SecureTracking:
secure position monitoring of children and people with
mental disabilities

Rui Diogo de Sousa Morais

Thesis to obtain the Master of Science Degree in
Information Systems and Computer Engineering

Supervisors: Professor João Pedro Faria Mendonça Barreto
Professor Ricardo Jorge Fernandes Chaves

Examination Committee
Chairperson: Professor José Manuel da Costa Alves Marques
Supervisor: Professor João Pedro Faria Mendonça Barreto
Member of the Committee: Professor Ricardo Jorge Feliciano Lopes Pereira

June 2015
Dedicated to my beloved parents and brother
Acknowledgments

My academic life would not be so enriching on a personal level without having the people I had on my side. It all starts with a freshman and ends with a passion that holds precious memories that bind us to the faculty. I owe all my being to all who walked and walk beside me. Good and bad memories, all part of me, all thanks to Maria Serralha, Miguel Brázia, Gonçalo “Xau” Ribeiro, Ricardo Paradela, Sérgio Isidoro, Filipe Cruzinha, Fábio “Milu” Almeida, Henrique Moisés, Bárbara Oliveira, Dário Ornelas and Tiago Soares. To my companions who pursue with me the magic dream and juvenile irreverence, always complying with the academic tradition, Francisco Patrocínio, Nuno Deus, Marta Santos, Catarina Rocha, Miguel Rodrigues, and many, many others. To all my friends of CPLEIC, Magnum Consilium Praxis do Instituto Superior Técnico, Irmandade da Capada and to all that shared with me all the days wearing Capa e Batina, when only Fado and Portuguese Guitars could verbalize our feelings. Thank you all, for everything.

Special thanks to my beloved parents and brother, my sources of human strength to whom I dedicate this thesis. To them I owe the opportunity of this life that I can now realize. Thank you so much. To my grandmothers Palmira and Fátima, and to my aunt/godmother Fátima, thank you for all the support and help.

To my supervisors, to whom I gave a lot of headaches, I thank you for the patience, the help and the opportunity of doing this thesis.
Resumo

Atualmente existem diversos sistemas dedicados a monitorizar a posição de objectos, veículos ou mesmo pessoas. Grande parte testes sistemas de monitorização focam-se na eficiência energética, implementando estratégias para maximizar a duração de bateria ao escolherem de forma cuidada os métodos de localização, ou usando sensores alternativos como acelerómetros e bússolas para estimar a posição actual. Ao monitorizar pessoas, nomeadamente crianças ou pessoas com deficiências mentais, surgem diversos problemas relacionados com segurança e privacidade. São poucos os sistemas de monitorização da posição que implementam mecanismos de segurança relativamente à informação transmitida e ao próprio processo de localização.

De forma a cumprir com estes requisitos, esta dissertação propõe o SecureTracking, um sistema capaz de monitorizar a posição de uma pessoa, aplicar algum controlo sobre essa posição através de vedações geográficas (geofencing), enquanto assegura confidencialidade, integridade, autenticação e frescura da informação transmitida. A solução proposta considera ainda mecanismos de anti falsificação da localização obtida. Foi implementado um protótipo funcional usando smartphones Android e um servidor dedicado que actua como intermediário de comunicação. Para alcançar os requisitos de segurança, foi usado o algoritmo AES-128 bits em modo CTS para cifrar e HMAC com SHA256 para assegurar a integridade e autenticação das mensagens. A solução implementada cumpre estes requisitos de segurança usando os algoritmos de menor consumo energético e praticamente sem introduzir tempo de atraso.

Palavras-chave: Monitorização da posição, vedação geográfica, comunicação segura, monitorização segura
Abstract

Currently there are many systems dedicated to monitor the position of objects, vehicles and people. Most of these monitoring systems focus on the energy efficiency, implementing strategies to maximize battery lifetime by carefully choosing the localization method, or using alternative sensors like accelerometer and compass to estimate the current location. When monitoring the position of a person, specially children or people with mental disabilities, security and privacy issues arise. Almost none of current tracking systems implement security mechanisms regarding the transmitted information and the localization process itself.

To fulfill these requirements, this thesis proposes SecureTracking, a tracking system capable of monitoring the position of a person, applying some control on that position through geofencing, while ensuring the confidentiality, integrity, authentication and freshness of the transmitted information. The proposed solution also considers mechanisms of anti-spoofing on the obtained location information. A fully functional prototype was implemented using Android smartphones and a dedicated server to act as intermediate. To meet the security requirements, the AES-128 bit algorithm in CTS mode was used to cipher messages and the HMAC with SHA256 was used to ensure authentication and integrity. The implemented solution meets all these security requirements while using the less energy-intensive algorithms and introducing almost no time delay.

Keywords: Position monitoring, geofencing, secure communications, secure tracking
## Contents

Acknowledgments ................................................................................. v
Resumo ........................................................................................ vii
Abstract ........................................................................................... ix
List of Tables .................................................................................... xiii
List of Figures ................................................................................... xv

1 Introduction ......................................................................................... 1
   1.1 Motivation ................................................................................ 1
      1.1.1 Ethics on human tracking ...................................................... 2
      1.1.2 Security on human tracking ...................................................... 2
   1.2 Objectives and Requirements ....................................................... 3
   1.3 Main contributions .................................................................. 4
   1.4 Dissertation outline ................................................................. 4

2 State-of-the-Art ................................................................................. 5
   2.1 Concepts ................................................................................ 5
      2.1.1 Geofencing ........................................................................ 5
      2.1.2 Localization Methods .......................................................... 6
   2.2 Related Work .......................................................................... 9
   2.3 Conclusion ............................................................................ 19

3 Proposed Solution ............................................................................ 21
   3.1 Proposed Solution Description .................................................. 22
   3.2 Monitored Device .................................................................. 23
   3.3 Tutor Device .......................................................................... 24
   3.4 Repository Server .................................................................. 25
   3.5 Requirements Fulfilment ........................................................... 25
   3.6 Conclusion ............................................................................ 25

4 Implementation ................................................................................. 27
   4.1 Mobile Components - Tutor and Monitored Devices ................. 27
      4.1.1 Software Artifacts ............................................................... 27
4.2 Fixed Components ......................................................... 32
   4.2.1 Repository Server .................................................. 32
   4.2.2 Tutor PC Application .............................................. 32
4.3 Communication Channels .............................................. 33
4.4 Security ........................................................................ 33
   4.4.1 Communication Security .......................................... 34
   4.4.2 Localization Security .............................................. 39
4.5 Implementation Discussion ............................................ 40

5 Characterization and Evaluation ..................................... 41
   5.1 Accuracy Characterization ............................................ 42
   5.2 Time Measurements .................................................. 44
      5.2.1 Localization Methods ........................................... 45
      5.2.2 MAC Algorithms ................................................ 45
      5.2.3 Cipher Algorithm ................................................ 46
   5.3 Battery Consumption ................................................ 47
      5.3.1 MAC Algorithms ................................................ 47
      5.3.2 Localization Methods ........................................... 47
      5.3.3 PIN Alert Impact ................................................ 49
   5.4 Discussion .............................................................. 49

6 Conclusions ..................................................................... 53
   6.1 Achievements .......................................................... 54
   6.2 Future Work ........................................................... 55

Bibliography .................................................................... 62

A Accuracy Maps .................................................................. 63
List of Tables

5.1 Path A - statistics on the distance between reported and fixed locations . . . . . . . . . . 43
5.2 Path B - statistics on the distance between reported and fixed locations . . . . . . . . . . 43
5.3 Path A - statistics on the difference between real and system accuracy . . . . . . . . . . . 44
5.4 Path B - statistics on the difference between real and system accuracy . . . . . . . . . . . 44
5.5 Time statistics for each localization method . . . . . . . . . . . . . . . . . . . . . . . . . . . 45
5.6 MAC Algorithms - time statistics using the first 80% of collected data (in ascending order) 46
5.7 AES using 128 bits key in CTS mode - time statistics using the first 80% of collected data 46
    (in ascending order) . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 46
5.8 Battery loss statistics using HMAC-SHA256 and CBC-MAC16 . . . . . . . . . . . . . . . . 47
5.9 Statistics on battery consumption per localization method . . . . . . . . . . . . . . . . . . 48
# List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Localization using lateration</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>Localization using angulation</td>
<td>7</td>
</tr>
<tr>
<td>2.3</td>
<td>SensTrack architecture</td>
<td>11</td>
</tr>
<tr>
<td>2.4</td>
<td>WheelLoc architecture</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>Enhanced Localization Solution architecture</td>
<td>14</td>
</tr>
<tr>
<td>2.6</td>
<td>Elderly Tracking System prototype</td>
<td>16</td>
</tr>
<tr>
<td>2.7</td>
<td>SET Initiated Reporting</td>
<td>16</td>
</tr>
<tr>
<td>2.8</td>
<td>SET Initiated Reporting</td>
<td>16</td>
</tr>
<tr>
<td>3.1</td>
<td>SecureTracking system overview</td>
<td>22</td>
</tr>
<tr>
<td>4.1</td>
<td>GCM messages flow</td>
<td>34</td>
</tr>
<tr>
<td>4.2</td>
<td>Messages types and their fields</td>
<td>38</td>
</tr>
<tr>
<td>4.3</td>
<td>Renew keys sequence diagram</td>
<td>39</td>
</tr>
<tr>
<td>5.1</td>
<td>Accuracy using A-GPS</td>
<td>43</td>
</tr>
<tr>
<td>5.2</td>
<td>Accuracy using Network</td>
<td>43</td>
</tr>
<tr>
<td>5.3</td>
<td>Battery consumption in normal operation and using PIN alerts</td>
<td>50</td>
</tr>
<tr>
<td>A.1</td>
<td>Accuracy using A-GPS (test 1)</td>
<td>63</td>
</tr>
<tr>
<td>A.2</td>
<td>Accuracy using Network (test 1)</td>
<td>63</td>
</tr>
<tr>
<td>A.3</td>
<td>Accuracy using A-GPS (test 2)</td>
<td>63</td>
</tr>
<tr>
<td>A.4</td>
<td>Accuracy using Network (test 2)</td>
<td>63</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

Tracking systems are becoming increasingly important in our daily life, and their use increasingly influences our behaviour. Keeping track of objects or even people has become usual in various and different scenarios. One kind of position tracking usage is mail monitoring. For example, online shopping often involves shipping orders by mail, so postal companies offer the possibility of tracking packages’ position to their customers. Tracking systems are also used to track employees’ position when they drive a company vehicle during office hours and by insurance companies in Pay-as-You-Drive (PAYD) models. Instead of having a fixed insurance fee to pay-per-year, in PAYD systems some data is collected from the vehicles, such as the kilometres driven, types of zones and roads, time of the day and average speeds. Another kind of use involves tracking personal location. Nowadays this kind of information is more accessible and shareable since more and more devices are equipped with sensors that keep their owner’s position. Over the past few years we have witnessed the growing use of smartphones [23, 24, 67] that are capable of, among other things, obtaining their position. Using this capability, most social network users voluntarily share their position to their friends. Even recent digital cameras with built-in GPS modules save the coordinates where a certain picture was taken in it as meta-data.

Such usage of the locating capabilities usually does not require much security efforts since the information is voluntarily shared and the purpose of these systems is, generally, recreational. But, on the other hand, if a tracking system’s objective is to maintain someone safe, the security requirements are much higher and many risks should be taken into account. A secure tracking system must aim, not only at obtaining the person’s position, but also at ensuring the proper security mechanisms to keep that information secure to ensure the person’s own security. The focus of this thesis is to provide a tracking system to monitor the position of children or people with mental disabilities, with geofencing capabilities and provided with adequate safety systems.

1.1 Motivation

Monitoring someone’s position raises some security and privacy issues. Depending on the target of that monitoring action, there is a trade-off between security measures and acceptable privacy loss. Tracking
the position of a healthy adult that represents no harm to the society may become abusive in terms of privacy and even raises ethical issues. There are some concerns that these actions could actually harm the trust relationship that should exist between children and their parents [43]. Another target group susceptible to be monitored is people with mental illness. Dementia patients are likely to wander and get lost, so although monitoring their position may result in some privacy invasion, it may also reveal itself as an important measure to keep them safer.

1.1.1 Ethics on human tracking

There are some research works in ethics on tracking systems aimed at human monitoring. In [40], Manoharan addresses the privacy and ethics on tracking humans and proposes a framework for tracking software with three points: the localization should be demand-driven; privacy of the monitored person should be respected and his location only revealed to authorized ones; and the framework must be technology-independent as far as possible. There are no ethical problems when GPS is used only as navigation assistance [41], but the same does not apply when that location data is transmitted to another location, as it raises privacy problems. However, currently there are already products that track one’s movements imposed by parents or caregivers who have immediate authority. When tracking dementia wandering, the authors of [61] state that one could either control their presence in protected areas or constantly monitor their position. In [39] the author tries to achieve an appropriate cooperation of technology regarding peoples’ needs. First it aims at Alzheimer sufferers’ properties and problems, conducting interviews. Most patients stated they are aware of their limitations but nevertheless want to remain socially active. Then, some research testing took place, where patients carried a GPS device and caregivers monitored their position using a PC. The tests revealed that the caregivers do not require information at all times, since they just want to know if the patient is having a blackout or is wandering. Both caregivers and patients stated that the communication should be subtle, not excessive and used only when needed. Regarding the patients feedback, they don’t want to be intrusively contacted. Also, they revealed it was important that the device’s design is simple and that the use of such devices could alleviate stress for families and relatives. To avoid situations when the patient forgets to carry the device, a possible solution is to implement the system in an object worn daily, which raises ethical issues regarding user consent and privacy.

1.1.2 Security on human tracking

The security aspect arises in tracking systems when the location is critical information about a person. Having as target groups children or mentally illness individuals, extra care must be taken into account. Contempt on security issues may result in scenarios where the location of a child may be visible to kidnappers, or dementia patients to be lost indefinitely. Some attacks on tracking systems may occur at different points of the information flow. Spoofing attacks may lead to incorrect localization, attacks on the communication channels may reveal the location to unauthorized individuals, or allow tampering with the exchanged messages pretending normal operation.
1.2 Objectives and Requirements

Our goal is to develop a platform to help relatives or tutors to consult the position of certain individuals under their responsibility. Additionally, these relatives or tutors may be informed if the monitored person moves outside a certain predefined secure area. This concept is especially useful when controlling children’s position in environments of high confusion (e.g., big cities) and exposure to several risks (loss, abduction). It may also be useful to track people with mental illness. If the system detects that the monitored person went outside a secure zone or entered a dangerous one it generates an alert to the person responsible for their monitoring. This concept is known as geofencing, detailed Section 2.1.1. Although there is no need for this warning, the responsible person may be periodically briefed about the position of their dependents, even without applying geofencing.

As a final product, it is desirable to have a system implemented in devices that are easy and comfortable to be carried, like smartphones. Such devices allow the development and installation of applications that take advantage of their sensors in order to locate themselves and report that information. Alternatively, a dedicated device could be developed (e.g., a chip to be carried in a backpack or a shoe), containing only the necessary modules to locate and communicate.

Both the process of obtaining the current location and transmitting the obtained information to another device requires security caution. The system operation is also affected by the available resources and the defined energy consumption policy directly affects the lifetime of the system. Due to these factors, the final platform must meet a set of requirements to maximize its usability and usefulness. The requirements defined for the SecureTracking system are:

- **Functional Requirements:**
  - **Localization** - The system must be capable of obtaining a person’s position using certain localization methods.
  - **Geofencing** - The system must offer the functionality of monitoring a person’s position in respect to certain geofences (concept explained in Section 2.1.1).
  - **Communication of Information/Warnings** - The system must be capable of communicating the position or a warning to the tutor if some geofence rule is violated.
  - **Position on-demand** - The system must allow a tutor to ask and get the monitored person’s position on-demand.

- **Non-Functional Requirements**
  - **Security** - The system must have certain security mechanisms to assure:
    - Confidentiality - Only authorized individuals can access the information about the position and warnings.
    - Integrity - The location information that the tutor receives matches the information sent by the monitored person. Similarly, the location determined by the device’s sensors should
be correct, that is, should effectively correspond to the real position of that person (within a certain accuracy)

- Authenticity - The location information received by the tutor has its origin solely at the monitored device.
- Freshness - The messages exchanged must be recent and not susceptible to replay attacks.

- **Performance** - The mobile system has limited energy and processing resources that should be used efficiently.
- **Longevity** - The system must be capable of operating autonomously for long periods without needing to replace or charge batteries.
- **Usability** - The system will be carried by children or people with mental disability, so it should be easy and comfortable to transport.
- **Ethics** - The system must take on account privacy and ethical issues as far as possible.

### 1.3 Main contributions

One of main contributions of this thesis is the system design. It allows many tutors to monitor the same person while ensuring the security requirements almost without the final user intervention. It is simple and complete. The implementation of the proposed solution is another important contribution, as it uses everyday devices such as smartphones and do not require developing and deploying new infrastructures to meet the requirements. There is a greater concern about security issues than the state-of-the-art, that are addressed with the appropriate mechanisms. Also, the experimental evaluation show the feasibility of SecureTracking as a monitoring system, demonstrating that a tracking system can provide security measures with little impact on performance and energy consumption.

### 1.4 Dissertation outline

The next sections of this thesis are organized as follows. Chapter 2 presents the State-of-the-Art divided in two subsections: one explaining some important concepts in the context of the related work and this thesis; and another containing an overview analysis of some related work, the problems they addressed or did not solve and the proposed or implemented solutions regarding tracking systems. After the State-of-the-Art, the Proposed Solution is presented in Chapter 3, taking into account the defined requirements and the analysis of the state-of-the-art. The Proposed Solution is followed by the Implementation chapter, where proposed solution is explained targeting an Android system. Chapter 5 explains how the implementation was characterized and evaluates it. The last chapter concludes this thesis, presenting an overall analysis of the proposed system. It also includes some remarks on possible future work.
Chapter 2

State-of-the-Art

There are various uses for tracking systems, from monitoring mail packages, employees’ vehicles, to tracking children on their path to school, or dementia patients. Different uses also imply different focus on functionalities, characteristics, different concepts and methods of localization. Taking in account the motivation for this thesis and the requirements proposed, it is important to analyse two aspects: the concepts involved and the related work on this area. Given that one of the proposed requirements is to allow monitoring a person's position related to a certain area, it is important to understand the concept of geofencing. On the other hand, the major functionality of this project is the localization process. Given that there are many sources, structures, techniques and security issues involved in these operations, an overview on them is also provided. After that, taking in account the context of this project, some related works are analysed, mapping their architectures, proposed solutions and unsolved problems to this thesis context and requirements.

2.1 Concepts

For a better understanding of the concepts involved in this project, they are explained in some detail in this section. The topics covered are geofencing and localization methods.

2.1.1 Geofencing

The geofencing technique [56] consists on the combination of two operations: first, regular monitoring of the position of some mobile device; and on the other hand the definition of a geographic area of interest. In this context, this area of interest is often called a geofence (geographic fence). The whole process consists on verifying if the position of the mobile device is inside or outside that geofence in order to generate an alert. It can also be defined a temporal validity by associating a valid schedule or calendar to a geofence.

Different techniques may be used to define the area that corresponds to the geofence:

- Polygon area - the geofence is defined as a polygon where the vertices are defined by coordinates.
• Point of Interest (POI) - the geofence is simply defined by the coordinate of the central point and a radius, defining a circular area.

• Route - allows to control if the mobile device follows or keeps close to a particular route.

In the context of this project, in order to apply some control on the position of the monitored person, it is important to apply the concept of geofencing to define areas where that person is allowed or disallowed to be at.

2.1.2 Localization Methods

There are various ways already studied and put into practice for positioning using mobile devices. Some of them are commonly used in smartphones everyday. These methods differ from each other in characteristics such as accuracy, speed, processing cost, resource and battery consumption. In this project, three of the most commonly used methods nowadays are addressed:

1. Using mobile network antennas (e.g., GSM antennas).

2. Using Wi-Fi Access Points.

3. Using (Assisted-)GPS.

Mobile Network Antennas

Cell-ID The localization process using mobile antennas is made using some parameters, namely the cell identification (Cell-ID) and the Location Area Identifier (LAI) [65]. A Base Transceiver Station (BTS) covers a certain area which is composed by various cells, each of them with a unique identifier. The cells are grouped in greater groups identified by the LAI. A mobile device always knows its Cell-ID (the cell where it is) and the LAI, and from those values it can estimate its location. It turns out that this estimate depends on many factors: the density of antennas in that area, interference that result in signal noise, the choice of the antenna to connect (it might not be the closest one [65]).

For example, Google has carried on the creation of a database with the location (although incomplete) of the antennas that serve the mobile network. This service may be used though the The Google Maps Geolocation API [21]. Being able to communicate with the servers that hold this information it is possible to locate the BTS that is serving a certain mobile device and determine its location. There are other services that also maintain information about mobile antennas and their locations that could also be used in localization (e.g., OpenCellID [25]).

The authors of [65] conducted a test to compare the location obtained using Cell-ID and GPS (the method that has greatest accuracy), obtaining a mean difference of about 500 metres in an European urban center. In [35] the authors determined an accuracy of about 500 metres in urban environment (micro cells) and an accuracy in a range of 1 to 10 kilometres in sub-urban environment.
Lateration & Angulation  Besides the localization using Cell-ID and LAI, it is also possible to obtain the location by triangulation of mobile antennas through lateration and angulation [42] [28].

Lateration consists in determining the position using the distance to different points. Making a parallel with this project, a mobile device may determine its location using the approximate distance to the mobile antennas. Using one antenna the only information that can be deduced is that the device is within a circular area which radius is that approximate distance. However, using the distance to several antennas it is possible to increase the accuracy of the localization by intersecting the areas of the different antennas. In a bi-dimensional plan, three reference points and the distance to each of them are needed to determine the position, so in the context of this project the distance to three antennas nearby is needed, as illustrated in Figure 2.1.

Angulation consists in using angles to determine a position. For example, being possible to know the direction of the signal of two antennas and knowing the distance between them, it is possible to determine the current position. However, this solution needs to know the distance between each antennas, as illustrated in Figure 2.2.

GSM Security  In terms of its security, the original architecture of the GSM network intended to authenticate its mobile users, maintain confidentiality of users’ data and signal information, the anonymity of theirs subscribers and the usage of the SIM card (Subscriber Identity Module) as a security module. However, the authors of [64, 29, 30] point out flaws in the existing structure, namely that the A5 algorithm, which is used in encryption, is considered weak and susceptible to cryptanalysis, the network does not provide end-to-end security, that is, in some points the messages are in plain text, and also it does not provide mechanisms for mutual authentication.

From the solutions presented by the authors of [64], the most appropriate is the end-to-end security, that is, establishing a secure channel at the end-entities. For the exchange of SMS messages, that measure may be the establishment of ciphering keys between those entities.
Wi-Fi Access Points

Localization using Wi-Fi Access Points (APs) signals, also known as Wi-Fi-based Positioning System (WPS), follows a similar pattern as Cell-ID positioning: analysing nearby Wi-Fi APs and knowing their approximate location allows a device to locate itself. Most mobile devices like smartphones and tablets have Wi-Fi modules and if they have access to a database of Wi-Fi APs’ location, this is viable solution in localization. To better understand how WPS works, its architecture is briefly explained below, just like its security analysis.

Wi-Fi-based Positioning System (WPS)  The WPS takes advantage of the identifiers of Wi-Fi APs to obtain an approximate location [18]. Wi-Fi APs may be identified by their Media Access Control (MAC) address and their Service Set Identifier (SSID), commonly known as the network’s name. The MAC address identifies an actual physical equipment and is visible in the packets sent by devices that participate in wireless communications, whether ciphered or not. The location of Wi-Fi APs has been carried out by different entities [19, 72, 46, 36]. This method is known as “wardriving” and consists in progressively detect Wi-Fi APs, register their identity, signal strength and map this information with their approximate location, usually with the aid of a GPS receiver. This data is stored in databases for later use. Locating process starts with a device detecting what Wi-Fi APs are nearby, collect their identities and query the database to try to obtain their locations. Assuming that those APs where previously mapped, the device can figure out its own approximate location. The accuracy of WPS is influenced by the number of Wi-Fi APs nearby and the fact that those may have or may have not been mapped already. Tests conducted by [59] revealed a mean accuracy of approximately 30 metres. Lately we have witnessed an increase in the deployment of Wi-Fi APs, in particular at homes but also in public spaces like stores, airports, shopping centers, so successfully mapping these APs facilitates WPS and increases its accuracy.

WPS Security  Regarding the security, WPS is susceptible to spoofing and jamming attacks [63]. A spoofing attack consists in inserting a fake signal to deceive the receptor, while a jamming attack actually prevents the receptor from receiving any signal. Another possible attack is the manipulation of the databases that hold the mapping between Wi-Fi APs and their location by injecting new data or corrupting existing one. As possible solutions the same authors propose sanity tests at the client through historical registry of APs to detect unusual discrepancies and higher security on the databases side to avoid tampering with the data.

Assisted-Global Positioning System (A-GPS)

Many mobile devices already have GPS modules capable of receiving satellites’ signals and locating themselves with high accuracy. The next section explains briefly how this localization method operates, its accuracy and energy consumption and the security issues.
**GPS** The GPS is composed by a constellation of 27 satellites (24 active and 3 in standby) [71]. The navigation system provides information about the position of each satellite and precise time set by the satellite atomic clocks. Each satellite is uniquely identified by a code that is also broadcast. To obtain its location, a GPS receptor calculates the distance to various satellites at the same time. A total of 3 satellites are needed to determine a position, however usually more are used to improve the accuracy. Fixing at a sufficient number of satellites to obtain a location may take some time, know as the Time-To-First-Fix (TTFF). This aspect has a great impact on the overall system performance, speed and time/energy consumption.

**A-GPS** However, this can be improved using Assisted-GPS (A-GPS) [32], which provides mechanisms to minimize the TTFF. In this case, a secondary source can be contacted to provide the almanacs and ephemeris, which give information on the current state of the satellite constellation, precise data on time, location of the satellites and data on their orbits. Obtaining this information in advance helps, accelerating the localization process by decreasing the TTFF. Tests conducted by [59] revealed a mean accuracy of 10 metres using GPS, and according to [35] the accuracy may vary between 10 to 100 metres in urban and 10 to 20 metres in suburban environment.

**GPS Security** A test conducted in [70] showed that GPS is susceptible to spoofing and jamming attacks. The authors successfully deceived a GPS receiver to report a fake position using a GPS Satellite Simulator. In [71] some countermeasures are presented to prevent these attacks. For example, it is possible to analyse the GPS signal strength and detect if it is too strong compared to a certain threshold or if it suffers sudden changes that could indicate a fake signal being inserted. Another possible technique is monitoring the number and identification of GPS satellites signals since GPS satellite simulators usually broadcast a greater number of signals against what is usually detected. Other sensors in the receiver like accelerometer and compass (if present) could be used to conduct sanity tests comparing the position with one using dead reckoning. Dead reckoning systems typically consists of motion sensors, such as accelerometer and compass, whose readings are used to estimate displacements by double integration [31].

### 2.2 Related Work

The state-of-the-art on tracking systems is vast on solutions, using various techniques to obtain the location, with different target audiences, that use different infrastructures and focus on different aspects. Most of them mainly aim at efficiently localization but disregard security measures. Various approaches were analysed regarding the main function (localization), their main features, but also taking in special account any security mechanisms of those solutions. Because the market for smartphones has grown in recent years, there are many solutions that take advantage on these devices' capabilities and sensors. Despite that, there are other solutions that introduce the creation or installation of new devices and infrastructures or networks of devices the work together in order to achieve localization. In this section
some of the analysed systems are documented in some detail.

As already mentioned, there are many solutions that take advantage of capabilities of smartphones to obtain the position. Regardless of the platform, most smartphones nowadays have location services and sensors, such as A-GPS, Wi-Fi-based positioning system (WPS) and Cell Tower positioning, and mobile applications are able to monitor them. Taking as point of interest the Android operating system, it is easy to find various applications under the subject of “GPS Tracking”. Usually these applications work with the same behaviour as a social network, making it easy to share current location to a group of persons. Applications like “Life360 – GPS Tracking” [37] and “Real Time GPS Tracker” [27], share the same set of features: creation of a user account and profile (by simply entering a user name, an e-mail address and a password), the need for internet connection to authenticate the user, share current location with a group of persons (“friends”) set by the user and the knowledge about the position of persons belonging to the same group.

So, these applications propose a system like a social network. To locate a person, they get information from the smartphone location sensors and share that information among the other users in the network. There are no concerns about security issues because the responsibility lies on the user when installing the application, granting it permission to access the smartphone sensors, and it is the users will to share their location with each other. This information is only shared when the user wants to, so it does not have the capability to know the current position of anyone, periodically or on demand.

Another type of mobile application is the “MobileMonitor” [1], which offers a more complete set of features. It works together with a private web portal where users can log in with their credentials. The smartphone application uploads to the system servers detailed information about the device’s state, like phone call and messaging history, last position, files and operating system data. The communication with the servers is ciphered and the information is only available in the private web portal. Concerning the tracking feature, the application uses A-GPS, Cell-ID and Wi-Fi to get the position information to send it to the servers where it is processed and compared with the geofences. In the private web portal it is possible to define geofences making usage of the Google Maps user interface. The user defines the notification policy, choosing to be notified if is inside or outside a geofence, making it possible to define secure and insecure areas. The notification is made via e-mail message containing the smartphone identification, event type (entering insecure or exiting secure area), a link to Google Maps indicating the position on the map and the date when that position was obtained.

This solution acts like a full supervisor of the mobile phone state. It collects much more information than just track the position, using too much bandwidth to upload all that information to the servers. Although the communication is ciphered, there is no information about cipher key exchanges processes, and there is no mechanisms on anti-spoofing.

The system in [16] presents an efficient solution to locate a person using an Android smartphone. The authors introduce the notion of Points of Interest (POI), and their method utilizes the distance to
them and the user’s transportation mode dynamically select the location provider and the interval they obtain the location. To determine the transportation mode, it is made an analysis to the accelerometer behaviour. The idle mode is associated with small and regular variations; the walking mode causes regular and patterned changes in short periods; the driving mode causes acceleration changes, smaller than the walking mode. The classification is made analysing the variance: for low, intermediate and high values, the transportation mode is defined as idle, driving and walking, respectively. The location provider selection is made taking into account the distance to the POI, the closer, the more accurate provider is selected. The authors state that this solution decreases the need to used GPS in about 96% and increase battery lifetime by about 75%.

This solution focus in efficiency when tracking a person. The mechanisms developed improve the uptime of the system, but it lacks methods to report the position to another entity and any security measures when obtaining the position.

Another tracking solution is SensTrack [75]. This system is divided in two parts, presented in Figure 2.3: one runs on a smartphone and collects position samples using the available providers (GPS or Wi-Fi); the other one, which runs on a remote server, processes the samples sent by the smartphone using the Gaussian Process regression. A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian (normal) distribution, and is fully specified by a mean function and a covariance function [55]. Gaussian Process Regression is the inference of continuous values with a Gaussian process prior. The GPS samples collection is not continuously active, instead, it is selectively activated at separate locations with the aid of the accelerometer and orientation sensors. For example, if those sensors detect a sudden velocity change or path turn, the system can activate the GPS to register the location.

Figure 2.3: SensTrack architecture [75]
Like the previous system, this solution focuses on energy efficiency, mainly GPS usage reduction due to its low availability (specially indoors) and high energy consumption. The authors achieved a 90% reduction on GPS usage and nevertheless managed to correctly monitor the position with high accuracy (average error of about 3 metres). Despite the great tracking capabilities, the system needs internet connection to connect to upload and reconstruct the path taken. There is no security measures on this communication, neither when collection the location samples.

The author of [11] proposes another energy efficient system with the addition of being concerned about the user privacy. It is stated that when receiving signal from various antennas, cell phones may make too much queries to know those antenna's location to locate themselves. According to the author, all these queries represent excessive resource consumption when doing geofencing, but also reveal the user's current position to the network provider who can monitor it. To work around this privacy issue, the proposed solution uses inverse positioning as an alternative. Instead of querying about the signals it receives, the mobile device checks locally if the signals it is receiving match the signals that correspond to the geofenced area. Only if these two parts match and the geofenced area contains Wi-Fi networks, the Wi-Fi module is activated to locate with more accuracy. If still necessary for more accuracy, the GPS localization can be activated, which also represents an efficiency mechanism.

Regarding the privacy, this system only reveals the geofence to the network provider. Despite that, the location information is held by the device itself, there is no communication with a second entity informing any event of entering or exiting geofenced areas. Also, it is needed to register the signals available in every desired geofence, which represents a limitation on creating geofences dynamically.

In [3] the authors propose UPTIME (Ubiquitous Pedestrian Tracking using Mobile phones), a system that applies dead reckoning to track a person. This system uses a technique of estimated step counting and gait-based algorithm to detect the step size. To detect a step, the authors analysed the magnitude of the 3 dimension acceleration and built a finite state machine capable of effectively detecting a step with an average error of about 5.7%. On the other hand, to classify the step size, the authors trained a classifier to distinguish three gait types: walking, jogging and running, reaching an accuracy of 97.7% on this classification. When performing the tracking task, it limits the average error to 6.9% in outdoor scenarios and 5.9% when indoors.

This tracking system is designed to be ubiquitous, accurate and energy efficient. But like the previous example, the location information remains on the device and is not communicated to anyone else. Also, the system does not provide geofencing capabilities, neither security measures of any kind.

WheelLoc [69] is another tracking solution that uses an indirect location strategy with a mobile phones, monitoring the device's sensors like accelerometer and magnetometer. The system assumes that the device already possesses a Cell ID database with their positions and also a street map. As presented in the architecture in Figure 2.4, WheelLoc obtains samples from the device's accelerometer and magnetometer to detect the motion state and estimate the velocity and direction in that state. Si-
multaneously, it obtains location via Cell-ID information and the database. Then, the system estimates
the turns taken and distance travelled. Combining the Cell-ID-based location, the street map for that
area and the estimation on distance and turns taken, the movement is traced by matching it with the
map using a Hidden Markov Model (HMM) and Viterbi algorithm (VA). A HMM is a statistical model com-
posed of a certain number of states, interconnected with each other with some associated probability
and where each state outputs a symbol, also with some associated probability [22]. They are called
“hidden” because only the symbol sequence is visible, the state sequence is not observed. The VA is a
dynamic programming algorithm to solve the problem of estimating the state sequence of a discrete-time
finite-state Markov process [26].

![WheelLoc architecture](69)

WheelLoc is able to obtain a location in 40 milliseconds with an accuracy of about 40 metres and
low power consumption (240 mW). Yet, the system is only focused in getting the location with efficiency,
which remains within the device and is not communicated to anyone else.

The Enhanced Localization Solution (ELS) [53] is another system than combines standard local-
ization techniques in mobile phones (GSM, Wi-Fi, GPS) with data from other sensors (accelerometer,
audio, compass, gyroscope) using Human Mobility modelling and machine learning techniques. As
demonstrated in Figure 2.5, this solution is inserted between the application layer and the operating
system localization manager (responsible for the location via GSM, Wi-Fi and GPS) which allows calling
this last only if necessary. During its normal operation, the ELS tries to predict the new position using the
Prediction Manager. To do so, every time a position is obtained it is sent to the Learning Manager mod-
ule which, periodically, applies a clustering algorithm to determine the points of interest. This information
is then passed to the Human Mobility Modelling Manager which builds a predicting graph to be used by
the Predicting Manager to predict the next movements. If this prediction doesn’t exist, the system then calls the operating system Location Manager that tries to obtain a location with the aid of the Sensor Manager (which collects data from the accelerometer, audio, compass and gyroscope sensors).

![Enhanced Localization Solution architecture](image)

**Figure 2.5: Enhanced Localization Solution architecture [53]**

The ELS acts like a middleware that aims at correctly get the location using prediction techniques, standard localization methods and data from other sensors. Like other solutions, it doesn’t communicate its location to anyone else, so the whole system operates only on the mobile phone and the information is only used by itself. There are no concerns on security.

In [14], the authors propose a tracking system to be used on a mobile phone with GPS with some security mechanisms. The solution includes the development of a Java application to be installed on a mobile phone and a web portal to check the location and apply geofencing. The application uses the GPS of the mobile phone to obtain its location with a custom period set by the user himself. Because the GPS may consume too much energy resources, it may be defined a longer period for energy saving purposes. After entering the username, password, the phone brand and interval options, the user can choose to use ciphered communication. After that, there is a step of configuration with the server which replies an ID to the application. When encryption is activated, it uses the device’s International Mobile Equipment Identity (IMEI) code to set a 128 bit encryption key, ciphers the GPS coordinates and sends this information to the server to be inserted into the database. The application uses the Advanced Encryption Standard (AES) algorithm to cipher the information. On the other hand, the web portal allows the definition of geofences with the aid of the Google Maps API. The geofences are circular, characterized by the coordinates of their centre and the radius. For each reported position, the system computes the distance to the centre of the geofence using the Haversine Formula and if it is greater than the radius than the system concludes that the position is outside the geofence. The Haversine Formula

![Haversine Formula](image)
is a formula of trigonometry used in navigation on a sphere to calculate the distance between two points using their latitudes and longitudes. If the geofence is violated (the user is outside its boundaries), an alert e-mail is sent to the person that created it. The owner of the monitored device uses the web portal to define a list of authorized users that can create and monitor geofences.

Unlike the other systems, this solution is focused on tracking and securely communicating that position to another entity, in this case a server with a database. There is not much information on the role of the IMEI in generating the ciphering key. The IMEI code is easily known and is vulnerable to spoofing attacks. There is no mechanism to regenerate and exchange the ciphering key if necessary. Although there is some concern about energy consumption when setting the localization update period to a larger value, the system only uses the GPS chip as a source of positioning, which represents a greater energy consumption compared to a Wi-Fi-based Positioning System. Also, the users that can monitor the device’s position are defined by its owner, so he can decide not to be tracked at all.

Another tracking system used in mobile phones is the Geo-Fencing [59] is presented as a framework with three main blocks: Fence, Activity Classification and Energy Consumption Profiles. The fence is a polygon and a buffer area in all of its edges. The authors introduce the notion of fence-level state transition to distinguish if the user is entering, exiting or between fences. Each transition triggers a different process. For example, if the system detects an entering event, it deactivates the GPS to save energy and only uses the cell tower’s signal strength variations to check if it remains inside or goes outside the fence. The framework uses the mobile phone’s sensors (accelerometer and compass) to distinguish between three major activity states: walking, driving and stationary. These states and the transitions between themselves affect the fence-level states and the transitions between themselves as well, namely the frequency of the positioning.

Regarding the energy consumption profiles, the authors monitored and analysed the energy consumption of various sensors (GPS, 3G, Wi-Fi, accelerometer and digital compass) in order to establish simpler states like “enable”, “disable” and “activation”. This framework combines its three blocks to define transportation modes, detect entering and exiting events on geofences a adjust the energy consumption. Although it is able to track the position, it does not provide mechanisms to communicate the position neither the fence-level states.

In [73] the authors propose a prototype elderly tracking system. As shown in Figure 2.6, this solution is composed by several components. The monitored device is an A-GPS terminal capable of 3G communication, also known as SET, or SUPL (Secure User Plane Location) Enabled Terminal. This device was designed with several energy management strategies (not described by the authors). The A-GPS terminal is provided with assistance GPS data by a single server, the Mobile Location Server (MLS). The GPS Reference Station (GSR) is a simple device that collects navigation messages from visible satellites and forward them to the A-GPS Collector and the AGPS Data Store. The A-GPS Collector transmits GSR data to the network and the A-GPS Data Store captures and reformats information of various GSR into assistance GPS data. All operation and information history is stored in a MySQL Database. The
care providers have access to a Web Client with monitoring applications implemented by a Web Engine.

![Diagram of system components and communication](image)

Figure 2.6: Components and communication between them in the solution in [73]

The system implements two methods: SET Initiated Reporting and Network Initiated Reporting. The first one, described in Figure 2.7, defines that the A-GPS terminal initiates the communication periodically with the MLS requesting assistance GPS data, sending its Cell-ID. The MLS checks the MySQL database to get the assistance data and sends it to the AGPS terminal so it can locate itself and report that position back to the MLS. Figure 2.8 shows the second method, where a third client asks the MLS for the AGPS terminal position. In this case, the MLS starts the communication with the A-GPS terminal (as opposed to the first method). The process goes on similarly from there and in the end the MLS reports the position back to the client that triggered the operation.

![Diagram of SET Initiated Reporting and Network Initiated Reporting](image)

Figure 2.7: SET Initiated Reporting  
Figure 2.8: Network Initiated Reporting

This system was tested and evaluated in various scenarios. First in an open field scenario, it was observed a mean error of about 15 metres when obtaining the position and a mean response time of
about 12 seconds from when the A-GPS initiates the localization process until it gets its position. After that, in an urban center, the system got a mean error of about 34 metres on the position and a delay of about 15 seconds. Finally, the prototype was tested on the target audience, elders, who carried the A-GPS terminal. The test concluded that the success localization rate depends on the way the user carries the terminal and that, as a matter of fact, that rate is null when the device is in the pockets. The individuals reacted in different ways when using the terminal. Some revealed themselves indifferent when others where uncomfortable. As a conclusion, it would be useful if the system was implemented in a mobile phone with GPS capabilities. In terms of lifetime, the system can operate for 7 days straight if the localization update period is no less than 10 minutes.

This solution allows geofencing using the Google Maps in the web client, but doesn’t mention any security measure when obtaining or communicating the location. It is composed of new devices so it has the overhead of not using existent platforms that could perform similar tasks, like being implemented directly on a mobile phone for comfort purposes.

The authors of [74] present an improvement of the monitoring algorithm of [73] based on the concept of Personal Common Locations (PCLs). A PCL is the combination of places where an elderly will spend most of his time and a set of states that translate into situations on which the elderly could be. A total of 4 states are introduced, organized in 2 super-states: the super-state “COMMON” includes the states “PCL” (the elderly is in a PCL) and “PCL2PCL” (the elderly travels among PCLs); the other super-state, “UNKNOWN”, holds the states “Satellite” and “Cell” (the subject location is unknown but has GPS signals on the first case, and don’t have them on the last).

The main objective here is to perform energy-saving tracking by detecting areas where the subjects spend more time and reduce the A-GPS usage and thus save battery. There are no changes regarding security measures.

The system in [60] is focused in tracking children’s position to be used in and on the way to schools. The solution is composed by three main components: a device that the children carries, a fixed system installed on the school and another device to the children’s parents or tutors. The device carried by the children has a GPS module, a GSM module and an audio recording system. The GPS module is responsible for obtaining the children’s position, which the GSM module reports to 2 receptors: the database installed on the school and the terminal held by the parents. On the other hand, the audio recording system can detect the child’s crying and trigger the GSM module to send a message warning the parents.

Although the system is designed to be carried by children, it does not analyses the security issues when obtaining the position and communicating that information. Since the only positioning system used is the GPS, this could represent too much energy consumption in comparison with another methods. The GSM communication is highly available so it is a good choice to report the position with low cost. Despite that, the system doesn’t provide geofencing capabilities.
In [44, 45] the authors present improvements to the experiments of the Hiroshima City Tracking System, which “is a safety support system children based on ad hoc network technologies”. Some objectives were established: (1) to be easily implemented and easy to add features, (2) to be able of monitoring many children efficiently, (3) adapted to the children movement to and from the school, (4) be low-cost and (5) be safe regarding suspicious individuals. The solution uses Android terminals to be carried by the children, a mesh network of computers with wireless LAN, called Tags, and a central server to save the children's tracking information. A group of children forms a cluster, built using a technique of Autonomous Clustering, from which there is always a representative terminal. This representative gets periodic information about all the members of the group via Bluetooth. When passing near a Tag, it requests the representative about group information and then reports back to the central server adding information about its location and time. This solution tries to meet the requirements mentioned above: the usage of Android terminals to help the installation of applications (1), the usage of the Autonomous Clustering technique (2 and 3), the deployment of the Tags mesh network which maintenance costs are lower than the Hiroshima City Tracking System that uses the mobile network (4). The system also uses the Secret Sharing scheme for end-to-end communication on the Tag mesh network and ciphering and information sharing through multiple channels.

Despite that this solution presents tracking and security capabilities, it also implies the installation of computers with wireless LAN in every scenario the target audience could be. This is a good solution for expected locations, but if the monitored subject moves beyond that area with the Tags, we no longer can track his position. The investment of deploying Tags would be massive to reach an acceptable coverage, and there are existing structures that could be used.

In [49] the authors propose a children tracking system without GPS. This solution aims to be energetically efficient, with low cost of the device to be carried by the children, while maintaining the tracking functionality. Thinking on the energy consumption and to avoid changing or charging the batteries too often, the authors chose not to include a GPS module. To address the lack of GPS this solution proposes that the children carry a wireless terminal and their position be estimated with the aid of previously installed fixed stations, named Transponders. The wireless terminals have a communication range of about 200 metres (with unobstructed visibility) and are installed at the children's backpacks. They are also equipped with a vibration sensor to tell if the children is walking or stationary. If it detects movement, the terminal communicates with the transponders with an interval of 3 minutes. If no movement is detected, the children may be stationary or it could indicate a malfunction, so the terminal will communicate with an interval of 1 hour. The Transponders form an independent wireless network and receive energy from solar panels and also have secondary batteries. The children’s position is considered the same as the nearest transponder. The proximity is based on the analysis of the reception electric field strength. To do so, during the communication between the terminal and the transponder, the last informs its signal strength. The authors tested the battery lifetime and concluded that a terminal with a GPS module would achieve an operating time of about 9 days. On the other hand, the system without GPS was able to operate for 5 months before needing for charge or replace batteries, so the choose to not include
the GPS has great impact on the energy consumption. During the positioning tests, the system could correctly locate the children in 61 out of 63 tests. On the failed tests, the distance between the closest and the next transponder was only 10 to 20 metres, which is not a big difference.

This system is another example of a tracking solution that would only work for a limited area covered by the transponders installation. In kidnap scenarios the children would rapidly leave that area and could not be tracked. Also, there is no security mechanisms on the communication between the terminals and the transponders.

2.3 Conclusion

In this section some important concepts were introduced for better understanding of some aspects of the state-of-the-art and this project itself. The notion of geofencing is related with one of the requirements proposed for this system: controlling the position of someone in comparison with certain areas. One could monitor if a monitored person is inside certain areas that may be defined as polygons, circular areas or even routes. There are different localization methods, varying in infrastructure used, accuracy and resources used. Three localization methods were analysed in more detail: using mobile antennas, Wi-Fi (WPS) and A-GPS. Positioning using mobile antennas uses some parameters like the Cell-ID and LAI so that mobile devices can approximate their position. This method is low in resources used, but its accuracy also is the worse, varying from 500 metres up to 10 kilometers. WPS uses Wi-Fi Access Points signals and databases that map their identity (MAC address and SSID) with their position, previously obtained the the aid of a GPS receiver. This method accuracy may varying depending on the density of APs and if they were already mapped. Tests revealed a mean accuracy of approximately 30 metres. Regarding the security, WPS is susceptible to spoofing and jamming attacks, and the databases that hold APs’ positions should be properly protected against corruption or injection of false information. Finally, the A-GPS was also analysed. It uses a constellation of satellites, whose signals are detected by GPS receivers. These receivers calculate the distance to at least 3 satellites and, knowing their orbits and position, they can locate themselves with an accuracy between 10 to 100 metres. A-GPS is also vulnerable to spoofing and jamming attacks.

In the second part, several tracking systems were analysed and documented. Although the main functionality of the solutions is the localization process, there are some different approaches that focus in some interesting aspects. The most common characteristic among them is the efficiency of the system. Many solutions include algorithms to handle the available resources, for example, dynamically changing the localization method depending on the transportation mode, or depending if they are near a certain location. Various systems implement dead reckoning as an alternative to other localization methods that consume more resources. Some solutions involve creating or installing new infrastructures to help in tracking. This kind of systems may be suitable when monitoring in controlled environment. But if, for some reason, the monitored subject moves away from the monitored area, tracking is no longer possible using only that solution. This can be really critical in situations of kidnapping or dementia patients unable to return to a safe location. Although there are many tracking systems approaches, security is,
somehow, disregarded. None of the analysed systems have mechanisms to detect or prevent possible spoofing attacks. Although some systems have ciphered communication when it is used, none of them authenticate the exchanged messages.
Chapter 3

Proposed Solution

Tracking systems that aim at monitoring humans should meet some requirements in terms of security, availability, ethics and performance, specially when applied to children or people with mental disabilities. Considering the related work analysed in the previews section, none of them fully meet the requirements. Most of them focus mainly in efficiency and disregard security issues related with the localization process and the communications they perform. Some solutions involve installing new infrastructures to help in tracking. These systems have the disadvantage of only controlling the position where they are installing. If the monitored person goes beyond the monitored area, the tracking system turns useless. The localization must be possible as far as possible, and currently there are already infrastructures that can be used to locate that cover a vast area with more or less accuracy. Also, installing new devices to help locating and cover a considerable area could be considered an unnecessary expense. Considering the system requirements previously presented, the concepts covered and the evaluation made to the state-of-the-art, this chapter presents the SecureTracking. The following presents an overview on the proposed system, presenting the case scenarios it applies, followed by a more detailed explanation on the role of each component. At the end of this chapter, the proposed solution is analysed against the requirements and the state-of-the-art.

The proposed system is suitable for scenarios where parents are concerned with kidnapping risks of their children, for example, in environments with much confusion where children may walk or travel without adult supervision. The Tutor devices are held by the parents, and with it they receive information about their children’s position, or alternatively, information of whether they are inside some secure area. Here we apply the concept of geofencing to check the presence in the defined areas. There can be multiple Tutor devices, and the Monitored device is able to report to them. A similar scenario is related with dementia patients or people with mental disabilities. In these cases, usually there is a responsible person, or more, assigned to them. The person with mental disability should carry the Monitored device, which reports to the Tutor device, held by the person(s) responsible for the monitoring task. In any case, the Monitored device must be capable of locating itself and communicate with the Tutor device. Both the Monitored and Tutor devices are mobile. Considering a scenario where the Tutor device may run out of battery or be temporarily unavailable for communications, the SecureTracking solution also assumes the
existence of an always-available repository that intermediates communications and stores messages to be used and checked out later if not properly delivered to the Tutor device. From now on, this component will be called the “repository server”.

### 3.1 Proposed Solution Description

This system assumes the existence of two major components that play the main roles of a tracking system: the Monitored device and the Tutor device. The Monitored device should be carried by the person whose position will be controlled. It must have built-in modules capable of obtaining its current location, processing capabilities to process and validate that information, and communication capabilities to send and receive information. On the other hand, the Tutor device should be carried by the person, or persons, that control the other’s position. From now on, this person is called the “responsible person”. He or she must be a person allowed to control the other’s position. Note that the expression “allowed to control the other’s position” means that the responsible person must have legal authority to perform such function. In any scenario, whatever it is, the SecureTracking system should not be used to illegally control a person’s position, as it is unethical and represents a clear violation of privacy.

The simplified architecture is presented in Figure 3.1. The ‘‘M’’ represents the Monitored side, and the “T” the Tutor side. The Monitored device gets its position using the mobile antennas, Wi-Fi APs or GPS. The communication is bidirectional between the Tutor and the Monitored devices. The figure also considers the presence of the repository server that acts as intermediate in the Tutor-Monitored devices communication, storing messages and sending them to the appropriate receivers.

![Figure 3.1: Proposed Solution - SecureTracking system overview](image)

For convenience and ease of usage, we assume that, in a first approach, both Monitored and Tutor
are designed to be implemented in smartphones (Android platform). The Tutor may also be available as an application for a personal computer. Using the smartphone capabilities and the existing support infrastructures for communication and localization, we can avoid unnecessary spending on creating and installing such support platforms. The communication channel is, mostly, the cellular network (using text-based SMS messages), given its low costs and high availability. Alternatively, if no signal is available, the communication may be performed via Internet (e.g., using available Wi-Fi connection).

### 3.2 Monitored Device

The main objective of the Monitored device is to be used to locate a certain person. This comprises the capabilities of obtaining its location, detect its presence in certain geofences and communicate the result of that process to another entity, in this case, the Tutor device. To do so, the monitored device must have a GSM Module, a Wi-Fi Module and an A-GPS Module. Being hard to implement, a dead reckoning method to navigate and estimate the position was discarded. The system assumes two major operating modes: “normal”, in which the device periodically obtains its location and processes it; and “rest”, where it is assumed that the monitored person is in a secure environment, and so, the monitored device does not get its location. For safety reasons, in the “rest” mode, the Monitored device still sends a periodic message to the Tutor to prove that the system is still operating and is not “dead”.

The position is obtained using the GSM, Wi-Fi and A-GPS Modules. Considering that, most of the time, knowing that a person is safe is enough, a lower accuracy is required to verify presence in certain areas. Also, using more accurate localization methods require more energy consumption. So, periodically, the device will turn on the Wi-Fi module (if not already on) so the device can detect nearby Wi-Fi APs to locate itself. To assure a decent accuracy, there is a defined threshold value that must be achieved. If the location accuracy does not meet this value, than the system activates the A-GPS localization. Regarding the A-GPS usage, if the device is indoors or under adverse environmental conditions, it may take too much time fixing on the GPS satellites, or even never be able to fix and get its location. Due to this possibility, there is a time-out event to avoid waiting forever for an A-GPS location. In these cases, and despite the bad accuracy, the location obtained with worse accuracy (obtained using GSM) is used, with a “bad accuracy” warning attached.

When a new position is detected, the system makes a simple comparison with the last position using the distance between them and the time since the last update. If the velocity is above a threshold, a message is sent to the tutor so he can investigate if it is a dangerous situation (a spoofing attack, a kidnap, or maybe just accuracy error). After that, the monitored device either processes the predefined geofences or just reports its position back to the tutor. Given the privacy constrains and ethical issues already addressed, when the geofence is made on the Monitored device, it does not report the precise location of the monitored person. Instead, the Monitored device only reports a message reporting the presence in the pre-defined geofences. Only if the the device locates itself outside a secure area it reports its position to help the responsible person to act.

Any communication between the monitored device and the tutor is ciphered and authenticated. The
keys to cipher and authenticate messages are exchange during initial setup, but can be regenerated at the tutor's request (process explained in Tutor's section). To assure freshness of the messages and avoid replay attacks, each message also contains a timestamp. To be valid, a message must have a timestamp greater than the last message received.

The monitored device also features the notion of “safe schedules”. These are time intervals defined by the tutor and sent to the monitored device in which it is, supposedly, in a safe environment. During this interval the device falls into "rest" mode. For example, during classes the tutor may assume a child is safe, constituting a safe schedule. This method helps save battery lifetime taking advantage of time intervals where there is no need for exhaustive operations.

### 3.3 Tutor Device

The tutor side aims at controlling the monitored person’s position and presence in certain areas. The tutor’s device communicates with the monitored one, sends requests or configuration messages, and receives location reports or warnings regarding geofences violation or position inconsistency.

As mentioned before, all communications between the monitored and tutor have to be confidential, fresh and authenticated. The keys for these security services are generated and exchange at first setup, but the tutor is responsible for regenerating them every pre-defined time interval in order to avoid or minimize attacks. To do so, the tutor application supports key regeneration. To send the keys to the monitored device, the tutor uses a pre-installed Key Encryption Key (KEK) dedicated to cipher the new key values.

The features provided include:

- Show monitored device’s position in a map.
- Create and edit geofences using the map to define coordinates and radius.
- Create and edit safe schedules.
- Keys regeneration.
- Position on-demand.
- Control messages:
  - Update period change.
  - New geofences.
  - New safe schedules.
  - Geofence in monitored/tutor.

The Tutor may also run as a personal computer application. This acts as a backup monitoring application or for more commodity to the responsible person. The application also holds the necessary keys to decipher and validate the received messages, being fully capable to be used as an alternative to the Tutor’s device application.
3.4 Repository Server

The repository server is a simple backup system to be used as a database for the monitored device's messages. The presence of this component helps keeping the system active if the Tutor’s device runs out of battery or is unable to communicate. The repository server does not hold any keys, so it cannot decipher nor validate received messages. Only the tutor’s applications may synchronize with the repository server, retrieve the messages and decipher and validate them. To perform its role, the repository needs to be able to communicate via text-based SMS or messages through the Internet, so it must have a GSM module and Internet connection.

3.5 Requirements Fulfilment

Taking in account the proposed requirements and the proposed SecureTracking system, this section details the proposed features to achieve the desired objectives. This system is considered a more complete solution, combining the tracking capabilities and some additional security measures. On a security point of view, the mapping between the requirements and the provided features is as follows:

- **Confidentiality** - Communication made between the Tutor and Monitored device is ciphered. The repository server does not have the capability of deciphering the messages, as they must be sent and received solely by the authorized persons and devices (Monitored and Tutor devices).

- **Integrity and Authenticity** - All messages carry a MAC to be used to check integrity and authenticate the sender.

- **Freshness** - All messages carry a timestamp that must be validated against the last received message(s).

- **Anti-spoofing** - The obtained position is checked against the last position to detect suspicious discrepancies. Also, the system analyses various parameters of the localization processes, namely GPS and Wi-Fi signals’ behaviour.

Regarding the ethics and privacy issues involved in tracking a human, this solution comprises a mode designed to minimize the impact on the monitored person perspective. Geofence may be performed on the Monitored or the Tutor device. When it is done in the Monitored side, it does not report the exact location when it is inside secure areas. The knowledge that the person is safe is enough to the responsible person.

3.6 Conclusion

Taking into account the related work that was analysed, none of the existing solutions can be considered complete in terms of tracking capabilities and security features to assure the location information, and consequently, the monitored subject safety. The solution proposed in this section, the SecureTracking...
system, addresses the security issues revealed for the localization processes and the communication channels. To do so, the communication has extra security mechanisms to ensure confidentiality, data integrity and authentication, and message freshness to avoid replay attacks. Also, regarding the localization process, the system considers anti-spoofing measures that helps to validate each location against the last obtained one to check for discrepancies.
Chapter 4

Implementation

This section presents the implementation of SecureTracking, the tracking solution proposed in Chapter 3 aimed at monitoring the location of children or people with mental disabilities. To build a prototype system, Monitored and Tutor devices were implemented on Android smartphones. As already stated, smartphones’ usage has grown in the past few years, and most of them have Wi-Fi and A-GPS modules, besides GSM and 3G communication. Also, the APIs for smartphone development, namely Android, are usually well documented and provide access to the device’s sensors, making them a viable solution. The Tutor’s PC application and the Repository Server were implemented in Java (for the application logic) and MySQL (for database management). The following sections describe the use of such platforms to implement the whole system and the choices made to meet the proposed requirements.

4.1 Mobile Components - Tutor and Monitored Devices

First of all, it is important to analyse the required behaviour for each component. The Monitored Device must obtain its position either periodically or on-demand. So, unless it receives a message from the tutor with instructions to perform some operation, this device should remain idle, regarding the SecureTracking context. Also, there is no user interaction with the Monitored device other than the initial setup or any configuration alteration ordered by a tutor. On the other hand, the Tutor’s device receives periodic information from the Monitored Device and performs user interaction, to check received information and to change the system configuration. Considering this, the Monitored and Tutor devices are implemented on Android smartphones to take advantage of the capabilities of these devices, as explained in the next sections.

4.1.1 Software Artifacts

Android applications can use up to four types of components: activities, services, content providers and broadcast receivers [6]. Activities represent screens (one-to-one relationship) to provide user interface, services are components that run in the background without user interface, content providers may be used to encapsulate data providing ways of defining its security and broadcast receivers are components
that respond to announcements that can be originated from the system itself or from applications. These announcements are represented and delivered as Intent objects, which can hold relevant information to be extracted and used when they are received.

The Monitored device’s application is mainly composed by a broadcast receiver that captures new messages from the Tutor and services that perform each task in the background without any user interaction. No activity is required for the Monitored device to properly perform its role, other than provide direct access to a configuration panel. The service responsible for periodically obtaining the location, uses the Android’s Alarm Manager to trigger an intent that calls the localization process. On the other hand, the Tutor’s application must have Android activities to allow user interaction (e.g., to check the received information or instruct the Monitored device with any command). Other than that, it must also have a broadcast receiver to listen to any message from the monitored device and services to process those messages even if the main activity is not running, so that every message is captured.

These modules were defined and implemented, each of them performing a different role: a Security Manager, that handles the cipher operations and authentication and integrity checking; a Localization Manager, responsible for the location requests and validation of that information in the Monitored device; a Communication Manager, used in every communications to the other components of the SecureTracking system; and the Geofence and SafeSchedule Managers, responsible for applying the geofences and the SafeSchedules, respectively. The main operations are based on the Android’s Alarm Manager and Broadcast Receiver, as explained below.

**Broadcast Receiver (Both devices)**

A broadcast receiver can be registered to a set of intents, operating as a filter. This is useful to turn the application into event-based instead of being constantly active and waiting for another task to finish. For example, requesting the device for location updates may take a while, especially if using A-GPS. So, whenever a component performs an operation that may take indefinite duration, intents are used to signal the starting or ending of those operations. The most important intents used in the designed prototypes are:

- **android.intent.action.DATA_SMS_RECEIVED**
  This is a system intent delivered when a data based SMS message is received. This filter is one of the most important in the SecureTracking system since it constitutes one of the communication channel. A data message is defined as a sequence of bytes sent using the short message service center to a destination address (phone number) to a specific port on the destination device. When this intent is received, the system deciphers and validates the message using the security mechanisms detailed in Section 4.4 and then processes the data in it.

- **com.google.android.c2dm.intent.RECEIVE**
  This is also a system intent and it is delivered when a message is received via the Google Cloud Service. This also represents one of the communication channels. Its usage and implementation are better explained in Section 4.3.
• **securetracking.locate**

These intent is broadcast periodically using the Alarm Manager at the monitored device. When detected by the broadcast receiver, it triggers the localization process. The usage of the Alarm Manager and the location management are explained below.

• **securetracking.newlocation**

When a new location is detected by the localization manager at the monitored device, it broadcasts this intent, adding information about the latitude, longitude and accuracy to be processed.

• **securetracking.scheduleevent**

The SafeSchedule events are also represented as intents fired using the Alarm Manager, and they hold information about the identification, start time, end time and type of the schedule. Upon receiving this intent, the system checks the type to determine if it is a start or end event. Being a start-event, the system falls into rest mode and an end schedule event is added to a pending map, using the identification as key and a ScheduleEvent as value. On the other hand if it is an end-event, the system checks if there is any schedule overlapping the one that just ended. It will only turn into normal operation if no other schedule is active at that time, checking if there is any pending end event. If none, then the system falls back to normal mode. This intent is used in the monitored device only.

**Services (Both devices)**

As already seen, the Monitored device normal operation does not interact with the user. Even the Tutor device may have some periods without user interaction. Despite this scenario, some support operations must take place even in the background. This is the reason why services are used in the implementation of both Monitored and Tutor devices. They run on the background, executing operations such as handling a new message or getting a new location information without requesting any user intervention. When the broadcast receiver detects any of the previously explained intents, it starts a service to process that request.

**Alarm Manager (Monitored device only)**

Since the period between two consecutive localization requests may be long, the Monitored device stays idle most of the time. Due to its lifecycle [5], an activity that is not focused (not showing in the screen) goes from “Resumed” to “Paused” and then “Stopped”. In the last stage, and despite being alive, an activity may be killed by the system if memory is needed elsewhere. A scenario where the Monitored application is shutdown because of memory needs is simply not acceptable, so in addition to having the broadcast receiver listening to new events, the Alarm Manager is used to trigger mandatory operations when needed. Alarms triggered by the Alarm Manager are capable of broadcasting an intent (detected, again, by the broadcast receiver) at set intervals even if the device itself is asleep. Being the case, and setting a “wakeup” flag, the alarm is capable of waking up the device’s CPU if its screen is off, ensuring
that the task is performed [9]. In this implementaion, the Alarm Manager is used to trigger the localization process and to trigger SafeSchedule-based events.

**Localization Manager (Monitored device only)**

The process of obtaining the current position runs in a dedicated service in the Monitored device application that initiates upon receiving a `securetracking.locate` intent, usually issued by the Alarm Manager at a predefined period. The SecureTracking localization strategy tries to minimize the average battery consumption and obtain the current location with acceptable accuracy. Since it first tries to obtain a location using the Wi-Fi APs, regardless of its state, the application saves the current state of the Wi-Fi, and if it not enabled, it turns it on. This allows the device for scanning nearby Wi-Fi APs to locate. After that, the actual localization process initiates. The Android API provides access to the system location services through the `LocationManager`. The `requestLocationUpdates` method is used using the `NETWORK` provider at first, which tries to obtain the current location using cell towers and Wi-Fi APs. To deal with coordinates and accuracy values, the Android definition and implementation of a “Location” is used. According to the Android API Reference [7], all locations obtained using the Android’s `LocationManager` include valid latitude, longitude, timestamp and accuracy information. The accuracy value of these locations is the radius of a circular area centered at that latitude and longitude, and, according to the the Android API Reference, there is a confidence of 68% that the real location is within that area. So, the smaller that accuracy value, the more precise is the information. Once the location is obtained, the system executes some tests to check acceptability of the accuracy and the truthfulness of that information. The accuracy must be below some threshold so the location information can be useful. For example, a location within a radius of 2 kilometres is not useful when trying to find a missing children. As shown if previews studies, locations obtained using the cell towers may reach an accuracy of several kilometres, so if such values are obtained it may be induced that no Wi-Fi APs could be used to locate. In this scenario, the A-GPS may be used as a location provider. Because the localization using GPS may take some time (e.g. bad weather conditions) or it may never succeed to locate (e.g. indoors), the system uses a timeout to prevent waiting indefinitely for location information. If the timeout triggers, the system reports the position obtained via `NETWORK` provider plus a bad-accuracy alert. The location report is made broadcasting a `securetracking.newlocation` intent holding the latitude, longitude and accuracy as extra data. This intent is then detected by the broadcast receiver, as already explained. To enable the Assisted-GPS and help decrease the TTFF, the Android system downloads the GPS almanacs and ephemeris, also known as XTRA Data. To force fresh assistance information, the system resets this aiding data and forces the download and injection of new one, including time re-synchronization.

**SafeSchedule Manager (Both devices)**

A SafeSchedule is characterized by an identification code, and the start and end times. Each time is simply formatted as hours and minutes using the `java.text.DateFormat` Java package, since they
are to be repeated every 24 hours. All SafeSchedules are privately and persistently stored. According to [8], files saved in the internal storage using private mode are not accessible by other applications, making this a viable solution to locally save critical information to be saved and loaded. The SafeSchedule Manager on the Tutor side enables its owner to visualize, create and edit them. When creating a new SafeSchedule, the user is prompt to enter a name, the starting and ending time (in the hours and minutes format). Although these SafeSchedules are created in the Tutor side, they must be sent to the Monitored device to be properly activated and used. On the Monitored device’s side, the SafeSchedules are loaded and dispatched by the SafeSchedule Manager. After saving them in the internal storage, the SafeSchedule Manager creates the start and end events for each SafeSchedule. These events are represented as pending intents scheduled using the Android’s Alarm Manager. The method `setRepeating` is used with one day as interval between consecutive executions. The pending intent is triggered at the specified time (start and end), and at that moment it broadcasts a ‘securetracking.scheduleevent’ intent containing information about the SafeSchedule it is related to. These intents are handled and the appropriate change in the functioning mode is made.

**Geofence Manager (Both devices)**

A SecureTracking geofence is a circular area characterized by an identification code, its center coordinates (latitude and longitude), its radius and the type, which can be secure or insecure. The type parameter allows the user to better adapt the system to different scenarios, allowing, for example, the definition of areas where the monitored device can be safe (secure geofences) or areas where it can not be for being unsafe or prohibited (insecure geofences). Like the SafeSchedules, the geofences are also saved locally in private mode in the internal storage, making them only accessible from this application. Geofences are present in the Monitored application to enable geofencing right in this device before further communication with the Tutor. This way, the Tutor may only be informed about the presence in secure/insecure areas. Again, in order to calculate distances and presence inside geofences, the geofence manager uses the Android’s notion of Location [7]. To determine the distance between two locations, the `distanceTo` method is used, which computes the approximate distance in metres between two locations using the World Geodetic System 1984 (WGS84) ellipsoid [7], which is the reference frame for the Earth used by GPS[15]. A location is inside a geofence if the distance between it and the center of the geofence is smaller than the latest radius.

**Google Maps Interface (Tutor devices only)**

Some operations are performed using the Google Maps API and interface. The Tutor application exhibits a frame with the map to help displaying some information, like the current geofences, the Monitored position and to allow easier managing of geofences.

The Google Maps frame shows some relevant information in this system context. For instance, every time the Tutor receives a message with the Monitored device’s position, it places a mark on the map showing its location. Also, the currently active geofences are also visible in the map. Secure and
insecure geofences are marked with difference colors.

4.2 Fixed Components

The two mobile devices involved (Monitored and Tutor devices) also communicate with two other components with different roles. To assist the communication between all components, there is a Repository Server; and as an alternative to the Tutor application in a smartphone, there is an implementation for a personal computer that replicates the same behaviour. The next sections explain in more detail the implementation of these two components.

4.2.1 Repository Server

The Repository Server is an always-on service, with communication modules (GSM and internet access) capable to communicate with both the Monitored and Tutor mobile devices and the Tutor computer application. As a prototype, the repository server was implemented using a smartphone with GSM, connected to a personal computer, and internet connection. Using the smartphone GSM communications it was possible to simulate a server with a GSM module, capable of receiving and sending SMS messages from and to the mobile devices involved. The internet connection allows communications via an alternative (or principal) channel. The communication model and the Internet messaging service are better explained in Section 4.3. Upon detecting a new message, the server stores it in a local database. The person responsible for monitoring the location may use both the smartphone and the computer application to retrieve the stored messages accessing the repository server providing the necessary credentials.

It is important to note that the repository server does not hold any secret keys, so it cannot decrypt any information by itself. The server never knows what information is in its databases. This has the security advantage that if the server security is breached, the attacker would still need the keys to decrypt the information in it.

4.2.2 Tutor PC Application

The Tutor PC application can be used as an alternative as the method of access to the SecureTracking features. Even if the Monitored device has plenty of battery and continues to report its location, the Tutor device may run out of battery, resulting that the person responsible for monitoring its position would be unable to play its role, the messages would be stuck in the repository server until the Tutor device is active again. This is the main reason for replicating the Tutor device’s behaviour and features to another platform.

As a prototype, the Tutor PC application was not fully implemented, but partially as a simple program written in Java capable of sending and receiving messages through the Internet. As future work a Graphical User Interface should be designed to show a map with the received locations. This application has access to the secret keys used to secure the communication channels, so it acts as one of the trusted
endpoints. Since this application is not always active and it can miss some messages in between, when it starts it must synchronize itself with the repository server to be up to date.

4.3 Communication Channels

The Monitored device needs to send its location to the Tutor devices (mobile and fixed) and the latter also need to send messages to the first, for example, asking its current location (location on-demand). Since the mobile devices have GSM modules, one obvious communication method is by sending and receiving SMS messages. According to [17], the process of manually selecting a contact and sending a text message of 55 characters showed little difference from idle mode in terms of average power consumption. The dominant power consumer is the screen. In the SecureTracking design, there is no user interaction when sending messages, so there is no need to turn the screen on. Tests conducted in [54] concluded that sending messages using 3G communication consumes more energy than using GSM. Also, GSM network is widely available, in contrast to Wi-Fi connections to the Internet that need often need adequate credentials. For this reason, the SMS messages are mostly used as a communication channel in SecureTracking.

Despite this, the device may be unable to access GSM network (for example, in basements) but may have an Internet connection that may be used to continue the normal operation of SecureTracking system. The Google Cloud Messaging (GCM) is a free service that allows to send messages from a server to Android devices, and currently supports bidirectional communication. It is easy to implement and is herein used as a prototype method for sending messages via an Internet connection. Messages from the server are temporarily stored in the Google’s GCM Servers, that act as intermediate in this communication, until the Android device is available, that is, connected to the Internet. The SecureTracking system adapts this messaging system by using the Repository Server to associate the Monitored devices to their Tutors, being an intermediate between them. Figure 4.1 shows the messages flow. One message from one Monitored device to its Tutor goes through the GCM Servers and it is stored in The Repository Server. From here the message is sent to the associated Tutor, which is retrieved from the databases. The message stays temporarily in the GCM Servers until the Tutor device is online and able to receive it. GCM allows sending messages with up to 4kb of payload, which is more than enough space to carry the necessary data.

4.4 Security

Despite the obvious focus in getting a location and communicating it to a person who is responsible for monitoring it, one of the main objectives of SecureTracking is to provide security mechanisms capable of maintaining some requirements not achieved by other monitoring systems. One security issue present is the vulnerabilities associated with the communication channels. Another security requirement is that the reported location should be analysed against possible spoofing attacks. The added mechanisms should use the resources available to secure the communication channels, determine if the location
obtained by the Monitored device could correspond to a true location (given the acceptable accuracy) or if it is possible that the device’s position is being spoofed. The next sections explain the implementation choices to fulfil these critical requirements.

### 4.4.1 Communication Security

As explained before, one of the defined requirements is to keep the communication channels secure, adding an extra layer of security mechanisms if necessary. Messages exchanged between the components of the SecureTracking system are sent via SMS messages or messages through the internet. There are two constraints that affect the decision on how to implement any security mechanisms representing an overhead on the original system. The first is the limited resources available, whether they are the processing capabilities or the limited energy available in the battery. The second is the limited size of a single SMS message and the fact that sending more messages than those strictly necessary may represent not only spending limited resources in the mobile device, but also monetary costs to its owner (usually monetary costs are associated to the sent messages). Therefore, the system design requires messages to be as small as possible to minimize the computational and monetary impacts. There are four concerns at this point: confidentiality, integrity, authentication and freshness of each message. For each case, the mechanisms used, the algorithm strength and cryptographic key sizes must be taken into account.

The components with an active role in the communications are the Monitored device, the Tutor device application and the Tutor PC application (the repository server does not have access to any secret key, so it is passive in this matter). All these prototypes are able to run Java applications. Being so, the algorithms, keys and all processes involved in generating, ciphering, deciphering, authentication and integrity mechanisms, and validation are implementations from Java API.
Confidentiality

To ensure confidentiality, end-to-end security must be achieved, so every exchanged message must be ciphered by the sender and deciphered by the receiver. Ciphering algorithms can use symmetric keys, meaning that only one secret key is used for both cipher and decipher operations, or they can use asymmetric keys, meaning that two different, although related, keys are used for cipher and decipher operations. Generally, using asymmetric keys, one of them is public and is used to cipher and the other one is kept private and is used to decipher. This means that if the system requires bidirectional secured communications, it must handle a different pair of keys for each endpoint because each player needs its own private and public keys. Also, the data block is bigger and the computation is higher. On the other hand, using a single secret key for both cipher and decipher operations is easier to manage but has the problem of securely distributing that key.

In [2] the authors study an approach on ciphering SMS text-based messages using asymmetric key algorithms. The conducted tests demonstrate that the resulted ciphered data usually gets bigger than the original message, thus resulting in more messages sent and more resources used. Also, the bigger the key size, the bigger the resulting ciphered message gets. The authors of SMSSec [38] proposed an end-to-end security system for SMS having both symmetric and asymmetric key algorithms in ciphering and message authentication codes. Regarding the symmetric algorithms, the authors opted for using the Advanced Encryption Standard (AES) for its speed, efficiency and for being considered a standard. The counter mode (CTR) was chose for being easy to implement and not introducing padding nor collisions. To authenticate messages, the HMAC was used having SHA256 as hash function, and not SHA1 due to known collision attacks.

Algorithms strength may be compared by the amount of work it takes to “break” them or determine the keys using approximately the same resources [12]. For example, a symmetric key algorithm like the AES with key size of 128 bits has approximately the same strength of the RSA algorithm with key size of 3072 bits, which is an asymmetric key algorithm. The key size difference is huge, and sometimes using larger sizes than actually required may negatively affect system performance, since larger keys take longer to be generated and to process data. On the other hand, too small size keys may not meet current security requirements, so a balance must be achieved. Since different algorithms may be used in a single system, it is important to remind that the overall protection strength is generally defined by the weakest algorithm and key size used. As pointed at [12], currently, and in the next 15 years, having a security strength of 128 bits is considered acceptable.

Many algorithms could be used and some analysis must be done to decide which achieves the better balance between the security requirements and the resources constraints. Twofish, the successor of the cipher algorithm Blowfish, is analysed together with AES in [57] regarding their performance, and the tests concluded that AES is faster than Twofish when ciphering text and images. The Data Encryption Standard (DES) is faster than AES [47], but it is considered insecure [10] and recommended not to be used [34]. Also in [47], the Triple-DES (3DES) algorithm is considered the slowest among DES, AES and Blowfish. Finally, the AES algorithm shows good performance not consuming much time to cipher/decipher [62]. Given the constrains and the previews analysis, SecureTracking uses the AES
(Rijndael) algorithm to cipher messages with a key of 128 bits.

Regarding the cipher mode, the main aspect to be considered is that the resulting ciphered data must be smaller than the defined size of a SMS message. This means that cipher modes that require the input to be a sequence of complete blocks of certain size are discouraged. Not being the case, usually what happens is that an incomplete block is padded with bits to match the required size using a padding algorithm. This would not be a limitation if the SMS message size was a multiple of a typical block size, which is not the case as shown in Section 4.4.1 - Message Format. Considering this, the Electronic Code Book (ECB), the Cipher Block Chaining (CBC) and Cipher Feedback (CFB) modes are rejected as they required padding. The Cipher Text Stealing (CTS) mode is a variant of CBC that allows arbitrary input sizes. If the last block is a “partial block”, that is, the bit size is smaller than the block size and padding bits are needed, they are taken (stolen) from the second to last block. Another possible mode is the Counter (CTR) that uses a counter that is incremented in each cipher operation.

Having as candidates the Counter (CTR) and the Cipher Text Stealing (CTS) modes, the authors of [58] point that despite that the CTR mode natively allows the input to be of arbitrary size, the CTS mode is a classical, standardized and elegant way of extending the CBC mode, which is widely used. The only limitation regarding the CTS mode is that it requires a minimum input size the same as the Initialization Vector. As shown in Section 4.4.1 - Message Format, the size of the message always fulfills this requirement. Having this in consideration, the CTS is the chosen cipher mode in SecureTracking.

The Java API already provides implemented cryptographic mechanisms via the package 'javax.crypto' [52]. So, the chosen cipher parameters are ‘AES/CTS/NoPadding’ [51], that is, Advanced Encryption Standard with secret key of 128 bits with Cipher Text Stealing mode and no padding (which is not required).

Authentication/Integrity

To provide authentication, some mechanism must be used to make sure that the message was originated from a well-known source. In the SecureTracking context this means ensuring that, for example, no one other than the Monitored device sends location information to the Tutor mobile device or PC application. Ultimately, if the system detects a message from an unknown source, it must ignore it. This requirement could be achieved using digital signatures based on Public-Private keys signing algorithms but, again, this would mean handling a pair of keys per each player in the communications instead of a single shared key. Also, comparing with other authentication methods, the key sizes are much bigger when using public-private cryptography, just like the computation overhead. Another way if providing authentication and also ensuring integrity is by using a Message Authentication Code (MAC), as defined in [48]. The Cipher Block Chaining Message Authentication Code (CBC-MAC) [33] ciphers a message in CBC mode but the result is just the last computed block, which then is the MAC. There are two differences between the CBC-MAC and the standard CBC encryption. The first is that, in the former, no Initialization Vector is (or can be) used. In other words, \( IV = 0^n \). The second is that the CBC encryption outputs every intermediate ciphered block and the CBC-MAC only outputs the last computed block. A MAC based on a cryptographic hash function is known as Hash-based Message Authentication Code
(HMAC). Given an input text, the HMAC algorithm uses a pre-shared secret key and a cryptographic hash function to compute a final MAC [20, 66]. The hash function must be carefully chosen because it can compromise the effectiveness of the HMAC algorithm. According to [20], the SHA-1 hashing function provides less than 80 bits of security strength and shall not be used in digital signatures after the end of 2013. Conducted tests, described in Section 5, compared the computation time between HMAC using SHA-1 and SHA-256 as hash functions and CBC-MAC. The results show that CBC-MAC is slower.

Considering the previews analysis and the fact that the size of the resulting MAC matters, as it influences the size of the sending message itself, the choice in SecureTracking was to use HMAC with SHA-256 as hashing function. Just as the implementation of the confidentiality mechanisms, the ‘javax.crypto’ package already provides HMAC functionalities with various hashing functions, so the parameters used are ‘HmacSHA256’ [51].

**Freshness**

Despite being ciphered and authenticated, exchanged messages are susceptible of replay attacks. In such situations, an attacker may intercept a message and retransmit it in another time. The message is already ciphered, has the original authentication made at the source and integrity is not violated, so retransmitting it will deceive the receiver that would validate the message and introduce it in the system as being legit. To prevent this kind of attacks, the freshness of the messages is guaranteed using timestamps. It is assumed that the devices have pre-synchronized clocks, configuration that is achieved by synchronizing with Network Time Protocol (NTP) servers. Upon receiving a message, the receptor extracts the timestamp in it and checks its validity by comparing with the latest timestamp received. To validate the just received message must have a timestamp grater than the lastest received. If not, the message is discarded. The implementation of this simple freshness system is made using the ‘java.util.Date’ Java package to obtain the current time in milliseconds, which is then appended to the message.

**Applying Security Measures**

There are three ways of joining cipher and authentication/integrity: one is to cipher the plaintext and then append a MAC of the original plaintext (Encrypt-and-MAC plaintext); another is to append a MAC of the original plaintext to it and then cipher (MAC-then-encrypt); and cipher the plaintext and then compute a MAC of the already ciphered data, appending that MAC to it (Encrypt-then-MAC). In [13] the authors managed to prove that the Encrypt-then-MAC method is the better, being the single safe against chosen-plaintext and chosen-ciphertext attacks. For this reason, this is the method used in SecureTracking to combine cipher and authentication/integrity mechanisms. Regarding the freshness mechanism, the timestamp is firstly appended to the plaintext, and then goes the cipher of them together, and finally the MAC of the ciphered data is computed and appended to it. To successfully handle a message, the receiver must first validate the MAC, then decipher the message and finally validate the timestamp ensuring the freshness of the message.
Message Format

To define the message format, the best approach is to have as reference the communication channel with more constraints, and that is the SMS message. Also, the priority goes to security, so the number of fields and their size are defined after deciding the security measures. The security mechanisms chosen in the previous sections define that the cipher mode does not alter the size of the original message to be sent (CTS cipher mode), the timestamp is a value in milliseconds, represented in Java as a field of 8 bytes, and the HMAC value is computed using SHA-256 which outputs a MAC of 32 bytes. In sum, the security overhead is of 40 bytes. Tests made showed that, when using the Android API for sending SMS messages, namely sending SMS as an array of bytes (sendDataMessage method from [4]) instead for text-based SMS messages, they have 133 bytes that can be used. So, reserving 40 bytes to security, the total available space for operational data is 93 bytes.

The SecureTracking defines a set of required messages, listed in Figure 4.2, each of them with different fields in it. This means that messages of the same “type” have the same fields and the same size, but different “types” of messages have different fields and sizes. By manually defining each field size in bytes and not using regular text with a separator character, we get more compressed messages since only the strictly necessary space is used.

<table>
<thead>
<tr>
<th>Position Report</th>
<th>OPCODE (1)</th>
<th>VALID (1)</th>
<th>LATITUDE (8)</th>
<th>LONGITUDE (8)</th>
<th>ACCURACY (4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Position On-Demand</td>
<td>OPCODE (1)</td>
<td>VALUE (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process Geofences in Monitored dev</td>
<td>OPCODE (1)</td>
<td>VALUE (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alert</td>
<td>OPCODE (1)</td>
<td>ALERT_CODE (1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New update interval</td>
<td>OPCODE (1)</td>
<td>VALUE (4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>New Geofences / SafeSchedules</td>
<td>OPCODE (1)</td>
<td>MSG_ID (4)</td>
<td>MSG_NMBR (1)</td>
<td>TOTAL_MSGS (1)</td>
<td>DATA (…)</td>
</tr>
</tbody>
</table>

Figure 4.2: Messages types and their fields (values are size of field in bytes)

All messages are identified with an OPCODE, a code that identifies the type of the messages and determines how the rest of the fields must be processed. Upon receiving a message with OPCODE of a Position Report, the Tutor device knows that the next field has 1 byte, which corresponds to a validation code meaning that this location is considered valid or may have been spoofed; and the next 16 bytes correspond to the latitude and longitude values (8 bytes each), followed by 4 bytes corresponding to the accuracy.
**Keys Management**

Cipher and authentication/integrity check operations require the establishment of secret cryptographic keys between the players of the communication. Since the chosen cipher algorithm uses a single key, it is a good choice to use different keys to cipher and to compute the HMAC, making the security system a little more robust.

On the other hand, it is recommended that keys are replaced after being used a certain number of times or after a time interval. SecureTracking implements a mechanism that regenerates both cipher and HMAC secret keys every 24 hours and distributes them among the communication end entities, creating a notion if a session. To distribute these keys, a KEK is used, which is a pre-shared secret key only used for this purpose. The KEK is used to cipher the message that carries the new keys as shown in Figure 4.3. This feature avoids overuse of the same keys for a long period, reducing chances of cryptanalysis. Since the frequency of messages is not very high during a normal operation (assuming periodic updates every hour or 30 minutes, for example), the keys are not overly used, and if the frequency of messages rises, the session period could be reduced.

![Renew keys sequence diagram](image)

**Figure 4.3: Renew keys sequence diagram**

### 4.4.2 Localization Security

The proposed solution assumes the existence of mechanisms to prevent spoofing attacks on the localization process. Unfortunately, the used platform does no provide much mechanisms to do so, nor the current Android devices have sensors capable of analysing the signals from GPS satellites. The anti-spoofing mechanisms could be implemented by analysing the GPS satellites signals and detecting unusual or abrupt changes in their strength. The Android API only provides access to the Signal-to-Noise-Ratio (SNR) measurement, which is the ratio of signal power to the noise power. If a satellite simulator is in place, even if it broadcasts a fake satellite signal that should have signal strength considerably greater, so easily detectable, it may also broadcast noise signals to balance the SNR. So, using only the SNR values, detecting spoofing attacks on GPS localization is much difficult or even ineffective.
Despite that, simple spoofing attacks may cause abrupt changes in the location itself. To help detecting these situations, SecureTracking implements a mechanism to validate the distance between two consecutive locations given the elapsed time. By calculating the velocity, and assuming a person is travelling by foot, the system can detect abrupt changes on the location using a threshold value suited to human walking speed. Additional improvement could detect the transportation mode and dynamically change that threshold value.

4.5 Implementation Discussion

Using Android smartphones to implement the Monitored and Tutor applications was a good choice taking in account that the Android API provides access to a lot of features. Access to the localization sensors, the private storage, the use of Java environment and access to many Java APIs facilitates the implementation of such system. Smartphones are widely used and almost every one have the necessary capabilities to fulfill the requirements of SecureTracking. The prove is that the mobile applications were fully implemented and are capable to operate normally. Despite that, it would be interesting to have implemented a more sophisticated anti-spoofing system capable of analysing the signals from GPS satellites. Currently, the Android system does not provide access to such measurements. Also, for future work it is necessary to compare the effort and feasibility of implementing this system using smartphone applications versus implementing a dedicated system that only runs the necessary modules and software artifacts instead of having an operating system with another applications running and sharing resources.

The use of the GCM as a mean to communicate over the Internet was a prototype option, but an alternative option would be to avoid communication with third-party servers and only have the Repository Server as intermediate. Further analysis as future work is crucial to compare solutions on the messaging system over the Internet.
Chapter 5

Characterization and Evaluation

Taking into account the prototype implementation, the requirements established for the proposed solution and the state-of-the-art analysed in the Section 2.2, it is important to characterize and evaluate the proposed SecureTracking system. The evaluation starts by considering the security measures, which introduce some overhead other than simply sending a SMS text message. The impact of the cipher operation, as well as the computation of the HMAC must be calculated regarding the time spent and the battery loss. Also, since there are three methods, provided in the implemented platform, of getting the current location, each one of them with different properties, they need to be characterized in order to evaluate the best localization strategy. For each feature, the different configurations were tested for typical conditions. To retrieve information about time and battery levels, a dedicated Profiler was implemented. For each operation, the time taken to execute it was calculated in nanoseconds using Java’s `System.nanoTime()` method. Since the Android system only provides battery level in integer values between 0 and 100 (like a percentage), the impact of SecureTracking can only be seen after prolonged activity and not at each operation level. The collected data was saved in log files in the device’s storage. The description, goals and results of each test are described in the following sections. The end of this chapter has a discussion on the obtained results.

The main target of evaluation is the Monitored device since it is the component with the most demanding requirements. The tests were conducted by installing the developed application in an Android smartphone “LG Optimus Pro C660”, running Android OS 2.3.3 with a 800 MHz CPU and 256 MB of RAM. The set of conducted tests is as follows:

- **Accuracy Tests**
  Metrics: accuracy in metres given by the Android API; accuracy in metres calculated manually
  - Network, Wi-Fi and A-GPS along a short path in residential area.
  - Network, Wi-Fi and A-GPS along a short path in industrial area.
  - Network, Wi-Fi and A-GPS along a long path in mixed area.

- **Time Tests**
  Metrics: elapsed time in nanoseconds
• Time taken by Network, Wi-Fi and A-GPS to obtain location
• Evaluation of MAC computation
• Evaluation of the cipher algorithm

• Battery Tests
  Metrics: battery level in percentage
  - MAC Computation.
    • HMAC with SHA-256
    • CBC-MAC of 16 bytes
  - Localization methods
    • Network
    • Wi-Fi
    • A-GPS

5.1 Accuracy Characterization

Since there are different localization methods that can be used in the implemented platform, the accuracy of each of them must be evaluated and validated. The Android API details that every location obtained through it includes a field that reflects an accuracy value with 68% of confidence. This accuracy notion assumes that errors are random with a normal distribution, so the standard deviation is represented by that 68% confidence circle, centered in those coordinates and with a radius equal to the accuracy value [7].

To evaluate the accuracy given by the Android system, a path was predefined and monitored by the Monitored device, set to report its location with an interval of 5 minutes. The localization methods were tested in separate, using Network (using Wi-Fi) and A-GPS. The path is 4km long, characterized by both residential and non-residential areas. The obtained results suggest that the system was able to locate itself near the real path. Figures 5.1 and 5.2 show examples of the accuracy difference between locating with A-GPS and Network. More tests are available in Appendix A.

From the obtained results depicted on maps it can be observed that the accuracy value given by the Android system is not always correct. These tests were made using a time interval to simulate a typical scenario where the Monitored device periodically reports its location. Due to differences in walking speed during the various tests, the location was not always obtained in the same spots. So, another set of tests was conducted, this time establishing well defined locations called fixed points, and instructing the device to locate on-demand using each of the localization methods. Two different type of areas were considered: one residential (Path A) and another one more industrialized (Path B). The accuracy given by the Android system (system accuracy) was then compared with the accuracy manually measured with the assistance of Google Maps to get the coordinates of each fixed point and the Haversine formula, implemented in Javascript [68], to compute the distance from them to those obtained by the Android...
system. The Haversine formula calculates the shortest distance between two points over the surface of the Earth. All confidence intervals were calculated for a confidence of 90%.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Distance (m)</th>
<th>Standard Deviation</th>
<th>Variance</th>
<th>Coefficient of Variation</th>
<th>Confidence Interval (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-GPS</td>
<td>21.25</td>
<td>21.43</td>
<td>459.12</td>
<td>1.01</td>
<td>10.17</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>24.83</td>
<td>25.48</td>
<td>649.22</td>
<td>1.03</td>
<td>12.10</td>
</tr>
<tr>
<td>Network-only</td>
<td>128.67</td>
<td>102.82</td>
<td>10570.97</td>
<td>0.80</td>
<td>48.82</td>
</tr>
</tbody>
</table>

Table 5.1: Path A - statistics on the distance between reported and fixed locations

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Distance (m)</th>
<th>Standard Deviation</th>
<th>Variance</th>
<th>Coefficient of Variation</th>
<th>Confidence Interval (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-GPS</td>
<td>53.06</td>
<td>39.44</td>
<td>1555.22</td>
<td>0.74</td>
<td>18.73</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>42.17</td>
<td>25.75</td>
<td>662.88</td>
<td>0.61</td>
<td>12.23</td>
</tr>
<tr>
<td>Network-only</td>
<td>171.25</td>
<td>88.06</td>
<td>7754.02</td>
<td>0.51</td>
<td>41.81</td>
</tr>
</tbody>
</table>

Table 5.2: Path B - statistics on the distance between reported and fixed locations

As it can be seen in Tables 5.1 and 5.2, the Network-only method has the worse accuracy obtained either manually or using the Android system, in both paths. On average, the measured accuracy using Network had an error of $128 \pm 48$ and $171 \pm 41$ metres relatively to the real location in paths A and B, respectively. The Wi-Fi method registered almost half of accuracy error in path B ($42 \pm 12$ metres) than in path A ($24 \pm 12$ metres), but the confidence interval was basically the same. Regarding the GPS measured accuracy, some abnormal values were registered in path B, resulting the very bad accuracy in specific areas. In particular, two fixed locations were partially under a dome, which limited the line-of-sight with clear sky. For this reason, path B registered a worst average accuracy error, $53 \pm 18$ against
21±10 registered in path A. The difference between the measured and system accuracy is detailed in Table 5.3 and 5.4. On average, for the Network-only method, an error of about 639±67 metres in path A was measured in comparison with 674±44 metres for path B, revealing that the estimation made by the system was much worse than the real accuracy. For the Wi-Fi method, path A registered greater difference than B. In that case, the system accuracy was worse than the real one. The A-GPS method registered greater difference in path B than A, 36±15 metres against 19±8 metres, noting again the interference of the dome in its accuracy. Regarding the coefficient of variation, we can conclude that the network-only method has less dispersed values, noting that unlike the other methods, this method is less influenced by the environment. The Wi-Fi localization accuracy depends on the presence and density of APs and the A-GPS method depends on weather conditions and clear line-of-sight with the sky.

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Difference (m)</th>
<th>Standard Deviation</th>
<th>Variance</th>
<th>Coefficient of Variation</th>
<th>Confidence Interval (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-GPS</td>
<td>19.41</td>
<td>17.69</td>
<td>312.84</td>
<td>0.91</td>
<td>8.40</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>118.34</td>
<td>74.87</td>
<td>5605.16</td>
<td>0.63</td>
<td>35.55</td>
</tr>
<tr>
<td>Network-only</td>
<td>639.00</td>
<td>141.88</td>
<td>20130.00</td>
<td>0.22</td>
<td>67.37</td>
</tr>
</tbody>
</table>

Table 5.3: Path A - statistics on the difference between real and system accuracy

<table>
<thead>
<tr>
<th>Method</th>
<th>Average Difference (m)</th>
<th>Standard Deviation</th>
<th>Variance</th>
<th>Coefficient of Variation</th>
<th>Confidence Interval (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-GPS</td>
<td>36.10</td>
<td>33.30</td>
<td>1108.80</td>
<td>0.92</td>
<td>15.81</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>40.08</td>
<td>55.30</td>
<td>3057.90</td>
<td>1.38</td>
<td>26.26</td>
</tr>
<tr>
<td>Network-only</td>
<td>674.08</td>
<td>93.12</td>
<td>8671.17</td>
<td>0.14</td>
<td>44.22</td>
</tr>
</tbody>
</table>

Table 5.4: Path B - statistics on the difference between real and system accuracy

5.2 Time Measurements

SecureTracking introduces some new operations associated with security that delay the transmission of information to the other entities, namely the tutor device or the repository server. These operations are (1) obtaining the current location (in the Monitored device only), (2) ciphering the message to be sent and then (3) computing the MAC value. For the first operation there are three different methods of execution (Network only, Wi-Fi and A-GPS), but all of them can be used and the time they take varies significantly depending on the circumstances. Although time matters, the main factors on the decision of the localization method are the energy consumption and their availability. To evaluate the possible candidates for the message authentication, tests were performed for the MAC algorithms, namely: CBC-MAC of 8 bytes (CBC-MAC8), CBC-MAC of 16 bytes (CBC-MAC16), HMAC with SHA-1 (HMAC-SHA1) and HMAC with SHA-256 (HMAC-SHA256). The CBC-MAC methods were deployed using the BouncyCas-
tle packages org.bouncycastle.crypto.macs [50], which produces, by default and according to the
API reference, a MAC value of half the size of the block size of the cipher. The cipher algorithm used
as base to produce the MAC was the AES, which has a block size of 128 bits, or 16 bytes. Therefore,
by default, the CBC-MAC algorithm using AES produces a MAC value of 8 bytes. Despite that, the API
allows the generation of MACs with size multiple of 8 bytes.

5.2.1 Localization Methods

It is important to measure the time required to obtain the current location using each of the localization
methods available in this system. It is important to note that both the Network-only and Wi-Fi methods
require a database that maps each cell antenna and Wi-Fi AP to its real location in order to use that
information when estimating the location of a device near them. On the other hand, A-GPS must fix
in satellites to calculate its location. Table 5.5 shows the statistics of each localization method. It is
important to note that if the A-GPS does not have the assisting data (almanacs and ephemeries), the
TTFF will be greater, and so will be the time to locate. For this reason, the assisting data was pre-
downloaded before each test. A scenario without pre-downloaded data was not considered as it would
require bigger time interval between each location request to provide equal conditions to each test.

<table>
<thead>
<tr>
<th>Localization Method</th>
<th>Average (ms)</th>
<th>Standard Deviation</th>
<th>Variance</th>
<th>Coefficient of Variation</th>
<th>Confidence Interval (ms)</th>
<th>Max (ms)</th>
<th>Min (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network-only</td>
<td>111.31</td>
<td>74.12</td>
<td>5493.11</td>
<td>0.67</td>
<td>5.45</td>
<td>942.44</td>
<td>55.68</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>5183.22</td>
<td>1920.32</td>
<td>3687631.29</td>
<td>0.37</td>
<td>141.26</td>
<td>23489.98</td>
<td>254.03</td>
</tr>
<tr>
<td>A-GPS</td>
<td>10934.24</td>
<td>6511.63</td>
<td>4240137.28</td>
<td>0.60</td>
<td>479.00</td>
<td>36711.63</td>
<td>1783.06</td>
</tr>
</tbody>
</table>

Table 5.5: Time statistics for each localization method

From the results it can be concluded that the A-GPS takes much more time, on average, than the
other two methods, about 10934.24 ms ± 4%. In average, Wi-Fi localization took about half the time,
5183 ms ± 3%. The fastest was the Network-only method, taking, on average, just 111.31 ms ± 5% to
obtain the current location. The average statistics have a confidence of 90%.

5.2.2 MAC Algorithms

The goal of these tests is to evaluate the computation time for the MAC generation. To perform these
tests the localization methods were disabled and another method was implemented to randomly gen-
erate dummy values for the latitude and longitude. This way, the test generates random values for the
coordinates, builds the message, generates the MAC value and ciphers it. Since this system was imple-
mented as an Android application, there is not much control on the behaviour of the thread scheduler
and manager. This impacts the time taken by some operations, which could sometimes become much
higher. The statistical analysis takes this into account by considering only a slice of the results for each
operation, in ascending order of time taken. First we calculated all these metrics for all the obtained
results. The variance for these values was excessively high, given many situations where the time was
heavily influenced by the operating system thread scheduler and manager. We then applied the metrics to 80% of the results (in ascending order) to exclude most of these cases where the thread scheduler yielded the SecureTracking thread.

<table>
<thead>
<tr>
<th>MAC Algorithm</th>
<th>Average (ms)</th>
<th>Standard Deviation</th>
<th>Variance</th>
<th>Coefficient of Variation</th>
<th>Confidence Interval (ms)</th>
<th>Minimum (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBC-MAC8</td>
<td>8.40</td>
<td>7.74</td>
<td>59.83</td>
<td>0.92</td>
<td>0.14</td>
<td>2.93</td>
</tr>
<tr>
<td>CBC-MAC16</td>
<td>8.28</td>
<td>7.29</td>
<td>53.11</td>
<td>0.88</td>
<td>0.13</td>
<td>2.80</td>
</tr>
<tr>
<td>HMAC-SHA1</td>
<td>2.29</td>
<td>1.39</td>
<td>1.94</td>
<td>0.61</td>
<td>0.03</td>
<td>1.27</td>
</tr>
<tr>
<td>HMAC-SHA256</td>
<td>3.67</td>
<td>3.91</td>
<td>15.25</td>
<td>1.07</td>
<td>0.07</td>
<td>1.48</td>
</tr>
</tbody>
</table>

Table 5.6: MAC Algorithms - time statistics using the first 80% of collected data (in ascending order)

The tests were divided in 10 samples of 1000 tests each, totalling 10 000 tests. Taking a look at Table 5.6, the CBC-MAC8 and CBC-MAC16 took approximately the same time, 8.40 ± 0.14 ms and 8.28 ± 0.13 respectively, the HMAC-SHA1 took 2.29 ± 0.03 ms and the HMAC-SHA256 3.67 ± 0.07 ms. All values were calculated with a confidence of 90%. We can conclude that, in descending order, the fastest algorithms are the HMAC-SHA1, HMAC-SHA256, CBC-MAC8 and CBC-MAC16. If we assume that the minimum time value observed is the fastest that each algorithm can execute, again we get that HMAC-SHA1 is the fastest, taking about 1.26 ms, followed by the HMAC-SHA256 with 1.48 ms, and then with similar values, by CBC-MACs. To have another perspective and to confirm the energy consumption, the battery level was recorded during a test with HMAC-SHA256 and another with CBC-MAC16. The results of this test, among others, are presented in the next section.

5.2.3 Cipher Algorithm

Ciphering data also introduces time overhead that must be characterized. As explained before, to measure the time required to generate the MAC value, a set of 10 samples of 1000 tests was conducted for each MAC algorithm. Each of these tests includes measuring the time required to cipher the generated message. Similarly to the MAC measurements, only a slice of 80% of the tests were considered, in ascending order of time taken, to exclude tests influenced by the operating system thread manager. As detailed in Section 4.4, the chosen cipher algorithm is the AES using a key of 128 bits in CTS mode. Taking into account the tests made on HMAC-SHA256, the cipher characterization is as depicted in Table 5.7. On average, the AES algorithm took 1.23 ± 0.0021 ms to cipher the messages, with a confidence of 90%.

<table>
<thead>
<tr>
<th>Average (ms)</th>
<th>Standard Deviation</th>
<th>Variance</th>
<th>Coefficient of Variation</th>
<th>Confidence Interval (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.23</td>
<td>0.11</td>
<td>0.01</td>
<td>0.09</td>
<td>0.0021</td>
</tr>
</tbody>
</table>

Table 5.7: AES using 128 bits key in CTS mode - time statistics using the first 80% of collected data (in ascending order)
5.3 Battery Consumption

The periodical process of the localization methods, ciphering or deciphering messages, or computing MAC values has impact on the battery usage. Being a limited and crucial resource, the main and most frequent operation were tested to check the impact on the battery lifetime. To measure the battery level, the Android API was used, namely by registering to the intent android.intent.action.BATTERY_CHANGED. This intent is sent by the system and contains various informations about the battery, including the level and charging state.

5.3.1 MAC Algorithms

As stated in the previews section, two MAC algorithms were tested in order to verify the true impact in the battery: CBC-MAC16 and HMAC-SHA256. The objective was to correlate the time that each algorithm takes to output a MAC value with the energy consumed. Each algorithm was used to authenticate the same amount of messages while recording the battery level. The obtained results depicted in Table 5.8 show that the CBC-MAC16 algorithm consumes more battery than the HMAC-SHA256. On average, CBC-MAC16 consumed more battery than HMAC-SHA256, 4.80±0.97% against 2.11±0.61% on the later, with a confidence of 90% and a sample of 10 sets of 100 executions each. Each of the executions involved generating random values for the coordinates, generating a message, generating the MAC value and ciphering the whole message, just like it would happen in a real scenario.

<table>
<thead>
<tr>
<th>MAC Algorithm</th>
<th>Average decrease (%)</th>
<th>Standard Deviation</th>
<th>Variance</th>
<th>Coefficient of Variation</th>
<th>Confidence Interval (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CBC-MAC16</td>
<td>4.80</td>
<td>1.87</td>
<td>3.51</td>
<td>0.39</td>
<td>0.97</td>
</tr>
<tr>
<td>HMAC-SHA256</td>
<td>2.11</td>
<td>1.17</td>
<td>1.36</td>
<td>0.55</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Table 5.8: Battery loss statistics using HMAC-SHA256 and CBC-MAC16

5.3.2 Localization Methods

The Wi-Fi localization involves turning the Wi-Fi chipset on (if not active already), keeping it active for some time and scanning for APs. Also, the GPS chipset is known to be a heavy energy consumer (as the state-of-the-art section explains). The Network-only localization requires 3G connection and availability to contact with a remote server with the database of the location of cell antennas. An alternative would be having a local database in the device, but it would not be viable because available databases are very incomplete. Fortunately, Android provides access to Google's databases, which are much more complete. The disadvantages are that it requires access to the internet, hence the use of 3G communication which may use more energy, and the process is like a “black box”, that is, the request is made and it responds with the location. It was not possible to uncover how is the database organized, the extent to which it is complete, what information does it hold or use and how the location is actually calculated.
To have a proper vision on the impact of the localization process on the battery lifetime, each localization method was executed the same number of times and with the same interval between consecutive requests. During the tests, the battery level was recorded. Some problems arose when implementing and executing these tests. It is very important that the testing conditions are as similar as possible. For example, the period of the requests needs to be sufficiently big so that each localization method can successfully get a location in it. If not, it could happen that a location request takes too much time, and in the meanwhile another request is made. In these situations, if one request successfully obtains a location, the other will report the same information because it is the most recent. The problem is that the time it takes to locate varies significantly, depending on the information used to locate, such as which antennas does it used, how many and which Wi-Fi APs have been scanned, how many GPS satellite can it fix to. This kind of behaviour is beyond our control. To properly define a useful period, some tests were made with increasing values of time between consecutive location requests, starting in just a few seconds. The value was established at 60 seconds, every localization method could finish successfully most of the times. Even so, in some situations, some test results were discarded because the localization took too much time or, probably due to implementation bugs, the test did not conclude with the expected number of requests. To better deal with these situations, the tests were divided in 10 samples of 50 tests each to easily keep track of the battery levels and to rapidly discard invalid tests. All tests that successfully ended as expected were accepted, no matter the results. A total of 500 tests were made, starting at no particular level of battery. For the Network-only tests, the 3G communication was left on to allow communication with internet by the application. The Wi-Fi tests were made using known Wi-Fi APs with Internet access to query Google servers. The Wi-Fi chipset was turned off before the test and turned on and off in each location request. The A-GPS were made having recent pre-downloaded assisting information about the almanacs and ephemeris (Xtras). If this was not made, the localization using GPS could take several minutes. All tests included ciphering and computing the MAC value of the message.

<table>
<thead>
<tr>
<th>Localization Method</th>
<th>Average (%)</th>
<th>Standard Deviation</th>
<th>Variance</th>
<th>Coefficient of Variation</th>
<th>Confidence Interval (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network-only</td>
<td>4.20</td>
<td>2.62</td>
<td>6.84</td>
<td>0.62</td>
<td>1.36</td>
</tr>
<tr>
<td>Wi-Fi</td>
<td>3.10</td>
<td>1.60</td>
<td>2.54</td>
<td>0.51</td>
<td>0.83</td>
</tr>
<tr>
<td>A-GPS</td>
<td>3.70</td>
<td>1.16</td>
<td>1.34</td>
<td>0.31</td>
<td>0.60</td>
</tr>
</tbody>
</table>

Table 5.9: Statistics on battery consumption per localization method

Table 5.9 depicts the obtained results of tests, which show that, surprisingly, the Network-only localization method consumed more energy than the other two, an average of $4.2\pm0.36\%$, against $3.10\pm0.83\%$ of Wi-Fi and $3.70\pm0.60\%$ of A-GPS. We associate this result to the fact that the 3G was used in every request to connect to the internet. It is actually known that this communication is in fact made because some tests were made with no internet access, resulting in failure to locate. The Wi-Fi localization method drained, on average, about 3.10% of the battery during the tests, and the A-GPS consumed about 3.70%. These results are too close, and this has an explanation. As already stated,
using Wi-Fi involves turning the chipset on, keep it active and scanning of APs. Also, having actual internet communication also drains battery. This favors the A-GPS method, where the GPS assisting data was downloaded before every test. Since the GPS module needs to fix at satellites, and that operation takes much time, not having the recent almanacs and ephemeris would result in much more time trying to fix at satellites, so much more time trying to locate, so much more battery consumed, as proved at the state-of-the-are analysis. The GPS usage is also influenced by its availability. Tests conducted indoors using A-GPS resulted in no fixed satellites at all and no reported location at all in lower floors of buildings (for example, in a second floor out of 3). However, locating with A-GPS in the last floor of a building was possible, mostly near windows (with some or better line of sigh).

5.3.3 PIN Alert Impact

One feature introduced by SecureTracking is to ask the owner of the Monitored device to enter a code periodically in order to acknowledge that the device is still, in fact, in the hands of its owner. Because this behaviour involves waking up the device and showing information in the screen, as opposition with a service running in the background, it has a greater impact on the battery consumption. The combination of having an alert signal (such as vibration or sound), showing an activity on the screen and turning the screen on so the person can confirm the code may result in some extra energy costs. To verify this impact, a test was conducted to compare two scenarios: (A) periodic location requests without PIN alert and (B) periodic location requests with PIN alert. The period between consecutive PIN requests was defined at 30 minutes as a mere example. The results are presented in Figure 5.3 and show that, in fact, this feature has some impact in the energy consumption. Having it as an optional gives the Tutor the option to decide to sacrifice a little more battery in exchange to have this mechanism, that may reveal that the Monitored device was left behind and is no longer reporting the monitored person’s real position.

5.4 Discussion

The SecureTracking solution aims at combining periodic position tracking and security measures. Adding this extra layer of processing represents an overhead to the normal operation of the device which has impact on the battery lifetime. To better argue on the impact of this solution, many implemented aspects were tested and some decisions were made by analysing time and battery metrics.

A portion of tests focused in analysing each localization method regarding the accuracy and the battery consumption. The results show that the Network-only localization method has very bad accuracy compared with Wi-Fi and A-GPS, despite having less variation. The average difference between the reported and real locations for the Network-only method was about 171 and 128 metres in residential and industrial areas, respectively. The Wi-Fi had better accuracy, having a difference of about 24 and 42 metres and the A-GPS got average differences of 21 and 53 metres in residential and industrial areas, respectively. The bad accuracy of Network-only, in addition of needing Internet connection, makes it an option of last resort. Still, having an idea of the location, even with bad accuracy, is better than
not having any indication about the location of the device. The accuracy of the A-GPS is influenced by weather conditions and surrounding buildings, which was the cause of having such big differences. Also, depending mainly on the A-GPS to obtain a location can result in many failed location reports if the monitored person is indoors. Regarding the battery consumption, the Wi-Fi has the disadvantage of having to turn the chipset on and scanning for nearby APs. On the other hand, if the device is already connected to a Wi-Fi AP, no operation is needed other than requesting the location. The localization using Wi-Fi proved to be a more balanced solution, with acceptable average real accuracy, and available both indoors and outdoors. This conclusion reinforces the localization strategy proposed in Section 3 and implemented in Section 4, which prioritizes the Wi-Fi usage over the A-GPS and lastly the Network-only. Regarding the battery consumption, the tests revealed that turning the Wi-Fi chipset on and off has some negative impact on this kind of localization method. The same impact was noticed in the Network-only method where the 3G communication was used to query Internet servers to get the location. The A-GPS proved to drain more battery than the Wi-Fi method, with an average of 3.7% against 3.1% on the later, but still is the second best option.

Another set of tests was made regarding the security measures, namely the MAC Algorithms. The results showed that the HMAC-SHA256 is the better choice between the selected candidates, has it provides proper security strength (relatively to HMAC-SHA1, for example) and takes, in average, less than half the time and less battery than CBC-MAC of 8 and 16 bytes.

The overall overhead of the security measures, in terms of time, is insignificant in comparison to the expected interval for updates. Considering the worst scenarios of average time consumption by summing a delay of $3.64 + 0.07 = 3.74$ ms for the MAC computation with $1.23 + 0.021 = 1.25$ ms for the cipher operation, we get an average of $3.74 + 1.25 = 4.99 \approx 5$ ms required for security mechanisms.
This value has no impact considering that the update intervals are expected to be around the tens of minutes.

The tests also allow to conclude that having the feature of periodically asking the monitored person to insert a code, acknowledging that the device is with him/her, has some impact on the battery. The test made using this feature consumed 4% more battery than another test where it was disabled. Nevertheless, it is still a simple method that can help detecting if the device is not with its owner and falsely reports his/her location.
Chapter 6

Conclusions

Currently, there are many systems capable of monitoring the position of a person, using existing platforms and infrastructures or implementing and installing new ones. Many of these systems focus on energy saving, implementing algorithms to detect the most appropriate localization method, or avoiding A-GPS based localization. There are solutions that use dead reckoning to estimate the current position based on the analysis of sensors like accelerometer and compass. Some of these systems also try to detect steps to estimate the distance travelled. Generally, position monitoring systems manage to detect the current position of a person and report that information to someone that is responsible for that person. This kind of information is critical and must be kept safe. Several security issues can put the person being monitored at risk. For example, in the case of a child or a mentally disabled person being monitored, leaking his/her location would make him/her vulnerable to many dangerous situations, like kidnapping or robbery. Few position monitoring systems provide security mechanisms to secure the communication channels used to report the position or to evaluate if it actually corresponds to the person's real location. Privacy is another issue associated with these systems, since they could become a mean of excessive control.

This thesis proposes the SecureTracking to provide means of monitoring the position of a person, applying some control to that position using the concept of geofencing and provide security measures regarding the communication channels and the localization process itself. The proposed architecture is composed of a device transported by the person whose position is being monitored, devices held by the person or persons responsible for that monitoring duty, and an intermediate system, a Repository Server. Localization is done by obtaining the current position using the Wi-Fi positioning system, A-GPS and cell antennas based localization, in that order of priority. This process takes advantage of the balance on accuracy, battery consumption, availability and time taken by Wi-Fi localization method. The A-GPS provides an alternative if no Wi-Fi APs exist nearby and there are decent weather conditions and good line of sigh with clear sky to get the current location, sacrificing some more time and energy. Lastly, if the previews methods fail, cell antennas' position may be used to locate, even if with bad accuracy. Another feature is the possibility of applying geofencing by defining areas where the monitored device (and its owner) can or cannot be at. Also, it is possible to define SafeSchedules, which are intervals of
time when there is no need to obtain and report the actual location but simply a “heartbeat” message to acknowledge the correct functioning of the system. The secure communication channels are achieved by ciphering all messages with the AES ciphering with a 128 bit secret key in CTS mode. Messages are secured in both integrity and authenticity aspects by computing a MAC value using HMAC with SHA-256 hashing algorithm. The freshness of each message is achieved by including a timestamp value in the message to ensure that older or repeated messages are detected. Regarding the localization process security, consecutive location reports are tested to detect abrupt changes that could denounce spoofing attacks on the monitored device. The system also provides a mechanism to detect if the monitored device is no longer reporting the correct location of its owner. This is achieved by periodically presenting a PIN screen where the monitored person must enter a code than only he/she knows. If the system does not receive a correct code in some interval of time, an alert is sent by the Monitored device to its Tutors because the device may not be near its owner any more. If the Tutor does not receive some expected location report, or a “heartbeat” message if in safe mode, it may indicate that something is not correct, for example, the Monitored device was switched off.

6.1 Achievements

Considering the requirements established in Section 1 and the state-of-the-art analysed in Section 2, the SecureTracking system is a balanced solution that:

1. Is able to monitor the position of a person with a given periodicity and report that information to someone responsible for it.

2. Provides geofencing and “position on-demand” capabilities.

3. Provides confidentiality, integrity, authenticity and freshness in every exchanged message.

4. Analysis consecutive location reports in order to detect simple spoofing attacks.

5. Allows the definition of SafeSchedules, which are intervals of time when there is no need to obtain and report location.

6. Provides a periodic PIN screen that aims at acknowledging that the monitored device is still in the possession of its owner.

The implementation of this solution was evaluated in Section 5 regarding time and battery consumption and the accuracy of the localization methods. The system was able to obtain correct locations along various paths in both residential and industrial areas. The tests show that the Network-only localization method achieves worse accuracy than Wi-Fi and A-GPS. Between the latest two, A-GPS is able to report location with better accuracy but is influenced by the line of sign with clear sky and is unable to obtain any location in lower floors when indoors. In the implemented prototype, the Network-only method consumed more battery than expected due to 3G communication needed to query for the location. Various MAC algorithms were taken in account and tested regarding time and battery consumption.
The results show that HMAC-SHA256 is the most balanced solution, being safer than HMAC-SHA1, and faster and more energy saving than CBC-MAC16. The security mechanisms that were implemented have practically no impact in terms of time overhead.

6.2 Future Work

Despite having a complete solution and its implementation, meeting all the requirements established in this thesis, some improvements can be applied and some options can be considered to provide a better experience when using the SecureTracking system. Some possible future work directions are:

- The platform used to implement the Monitored device was able to provide access to many features, APIs and sensors that enabled the proper localization process and secure reporting of that information. However, the only anti-spoofing measure that was able to be implemented was a simple system than evaluates consecutive location reports and detects abrupt changes based on the velocity. Having the capability of properly analysing the strength of GPS signals would provide much better anti-spoofing measures and enable the detection of GPS signal emulators, for example.

- Dead reckoning should be further evolved and analysed as an option on both localization methods and assisting in anti-spoofing measure. Some state-of-the-art proved that this kind of positioning systems could achieve great accuracy with much less energy consumption than the called “tradi-cional” localization methods (A-GPS, Wi-Fi and CellID). It could also be used as a sanity test for the locations obtained with other methods, in order to detect discrepancies that could denounce location spoofing.

- Having the system implemented on a smartphone has the disadvantage of having and operating system that has to deal with other applications running “simultaneously”. This means than they share CPU, RAM, storage and battery with the SecureTracking application. Having a dedicated system, with hardware and software used solely by this system should be analysed in terms of feasibility.

- Regarding the Internet communication channel, some other options should be analysed other than using the Google Cloud Messaging to support a final decision on which message transmission system to use. One possible option could be implementing direct communication with the Repository Server.

- The Tutor PC Application can be improved, providing a better interface to the user identical to the one in the mobile application, like visually showing the location of the Monitored device and allowing user interaction with a map for geofencing operations.
Bibliography


Appendix A

Accuracy Maps

Figure A.1: Accuracy using A-GPS (test 1)

Figure A.2: Accuracy using Network (test 1)

Figure A.3: Accuracy using A-GPS (test 2)

Figure A.4: Accuracy using Network (test 2)