 includes the acquisition configuration entry and the collected data from the scenario of section 5.3.3.

```json
d0a3ecf6e813d6b9ab50c31c84e53c7e71eb92

```

Figure D.1 presents the timeline of events of section 5.3.4.

```
829028f9722bdc716e6f73a47451d17b9bbfea06

```

**Appendix E**

Table E.1 summarizes the results of our experiments with the light sensor.

<table>
<thead>
<tr>
<th>Scenario tested</th>
<th>Illumination value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Device in a pocket</td>
<td>4.68</td>
</tr>
<tr>
<td>Room without light tubes on</td>
<td>4.72</td>
</tr>
<tr>
<td>Room without light tubes on, near 4 PC monitors</td>
<td>21.16</td>
</tr>
<tr>
<td>Room with fluorescent light tubes</td>
<td>246.66</td>
</tr>
<tr>
<td>Right under fluorescent light tube</td>
<td>688.48</td>
</tr>
<tr>
<td>20cm from 4 fluorescent light tubes</td>
<td>6483.15</td>
</tr>
<tr>
<td>Cloudy noon daylight</td>
<td>8834.43</td>
</tr>
<tr>
<td>5cm from 4 fluorescent light tubes</td>
<td>9891.19</td>
</tr>
<tr>
<td>Sunny morning 08:30 under thin shadow</td>
<td>10874.4</td>
</tr>
<tr>
<td>Sunny morning 09:30 facing sun</td>
<td>( \gg 1000 ) (spikes from ( 1 \times 10^3 ) to ( 6 \times 10^4 ))</td>
</tr>
</tbody>
</table>
Daylight fluorescent tubes give an illumination value approx. $10^4$ when the distance between the tube and the sensor is kept to a minimum. In our experiments, we noticed that illumination values are greater than $10^4$ when the device is outdoors on a sunny day. While these tests are only indicative, this sensor may be useful for a forensic investigation process, in the sense that if it is combined with location data it may give specific information regarding the context of the device (i.e. device is indoors or outdoors).