ISSN 2007-9737

Securing mHealth Applications Using IoTsecM Security Modelling: Dentify.Me mApp Case Study for Urgent Care Management

Ponciano J. Escamilla Ambrosio[1], David Robles Ramírez[1], Shada Alsalamah[2,3], Theo Tryfonas[4], Sandra Orantes Jiménez[1], Abraham Rodríguez Mota[1], Sakher AlQahtani[5], Thamer Nouh[6], Hessah Alsalamah[2], Shahad Almutawaa[2], Hend Alkabani[2], Mshael Alsmari[2], Nouf Alashgar[2], Abeer Alrajeh[2], Heba Kurdi[2,7]

[1] Instituto Politécnico Nacional,
Centro de Investigación en Computación,
Mexico

[2] King Saud University,
College of Computer and Information Sciences,
Saudi Arabia

[3] Massachusetts Institute of Technology,
Media Lab, Cambridge, Massachusetts (MIT),
USA

[4] University of Bristol, Faculty of engineering, Bristol,
United Kingdom

[5] College of Dentistry, KSU,
Saudi Arabia

[6] College of Medicine, KSU,
Saudi Arabia

[7] Cambridge, Mechanical Engineering Department,
USA

{pescamilla, dinora, arodrigm}@cic.ipn.mx, davrobles28@gmail.com, shada@mit.edu,
{asakher, edu.sa, tnouh, halsalamah, almutawaa, alkabani, alsmari, alashgar, alrajeh,
kurdi}@ksu.edu.sa, theo.tryfonas@bristol.ac.uk

Abstract. Mobile devices and the Internet of Things (IoT) are revolutionizing today's digital sectors, including healthcare. eHealth services delivery enables integrated mHealth care and informed-decision making for emergency medical services, especially in the event of disasters when every second could mean the difference between life or death. The risk of cyber-attacks directed to mHealth applications can compromise the availability and integrity of patient information, crippling care mobility and sometimes threatening patients' lives if decisions are made based on invalid information. Such risks can be treated by considering appropriate information security controls at the early stages of the mobile Application (mApp) development lifecycle for mHealth model of care. However, most developers consider security at a later stage, and even if they do, there is a lack of an appropriate tool to help them represent security requirements in design models. This has proven to be bad practice, resulting in insecure mApp development. This paper aims to bridge this gap by equipping analysts with the tool necessary to identify risks and treat them while designing the application. Therefore, we propose the approach referred to as Internet of Things Security Modelling (IoTsecM) for

mApp security modelling in mHealth. IoTsecM is a UML extension to model identified security controls against possible attacks to guarantee the existence of a security analysis and security mechanisms. Results show that IoTsecM, first, allows mHealth designers to apply and depict non-functional security requirements with the functional requirements. Second, its annotation illustrates meaningful information security requirements at early design stages as part of the mHealth application development lifecycle and not afterwards.

Keywords. mHealth, mobile application design, information security, internet of things, modelling, UML, SysML, UML extension, security controls, disaster management.

1 Introduction

The Internet of Things (IoT) represents a revolutionary transformation of the traditional Internet into an interconnected network of "smart objects", simply referred to as "Thigs", that not only collect data from the environment (they have sensing capabilities) and interact with the physical world (they can perform actuation, command and control over other things), but also use the Internet to provide services for information transfer, analytics, applications, and communications [1]. The main postulate of the IoT is that everything can be connected to the Internet, anywhere, at any time.

This means that a plethora of objects (for example, smart cameras, wearables, environmental sensors, home appliances, and vehicles) are 'connected' and generating massive amounts of data among an increased number of IoT users, services and applications across different disciplines. The collection, integration, processing and analytics of these data enable the realization of smart cities, infrastructures and services for enhancing the quality of human life.

There are many relationships between IoT things, mainly controlled through a mobile Application (mApp) or a web Application (webApp). Accordingly, new and lighter protocols have been developed to optimize IoT devices communications considering their respective constraints. The wave of interest towards adopting smart environments is evident as the number of IoT devices increases and thus the number of users.

According to Gartner [2], the number of IoT devices has increased by 31% from 2016 to 2017 reaching up to 8.4 billion in 2017 and it is expected to reach up to 20.6 billion in 2020 [2]. According to GSMA [3], wearable devices (wearables in short) are expected to occupy 13% of the devices used in IoT in the next five years, which explains the heavy investments by leading technology companies such as Google and Samsung in those devices [3].

The ubiquitous nature of IoT technology has made it particularly attractive to the healthcare sector, where it can radically change the way in which healthcare is delivered [4]. The potential improvements that IoT can bring to healthcare delivery services include better outcomes and increased efficiency, which will make healthcare affordable, acceptable, and available to anyone and anywhere at any time [5].

In general terms, eHealth refers to the use of information and communication technology, especially the Internet, to improve or enable health services and healthcare delivery [6, 7]. Hence, the use of IoT technology to implement eHealth services is a natural progression [8, 9].

In this context, mHealth (or mobile health) refers to the use of mobile devices, wearables and sensing technologies either by consumers or providers, for monitoring health status or improving health outcomes, including wireless diagnostics and clinical decision support [10].

In security terms, however, as the number of entities connected to the Internet grows, the attack surface grows as well. The IoT drastically expands the size and scope of what security teams need to protect as attacks can be performed targeting unsecured IoT devices. Hence, IoT systems impose new challenges that were not present in the traditional Internet, in terms of identifying security requirements and depicting security controls.

Commonly, the security requirements within IoT systems are reviewed as an after-thought, even when the information handled by those systems is very sensitive in most cases. In particular, the risk of cyber-attacks directed to IoT mHealth applications can compromise the availability and integrity of patient information, crippling care mobility and sometimes threatening patients' lives if decisions are made based on invalid information [11].

In this work, the recently proposed approach referred to as IoTsecM [12-13] is used for mApp security modelling in mHealth. IoTsecM is a UML extension which models identified security controls against possible attacks, resulting from a comprehensive security analysis practice, to guarantee the existence of a security analysis and security mechanisms from the designing stages of an IoT system, viewed from a software development life cycle, for example, analysis, design, implementation, and testing, within the waterfall development life cycle. Therefore, by applying IoTsecM, the security requirements analysis can be realized by identifying the particular system vulnerabilities and threats. As such, the IoTsecM UML extension is useful to depict the security controls from the analysis stage until the design stage, where the security mechanisms have been proposed and modelled.

The remainder of this paper is organized as follows. In Section 2, the Literature review of related works is presented. In Section 3, the IoTsecM approach is reviewed along with its mean features. In Section 4, the Dentify.Me mApp case study is presented. Section 5 develops the security modelling in Dentify.Me mApp using IoTsecM. Finally, discussion and conclusions to this work are provided in Section 6.

2 Literature Review

2.1 IoT Security Modelling

There are several approaches, which consider IoT security according to different viewpoints. The security goals in the IoT depend on the security requirements of each system, the computational power and the energy supply. Therefore, there are many security requirements within the IoT universe, nonetheless, in [27], a table of security requirements from the infrastructure point of view is presented (see Table 1). As seen in Table 1, in this proposal, the IoT environment is split into three categories: System Dependability, Communication Stack and User and Service Privacy. For each one of these categories, the authors find some requirements related to Confidentiality, Integrity and Availability, then they propose some security components to target the security goals: AuthN (authentication module), AuthZ (authorization module), IM (identity management), KEM (key exchange management) and TRA (trust and reputation authority). Nevertheless, the IoT involves more security requirements, for instance, in many scenarios tamper protection is also needed as it is authorization, not just for the service layer but also for the physical layer.

In [28], some high level security requirements are given: resilience to attacks, data authentication, access control, client privacy, user identification, secure storage, identity management, secure data communication, availability, secure network access, secure content, secure execution environment and tamper resistance. In this approach, there are more security considerations for IoT, for instance, the different layer requirements, according to a respective IoT architecture.

In [29], security requirements for IoT systems have been obtained considering the Industrial Internet Reference Architecture (IIRA) point of view. Hence, in their work, the authors envision four viewpoints: business viewpoint, usage viewpoint, functional viewpoint and implementation viewpoint. Consequently, they define the security requirements according to these viewpoints. For the business viewpoint, they are focused on the return on investment for security in the context of other considerations, such as performance or consumer satisfaction. For the usage viewpoint, they propose: security monitoring, security auditing, security policy management and cryptography. For the functional viewpoint, they provide common security functions: security audit, identify verification, cryptographic support, data protection and privacy, authentication, identity management and physical protection. For the implementation viewpoint, the authors outline common security issues: end-to-end security, protected device-to-device communications, confidentiality and privacy of data collected, remote security management and monitoring.

2.2 mHealth as Part of eHealth

Emerging healthcare delivery models help improve the patient's quality of life by facilitating the delivery of holistic, integrated, patient-centered care. eHealth is the most recent health care delivery

Table 1. Security requirements involving the security infrastructure

System Dependability	Communication Stack		User and Service Privacy
	Network Layer	Service Layer	
Service Availability	Network-level Anonymization	Service Access Control/Authorisation	User privacy protection when using Infrastructure
Infrastructure Availability	Confidentiality	Service Authentication	User privacy protection when using Services
Infrastructure Integrity		Service Reputation Metering	Privacy protection of Service towards User
Infrastructure Trust		Service Trust	
Non-repudiation			
Accountability			

model in our modern world, which utilizes Information and Communications Technology (ICT) and modern sensor technologies in the delivery of health treatment to enhance collaboration, communication, and coordination in the health sector [7-14]. mHealth is one of the emerging models of eHealth that has been maturing since the mid 90's and it still does not have a standard definition [15-16]. However, the National Institutes of Health (NIH), [17] defines mHealth as: "The use of mobile and wireless technologies along with wearable and fixed sensors for the improvement of health outcomes, healthcare services, and health research'' [18]. mHealth is considered the one form of eHealth delivery models that concentrates on coordinating integrated care and actively managing it remotely in our world today [16].

Therefore, mHealth supports health by heavily incorporating mobile devices into the healthcare delivery model to mobilize care delivery to reach patients like never before, whether in emergency medical services or home care. mHealth involves the use of many services and utilities supported by a mobile phone device, such as text messaging, voice recognition, Global Positioning System (GPS), and Bluetooth technology [7].

Consequently, mHealth is expected to grow and occupy 10% of IoT devices in the next five years [3] and is predicted to have a massive impact on the way healthcare is delivered in our modern world as it enables affordable personal management of the user's well-being.

According to this data, the IoT systems related to mHealth must consider security in the analysis and design stage, since the information that mHealth collects is very sensitive and it would involve huge risk if it lies on the wrong hands, such as malicious attackers.

2.3 mHealth Technologies in Disaster Management

A number of mHealth technologies have been designed to save victims' lives in disasters by equipping them with the right tools that would facilitate their call for emergency. First, the international Morse code distress signal "SOS" is an internationally recognized call for help. It is used especially by ships in distress. Nowadays, many devices and applications have integrated SOS as one of their features. For example, Apple introduced the SOS feature in Apple Watch as one of its main features and it can be used by pressing and holding a side button; no matter where you are in the world, SOS initiates a call with local emergency services, and it can also send a message to quickly alert registered emergency contacts [19].

Second, in recent years, mApps helped in reducing the harm and damage done by either natural or human disasters, such as generating early warning systems, aiding in emergency coordination, and improving public communications in times of crisis [20].

mHealth, in particular, has played an important role in emergency aid in the event of a disaster where it enables victims to record personal medical information. This enables the rescue team to give the victim the right treatment based on informed-decisions [21].

Third, location detection technologies have enhanced the service delivery by emergency services [22]. For example, GPS plays a vital role in relief efforts for global disaster responses, where a rescue team uses GPS to detect the disaster's location. This makes it faster and easier to obtain accurate coordinates that locate the victim in real-time in the middle of a disaster [23].

Finally, in disasters, a victim can use their mobile phone to ask for help by calling or texting. However, this is sometimes not an option when, for instance, an access overload occurs at a time when everyone is trying to call in or out, or when a victim is unable to reach out to their phone either because of their injury, or if buried under rubble.

In such cases, mobile wearable devices with a touchscreen display (such as smart watches) would be a more suitable option, especially if designed to be worn on the wrist and so attached at all times while controlled through an interactive user-friendly interface which is hands-free [24]. Therefore, a wearable device can be helpful in the event of a disaster where it is easy to use, available and faster to reach while worn on the wrist, and it provides speedy access whilst offering features such as speech recognition.

Most wearable device models support wireless communication using technologies such as Bluetooth, Wi-Fi, and GPS [25]. In recent times, smart watches are being increasingly used; Apple iWatch [25], in particular, is considered one of the most famous and best-selling wearable devices [26]. By using Apple iWatch, users can now access information in a way that is both distinctly personal and unobtrusive [25].

Conversely, in security terms, the deployment of mobile devices and wearables into the mHealth ecosystem presents a trade-off between great opportunities and risks, and so the aim is to maximize the rewards of this model of care while mitigating risks for all stakeholders in this ecosystem.

3 IoTsecM Security Modelling

Developing mApps for time-sensitive situations demands a great deal of systems analysis and design to ensure the solution balances between the value those applications add to victims in saving their lives and the risks they pose. The software development life cycle goes through several activities to reach the final software application, for example, analysis, design, implementation, and testing, within a waterfall development life cycle [13]. Furthermore, IoT mApp security analysis will be fundamental since the information that IoT systems handle is very sensitive. It is not just passwords or virtual files; human life could be threatened if an IoT system fails.

Although it is impossible to propose a unique solution to the security concerns within the IoT system, an analysis of the common attacks can lead us to a better understanding of the security and the well-placed security controls will help to shrink the attack surface. However, IoT, just like any informatics system, must guarantee three security goals: Confidentiality, Integrity and Availability (CIA). Nonetheless, in IoT, these goals are wider because there are new actors as sensors and actuators. Furthermore, there are many application domains in the IoT, each one with countless systems and applications; each system has its own security requirements since the assets are particular and application-dependent. Therefore, a specific analysis for each system must be done, where security mechanisms can be proposed, attending to the threats and vulnerabilities related to the system at hand.

In this context, the IoTsecM approach [13] proposes a UML/SysML extension nomenclature to take into account security requirements along the analysis stage within a well-defined development life cycle, such as the waterfall model. Some of the nomenclature components have been proposed in the IoT-A proposal [27] as security modules, however they were not proposed as a UML/SysML extension. The IoTsecM proposal is a UML/SysML extension which provides a set of well-defined elements which abstract the security concerns of IoT systems, and allows for them to be deployed over the UML/SysML diagrams [13].

Therefore, it is a graphic representation which integrates fourteen security elements depicted in a nomenclature, and encapsulated in stereotypes. The IoTsecM nomenclature encapsulates the security concerns of IoT systems in the stereotypes, and such stereotypes are depicted in Table 2. Each one of the elements encapsulates an IoT security service, and has a short representation (nomenclature elements), the corresponding UML extension mechanism and the metaclasses extended by the element. These elements are used inside of the extended UML/SysML diagrams because they are high level abstraction security requirements and they encapsulate the CIA security goals. These actors are incorporated in use case diagrams; besides, the nomenclature can be applied as use cases if the security requirements request it. The D, N, KM and IM elements also can be modelled as use cases.

This will allow for a more agile design process for security requirements, because even the developers who are not involved in security can recognize these elements. In the class diagram, some nomenclature elements can be modelled as classes, and, in fact, according to the IoT security requirements analysis performed in [13], these elements are: N, Z, C, D, B&B, KM, IM, T&R, CA and TP. The TP, SS and SC elements are constraints depicted as [TP], [SS] and [SC], and these three elements are used mainly in the use case diagrams and in the UML behavior diagrams. A sequence diagram depicts objects interactions chronologically; therefore, the classes mentioned before, that apply the nomenclature, appear in the sequence diagram as objects.

As described, IoTsecM integrates a profile which addresses the designing and modelling of IoT systems considering a security architecture; it helps to depict the system security concerns and the security mechanisms should be in an accurate place, in order to protect the system against a real threat or attack. Once the possible attacks over the system at hand are identified, the developer will then be able to establish a way to provide protection or countermeasures against those attacks, and therefore they would be able to find the right place for the right countermeasure for a particular attack or threat.

Threat modelling can be achieved by different methods, and there is not a unique methodology which helps to mitigate the system risks. The main objective in threat modelling is to know the system threats and vulnerabilities, which surely would be exploited by a motivated attacker if countermeasures are not there to prevent them. Along with threat modelling, attack analysis also needs to be performed regarding all the ways in in which an attack can be successful, with the aim of showing how the vulnerabilities of an IoT system can be exploited.

For this analysis, attack process and attack trees are commonly used. Attack trees are one way to model the attacker behavior against the system assets [34]. Typically, an attack is grouped in a sequence of sub attacks or other activities that are individually focused on obtaining an immediate target. The attack trees let us model these sub attacks and the steps that need to be followed to obtain the target.

This attack representation helps to conceptualize, visualize, and communicate a better understanding of the sequence of vulnerabilities that can be exploited.

In the next section, the IoTsecM profile is applied to the Dentify.Me mApp case study, and threat modelling and attack trees are obtained. Based on the analysis performed, and using the IoTsecM profile proposed, security countermeasures are derived and represented in the IoT system's architecture.

4 IoTsecM Security Modelling

4.1 Victim-Centred mApp Solution

Every single day, humans around the globe are subject to many calamitous incidents and disasters. In fact, horrifying mass fatality incidents and disasters have been witnessed in every corner of the planet, whether as a result of natural disasters, war or violent acts. In such incidents and disasters, all efforts should be combined to rescue surviving victims, find those missing, and identify those found. Firstly, all injured victims need to be located or reported to be provided with first-aid paramedic services.

Table 2. IoTsecM nomenclature

Element	Name	Extension mechanism	Base meta-class(es)
N	Authentication	Stereotype	Class, use case, component, block, activity and state
Z	Authorization	Stereotype	Class, activity, component, block, state and use case
C	Cypher	Stereotype	Use case, component, block, class
D	Decipher	Stereotype	Use case, class and component
SS	Secure Storage	Stereotype	Link, property, association, communication path and constraint
SC	Secure communication	Stereotype	Constraint, communication path and link
KM	Key management	Stereotype	Class, component, block, device and association class
T&R	Trust and Reputation	Stereotype	Class, block and component
IM	Identity management	Stereotype	Class, block, component, activity and association class
Ps	Pseudonym	Stereotype	Actor and constraint
CA	Certification authority	Stereotype	Class, block, component and device
RA	Registration authority	Stereotype	Class, block and component
TP	Tamper protection	Stereotype	Constraint and property
BM	Behaviour monitor	Stereotype	Class, block, component and device

Secondly, missing victims need to be identified and listed so all Disaster Victim Identification (DVI) efforts can be best exploited to find the missing victims when every second could mean the difference between life or death.

Finally, deceased victims need to be identified to bring them justice in case of a homicide or peace of mind to their friends and family. However, rescue teams rely on victims or eyewitnesses to report incidents manually or on post-active systems for receiving emergency calls.

Furthermore, the generation of a missing persons' list takes a considerable amount of time and is mostly not accurate initially until it starts receiving

reports from families and friends. Moreover, all pre and post-disaster data, also known as Ante-Mortem (AM) and Post-Mortem (PM), respectively, are collected and matched manually by DVI teams for missing persons' identification. Finally, there are no holistic pro-active systems today in the IT market to address all of these three issues at once.

Dentify.Me mApp [34] is a proactive application that proposes a victim-centered solution that automates the process of requesting S.O.S and locating victims, generating missing persons' lists, and collecting and delivering AM data. Using semi-structured interviews, we identified and gained a good understanding of AM and PM data needs, the teams involved, and information sharing and information security needs and challenges. Furthermore, in line with supporting social policing concepts, Dentify.Me mApp engages with eyewitnesses to automate the incident detection and identification process.

Finally, Dentify.Me mApp automatically identifies and locates potentially affected victims, alerts the rescue team so that they are able to locate survivors, generates a list of missing people, and collects and delivers AM data to the DVI team. This paper fully designs a secure mHealth mApp solution for disaster management ecosystem. This proposed solution is hoped to have a significant impact on mass fatality-victims by facilitating the rescue of survivors, finding the missing, and identifying those found to bring justice to the victims and peace of mind to their loved ones.

4.2 Designing CIA into Dentify.Me mApp

Although the Dentify.Me mApp team have fully designed, implemented and tested the proposed mobile health solution, the security analysis for the mApp's CIA was not realized. Therefore, a well-motivated attacker may collude with malicious people to injure or even kill people; the informatics attacker may attack the system's infrastructure and assets, provoking the system to not respond or respond incorrectly. Furthermore, a successful cyberattack on the Dentify.Me mApp may be even more catastrophic since the rescue teams would not be able to react at the right time and in the right way and, as a consequence, could even lead to increased numbers of fatalities and injured people.

Consequently, the security analysis must consider the effects that an attack can cause, in particular, in systems related to eHealth and especially in mHealth systems.

In this paper, the security analysis of Dentify.Me mApp was developed. The method followed to perform the threat modelling and security controls analysis and design is based on [30]; however, this process was customized and extended in order to add the controls modelling proposed in IoTsecM. The process followed is summarized in the following steps:

- Identify assets,
- Create an IoT system architecture overview,
- Decompose the IoT system,
- Identify threats,
- Document threats,
- Propose controls for each threat,
- Propose a system architecture depicting security controls.

A short description of the development of each of these steps is presented in the next section. However, for a detailed discussion of each one of these stages, and for the matter of space, the interested reader is referred to [12, 13].

5 Security Modelling in Dentify.Me mApp

5.1 Assets Identification

First, a clearer idea of the system is needed. As mentioned earlier, the study case is the Dentify.me mApp, which has already been designed [34], nevertheless the security requirements have not yet been considered, and so security analysis and threat modelling needs to be performed. As mentioned in the steps introduced before, the assets identification within the system offers an understanding of what needs to be protected, and may include: people, hardware, software, procedures and data information. It is an inventory of all the items (virtual or physical) that are important for the system or for the organizations and can also be to the attacker's interest. In the Dentify.me mApp, an assets classification is provided in order to identify the assets according to their particular type.

The list is introduced as:

- People,
- Eyewitness,
- Injured and missing victims,
- Rescue team,
- DVI team,
- Hardware,
- Mobile,
- Wearable devices,
- Desktop,
- Printer,
- Device for recognition,
- Switch,
- Router,
- Cable for connection,
- Software,
- Firewall software,
- Operating system software, Network operating system,
- Procedure,
- User registration in app,
- Basic information from all users,
- Identification and location of victims,
- List of potential missing people,
- Request a response from every victim on that list,
- Start timing and generate four sub-lists:
    - Sub-list: Alive and Well Victims,
    - Sub-list: Alive and Injured Victims,
    - The list is shared with the rescue team,
    - Sub-list: Pending Confirmation,
- The list is shared with DVI team,
- Create a list for missing people sub-list,
- App allows the victim to trigger S.O.S. request,
- Rescue team is able to access Affected Victims List,
- Data/Information,
- Basic information from all users,
- Data (Pictures/Video/Notes),
- Information,
- Networking,
- Server,
- Host,
- Clients.

5.2 IoT System Architecture Overview and Decomposition

Once the assets are identified, they are placed together in an architectural view in order to observe their associations; the Dentify.me mApp architecture can be found in [34]. The class diagram is used as a conceptual diagram in order to show how the assets are located within the system and to observe how they are associated. In Fig. 1, the original Dentify.Me mApp architectural view is shown (without cybersecurity requirements represented).

5.3 Threats Identification Using Attack Trees

At this point, the system architecture was depicted and analyzed, the assets identification was performed, and hence it is time to analyze the security requirements. Attack trees are an orderly and sequential way of describing the sub-attacks that compromise a system; they are a useful tool to conceptualize and visualize possible attacks, where the designer must put himself/herself in the attacker's shoes to devise all the different ways in which an asset can be compromised. This analysis results in the underlying root causes of attacks, allowing for the creation of attacker profiles, in order to make decisions about the possible mechanisms and security controls needed to protect the system against certain attack vectors and thus reducing the attack surface.

Building an attack tree is not an easy task since it must consider, as far as possible, the entire attack surface. It is recommendable to do it within a work group where at least two people can build it together. The tool used for this purpose is named SecureITree, an attack tree modelling tool built by the Canadian company Amenaza (the Spanish word for threat) [32]. In this tool, the root node represents the end objective and the children nodes depict the different sub attacks in order to accomplish the overarching goal.

The nodes can be AND operator, OR operator, or a LEAF. The AND operator means that all of the children nodes are needed to accomplish the parent node. On the other hand, the OR operator means that any of the children nodes satisfy the parent node.

ISSN 2007-9737

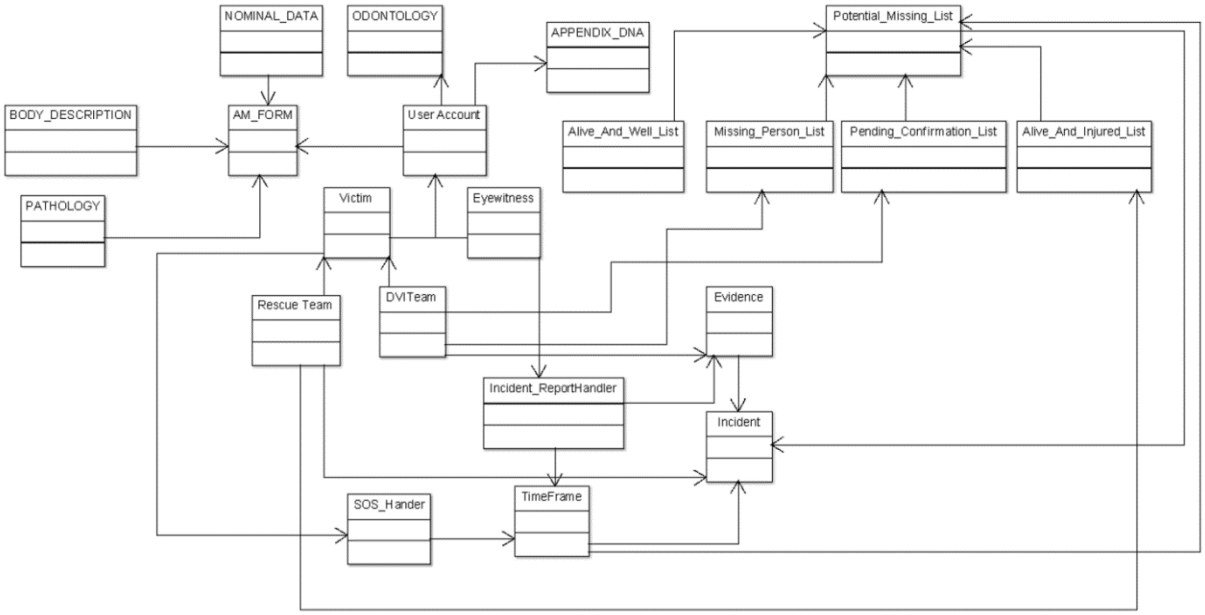

Fig. 1. Dentify.Me mApp original architectural view

The first attack tree to analyze is the one with the parent node “Eyewitness unable to report Incident” shown in Fig. 2; meaning that the witness is not able to report an incident because some sequence of attacks is being executed by an attacker.

The attacks related to social engineering are particularly difficult to address since they rely on human mistakes, and the countermeasures against this kind of attack are to educate people to be aware of the possible ways in which an attacker may achieve their goal.

For instance, the attackers may provide the right credentials, and hence stronger policies for physical human access control need to be enforced; such attacks, however, are out of the scope of IoTsecM.

The attacks and sub attacks to the Dentify.me App assets are described in this section; however, not all the attacks are described, since some of them are not a relevant part of the attack vector. Therefore, the attacks description provided below involves the relevant attacks accompanied with the corresponding countermeasures. The nodes that are the offspring of “Eyewitness unable to report Incident” threat (see Fig. 2) are:

- Error in data entry: The sub attacks described below are attempts to provoke an error in the data entry. To mitigate this attack, the sub attacks need to be mitigated and, as a consequence, this attack will be mitigated as well.
- Malicious app changes the data.
- Download and Install malicious app.
- Phishing attack: With malicious links and another technique, the attackers may force the users to install malicious apps in their smart phone; in this attack, these malicious apps provoke data changing and, consequently, errors in data entries. The countermeasure against this attack is to have an anti-virus installed within the smart phone, which prevents the malicious apps installation.
- Colluded app: An app is installed which is colluded with another app that provokes the error in data entry. The countermeasure against this attack is to encapsulate (sand boxing) the Dentify.Me mApp, in order to deny access to any other app within the smart phone to the Dentify.Me mApp. Therefore, the

countermeasure is an authorization process between the smartphone apps.
- Error in the app:
- The app was not loaded.
- Malicious App Installed: The malicious app does not allow the Dentify.Me mApp to load, hence, the eyewitness is unable to report the incident. The countermeasure against this attack is an authorization process between the smartphone apps.
- No Internet connection.
- Wi-Fi jamming.
- False Wi-Fi Access Point (AP): The attacker creates a false access point, then the data sent from the smart phone cannot be received by the Incident_ReportHandler or it may be modified. These kinds of attacks are part of the MITM (Man-In-The-Middle) attacks. The countermeasure against this attack is to provide an authentication method between the eyewitness and the Incident_ReportHandler.
- No mobile data.
- No cellular network.
- Cellular Network Jammer Attack: This kind of attack is very common in terms of compromising a wireless environment. For example, in communication between the eyewitness and the Incident_ReportHandler, the goal is to drop the signal to a level where the communication is interrupted. Typically, older wireless area networks are the most vulnerable to the success of this kind of attack, since current networks are able to adapt to unintentional or intentional interference. The countermeasure proposed for this attack is an intrusion prevention system (IPS), since it should be able to detect the presence of any unauthorized client device.
- Camera Malfunction:
- Malicious App disables the camera.
- Download and Install malicious app: These attacks involve the installation of malicious apps, and these were previously described, as well as the controls proposed. The only change is that these attacks attempt to provoke a camera malfunction.
- Phishing attack.
- Colluded Apps.
- Break camera.
- Mobile battery discharged:
- Malicious app changes the battery level.
- Steal the eyewitness mobile phone:
- Direct attack to the eyewitness mobile: This is a physical attack; the attacker steals the eyewitness' mobile phone.
- Server does not respond:
- DoS (Denial of Service) attack to the server: This attack consists of sending many requests to the server, in order to make it attend to just the false and malformed requests whereas they deny any other requests, even the authenticated requests. The control against this attack is an intrusion detection system (IDS) or an intrusion prevention system (IPS).

The second threat analyzed is the "Rescue Team cannot access affected Victim List" which means that the rescue team does not have access to the affected victim list; this is displayed in Fig. 3 and it considers the following possibilities or sub attacks:

- No Internet connection.
- False Access Point: A Man in the middle (MITM) attack is performed by an attacker; the attacker creates a false access point and therefore is able to receive, modify or block the communication between the rescue team and the affected victim list. The countermeasure proposed against this attack is the authentication of the rescue team, and a trusted victim list.
- Wi-Fi jammer: The countermeasure against this attack is an IDS or IPS control.
- Access a false affected Victim List.
- Modify, delete, observe the list.
- Password force Brute-Attack: The attacker performs an attack against the server access mechanism by a brute-force attack, which means that, according to a password dictionary, the attacker tries each one of the possible combinations. The countermeasure against this attack is a well-defined authorization mechanism and an IPS in the server.
- Social engineering.
- Denial of Service attack: The countermeasure against this attack is an IPS placed in the server, which is able to monitor the port server.

The next attack tree has a root node "Communication interception from mobile", and

ISSN 2007-9737

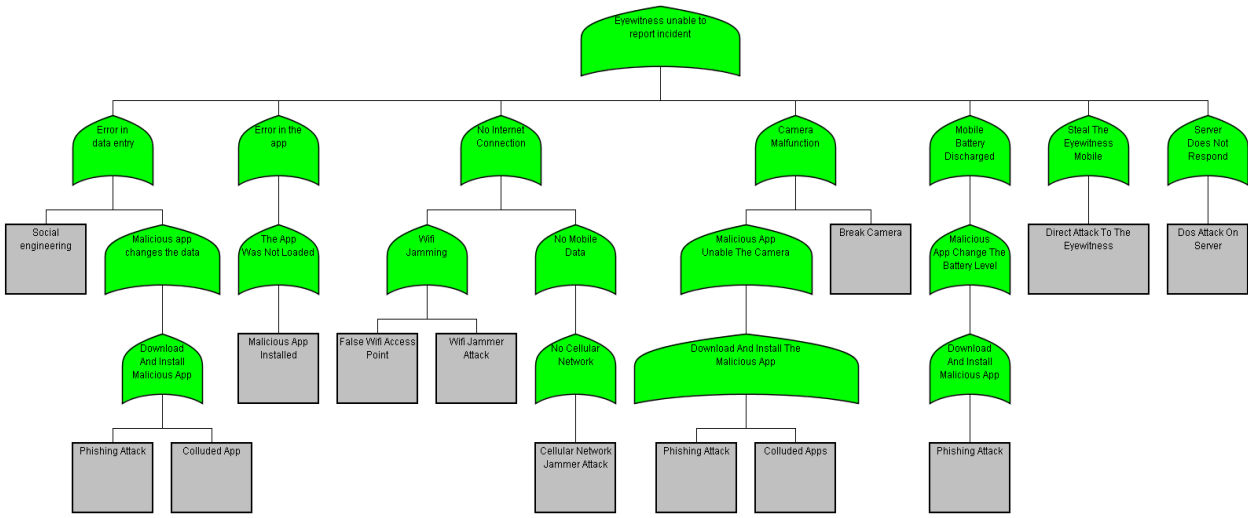

Fig. 2. Dentify.Me mApp architectural view

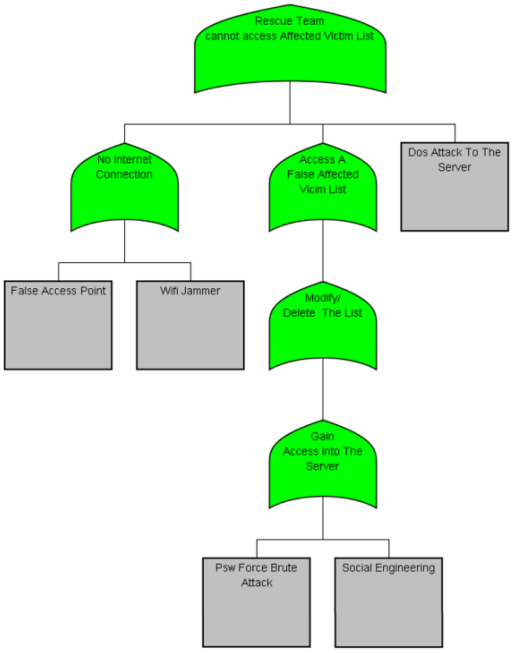

Fig. 3. Rescue Team cannot access affected victim list attack tree

this means that an attacker aims to eavesdrop the data traffic from the mobile to the server. The attack tree regarded for this threat is shown in Fig. 4:

- Communication interception from mobile.
- Man in the middle (MITM) attack: The attacker performs an MITM attack in order to intercept the communication, and this can be performed

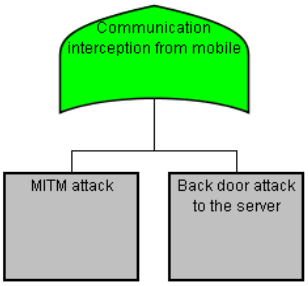

Fig. 4. Communication interception from mobile attack tree

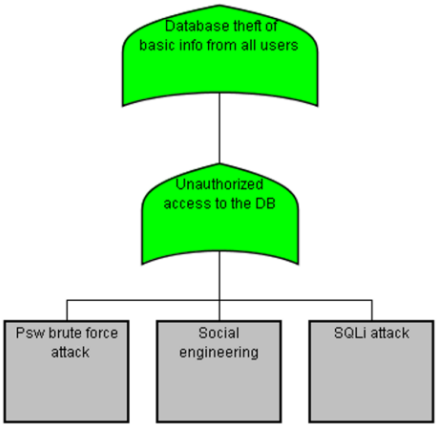

Fig. 5. Database theft of basic information from all users attack tree

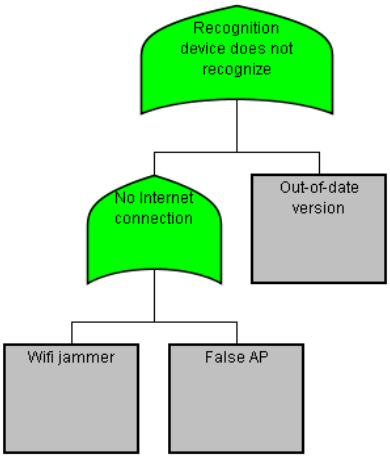

Fig. 6. Recognition device does not recognize attack tree

in several ways, however the countermeasure proposed against this attack involves authentication and authorization controls from the mobile to the servers.

– Back door attack to the server: The attacker identifies a vulnerable service, and performs a back door attack in order to take control of the server, in this case, taking control of the server communications. The countermeasure against this attack is an IPS control in the server and authentication mechanisms in each service for the mobiles.

The next attack tree is “Database theft of basic from all users” where an attacker tries to modify, delete or observe the database and this would imply unauthorized access to the DB. The attack tree is shown in Fig. 5, and deployed as follows:

– Database theft of basic information from all users.
– Unauthorized access to the DB.
– Password brute force attack: This is an attack against the server access mechanism by a
– brute-force attack. This means that, according to a password dictionary, the attacker tries each one of the possible combinations to gain access to the database. The countermeasure against this attack is a well-defined authorization mechanism and an IPS in the server in order to identify and react against the malicious behavior.
– Social engineering.
– SQL injection attack: This attack is performed by entering crafted data that provokes the input to be interpreted as part of SQL query instead of data. The countermeasure against this attack is sanitization and validation that could be part of an authorization mechanism. The objective is to ensure that any malicious characters are not passed to an SQL query for data.

The next case considered is “Recognition device does not recognize”, and this means that the mobile camera or any other device dedicated to recognizing does not work properly. This could be because of a mobile malfunction or because of an attack and, therefore, the attack tree for that threat is modelled and depicted in Fig. 6. The countermeasures against the two attacks are described as follows:

– Wi-Fi jammer: The countermeasure against this attack is an IDS or IPS control. However, as mentioned, the recent Wi-Fi AP contained anti-jamming technology; thus, another recommendation is to not use old Wi-Fi AP.
– False AP: The countermeasure against this attack is an authentication mechanism in order to verify the identity of the recognition device.

The analysis of the "Unauthorized access to router" threat (Fig. 7) is when an attacker crosses the router because no security mechanism is implemented in the children nodes; the analysis of the results is as follows:

– Unauthorized access to router.
– Old equipment.
– Attack software or hardware vulnerability: The countermeasure against this attack is to place only current equipment for the Dentify.Me App system.
– No access control mechanism implemented: The countermeasure against this attack is an authorization mechanism implemented in the router.

The next attack tree considers the "Unauthorized access to the lists" (Fig. 8), and this means that an attacker will focus his efforts on accessing the database server with the aim of modifying, observing or deleting the Dentify.Me App. The sub-attacks considered are:

– Unauthorized access to the lists.
– Social engineering.
– Get in with right credentials.
– Password brute force attack: This attack has already been described; the countermeasure against this attack is a blend of an authorization mechanism and an IPS control.
– Social engineering.
– Physical penetration.
– Social engineering.
– Credentials falsification.

The non-repudiation security requirement is a sub-branch of the general availability requirement. In this case, an authorized actor needs to be able to access the resources requested. In the case of Dentify.Me mApp, the name of the attack against this security requirement is "DVI team cannot access a Pending Confirmation List" and the

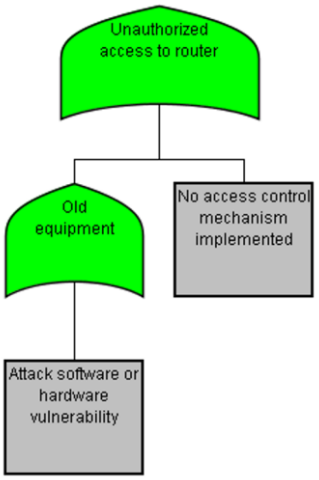

Fig. 7. Unauthorized access to router attack tree

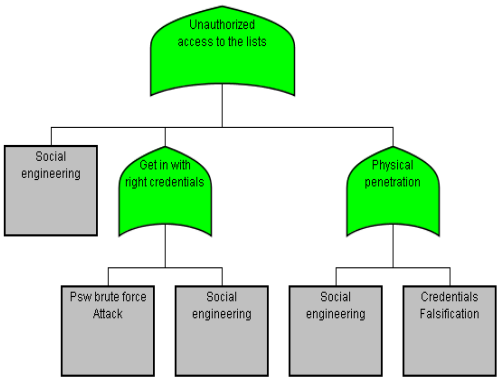

Fig. 8. Unauthorized access to the list attack tree

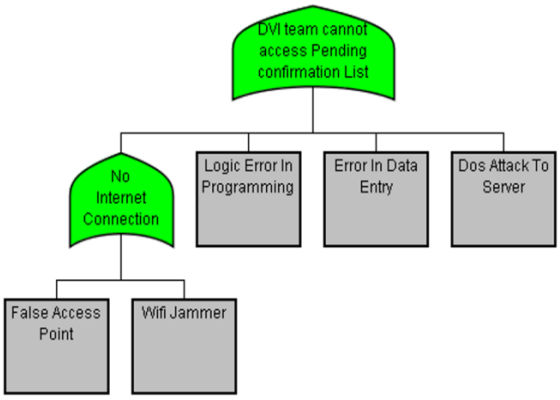

Fig. 9. DVI team cannot access pending confirmation list attack tree

analysis results are shown below (Fig. 9). The controls against these attacks have been described previously, hence they are not described in this section, and only the sub-attacks names are shown:

- DVI cannot access the Pending Confirmation List.
- No Internet connection.
- False access point.
- Wi-Fi jammer.
- Logic error programming.
- Error in data entry.
- DoS attack server.

Some attacks are solved with users' good practices; others need to be addressed during the programming of the application, such as social engineering attacks, and these are usually accomplished in three ways:

- Persuading the user: In the case of Dentify.me mApp users, this is when the user downloads a malicious application that blocks the camera, for instance.
- Physically get into the servers: Using fake credentials or convincing people are the most common practices.
- Remote access: With the correct passwords obtained indirectly or directly from users and programmers, for example.

Therefore, users should be made aware so that they do not install applications that appear malicious, for example, applications that ask for permissions that are not congruent with the application's intended performance. If this recommendation is followed, then the risk of having an application that blocks a hardware asset will be greatly reduced, assuring the correct functioning of Dentify.Me App, especially in urgent situations.

Another form of attack is phishing, where a malicious link is given to download some software. This can be done through fake emails or web pages that contain them, therefore users and members of the DVI team and Rescue team are advised not to download anything outside of the necessary and validated applications.

In the case of Jammers, these are used when the attacker intends to gain time for a physical attack; for example, it can be used to deny the opponent the opportunity to communicate in time in a critical situation. The countermeasure for this attack could be very complex; in the case of Dentify.Me mApp, it is recommended to have other data outputs, as is usually the case: Mobile and Wi-Fi data. Since the application will be running in a non-controlled environment, no featured protocol can be implemented because Access Points would need to know this.

A direct attack on the Eyewitness is very difficult to avoid, since it cannot be predicted, however, if the witness observes the incident, he or she should act cautiously and calmly, find a safe place where they can continue observing, whilst considering their own and others' safety.

The DoS Attack is when an attacker or a group of attackers make many requests to one service, with the aim of "saturating" the server and keeping it busy by responding to empty requests, therefore the server is not able to attend to any current requests. The countermeasure proposed against this attack is a behavior monitor, which is usually an Intrusion Detection System (IDS) or an Intrusion Protection System (IPS). This element is represented in the class diagram as a <<BM>> stereotype instance and it is placed in the IncidentReportHandler as shown in Fig. 11.

The man in the middle attack (MITM) is where the attacker places himself between two assets which exchange data, and the attacker inspects ,traffic and finds usernames, passwords, or any data sent in plain text.

This is the reason why the countermeasure against this attack in the Dentify.Me mApp study case uses point to point encryption, in addition to using encrypted network connections (HTTPS or VPN). Within the Dentify.Me mApp study case, it was aimed to encrypt the communications between the mobile device and the server, that is between the Eyewitness and the Incident_ReportHandler, and this is represented using two extension elements of IoTSecM: <<C>> and <<D>> stereotypes.

For attacks on the database, it is firstly considered an SQL injection (SQLi) attack, where an attacker can execute malicious SQL statements that control the database of web applications (Relational Database Management System - RDBMS). If an SQL vulnerability is exploited, an attacker could bypass the web application's

authentication and authorization mechanism in order to observe, modify or delete content from the entire database. According to OWASP [26], avoiding SQL flaws is simple: to stop writing dynamic queries prevents malicious SQL statements from affecting the logic of the executed query. In [32], a set of simple techniques is provided to prevent SQL Injection:

- Use prepared statements.
- Use of stored procedures.
- Whitelist input validation.
- Escaping all user supplied input.
- Enforcing least privileges.
- Performing whitelist input validation as a secondary defence.

For Dentify.Me mApp, it is proposed to follow the recommendations given by OWASP although, as an addition, we propose database encryption, since the requests to the database are not of an urgent nature, nevertheless the user´s information is very sensitive as well as the lists, therefore the <<C>> instance is proposed to protect the information confidentiality. The <<C>> stereotype will be used as a crypto-module by the Potential_Missing_List, Eyewitness and User Account.

5.4 System Architecture Depicting Security Controls

The IoTsecM use case for Dentify.Me mApp is shown in Fig. 10 where there are six actors: User, Victim, Eyewitness, Rescue Team, DVI Team and Emergency Contact. According to the analysis performed before, authorization is required for the Users, Eyewitness, Rescue Team, Emergency Contact and DVI team, and this will protect the system against unauthorized access and, as a consequence, it will not allow unauthenticated actors to access the resources.

As mentioned earlier, a text box over the actor's head within a Z depicts that the actor must be authorized. The Z element normally implies that an N instance is implicit. A secure communication ([SC]) requirement is needed as well, in order to indicate that the channel between the user and the account management needs to be secure. SC will protect the information against MITM attacks since the attacker would be able to observe the communication flow, nevertheless they would not be able to understand them. The use case diagram is shown in Fig. 10.The IoTsecM class diagram depicts the system architecture where the security classes are shown, see Fig. 11.

Once the threat analysis is performed, the use case diagram regarding security requirements is obtained, then the IoTsecM class diagram is proposed, where the security requirements are depicted with the functional requirements within the same diagram. This provides a better understanding of the security necessities, the costs, and the system reliability. The <<C>> stereotype instance is used by the AM_FORM class to provide a confidentiality requirement; this same element is utilized by the Potential_Missing_List, Eyewitness and User_Account. The <<Z>> stereotype instance depicts an authorization mechanism which provides protection against the attacks described earlier. The <<Z>> element authorizes the Rescue Team to access the victim information.

As shown in Fig. 11, the Z element normally requires an authentication mechanism (N element), which helps to first authenticate the actor, then once it is authenticated, an assertion is passed to the Z element and it will verify its access control list (ACL), or any other control implemented to authorize the actor and guarantee some rights, whether it is the writing, executing or reading rights.

The <<BM>> stereotype is applied to the Incident_ReportHandler, and this indicates that an IPS needs to be designed in the next stage of the development life cycle; these security mechanisms will mainly help to prevent the DoS attack.

6 Conclusions

One of the main characteristics of IoT systems is the integration of "intelligence" in the objects around us. This is achieved by providing objects (things) with sensing, acting, storage, and processing capabilities, but overall by providing them with interconnectivity via the Internet, in order to provide services to different users. This interconnectivity concept, anywhere, at any time, using any network, facilitates the development of a myriad of applications in many domains.

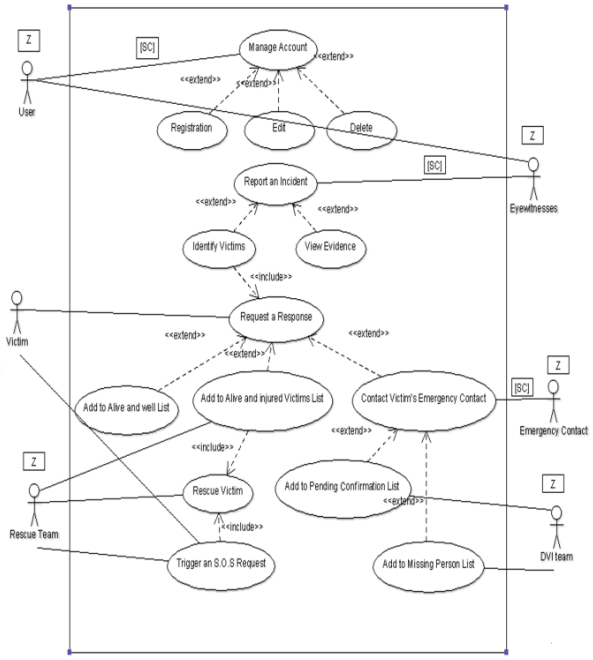

Fig. 10. IoTsecM use case diagram for Dentify.Me mApp

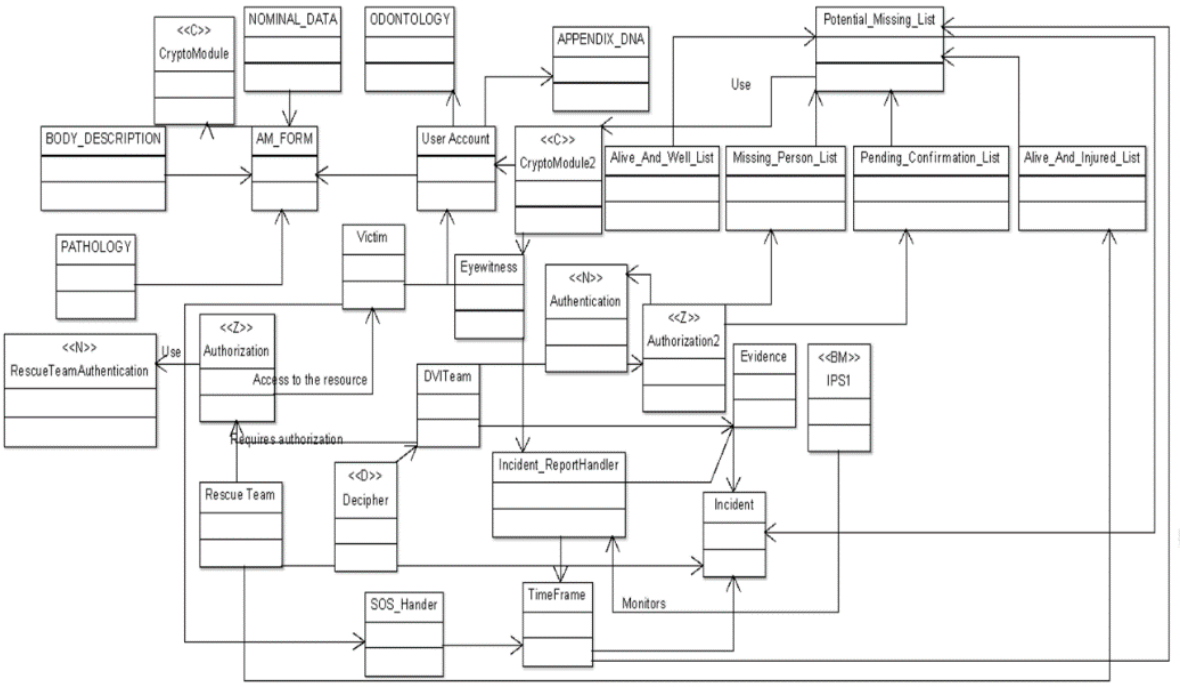

Fig. 11. IoTsecM class diagram for Dentify.Me App

Perhaps one of the domains where IoT is finding a niche of opportunity for success is in eHealth and mHealth applications. The mHealth model heavily depends on mobile devices to mobilize care delivery to reach patients whether in emergency medical services or home care. However, the risk of cyber-attacks directed at IoT mHealth applications can compromise the availability and integrity of patient information, crippling care mobility and sometimes threatening patients' lives if decisions are made based on invalid information.

Therefore, in this work, the approach referred to as IoTsecM has been used for mApp security modelling in mHealth. IoTsecM is a UML/SysML extension which models identified security controls against possible attacks to guarantee the existence of a security analysis and security mechanisms from the designing stages of an IoT system.

In this research, the IoTsecM approach helped to identify and represent the security and functional requirements in UML/SysML notation in the Dentify.Me mApp in the domain of disaster management. Threat modelling was followed which enabled, first, the identification of assets that were relevant elements for the system functionality and, therefore, targets for the attackers. Once the assets were identified and the system architecture was obtained, threat modelling was realized using attack trees to reveal the sequence of attacks targeted to a particular threat, and the results of such attacks and sub-attacks were identified as attack vectors, which are all the possible methods an attacker can use to reach the threat objective.

The functionality of IoTsecM helped to identify the security requirements in order to mitigate the possible attacks. Once the controls were identified and proposed, the IoTsecM profile allowed for the representation of such controls as security requirements within the system architecture and associations. The security requirements were considered in the analysis stage in this case study, allowing for the representation of the proposed security controls which will mitigate the attacks regarded in the attack trees. The security analysis must be considered a fundamental process when IoT systems involve sensitive information, since the consequence of the absence of security mechanisms would present huge risks to human lives.

IoTsecM represents a very useful tool which helps understand and consider the security of IoT ecosystem before it is implemented in physical objects, and the results of such consideration may lead to more powerful processors in nodes, the establishment of refined policies, data encryption and IDS or IPS implementation. These security controls and mechanisms undoubtedly change the system architecture, design and deployment, and due to the security mechanisms identified, both companies and developers can save both money and lives.

Finally, it is expected that the IoTsecM approach will facilitate the building of security awareness and consideration, specifically for IoT ecosystem, to address risks to IoT applications in digital sectors beyond healthcare. IoTsecM depicts and models security requirements and it is a UML/SysML extension, hence it is visual and therefore helps in terms of the conceptualization and representation of security requirements. In terms of other state of the art approaches, and to the authors' knowledge, there is no other UML/SysML extension, which covers all these aspects.

Acknowledgements

The authors from Instituto Politécnico Nacional would like to acknowledge financial support from CONACY and IPN-SIP under grants 1894 and 20180460; the authors from King Saud University extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this research through research group no. (RGP-1438-002).

References

1. **Escamilla-Ambrosio, P.J., Rodríguez-Mota, A., Aguirre-Anaya, E., Acosta-Bermejo, R., & Salinas-Rosales, M. (2018).** Distributing Computing in the Internet of Things: Cloud, Fog and *Edge Computing Overview NEO´16,* pp. 87–115, DOI: 10.1007/978-3-319-64063-1_4.

2. **Gartner (2017).** *Gartner Says 8.4 Billion Connected Things Will Be in Use in 2017, Up 31 Percent From 2016*. http://www.gartner.com/ newsroom/id/ 3598917.
3. **GSMA (2017).** *The Impact of the Internet of Things: The Connected Home.* https://www.gsma.com/ newsroom/wp-content/uploads/15625-Connected-Living-Report.
4. **Venkatramanan, P. & Rathina, I. (2014).** *Healthcare leveraging internet of things to revolutionize healthcare and wellness.* IT Services Business Solutions Consulting.
5. **Maksimović, M. (2018).** Implementation of Fog computing in IoT-based healthcare system. *JITA-Journal of Information Technology and Applications,* Vol. 14, No. 2, pp. 100–107. DOI: 10.7251/ JIT1702100M.
6. **Oh, H., Rizo, C., Enkin, M., & Jadad, A. (2005).** What Is eHealth (3): A Systematic Review of Published Definitions. *Journal of Medical Internet Research*, Vol. 7, No. 1. DOI: 10.2196/ jmir.7.1.e1.
7. **Eysenbach, G. (2001).** What is e-health?. *J Med Internet Research*, Vol. 3, No. 2, pp. 1–2. DOI: 10.2196/jmir.3.2.e20.
8. **Swiatek, P. & Rucinski, A. (2013).** IoT as a service system for eHealth. *e-Health Networking, Applications & Services (Healthcom´13) IEEE 15th International Conference,* pp. 81–84. DOI: 10.1109/HealthCom.2013.6720643.
9. **Farahani, B., Firouzi, F., Chang, V., Badaroglu, M., Constant, N., & Mankodiya, K. (2018).** Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. *Future Generation Computer Systems,* Vol. 78, No. 2, pp. 659–676. DOI: 10.1016/j.future. 2017.04.036.
10. **Kumar, S., Nilsen, W.J., Abernethy, A., Atienza, A., Patrick, K., Pavel, M., & Hedeker, D. (2013).** Mobile health technology evaluation: the mHealth evidence workshop. *American journal of preventive medicine*, Vol. 45, No. 2, pp. 228–236. DOI: 10.1016/j.amepre.2013.03.017.
11. **Papageorgiou, A., Strigkos, M., Politou, E., Alepis, E., Solanas, A., & Patsakis, C. (2018).** Security and Privacy Analysis of Mobile Health Applications: The Alarming State of Practice. *IEEE Access*, Vol. 6, pp. 9390–9403. DOI: 10.1109/ ACCESS.2018.2799522.
12. **Robles-Ramirez, D.A., Escamilla-Ambrosio, P.J., Acosta-Bermejo, R., Aguirre-Anaya, E., Rodríguez-Mota, A., & Reyes-Torres J.J. (2017).** IoTsec: UML extension for Internet of things systems security modelling. *Proceedings of the International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE2017),* pp. 151–156. DOI: 10.1109/ ICMEAE.2017.20.
13. **Robles-Ramírez, D.A., Escamilla-Ambrocio, P.J., & Tryfonas, T. (2018).** *IoTsecM: UML/SysML Extension for Internet of Things Security Modelling.* MSc Dissertation. Instituto Politécnico Nacional, Centro de Investigación en Computación, pp. 151–156. DOI: 10.1109/ICMEAE.2017.20.
14. **Powell, J. (2009).** *Integrating healthcare with ICT, in Integrating Healthcare with Information and Communications Technology.* W. Currie and D. Finnegan, Eds. Oxford: Radcliffe Publishing Ltd, ch. 4, pp. 85–94.
15. **Waegemann, C.P. (2016).** *mHealth: History, Analysis, and Implementation.* A. Moumtzoglou. Ed. IGI Global, pp. 1–19. DOI: 10.4018/978-1-4666-9861-1.ch001.
16. **Moumtzoglou, A.S. (2016).** *Tailored M-Health Communication in Patient-Centered Care.* A. Moumtzoglou, pp. 1–22. DOI: 10.4018/978-1-4666-9861-1.ch016.
17. **National Institutes of Health (2016).** *Who We Are — National Institutes of Health (NIH). http://* nih.gov/about-nih/who-we-are
18. **Koumpouros, Y. & Georgoulas, A. (2016).** mHealth R&D Activities in Europe. *M-Health Innovations for Patient- Centered Care,* pp. 20–51.
19. **Apple Support. (2016).** *About SOS on apple watch.* https://support.apple.com/en-us/HT206983.
20. **West, D.M. & Bleiberg**, **J. (2016).** *Five ways teachers can use technology to help students.* https://www.brookings.edu/research/how-mobile-devices-are-transforming-disaster-relief-and-public-safety/.
21. **Callaway, D.W., Peabody, C.R., Hoffman, A., & Cote, E. (2012).** *Disaster mobile health technology: Lessons from Haiti.* Cambridge University Press, Vol. 27, No. 2, pp. 148–152. DOI: 10.1017/ S1049023X12000441.
22. **Australian Law Reform Commission. (2016).** *Impact of developing technology on privacy*. http://www.alrc.gov.au/publications/9.%20Overview%3A%20Impact%20of%20Developing%20Technology%20on%20Privacy/location-detection-technologies.
23. **GPS.gov. (2011).** Public safety & disaster relief applications. http://www.gps.gov/applications/ safety/.
24. **Oxford Dictionaries. (2016).** Smartwatch https://en.oxforddictionaries.com/definition/smartwatch.

25. **Apple. Inc, "WatchOS + Apps". (2016).** http://developer.apple.com/watchos/.
26. **OWASP. (2018).** *SQL Injection Prevention Cheat Sheet.* https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet.
27. **Serbanati, A. & et al. (2012).** *Internet of Things Architecture, Concept and Solutions for Privacy and Security in the Resolution Infrastructure.* EU Proj. IoT-A, Proj. Rep. D4. 2, No. 257521.
28. **Babar, S., Mahalle, P., Stango, A., Prasad, N., & Prasad**, **R. (2010).** Proposed security model and threat taxonomy for the Internet of Things (IoT). *International Conference on Network Security and Applications,* Vol. 89, pp. 420**.**429. DOI: 10.1007 /978-3-642-14478-3_42.
29. **Lin, S.W., Miller, B., Durand, J., Bleakley, G., Chigani, A., Martin, R., Murphy, B., & Crawford, M. (2017).** The Industrial Internet of Things Volume G1: Reference Architecture. *Industrial Internet Consortium (IIC), Tech. Rep.*
30. **Russell, B. & Van-Duren, D. (2016).** *Practical Internet of Things Security.*
31. **Almutawaa, S.N., Alkabani, H.M., Alsmari, M.A., Alashgar, N.H., & Alrajeh**, **A.S. (2017).** *Dentify.Me App.*
32. **Amenaza**, http://www.amenaza.com/.
33. **Russell, B., & Van Duren, D. (2016).** *Practical Internet of Things Security.* Packt Publishing Ltd.
34. **AlQahtani, S., Alsalamah, S., Adserias-Garriga, J., Aschheim, K., Riley, A., Silva, R., & Nuzzolese, E. (2018).** Rescue the Living, Find the Missing, and Identify the Found: The Dentify.Me App. *Presented in: the 70th American Academy of Forensic Sciences (AAFS) Annual Scientific Meeting,* Vol. 19-24, pp. 569.

Article received on 02/12/2018; accepted on 28/05/2019.
Corresponding author is Ponciano J. Escamilla-Ambrosio.