UNIVERSIDADE DA BEIRA INTERIOR
Engenharia Informática

Identity Management and Authorization Infrastructure in Secure Mobile Access to Electronic Health Records

Final version after presentation

Pedro Miguel Freitas Moura

Dissertação para obtenção do Grau de Mestre
Engenharia Informática
(2º ciclo de estudos)

Orientador: Prof. Doutor Paulo Fazendeiro
Co-orientador: Prof. Doutora Ana Ferreira
Co-orientador: Prof. Doutor Pedro Inácio

Covilhã, Fevereiro de 2018

Acknowledgment

A sincere thanks to all who help me in the conclusion of this Master Thesis.

I would like to thank Professor Paulo Fazendeiro and Professor Pedro Inácio for taking me under their wing and mentor me with precious knowledge and wisdom. I learned a lot from them. Also for all the constant words of encouragement and for supervising my Master Thesis.

A thanks to Ana Ferreira for all the patience, kind support and scientific wisdom provided.

I thank my family because without them nothing in my life would have been possible. Lastly, I would like to thank all my friends for the patience and support.

Thank you all!

Abstract

We live in an age of the mobile paradigm of anytime/anywhere access, as the mobile device is the most ubiquitous device that people now hold. Due to their portability, availability, easy of use, communication, access and sharing of information within various domains and areas of our daily lives, the acceptance and adoption of these devices is still growing. However, due to their potential and raising numbers, mobile devices are a growing target for attackers and, like other technologies, mobile applications are still vulnerable.

Health information systems are composed with tools and software to collect, manage, analyze and process medical information (such as electronic health records and personal health records). Therefore, such systems can empower the performance and maintenance of health services, promoting availability, readability, accessibility and data sharing of vital information about a patients overall medical history, between geographic fragmented health services. Quick access to information presents a great importance in the health sector, as it accelerates work processes, resulting in better time utilization. Additionally, it may increase the quality of care. However health information systems store and manage highly sensitive data, which raises serious concerns regarding patients privacy and safety, and may explain the still increasing number of malicious incidents reports within the health domain.

Data related to health information systems are highly sensitive and subject to severe legal and regulatory restrictions, that aim to protect the individual rights and privacy of patients. Along side with these legislations, security requirements must be analyzed and measures implemented. Within the necessary security requirements to access health data, secure authentication, identity management and access control are essential to provide adequate means to protect data from unauthorized accesses. However, besides the use of simple authentication models, traditional access control models are commonly based on predefined access policies and roles, and are inflexible. This results in uniform access control decisions through people, different type of devices, environments and situational conditions, and across enterprises, location and time.

Although already existent models allow to ensure the needs of the health care systems, they still lack components for dynamicity and privacy protection, which leads to not have desire levels of security and to the patient not to have a full and easy control of his privacy. Within this master thesis, after a deep research and review of the stat of art, was published a novel dynamic access control model, Socio-Technical Risk-Adaptable Access Control modEl (SoTRAACE), which can model the inherent differences and security requirements that are present in this thesis. To do this, SoTRAACE aggregates attributes from various domains to help performing a risk assessment at the moment of the request. The assessment of the risk factors identified in this work is based in a Delphi Study. A set of security experts from various domains were selected, to classify the impact in the risk assessment of each attribute that SoTRAACE aggregates. SoTRAACE was integrated in an architecture with requirements well-founded, and based in the best recommendations and standards (OWASP, NIST 800-53, NIST 800-57), as well based in deep review of the state-of-art. The architecture is further targeted with the essential security analysis and the threat model.

As proof of concept, the proposed access control model was implemented within the user-centric architecture, with two mobile prototypes for several types of accesses by patients and health-care professionals, as well the web servers that handles the access requests, authentication and identity management.
The proof of concept shows that the model works as expected, with transparency, assuring privacy and data control to the user without impact for user experience and interaction. It is clear that the model can be extended to other industry domains, and new levels of risks or attributes can be added because it is modular. The architecture also works as expected, assuring secure authentication with multifactor, and secure data share/access based in SoTRAACE decisions. The communication channel that SoTRAACE uses was also protected with a digital certificate. At last, the architecture was tested within different Android versions, tested with static and dynamic analysis and with tests with security tools.
Future work includes the integration of health data standards and evaluating the proposed system by collecting users' opinion after releasing the system to real world.

Keywords

Access Control, Authentication, Cryptography, e-Health, m-Health, Mobile Computing, OWASP, Privacy, Risk Adaptable Access, Security.

Resumo

Hoje em dia vivemos em um paradigma móvel de acesso em qualquer lugar/hora, sendo que os dispositivos móveis são a tecnologia mais presente no dia a dia da sociedade. Devido à sua portabilidade, disponibilidade, fácil manuseamento, poder de comunicação, acesso e partilha de informação referentes a várias áreas e domínios das nossas vidas, a aceitação e integração destes dispositivos é cada vez maior. No entanto, devido ao seu potencial e aumento do número de utilizadores, os dispositivos móveis são cada vez mais alvos de ataques, e tal como outras tecnologias, aplicações móveis continuam a ser vulneráveis.

Sistemas de informação de saúde são compostos por ferramentas e softwares que permitem recolher, administrar, analisar e processar informação médica (tais como documentos de saúde eletrónicos). Portanto, tais sistemas podem potencializar a performance e a manutenção dos serviços de saúde, promovendo assim a disponibilidade, acessibilidade e a partilha de dados vitais referentes ao registro médico geral dos pacientes, entre serviços e instituições que estão geograficamente fragmentadas. O rápido acesso a informações médicas apresenta uma grande importância para o setor da saúde, dado que acelera os processos de trabalho, resultando assim numa melhor eficiência na utilização do tempo e recursos. Consequentemente haverá uma melhor qualidade de tratamento. Porém os sistemas de informação de saúde armazenam e manuseiam dados bastantes sensíveis, o que levanta sérias preocupações referentes à privacidade e segurança do paciente. Assim se explica o aumento de incidentes maliciosos dentro do domínio da saúde.

Os dados de saúde são altamente sensíveis e são sujeitos a severas leis e restrições regulamentares, que pretendem assegurar a proteção dos direitos e privacidade dos pacientes, salvaguardando os seus dados de saúde. Juntamente com estas legislações, requerimentos de segurança devem ser analisados e medidas implementadas. Dentro dos requerimentos necessários para aceder aos dados de saúde, uma autenticação segura, gestão de identidade e controlos de acesso são essenciais para fornecer meios adequados para a proteção de dados contra acessos não autorizados. No entanto, além do uso de modelos simples de autenticação, os modelos tradicionais de controlo de acesso são normalmente baseados em políticas de acesso e cargos pré-definidos, e são inflexíveis. Isto resulta em decisões de controlo de acesso uniformes para diferentes pessoas, tipos de dispositivo, ambientes e condições situacionais, empresas, localizações e diferentes alturas no tempo. Apesar dos modelos existentes permitirem assegurar algumas necessidades dos sistemas de saúde, ainda há escassez de componentes para acesso dinâmico e proteção de privacidade , o que resultam em níveis de segurança não satisfatórios e em o paciente não ter controlo directo e total sobre a sua privacidade e documentos de saúde. Dentro desta tese de mestrado, depois da investigação e revisão intensiva do estado da arte, foi publicado um modelo inovador de controlo de acesso, chamado SoTRAACE, que molda as diferenças de acesso inerentes e requerimentos de segurança presentes nesta tese. Para isto, o SoTRAACE agrega atributos de vários ambientes e domínios que ajudam a executar uma avaliação de riscos, no momento em que os dados são requisitados. A avaliação dos fatores de risco identificados neste trabalho são baseados num estudo de Delphi. Um conjunto de peritos de segurança de vários domínios industriais foram selecionados, para classificar o impacto de cada atributo que o SoTRAACE agrega. O SoTRAACE foi integrado numa arquitectura para acesso a dados médicos, com requerimentos bem fundados, baseados nas melhores normas e recomen-

dações (OWASP, NIST 800-53, NIST 800-57), e em revisões intensivas do estado da arte. Esta arquitectura é posteriormente alvo de uma análise de segurança e modelos de ataque.

Como prova deste conceito, o modelo de controlo de acesso proposto é implementado juntamente com uma arquitetura focada no utilizador, com dois protótipos para aplicações móveis, que providênciam vários tipos de acesso de pacientes e profissionais de saúde. A arquitetura é constituída também por servidores web que tratam da gestão de dados, controlo de acesso e autenticação e gestão de identidade. O resultado final mostra que o modelo funciona como esperado, com transparência, assegurando a privacidade e o controlo de dados para o utilizador, sem ter impacto na sua interação e experiência. Consequentemente este modelo pode-se extender para outros setores industriais, e novos níveis de risco ou atributos podem ser adicionados a este mesmo, por ser modular. A arquitetura também funciona como esperado, assegurando uma autenticação segura com multi-fator, acesso e partilha de dados segura baseado em decisões do SoTRAACE. O canal de comunicação que o SoTRAACE usa foi também protegido com um certificado digital.
A arquitectura foi testada em diferentes versões de Android, e foi alvo de análise estática, dinâmica e testes com ferramentas de segurança.
Para trabalho futuro está planeado a integração de normas de dados de saúde e a avaliação do sistema proposto, através da recolha de opiniões de utilizadores no mundo real.

Palavras Chave

Autenticação, Controlo de acesso, Criptografia, e-Health, m-Health, Computação Móvel, OWASP, Privacidade, Accesso Adaptavel ao Risco, Segurança.

Contents

List of Figures

List of Tables

Acronyms

AACP Adaptable Access Control Policy

ABAC Attribute Based Access Control

ABE Attribute Based Encryption

ACL Access Control Lists

AEAD Authenticated Encryption with Associated Data

AES Advanced Encryption Standard

API Application Programming Interface

AT Authorization Token

ATM Automated Teller Machine

BTG Break the Glass

CA Certificate Authority

CBC Cypher Block Chaining

CC Citizen Card

CHAP Challenge Handshake Authentication Protocol

CIA Confidentially Integrity Availability

CINTESIS Center for Health Technologies and Services Research

CNPD Comissão Nacional de Proteção Dados

COA Ciphertext-only attack

CRUD Create, Read, Update, Delete

CSPRNG Cryptographically Secure Pseudo-Random Number Generator

CTR Counter

DAC Discretionary Access Control

DoS Denial of Service

DDoS Distributed Denial of Service

DHE Diffie-Hellman Ephemeral

DNS Domain Name System

DREAD Damage, Reproducibility, Exploitability, Affected Users, Discoverability

EC Ecliptic Curve

eHealth Electronic Health

EHR Electronic Health Record

ENISA European Network and Information Security Agency

FISMA Federal Information Systems Management Act

GCM Galois/Counter Mode

GDPR General Data Protection Regulation

GIS Geographical Information Systems

GPS Global Positioning System

HIPPA Health Insurance Portability Accountability Act

HIS Health Information Systems

HL7 Health Level Seven

HMAC Hash Message Authentication Code

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

IA Information Accountability

IDE Integrated Development Environment

IdM Identity Management

IdP Identity Provider

IEC International Electrotechnical Commission

IEEE Institute of Electrical and Electronics Engineers

IMEI International Mobile Equipment Identity

IP Internet Protocol

ISO International Organization for Standardization

IT Information Technology

IV Initialization Vector

JAX-RS Java Application Programming Interface for Representational State Transfer Web Services

JDBC Java Database Connectivity

JDK Java Development Kit

JSON JavaScript Object Notation

JSR Java Specification Request

LAN Local Area Network

MAC Mandatory Access Control

MAuthC Message Authentication Code

mHealth Mobile Health

myHEnCE my HEalth Control Everywhere

NanoSTIMA Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics

NIST National Institute of Standards and Technology

OASIS Organization for the Advancement of Structured Information Standards

OFELIA Open Federated Environment for Leveraging of Identity and Authorization

OS Operating System

OTP One Time Password

OWASP Open Web Application Security Project

PHR Personal Health Record

PIN Personal Identification Number

PII Personal Identifying Information

PKI Public Key Infrastructure

PRNG Pseudo Random Number Generators

QR Quick Response

RAdAC Risk-Adaptable Access Control

RBAC Role Based Access Control

RC Rivest Cipher

ReBAC Relationship-Based Access Control

RDBMS Relational Database Management System

REST Representational State Transfer

RFC Request For Comments

RSA Rivest, Shamir, Adleman

RuleBAC Rule-based Access Control

SAML Security Assertion Markup Language

SDLC Software Development Life Cycle

SHA Secure Hash Algorithm

SitBAC Situation-based Access Control

SOAP Simple Object Access Protocol

SoTRAACE Socio-Technical Risk-Adaptable Access Control modEl

SP Service Provider

SQL Structured Query Language

SSID Service Set Identifier

SSL Secure Socket Layer

SSO Single Sign-On

STRIDE Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of privilege

TMAC Team-based Access Control

TLS Transport Layer Security

TRBAC Temporal Role-Based Access Control

UAP User Activity Profile

UML Unified Modeling Language

URL Uniform Resource Locator

URI Uniform Resource Identifier

WPA Wi-Fi Protected Access

WSDL Web Services Description Language

XACML eXtensible Access Control Markup Language

XML eXtensible Markup Language

XSS Cross-Site Scripting

Chapter 1

Introduction

1.1 Problem Definition and Motivation

Smartphones and Internet are ubiquitous emerging technologies, present everyday in humans lives through handheld devices, desktops or machine-to-machine communications which give us the possibility of access information, communication, shopping, view email and social networks, etc. Organizations or single persons depends on these technologies to perform their daily tasks, to stay connected to everyone and everything, everywhere. Companies are migrating from paper to digital, to centralized, hosting their data and information online, to be accessible by all the respective authorized. This road lead to in 2015, a record-setting total of nine megabreaches, and the reported number of private exposed identities reached to 429 million [1]. In some cases, users or employee's private information can be captured by a malicious adversary or other illegal users, and they can analyze the obtained information then leading to inestimable loss and threat.

Internet is widely and globally used, the massive growth brings enormous and varied services, that help in the evolution of the world and in the improvement of the quality of life. Because most services are unrelated, users need to have a separate identification and different accounts for each service, and personal information stored and fragmented by different operators and services, in different places. However, use and explore the options and enormous value of Internet services (e.g. communication, shopping, cloud services, big data) can rapidly increase risks of security as well as the lack of privacy. Analyzing and tracking information on Internet services become an interest for companies, who aim to gather, store, share and reuse our personal data without users authorization to get profit. If we look closer to some frequently used services, Amazon monitors ours shopping preferences, Google knows our browsing habits and visited pages and Facebook catches all information about us, from location, friendship and photos to private talks and real time videos [2]. Besides that mobile operators also record with who, when and what we talk. From health questions to shopping habits, our web life and search history contains some of the most personal information that search technology giants such as Google, Facebook and Bing carefully store, to share with advertisers, publicity companies and the government [3]. Many users don't give importance to privacy, many may say they has nothing to hide. But if confronted with the possibilities of leak all their emails, chat talks, etc, to the Internet, they may consider being more careful [3]. Companies and scientists can no longer guarantee total privacy. With the growth of privacy-aware conscious users, privacy and security needs to be a target as primordial and primary themes for investment and research, with the objective of empower privacy, anonymization and security base requirements: integrity, availability and confidentiality [4].

A fundamental aspect of modern information society is security which has now become more than ever an asset of great importance in almost every area. Nowadays confidentiality and privacy of computational frameworks, along with high-level authentication and authorization, are

considered essential assets of any system. In Electronic Health (eHealth) security is essential to assure privacy, Confidentially Integrity Availability (CIA), along side with proper authentication and access control. Along side with these, data storage and protections, secure communication and digital certificates are key concerns that need to fulfilled. The need for protocols which secure the distribution of data while protecting the privacy of users and the reliability of results lead to the development of cryptographic protocols to address these issues.

Health care institutions in the past decade are taking the same road of companies, with the recursive replacement of paper-based health systems for electronic-based records such as EHR or PHR [5, 6], supported by Health Information Systems (HIS). Both EHR and HIS are components of eHealth. HIS are tools and software to collect, manage, analyze and process medical information (such as EHR and PHR), empowering the performance and maintenance of health services [7, 6]. Beside these, HIS must transmit information securely and assure accessible information for multiple authorized users.

HIS empower availability, readability and accessibility and promote data sharing of vital information about a patient's overall medical history between geographic fragmented health services (pharmacies, hospitals, local health centers) [8, 9]. There are legislation available in the health sector, such as Health Insurance Portability Accountability Act (HIPPA) [10, 11] in America legislation, Recommendation No. R (97) 5 in European legislation [12], Regulation 2016/679 of European congress [13] and Portuguese Comissão Nacional de Proteção Dados (CNPD) to personal genetic information and health information in Law nº 12/2005 [14]. These strict legislations enforce rules for the privacy and security of patient's health information.

In eHealth HIS systems must generate event logs that give us basic data about the 'who?', 'what?' and 'when?' aspects of information use [15]. However, the 'why?' (situation, context, purpose) aspect is much harder to determine and is excluded from the log systems. And with emerging of mobile technologies the 'where?' and 'which device?' questions are also important to record and analyze. Information sharing in healthcare service delivery demands privacy management through Information Accountability (IA), which refers to holding the users answerable for their accesses, actions and the ramifications of those actions [15]. Through this, with an intelligent HIS and IA, new access control restrictions can be dynamically applied, attacks can be prevented and better decision can be provided.

With the exponential growth of HIS, security threats has increased significantly in recent years [16, 6]. HIS enhance the performance and maintenance of health services [6, 7] but their storage of highly sensitive data raises serious concerns regarding patient' privacy and safety [17]. Even though health data are subjected to legal and regulatory restrictions [10, 12, 13], according to [1], in 2015, 39% of all data breaches that occurred within the Services sector were attributed to Healthcare. Therefore, storing health information in electronic form raises concerns about patients health, privacy and safety [17]. The collaboration of informatics and healthcare professionals is also a must for dealing with all the security and privacy concerns cases and handling the deluge of data. The reliability and availability of HIS is also important, people need to trust in these systems otherwise will not use them.

The most ubiquitous device that people now hold is the smartphone. The acceptance and adoption of these devices is growing due to their portability, availability and improved ease of use [18]. HIS are widely including smartphones in core functions and tasks. Health professionals can use smartphones to access patient records (e.g EHR), to view exam results, to share and ask for second opinion diagnosis, and to prescribe medications [17]. The other end user, respectively the patient, can use smartphones to access and update their medical records, to control access

to their medical records, to monitor their health statistics and to view their prescriptions [19]. However, mobile applications are vulnerable. They are deliberately programmed with privacy leaks of information for track position and target advertising [20]. In 2017 the Check Point Mobile Threat Prevention has detected a severe infection in 36 Android devices models, the malware was not downloaded to the device as a result of the users use, it was already present on the devices before the users received them. This means it comes installed from the production source or reseller point. The malicious apps were not part of the official ROM supplied by the vendor, and were added somewhere along the supply chain [21].
Google report Micro-moments [22] proved this with many interesting facts about the way people use their mobile devices. In this research, Google said that 68% of phone users say they check their phone within 15 min of waking up and 87% always has their smartphone at their side, day, and night.
The worldwide smartphone market grew 1.1% year over year, and in middle of 2016 reach 363.2 million shipments. Table 1.1 presentes the values of Operating System (OS) smartphone global market share [23]:

Period	*Android*	*iOS*	*Windows Phone*	*Other*
2015 Q4	79.6%	18.7%	1.2%	0.5%
2016 Q1	83.5%	15.4%	0.8%	0.4%
2016 Q2	87.6%	11.7%	0.4%	0.3%
2016 Q3	86.8%	12.5%	0.3%	0.4%

Table 1.1: OS Smartphone Global Market Share 2015Q4-2016Q3.

This domain in the mobile market happens because Android is open-source, provides the most cheap devices and software in the market, with an enormous amount of applications which are developed in the most used programming language Java and is internal system is Linux-based.

The right balance between transparency and privacy at the point and purpose of need is required, as this is believed to be a precondition for service improvement, data quality and productivity and a powerful driver for auditing and accountability in healthcare [24]. Transparency is about verifying that both security measures and privacy are in place and if any violations have occurred [24]. Traditional access controls are mostly based on predefined access policies and roles. With the adoption of mobile devices there is a need to search more innovative, original, flexible, adaptive, transparent and more resilient access control models, that are required for more heterogeneous requests [25]. Access control, is only reachable after a reliable and secure authentication phase. Authentication is the process of verification that an individual, entity or website is who it claims to be and it is a crucial phase in all services to reach privacy and security. To do this there is the need to provide the individual with a trusted digital identity composed of a set of personal data attributes that can be used to characterize a user.
There is a need to explore existing problems and innovate new paths to ensure reliability, transparency, privacy, confidentiality, availability and integrity in HISs and eHealth that needs to integrate and resolve some the big challenges of accessing centralized shared big data:

- Secure, standard and centralized authentications methods, with an identity manager validating end-user authentication from mobile devices.

- Fine-grained dynamic access control, which varies and takes different access options depending on non-static environmental variables, system variables and users behaviour.

- Practical and available data usage and purpose-specific privacy policy specification and compliance.
- Interfaces that are socio-technically intuitive, usable and secure transparency.
- Provide data anonymization and information protection.
- Use standards for centralized data storage.

1.2 Research Questions

During the elaboration of this work the following research questions were used as main guidelines for the study and developments efforts:

- Can users/patients private information be kept as secure as possible during rest and transmission, and keep it under their direct control?
- How much information is needed to provide dynamic access control at the point and for the purpose of need, while guaranteeing the necessary balance between transparency and privacy?
- What are the most common access control models? And what are the best and most used methods for authentication and identity management protocols?
- Which environmental variables (e.g. type of wireless connection, Global Positioning System (GPS) location) and factors of human behavior (e.g. denied of services times, wrong password) can be delegated to the behaviour and permissions of the access control middleware?
- How to provide central identification and authorization without compromise users credentials under each request to different institutions and data? How to define a secure cryptographic token to identify user in data request through all institutions? How to keep the anonymity of the patient data?
- What are the challenges, requirements and threats on eHealth? What are the best techniques, measures and cryptographic algorithms to assure secure authentication, dynamic access control and CIA in eHealth? And how to use them to fulfill the legislations?
- What are the security requirements to reach the high level security in eHealth? What are the best recommendations, algorithms and techniques?

1.3 Objectives

Systems technically validated as secure against software attacks may still be unsecure and vulnerable against non-technical attacks (e.g, social engineering, phishing). Such failures are common since humans do not perceive security as a primary goal [25] and do not properly assess risks when using computer systems. There is a need for socio-technical security analysis framework to automatically detect and test human behaviour, environmental variables and technical interactions to identify those with vulnerabilities, which can lead to possible attacks and exploitations. Techniques of access control and authentication are most popular to protect unauthorized accessing from attackers.

The policies and legislations applied to health data imply strict access controls from the institution itself (e.g. roles, team, task, context) but also from the point of the individual owner of that data or accessing it for providing the healthcare service. Also the health data regulations implies that the owner manage the access control and operations over their related information. So it is implicit the need to provide the individual with a trusted digital identity composed of a set of personal data attributes that can be used to characterize a user.

One of the objectives is to research a decentralized privacy and user-centric model for identity management and access control on the aggregation of users private health data, distributed and protected by an IdP server, whose access is mediated by the persons smartphone. Other objective is to reach the high level of security. To do this, a system threat analysis needs to be performed, to evaluate security requirements in eHealth and research/implement cryptographic mechanisms, data storage protection and provide secure communication.
So in this work it is proposed a real time user-centric aggregation implemented within smartphones which are currently recognized as essential for enhancing security and privacy as well as providing flexible user-centric architectures, where it is vital to employ stronger cryptographic and security mechanisms. This study will focus on providing innovative, reliable, secure and centralized authentication and a new transparent, dynamic and novel access control model that can integrate adaptability and privacy together with behaviour and contextual attributes at the point and purpose of need.
In resume, the objectives for this research are to:

- Research and analyze existing access control models and define a new dynamic access control based on environmental variables(e.g. type of wireless connection, type of wireless encryption, GPS location), factors of human behavior (e.g. denial of services times, wrong password), type and sensitivity of resources (video, figure, text, very private) and system/intuitions predefined rules, with delegated access thought relationship based (related health professional, family, etc) and with break the glass methodology. Implement the new access control model.
- Research and analyze methods and protocols to perform central authentication and identity management. Define and implement a central secure authentication architecture with multifactor authentication.
- Define a set of SPs, an IdP, two android application (for patient and for health professionals), and model the respective HIS to create, view and for request permissions to acquire temporary/conditional access to a specific users identity attributes or EHR.
- Deeply review the challenges, requirements and threats on eHealth.
- Define and implement means that allow patients to access their EHR, allowing patients to customize access control rules and take full responsibility and governance over their health information .
- Study anonymization techniques can be used in eHealth.
- Perform threat analysis and the attack model over the defined system.
- Research and implement the recommended security protocols and cryptographic techniques for achieve data protection, CIA, reliable communication, secure authentication and dynamic authorization.

- Test and deploy the defined model.

1.4 Main Contributions

The present work reports, sometimes in a synthesized way due to space constraints, the following main contributions:

- The main scientific contribution is a novel access control model, namely SoTRAACE. SoTRAACE [26] was published and presented at the Institute of Electrical and Electronics Engineers (IEEE) 51st International Carnahan Conference on Security Technology, in Madrid in October 2017. SoTRAACE was also presented in a conference Macro-to-Nano Human Sensing: Towards Integrated Multimodal Health Monitoring and Analytics (NanoSTIMA) 3.5, in Porto in May 2017.
- The risk evaluation that SoTRAACE performs, is based in a conducted Delphi study within this master thesis, is also a contribution for the state of the art.
- A innovative HIS composed by web servers and two mobile applications, that enables user's in the health sector to view, share and manage health data.
- A literature review on security analysis in eHealth whose findings and recommendations are also valuable for different areas of application.

1.5 Document Organization

The remaining of this master thesis is structured as follows:

- Chapter 2 : Presents a review and the state of the art of security definitions and vulnerabilities, authentication and identity management, access control and eHealth.
- Chapter 3 : Presents the main scientific contribution, a novel access control model named SoTRAACE.
- Chapter 4 : Describes the architecture of the implemented HIS framework (mobile applications, web services and database). Outlines requirements and the respective essential diagrams. Describes how SoTRAACE it is integrated in the framework.
- Chapter 5 : Contains the security analysis of framework and discussions, and the respective measures. Also presents the details of the implemented security measures to fulfill the security analysis and requirements. Contains more details about SoTRAACE and his risk evaluation method, based in a Delphi study.
- Chapter 6 : Introduces features of the technologies used to implement the framework. Also contain implementation details, a demonstration of mobile application prototype and web service, and the final tests.
- Chapter 7 : Discussions, conclusions and future work.

Chapter 2

State of the Art

2.1 Introduction

Nowadays every person, entity or company uses computer software to execute their main tasks. So it is in the user's interest that the software is correctly developed, totally functional, without bugs and secure. The methodology within the Software Development Life Cycle (SDLC) process can vary across organizations and personal development necessities, but standards such as International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 12207 [27] represent processes that establish a life cycle for software, and provide a standard for building and maintaining software.
The intent of a SDLC process it to ensure that good high quality and cost-efficient software is developed. There are various SDLC models, a recent list can be found here [28]. In general all models include the following hierarchic steps:

1. Planning and Requirement Analysis.
2. System Design.
3. Implementation and Coding.
4. Testing and Documenting.
5. Deployment and Maintenance.

In the steps enumerated above it is not well-marked where to include the security modeling. Security takes time, cost and performance to implement and maintain. Because, in theory, security is not essential on the road of development for a functional software, is often ends up being excluded from the SDLC. The lack of time to release the software to the market or to the client, and lack of resources for the project are the main causes for the exclusion of the security analysis, measures and operations in the SDLC [29]. In the vulnerable software world that we live in this can lead to enormous consequences, which we will get to later.

But first, is important to define security in common words. In the Oxford English dictionary is present two good definitions for the term security [30]:

"The state of being free from danger or threats."

"The protection of a person, building, organization, or country against threats such as crime or attacks."

Cambridge English dictionary provides a more specific definition (slightly adapted in the scope of this work) for security related to Internet and Information Technology (IT) [31]:

- *Security is the protection of information and resources (physical or virtual) against threats, illegal access, being stolen or used wrongly or illegally.*

Security and threats are directly related. Security exists to prevent the negative and malicious effects of the threats. Threats try to take advantage of system malfunctions/imperfections and security vulnerabilities to complete their malicious and dangerous intentions. In technology, threat can be defined as an event or method that can potentially cause the theft, destruction, corruption, or denial (of use) of either service, information, resources, or materials [32].

Another important concept that needs to be defined is privacy [33]:

"... not having things known about you that you don't choose to have known, or at least you know that they are known and by whom."

In general, privacy is constructed by the social world, because without other people or entities, there isn't the need for protecting or hiding. Essentially is a person or entity desire to control access to his or her personal information, to be seen somewhere, to be followed nowhere, etc [5]. The privacy preservation problem is present in all countries, and has a big effect on human life and institutions, as it touches upon social, cultural, economic, and political aspects. Privacy is influenced by legislation and legal changes (e.g.[10, 12, 13]), such as the right of free speech [34], changes in technology, rules inside communities, changes in journalistic practice and country government rules.

The rest of this chapter presents a review of the state of the art of the thesis main topics of research, namely security vulnerabilities and threats, authentication and identity management and access control. Also a deep review about the the actual state and the important concepts of eHealth, which is the sector of this study.

2.2 Security Definitions and Vulnerabilities in IT

2.2.1 CIA Triad

There are many global standards for deploy security in SDLC, defined by ISO and IEC, present in ISO/IEC 27000 family - Information security management systems [35]. ISO standards are globally accepted and their value recognized in all markets. The ISO standards brings benefits to organizations business processes, leads to improving performance, reducing business risk and help organizations to becoming more sustainable and encourage innovation [36].
A more basic and fundamental principle in security is the CIA triad. The CIA triad is a model for security policy development and maintenance, worldwide used to identify problem areas and important solutions for information security. It's the heart of information security. In brief words, CIA triad ensures that only authorized users (confidentiality) have access to accurate and complete information (integrity) when required (availability)[37].
The importance of CIA is globally recognized. In America, in order to secure the information systems, the American government has issued a series of documents as part of the National Institute of Standards and Technology (NIST) risk management framework [38], including Federal Information Systems Management Act (FISMA) of 2002 [39], to enforce the use of CIA attributes in the establishment and maintenance of security controls [4]. These attributes are further connected to specific security control selection in NIST 800-53, named Recommended Security Controls for Federal Information Systems [4, 40].

- Confidentiality

FISMA defines confidentiality as [39]:
"Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information…" [44 U.S.C., Sec. 3542].
Confidentiality is related with the principle of least privilege. This principle encourages system designers and developers to allow running code only if the permissions needed to complete the required tasks and no more, and states that access to information, documents, assets, etc, should be granted only on a need to know basis, so that information which is only available to some should not be accessible by everyone [41]. The main cryptographic function that supports confidentiality is encryption. Encryption transforms plain text into cipher text which cannot be read easily. A loss of confidentiality can be defined as unauthorized revelation of information.

- Integrity

FISMA defines integrity as [39]:
"Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity…" [44 U.S.C., Sec. 3542].
Integrity ensures that information travelling through a communication channel, from a source to a destiny, is not changed. To fortify information integrity, the information can only be changed by authorized accesses. Defined rules for information manipulation and force access control policies are an import for assure internal integrity. Cryptographic hash functions are used to verify integrity, where in a hash of a particular set of data is calculated before transit and is sent along with the original message [42]. At the destiny side, the hash of the received message is computed and compared with the hash received. If both hashes are different, it means that the message has lost its value. Otherwise message integrity is confirmed. A loss of integrity can be defined as an unauthorized or unexpected modification or destruction of information.

- Availability

FISMA defines Availability as [39]: "Ensuring timely and reliable access to and use of information…" [44 U.S.C., SEC. 3542]. Availability enforces that the services and information of an organization are available when requested by an authorized party or service. Availability disruption can occur as a side effect of some problems, such as poor exception management, buffer overflows, too much request/access in simultaneous, human error, command injection, and other coding mistakes. Also environment variables can lead to loss of availability, such as earthquake or floods. Denying access to information is a very common attack nowadays. For example, websites can be taken down by Distributed Denial of Service (DDoS) attacks. The primary aim of DDoS attacks [43] is to deny users of the website access to the resources of the website. Such downtime can be very costly in some companies. For example, in 2000 Yahoo received a DDoS attack and was down for 3 hours, leading to US$500k in losses[44].

2.2.2 Vulnerabilities in IT

In the past decade, computer and internet networks have grown at an enormous rate. Organizations are building networks with larger scales, and need access to information, databases, reports and means of communication with the outside of the organizations. Therefore connec-

tivity with the global internet has become indispensable. Along with this network growth, has come an explosion in the use of computer and internet networks as a means of illicit access to computer systems. Nowadays, the intruders are everywhere, and they are constantly trying to gain entry into remote computer systems, capture passwords and sensitive information. The current internet environment is vulnerable to various attacks such as replay attack, guessing attack, modification attack, stolen- verifier attack, identity theft and denial-of-service attack [45]. Is not defined where SDLC include security modeling or security scans. Security takes time, it is expensive and performance costs tasks, that need to be done by a security specialist. Companies only focus in delivering functional software, and achieving short deadlines. At the end of the SDLC, given the lack of resources, companies not include security in their projects. Other main reasons [46] for the exclusion and lack of importance given to security are present in 2.1 (figure taken from [46]).

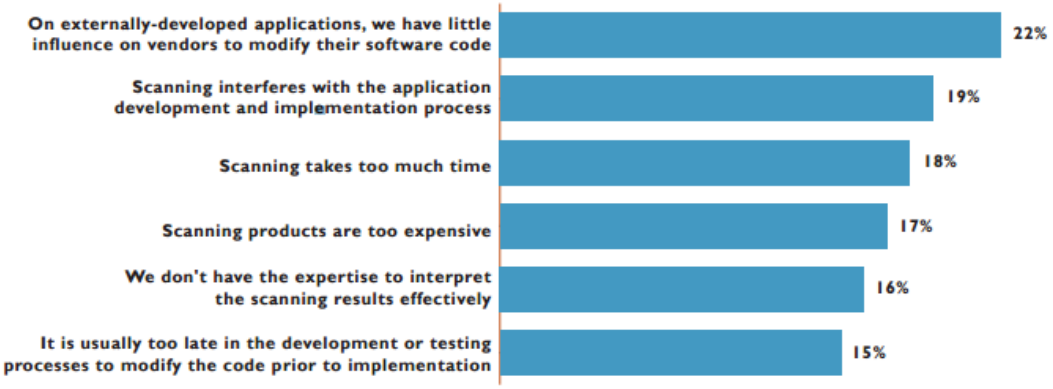

Figure 2.1: Reasons for not including application security scans.

Information security professionals are always searching for weaknesses and vulnerabilities in the system, to prevent attackers to gain access for malicious information manipulation or theft. According to 2015 survey [46], the security concerns scale are present in 2.2 (figure taken from [46]).

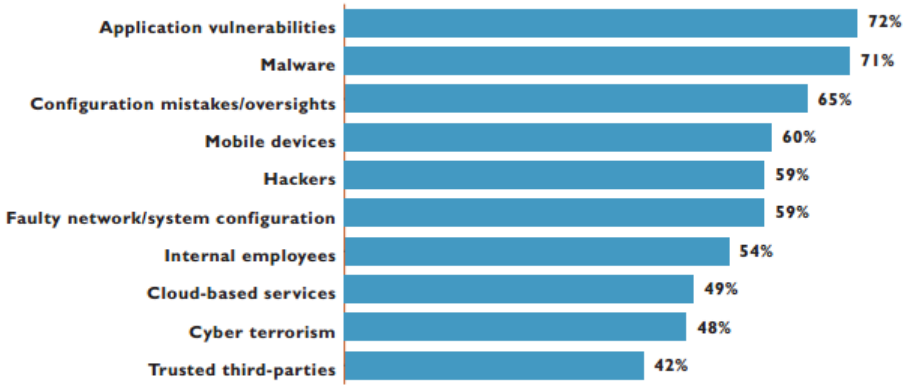

Figure 2.2: Top 10 security concerns.

Symantec Internet Security Threat Report, reports that in 2014 the leak of personal financial information scaled from 17.8% to 35.5%, where the username & password exposure was 13%

[47]. Considering that most common users maybe use same username and password for multiple systems and platforms, this is a critical issue.
Another related issue is the use of weak passwords or single factor authentication. According to [48], weak passwords or single factor authentication are a big vulnerability in most visited sites, such as social networking. Complex and hybrid passwords (numbers, upper and lower case letters and characters) are the first approach. The second is the use of multifactor authentication, which currently seems to be a good solution. In [49] is present a good tutorial for reach password account security.

There is nothing 100% safe. Where there is a lock there is also a key to open it. Exhaustive tries with small hooks can work or in extreme cases it can be broken down with brute force. Attackers are patient, fault tolerant and have all the time in the world to reach their objectives. They study all the possibilities and details in a system (Domain Name System (DNS), Internet Protocol (IP), OS, running services, versions of services, etc) before proceeding to the attack. Abraham Lincoln once said : "If I had eight hours to chop down a tree, I'd spend six hours sharpening my ax." Translating to our context, before proceeding to an attack, analyze and recognize all details in the environment. This is the way that attackers think.

To avoid the attacking attempts it is essential to know the most common errors and vulnerabilities. The Open Web Application Security Project (OWASP) [50] is a non-profit organization dedicated to providing unbiased, practical information about application security. OWASP is globally recognized and supports security agencies, countries, National & International legislation, standards, guidelines, committees and industry [51]. The OWASP Top 10 represents a broad consensus on the most critical web application security flaws. The errors on this list occur frequently in web applications, are often easy to find, and easy to exploit. They are dangerous because they will frequently allow attackers to completely take over your software, steal data, or prevent your software from working at all. In table 2.1 is present a list with the top 10 web vulnerabilities in 2017 [52] done by OWASP.

The software development habits and SDLC in industry and companies, seems to not invest and change in order to block these vulnerabilities forever. Besides that, in 2015, the number of zero-day vulnerabilities discovered raised to 54, a 125% increase from 2014 [53]. Thus, in average a new zero-day vulnerability was found every week in 2015. These numbers show that although the vulnerabilities are almost the same, non-reliable and insecure software development allows attackers to exploit system failures and develop new attacks. Zero-day vulnerabilities, and other malwares, are used to create ransomware, a particularly nasty type of malware that encrypts data or blocks access to a computer or data, and demands money to release it. In 2017 WannaCry ransomware emerged from a vulnerability first revealed to the public as part of a leaked stash of American National Security Agency related documents in order to infect Windows devices and encrypt their contents, before demanding payments of substantial amounts of money for the key to decrypt files. More than 400,000 machines infected and 98 percent of victims were using Windows 7, and it had taken root in 150 countries.

In recent year the smartphone appears. According to Statista, the overall number of mobile phone users reached 4.43 billion in 2015. The statistic for 2017 is that the number of smartphone users is forecast to reach 4.77 billion. So the smartphone penetration is fore-casted to continue to grow, expecting to reach 67% by 2019 [54]. And according to StatCounter, 37% of

OWASP Top 10 Vulnerabilities 2017	
1 - Injection	Injection flaws, such as Structured Query Language (SQL) or OS, occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing data without proper authorization.
2 - Broken Authentication and Session Management	Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, or session tokens, or to exploit other implementation flaws to assume other users identities
3 - Cross-Site Scripting (XSS)	XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.
4 - Broken Access Control	Restrictions on what authenticated users are allowed to do are not properly enforced. Attackers can exploit these flaws to access unauthorized functionality and/or data, such as access other users accounts, view and modify sensitive files, change access rights, etc.
5 - Security Misconfiguration	Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform.
6 - Sensitive Data Exposure	Many web applications and Application Programming Interface (API)s do not properly protect sensitive data, such as financial, health, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct credit card fraud or identity theft
7 - Insufficient Attack Protection	The majority of applications and APIs lack the basic ability to detect, prevent, and respond to both manual and automated attacks. Attack protection goes far beyond basic input validation and involves automatically detecting, logging, responding, and even blocking exploit attempts.
8 - Cross-Site Request Forgery	An attack forces a logged-on victim's browser to send a forged Hypertext Transfer Protocol (HTTP) request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application.
9 - Using Components with Known Vulnerabilities	Components, such as libraries, frameworks, and other software modules, almost always run with full privileges. If a vulnerable component is exploited, such an attack can facilitate serious data loss or server takeover.
10 - Underprotected API	Modern applications often involve rich client applications and APIs, such as JavaScript in the browser and mobile apps, that connect to any kind of API. These APIs are often unprotected and contain numerous vulnerabilities.

Table 2.1: Top 10 2017 Vulnerabilities

website visits in 2015 were generated by mobile web browsers [55]. Moreover, as refereed in [56] mobile data was expected to growth annually with a rate of 60% , reaching 25 exabytes per month in 2020.

The smartphones are always close to each person, are transported everywhere, allowing access to the Internet anywhere and anytime. With this potential, smartphones are a growing and tasty target for the attackers. OWASP produced a list of the mobile Top 10 vulnerabilities until 2016 [57], present in table 2.2.

OWASP Top 10 Mobile Vulnerabilities 2016	
M1 - Improper Platform Usage	This category covers misuse of a platform feature or failure to use platform security controls. It might include Android intents, platform permissions, misuse of TouchID, the Keychain, or some other security control that is part of the mobile operating system.
M2 - Insecure Data Storage	M2 + M4 from Mobile Top Ten 2014. This covers insecure data storage and unintended data leakage.
M3 - Insecure Communication	This covers poor handshaking, incorrect TLS or Secure Socket Layer (SSL) versions, weak negotiation, cleartext communication of sensitive assets, etc.
M4 - Insecure Authentication	This category captures notions of authenticating the end user or bad session management. This can include: - Failing to identify the user at all when that should be required - Failure to maintain the users identity when it is required
M5 -Insufficient Cryptography	The code applies cryptography to a sensitive information asset. However, the cryptography is insufficient in some way. This category is for issue where cryptography was attempted, but it wasn't done correctly.
M6 - Insecure Authorization	This is a category to capture any failures in authorization (e.g., authorization decisions in the client side, bad access control forced browsing, etc).
M7 - Client Code Quality	This was the "Security Decisions Via Untrusted Inputs". This would be the catch-all for code-level implementation problems in the mobile client. This would capture things like buffer overflows, format string vulnerabilities, and various other code-level mistakes where the solution is to rewrite some code that's running on the mobile device.
M8 - Code Tampering	This category covers binary patching, local resource modification, method hooking, method swizzling, and dynamic memory modification. Once the application is delivered to the mobile device, the code and data resources are resident there. An attacker can either directly modify the code, change the contents of memory dynamically, change or replace the system APIs that the application uses, or modify the application's data and resources.
M9 - Reverse Engineering	This category includes analysis of the final core binary to determine its source code, libraries, algorithms, and other assets.
M10 - Extraneous Functionality	Often, developers include hidden backdoor functionality or other internal development security controls that are not intended to be released into a production environment. For example, a developer may accidentally include a password as a comment in a hybrid app.

Table 2.2: Top 10 Mobile Vulnerabilities 2016.

Many enterprises has already understood the importance of security by design and are willing to invest more and more resources in security-related projects. The market demand for experts with a security and privacy background both in academia and industry has grown drastically [58].

Applications without security architecture are as bridges constructed without finite element analysis and wind tunnel testing. The need for application security in the form of security architecture is every bit as great as in building or bridge construction [59]. Application architects are responsible for constructing their design to adequately cover risks from both typical usage, and from extreme attack. Security is now expected, not an expensive add-on or simply left out. Security architecture refers to the fundamental pillars: the application must provide controls to protect the confidentiality of information, integrity of data, and provide access to the data when it is required, and only to the right users. The best system architecture designs and detailed design documents contain security discussion in each and every feature, how the risks are going to be mitigated, and what was actually done during coding. Security architecture starts on the day the business requirements are modeled, and never finishes until the last copy of your application is decommissioned. Security is a life-long process, not an one shot accident [59]. Some examples of security by default principles are minimize attack surface, principle of Least privilege area and separation of duties [59].

2.3 Authentication and Identity Management

2.3.1 Authentication

Authentication is usually the first measure to protect a secure communication, which could exclude malicious adversaries from a communication system. Authentication is the process of reliably identifying subjects by securely associating an identifier and its authenticator [60]. Its a verification that an entity is who/what it claims to be using a password, physical such as an authentication card, biometrics such as a fingerprint, eye iris, or distinctive behavior such as a gesture pattern on a touchscreen. To implement this phase, some specifications are essential to pay attention. The password complexity, password storage, how password are transmitted in a channel, etc. OWASP authentication specifications [61] are excellent, based on NIST and ISO, and provide good guidelines to help build an effective authentication phase. Also follow legislations rules its essential. For instance [10] enforces that each user must have a different identifier number. The most used attacks in the authentication phase are brute force attacks, replay attacks and man in the middle. In a brute force attack, if an attacker is able to guess passwords without the account becoming disabled due to failed authentication attempts, the attacker has an opportunity to continue with a brute force attack until the account is compromised. Automating brute-force/password guessing attacks on web applications is a trivial challenge. Password lockout mechanisms should be employed that lock out an account if more than a preset number of unsuccessful login attempts are made. Password lockout mechanisms have a logical weakness. An attacker that undertakes a large number of authentication attempts on known account names can produce a result that locks out entire blocks of user accounts. Given that the intent of a password lockout system is to protect from brute-force attacks, a sensible strategy is to lockout accounts for a period of time (e.g., 20 minutes). This significantly slows down attackers, while allowing the accounts to reopen automatically for legitimate users. Also, multifactor authentication is a very powerful deterrent when trying to prevent brute force at-

tacks since the credentials are a moving target. When multifactor is implemented and active, account lockout may no longer be necessary [61].
The most used types of multifactor authentication:

- Something you know : Personal Identification Number (PIN), password, passphare, security question.
- Something you have : Phone, credit card
- Something you are : Biometric, fingerprint, blood sample.

In case of replay attacks, nowadays becomes easy to capture and replay passwords commonly used to authenticate users. Many Internet protocols send their passwords in clear text, anyone who can read network traffic can gain access to whatever is protected by clear text passwords. Attackers can use network management tools to sniff packets to discover clear text passwords, thereby gaining unauthorized access to systems using clear text reusable passwords [62]. Or even capture the same hash value or cryptogram in the authentication phase, that is sent any time that the user authenticates. If the authentication always generates the same cryptogram an attacker has more chances to perform a successful Ciphertext-only attack (COA). This task becomes easier with the emerging of software to network monitoring and analysis [63]. Challenge Handshake Authentication Protocol (CHAP) and Lamport scheme are widely used protocols that ensures that a replay attack cant be performed. Both provide authentication using One Time Password (OTP).
Lamport brought up the first OTP authentication scheme using one-way hash function [64], after that there have been many subsequent researches. The Lamport scheme uses OTP authentication, in which every password transmitted to server is different each time, providing reasonably stronger user authentication. It calculates successive hash values over the password, in such a way that an hash value password can only be used once. Is called OTP because it is usable exactly once.
The OTP authentication system is defined in Request For Comments (RFC) 1938 [65]. This protocol provides authentication that is secure against replay attacks and man in the middle passive attacks, but no secure against man in the middle active attacks. The CHAP, defined in RFC 1994 [66] verifies the identity of the user by means of a three-way handshake. Is considered a strong authentication protocol, and is very used in Internet, important software and OS, such as Cisco IOS [67].
This protocol provides protection against replay attack by using of an incrementally changing identifier and a random number only used once (usually called nonce).

Beside the generic researches for authentication methods, some had been made for healthcare (e.g. [68, 69, 70, 71]). However, no one can be considered 100% secure and some of them have been proved to be insecure against known attacks.
Huang et al. [70] illustrated a system model and an adversary model in wireless health monitoring system, and then proposed an identity-based authentication scheme for the context privacy preservation. Unfortunately, the authors in [69] found that the scheme in [70] lacks protection for privacy and security.
Later, Kim [71] analyzed the security and privacy weaknesses in the paper [69]. In order to remedy those weaknesses, Kim proposed an authentication scheme [71] based the assumption, in 2014. However in [68] were found some serious flaws in the scheme.

More recent in 2015 Mao et al. in [68] show there are three flaws in Kim's privacy preservation authentication scheme [71]. Second, they propose a new smart based authentication in Healthcare. This proposed authentication scheme is different from the previous approaches and can overcome the weaknesses of Kim's scheme.
Quick Response (QR) codes have grown in last years with the emerge of mobile devices. For instance a mobile device can read a QR code and show what is stored in that QR code. They are limited in the amount of information that can be encoded and mostly store information about cryptographic information (secrets, keys, certificates) or a Uniform Resource Locator (URL) to a website. This code become a benefit for advertising strategy, because provides a ways to access a companies website more quickly than by manually entering a URL. For fast authentication these codes can be used in eHealth. Chang et al. implement a HIS mobile application with QR code for authentication [72]. Also in Open Federated Environment for Leveraging of Identity and Authorization (OFELIA) [73] the first engagement between patient and doctor is performed through a QR code.
With the evolution of technology, what is now considered secure authentication can become insecure in the future. So investigation needs to going on. In this new fragmented and mobile world, simple authentication isn't enough. Exits the need for manage and store the users identity. Thus, identity management is a form of on going authentication presents in most of the services nowadays.

2.3.2 Identity Management

Due to the massive organic growth of the Internet, with its unaccountable number of unrelated services, users personal data is currently completely scattered all over the network. This is the direct result of the current need to create different user accounts for the numerous Internet services that are being run by different operators. However this fragmentation of identity data can in some way be seen as a positive feature, because this means that no single system is capable of completely identifying a person identity attributes, in other words, user identity data on the Internet is naturally decentralized [73]. Because identity data is decentralized, anonymization must be ensured. Anonymization ensures that the user may use a resource or a service without disclosing owner identity. This is a very useful tendency it should be explored to improve upon the users privacy. Other related method is the use of pseudonyms. A pseudonym is a fictitious name used to conceal the user original identity, this way can not be identify directly. When a pseudonym is associated to data, we obtain pseudonymous data, which is nether less than personal data that cannot be attributed to a specific data subject without the use of additional information [74]. Thus this is a reinforcement to users privacy.

The interest on users digital identity has been increasing dramatically over the recent years due to its highly strategic commercial value for the market [75, 76]. Internet application providers, companies like Google, Facebook and even Microsoft, are currently under a fierce competition over the hearts and minds of users for their personal data. Their main purpose is to create enormous monopolized centralized databases of user identity attributes as they allow them to produce highly accurate user profiles that they can then monetize very efficiently for marketing purposes [76]. These global companies harvest and aggregate personal data in such a large scale that it will very soon represent a major global threat to personal security and privacy the like of which the world has never seen [73]. Due to this competition, users are placed in a complex scenario where they do not fully understand how their identity and privacy is being

negotiated between a set of internet services, sometimes without their consent or control [77]. For the purpose of aggregate users data, Google, Facebook and other major companys have included services for Identity Management (IdM) and authorization, using protocols like OpenID [78], OAuth2 [79] and Security Assertion Markup Language (SAML) [80]. These are employed as standardized mechanisms to build Single Sign-On (SSO) systems and attribute sharing based on cryptographic token [79, 80]. Thus they serve as IdPs. SSO allows the users to remember just one password or at least much fewer passwords. The most apparent benefit is that users can move between services securely and uninterruptedly without specifying their credentials each time [81]. However to share or give access to highly sensitive data like bank accounts, EHRs or the current geographic position to these companies profit, constitutes a highly risk and violates privacy and confidentiality [82]. Once a user shares this kind of data he immediately loses control over it, not to mention that if these companies suffers an attack, millions of highly detailed personal attributes can be immediately compromised [73].

The SSO is provided by the use of specific protocols. SAML [80] is an eXtensible Markup Language (XML)-based open standard, developed and published by Organization for the Advancement of Structured Information Standards (OASIS). Is a language for securely employ authentication, authorization, and attribute information, expressed in the form of assertions about subjects [83]. SAML uses secure tokens which are digitally signed and encrypt messages with authentication and authorization data, such as an users email and company role. These tokens are passed from an IdP to a cloud application with an established trust relationship. One major goal of SAML is SSO, but can also be used for authorization purposes.
OAuth [84] is one of the fastest growing community-based specifications that allows any user to delegate his access right in a more user friendly and secure way. Is a protocol that allows an application to authenticate against a server as an user, without requiring passwords or any third party server that acts as an IdP. It uses a token generated by the server, and provides how the authorization flows most occur, so that a client, such as a mobile application, can tell the server what user is using the service. The recommendation is to use and implement OAuth 1.0a or OAuth 2.0, since the very first version (OAuth1.0) has been found to be vulnerable to session fixation [61]. OAuth 2.0 relies on TLS for security and is currently used and implemented by companies such as Facebook, Google, Twitter and Microsoft.
The OpenID Connect [78] protocol is a simple identity layer built on top of the OAuth 2.0 [79] protocol. It allows SP to verify the identity of their end users by taking advantage of the authentication services provided by an associated OAuth service. This protocol is also capable of providing basic profile information about the end user by providing the web application developer with an identity/authentication API based on Representational State Transfer (REST)ful web services [85]. OpenID allows users to sign into multiple different web applications with a single account, in SSO mode and at the same time control which of the user identity attributes can be shared with each one of these web applications [73].

Table 2.3 shows the comparison of the various authentication models [80, 85, 79, 86, 81].
A IdM can be defined as the creation, management and use of user identities in an infrastructure [87, 81]. An IdM system have three main entities [88, 89]:

- IdP : responsible for generating identities, for maintaining user attributes and for authenticating users.
- SP : which offers resources and main services to users.

Authentication Model	Log-On Credentials	SAML	OAuth 2	OpenID
Authentication Mechanism	Biometrics or Password Password sharing through network	Token based No password sharing	Token based No password sharing	Token based No password sharing
Digital Signature for Token	Not Applicable	Yes Option for Digitally Signing the Token	No Digital Signatures are not used	Yes Protected with a digital signature, or message authentication code
Metadata	Not Applicable	Yes Uses pre-agreed metadata file for communication	No Uses only consumer key and consumer secret for communication	No Uses only consumer key and consumer secret for communication
Single Sign On Support	No Each application needs separate Sign On	Yes Supports only Web Single Sign On	Yes Supports Web and Native application Single Sign On	Yes Supports only Web Single Sign On
User Identity	Needs to be specified in each users session and application life cycle.	Yes Contains user identity information	No User identity information is not available in the Token	Yes Contains user identity information
Verification with Identity Storage System	Not Applicable	No No validation with the Authorization Server	Yes Validates the token with the Authorization Server	Yes Validates the token with the Authorization Server
Security Risks	Depends on system security design, communication protection, cryptography, security of the credentials, data and credentials storage, end point protection, etc. Can lead to replay attacks, phishing, database intruders, man in the middle, brute force, SQL injection...	XML Signature Wrapping to impersonate any user.	Phishing. OAuth 2.0 does not support signature, encryption, channel binding, or client verification. Instead, it relies completely on TLS for confidentiality.	Phishing. Identity providers have a log of OpenID logins, making a compromised account a bigger privacy breach.

Table 2.3: Comparison of authentication models.

- The user or device, the entity that uses a service and needs to be authenticated. This includes Personal Identifying Information (PII).

IdM systems follow models classified as traditional, centralized, federated and user-centric [89, 90].
In the traditional model, the SP operates as both SP and IdP. In this model there is no identity sharing among SPs. Thus, for each SP, the user has different identifiers and credentials [89, 90].
In the centralized model, there is only one IdP trusted by users and SPs. The IdP share information of users authentication information among SPs and SSO are possible. However, the IdP is a single point of failure. Also, as the IdP has control over users identity information, it may do whatever it wants with such information [89, 90].
In the federated model, IdPs functions are shared among several IdPs, localized in different security domains. A federation is composed by IdPs and SPs of different domains. SPs accept the authentication token issued by an IdP, due to trust relationships established among IdPs and SPs in the federation. Federated model solves the single point of failure problem of the centralized model and offers facilities to the users, because they do not have to authenticate many times, as well they do not have to cope with many identities [89, 90]. However, in the centralized and federated models there is a lack of user control over identity information stored on the IdP, because the IdP controls such information and can disclose it to third parties (e.g. other non federated SP).
User-centric model solves this problem. This model aims to give more control to the user over transactions that involve his identity data [89].
Regarding this work, healthcare services are fragmented in various different geographic institutions (private/public hospitals, local healthcare center) and different services (laboratories, pharmacies, hospitals, nurse centers, etc) which present a need to use standardized mechanisms to build SSO systems and IdM protocols to manage users identity through all institution and services in a standardized way. Besides that data protection legislations [10, 12, 13] enforces users privacy, establishing that users have control over their identity information and over their digital records.
There is a paradox for IdM in eHealth. User-centric IdM model is more appropriate to eHealth applications, because it allows users to have control over identity information (e.g. user attributes) and over the release of such information [90]. Thereby, legal users privacy requirements can be met. In some circumstances, different users (e.g. patient, health professionals) localized in different security domains may need access to patients health data. In such situations, users may not use the same IdP for authentication, what makes the federated model more adequate [90].

There are several related works in authentication and IdM in eHealth that were the base of the proposed solution in this thesis:

- A solution for secure access to EHR using mobile device is proposed in [91]. Four entities compose the solution: (i) user, who wants to access the EHR, (ii) SP, which provides the EHR service, and (iii) two different authentication services, which together authenticate the user to the SP. This solution enables secure communication and authentication between an user (using a mobile application) and a SP. Hypertext Transfer Protocol Secure (HTTPS) protocol and two factor authentication (PIN code and OTP) are used as security mechanisms. However, the proposed solution does not address the publication of users health data in an SP. The authentication services are centralized and are not a

widely known solution, what affects the interoperability. Use of medical devices as SPs or as publishers of users health data in SPs (Machine-to-Machine - machine-to-machine communication) is not addressed.

- A scenario of a health SP that wants to access patients data stored in another Health SP is addressed in [92]. An approach of federated IdM is proposed, where an IdP in the same domain of an health SP has a trust relationship with IdPs of other domains. For protecting patients privacy, each health SP uses a local identifier for a patient. An algorithm proposed by the authors is used for converting the patients local ID into a global ID, used to refer to the patient within the federation. An IdP that receives a data request referring to the global ID can discover the users local ID. A trusted third party, called mediator, is proposed for helping in this conversion. Mediator does not store patients local IDs, ensuring that there is no user tracking in Health SPs. In this work, the protocol for the exchange of messages is proprietary what affects interoperability and machine-to-machine communication is not addressed.
- Campos et al. proposed an IdM system for eHealth based on service oriented architecture [93], in which systems expose their functionalities as services. This IdM respects the European legal requirements [12, 13] and Portuguese legal requirements [14]. An user-centric approach is used, enabling the patient to control the release of identity attributes to SP, as well as the choice of the most appropriate identity for each access. Users identity is registered in a central IdP, which is responsible for the creation of national e-IDs for each user. In this work, the use of SAML guarantees interoperability of attributes and SSO authentication. Nevertheless, machine-to-machine communication is not addressed.
- Proposal described in [94] aims to increase user privacy by using identity pseudonymization, metadata obfuscation and anonymous authentication. In the proposed mechanism, the user may divide his identity into several sub-identities, which have data chosen by the user. For each sub-identity, a pseudonym is created and the user can choose the sub-identity he wants to use in each situation. However, the proposal provides a proprietary mechanism, what affects interoperability with other systems. The work focuses just on user IdM and does not address medical devices publishing user's health data.

2.4 Access Control

There are two main types of access control: physical and logical. Physical access control limits access to campuses, buildings, rooms and physical IT assets, protected by card readers, fingerprint readers, key-locks, security guards, etc. Logical access limits connections and requests in a computer networks, system files, database through authentication protocols, login-password combinations, digital signatures and certificates, biometrics, etc. However in some cases, logical and physical access can be merged in a system. For instance in a case of an Automated Teller Machine (ATM) we need a physical credit card (something you own factor) and the respective logical PIN (something you know) to access.
IT general control should demonstrate that the organization has a procedure or policy in place for technology that affects the management of fundamental organizational processes such as risk management, change management, disaster recovery and security. Have access control polices and middleware are fundamental to reach the desired levels of privacy, confidentiality

and integrity in any secure environment [95].

Authorization and authentication are fundamental to access control. They are distinct concepts but often confused. Authorization, in fact, is dependent on authentication [95].
Authorization, as part of the access control level, is the process where requests to access a particular resource should be granted or denied. It should be noted that authorization is not equivalent to authentication - as these terms and their definitions are frequently confused. Authentication is providing and validating identity. Authorization includes the execution rules that determines what functionality and data the user (or administrator) may access, ensuring the proper allocation of access rights after authentication is successful [96].

Access control is employed to enforce security requirements such as confidentiality and integrity of data resources (e.g, files, database tables) to prevent unauthorized access and use of resources (e.g, programs, processor time, expensive devices), or to prevent denial of service to legitimate users [97]. The distinction between authorized and unauthorized accesses is made according to an access control policy, rules, governs decisions, documentation and processes of determining to which subjects (users, devices or processes) that should be granted access and the objects to which they should be granted access. Access controls also govern the methods and conditions of enforcement by which subjects (users, devices or processes) are allowed to or restricted from connecting with, viewing, consuming, entering into or making use of identified information resources (objects) [96, 98].
Access control is a key feature of HIS. In eHealth systems this means protecting patients privacy assuring confidentially, assuring the best possible care for the patient. These depends on the health professionals having access to the information they need to make the wisest and better decisions. Care processes are often unpredictable and hard to map to strict access control rules. In emergency or unexpected situations, health professionals need to be able to bypass access control. In a crisis, availability of information takes precedence over privacy concerns. This duality of concerns is what makes access control in healthcare systems so challenging and interesting as a research subject [99].
Access control must be coupled with auditing. Audit controls concern a posteriori analysis of all the requests and activities of users in a system, this process ensure that authorized users do not misuse their privileges . Extensive auditing is important to ensure traceability of user actions, in this case in mobile agent actions [100, 15]. For enable audit in order to reinforce access control, mechanisms for session management are essential [101]. Modern and complex web applications require the retaining of information or status about each user for the duration of multiple requests. Therefore, sessions provide the ability to establish variables(such as access rights and localization settings) which will apply to each and every interaction an user has with the web application for the duration of the session. This way access control can be adapted based in users behaviour and histories, through a data mining of a logging mechanisms. EHRs systems needs to generate event logs that give us basic data about the ’who?’, ’what?’, ’why?’, ’where?’ and ’when?’, etc, for better maintenance an empowering of health services [15]. To reach the reliable output of these records, there is a need to set session management protocols in the access control middleware. Therefore, NIST SP800-53 [40] suggests five controls related to session management. They are 1) Concurrent Session Control, 2) Session Lock, 3) Session Termination, 4) Session Audit, 5) Session Authenticity.

In 1969 Lampson [102] has released a formal definition of access control. This first model has

a set of subjects and objects and it associates them to a list of possible operations. After this model, many have arisen. In 1975, the first multilevel model was presented by Bell and LaPadula [103]. Such a model consists of four access levels and access labels, that are unclassified, confidential, secret, and top secret. Users and files are classified with those access labels. An user with ”Confidential” access should not be able to read files marked as ”Top Secret” (a higher level of secrecy), but can read files with ”Unclassified” and ”Confidential”. The Bell–LaPadula Model model is used for enforcing access control in government and military applications [104]. Later the Discretionary Access Control (DAC) and Mandatory Access Control (MAC) models were defined, that today have become two traditional access models, from which almost all the others arise. The DAC model allows the user, the owner of the resources, to grant or deny access to the resources to other users. If a subject is the owner of an object, the subject is authorized to grant or revoke access rights on the object to other subjects at his discretion [105].

DAC can be represented by an access control matrix that indicates which subjects (one row for each) can access which objects (each column) via which modes (the cell contents) [97]. Also can be represented by Access Control Lists (ACL), which is highly inefficient and inflexible considering the fact that each record needs to be accompanied with a separate list [97].

The MAC model is derived directly from the model of Bell-LaPadula [103]. In fact, all subjects and objects are classified based on predefined security sensitivity levels that are used in the access decision process [105]. The MAC defines access rules between subject levels and resource levels, typical rules being Read Down and Write up to ensure the privacy, and Read Up and Write Down to guarantee integrity [106].

Later, Ferraiolo and Kuhn [107] defined a first Role Based Access Control (RBAC) model, including the concepts of users, operations, sessions, groups and defining the concept of role. In this way, a more streamlined management of the policies in an enterprise system is allowed [106]. Also in a deep work Ferraiolo et al. [108] present NIST RBAC model.

Instead of dealing directly with privileges (permissions) per user, the users are merged into roles, and each role is associated with labels and privileges. Roles can be created and added as much as the system requires. Create, Read, Update, Delete (CRUD) operations are defined in the system (more can be defined e.g, append) and can be associated with a privilege that is assigned to a role. The RBAC model as a whole is fundamentally defined in terms of individual users being assigned to roles and permissions being assigned to roles. As such, a role is a means for naming many-to-many relationships among individual users and permissions. In RBAC users can have multiple roles, roles can have multiple users, roles can have multiple permissions, permissions can have multiple roles, users and permissions can be related to multiple objects, and objects can be related to many users and permissions. In this illustrative case, operations is element of permissions. RBAC is receiving increased attention as a generalized approach to access control because it provides several well-recognized advantages [105]. Nowadays RBAC is the most used access control model in healthcare, and various access control models emerge based on him, for example [5, 109, 106, 110, 111]. The core RBAC includes five basic elements [108, 109, 5, 112]: Users, Roles, Objects, Operations, and a set of Sessions, where each session is a mapping between an user and an activated subset of roles that are assigned to the user. This model also have five relations [108, 109, 5, 112], which are the User-Assignment (UA), the Permission-Assignment (PA), the User-Session (US), the Session-Role (SR), and the set of Permissions (PRMS). In figure 2.3 presents, based on the original NIST RBAC [108], the basic operation of core RBAC:

In table 2.4 a brief comparison between the base models of access control DAC, MAC and RBAC, is described [96, 106]:

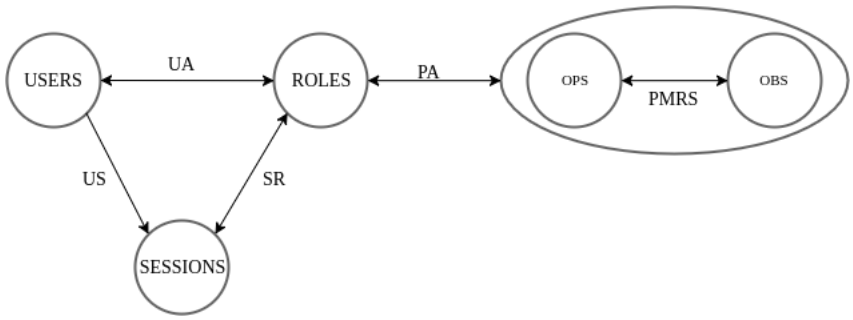

Figure 2.3: Core Role Based Access Control

	Advantages	Problems	Areas of caution
DAC	- Easy to use and administer. - Ownership-based, flexible, does not provide a high degree of security, and hence low assurance. -Aligns to the principle of least privileges.	- Documentation of the roles and accesses has to be maintained stringently. - Multi-tenancy can not be implemented effectively unless there is a way to associate the roles with multi-tenancy capability requirements.	- While granting trusts. - Assurance for DAC must be carried out using strict access control reviews.
MAC	- High level of security, and hence high assurance, but less flexible. - Only an administrator can grant access. - Information flow control rules.	- Difficult and expensive to implement. - Not agile.	- Classification and sensitivity assignment at an appropriate and pragmatic level. - Assurance for MAC must be carried out to ensure that the classification of the objects is at the appropriate level.
RBAC	- Principle of least privilege. - Easy to use and administer. -Roles are assigned based on organizational structure with emphasis on the organizational security policy. - Built into most frameworks. - Able to express DAC, MAC, and user specific policies using role-hierarchy and constraints	- Documentation of the roles and accesses has to be maintained stringently. - Does not support data based access control. - There is a tendency for scope creep to happen e.g. more accesses and privileges can be given than intended for.	- Assurance for RBAC must be carried out using strict access control reviews. - Roles must be only be transferred or delegated using strict sign-offs and procedures. - When an user changes his role to another one, the administrator must make sure that the earlier access is revoked.

Table 2.4: Comparison between the most used access control models.

RBAC model is policy based and can be adapted to be match with data legislations requirements, such as HIPPA privacy guidelines for accessing patient health records and ISO norms for health-care. Various models have been defined for healthcare sector, whose aim is to regulate access to data and the services. A variety of proposed privacy aware RBAC solutions to provide access control in shared EHR repositories can be found in [113, 106, 114, 115, 116, 117]. RBAC is by many considered particularly well-suited for HIS, because it provides several well-recognized advantages like simplicity and ease of administration, flexibility (to adapt to institution rules and legal legislations) and scalability [111, 99].

There are many subsequent access control models, most derived from RBAC. Temporal Role-

Based Access Control (TRBAC) [118] is an extension of the RBAC model, which allows a temporal enabling and disabling of the role. This can be used is time schedule accesses, during the shift of each professional in a hospital.
Attribute Based Access Control (ABAC) [119, 120] is identified as an access control model which is similar to RBAC in the sense that it also adopts a policy driven approach. ABAC is suitable in adapting to dynamic access requirements in EHR systems. Due this similarity with RBAC, and because ABAC can bring more restriction on access control policies [120], ABAC is also one of the most used in EHR systems [121].
Other model that is similar to RBAC is Rule-based Access Control (RuleBAC) [122], which allows the specification of access rules for online resources where authorized subjects are denoted in terms of the relationship type, depth, time, local, and trust level existing between users in the network.
Team-based Access Control (TMAC) [123] was proposed to extend RBAC so as to introduce the concept of teams. An user is engaged in one or many groups. The content of his/her access control privilege needs to be changed depending on his/her group which he/she is supposed to belong. In healthcare systems, beside define role (e.g. nurse, doctor), this can be useful for group health professionals in teams by floor, task (e.g. operation, assisted exams, research) or department.
More recent, appears Situation-based Access Control (SitBAC) [5, 114], containing the definition of situation in the model. Access restrictions are applied based in situations (e.g, allow access for just blood type data during on month). Location has been taken in variable in some access control models, most of them using GPS technology, an excellent example in [124]. These models uses Geographical Information Systems (GIS) as support to make the best evaluations about location and parameters. Other example is Geo-Spatial Access Control [125]. Position is further used to control access, for instance, if a doctor is outside range from hospital, can not access data from his patients.
Based the concept of social networks, the Relationship-Based Access Control (ReBAC) [113] is characterized by the explicit tracking of interpersonal relationships between users, and the expression of access control policies in terms of these relationships. It explores what it takes to widen the applicability of ReBAC to application domains other than social computing. To this end, an archetypical ReBAC model is present here [113], design with the objective of capture the essence of the paradigm, that is, authorization decisions based on the relationship between the resource owner and the resource accessor in a social network maintained by the protection system. This model can be extended to various areas, including healthcare due to the relationships between patients and various health professionals. For instance, patients with Alzheimer disease, schizophrenia, or any other mental illness – or loss of mental faculties due to old age – should be allowed by the access control system to securely delegate the management of their medical records to someone they trust (family, friend, health professional). In [113], access control policies are declarative and qualitative.
For traditional access control models there is usually the assumption that access permissions are known in advance, and that the rules have been set up correctly, but in real settings, errors are made and unanticipated or emergency situations may occur. These expresses the need to a more flexible and adaptable approach be adopted in these cases. The Break the Glass (BTG) [109] policy is used in order to break or override the access controls in a controlled manner, to give certain permissions in case of emergency or other unanticipated situations. The BTG-RBAC [109] appears to satisfy these needs. For instance the HIPPA act specifies the need for BTG as is described in [126]. BTG is needed when normal access controls to processes are insufficient and

an emergency access control mechanism is required. Examples of emergency situations that might require BTG could be account problems (e.g. user has not been given the proper roles or permissions), authentication problems (central authentication system failure) or authorization problems (e.g. an emergency situation such in an ambulance with the patient unconscious) [126, 109].
The first Risk-Adaptable Access Control (RAdAC) model [127] is an example that recognizes in some situations, the consequences to an organization of not sharing information might be worse than of sharing it. The security risk has to be balanced against the operational need, and provide the best access decision. The main difference from traditional models is that RAdAC provides flexibility to adapt access control decisions according to the situation and context at the moment of the request. Security policy grants or denies can be reversed according to the operational need at the time of the requested access. Further some RAdAC appears, but none includes social and behaviours factors. For instance, Sandhu et al. [128] proposed a fine RAdAC model, but this not include trust levels, human behaviour or even evaluation for different network connections. Similar work needs to be done in the healthcare environment as this also requires more dynamic characteristics than access control policies usually allow.

Already exist some access control programming techniques and languages for policy specification [125]. Two of the most relevant initiatives are the standards eXtensible Access Control Markup Language (XACML) [129] and SAML [80, 83], both developed and published by OASIS.
XACML is a general purpose, flexible, and powerful language for specifying and enforcing access control rules following the ABAC model. The XACML language in effect protects content from unauthorized use in enterprise data exchanges, and is developed and and written in XML, which is understood in most global environments [129, 111]. Many models of access control use XACML in their investigation and prototypes (e.g.[111, 125, 130].
SAML can be used to manage authorization. It uses secure digitally signed tokens and encrypted messages with authentication and authorization data, such as an users email and company role. There are other related works which focus encryption techniques to provide CIA in access control models [131].
Augusto et al. in OFELIA [73], presents a framework for user centric identity management that provides an identity/authorization versatile infrastructure that does not depend upon the massive aggregation of users identity attributes to over a versatile set of identity services. In OFELIA personal attributes are distributed among and protected by several otherwise unrelated Attribute Authorities. Only the user mobile device knows how to aggregate these scattered Attribute Authorities identity attributes back into some useful identifiable entity identity. The mobile device thus becomes the means by which the user can asynchronously exercise discretionary access control over their most sensitive dynamic identity attributes in a simple but highly transparent way. OFELIA relies on OpenID and digital wallets.
Li et al. [132] propose patient-centric and fine-grained data access control through a multiple-owner settings model. In this work, patients as owners of healthcare data can generate their own decryption keys utilizing Attribute Based Encryption (ABE) and then distribute them to their authorized users. [131].
Barua et al. [133] propose a patient-centric access control scheme which uses ciphertext-policy ABE. This method determines different access rights for users according to their roles, and then assigns different attribute sets to them. Later, in [134] Barua et al. [134] also suggest hybrid security policy for wireless body area networks with Quality of Services for secure eHealth care system. In this method, cryptographic approaches such as public key cryptography are used for

session key management and private key cryptography is used for data encryption in wireless body area networks environment [131].

All these models, separated, have some limitations in the use of EHR systems. For example, the data in the EHR system are mostly clinical documents, and for this reason it is not easy to identify the owner of the document.
There are several subjects, such as the author of the document, the holder of the document and the patient, that could be considered the owner of the document, depending on the point of view. Therefore, the DAC model cannot be the only one used to manage the access policies to the EHR system. In the DAC model, the owner of the data stored in the EHR is identified. Furthermore, the DAC model is not scalable in a system of large dimensions, because it requires the definition of ACL or of a matrix (user/ objects /operations), which is, in this case, a complicated management. Instead, in the MAC model security labels are associated to resources. It is difficult to use only this model in EHR systems because a given document could be characterized by different security levels depending on the patient or on the health care organization responsible for the data. This model is devoid of the flexibility necessary to be used alone for an EHR system. The RBAC model is static, since the association among roles, operations and objects is made upstream and it is defined by the system. The RBAC model is characterized by a greater flexibility compared to the MAC model and it is easier to handle compared to the DAC model, but it still has many limitations on its use in an EHR system, such as the need to define common and shared roles for healthcare organizations and the lack of flexibility and dynamicity, that is the possibility of policy management by the patients, who in this model cannot change these repeatedly [106]. For instance, to have more flexibility, the attribute-based access control model brings additional attributes associated with the role are used. And other models can bring more flexibility, such as RuleBAC and ReBAC, but separated this is insufficient.

2.5 Electronic Health

Population growth have required a more broad and efficient health system. The basic requirements of quality in health care for this new age are safety, effectiveness, patient-centered, timeliness, efficacy and equity [135]. In healthcare many of the treatments are dependent on how quickly it is obtained a correct diagnosis. The delay in getting diagnosis and, from that applying a correct treatment, can lead to unexpected evolution of the symptoms, making the patient condition worst. In healthcare all the time is precious. It becomes easier apply the best treatment in the patient if the health professionals access information and do research about symptoms or medications faster.

eHealth is the use of information, Internet and communication technology to reinforce health and health care. It refers to forms of prevention and education, diagnostics, therapy and care delivered through digital technology, independently of time and place. As an expansible area, it includes associated notions such as telemedicine, Mobile Health (mHealth), telecare, public health, mental health or telehealth [135].
The use eHealth environment provides many potential benefits [136], for instance improving the quality of care, reducing medical errors, enhancing the readability, availability and accessibility of information and medical records [9]. Beside this, the adoption of eHealth enables more informed decision making and enhanced quality of care, saves lives through remote consulta-

tions, whether urgent or diagnostic, creates more efficient, convenient and potentially more cost-effective delivery of care, facilitates earlier and more accurate diagnoses, provides greater and faster access to a patients medical history, reducing the risk of negative drug interactions or poor response to a course of treatment [137, 9]. Also improves administrative efficiency and coordination, allows rural residents to receive expert diagnosis and treatment from medical centers avoiding long waiting lines in the hospitals and long, tiring trips, increasing independence for patients. Also important, enhances senior wellness and preventative care through telemedicine and remote in-home monitoring [138]. Forms of eHealth that reduce health care costs and medical error's are EHR and PHR, telemedicine services, portable patient-monitoring devices, mHealth, operating room scheduling software, robotized surgery, blue-sky research on the virtual physiological human [139].

The EHRs is the keystone of a HIS. Has been touted for years as an essential part of the multifaceted face of medicine in the information system era. While the benefits of adopting EHR have been detailed in numerous proposals for both healthcare organizations and national initiatives, privacy advocacy groups insist that the issues around privacy have not been addressed adequately at a technical or a business process level [140].
EHRs are electronic versions of the paper charts in your doctors or other health care providers office, created on geographic fragmented institutions. An EHR may include your medical history, notes, and other information about your health including your symptoms, diagnoses, medications, lab results, vital signs, immunizations, and reports from diagnostic [137], included in different ares of health (radiology, dental health, mental health, cardiology, etc). These records also can include a not limited to personal information such as name, address social security number, and birth date.
Similar to this, a PHR is an electronic application used by patients to maintain and manage their health information in a private, secure, and confidential environment. Are managed by patients and are separate from (not replace) the legal record of any health care provider. A PHR can be or not shared, for instance, a patient can create a PHR in his mobile, do operations in it, but never share with anyone [141].
The information in EHRs can be shared with other organizations involved in your care if the computer systems are set up to talk to each other. Unfortunately, medical information about a particular individual is currently maintained by numerous different healthcare providers, and is stored in isolated databases in various incompatible formats.
Information in these records should only be shared for purposes authorized by law or by the owner. The owner must have privacy rights whether your information is stored as a paper record or stored in an electronic form. The same federal laws that already protect your health information also apply to information in EHR [137].
In section 1.1 was instantiated legislations in eHealth. These legislations [10, 11, 142, 12, 13, 14] enforce rules of privacy and management. They enforce that the users have rights over their own health information, regardless of its form. Whether your record is in paper or electronic form, under the privacy rules user have the basic right of to see or get a copy of your medical record, control with who to share their health information securely over the Internet (their families, doctors or others health professional), to request to have any mistakes corrected, to get a notice about how your health information is used, changed and shared, to say how and where you want to be contacted by your health care provider, and to file a complaint if you think any of these rights have been violated. In addition to legislations, ISO and Health Level Seven (HL7) has defined some standards related to eHealth :

- ISO 27799:2016 : Health informatics - Information security management in health.
- ISO/TC 215: Standardization in the field of health informatics, to facilitate the coherent and consistent interchange and use of health-related data, information, and knowledge to support and enable all aspects of the health system.
- ISO 13606-1, ISO 13606-2 , ISO 13606-3, ISO 13606-4 , ISO 13606-5 : Health informatics - EHR communication.
- HL7 Standards : Standards that define how information is packaged and communicated from one party to another, setting the language, structure and data types required for seamless integration between systems.
- openEHR: a virtual community working on means of turning health data from the physical form into electronic form and ensuring universal interoperability among all forms of electronic data. The primary focus of its endeavour is on EHR and related systems.

In past years eHealth has grown a lot, with various companies migrating to the health market to do revenue. In a longitudinal study, from 1999 to 2002 the adoption of eHealth grows 788% [143]. In United States, the 2011 National Ambulatory Medical Care Survey revealed that the 55% of U.S. physicians had adopted the EHR technology [144].
Despite all advantages and useful components, services, technologies (and so on), there exists a gap between postulated benefits and actual outcomes, while the potential of eHealth is celebrated, robust results in a variety of care contexts lag behind expectation. Besides this, health authorities and institutions generally welcomed some early or non-proved developments, despite worries about the quality of online health information and other digital hazards such as privacy or data security [135].

The use of mobile technologies and wireless networks in health gave origin to the term mHealth. Is defined as the medical and health practice supported by mobile devices, such as mobile phones, tablets, monitoring devices, or other mobile devices. mHealth involves the use and capitalization on a mobile device core utility of voice and short messaging service, as well as more complex functionalities and applications including general packet radio service, third and fourth generation mobile telecommunications (3G and 4G systems), GPS, and Bluetooth technology [145].
As referenced and proved in section 1.1, smartphone are the most ubiquitous device that people now hold. Their portability, availability and improved ease of use leads to their acceptance and domination in the market [18], and they offer various opportunities to create innovative mHealth solutions.
Healthcare providers show increasing interest, excitement and more than a third of physicians report recommending mHealth apps to patients. mHealth have a strong impact on healthcare monitoring and alerting systems, clinical and administrative data collection, improvement of communication between health professionals and patients, record maintenance, healthcare delivery programs, medical information awareness, detection and prevention systems, drug-counterfeiting, ambient Assisted living, and so on [17, 146, 147]. For instance, health professionals can use smartphones to access and create patient records (e.g, EHR), to view exam results, to share and ask for second opinion diagnosis, and to prescribe medications [17]. On the other hand, the patient, can use smartphones to access and update their medical records, to control access to their medical records, to monitor their health statistics and to view their prescriptions [147]. The impact of mobile applications can also improve the reducing healthcare

by use of remote analysis services, remote mHealth monitoring technologies, at-home triage services and telemedicine appointments [148, 149].
However, barriers continue to exist, impeding full adoption of mHealth apps in a prescriptive and integrated manner. These barriers include lack of scientific evidence to lack of integration into workflow systems, regulatory and privacy unknowns and lack of provisions for reimbursement. These barriers are further magnified by a complex healthcare system with limited interoperability both within and across healthcare organizations [150].
mHealth applications do not need approval to be published and any developer can upload their mobile healthcare applications on the global market. Mobile applications are developed with intentional leaks to track user and target advertising [20], this represents many risks to the privacy and security of the users. With the increasing use of smartphones for healthcare purposes, more and more people now share their personal healthcare information using a variety of applications. The vast number of existing mHealth applications creates a serious problem for users, as often times they are unaware of how their data are managed and used and by whom. It can also cause physical harm by providing wrong and incorrect information and poorly developed features [148]. Research has demonstrated that many medical applications currently available in mobile applications stores have flaws that could prove detrimental for medical practitioners and their patient [151, 148].

eHealth brings enormous advantages, but because EHR systems are often web-based, many patients fear that also exposes their medical history and personal data to anyone with an Internet connection.
Lack of adequate protection in sustaining the CIA aspects leads for investigation to the potential threats particularly in HIS domain. Also poor organization and implementation of security controls or low awareness of risk analysis practices within public and private sector especially in healthcare organizations also need particular attention.
Many health services and institutions use devices and software out of date (e.g, Windows XP), versions of OS that has not received publicly available security updates for some time, and even those which are running on newer operating systems are often sporadically maintained. For an attack which relies on using a hole fixed less than three months ago, just a slight oversight can be catastrophic. Attacks on healthcare providers across the world are at an all-time high as they contain valuable private information, including medical records.
Given that medical records contain a wealth of information that can be used for identity theft and fraud (such as social security number, address or claims data), personal health information carries a higher value on the black market than other industries.
A survey conducted in 2005 by Harris Interactive of Rochester, in New York that found that 70 percent of people in the U.S were very concerned that personal medical information would be leaked because of weak data security [140]. To prove that health information is at risk, in Christus St. Joseph Hospital, Houston Texas has found 16 thousand records compromised by theft, University of Chicago Hospital reported an employee found selling patient data and Wilcox Memorial Hospital, Kauai, Hawaii 1 hundred and 30 thousand records also compromised by theft [152]. Other famous incident, a Department of Veterans Affairs database containing sensitive personal health information of 26.5 million military veterans, including their social security numbers and health problems was stolen by an employee who took the data home without authorization [132].
According to survey in [153] of 223 healthcare executives, about 80 percent of U.S based healthcare executives have reported compromise of their organizations information technology by cy-

ber attacks. Other survey in [154] reports that in the first 4 months of 2015, more than 99 million healthcare records have been reported to be exposed through 93 separate attacks.
HIS (which includes EHR systems) security threats have increased significantly in recent years. For instance, during the period from 2006 to 2007, over 1.5 million names were exposed during data breaches that occurred in hospitals alone [155].
Using HIS in health institutions brings two types of threats, internal and external [156]. An internal threat includes various types of employees behavior such as employees ignorance, curiosity, recklessness, inadequate behavior, taking someone else password and giving password to another employee. External threat includes viruses and spyware attacks, hackers and intruders in premises [156].
The study [157] identified 22 types of threats according to major threat categories based on some previous researches and ISO/IEC 27002 (ISO 27799:2008):

1. Power failure/loss.
2. Network Infrastructure failures or errors.
3. Technological Obsolescence.
4. Hardware failures or errors.
5. Software failures or errors.
6. Deviations in quality of service.
7. Operational issues.
8. Malware attacks (Malicious virus, Worm, Trojan horses, Spyware and Adware).
9. Communications interception.
10. Masquerading.
11. Unauthorized use of a health information application.
12. Repudiation.
13. Communications infiltration.
14. Social Engineering attacks.
15. Technical failure.
16. Deliberate acts of Theft (including theft of equipment or data).
17. Acts of Human Error or Failure.
18. Staff shortage.
19. Wilful damages.
20. Environmental Support Failure/Natural disasters.
21. Terrorism Attacks.

The study in [157] shows that the most critical threat is the power failure. This is due to power failure of server, air-conditioning failure or interruption by SPs. Besides that, acts of human error or failure threat also show high frequency of occurrence in HIS. Furthermore, in acts of human error threat, one of the greatest threats to HIS is the entry of erroneous data by staff. These kind of incidents happens due to lack of awareness and good practices among the staff, which includes the shared accounts to gain privileges or share unauthorized data [157].
IT security in healthcare systems, services and applications is positioned as a major concern due to the high privacy and confidentiality requirements of sensitive healthcare data. eHealth faces many security challenges The study in [158] introduce a complete and structured summary of eHealth security challenges. European Network and Information Security Agency (ENISA) in Security and Resilience in eHealth report [159] also introduced a section with eHealth security challenges. The next enumerated list [158, 159] presents a selections of the most important challenges in eHealth:

1. **Computational and memory limitations**: Most of health devices are embedded with low-speed processors, not powerful in terms of its speed. In addition, these devices are not

designed to perform computationally expensive operations. Therefore, finding a security solution that minimizes resource consumption and thus maximizes security performance is a challenging task. Also their memory may not be sufficient to execute complicated security protocols [158].

2. **Mobility**: In general, healthcare devices are not static but mobile in nature. Such devices are connected to the Internet through an Internet service provider. For example, a wearable body temperature sensor or a heart monitor may be connected to the Internet and notifies the concerned caregiver of the users conditions. Such wearables are connected to the home network when the user is at home, whereas they are connected to the office network when he or she is at office. Different networks have different security configurations and settings. Therefore, developing a mobility-compliant security algorithm is a serious challenge [158].

3. **Devices and scalability**: The number of health devices has increased gradually, and therefore more and new devices with new specifications and OS are getting connected to the global information network. Therefore, designing a highly scalable security scheme without compromising security requirements becomes a challenging task. Other challenge lies in designing a security scheme that can accommodate even the simplest of devices [158].

4. **Communications media and network security**: In general, health devices are connected to both local and global networks through a wide range of wireless links such as Zigbee, Z-Wave, Bluetooth, WiFi, GSM, WiMax, and 3G/4G. Wireless channel characteristics of these networks make traditional wired security schemes less appropriate. Therefore, it is difficult to find a comprehensive security protocol that can treat both wired and all wireless channels characteristics equally [158], and assure network security.

5. **Systems availability**: is the basic feature for achieving continuity of electronic healthcare. It is about continuous accessibility of critical health information by authorized professionals in order to ensure the best healthcare services. Systems availability may relate to physical systems function (e.g. networks, storage) and affect significantly the healthcare delivery. In a hospital, if the network is down, the healthcare professionals cannot access patients data and cannot prescribe. Generally, the more digitized the health sector in a country, the more the health services are affected by interruptions in eHealth infrastructures. So there is needed to ensure that systems are always available as resistant to DDoS attacks [159].

6. **Lack of interoperability and data standardization**: eHealth infrastructures include many diverse systems, databases, devices and applications interconnected at various scales. A core issue for an effective and secure use of these services is to ensure a high level of interoperability and guarantee that data is transmitted safely through individual data systems, health service institutions, healthcare providers and patients and, on the other hand, that the recipients system is able to use the information received in order to proceed in various actions. For example, the vocabulary used in EHR, namely the terminologies, the classifications, the metadata, or the cloud services, must be based on universally applied standards and an agreed-upon framework or some open protocols/APIs for secure information exchange and services integration. To improve interoperability global institutions needs to agree and follow standards, which is hard to achieve. The lack of interoperability may also affect the security updates in an eHealth services network [159].

7. **Authentication and access control**: One of the greatest vulnerabilities in eHealth data security is sharing data between third parties and insiders (breaches by employees). This indicates authentication (2.3) and access control (2.4) as keys security features in eHealth infrastructures [159]. The challenge relies in define and implement secure authentication protocols and reliable dynamic access control models.

8. **Data integrity, message authentication and data confidentiality**: One of the most common cyber security challenges in all eHealth (and other areas) is ensuring quality and integrity of the data that are stored and exchanged for clinical and administrative purposes [159]. Integrity can be assured through cryptographyc hash functions. Also message authentication is the process of simultaneously verify both the data integrity and the authentication of a message. To calculate the message authentication and integrity in same process it can be used a Message Authentication Code (MAuthC) or Hash Message Authentication Code (HMAC). eHealth data is restricted with legal laws, that enforce confidentiality, which is essential to achieved full privacy. All these can be a challenge due to lack of interoperability, computational and memory limitations, and some devices may no support more strong algorithms. Also cryptographic algorithms are always in change, old ones become obsolete and insecure, so is a challenge relational with dynamic security updates.

9. **Counter-Tampering techniques**: An attacker may tamper with devices and then may later extract cryptographic secrets, keys, modify programs, or replace those with malicious nodes and code. Counter-Tampering techniques are a way to defend against such attacks [158].

10. **Data loss**: The digitalization of information and the high level of eHealth services penetration in the healthcare sector mean that a significant amount of vital, personal and confidential data are stored in digital format. The protection of the data from loss is considered to be very important. On the other hand, sometimes it is impossible to avoid ending up in such a critical situation (e.g. software and hardware faults, network faults, security attacks, and natural disasters), so data recovery and the time-frame that it can be achieved is closely related to data loss. Common causes of data loss are unauthorized access to clinical patient data by IT vendors and by healthcare organizations personnel and the back-up policy [159]. To avoid unexpected losses, data must be performed a back up regularly. But the lack of time, resources or technicians difficult these tasks.

In the chapter 4 and 5 it will be present the requirement analysis, diagrams, the techniques, technologies and models which will be used to implement the system prototype.

2.6 Conclusion

This chapter contains a deep review of the main topics of this master's thesis. Security sometimes is left to the end of project, and it can even be excluded for lack of time or resources. To achieve an overall secure system it's essential to understand and study the most dangerous threats and how to prevent or mitigate them. Otherwise highly important data can be compromised or forged, or even critical systems that depend on high availability can be compromised. Within this chapter it is perceived, not only the main vulnerabilities and security breaches, but

also the key role that authentication/identity management and access control has in every system, in order to avoid unauthorized accesses. The evolution of exploits and threats requires updated systems. Due to the challenges and limitations in eHealth, achieving up-to-date protection can be a difficult task to be accomplished. eHealth is in constant growth, however barriers seems to limit the full potential that eHealth can bring to our world.

Chapter 3

SoTRAACE- Socio-Technical Risk-Adaptable Access Control modEl

3.1 Introduction

The traditional solutions for access control are based on predefined access policies and roles and are inflexible because the access control policy is hard-coded and pre-set into decision logic or database restrictions. More, they assume uniformity of people role, devices , environments and some of situational conditions, across the enterprise/location, time and connection. There is a lack and a need for new and subsequent RAdAC [127] models. With this new mobile paradigm of anytime/everywhere, from different mobile devices and Internet wireless connections, there is a need to search for more innovate, flexible, adaptive, dynamic, transparent and more resilient access control models, that are required for more heterogeneous requests. Although many of the latest models allow you to ensure the needs of the EHR system, they still lack components for dynamicity and privacy protection, which leads to not have desire levels of security and to the patient not to have a full and easy control of his privacy. Integrate and merge some of these models in a new dynamic, adaptable and secure privacy-aware models can be the correct research path. Within this master thesis, was researched and published a novel dynamic access control model, SoTRAACE [26], which can model the inherent differences and security requirements that are present in each of the described scenarios. To do this, SoTRAACE aggregates attributes from various domains to help performing a risk assessment at the moment of request. To provide a more secure and transparent access decision, SoTRAACE integrates data from context and location (e.g., wifi connection, Service Set Identifier (SSID), GPS location), type of device (e.g., International Mobile Equipment Identity (IMEI), operating system), user profiling and log trace (e.g., previous similar accesses, denied accesses), institution or legal requirements (e.g., [13]), type and sensitivity level of the requested resource (e.g., blood test, radiological or cardiology exam), unanticipated situations (e.g., BTG features) as well as relationships between the patient and her family or healthcare professionals (e.g., delegation features, Social Network System). SoTRAACE will be included in the implementation of system prototype.

3.2 SoTRAACE Model

A first step in the development of an access control model is the identification of the objects to be protected, the subjects that execute activities and request access to objects, and the actions/operations that can be executed on the objects. The SoTRAACE model is presented in figure 3.1 and its components are described next.

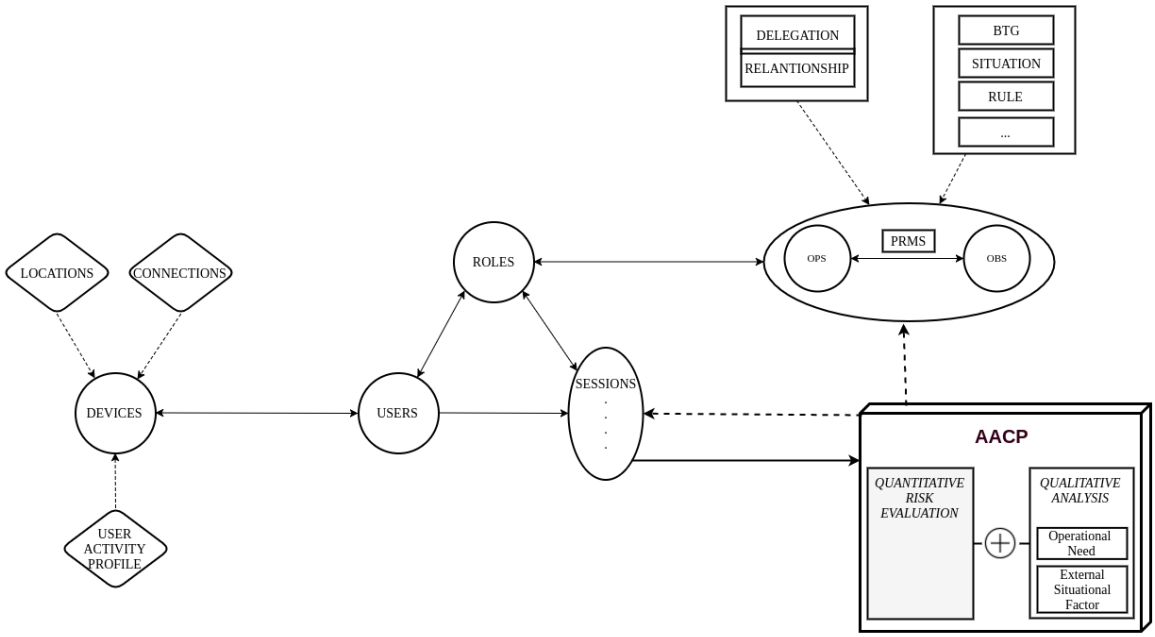

Figure 3.1: SoTRAACE - Socio-Technical Risk-Adaptable Access Control Model.

SoTRAACE NIST RBAC-BASED Components

Users: a User entity is a human being requesting an access operation to an object, through a mobile device.

Roles: Users can be mapped to different roles, each role with different associated permissions and restrictions. In this paper our use-cases focus on a patient-centric solution only.

Sessions: Each session is associated to a user and their roles, and registers what a user does (tracking logs) for a specific period of time, what resources were accessed and what operations have been made, from which device and with which connections. With this, the system has enough data to learn about user's legitimate and compromised access behaviour and history overtime, enabling the building up of heuristics/profiling data for subsequent access requests. In our model, each session will provide information to decide in which conditions the requested data will be retrieved to the user and be, afterwards, adaptable according to the risk calculation. More in the description of the Adaptable Access Control Policy (AACP) component.

Objects (OBS): An object is an entity that contains or receives information.
In Healthcare, an object can have different degrees of granularity representing a complete HIS or just documents (e.g discharge notes), or even a particular set of data within a document (e.g genetic information), each potentially having different degrees of sensibility.

Permissions (PMRS) and Operations (OPS): An operation is an executable of some function for the user.
A permission is an approval to perform an operation on one or more objects.
The next items describe examples of existing constraints that can turn the use of PRMS in SoTRAACE more flexible.

Break The Glass: BTG [109] is used to override pre-defined access in a controlled manner, with strong auditing measures to make users responsible for their requests and justify them, whenever necessary [109]. In SoTRAACE, if a patient in a hospital is incapable of communicating if s/he is allergic to a specific medication and the nurse treating him/her can have the justified possibility of overriding pre-defined policies and accessing patient's allergy data, before administering it.

Situation and Rule: Situation factors are considered in our model [5]. One example of a situation factor is a request for data access for research purposes. This case comes with different obligations/constraints than when compared to a normal data request. Situation can also be a location, the user or the object, or can be external to all defined parameters. Local situations are directly applied to the access control policy (PRMS: OPS, OBS).
Specific static rules can also be set by the user, health professional or institution and be directly applied to the access control policy.

Relationship/Delegation: Users can establish a relationship with family, friends or healthcare professionals using ReBAC [113]. What differs for delegation is mostly the fact that relationship uses the direct concept of Social Networks Systems. In healthcare: Bob as an obstetrician requests access to Maria's pregnancy data. If she accepts the request, a relation is created in PRMS, to which she can add constraints.

SoTRAACE new components

Devices: A Device is defined as an entity that aggregates several contextual, behavioural and situation attributes (e.g., location, connection) and allows the user to request access to objects in the session, taking into account these previously collected attributes, which are used to evaluate the risk in the AACP. In our study, the Device is a mobile device. A device can only allow access if the respective IMEI is registered within the user's profile at the server's side otherwise, the user needs to add that device (using a multi-factor authentication). We can define other device attributes to use in the risk evaluation such as the type of OS (e.g., android is proven to be more unsecure than iOS) or the use of older, non-updated versions, which are more prone to be exploited.

User Activity Profile (UAP) : A UAP is defined as a set of attributes that contains history information regarding all associated user's devices and locations through all the user's sessions, as well as information about user's previous accesses and used connections. To enable audit and learn about the user's behaviour, each user's session is registered in the UAP. This allows an easy search for malicious or legitimate behaviour over different sessions.

Locations: In our model, Locations is defined as a set of attributes. We will use GPS sensors from mobile devices to track the most common user's locations to build a user profile location and check access and calculate risk having into account location history and other current parameters. A profile history of regular access from Portugal may help to raise suspicious of unauthorized access when there is evidence of interleaved accesses from Australia within a reduced time gap. A user may not want to share his/her location due to privacy issues. In this case, the locations set will be empty. If a user wants to add a new device or new location, a multifactor authentication is required.
UAP, Session and Device can have associated/registered none or many Locations.

Connections: A Connection is defined as a set of attributes and is a communication tunnel that binds two end points (e.g, device and server or device and user) to exchange information. In mobile devices, the first evaluation is to determine if the connection is made to a mobile Service Provider (e.g, 3G,4G) or to a wireless network. SoTRAACE can evaluate the encryption algorithm, the length of the encryption key, used protocols, if the connection is password protected, the SSID, and so on.
It also compiles a few questions to help with the risk connection evaluation: *How many users are connected to the wireless network? How many wireless networks are available in the vicinity?*
UAP, Session, Device can have one or more Connections.
The next subsection presents the main engine of SoTRAACE where risk adaptable features can verify and adapt to the environment and user who is requesting access.

Adaptable Access Control Policy

RAdAC [127] introduced a base definition for the core characteristics of security risk evaluation, operational need, external situation factors and adaptable access control decisions. We will adapt them into our model and add other features.
For quantifying the security risk of each request, the AACP aggregates, in real time, all attributes that are instantiated in the session, namely connection, location and the user activity profile from the device. It also aggregates data from the object descriptive metadata (e.g., type, sensitivity level of the requested resource, owner, institution/company related) as well as the object logs (who/when/where that object was accessed or changed). Each attribute can contain exploitable threats that will be used to perform the quantitative risk evaluation by anticipating how and what is the probability of that security flaw being exploited (e.g. without the use of https, there is a higher probability that user credentials can be stolen).
The quantitative risk evaluation is complemented with more qualitative measures such as the operational need and external situation factors to provide a more accurate, secure and adapted access decision.
This can be understood as the need to access the requested object and can influence more or less the already measured/quantified risk. In our model, the operational need is dynamic as is also defined through other aggregated attributes (connection, location, device, etc), roles, user and situations but evaluated at a different, less objective but more human behavioural light.
As an example of a more qualitative risk evaluation: if a nurse is trying to access a medical record at a different time from her normal working hours, using a different device and connection, the calculated quantitative risk will be higher than usual.

However, a more qualitative analysis may attenuate that risk if it confirms that the nurse is accessing data that is customary. In this case, auditing can register some warnings and visual security restrictions can be applied as a preventive measure.

After having calculated the total risk, AACP specifies a set of rules (the decision) that can be applied to PRMS, and under which conditions. These dynamic decisions and rules can: 1) block or allow the access; 2) enforce the fragmentation of the object and just allow access to some fragments; 3) block one or more operations to the object; 4) trigger other hidden security protocols to better avoid the risk; 5) in certain situations different levels of security can be afforded; 6) in situations of extreme operational need and high risk, more secure channels for communications or different cryptography techniques can be used.

However, these high levels of security can be heavy performance wise. For instance, if the risk is low and the operational need is also low, perhaps is not necessary to waste such heavy-cost in security resources but opt for more user interaction security options, which can also empower the use of older devices that some users can still have. To do this, AACP decision rules can also be applied at the Session level where AACP can change the visualization of the requested data, providing dynamic ways to present the object to the user, containing still the requested object but presented, perhaps, in a more categorized/ordered way, not showing all data at once or hide some data that AACP has some degree of certainty that is never useful to that patient.

Transparency is also a must and the user can find out, at all times, what the model is doing, why is doing it, with the option to provide information and tips about risk evaluations and past decisions. Finally, past decisions and respective parameters provided by the AACP will be used to help decide each subsequent decision. This knowledge can be used to improve algorithms that determine the risk, operational need and the rate of positive access control decisions, to build more accurate UAP and object logs. The full scientific paper is presented in appendix A.1.

3.3 Towards a flexible risk evaluation

SoTRAACE is modular and adaptable, the parameters and levels of risk of each evaluated parameter can be changed according to the needs of each environment. Based in the review in chapter 2.5, RFCs about network connections and respective types and encryption and some debates with some elements of Center for Health Technologies and Services Research (CINTESIS) research unit, the figure 3.2 presents the technical environment attributes that SoTRAACE uses as input and respective rank of risk of each attribute. These attributes are used in the developed system, aggregated at the Android application and sent to IdP (where SoTRAACE reside). As is illustrated in figure 3.2 each risk factor has a set of different attributes that has different quantitative values (low =1 , medium = 2 and high =3).

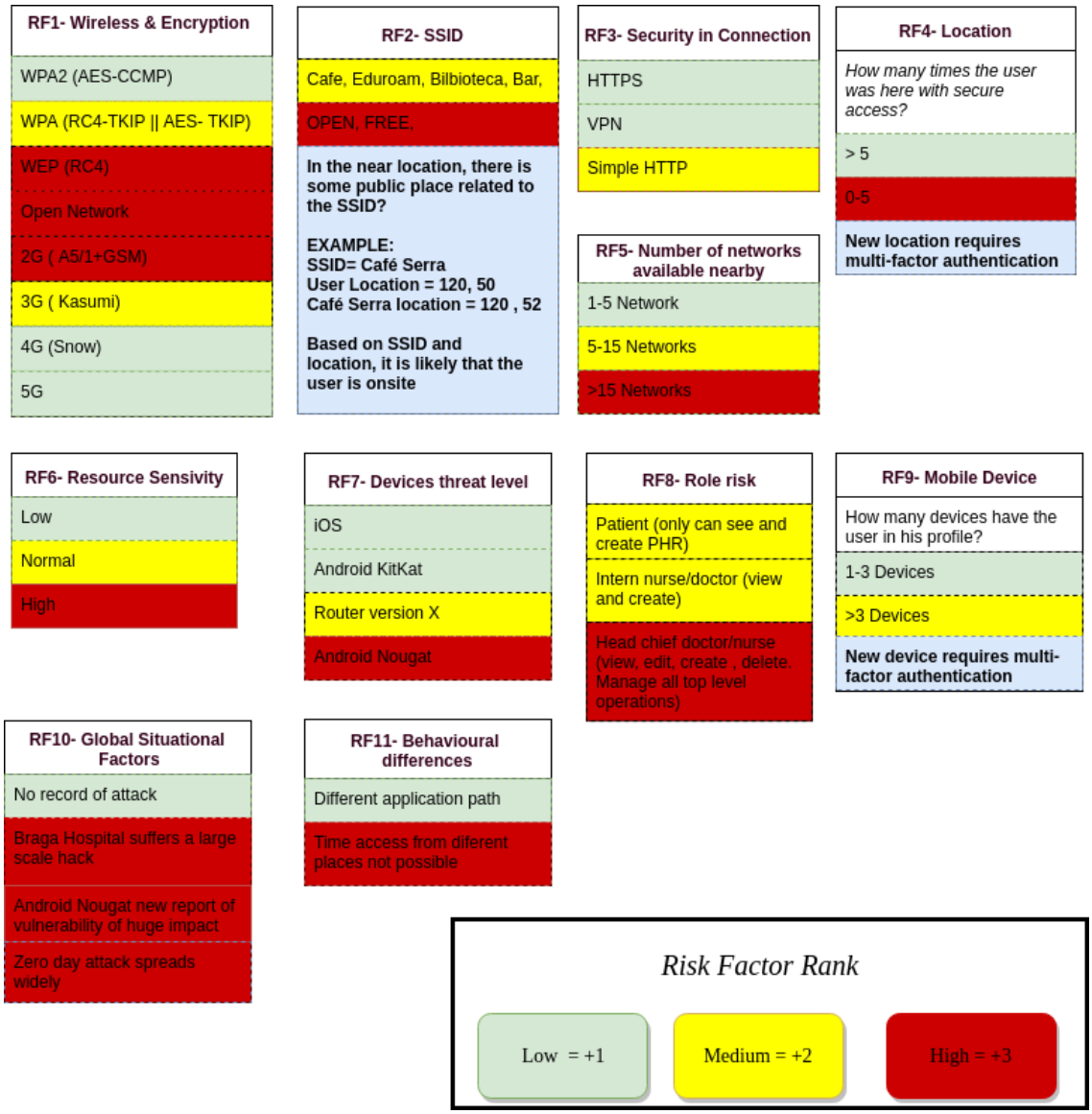

Figure 3.2: Technical environment attributes and respective ranks.

The risk of the aggregated attributes are further computed as an weight mean, in the IdP. Each attribute has distinct weights in this risk computation. The weights were based on a Delphi study [160]. The Delphi method [160] is a structured communication method. Relies on a panel of experts in the research area, and this method is structured, systematic and interactive. It was used a questionnaire, where the experts answer in two or more rounds (in this study it was two rounds). At the end of each round, it is provided an anonymised summary of the experts forecasts from the previous round. With the provided summary, the experts are encouraged to revise their earlier answers in light of the replies of other members of their panel. The objective is that during this process the range of the answers will decrease and the group will converge towards the most accurate answer [160].

In the questionnaire experts have to choose the weight of each attribute mentioned in figure 3.2. The list of selected experts contains both professionals from industry, education and research areas. The questionnaire is presented in appendix A.2. The means and standard deviations of each question, of the Delphi study first and second round, are shown in table 3.1. As expected the standard deviation in most cases reduces from the first to the second round, converging to a consensus of opinion among the experts.

	First Round			Second Round		
Question	**Mean**	**Std Dev**	**Ranking**	**Mean**	**Std Dev**	**Ranking**
RF1 - Type of wireless connection and respective encryption	4,25	0,87	3	4,25	0,45	3
RF2 - Patterns in the SSID,or profile of a wireless connection	3,83	1,03	7	3,67	1,23	8
RF3 - Security mechanisms of the communication protocol	4,58	0,51	1	4,75	0,45	1
RF4 - Location where the request is being made	2,58	1,08	10	2,89	1,19	9
RF5 - Number of wireless networks reachable in the present location	2,00	0,60	11	2,25	0,62	11
RF6 - Information sensitivity of the requested health record	4,42	1,16	2	4,25	1,14	4
RF7 - Known vulnerabilities associated to a device / OS version	4,17	0,83	5	4,17	0,83	5
RF8 - Role of the person trying to access a resource	3,92	1,31	6	3,92	1,16	6
RF9 - Number of mobile devices that the user has registered	2,67	1,37	9	2,42	1,16	10
RF10 - Occurrence of recent reports of global security threats /vulnerabilities	3,67	0,89	8	3,83	0,83	7
RF11 -Observable behavioural differences regarding time and location of the person who is performing the request	4,25	0,75	4	4,50	0,67	2

Table 3.1: Delphi study first and second round results.

The respective final results extracted from the Delphi final round are converted to weights and are reallocated in the following weight mean to calculate the risk:

$$Final\_Risk = \frac{\sum_{i=1}^{11} RFi \times Wi}{\sum_{i=1}^{11} Wi}$$

RF= Risk Factor W= Weight

Finally, the risk level output is a number between 1 and 3, and is mapped to new security controls and access restrictions:

1. In all requests give feedback to the user about the most unsecure attribute in the request (e.g, send message advising that Wi-Fi Protected Access (WPA)2 are more secure than open network).

2. If `final_risk<=1.6`, no restrictions will be applied. With the security solutions presented in this chapter, the system is protected.

3. If `final_risk>1.6 && final_risk<=2.2`, the system will use Authenticated Encryption with Associated Data (AEAD) to compute end to end encryption.

4. If `final_risk>2.2`, the system will use AEAD, and the access to resources with high sensitivity will be denied.
5. BTG requests : besides have extreme risk, the urgent situation of the request always requires access to the data. These requests always use AEAD and the data is fragmented and sent in parts(R00, R01, R02). In the client side the parts are merged and the file presented.

Note:

- More levels of security can be added as final risk output. Also more risk factors can be added, SoTRAACE is very modular. But for initial implementation and tests, just the previous will be included.

In future research work, the authors of the SoTRAACE [26] aim to fine tune this risk evaluation method and provide means to build as accurately as possible the user profile, which aims to provide a new scientific publication in near future.

3.4 Conclusion

SoTRAACE is based on standard access control models as well as other peer-reviewed models but also integrates some new features that the authors believe are very important in today's mobile paradigm. This integration enriches the model and at the same time provides easy adaptation to various technical, contextual and user needs. This is the case of the healthcare domain where heterogeneous health professionals, type of data, different security levels, regulations and need of accesses, require an adaptable but still secure model. Although we could only describe two use cases with the patient role, due to space constraints, it would be easy to adapt to different and more complex cases.
SoTRAACE integrates novel features such as the analysis and evaluation of socio-technical risk. It calculates not only a quantitative measure of all the parameters but also a more qualitative one that either minimizes or strengthens the objective value. It provides a more complete analysis without just bluntly denying or providing access to resources independently of other vulnerabilities that should be considered in the access decision. Our model also takes advantage of user and object profiling data - so strong audit features need to be used not only to control but also at the service of the user - to better calculate both types of risk and providing more decision options, as well as better usability.

SoTRAACE access decisions can include access with: 1) extra restrictions; 2) improved security mechanisms without affecting user's access to the object; 3) warnings to the user, if s/he thus requires; 4) and adaptable security visualization without compromising both security and user's request. This fine tuning is helpful to adopt SoTRAACE to different domains or types of security requirements as for instance, in government or banking where security is highly demanded, AACP can take more restrictive and controlled decisions than, for instance, in research or education where it is more important to have access to more information, faster, but in a secure and trustable way. Secure visualization for common users is still a very incipient field but the authors believe that it is a domain to focus research. A simple, clean and organized view of data can possibly prevent many security hazards, in different situations. We plan to focus more work on this topic. Also, our model can be adopted in systems with other types of devices.

Chapter 4

Requirements Analysis and System Architecture

4.1 Introduction

The analysis of requirements is an essential process for the development of a product, because it is in the requirements analysis that the objectives to be developed are precisely defined. This chapter presents the design, requirements and architecture of the developed HIS. It starts by presenting the conceptual design and requirements of the system proposal and then presenting Unified Modeling Language (UML) diagrams of the main actions and procedures. UML helps to visualize the system architecture in a standard way. The use of UML visually supports the system specification and the development process, in order to lead the product under development to the success. Following the UML, system architecture and specifications are presented. Afterwards the technologies used are introduced. At last, this chapter refers to the security requirements and the attack model of the system.
The final goal is to deploy a user-centric HIS, for patients and different health professionals positions. In an initial approach, to better assure security and functional tests, the system will only be designed and implemented for doctors and patients. This way all subsequent added positions have the basis tests performed and security assured, avoiding threats, bad configurations and bugs.

4.2 Requirements Analysis

The system will be divided in three major frameworks: mobile applications, web services (IdPs) and SPs with data bases. The mobile framework comes with two mobile applications named myHEnCE and myHEnCEPRO, the first for patients and the other for health professionals, with different options.
The basic requirements of the system were established after interviews with health related professionals and health researchers at CINTESIS, Faculty of Medicine, University of Porto. Also the deep review in section 2.5 enables a more concrete requirement analysis. The privacy of the end user (in this work, essentially the patient) is an essential requirement. Users data must be protected from unauthorized access, and only can be viewed by user authorized institutions and health professionals. Also the requirements for anonymity ensure the users identity protection related to a subject, object or an operation. This is a very useful tendency that the system should explore to improve upon the users privacy. A methodology to achieve anonymization is the use of pseudonyms. When a pseudonym is associated to data, we obtain pseudonymous data [74], which is data that cannot be directly associated to an user, reinforcing users privacy.
Considering the review at section 2.5 and the legal requirements [10, 11, 142][12, 13, 14], the patient:

- Should have the right of control over their own clinical documents and must trust in the HIS.

- Should have the ability to change (add or remove) at any time the rights of access to their documents.
- Should be able to hide their documents from a specific health professional.
- Create PHRs and share it with health professionals at any time.
- Have its own data anonymized.
- Health professionals must have the option to request access to the patient clinical documents and to communicate via messages with them.

Besides data, attributes such as location, devices and sensitive profile information must be protected as well. To assure this, the adaptable access control evaluation is embedded at the IdPs and all attributes are aggregated there, never reaching the SPs. These attributes help SoTRAACE to perform a risk evaluation and perform the best access decision at the moment of each request. The ACL exists in the IdP, but must also exist in the federated SPs. For instance, when the internet connection fails in a institution, the internal Local Area Network (LAN) of the institution checks the internal ACL data base for permissions, without the need to connect to an outside IdP. The database in the IdP is the one that contains the main ACLs. The health data is stored in institutional databases. The local SP ACL are synchronized with the main ACL in the IdP. For the system and services to remain available to authorized parties, a set of IdPs must exist. If one fails, another takes place.
The system can have a lot of geographically fragmented federated institutions (SPs), that can share data between them, if the user consents. The user can not be obligated to perform a login each time he requires an EHR from different intuition. Based on the review in section 2.3, using SSO allows the users to remember just one password or at least much fewer passwords. Thus, the users can move between services securely and uninterruptedly without specifying their credentials each time. Also multifactor authentication must be present in the system, to protect stolen devices and access from new locations in cases of stolen accounts.
The user should have access to all his clinical documents history and logs. Questions such as who, when and where his his documents were accessed, and who changed them it must always be recorded and available to the user. There are some cases when time is precious and cannot be lost in set permissions into mobile phone. In cases of extremely emergency, such as an unconscious patient in an ambulance that cannot give access to objects, the BTG mechanism needs to allow emergency access to the health professionals. This BTG access must be well defined and always recorded in the logs system. Also data loss can not happen, secure backups must be done in short periods of time.
Moreover, all the system must be secure. The management of health data can be critical, all the end point and communications must be protected. Security mechanisms assure CIA empowering patient privacy. In section 5.2 the security requirements are analyzed and explained.
According to 1.1, and after a deep and structured study and analysis of global smartphone OS market share, it was decided that the initial OS target for the final implementation and test of this research is Android [161].

4.2.1 Use Cases

In this subsection use case diagrams are presented and explained. Only the most important use cases diagrams are presented. Use case diagrams represent how a user interacts with the features of the system and provides a graphical overview of the functionality provided by a system in terms of actors. A *Patient*, with the android application myHEnCE installed in his device, wants to experiment the full options of this application. The diagram in figure 4.1 shows the generic use cases for the *Patient*. Initially *Patient* needs to perform login. The login includes a previously registered account (more explained in the next subsection). Depending if the location and device IMEI are new, the login extends a multifactor authentication protocol. After the login phase, the *Patient* can choose between the options i) add new device or location, ii) read messages, iii) create PHR, iv) create new relationships and v) view available EHRs. Each of the options have subsequent options, that will be explained in detail in the sequence diagrams, at 4.2.2.

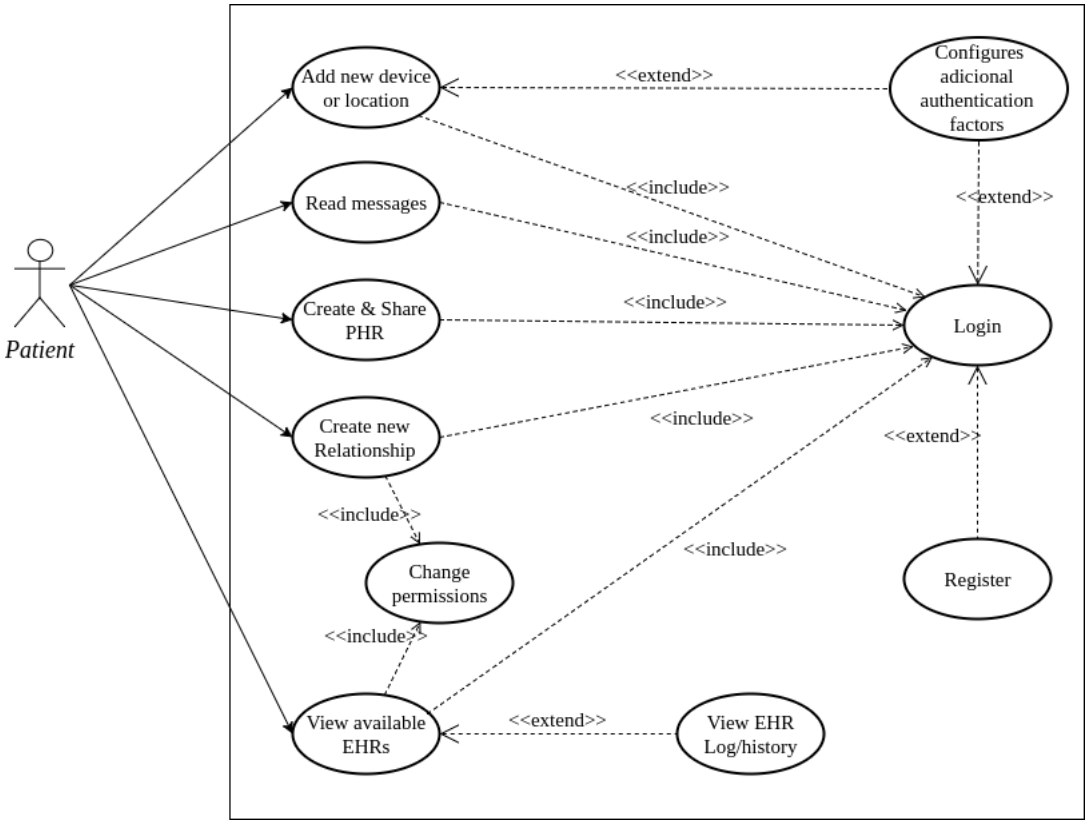

Figure 4.1: Patient use case.

The diagram in figure 4.2 shows the generic use cases for a health professional, in this case *Doctor*. As first phase the *Doctor* needs to perform login. The login includes a previously registered account, which is responsibility of each *Federated Institution* (not approached in this work). In the login phase, a *Doctor* always needs to perform the multifactor protocol. To do this he needs to insert his medical card in the card reader. In the first tests, the *Doctor* only needs to insert the card number as multifactor. It was assumed that the *Doctor* device is provided and registered by the *Federated Institution*. After login, a *Doctor* can choose a patient from his list or search for a patient. He can see the generic information of the patient after selecting him. After this some options are presented to the *Doctor*, if he doesn't have access to the selected patient EHRs, he can request permission. Also *Doctor* has other options, such as the use cases i) generate registration QR code, ii) create new EHR, iii) create new message, iv) view patient PHR, v) view patient EHR, and others in the diagram.

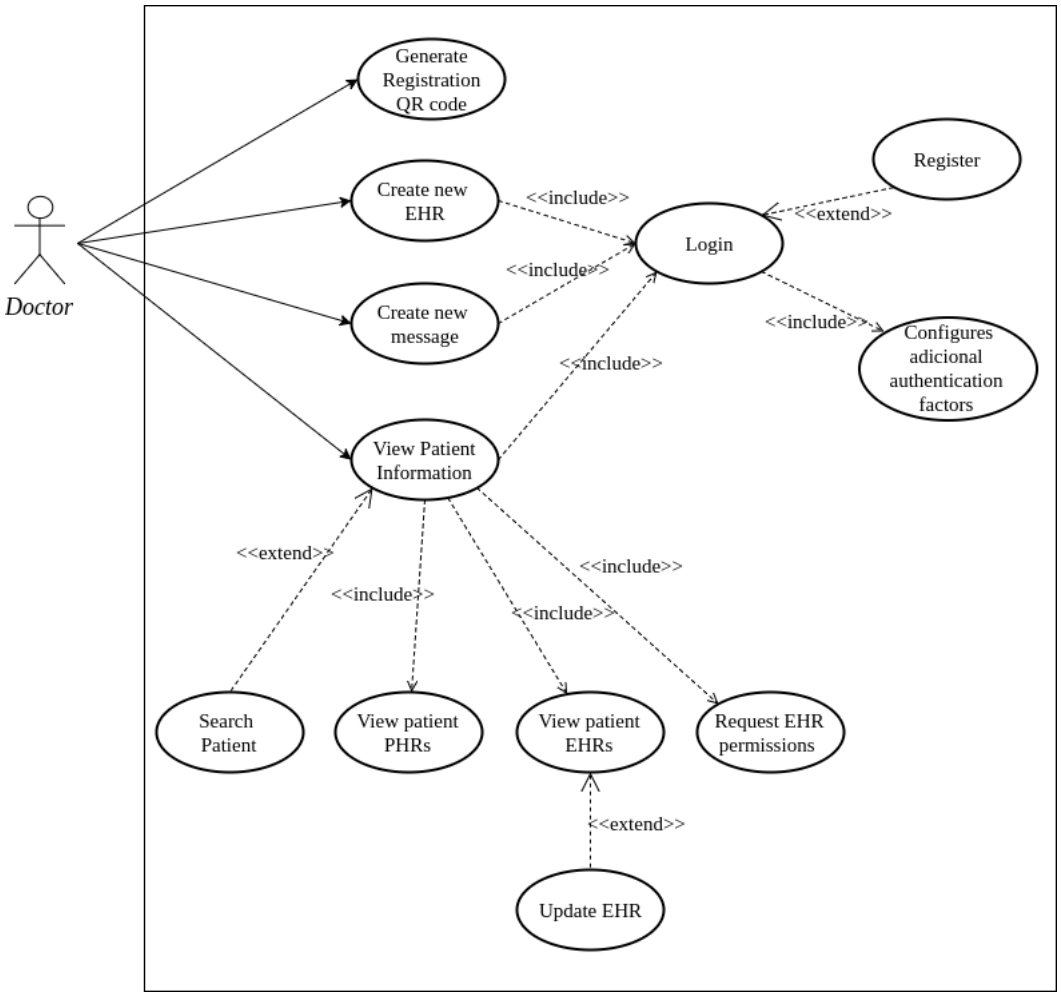

Figure 4.2: Health professional use case.

The diagram in figure 4.3 shows the use case for *Doctor* to create a new EHR. This use case assumes that the login phase was successful. To create the EHR, the doctor needs to i) insert patient identification, ii) insert institution, iii) define episode attributes iv) choose treatment and/or prescription, v) insert resume/observations and vi) choose the sensitivity of the new EHR. The *Web Service* manages to get and define the other necessary data, such as i) recognize date and hour, ii) identify the health professional, etc.

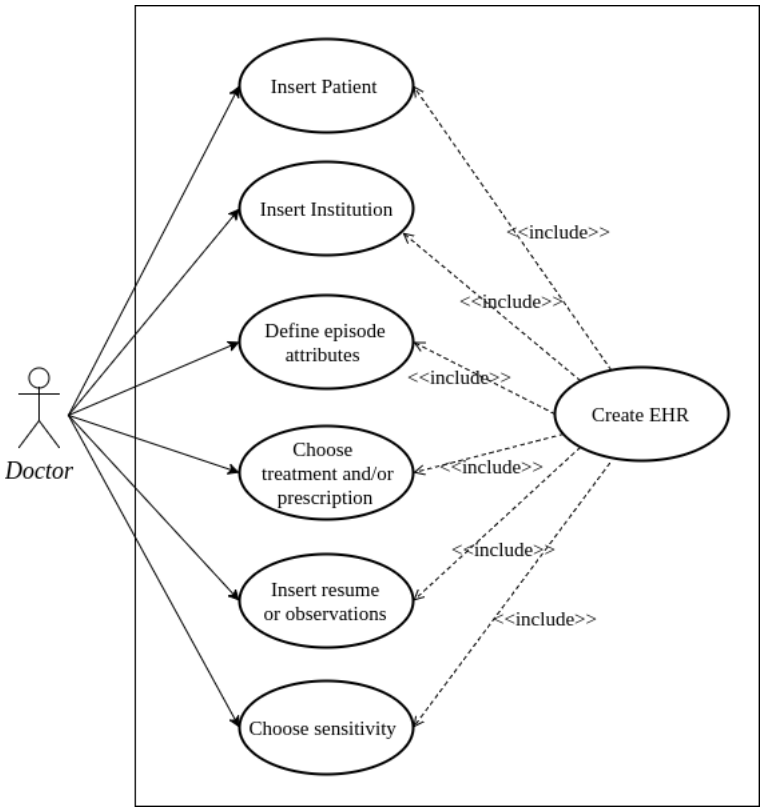

Figure 4.3: New EHR use case

4.2.2 System Sequence Diagrams

To better understand the interactions and processes between all parts of the system, it was chosen the use of system sequence diagrams (in detriment of the simple sequence diagrams). The elements participating (exchanging messages) in a system sequence diagram are Actors and/or Systems. The messages exchanged by these elements could be of any type depending on the systems (from web service calls to data input from a human). The security requirements, measures and implementations in the system will be explained in chapter 5.
The diagram in figure 4.4 shows the sequence for download and installation of the Android application.

The diagram in figure 4.5 shows the sequence of a *Patient* registration in the system. As pre-requirements the *Patient* needs to have the application installed in his device. Also the *Patient* needs to physically go to a *Federated Institution* to request a registration. With this, the identity is assured with *Patient* presence and Citizen Card (CC). An authorized *health professional* verifies the identity of the *Patient* and inserts the necessary identifiable data (e.g, email, name, CC number) to fulfill the registration request. The device checks if the *health professional* GPS

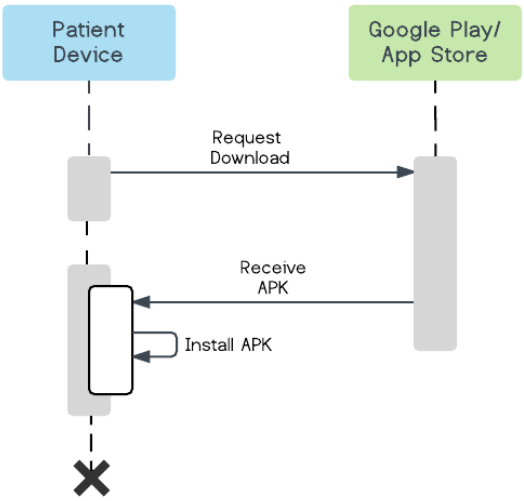

Figure 4.4: Install application sequence diagram.

location matches the respective institution location. After the *health professional* device generates a QR code with the *Patient* data, and the URL to complete this process. The *Patient* uses his device to read the QR code and the device automatically does the final request to the IdP. And with this auto-enrolment the registration process is concluded.

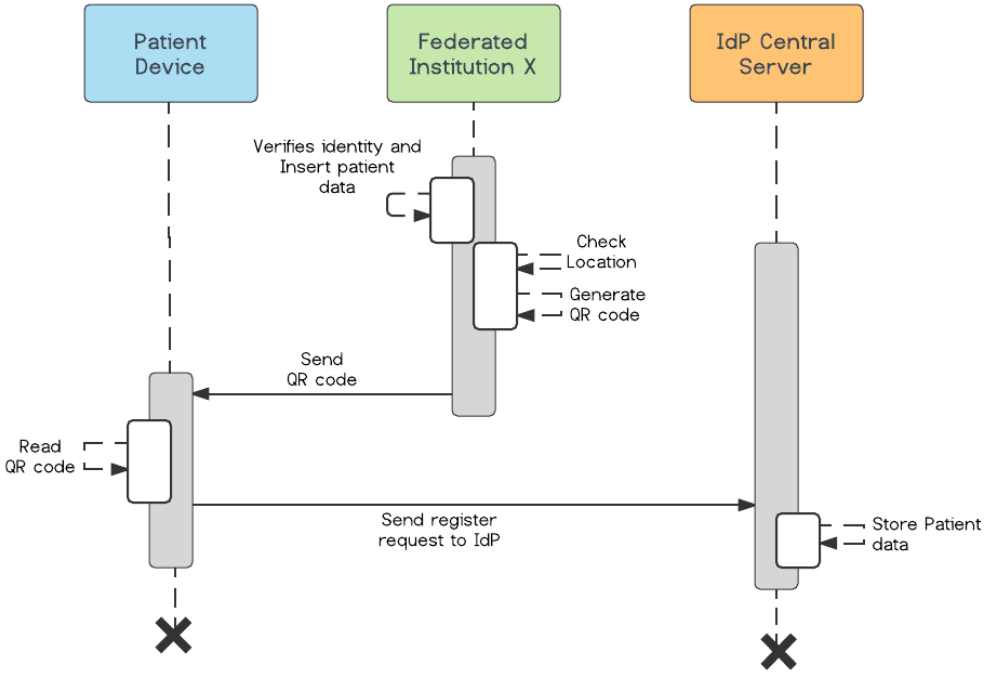

Figure 4.5: Patient registration sequence diagram.

The diagram in figure 4.6 shows the sequence of a *Patient* login in the system. As pre-requirements the *Patient* needs to have a registered account. In the first step the *Patient* requests login, sending his login credentials, also the mobile application collects the GPS location and device IMEI (to identify the device). The IdP validates the login credentials. It also checks the *Patient* profile to see whether the device and location are already known. If negative, it will be needed a multifactor authentication, explained at figure 4.7. If all positive, SoTRAACE risk evaluation is performed, which dynamically changes all the queries that are made to all *Federated Institutions* where the *Patient* has health information and EHRs. For example, if the risk is considered high, the name of the institutions can be omitted from the query and from the query result. After querying the *Federated Institutions* where the *Patient* has data, the data is aggregated at the IdP and merged to create a list. At last, a cryptography Authorization Token (AT) is generated for *Patient* authentication and authorization through all SP. This AT is stored at IdP. The list and the AT are sent to the *Patient* device.

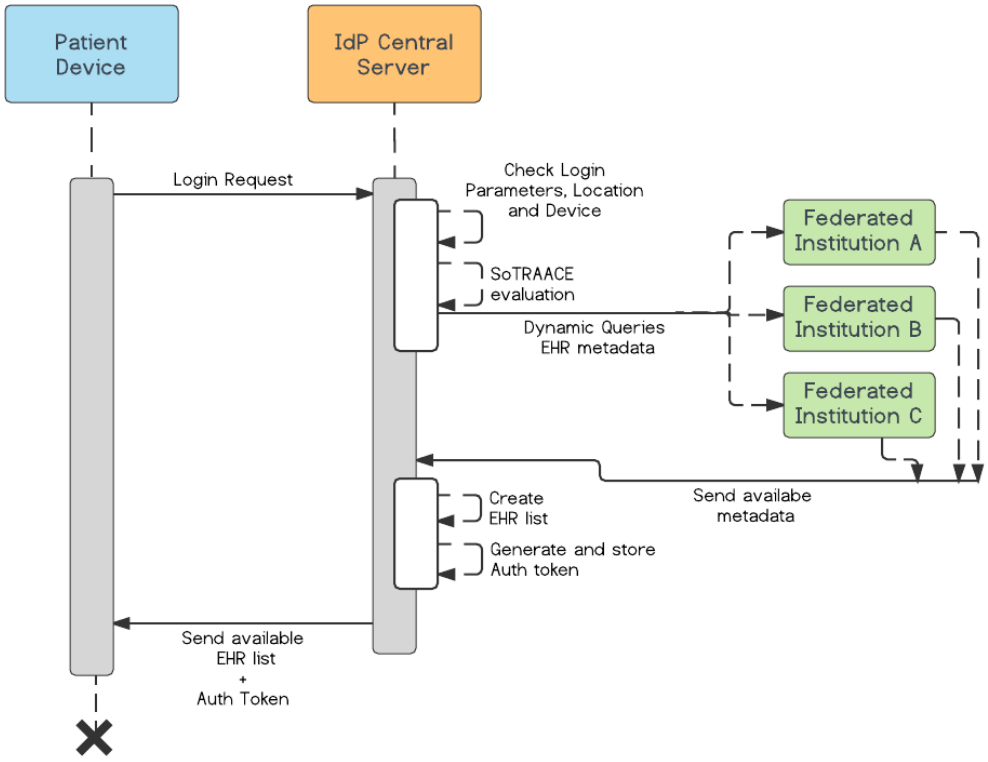

Figure 4.6: Patient login sequence diagram.

The diagram in figure 4.7 shows the sequence of a *Patient* login in the system, with the obligation of a multifactor authentication. The *Patient* receives a random secret PIN in his email, each time the multifactor is required. When the *Patient* requests login, the IdP verifies if he is using a new, non-registered device or location, that are not in his profile of past accesses. IdP requests a multifactor authentication to the *Patient*. The *Patient* check his email, and sends the secret PIN to the IdP, which verifies this secret authenticity. If everything matches, IdP stores the new device IMEI or/and location in the user profile, and notifies the *Patient*.

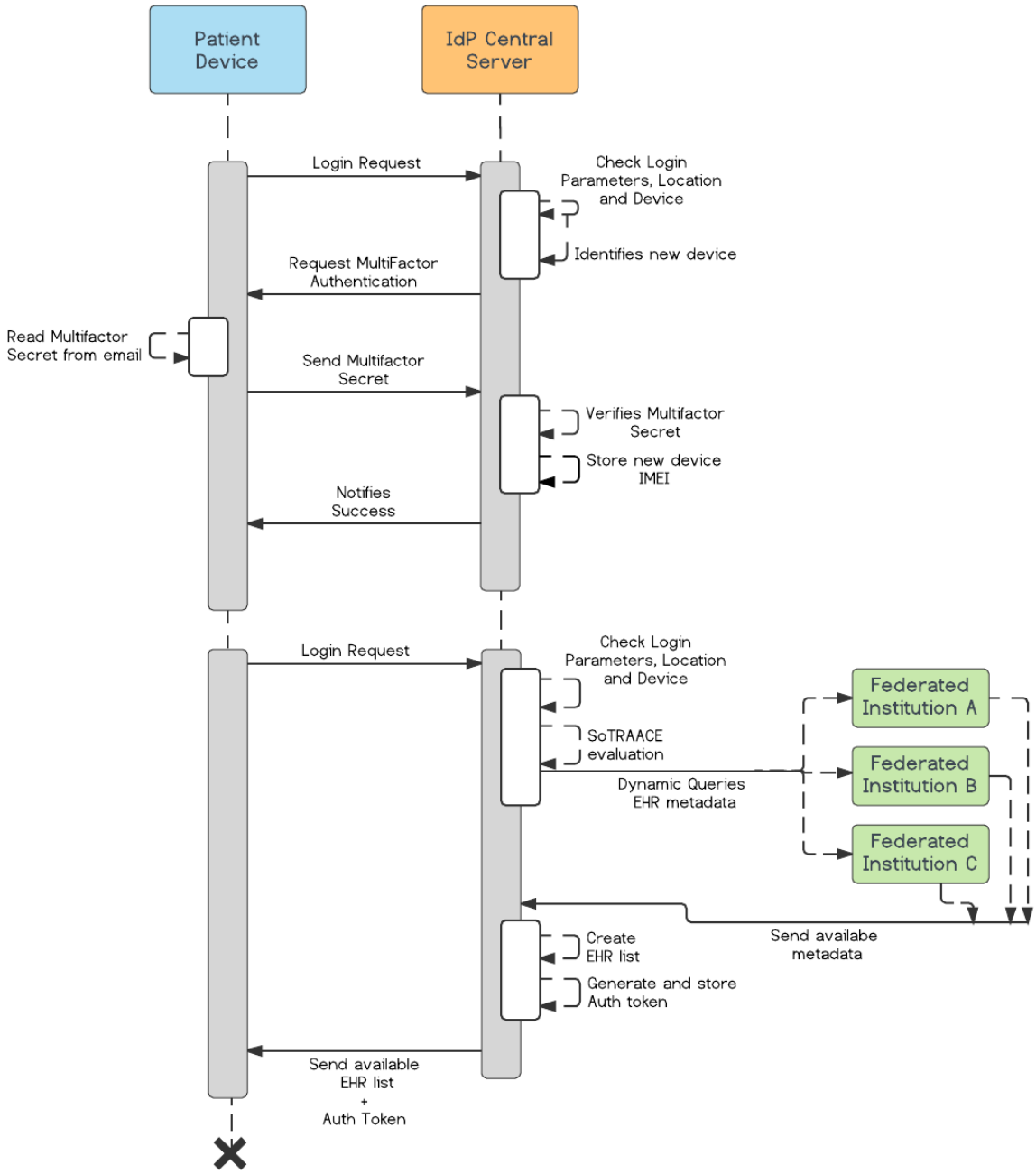

Figure 4.7: Patient login with multifactor sequence diagram.

One of the objectives of the system is to ensure privacy of the *Patient*, and that he has total control over his data and EHRs/PHRs. SoTRAACE is the access control model for the implementation. This model needs user data at each request (e.g, location, connection, etc) to perform the risk evaluation and the final access decision. However, to assure privacy, these personal informations can not be on the *Federated Institutions* (SP) side. Sensitive personal information is always stored at IdP. Also each request generates a log, to help build the *Patient* profile, and enable audit. Considering this, the figure 4.8 shows the sequence of a *Patient* choosing an EHR from the initial list. This request goes with an AT directly to the *Federated Institution*. The *Patient* device sends the necessary data to the IdP for SoTRAACE do his work. The *Federated*

Institution validates the AT with the IdP. If the AT is valid, a record is created in the log system. After this SoTRAACE evaluates the risk for that request and adapts the query (e.g, if the risk is high, some parts of the EHR are omitted). Finally the EHR is sent to the IdP, and the IdP determines the best protections and access decision based on SoTRAACE, and sends it to *Patient*. The *Patient* views the EHR and changes the access permissions. Those permissions changes are updated in the IdP SoTRAACE ACL and at the *Federated Institution* ACL. Finally the *Patient* is notified about the success/failure of his changes.

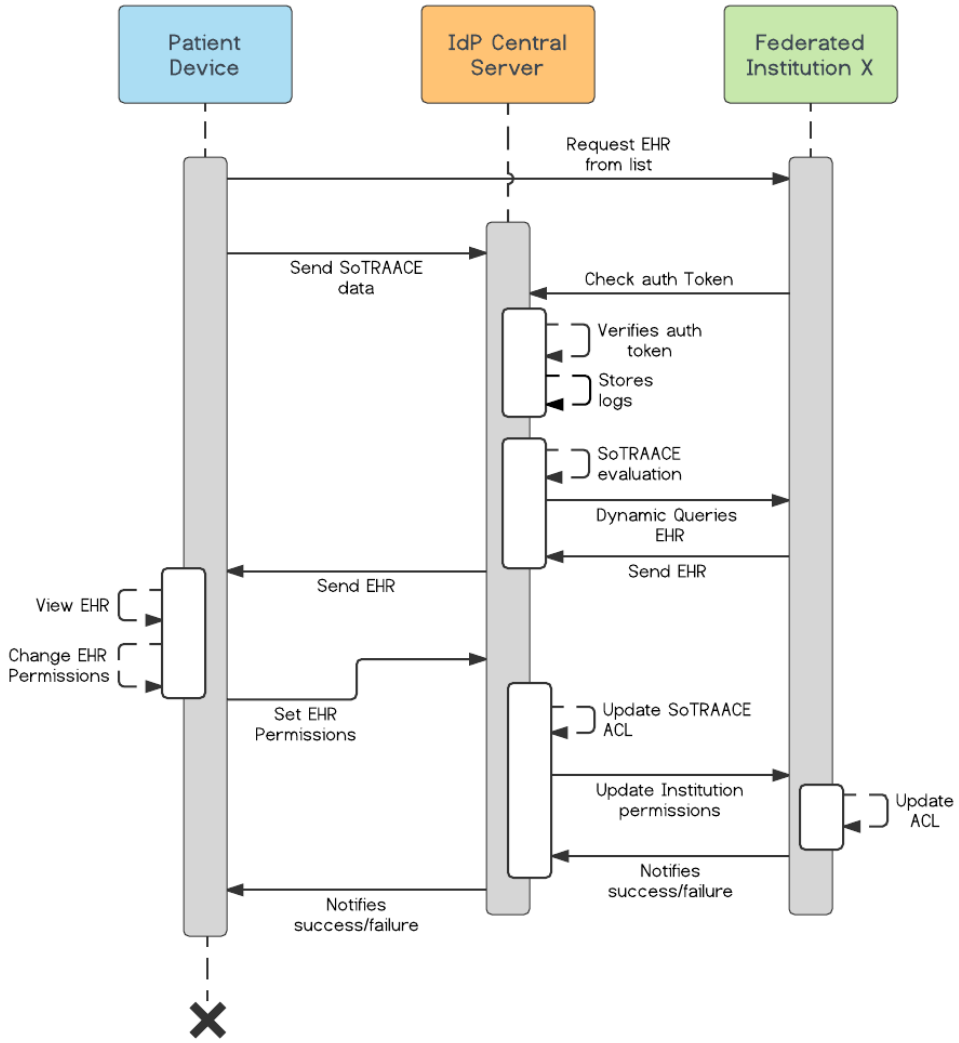

Figure 4.8: Patient EHR access with permission change sequence diagram.

The diagram in figure 4.9 shows the sequence of a *Doctor* in the process of requesting access to a *Patient* EHR or PHR. As a pre-requirement the *Doctor* needs to perform login in the system, with multifactor authentication. To do this he needs login/password combination, and to insert his identifiable doctor card in the card reader. For first tests and simulation, this step will be simulated just by the doctor inserting the number of the card manually.
Then the *Doctor* searches for the *Patient* he wants do request data access from. After that he chooses the type of health record and the area (e.g, orthopedics, radiology, general). The access request is sent to IdP, and is enforced a new multifactor authentication (this protects

against the use of lot time open computer sessions). After that the IdP validates this process and the access request is sent to the *Patient*. Upon visualizing it the patient can accept the request. The new permissions are stored in the IdP ACL, and *Doctor* is notified about the *Patient's* decision. In the case of an EHR, the permissions are also changed at the *Federated Institution* ACL. In the case of a PHR, which is stored at *Patient's* device and IdP, the only ACL is at the IdP. At last, *Doctor* sees the EHR/PHR.

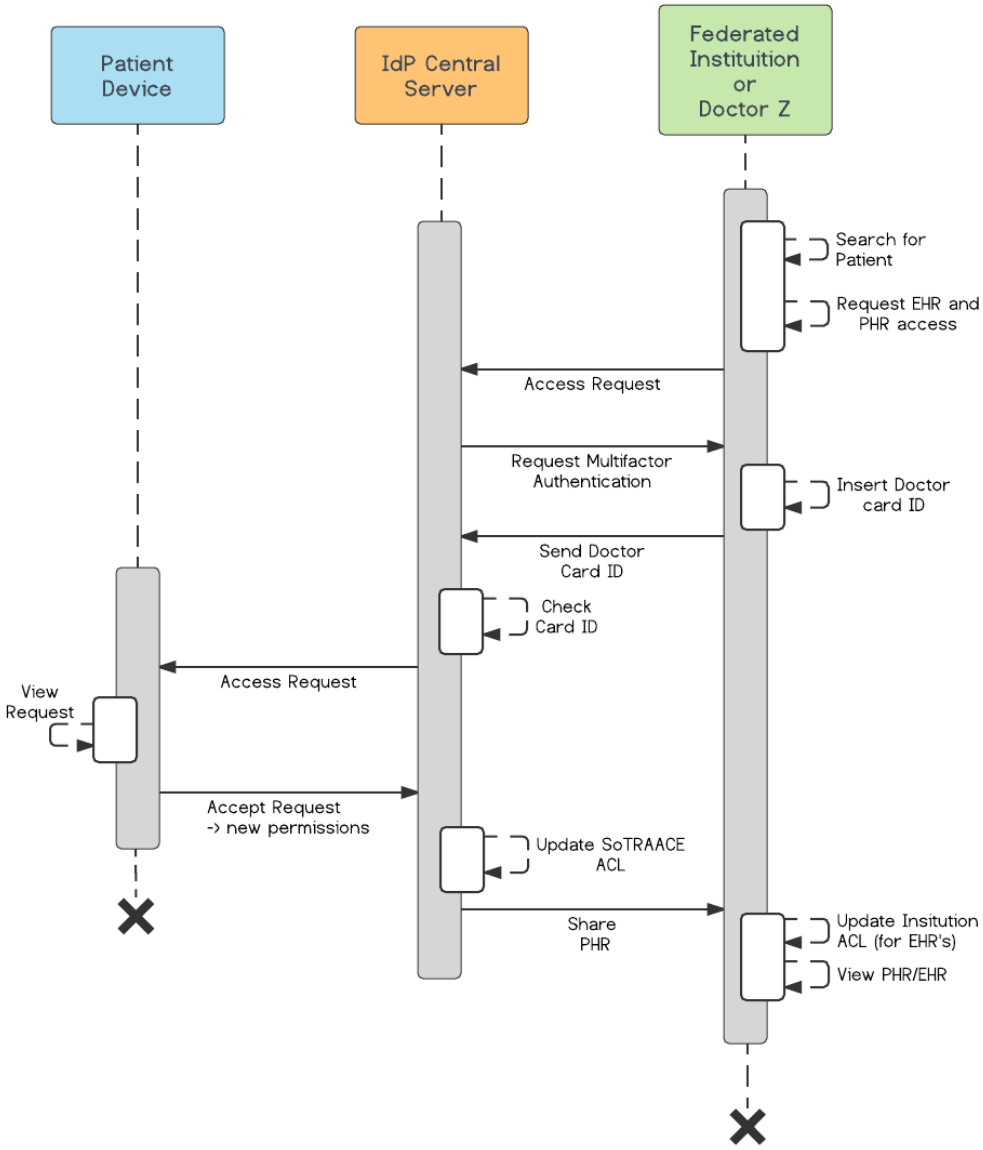

Figure 4.9: Health professional EHR or PHR access request sequence diagram.

The diagram in figure 4.10 shows the sequence when a *Patient* creates or updates a PHR and sets permissions for it. After create/update and set/change permissions, the mobile application contains SoTRAACE that tests the risk and allows or denies the sharing of the PHR. After this evaluation, the PHR is shared and stored in the IdP, as well as the permissions in the ACL. Finally each related health professional is notified about the share or removal of permissions.

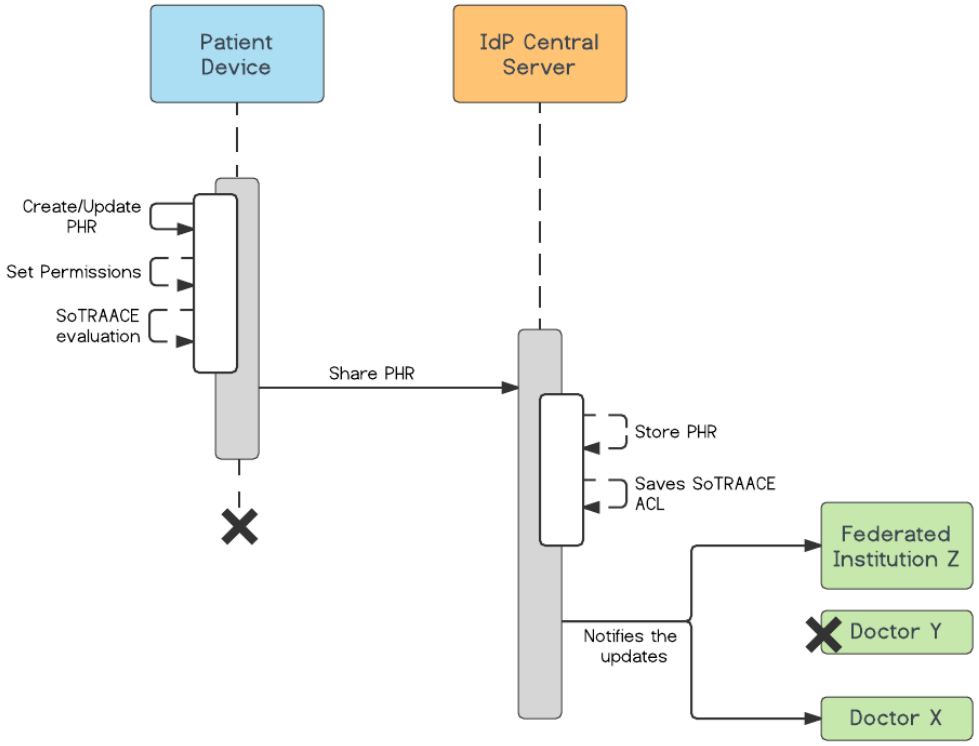

Figure 4.10: Patient share PHR sequence diagram.

4.3 System Architecture

The goal is to design and implement two Android applications, with web based central authentication and authorization IdP, to secure access and share health data stored in data bases of geographically fragmented SPs.

In the mobile node it is needed the use of SQLite database to store the user PHR (and later enable the sharing). For the risk-adaptable decisions in the access control layer, the android application needs to manage data about user locations and connections. Also location is used in the authentication layer to identify the user. The alert system warns the user about some important aspects of his interactions. It can be used to warn about the risk of the access to a specific data in a dangerous context, can release alerts to teach the patient to get better decisions and security, warn about the existence of a new EHR or a change in one, inform the user of new access requests, new messages etc. The message system is unidirectional. Only the health professionals can send messages to patients. If they want to keep an ongoing conversation, via this message system they can share his/her email or phone number. This way is avoided the patient sending an exorbitant number of messages to the health professional. Otherwise the flood of messages will reduce the user experience of the health professionals.

Each IdP contains an authentication layer (to manage authentication of users and control their identity), an access control layer (based on SoTRAACE), a SQL database to store the ACL permissions, assist the layers of authentication and access control and to store user profile (with all past requests and attributes). Also the IdP contains a log system, to enable audit.

SPs use SQL databases to store the health data and logs. In these databases are also stored the ACL permissions, that are synchronized with the main service, the IdP. In each SP the respective logs for later audit are also stored. It is important to store information to a better version of

control over the health data (e.g, who, when, where it was changed the data).
In figure 4.11 the generic architecture is graphically established, with the respective representation of the different types of communications that are used.

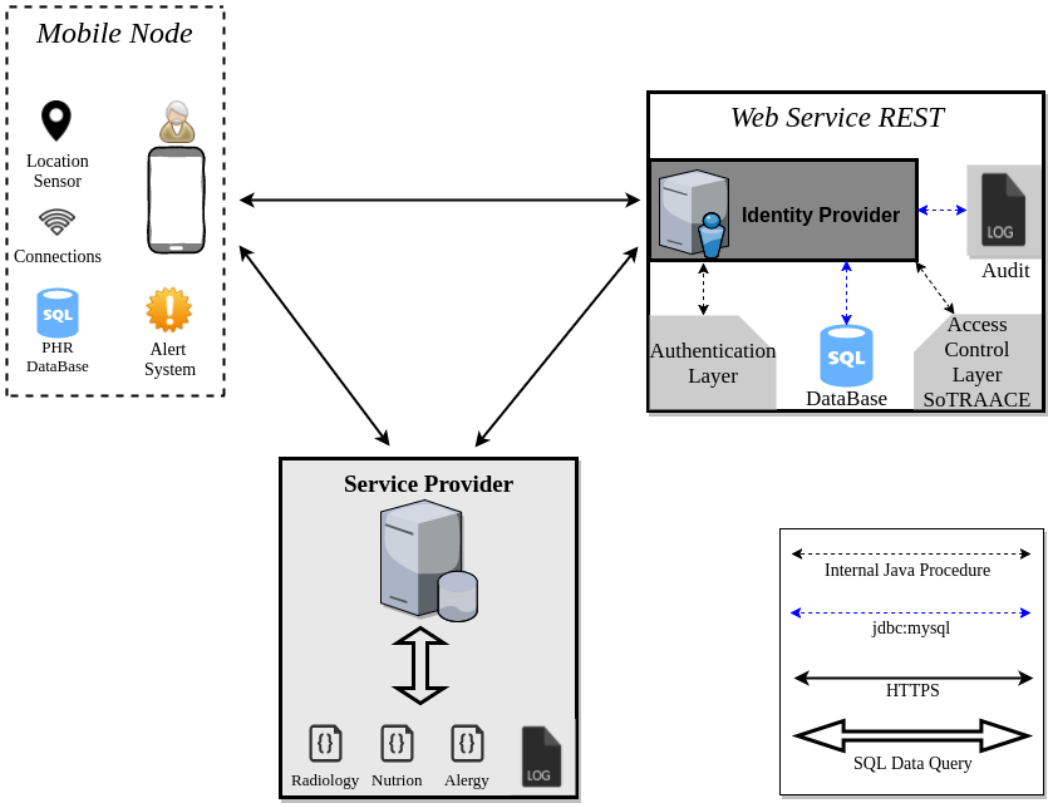

Figure 4.11: Generic System Architecture.

4.3.1 First Enrolment and Keys Exchange

To be able to access the medical data and use the myHEnCE application the patient needs to have a pre-registered account. To do that he needs to present himself in a federated institution and bring his CC, this way we assure the identity of the patient in this first enrolment. Then the patient meets the doctor, and starts the registration process. Along with his presence and CC, the patient needs to define an authentication password.
A key concern when using passwords in authentication is password strength. A strong password policy makes it difficult or even improbable for one to guess the password through either manual or automated means [61]. Following OWASP [61] rules and NIST [40], password length should be at least 10 characters, and complexity must have numbers and letters upper and lower case. Also it is critical for any application to store passwords using the right cryptographic technique. The chosen password by the patient must be protected. Therefore, the password storage needs to be protected with a strong cryptographic hash function from the start(more explanation about secure password storage at 5.4).
Next the IdP receives a request for a new patient and a QR code is generated. The QR codes are displayed at SP computers or at the health professional myHEnCEPRO for an auto enrollment process of smartphones into health institutions. QR codes are a very convenient way of conveying a reasonable amount of secret shared information to a smartphone that would otherwise be very cumbersome to input by hand by the user. This usage of QR codes to share secret information between eHealth systems and smartphones (e.g, cryptographic keys, certificates, salt), can in

a way, be seen as the establishment of a rather new special security layer by taking advantage of the analog security properties of the optical channel that is employed during the scanning of the QR codes by the smartphone. In other words, the QR codes can be used to simplify and make practical the enrollment process between the web service, institutions and the user's smartphone.

Also at the moment of the registration, a pseudonym is generated for the patient. The pseudonyms are used in the SPs to identify and associate health data to anonymous users. Thus, the user privacy is enforced, by hiding the true identity of the data owner, that can have EHRs with personal and important information. Only the IdP can map the pseudonym to the real owner. The IdP is the only that uses the CC to identify the patient. Considering *H* a cryptographic hash function and a salt as a secure random number (more explanation of secure random numbers at section 5.3), the long pseudonym can be defined as:

- String longPseudonym = H (H(CC||name) ||salt)

Only secure and updated cryptographic algorithms are used in the system framework. A secure hash function should have at least 256 bits (64 characters), which is a huge string to identify each user. Thus, the first ten characters are extracted from the long pseudonym using the subString method:

- String pseudonym = longPseudonym.substring(0,10);

Note that, if there is a hash collision for these ten characters, the system will generate a new salt and calculate a new pseudonym, until it finds an available value. In figure 4.12 the registration architecture is graphically established:

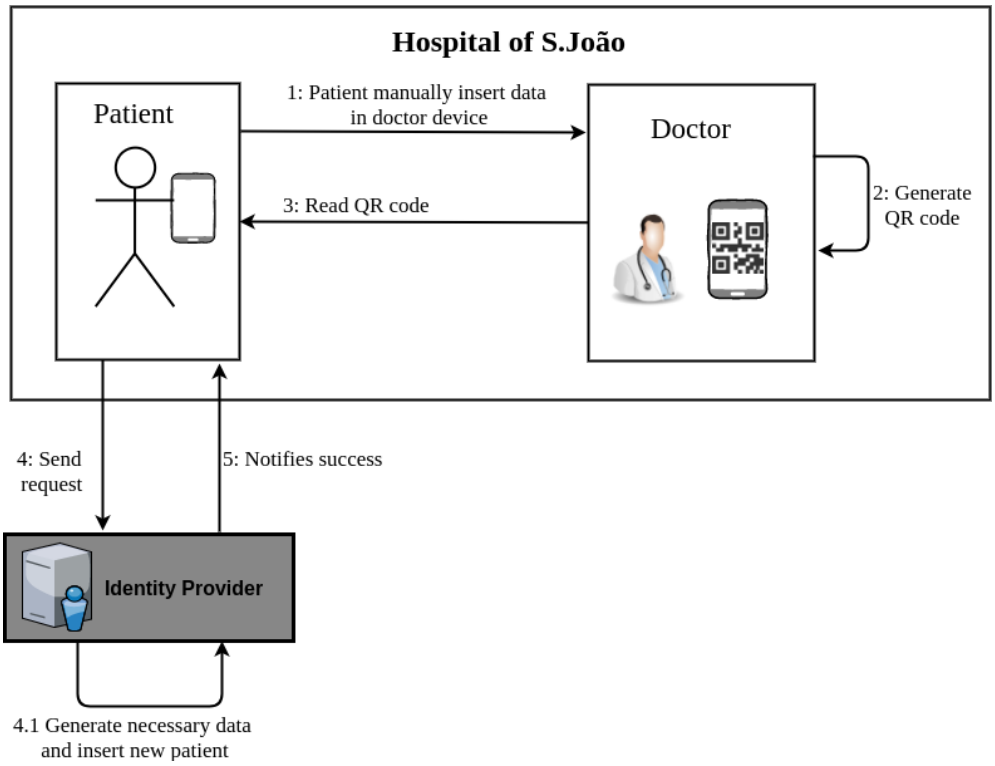

Figure 4.12: Patient's registration architecture.

The IMEI of the device used in the registration process is automatically added to the patient's allowed devices.

4.3.2 Authentication and Authorization Architecture

The authentication architecture follows the rules in OAuth2 protocol [79] and the OWASP cheat sheet for authentication [61]. Central authentication is made at IdP, which is a secure trust system that user can rely on. This way users don't need to worry about all institutions credentials security and storage. Users credentials are not scattered over the federated institutions, and the passwords are stored at IdP. SP doesn't know any of this. It just validates the AT with the associated pseudonym.
The pseudonym is generated in the registration process. The login phase with credentials, is protected with the CHAP to use OTP (more details at section 5.6). In this step is also sent to the IdP the IMEI of the user mobile and his location. Based on [124], location is used as attribute for authentication. A range of distance can be defined to a more adaptable system authentication, for this work it is considered a range of 50 kilometers. The IMEI is also used to validate authentication. A user can have multiple devices associated to him. Although that, each time the user accesses from a unrecognized IMEI or outside the location range, it needs to perform a multifactor authentication (more details at section 5.6).
After validating the credentials and attributes, the IdP delivers a AT, and that AT is used to request services in all the federated SPs. The user doesn't need to insert credentials for that session ever again. The SP checks the authenticity of the AT with the IdP. The IdP validates or not the authenticity of the AT and sends the answer to the SP, that allows or not the service requested.
So the AT can be seen as a secure digital object that an authorized person possesses and presents it to have direct access to his or other person resources.
Based on [79, 131, 73], an AT request, made in the authentication phase, uses the following request parameters:

[grant_type] : The type of credentials authorizing the request for an AT. This parameter must have a value of either *password* or *refresh_token*. For this initial implementation it will only be used the *password* value.

[client_id] : The automatically-generated unique ID of the client application requesting the AT.

[client_secret] : The shared secret string the instance and the OAuth application use to authorize communications with client-server. The client_secret is a secret known only to the application and the authorization server.

[username] : The user account name that wants to be authenticated to obtain the AT.

[password] : The password for the user account that wants to be authenticated to obtain the AT.

For the previous request, the web service produces a JavaScript Object Notation (JSON) response containing the following parameters as name:value pairs.

[scope] : The amount of access granted by the AT. The scope is always *useraccount*, meaning that the AT has the same rights as the user account that authorized it.

[token_type] : The type of token issued by the request as defined in the OAuth RFC. The token type is always *Bearer*, as defined in OAuth RFC.

[expires_in] : The lifespan of the AT.

[access_token] : The string value of the AT. Considering *H* a cryptographic hash function with 512bits, a salt as a secure random number (more explanation of secure random numbers at section 5.3), the AT is generated as follow:

- AT = (pseudonym || role || H(pseudonym || IMEI))

An authorization request answer example using the AT in JSON format:


```
{"scope":"useraccount",
"token_type":"Bearer",
"expires_in":1010,

"AT":"201AB241SPATIENTkB4nNd4tfZ2jp0shdcDVVr1tbMYgkfJBrJcgawZMJZsBwJ9L8usttbjhSqMZ
kKTj+jrHkqJeosvpjGdEqyowwQ=="}
```


With the initial nine characters '201AB241S' the web service can identify the user pseudonym, and then validate the combination. Thus, when the user performs a request to the SP, it will only send the pseudonym provided by the IdP. If the patient logs out or the session expired, the token is revoked. The patient has only one AT per session. In local appointments with the need of identifying patient data, the SP needs to contact the IdP to validate the pseudonym of the respective patient. Thus, only the IdP knows which data belongs to whom.
To grant authorization to other users or health professionals to a determined resource, the user application generates a new AT.

- AT = H(patientCC || DoctorID || salt)

. The new AT is stored in the ACL database, to ,when needed, perform double authorization check(AT and ACL.


```
{"scope":"useraccount",
"token_type":"Bearer",
"expires_in":1010,
"AT":"HTCS+dAJO+y36WzOXv9GuswrE1Gv6a0uexSGAe5Xyx4Jr2z3fw8r5jU2My+zTaf6kTTxKnmuUM
xQtPlqvnlNyg=="}
```


These previous ATs, are recorded until the patient revokes then or they expire. Each health professional has a list of tokens, that are recorded in a separated table in the database. To access an EHR, the authenticity of the AT is verified by performing a semantic match between the health professional granted ATs (in the table of health professionals granted tokens) and the AT + ACL in the authorization table (to that specified EHR).
The authentication and authorization relies on TLS encryption to protect the user's credentials and AT during transmission (more details about TLS at section 5.5).

In figure 4.13 the authentication architecture is graphically established, with the respective representation of the different types of communications that are used.

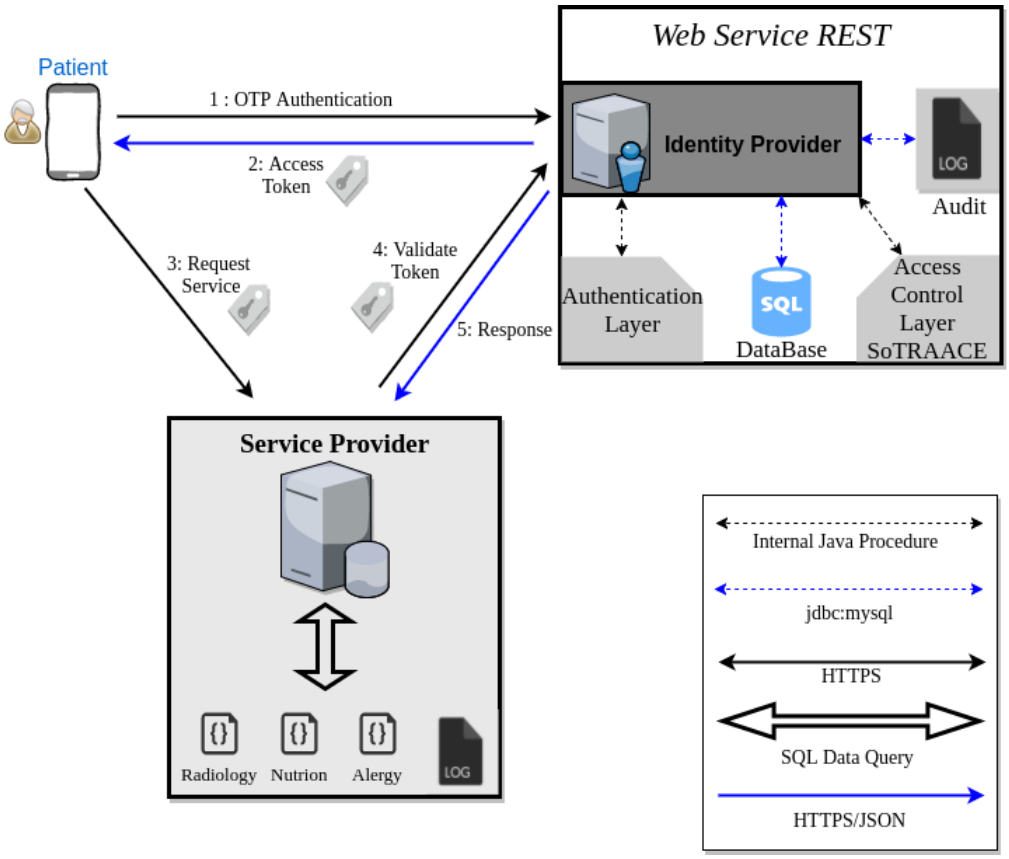

Figure 4.13: Patient authentication architecture.

4.3.3 Web Service

The health data in the system has the necessity to be synchronized between intuitions databases, mobile device applications and main databases. A Web service [162] is a method of communication between two electronic devices over a network, it is a solution used in systems integration and communication between different applications. With this technology it is possible that new applications can interact with those that already exist and that the systems developed on different platforms are compatible. Essentially, a Web service uses the features of the software application available over the network in a standardized way. Distributed Web services features or services can run on different hardware or OS, can be written in different programming languages and with different technologies. Two of most common web services styles of use are the Simple Object Access Protocol (SOAP) and REST.

SOAP [163] is a lightweight protocol for exchange of information in a decentralized, distributed environment. It uses XML to format the messages, and consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses. SOAP has specifications, such as Web Services Description Language (WSDL), and can potentially be used in combination with a variety of other protocols, such as HTTP.

REST was introduced and defined in 2000 by Roy Fielding in his doctoral dissertation. REST [164] is an architectural style for designing distributed systems. It describes a set of architectural principles by which data can be transmitted over a standardized interface (such as HTTP). It is not a standard but a set of constraints, such as being stateless, having a client/server relationship, and a uniform interface. REST is not strictly related to HTTP, but it is most commonly associated with it. RESTful is typically used to refer to web services implemented in REST architecture.
Principles of REST [164]:

- Resources expose easily understood directory structure Uniform Resource Identifier (URI)s.
- Representations transfer JSON or XML to represent data objects and attributes.
- Messages use HTTP explicit methods to map CRUD operations to HTTP requests.
- Stateless interactions store no client context on the server between requests. State dependencies limit and restrict scalability. The client holds session state.

The key component of a REST architectural is that RESTful applications must be stateless. This means in a RESTful application no session state is stored on the server. All of the information needed to satisfy the request is carried in the request message itself. A client can therefore cache a representation of a resource, which can significantly improve the application's performance, where a service explicitly allows it.
Java defines REST support via the Java Specification Request (JSR) 311. This specification is called Java Application Programming Interface for Representational State Transfer Web Services (JAX-RS). In table 4.1 is a brief comparison between REST and SOAP, which clearly shows that REST brings more advantages.

REST	SOAP
Not XML protocol based	XML based messaging protocol
Without specifications	With specifications (e.g, WSDL)
Doesn't enforce message format, can be XML or JSON	Enforces message format as XML
Light weight – due to the usage of JSON	Heavy weight – due to the usage of XML
Easy to parse the response	Bit difficult to parse the response

Table 4.1: Main differences between REST and SOAP.

In most common cases, Web services use JSON and XML to data exchange. JSON is a lightweight text-based open standard designed for human-readable data interchange. It is less verbose than XML and more simple. This simplicity of JSON has resulted in its widespread use, especially as an alternative to XML.
Web service communications for mobile computing can result in unacceptable performance overhead. In eHealth time is crucial, overheads and low performances must be avoid. This potential problem comes from two factors. First, the encoding and decoding of SOAP XML-based verbose messages consumes resources, therefore Web service participants, particularly mobile clients, can suffer from poor performance. Secondly, wireless communication (e.g, wifi, 3G) are not so fast and efficient as wired communications. This is caused by restrictions on the mobile environment due to limiting the speed of the processor, the limited battery life and slow, unreliable and intermittent connections. For these reasons, the Web service architecture chosen (for the IdP and all SPs) is REST with JSON for data interchange.
For firsts tests the system has one IdP and two SP, namely Hospital Braga and Hospital São João-Porto.

A class diagram is a representation of the structure and relationships of all classes that serve as a model for objects. The IdP class diagram is presented in figure 4.14.

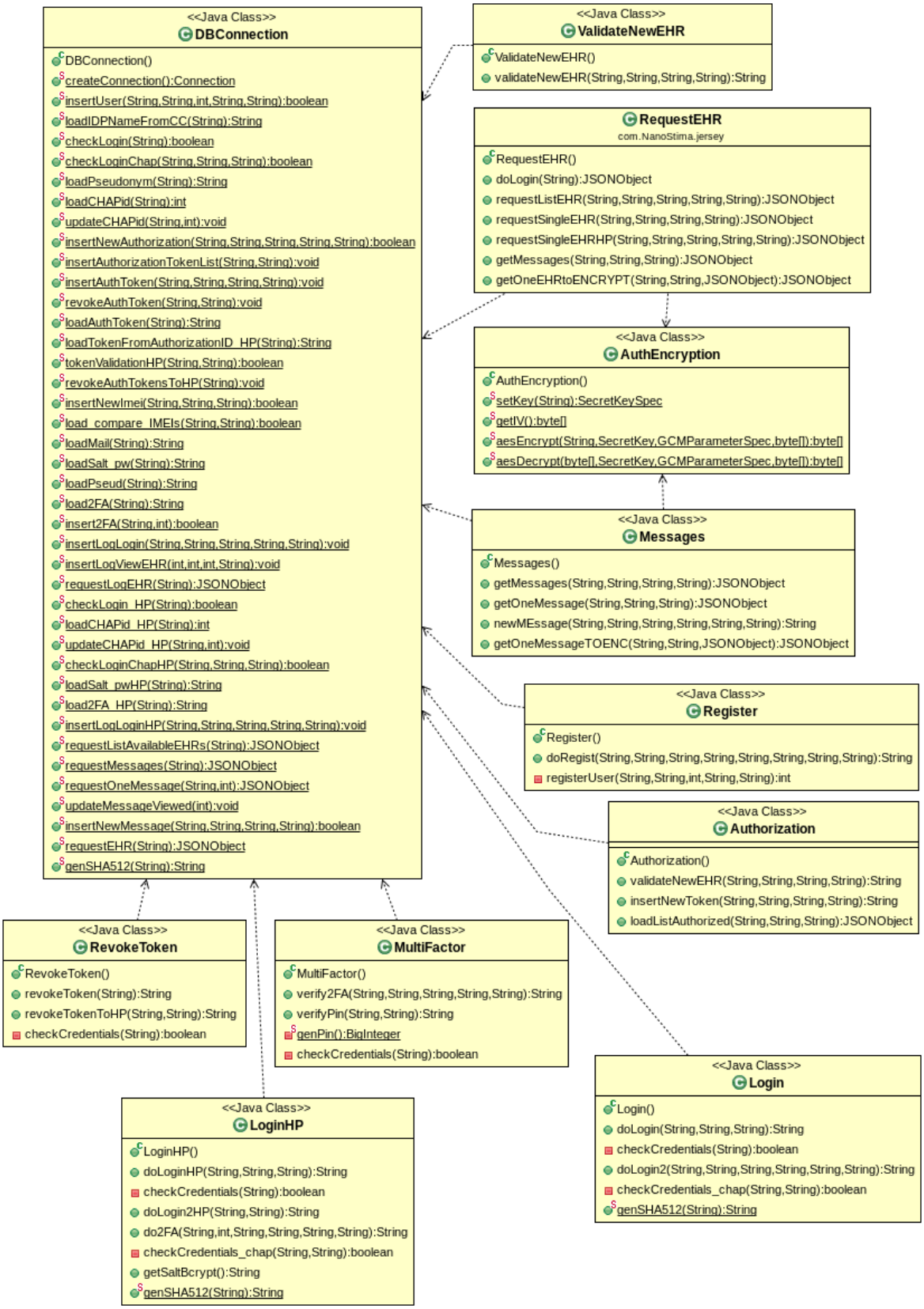

Figure 4.14: IdP class Diagram.

The SPs class diagram is presented in figure 4.15.

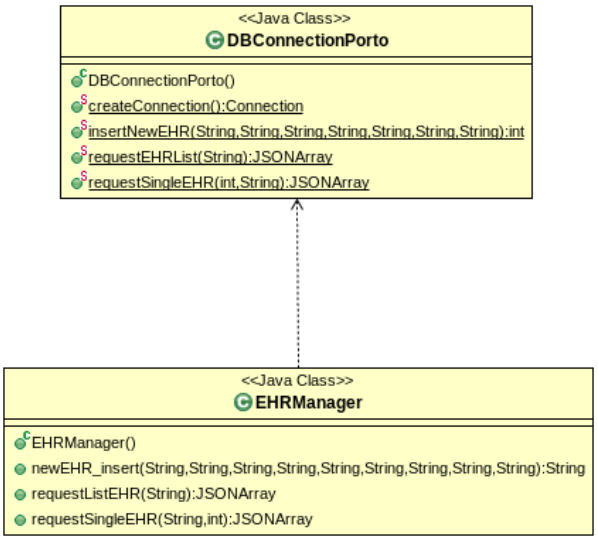

Figure 4.15: SP class Diagram.

4.3.4 Data Base Model

As approached in section 2.5, health data must follow global recognized standards, such as openEHR. In future work, and working with the company CINTESIS, the goal is to integrate this thesis framework (mobile applications and IdP) with standardized data, following openEHR norms. For initial tests in this work, each EHR in each SP has, along side with the identity of the patient owner and respective health professional who created it, just six fields: id, episode, treatment, resume, date and sensitivity. These fields are recorded in a database table along side with the owner pseudonym, the health professional who created, and the institution identification. The IdP that is the main focus of this work has a more complex data base. The IdP data base entity relationship diagram is presented in image 4.16.

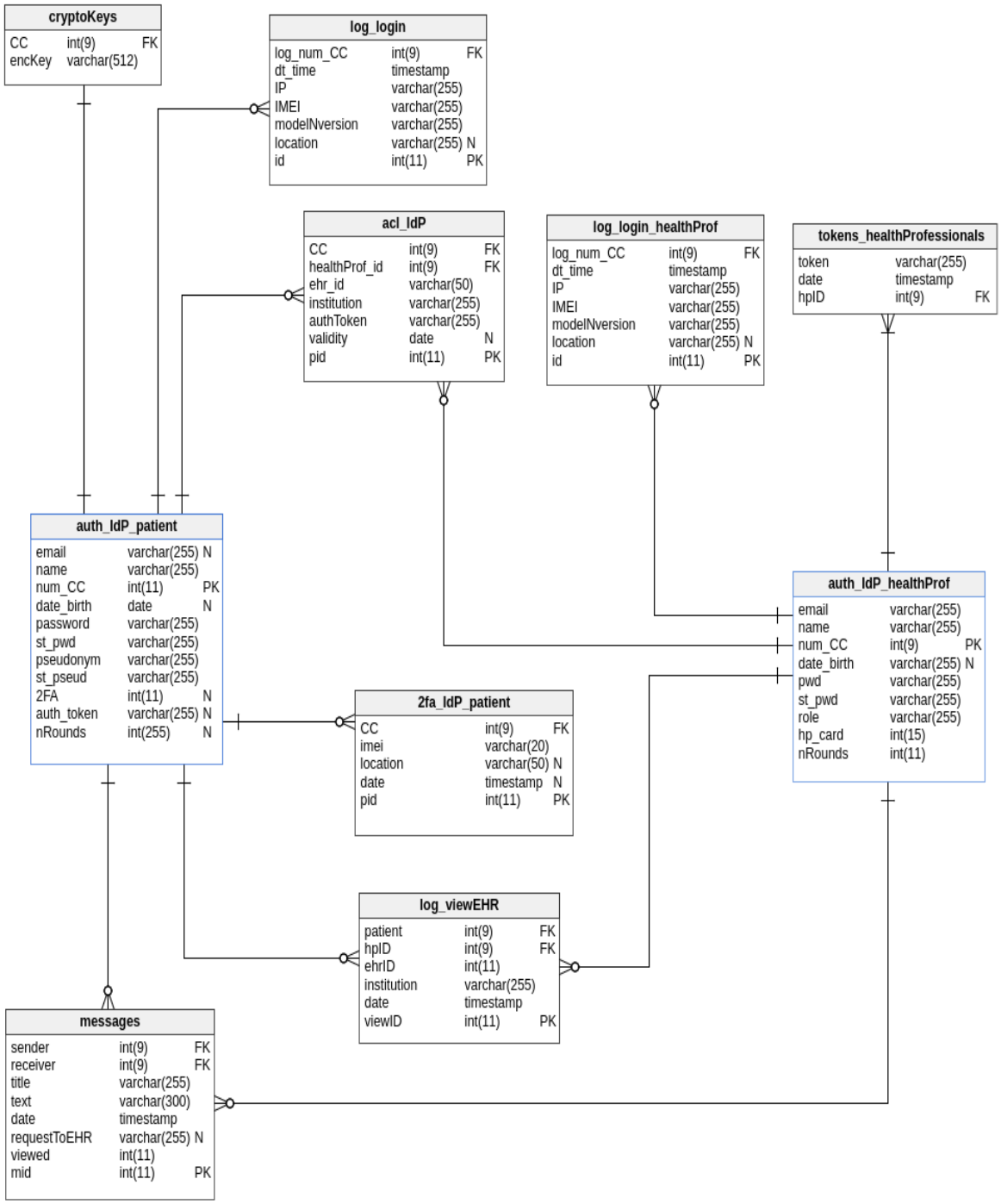

Figure 4.16: IdP entity relationship diagram.

The SPs data base entity relationship diagram is presented in image 4.17.

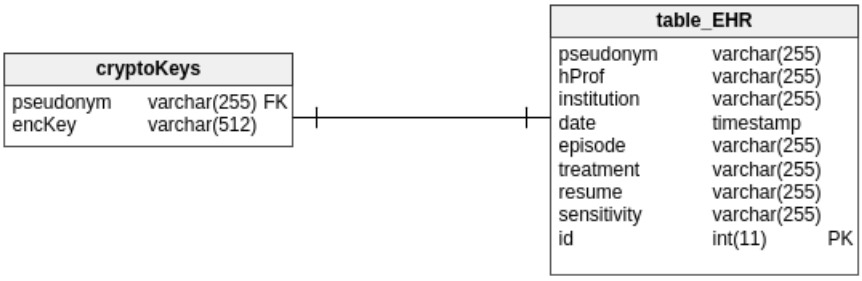

Figure 4.17: SP entity relationship diagram.

4.4 Conclusion

A good requirement analysis is the basis for a good system. The requirements were deeply reviewed and studied before defined. They are based in official legislations and in the discussion/reunions with some workers of CINTESIS (some related to health sector). The requirements are also based in some public speeches at CINTESIS, namely the ones performed by Dr. Hans Ossebaard, an international health care innovation advisor and also public speech named 'O Novo Regulamento Europeu de Proteção de Dados e a Agenda da Area da Saude', with many talkers, the most important was Dr. Isabel Cruz, the main secretary of Portuguese national commission of data protection. This chapter also present the integration in the system of the novel access control model SoTRAACE, published in this thesis context. The framework is modular and adaptable, easy to continuous integration and expansion.

Chapter 5

Security Analysis and Measures

5.1 Introduction

This chapter introduces the security solutions proposed to fulfill the security requirements in the framework. To fulfill the security requirements, secure and up to date cryptography algorithms must be used with unpredictable key with the correct sizes. This is extremely important, because no matter how many layers of security you have or how much resources you spend on it, if you are using unsecured algorithms with weak keys, all system is compromised. The choice of the security solutions relies on OWASP cheat sheets and NIST Special Publication 80057 [165]. To realize and understand why these security solutions were chosen, this chapter also includes a systematical theoretical and practical review of the respective solutions. Besides the security controls in the framework, a risk evaluation method for SoTRAACE, based in a Delphi study is presented.

5.2 Security Requirements and Attack Model

The most important aspect of the software design process is threat modeling, which include the analysis of security requirements and attack model. Security and privacy become more critical in remote systems where patient data needs to be shared among the medical authorities and doctors. Also, data acquired from the remote patient must be protected while being transmitted over the network. The security requirements were defined through the deep security review in all chapter 2, an analysis of OWASP [50] articles, ISO 18308 [166, 167] and ENISA health specifications [159]. To improve and optimize the implementation of the HIS it's crucial to identify and analyze distinct features of eHealth security and privacy, including security requirements from the healthcare perspective. The focus goes to the next security requirements:

1. Confidentiality: ensures that the medical information is only accessible to authorized users and protected from unauthorized access.

2. Integrity: ensures that received medical data stays intact when is at transit, and not changed by an adversary or any error.

3. Authentication: enables an health device to guarantee the identity of the user with which it is communicating. The use of multifactor authentication reinforces the authentication phase. Also users must identify the server, by means of a certificate.

4. Message Authentication: enforces that determined message comes, with integrity check, from whoever send it.

5. Non-repudiation: indicates that a node cannot deny sending a message already sent before. Or a health professional can not deny any change that he made to a file before.

6. Authorization and access control: only authorized users are available for view and use network services or resources.
7. Risk analysis: the access control model that will be used, namely SoTRAACE, needs a risk evaluation at the moment of each request to perform the access dynamic decision. Thus there is needed to use a good risk analysis method.
8. Fault Tolerance: A security scheme should continue with their security services even in the presence of a failure.
9. Semantic interoperability: enabling the ability to share data between systems that can be understood at the level of formally defined domain concepts to support automatic processing of data at the receiving system.
10. Audit trail or audit log: recording activities of information system users in chronological order, which enables prior states of the information to be faithfully reconstructed. It should contain information about access to and modifications of data as well as the nature of each access and/or modification.
11. BTG access: A restricted group of health professionals has the option to access medical documents without the permissions of the patient, in cases of extreme emergency. These cases may require need different security levels than the normal.
12. Patient's access: allowing the patient access to all his EHR information subject to jurisdictional constraints.
13. Data Sharing: ensure that only health professionals that have users grant can see their medical documents.
14. Data at rest: data at rest in IT means inactive data that is stored physically in any digital form (e.g. databases, data warehouses, archives, off-site backups, mobile devices etc.). Database can be compromised and viewed by an attacker. So data at rest must be protected with strong encryption.
15. Data at transit: when data is transmitted, the communication channel must be protected. Besides the protection of the channel, data can not be transmitted in clear text, must be always encrypted.
16. System Unpredictability : security depends on generated values being unpredictable. For instance if cryptographic keys are easy predictable, data can be tampered and patient's health compromised.

- Attack Model

The attack modeling is the process of identification and characterization of the attacks which a system is vulnerable. Next list contains common attacks and vulnerabilities that the system needs to prevent:

- Database intruders.
- Password theft.
- Replay attacks.
- Identity theft and broke authentication.
- SQL injections.
- Brute force.

- Sniffing data in transit (man in the middle passive).
- Data tampering (man in the middle active).
- Lookup and rainbow tables and dictionary attacks.
- System predictability.

The essential three steps in attack models are:

1. Threat identification and categorization.
2. Threat modelling and countermeasures.
3. Threat quantification based on risk.

Spoofing, Tampering, Repudiation, Information disclosure, Denial of Service, Elevation of privilege (STRIDE) is a threat categorization model developed by Microsoft for planning about computer security threats [168, 169]. It provides a mnemonic for security threats in six categories.

- **S**poofing identity : for vulnerable actions aimed to illegally access and use another users credentials, and fake identity (authentication).
- **T**ampering with data : involves the malicious modification of data. Examples include unauthorized changes made to persistent data, such as that held in a database, and the alteration of data as it flows between two computers over an open network, such as the Internet (integrity).
- **R**epudiation : threats are associated with users who deny performing an action without other parties having any way to prove otherwise (non-repudiation).
- **I**nformation disclosure : threats that involve the exposure of information to individuals who are not supposed to have access to it. For example, the ability of users to read a file that they were not granted access to, or the ability of an intruder to read data in transit between two computers (confidentiality).
- Denial of Service (DoS) : attacks deny service to valid users, for example, by making a Web server temporarily unavailable or unusable. This have impact in system availability and reliability.
- **E**levation of privilege : an unprivileged user gains privileged access and thereby has sufficient access to compromise or destroy the entire system. Elevation of privilege threats include those situations in which an attacker has effectively penetrated all system defenses and become part of the trusted system itself, a dangerous situation indeed (authorization).

In our system the primary identified threats are in the next list categorization is based on STRIDE :

- System predictability (**S + I + E**).
- Password theft (**S + I**).
- Broken authentication or identity theft (**S + R + I**).
- Broken authorization (**S + I + E**).
- Data on transit exposure or tampering (**T + I**).
- Data storage exposure (**S + T + I + E**).

For threat identification and modeling, one of the most used models in the area of software audit and attack modeling are the threat tree. In next figures, threats are further analyzed

by exploring the attack paths, the root causes (e.g. vulnerabilities, depicted as orange blocks) for the threat to be exploited, and the necessary mitigation controls (e.g. countermeasures, depicted as green blocks) [168].

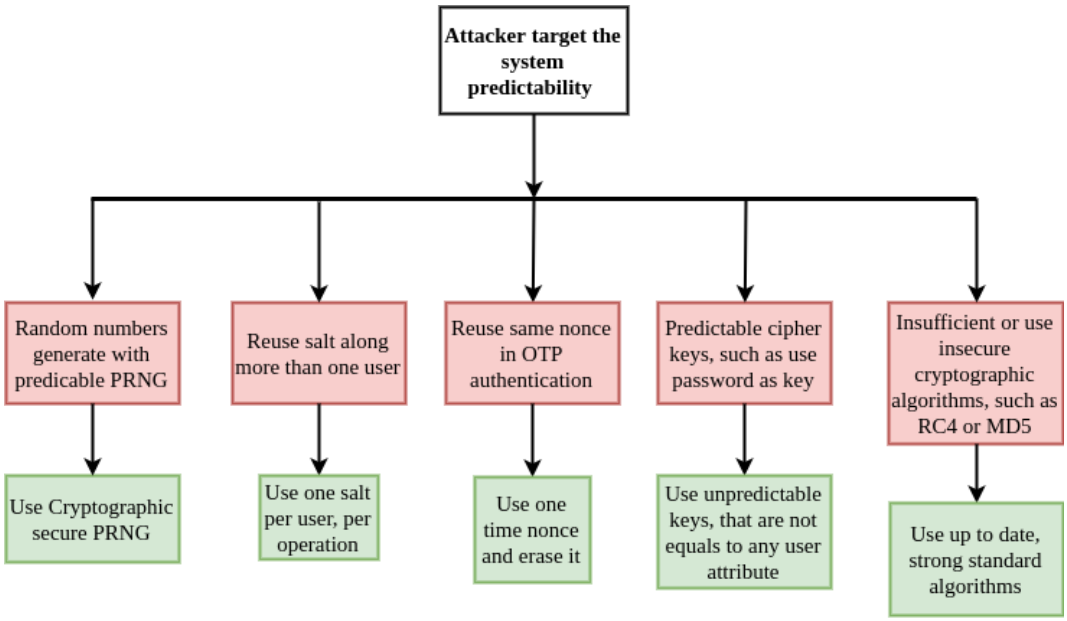

Figure 5.1: System predictability threat tree.

Figure 5.1 shows the threat tree for when an attacker targets the predictability of the system.

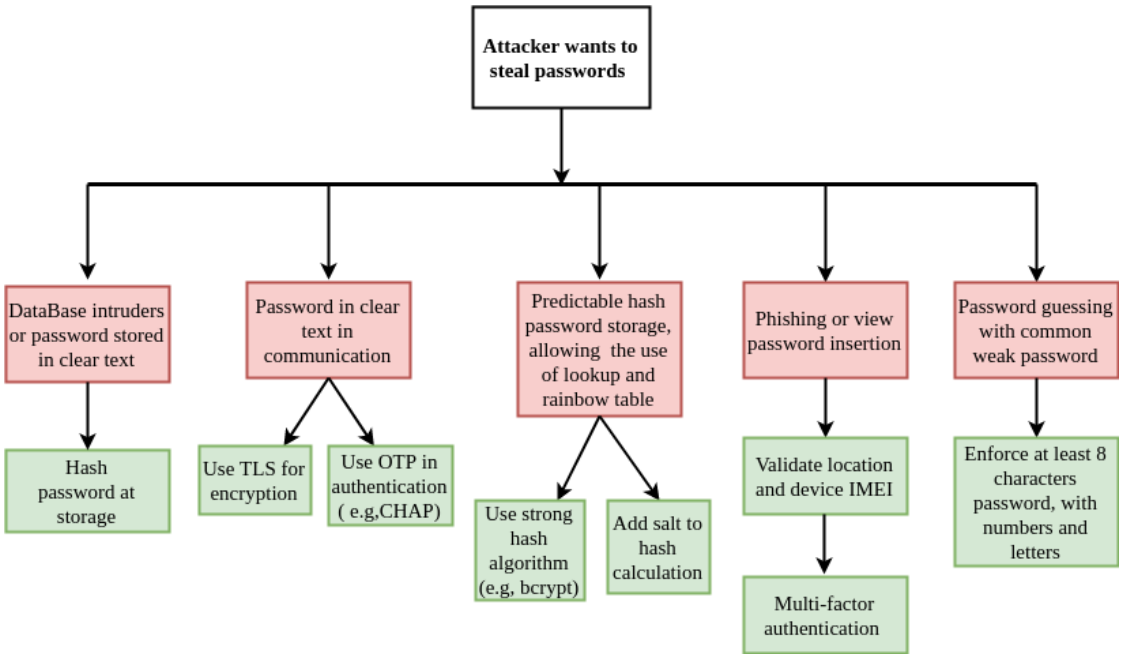

Figure 5.2: Password theft threat tree.

Figure 5.2 shows the threat tree for when an attacker wants to steak an user's password.

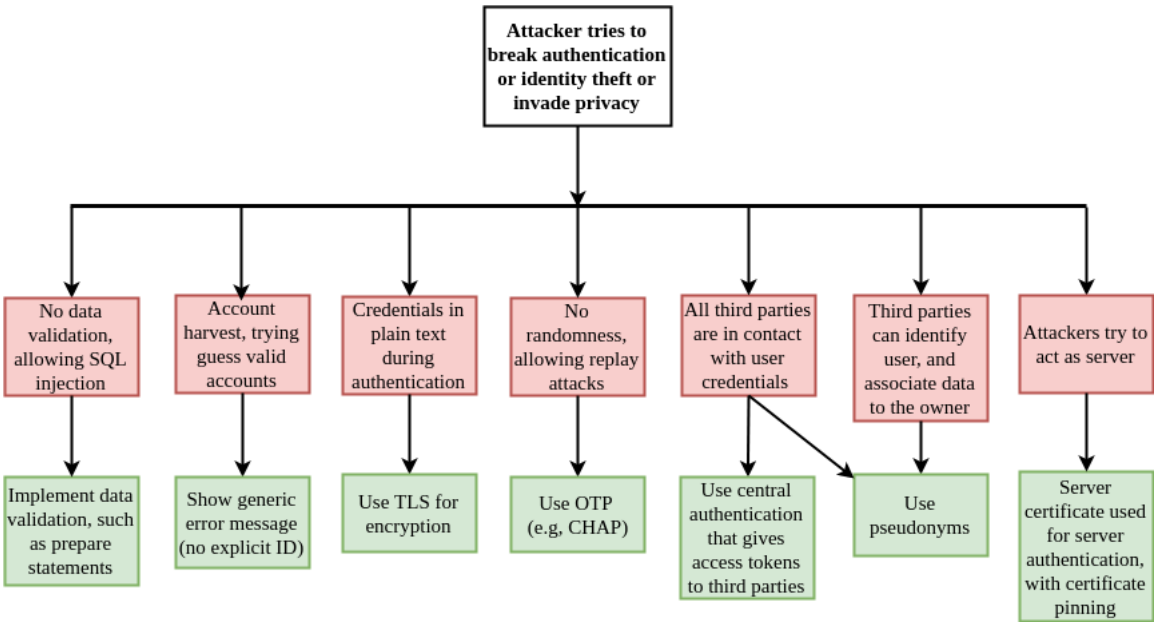

Figure 5.3: Broken authentication and identity theft threat tree.

Figure 5.3 shows the threat tree for when an attacker wants to break authentication or perform theft user identity.

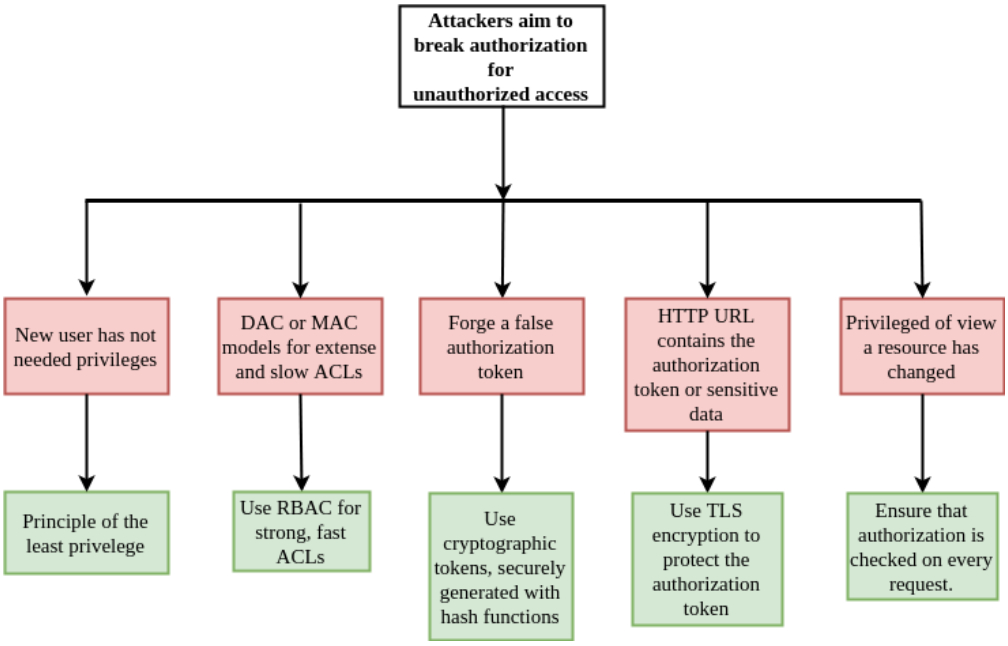

Figure 5.4: Authorization threat tree.

Figure 5.4 shows the threat tree for an attacker when he tries to gain unauthorized access by breaking authorization.

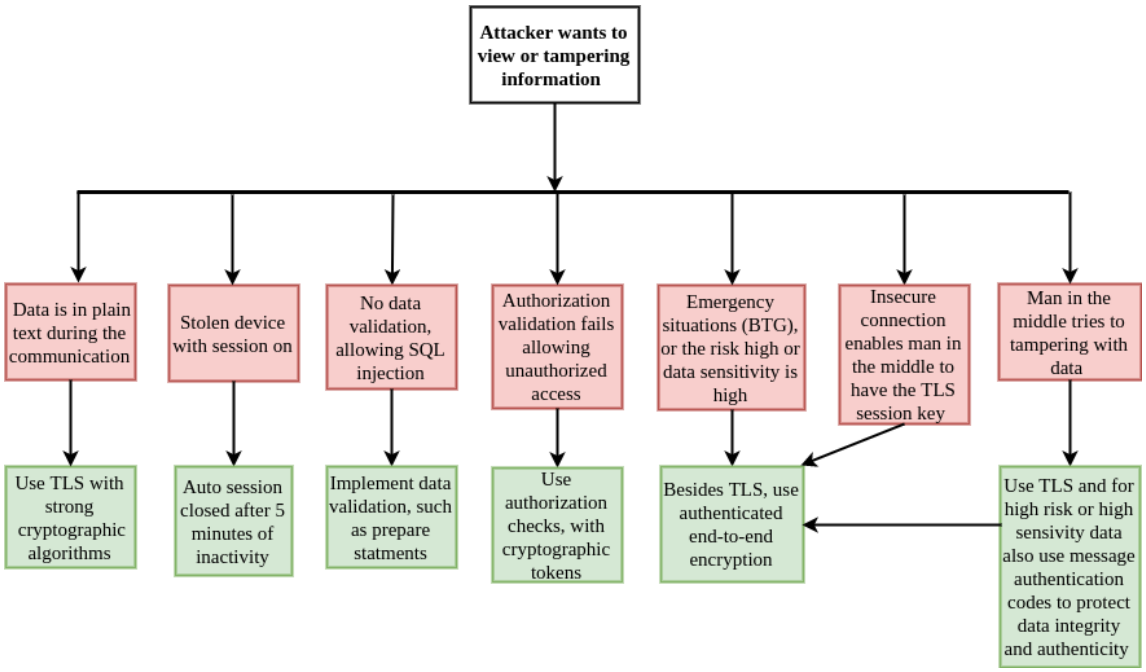

Figure 5.5: Information disclosure or tampering with data during communication threat tree.

Figure 5.5 shows the threat tree for information disclosure or tampering with data.

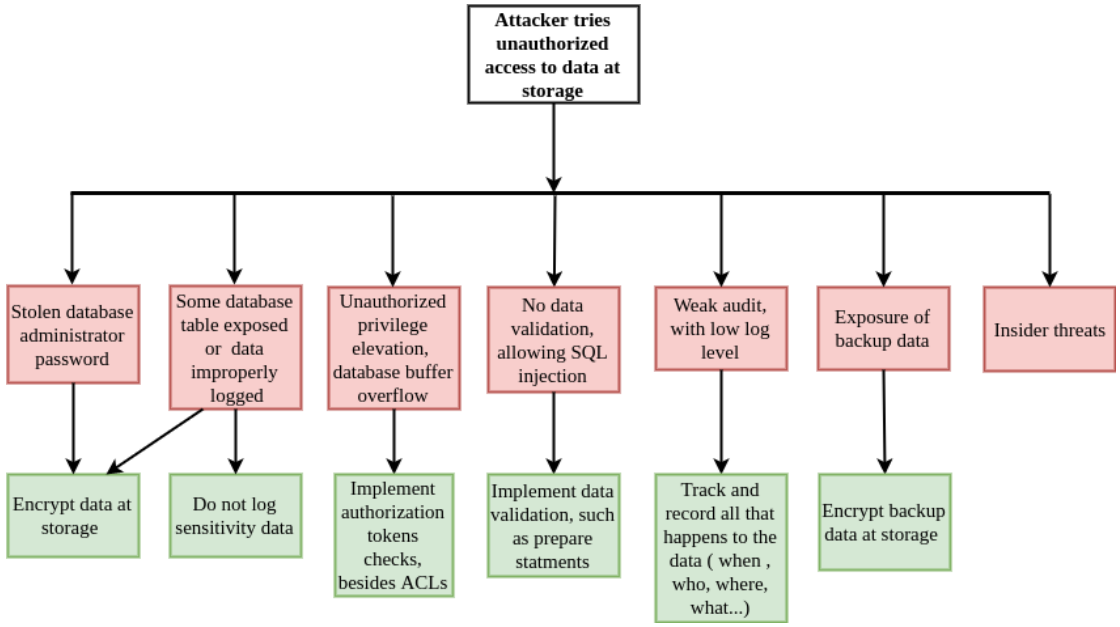

Figure 5.6: Data at storage disclosure threat tree.

Figure 5.6 shows the threat tree for when an attacker targets the data at storage in the database. The Microsoft Damage, Reproducibility, Exploitability, Affected Users, Discoverability (DREAD) [168, 170] is a threat-risk ranking model. As the acronym implies, the technical risk factors for impact are Damage and Affected Users, while the ease of exploitation factors are Reproducibility, Exploitability and Discoverability. This risk factorization allows the assignment of values to the different influencing factors of a threat. DREAD is a mnemonic for several factors, which must classified each vulnerability from 1 to 10:

- **D**amage: How big would the damage be if the attack succeeded? Value 10 represents high

damage in the system.

- **R**eproducibility: How easy is it to reproduce the attack? Value 10 represents that the attack is easy to reproduce.
- **E**xploitability: How much time, effort, and expertise is needed to exploit the threat? Value 10 means that is easy to exploit.
- **A**ffected Users: If a threat were exploited, what percentage of users would be affected? Value 10 represents a lot of users are affected.
- **D**iscoverability: How easy is it for an attacker to discover this threat? By default,(and according to some indications [168]) this value is always 10, due to the impressibility.

To obtain the final risk ranking output, is computed the arithmetic mean: (**D** + **R** + **E** + **A** + **D**)/5. The high risk threats are the ones that need more resources and dedication to prevent. Based on the DREAD recommendations made by Microsoft [170] and OWASP [168], the threat trees in section 5.2 have the next DREAD classification:

1. System predictability (in figure 5.1): *D= 8*, predict key, salt and nonce's can lead to big impact in system, such as data exposure or identity theft. *R= 3*, the attack is hard to reproduce, it is need analyze network traffic, or get source code access and analyze it. *E= 2*, the attack needs to have a lot of knowledge and time to exploit the vulnerability. *A= 2*, only one user is affected, because each user has its own keys, salts and secret parameters. *D= 10*. **Risk**= (8 + 3 + 2 + 2 + 10) / 5 = 5, medium risk.
2. Password theft (in figure 5.2): *D = 9* with the password, an attacker can forge the authentication and theft identification, leading to full control of user data. *R= 4*, can be hard to reproduce, because the attacker besides the password needs the user ID and device IMEI for authentication. *E= 8*, there are different ways to exploit this vulnerability, some of them are easy and without great knowledge, that can be simply to guess a weak password or phishing. *A= 5*, essentially each user per password is affected, however in cases that an user has high privileges, can influence a group. *D= 10*. **Risk** = 7.2, medium-high risk.
3. Broken authentication or identity theft (in figure 5.3): *D= 9* with authentication credentials an attacker has access to full control of user data. *R= 8*, can be easy to reproduce with a simple SQL injection. *E = 5*, an attacker needs some technical knowledge to exploit this threat. *A= 8*, essentially each user per account is affected, however in cases that an user has high privileges, can influence a group. Or when the attack acts has a server, a group of users can be vulnerable to. *D=10*. **Risk**= 8, high risk.
4. Broken authorization (in figure 5.4): *D= 9*, this threat can lead to the disclosure of confidential data. *R= 3*, besides the authorization token, is enforced strong ACLs to double check permissions. *E= 3*, only with advanced knowledge the threat can be exploited. *A= 3*, all EHRs of one user that are targeted of unauthorized access, implies damage on the user and his family. *D= 10*. **Risk**= 5.6, medium risk.
5. Information disclosure and tampering with data during communication (in figure 5.5). *D= 10*, this threat can lead to the disclosure of confidential data and tampering with data which can have huge impact in patient's health. *R= 9*, can be reproduced in many ways, for instance by capturing plain text network traffic or SQL injection. *E= 5*, requires connection in the same network to capture network traffic or broke the authentication phase. Only a

skilled attacker can exploit this threat. *A= 6*, each owner of the EHR and his family. If the attacker can connect to one health institution network, the impact grows. *D= 10*. **Risk**= 8, high risk.

6. Data storage disclosure (in figure 5.6: *D= 10*, access to database can lead to the disclosure of confidential data, tampering with data, privilege elevation and identity theft. *R= 7* can be reproduce in some cases easily, if the proper security constraints are not implied. *E = 7*, an attacker with some knowledge or an insider worker can exploit the threat with low effort. *A = 10*, with database access an attacker gains access to all users' data.*D= 10*. **Risk**= 8.8, high risk.

5.3 Pseudo Random Number Generators

An important challenge in security is the generation of cryptographically strong random number. Pseudo Random Number Generators (PRNG) are used in a variety of cryptographic and security applications for generate secure random numbers. The output of such algorithms depends on an initialization value, known as seed. These applications give rise to two distinct and not necessarily compatible requirements for a sequence of random numbers: randomness and unpredictability [171].

1. Randomness: commonly, the concern in the generation of a sequence of allegedly random numbers has been that the sequence of numbers be random in some well defined statistical sense. The following two criteria are used to validate that a sequence of numbers is random:
    - Uniform distribution: the distribution of bits in the sequence should be uniform. That is, the frequency of occurrence of ones and zeros should be approximately equal;
    - Independence: no one subsequent in the sequence can be inferred from the others;
2. Unpredictability: the requirement is not just that the sequence of numbers is statistically random but that the successive members of the sequence are unpredictable. With 'true' random sequences, each number is statistically independent of other numbers in the sequence and therefore unpredictable;
    - Forward unpredictability: if the seed is unknown, the next output bit in the sequence should be unpredictable in spite of any knowledge of previous bits in the sequence;
    - Backward unpredictability: it should also not be feasible to determine the seed from knowledge of any generated values. No correlation between a seed and any value generated from that seed should be evident.

In terms of randomness, the requirement for a PRNG is that the generated bit stream appears random even though it is deterministic.
PRNGs are crucial pieces inside the context of computational simulation, because allow to create random situations in which an event of the real world is affected. Thus, one fail in terms of randomness quality can result in a deficient reproduction of the simulation model, possibly leading to false conclusions or security failures.
At the field of information security and cryptography, the PRNGs are often the source of passwords, cryptographic keys, salts and nonces. In cryptography is used a specific type of PRNGs, the Cryptographically Secure Pseudo-Random Number Generator (CSPRNG)s, which has some

proprieties that are more fit to this context. CSPRNGs are very different than ordinary pseudo-random number generators. As the name suggests, CSPRNGs are designed to be cryptographically secure, meaning they provide a high level of randomness and are completely unpredictable. We don't want to random generated strings to be predictable, so we must use a CSPRNG. All nonces, salts, Initialization Vector (IV)s, and keys of the implemented framework are derived from a next Java CSPRNG. All random keys, nonce or salt, must have a secure size (16bytes) and unique per user. To store a integer with this size, we need to use a *BigInteger*. Next code sample shows how to generate a secure random number.

```
import java.security.SecureRandom; //provides a strong CSPRNG library.
//SHA1 hash function to generate a stream of random numbers.
SecureRandom prng = SecureRandom.getInstance("SHA1PRNG");
//BigInteger with a RANDOM stream of 128 bits = 16bytes
BigInteger nounce = new BigInteger(prng.generateSeed(16));
//next line makes sure that the generated number is positive!
nounce = nounce.abs();
```

5.4 One Way Hash Functions and Password Storage

The most important aspect of a user account system is how user passwords are protected [52, 172]. User account databases are hacked frequently, so you absolutely must do something to protect your users passwords if the website is ever breached. If passwords are stored in plain text and the database is compromised an attacker can read all passwords. Even worse, some users use the same password to a set of online services, if one of them is compromised, an attackers can gain access to others services. The best way to protect passwords is to employ password hashing. Common vulnerabilities allow the theft of protected passwords through attack vectors such as SQL Injection, artifacts such as logs, dumps, and backups.

Hash algorithms are one way functions. They turn any amount of data into a fixed-length output that cannot be reversed. They also have the property that if the input changes at least a digit, the resulting hash is completely different. This is great for protecting passwords, because we want to store passwords in a form that protects them even if the password file itself is compromised, but at the same time, we need to be able to verify that a users' password is correct. In a client server architecture, the hash function is performed on the server side.

However, if two users has the same password, they will have the same password hashes. We can prevent these attacks by randomizing each hash, so that when the same password is hashed twice, the hashes are not the same. So it can be added some salt. Salt is random bits of data that helps protect against dictionary and other precomputation attacks, such as rainbow tables. It is used to ensure that the same plain-text will not consistently hash to the same output value. A salt must be truly random, generated with CSPRNG. The salt does not need to be secret. Just by randomizing the hashes, lookup tables, dictionary attacks, and rainbow tables become ineffective. The salt needs to be unique per-user per-password. Every time a user creates an account or changes their password, the password should be hashed using a new random salt.

Salt ensures that attackers can't use specialized attacks like lookup tables and rainbow tables to crack large collections of hashes quickly, but it doesn't prevent them from running dictionary or brute-force attacks on each hash individually. Top computers and custom hardware can compute millions of hashes per minute, so these attacks are still very effective. To make these attacks

less effective, strong hash functions should be used. Secure Hash Algorithm (SHA) hashed secure passwords are faster and still secure, but able to be cracked with today fast computers. So the goal is to make the hash function slow enough to block attacks, but still fast enough to not cause a noticeable delay for the user. This feature is essentially implemented using some machine intensive algorithms such as argon2, bcrypt or scrypt. These algorithms take a work factor (also known as security factor) or iteration count as an argument. This value determines how slow the hash function will be.

The bcrypt hash function uses the parameters cost, salt, and key as input. The number of executed loop iterations is exponential in the cost parameter and makes heavy use of the Blowfish encryption function. The salt is used to block rainbow table attacks. Bcrypt is an over time adaptive function, the iteration count can be increased to make it slower, so it remains resistant to brute-force search attacks even with increasing computation power [173].

Following NIST[40] and OWASP[172, 174], bcrypt is the cryptographic hash function chosen to protect the passwords in the system. The user password is protected since the registration process with the bcrypt. The salt added to bcrypt hash computation is generated with bcrypt secure random library, and stored in the database. The following code sample shows how to generate a secure and random salt with bcrypt, and finally how to compute the final hash.

```
//log_rounds parameter determines the complexity
int log_rounds = 12;
SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
// the work factor is 2**log_rounds
String saltBcrypt = BCrypt.gensalt(12,sr);
//finally calculate the hash
String SecuredPwHash = BCrypt.hashpw(plainTextPassword, salt);
```

5.5 Secure Communication with Transport Layer Security

TLS [175] is a protocol that provides security for communications between client and server by implementing encrypted data and certificate-based authentication.

TLS is one of the most common ways of integrating secure communications on the internet, as it is a mature protocol that is well-supported by every major browser and a number of well-respected organizations provide third party TLS authentication services. It relies on Public Key Infrastructure (PKI).

The most common way that TLS is integrated into Internet communications is through the HTTPS protocol. Calling HTTPS a ”protocol” is not entirely accurate, as it is simply a combination of the HTTPS and TLS protocols. When we say a message was sent using HTTPS, what we are actually saying is that the message was first encrypted using TLS, transmitted and received using normal HTTP protocol, and then decrypted by the receiver, also with TLS.

The primary benefit of TLS is the protection of web application data from unauthorized disclosure and modification when it is transmitted between clients (web browsers) and the web application server, and between the web application server and back end and other non-browser based enterprise components. The server validation component of TLS provides authentication of the server to the client. If configured to require client side certificates, TLS can also play a role in client authentication to the server. However, in practice client side certificates are not often used. Username and password based authentication models are more often used for the

clients. In the proposed system, only the server will present its certificates, clients will be authenticated through username and password. TLS also provides two additional benefits that are commonly overlooked, the integrity guarantees and replay prevention. A TLS stream of communication contains built-in controls to prevent tampering with any portion of the encrypted data. In addition, controls are also built-in to prevent a captured stream of TLS data from being replayed at a later time [52, 176]. Besides, is the main defence against man-in-the-middle attacks.

Thus, TLS secures communication by providing message encryption (data at transit and confidentiality), integrity, and server authentication.

In order for public key encryption to provide secure communication, one more of the communicating parties must have some way of proving to the other that they are, in fact, who they claim to be. TLS provides this proof by requiring that one or more of the parties present a digital certificate into the initial negotiation of the connection, prior to the transmission of any encrypted data. This process is called handshaking.

To ensure that the certificate is a valid proof of identity, TLS contacts a trusted third party server specified in the certificate, called a Certificate Authority (CA). A CA is a trusted company that agrees to vouch for the identity of a site, usually for a fee. Generally, the more widely the CA is known as a reputable organization, the more they will charge you per year to verify any web site identity. Examples of well-respected CAs include Verisign and Digicert.

Figure 5.7 presents an architecture of how TLS handshake works.

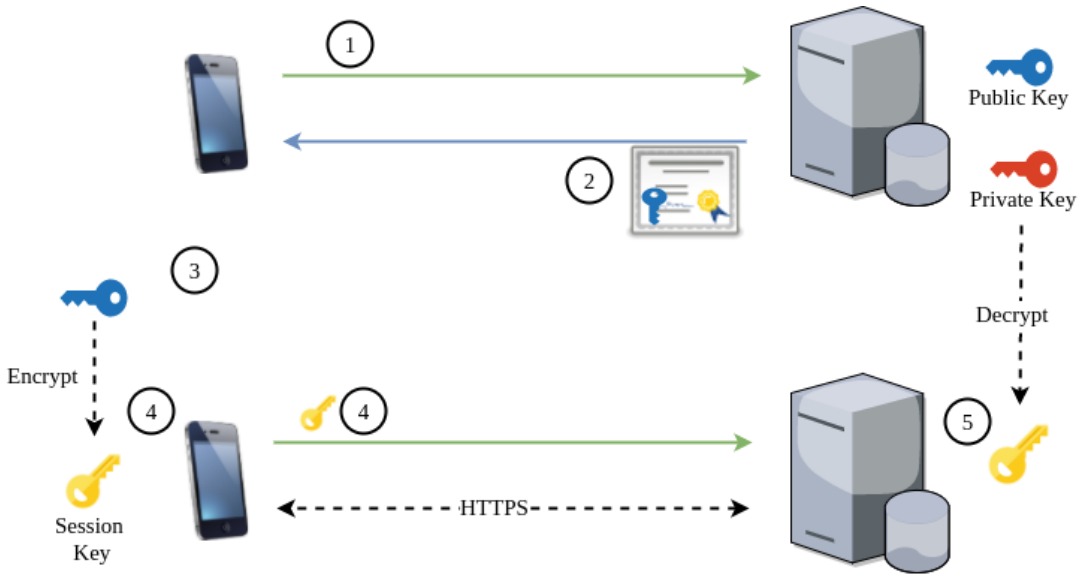

Figure 5.7: TLS handshake.

1. Client request a TLS connection.
2. Server responds with his signed certificate, which includes his public key.
3. Client verifies the server certificate.
4. Client generates a symmetric key, encrypt it with server public key, and transmits it to the server.
5. Server decrypt the symmetric key with his private key.

- TLS session established! Both server and client have a session symmetric key that only the two know, and it will be used it to encrypt all messages in the communication.

- TLS Configurations

As already explained in subsection 4.3.3, SPs and IdPs will run on top of REST architecture with Tomcat server. Manly for the first tests, a self signed certificate will be used, not signed for any CA. For this purpose, the main possibilities to create a self signed certificate are to use openSSL or Java keytool. For now, Java keytool, which is included in Java Development Kit (JDK), will do the purpose. The following terminal command sample shows how to do it.

```
root@root: usr/lib/jvm/java-8-oracle/bin# keytool -genkey -alias tomcatcert
-keyalg RSA -keystore ~/Desktop/keyStore
```

Now we have created a new certificate named `tomcatcert`, which locate in `~/Desktop/keyStore`, as shown the figure 5.8.

```
root@[redacted]:/usr/lib/jvm/java-8-oracle/bin# keytool -list -keystore ~/Desktop/keyStore
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

tomcatcert, 3/out/2017, PrivateKeyEntry,
Certificate fingerprint (SHA1): C4:A1:5B:7E:A0:B4:93:7F:D3:76:8B:B7:35:F8:77:B9:61:FA:90:CC
root@[redacted]:/usr/lib/jvm/java-8-oracle/bin#
```

Figure 5.8: List certificates in a keystore.

TLS has a lot of cipher suits available. When a client connects to a server to set up a secure connection, both parties will negotiate about which cipher suite to use. The strength of the encryption used within a TLS session is determined by the encryption cipher negotiated between the server and the browser. In order to ensure that only strong cryptographic ciphers are selected the server must be modified to disable the use of deprecated and weak ciphers and to configure the ciphers in an adequate order. It is recommended to configure the server to only support strong ciphers and to use sufficiently large key sizes. Rivest, Shamir, Adleman (RSA), Advanced Encryption Standard (AES) and Ecliptic Curve (EC) with Diffie-Hellman Ephemeral (DHE) are the most viable options. AES is not as strong as the others, but in Galois/Counter Mode (GCM) provides AEAD, that assures confidentiality, integrity and authenticity (more at 5.7). Following Mozilla Security TLS [177] recommendations and OWASP specifications on TLS [176], here is the final list of cipher suits:

```
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256, TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA,
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256, TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_128_CBC_SHA, TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_RSA_WITH_AES_256_GCM_SHA256, TLS_RSA_WITH_AES_256_CBC_SHA
```

Important notes:

- The Rivest Cipher (RC)4 ciphers should not be used, but in some cases they are needed for Internet Explorer on Windows XP compatibility reasons.

- The DHE ciphers are good, since they provide forward secrecy. But the EC with DHE ciphers are a stronger and better alternative to the DHE ciphers, and use a 571 bits elliptic curve key, which provides more than enough security.

At last, add at Tomcat `server.xml` file a new Connector element to enable TLS:

```
<Connector SSLEnabled="true" address="0.0.0.0"
ciphers="TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA,TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_128_CBC_SHA256,TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA256,TLS_RSA_WITH_AES_256_GCM_SHA256,
TLS_RSA_WITH_AES_256_CBC_SHA"
clientAuth="false" keyAlias="tomcatcert" keystoreFile="~/Desktop/keyStore"
keystorePass="*******" maxThreads="150" port="8443" protocol="HTTP/1.1"
scheme="https" secure="true" sslEnabledProtocols="TLSv1,TLSv1.1,TLSv1.2"
sslProtocol="TLS"/>
```

This is the configuration for the IdPs and th SPs.
Finally, in the client side (mobile application) the certificate needs to be validated. The application needs to have the server certificate, in order to validate the received certificate from the server when creating the TLS channel. So first the certificate is exported from the keystore:

```
root@root:~# keytool -export -alias tomcatcatcert -keystore ~/Desktop/keyStore
-file /home/workspace/IdPcert.cer
```

Next step is to use BouncyCastle to create a new keystore in the format `.bks`, to store the server certificate:

```
root@root:~# keytool -importcert -v -trustcacerts -file
/home/workspaceEclipse/IdPcert.cer -alias tomcatcert
-keystore /home/workspace/bouncyKeyStore.bks -provider
org.bouncycastle.jce.provider.BouncyCastleProvider
-providerpath /home/workspace/bcprov-jdk15on-158.jar -storetype BKS
-storepass ******
```

At last, the keystore `bouncyKeyStore.bks` is copied to the directory `res/raw/` in the mobile application. Now the mobile application can use this keystore to validate the received certificate from the server.
The server certificate is verified in each activity in the mobile applications. If for some reason this verification fails, the HTTPS requests will be blocked.

5.6 Secure Authentication

The authentication phase, after a successful registration, is the first engagement between the user and the Web server (IdP). Besides means to assure that the user is really who he claims to be, this phase must provide sufficient techniques to provide security and reliability. In case of

misconfigurations or lack of security techniques all user interactions from this point on can be compromised and targeted with unauthorized and malicious attacks, such as data tampering. As explained in 4.3.2, the IMEI and location are also used in the authentication phase, any time these attributes change, the user needs to enter in a extra level of security in authentication. This level is a multifactor authentication, which represents a PIN that is sent to user email, and user needs that pin to perform login, besides a user/password combination. It is also important to prevent brute force attacks and replay attacks.

In subsection 2.3.1 CHAP is introduced. This protocol ensures that passwords are never sent in clear text in the communication (besides have a TLS) and provides protection against replay attack by using of an incrementally changing identifier and a nonce. Thus, enables authentication with OTP, with different hash values, also avoiding COA.

Nonces (know as number used only once) are random bits of data that are often used as input to cryptographic protocols and algorithms, including many message authentication codes and some encryption modes. This suits to prevent replay attacks and keep the mutability. Nonces and salts are similar and serve related purposes, but are not identical. Both are typically randomly generated, usually secret, and serve to prevent attacks that would otherwise be possible against the system. They differ mainly in the context in which they're used, and in the consequences of repeats - a duplicate salt is unimportant, but a duplicate nonce can have dire consequences. So a nonce should be generated with a CSPRNG and discarded after one use.

CHAP works in a simple way. Lets suppose the user *Client* wants to do an authentication request in the authentication *Server*. Initially the user and the authentication system need to pre-share, in safe way, a secret of authentication *Secret*. After that, assuming that *ID* is the incrementally changing identifier and H is a secure SHA512 hash function:

- *Server* : Generates a nonce with safe size (128 bits).
- *Server* sends to *Server* the ID and the nonce.
- *Client* : Calculates *ValueUser* = H(ID||nonce||Secret).
- *Client* sends to *Server* the *ValueUser*, containing the authentication hash value.
- *Server*: Calculates himself *ValueServer*= H(ID||nonce||Secret), and compares *AuthValue-Server* == *AuthValueUser*. If true, authenticates *Client* and increment ID. If false, does not authenticate *Client*.

In the system framework the secret is the user password. Note that the password in server side is protected and stored with bcrypt. So at the client side it is necessary to perform a bcrypt hash computation over the password inserted by the user, and set *secret= bcrypt (password)*. At the server side, it is just necessary to read the password of the database. Figure 5.9 presents an architecture of how CHAP authentication operates in our framework.

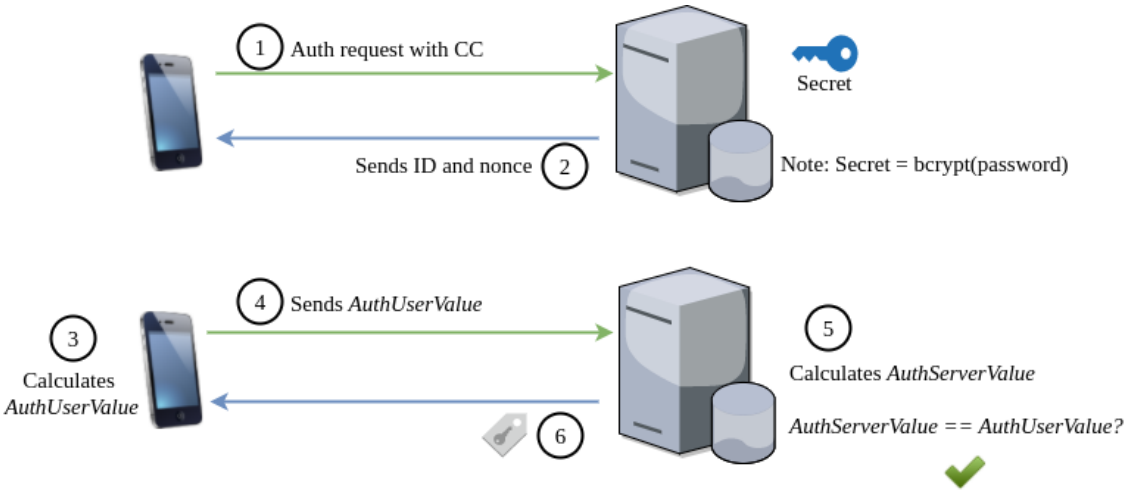

Figure 5.9: Secure authentication architecture.

1. User application requests authentication with the CC number. The password is not sent in this phase.

2. Server verifies the CC number. Reads the respective user salt, ID and generates a CSPRNG nonce, and sends both to user.

3. User application calculates Secret = bcrypt(password || salt).
   After it calculates *AuthUserValue*= H(ID||nonce||Secret)

4. Sends *AuthUserValue* to the server.

5. Server loads the bcrypt(password || salt) from database and sets it as the Secret. Then it calculates himself *AuthServerValue*= H(ID||nonce||Secret), and compares *AuthServerValue* == *AuthUserValue*.

6. If true, authenticates the user, increments the ID and sends to the user the authentication token. If false, it does not authenticate the user.

If authentication was successful is generated the token. Note that when the token is retrieved too the application is protected by the TLS connection. The hash algorithm that our framework uses in this step is SHA512, because login needs to be secure and quite fast, otherwise user experience is bad. SHA512 is fast and still secure nowadays. In step 1 are also sent attributes, such as OAuth **[client_id]** and **[client_secret]**, the IMEI and location of the user. The **[client_secret]** must be sufficiently random to not be guessable, which means it will use libraries with CSPRNG. In this case, because the **[client_secret]** is generated out of the application, openSSL is used, which contains a cryptographically secure library to generate a random 256-bit value and convert it to a hexadecimal representation.

```
root@root:~# openssl rand 32 -hex
d2df4e7e66efa5e448cd40735b7ff35c1aaf104d96fde6f354c2ea72e12086fb
```

The **[client_secret]** for myHEnCE is *d2df4e7e66efa5e448cd40735b7ff35c1aaf104d96fde6f354c2ea72e12086fb* and for myHEnCEPRO is *235a8c6dc2632810a148faa041f662e7baeec7e44a70e009cc5711c0bd75c7a3*. The **[client_id]** has the value *139138465_app_Nan0ST1MA_com*.

The **[client_id]** and **[client_secret]** are validated in every Android activity that makes a request, along with a generic validation if the username is not null. Also, the generated ATs are always validated when is suppose to.
Note that when the user fails one single time, when requested, the multifactor authentication, the token is revoked and is forced to restart the application. This hard strict police avoids brute force on the multifactor authentication. It is strict because the pin has a reduce length, 6 digits. To prevent brute force attacks on the normal authentication, an user only can only fail login 3 times. After that the application doesn't allow any more tries before the user restarts the application. As was previously said the system must record logs of all processes. To avoid DoS attacks, only the successful authentications are recorded in the database.

5.7 Authenticated Encryption with Associated Data

Protect the communication with TLS should be enough to secure data on transit. However, in cases where the data has a high level of confidentiality or the channel of communication has low protection, etc, end to end encryption is an interesting and good security option to adopt. The most used techniques to end to end encryption are Symmetric-key algorithms (e.g, AES) and Public-key cryptography (e.g, RSA). But then comes the question, is encryption enough to ensure confidentiality, integrity and authenticity of the data ? No. Encryption must always be combined with message integrity and authenticity protection [174, 97]. To do this, AEAD must be used with recommended modes (e.g,AES-GCM), which are specified in NIST approved modes and ISO/IEC 19772 [174].
AES-GCM [178] is a block cipher mode of operation that provides at the same time confidentiality, integrity and authenticity on the data. It supports high speed authenticated encryption and protection against bit-flipping attacks. It can be implemented in hardware to achieve high speeds with low cost and low latency. Software implementations can achieve excellent performance by using table-driven field operations. It contains an AES engine in Counter (CTR) mode and a Galois Hash module. It uses mechanisms that are supported by a well-understood theoretical foundation, and its security follows from a single reasonable assumption about the security of the block cipher [178, 97]. These modes require only one key. In general, the tag sizes and the IV sizes should be set to maximum values. Recent versions of openSSL and Crypto++ provide good implementations and also its available in the TLS cipher suites. Due to these previous reasons, the end to end encryption process in our system is based in AES-GCM.
GCM has two operations, authenticated encryption and authenticated decryption [178].
The authenticated encryption operation has four inputs, the key, the IV, the plain text and additional authenticated data. And has two outputs, a ciphertext and an authentication tag.
The authenticated decryption operation has five inputs, the key, the IV, the ciphertext, authenticated data, and the authentication tag. It has only a single output, either the plain text value or a special symbol that indicates that the inputs are not authentic.
The systems uses AES-GCM in 128 bits mode. To increase the unpredictability and randomness of the system, the 16 bytes key used in GCM encryption/decryption is the AT of user. At each authentication the token is renewed, so also the key changes at all sessions. For the additional authenticated data it will be used the user CC (unique for user). The 16 bytes IV is a secure random generated with a CSPRNG, unique for each encryption call.
More details about AES-GCM and how it computes are present in the original specification document [178].

5.8 Data Base Model and Protection

SQL injection can have a huge impact in a system. This kind of injection flaws are introduced when software developers create dynamic database queries that include user supplied input. A SQL injection attack is an insertion or injection of a SQL query via the input data from the client side to the application. When a successful injection exploit occurs, an attacker can read sensitive data from the database, modify database data with CRUD operations, execute administration operations on the database, recover the content of a given file on the file system and in some cases issue commands to the OS [179]. The typical ways to avoid SQL injection are stop writing dynamic queries and/or prevent user supplied input which contains malicious SQL from affecting the logic of the executed query. To do these, the primary defenses is the use of Prepared Statements. The android applications (myHEnCE and myHEnCEPRO) and the web services (IdP and SPs) are implemented in Java.
The following code example uses a PreparedStatement, Java implementation of a parameterized query, to execute the same database query.

```
 //inputs name and date must! be performed input validation to detect attacks
String name = request.getParameter("name");
String date = request.getParameter("date");
String query = "SELECT transaction_value FROM table_users WHERE username = ? AND
    transaction_date = ? ";

PreparedStatement pstmt = connection.prepareStatement( query );
pstmt.setString( 1, name);
pstmt.setString( 2, date);
ResultSet results = pstmt.executeQuery( );
```

It was assured that during the development of the Web services and both android applications, all query's to the databases use PreparedStatement (like the previous example), to protect SQL injection attacks.
Other important database privacy protection is the use of pseudonyms to achieve data anonymization. This measure is discussed and explained in subsections 4.3.1 and 4.3.2.
At last, it was designed a scheme to protect the data at rest in the databases. It relies in the use of a symmetric key per user, and a pair of asymmetric keys. Each user has it is own symmetric key, securely generated with a CSPRNG at the moment of the registration and since that moment protected with the asymmetric keys. That symmetric key is used to encrypt some data of user in the database. The symmetric key is stored in a isolated table, associated to the user identification in each Web server (CC in the IdP or pseudonym in the SPs). In the IdP almost all personal data at rest is encrypted. In the SPs only the EHR meta-data (pseudonym, data, institution, EHR_ID) is kept in clean text, because the key needs to be associated to a user, and if the pseudonym is encrypted there is no way to do that. The pair of asymmetric keys are used to protect (encrypt and decrypt) the symmetric key, and are stored in the file system of each Web service. The up to date cryptographic algorithms used are AES for symmetric encryption and RSA for asymmetric encryption. Let us suppose Bob is authenticated, and requests an EHR at rest. Considering the encrypt mode *E* and the decrypt mode *D* The steps that the Web service does are:

1. Loads the encrypted EHR and the encrypted symmetric key *eK* from the database.
2. Loads the asymmetric private key *sK* from the file system.

3. Obtains the plain text format of the symmetric key *K = D (sK,eK)*.
4. Finally computes *D (K,EHR)* and obtains the plain text EHR. The TLS channel will ensure the security of the communication and end to end encryption can also be used in some cases.

Note that the EHR and the symmetric key are never in clear text in the database! They are loaded to temporal variables in the Web service, and there are decrypted and able to use in other operations.
At last, a note to that when the account is created, the symmetric key is protected as follow:

- Load the asymmetric public key *pK* from the file system.
- Computes *eK =(pK,K)*, and stores the *eK* in the database.

Note that the previous process is only made once. Unless the symmetric key is compromised, then a new one needs to be generated. In this case, all data of that user needs to be decrypted with the compromised symmetric key, and then encrypted with a new symmetric key.

5.9 Conclusion

This chapter presents the security analysis of the framework, which is just a first step towards a full comprehension of risks and vulnerabilities. The security analysis using attack trees to model goals and threats is far from complete. These are just initial steps towards a global understanding of the system. Following that, the chapter presents the security controls and algorithms used in the framework implementation. These decisions were deeply researched and based in best recommendations from OWASP [50], NIST 800-53 [40] and NIST 800-57 [165]. Good and recognized recommendations are the first step to a reliable system/framework. To conclude this chapter, the cryptographic algorithms, protocols and methods used to secure the system architecture are:

- Hash computation: SHA512.
- Password Hashing: Bcrypt.
- Asymmetric encryption: RSA2048 bits.
- Data at end-to-end protection with symmetric-key algorithm (in cases of high risk): AES128 bits GCM mode.
- Data at rest protection with symmetric-key algorithm: AES128 bits Cypher Block Chaining (CBC) mode.
- Data in transit protection: TLS.
- Replay attacks protection: CHAP with SHA512.
- SQL injection: Prepared Statements.
- Generation of keys, salts, nonces: CSPRNG.
- Prevent brute force attacks: limited tentatives of failed authentication.
- Access control model: SoTRAACE
- Authorization: ACLs and OAuth.

Chapter 6

Implementation, Demonstration and Testing

6.1 Technologies Used

6.1.1 Android Framework

As showed in section 1.1 Android OS is widely adopted open-source project. Thus as a first implementation for myHEnCE and myHEnCEPRO the target was the Android OS. The native programming language used in Android development is Java. The Android OS uses Dalvik virtual machine that provides a platform-independent programming which allows the application to be executed the same way in any platform, independently of hardware and operating system. The Integrated Development Environment (IDE) used was Android Studio, the official IDE for Google's Android OS development. This provides the fastest tools (code editing, debugging, performance evaluation, compilation and instant running) for the creation of all types of Android Applications. The data records on the mobile device are stored under SQLite database. The Android layouts design are made using XML. The Android Studio IDE provides a virtual mobile device emulator (e.g, Nexus 5) to test and evaluate the final application.

To handle the asynchronous android client requests the android application uses LoopJ Android Asynchronous HTTP Client library [180]. It provides an asynchronous callback-based HTTP and HTTPS client for Android built on top of Apache HttpClient libraries, which is used by Pinterest, Instagram and others. Most Android devices allow to determine the current geo location. This can be done via a GPS module, via cell tower triangulation and via wifi networks. Google Play provides the fused location provider to retrieve the devices last known location. To use the location manager of Google play service and the LoopJ library there is the needed to add dependencies using Gradle buildscript in the application file *build.gradle*:

```
dependencies {
    compile 'com.loopj.android:android-async-http:1.4.9'
    compile 'com.google.android.gms:play-services:9.2.0'
    compile 'com.google.android.gms:play-services-location:9.2.0'
}
```

The *AndroidManifest.xml* contains essential information about the Android Application, needed for the system execute the code.This file contains information of the package, including components of the application such as activities, services, broadcast receivers, permissions, content providers, etc. It is responsible to protect the application when accessing any protected parts by providing the permissions and the Android API declares the application to use. Initially some permissions need to be added to this file, because we need to handle internet connections, access wifi informations (e.g, type of wifi connection and encryption), access internal device informations (e.g, IMEI) and location data (GPS).

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
//Allows applications to access information about wifi networks
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
//Allows read only access to phone state, including the information of the device,
    current cellular network information, etc
<uses-permission android:name="android.permission.INTERNET" />
//Allows applications to open network sockets
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
//Allows an app to access precise location
```

To run and test the Android application the API version used was the 24 and 25.

6.1.2 Web Service Specifications

RESTful API is a flexible way to provide different kinds of applications with data formatted in a standard way, which is very important in eHealth. It helps to meet integration requirements that are critical to building systems where data can be easily combined and extended. Also it is easier to use as it provides the use of JSON. IDE Eclipse Neon Enterprise Edition was used to build the Java Web service and configured to use Apache Tomcat servlet container (often referred to as Tomcat server). Also the Web service uses Jersey libraries and tools. Jersey [181] RESTful Web services framework is open source, production quality, framework for developing RESTful Web services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) reference implementation.

6.1.3 Data Base Specifications

The Relational Database Management System (RDBMS) MySQL, with the phpMyAdmin administration tools, were chosen ones to create and manage the database. To connect the Java based Web services (IdP and SPs) to the MySQL data base the official driver connector Java Database Connectivity (JDBC) was used. The data base connection Java configuration in the Web services follow the next block of code:

```
public class Constants {
    public static String dbClass = "com.mysql.jdbc.Driver";
    private static String dbName= "users";
    public static String dbUrl = "jdbc:mysql://localhost:3306/"+dbName;
    public static String dbUser = "root";
    public static String dbPwd = "********";
}
```

Every executable path on the Web services are logged to the console for later audit. Sensitive data, such as passwords, are not logged to console. These logs can also be later exported to a document. Besides, some complementary data is also stored in the database (timestamps, location, IMEI), all successful authentications, when was token issued and revoked, etc).

6.2 Demonstration and Validation

6.2.1 Registration and Authentication

For the initial tests, registration with QRcode was not implemented. This process is made with the patient inserting his credentials manually in myHEnCE application. In figure 6.1 is a screenshot demonstrating the registration activity in myHEnCE application. In this step there is an alarm that informs the user if the registration was successful or not. After a successful registration, the remaining parameters for the patient authentication are automatically generated (salts, pseudonym, bcrypt of password). At this moment the AT is null and not issued. The server certificate is verified in each activity. If for some reason this verification fails, the HTTPS requests will be blocked.

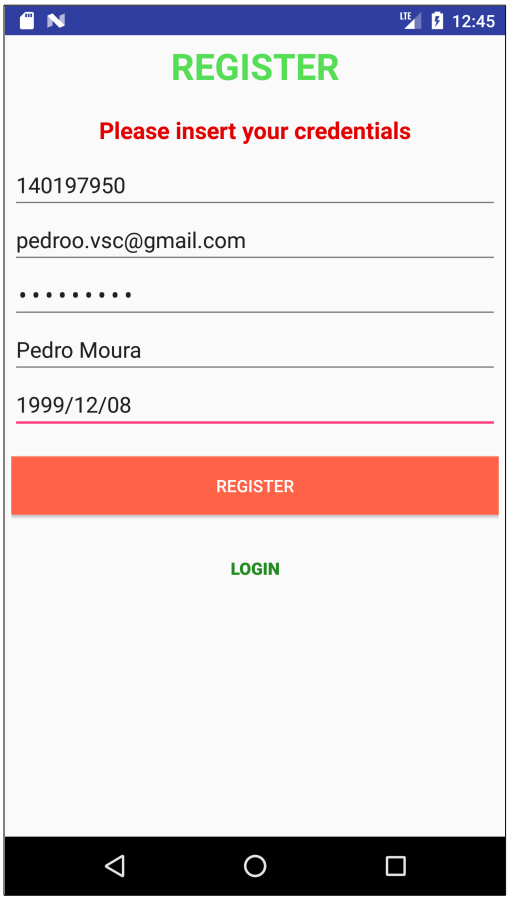

Figure 6.1: Patient's registration with myHEnCE.

After the registration the patient needs to perform authentication. Figure 6.2 is a screenshot demonstrating the authentication activity in myHEnCE application. After a successful authentication the AT is issued and recorded in the IdP.

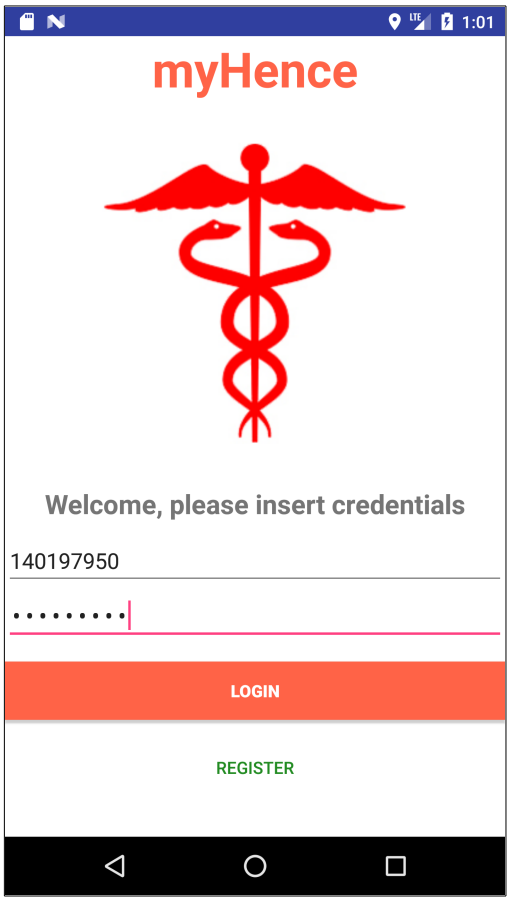

Figure 6.2: Patient's authentication with myHEnCE.

There are two more validations within patient's authentication. The number of failed authentication tentatives and if it is needed multifactor authentication (new device and new location verification). Figure 6.3 and figure 6.4 present screenshots demonstrating how the application deals with these two measures.

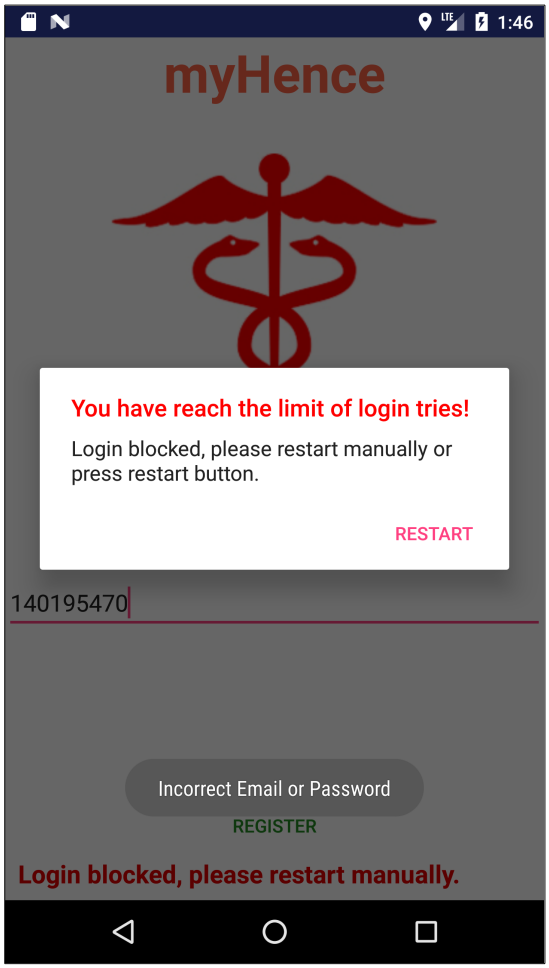

Figure 6.3: Authentication blocking brute force in myHEnCE.

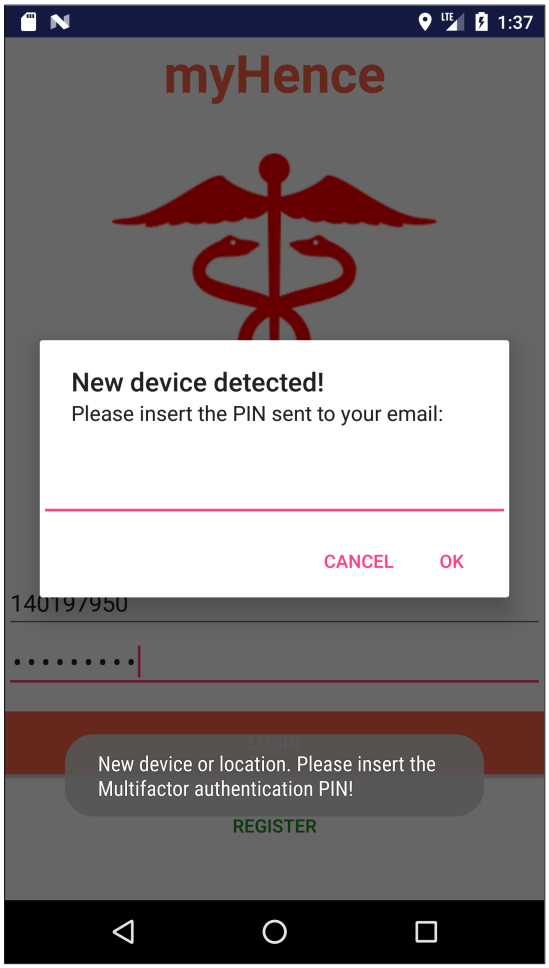

Figure 6.4: Multifactor authentication in myHEnCE.

An email is sent to the patient with the PIN for multifactor authentication and has the following format:

```
Title: PIN to 2FA
Sender : ubi.nanostima@gmail.com
Receiver : pedroo.vsc@gmail.com
Body: This is your PIN :22265
```

If the patient reaches a successful authentication, the IdP stores in the database a record of the authentication attributes: CC, IMEI, IP, date, and the model and version of the used device. Only successful authentications are recorded to avoid attempts of DoS attacks.

After evaluating of the legislation and debating with CINTESIS researchers, it was defined that the registration for health professionals using myHEnCEPRO cannot be performed in the application. This process must be handled by a high level institution, and cannot just be handled within the application.

For health professionals, the authentication process and application activity in myHEnCEPRO are very similar to the patient with myHEnCE. Note that to the health professionals is not issued the initial AT. As said previously, they have a set of tokens delegated by the patients. Multifactor authentication is always requested for health professionals. It is expected to be handled with a medical card and a card reader, but for the initial tests only the card number is needed. In the

same way it is protected against brute force attacks as is myHEnCE. Also the devices provided to the health professionals by the institutions are already registered in the institution system.

6.2.2 Main Menu

In figure 6.5 is a screenshot demonstrating the patient's main menu activity and the respective options in myHEnCE application. And in figure 6.6 is a screenshot demonstrating the health professionals main menu activity and the respective options in myHEnCEPRO application.

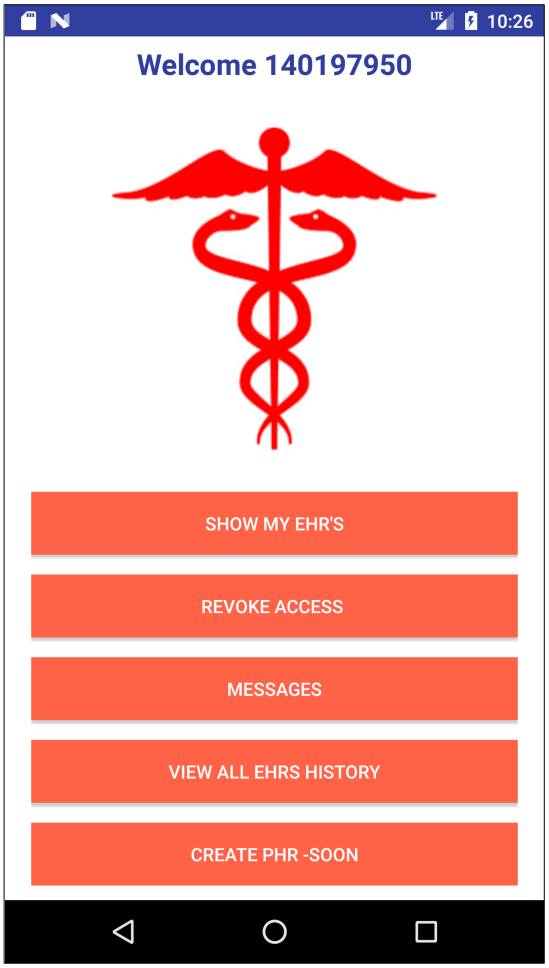

Figure 6.5: Main menu in myHEnCE.

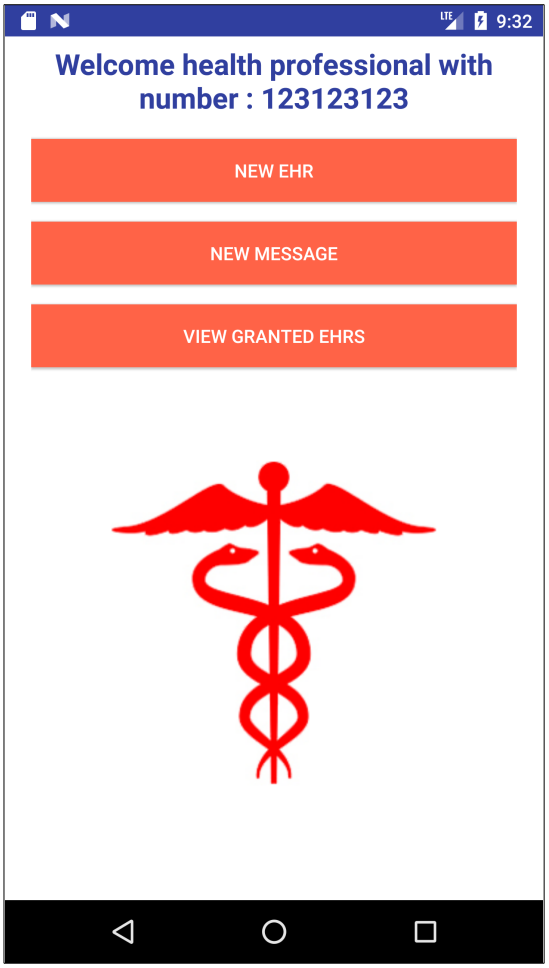

Figure 6.6: Main menu in myHEnCEPRO.

6.2.3 Message System

The message system allows unidirectional communication, providing means to the health professionals to send messages to the patients. In figure 6.5 is a screenshot demonstrating the activity where a health professional can create and send a message to one patient, using myHEnCEPRO application. Health professionals can reach this activity by pressing the button 'New message' in the main menu activity. In this step there is a verification if the patient exists and an audio?? alarm that informs the user if the creation was successful or not.

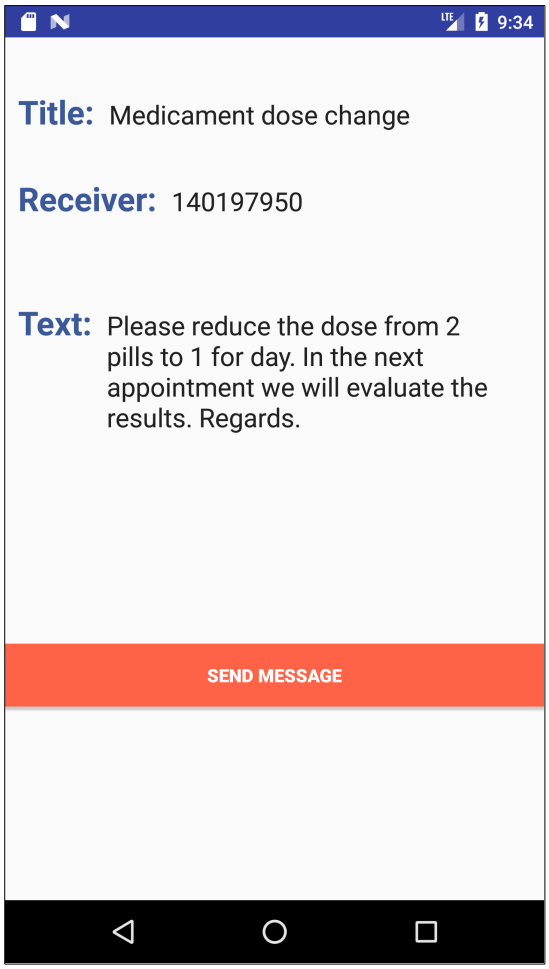

Figure 6.7: Create a new message in myHEnCEPRO.

In myHEnCE the patient can view the list of his received messages by pressing the button 'Messages' in the main menu activity (screenshot in figure 6.8). The patient can then select one message from the list, and view the content, as the screenshot in figure 6.9 shows.

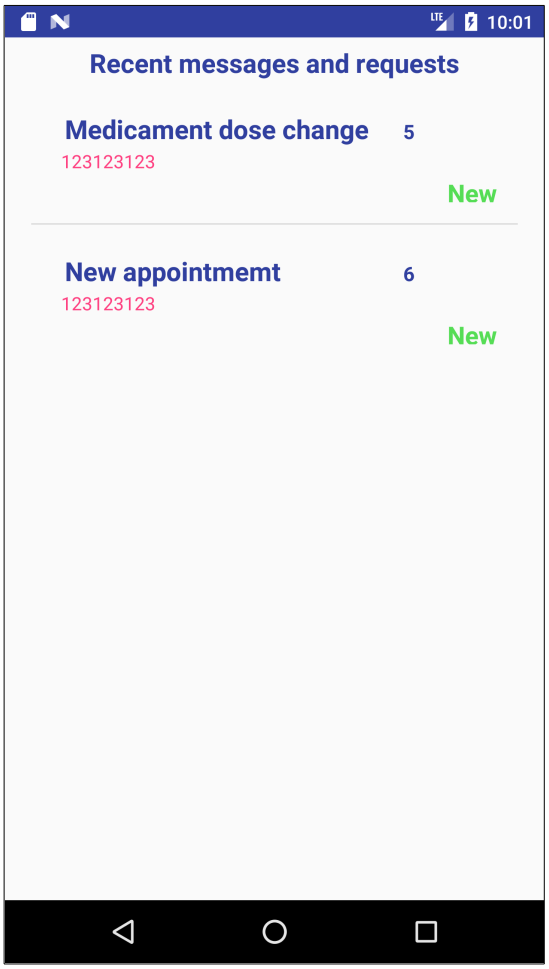

Figure 6.8: List of patient messages at myHEnCE.

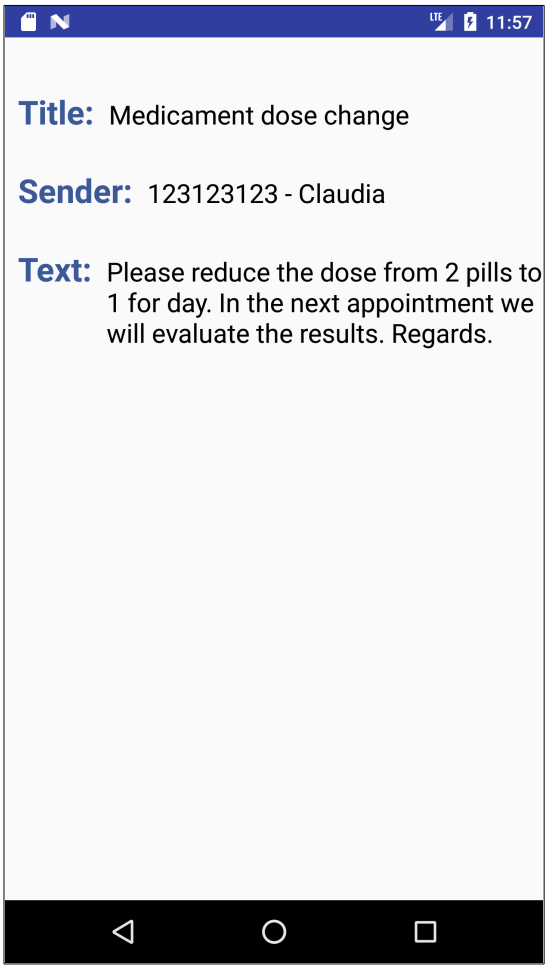

Figure 6.9: Visualization of one message at myHEnCE.

6.2.4 EHR Management System

In myHEnCEPRO, by pressing the button 'New EHR' on main menu activity, the health professional can create a new EHR. The health professional who creates a new EHR automatically gains permission to view it. Two screenshots of this activity are in figures 6.10 and 6.11. In this step there is a verification if the patient exists and an audio ??alarm informing the user if the creation was successful or not. As said previously, the composition of the EHR at this point is simple, later will be integrated with openEHR standards. The field 'Sensitivity' at this point is decided by the health professional, and later used in SoTRAACE risk evaluation.

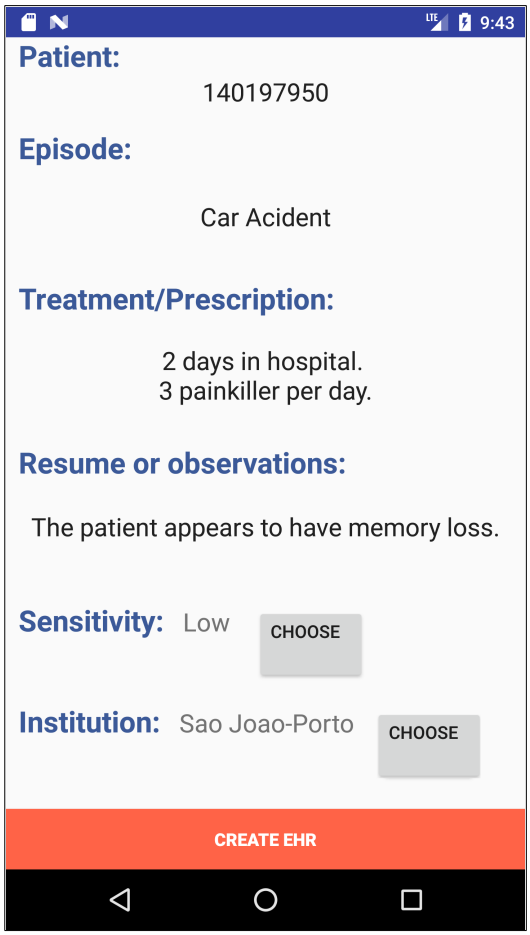

Figure 6.10: Create new EHR in Porto using myHEnCEPRO.

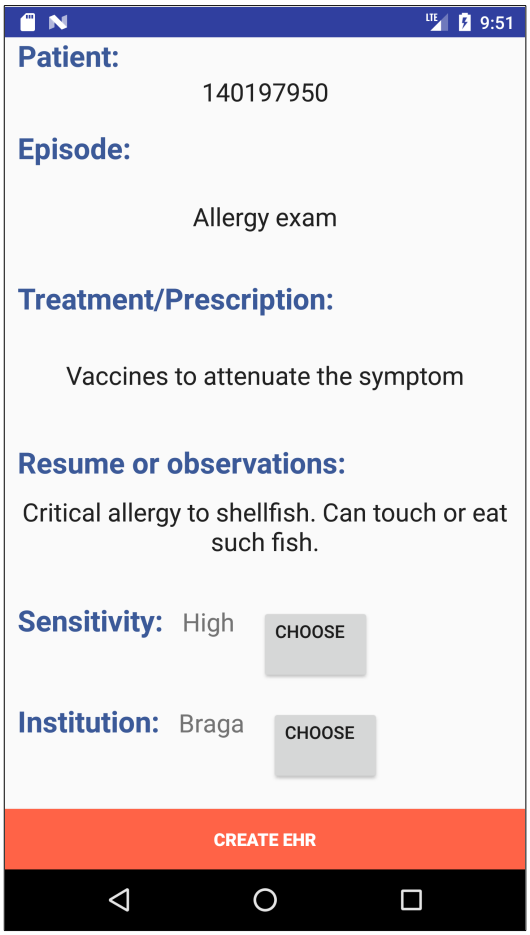

Figure 6.11: Create new EHR in Braga using myHEnCEPRO.

On the patient's side using myHEnCE, by pressing the button 'Show my EHRs' on the main menu activity, the application presents a list with the available EHRs of the patient (screenshot in figure 6.13). At this point the AT is validated. If the token is not valid, the list comes empty and the patient is warned about that. The query to present that list is filtered by institution (screenshot in figure 6.12). The system is highly modular and adaptable, it is easy to add a new query filter to be made by date, patient's CC, etc.

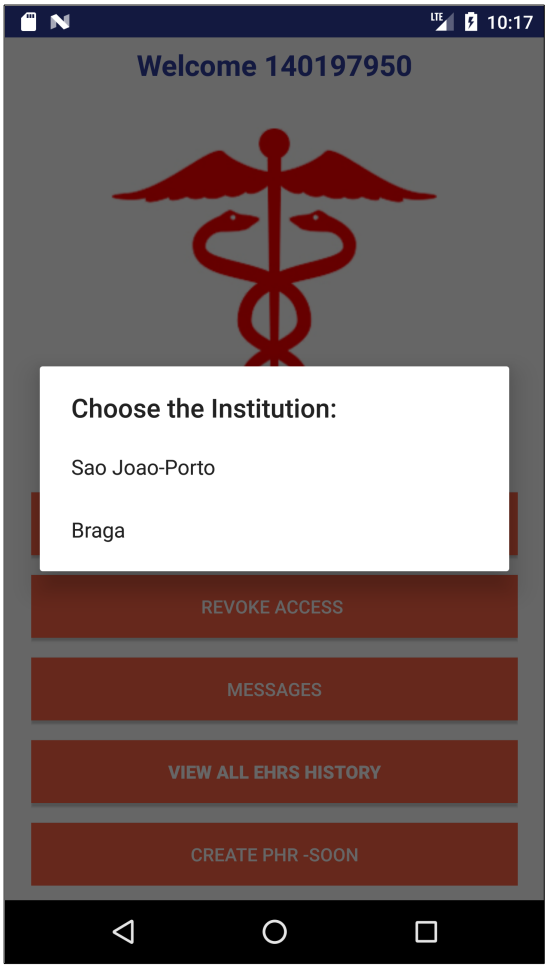

Figure 6.12: Search EHR by institution using myHEnCE

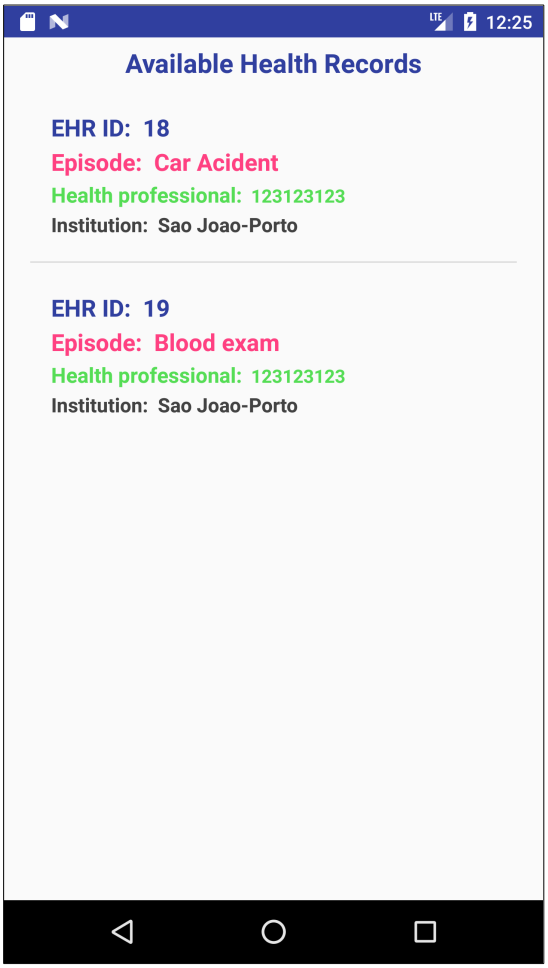

Figure 6.13: List of available EHR using myHEnCE.

After selecting one item from the list, the EHR is shown in a new activity (screenshots in figure 6.14 and 6.15). In this request the `final_risk<=1.6`, so no restrictions are applied, only informative advice about security is provided. Obviously this advice is provided after analyzing the attributes sent to the IdP. The AT is also validated in this step. If the AT is invalid, the user is warned and the EHR is not shown.

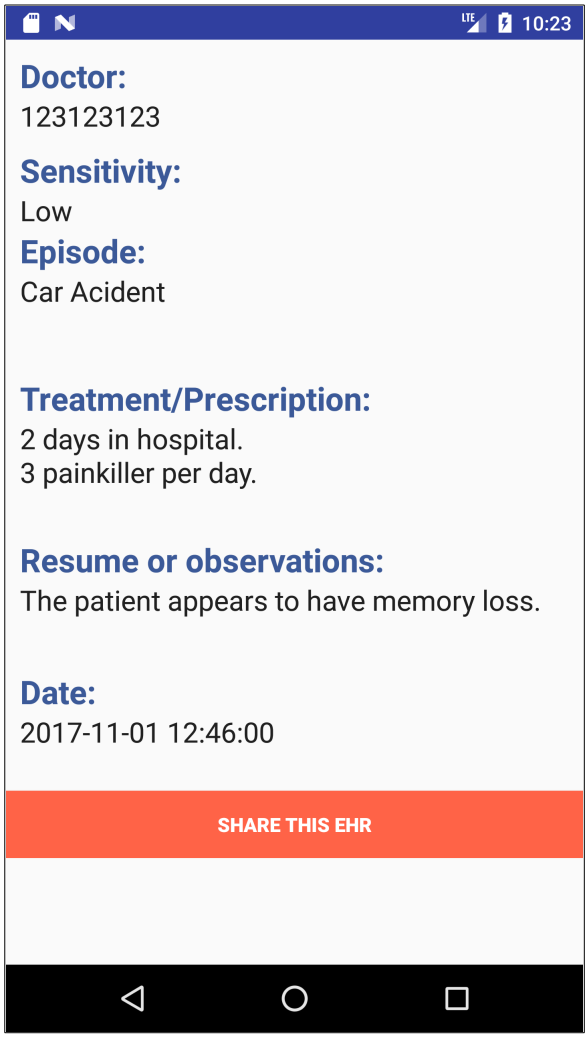

Figure 6.14: Visualization of one EHR using myHEnCE with low risk.

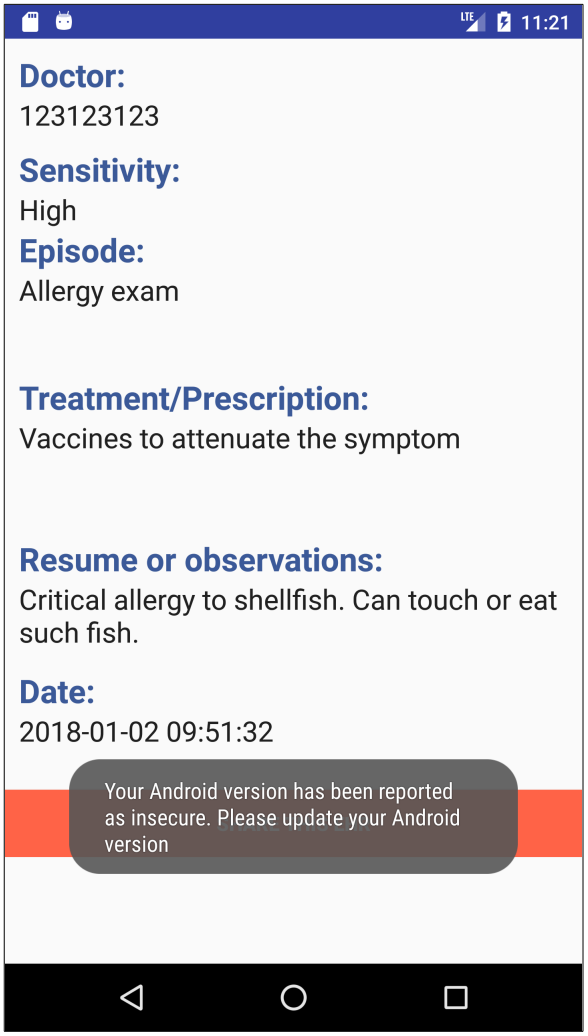

Figure 6.15: Visualization of one EHR using myHEnCE with low risk and advice.

Screenshot in figure 6.16 presents the result of a request with medium risk (`final_risk>1.6 && final_risk<=2.2`). In this case some items can be hidden and an end-to-end encryption is applied. The patient is advised of this procedure and what is more unsecure in that request. Figure 6.17 presents a screenshot with the result of a request with high risk (`final_risk>2.2`).

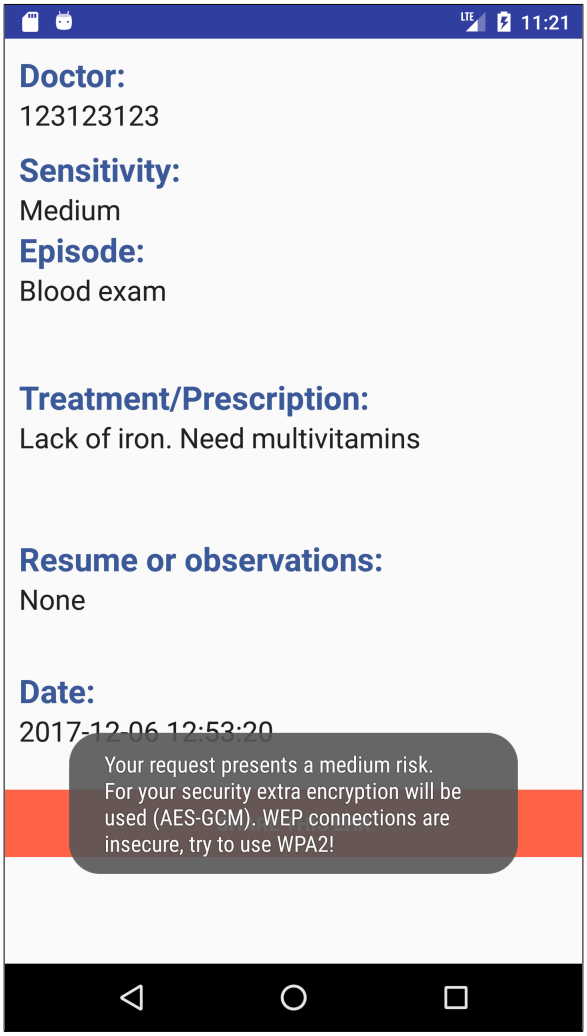

Figure 6.16: Visualization of one EHR using myHEnCE with medium risk.

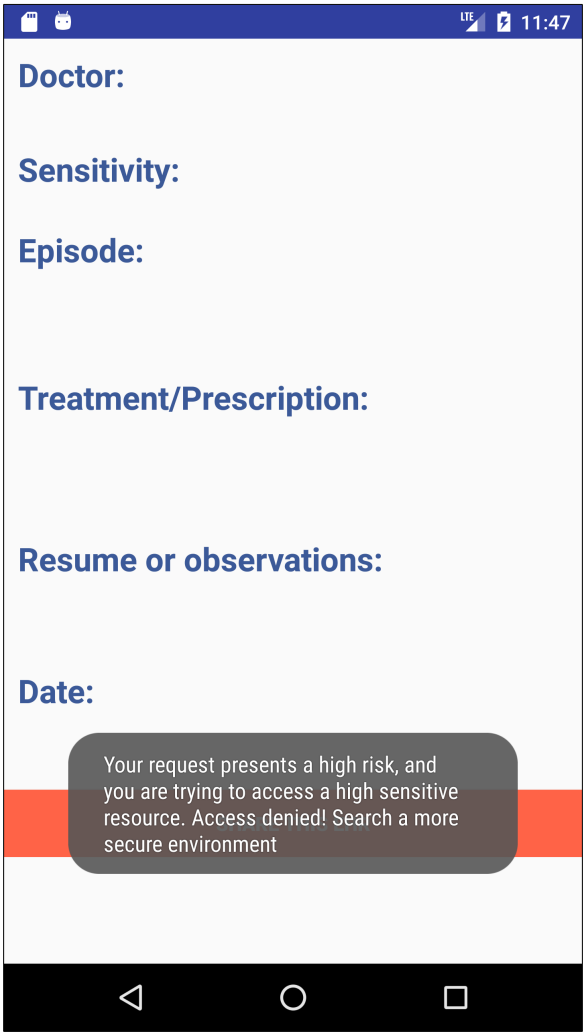

Figure 6.17: Visualization of one EHR using myHEnCE with high risk.

The patient can view the history list of who, when, and where accesses were made to his EHRs. This option is available in the main menu activity, by pressing the button 'View all EHRs history', which navigates to a new activity and shows that list (screenshot in figure 6.18).

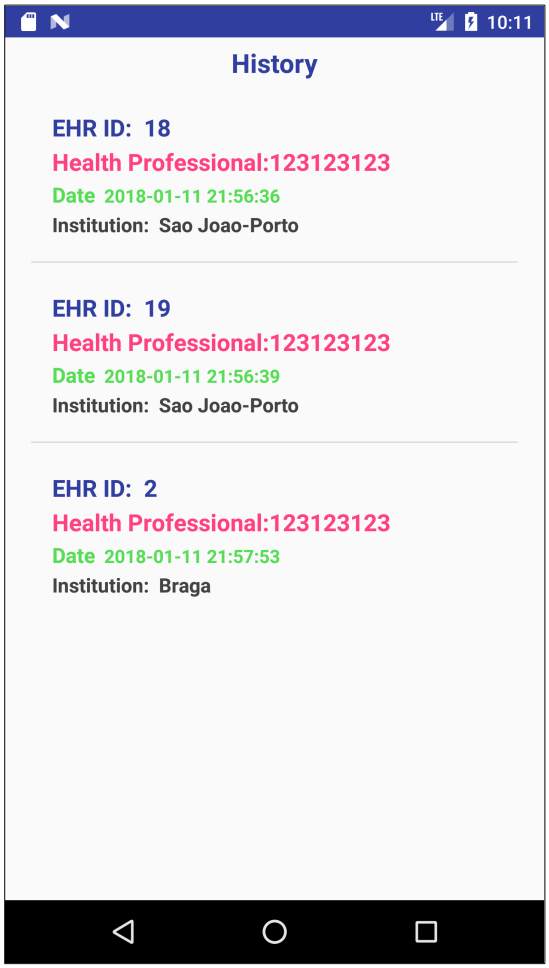

Figure 6.18: Visualization of view history of all patient's EHRs using myHEnCE.

6.2.5 EHR Authorization System

When the patient is viewing each EHR (screenshot in figure 6.14), the activity has a button 'Share this EHR' that allows the patient to share that specific EHR. By pressing this button the patient needs to insert the CC of the health professional that he wants to share the EHR with (screenshot in figure 6.19). The system generates an AT, and inserts it in the list of health professional ATs, and the permission in the ACL. To this use case, sharing is made to a different health professional from the one who created him (in this case share with health professional with CC 485368951). In this step there is a verification if the health professional exists, and an audio?? alarm informs the user if the sharing tentative was successful or not.

In myHEnCEPRO the health professional can view a list with the EHRs that he has or already has had access permissions in the past (screenshot in figure 6.20). This option is present in the main menu activity, by pressing the button 'View granted EHRs'. Consequently each EHR can be selected and presented.

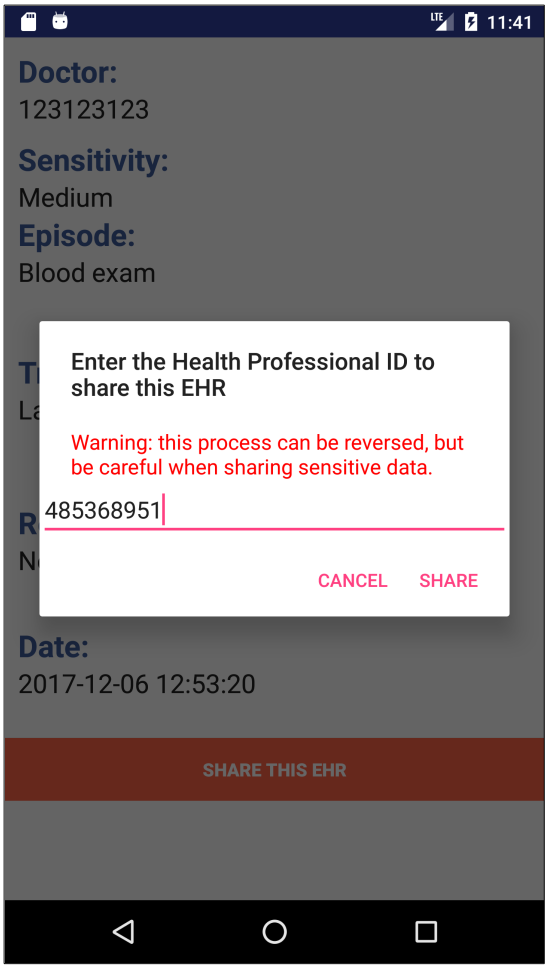

Figure 6.19: Share of a specific EHR using myHEnCE.

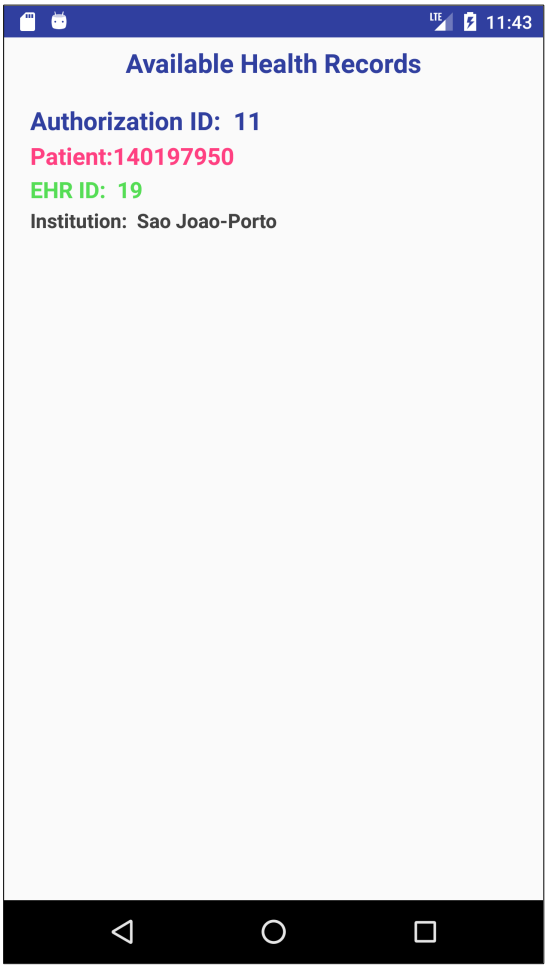

Figure 6.20: View of shared EHRs using myHEnCEPRO.

On the other side, the patient can revoke the access to all his EHRs, to a specified health professional. This option is present in myHEnCE main menu activity, by pressing the button 'Revoke Access'. Then, the patient needs to insert the CC of the health professional that he wants to revoke all access permissions (screenshot in figure 6.21). The respective AT is revoked from the list of health professional ATs, but not from the ACL, because all previous accesses must be recorded and logged. In this step there is a verification if the health professional exists, and an audio?? alarm informs the user if the revoke tentative was successful or not.

In myHEnCEPRO when the health professional views the shared list of EHRs (screenshot in figure 6.20), if he tries to access a revoked EHR, he cannot view it and hears an alarm informing that he can no longer have access permissions to that EHR (screenshot in figure 6.22).

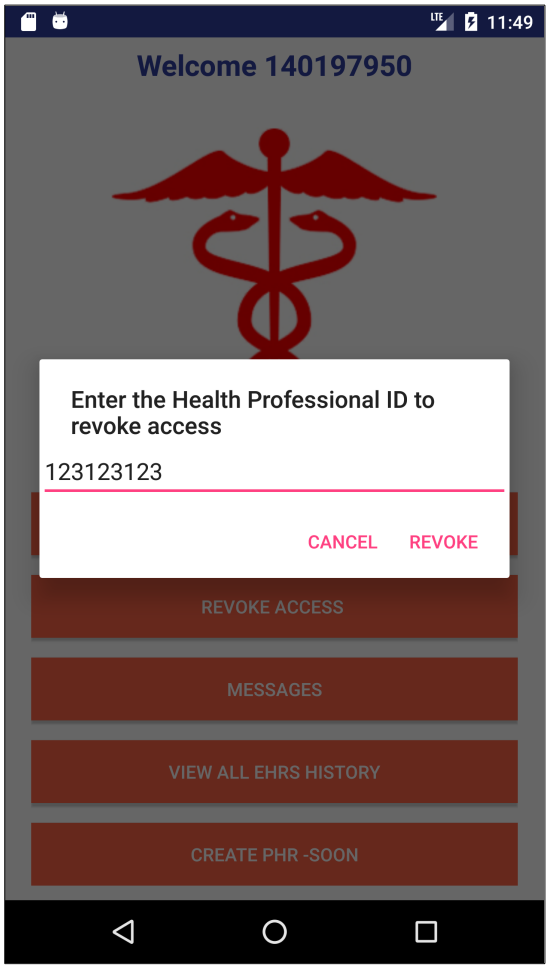

Figure 6.21: Revoke access to an EHRs using myHEnCE.

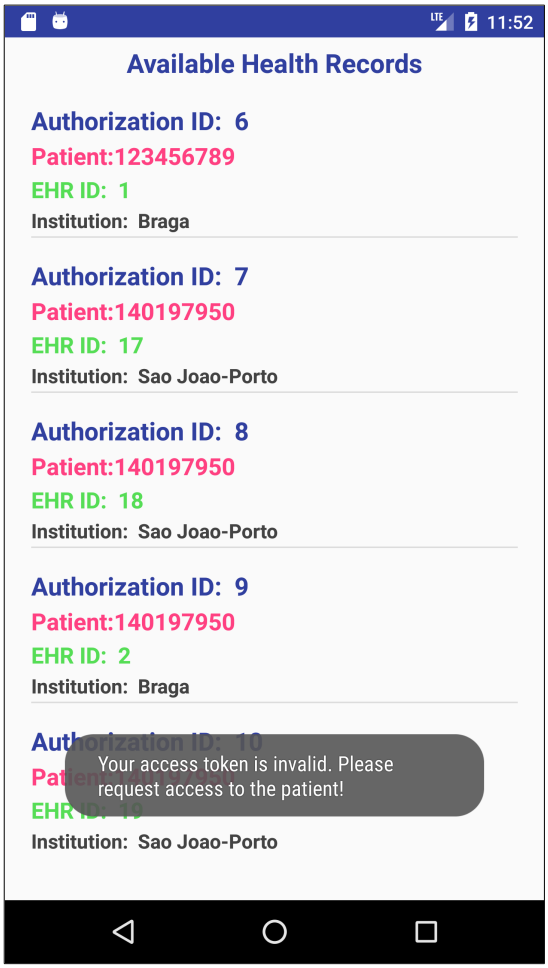

Figure 6.22: Access a revoked EHRs using myHEnCEPRO.

6.3 Testing

Testing is important to detect and fix errors and bugs. With the correct tests is possible to get an overview of the performance and check the fulfillment of the requirements. Major security flaws can be avoided with a good and iterative testing.

Besides the classic static analysis, some tools were used to eliminate bugs and security checks. To test the RESTful Web services (IdP and SPs) Advanced REST Client [182] was used as it makes a connection directly to the socket giving full control over the connection and URL request/response headers. This way it is possible to analyze and test all headers before inserting them in the mobile applications. Manual audit weak TLS cipher levels can be performed with openSSL and the following command:

```
root@root:~#  openssl s_client -connect 192.168.1.138:443
```

The checklist for testing weak encryption, bad hash and not recommended algorithms is based in OWASP recommendations for test weak encryption [183].

The mobile apps were tested in Android versions Marshmallow and Nougat, and with a physical device Huawei p9 lite. Besides the static and dynamic analysis to eliminate the bugs, to test the

privacy preservation and information leaks, myHEnCE and myHEnCEPRO were tested with Droid-Infer [20]. This tool is a type-based system (namely safe, tainted, poly), which performs static taint analysis without running application (in this case without the need to install the android application in some device), to check leaks of information. The possible leaks are considered to occur by console logs or HTTP requests. And the sources are mostly from `get()` methods [20]. To use this tool there is the need to have Android installation `.apk` of the application, generated in Android Studio and an UNIX-based OS to run the tool. The `.apk` must be generated and digitally signed. But, for test purposes the non-signed `.apk` was used. The results are equal in both mobile applications:

```
INFO:Source:10
INFO:Sinks:109
```

These values are attenuated because the application needs to collect some sensitive data (such as location, IMEI) to calculate the risk and handle authentication, and the requests are based in HTTPS. Nevertheless the code must be reviewed to reduce the number of sources and sinks.

Chapter 7

Conclusion and Future Work

This chapter presents a discussion and synthesis of the main achievements of this thesis and points to several directions for future work.

7.1 Conclusions

With the arrival of the new General Data Protection Regulation (GDPR) in May of 2018 [184], all HIS must be reviewed to match the new requirements. It is fundamental to perform the essential security analysis and the requirements checklist to ensure user's privacy and rights. The integration with the new GDPR will have costs and time to companies to adapt to the new reality, which may slow the evolution of eHealth. However it is essential to integrate full security controls in order to release the full potential of eHealth. That potential is not being fully used, with still a long road ahead. In this work we followed a uniform methodology for health data organization and storage, aiming at a simplified access and extraction of information. But different geographically fragmented institutions still present lack of interoperability and single data standardization. The implementation of central authentication and access control, can be a hard task because each institution can follow different standards for storage, different databases types, in general different systems. Integrate all in a central system is an on-going challenge. Also the performance of this kind of services depends of the computational and memory limitations of the hardware in each institution. The handling of several requests at the same time can lead to delays or time outs, reducing availability.

The main objective of this dissertation was the research and implementation of a ubiquitous solution for sharing and accessing health data, in a transparent and secure way. It was proposed an user-centric architecture, based in recommendations from OWASP [50] , NIST 800-53 [40] and NIST 800-57 [165], to provide to the patient means to keep his/her private health data as close as possible. To do this was carried out with the implementation of two mobile applications (one for patients and other to health professionals), an IdP and a set of SP. Within the research process, it was found a lack in the existent access control models.
This led to the innovation of a new dynamic access control model, namely SoTRAACE [26], published and presented at the IEEE 51st International Carnahan Conference on Security Technology, in Madrid. This model fills the gap between restrict and predefined policies in access control, providing a flexible access policy based on the assessment of the aggregated environmental risk, determined at the precise moment of the request. As a complement, the risk evaluation is based in a Delphi study, conducted engaging a set of security experts. The results were used to determine the relative weights of the identified risk factors. These weights are essential to calculate the risk level that SoTRAACE receives to make its access decisions. The proposed access control model faces some challenges, such as it is hard to compute the (humanly related) operational need without more information regarding users' tastes, experiences and social interactions beyond the ones that are within the reach of the mobile devices. Also, GPS location

can only be used if it is turned on, and accept to acquire location by the user. The authentication and identity management could use third parties services, such as Google and Facebook authentication. However, the way how these companies manage user data is not clear. So this option was excluded.

The architecture works as expected, assuring secure authentication with multifactor, and secure data share/access based in SoTRAACE decisions. At last, the architecture was tested within different Android versions (Marshmallow and Nougat) and with a physical device Huawei p9 lite. Also targeted with static and dynamic analysis. These analysis prove that the system works without bugs and unexpected behaviour.
To audit the certificate, security algorithms and measures was used openSSL [185]. The results of the audit process prove that only recommended algorithms [183, 165] are used in the system, and that the certificate is correctly configured [176, 177].
Leaks of information can compromise the privacy of the user's. To check the possibles sources of sensitive information and their respective leaks was used DroidInfer [20]. The results presents 10 sources and 109 sinks. The reason for these high numbers is that the mobile applications collects sensitive information (such as location, IMEI) to calculate the risk and handle authentication. Such information is considered as source by DroidInfer. And because it flows through a HTTP/HTTPS request, DroidInfer consider it as a sink. The system only uses HTTPS requests to protect the transmission of these sensitive information. Yet, the code must be reviewed to reduce the number of sources and sinks.
Also Advanced REST Client [182] was used to test and verify the URL request/response headers.

7.2 Future Work

The main objectives for this work were achieved, but there is still room for additions and improvements, as future work. The synthesized HIS platform has a wide range of possibilities for improvements and additional modules, such as the management of PHR, BTG access and delegations based in relationships. As an extension of this work, all health data must be standardized, for instance with openEHR [186] standards.
To avoid a lack of availability in the HIS, all web servers must be replicated, one node with two servers, master and slave. If for some reason the master is down, the slave takes place. Data synchronization should be at real time between them to ensure no data is lost. Future work includes the implementations of the mobile applications in other OS, such as Apple iOS.

The risk evaluation method based in the Delphi study should be more fine-tuned and extend the security controls applied to the risk output. Also there is the aim to improve human interaction, behavioral algorithms to better calculate operational need metrics.

Also future work includes testing the system with more security tools, such as SSLLabs [187] for certificate validation, and OWASP Zed Attack Proxy [168] for vulnerabilities scan.
Most important, future work must consider collecting users' opinion by releasing the system to the real world, and testing more thoroughly our framework with many concurrent users.

Appendix A

Appendix

A.1 SoTRAACE-Socio-Technical Risk-Adaptable Access Control modEl

SoTRAACE - Socio-Technical Risk-Adaptable Access Control Model

Pedro Moura* †, Paulo Fazendeiro†, Pedro Marques*,
Ana Ferreira*
*CINTESIS - Center for Health Technologies and Services Research,
Faculty of Medicine, University of Porto
pedromoura@med.up.pt, pmarques@med.up.pt, amlaf@med.up.pt
†Department of Computer Engineering, University of Beira Interior
pandre@di.ubi.pt

***Abstract*—Within the necessary security requirements, access control measures are essential to provide adequate means to protect data from unauthorized accesses. However, current and traditional solutions are commonly based on predefined access policies and roles and are therefore inflexible by assuming uniform access control decisions through people's different type of devices, environments and situational conditions, and across enterprises, location and time. We live in an age of the mobile paradigm of anytime/anywhere access as the smartphone is the most ubiquitous device that people now hold. In this new age, access control models need to determine adaptable access decisions based on multiple factors aggregated at the moment of request and not just perform a predefined comparison of attributes. This paper presents a new access control model: SoTRAACE - Socio-Technical Risk-Adaptable Access Control Model. This model aggregates attributes from various domains to help performing a risk assessment that is balanced against the operational needs at the moment of each request, so to provide the most accurate and secure access decision. As a proof of concept, SoTRAACE is used to model and compare two different use case scenarios in the healthcare sector.**

***Index Terms*—Health Data privacy, Risk Adaptable Access, Socio-technical Systems, Ubiquitous Mobile Access.**

I. Introduction

Smartphones are the most ubiquitous devices that people hold nowadays. Due to their portability, availability, ease of use, communication, information access and sharing within various domains and areas of our daily lives [1], the acceptance and adoption of these devices is still growing. They allow share and access to Internet services and data anywhere and anytime. A Google report [2] shows this with many interesting facts about the way people use their mobile devices. In this research, Google found that 68% of phone users say they check their phone within 15 minutes of waking up and 87% always have their smartphone at their side, day and night.

However, due to their potential and raising numbers, smartphones are a growing target for attackers and, as with other technologies, mobile applications are very vulnerable [3].

In the healthcare domain, smartphones can bring many advantages to tackle heterogeneous needs of stakeholders. On one hand, health professionals can use smartphones to access and create patient records (e.g, Electronic Health Record (EHR)), to view exam results, to share and ask for second opinion diagnosis, and to prescribe medications [4]. On the other hand, the patient, can use smartphones to access, update and control access to their medical records, monitor their health statistics and view their prescriptions [5].

Health Information Systems (HIS) can empower the performance and maintenance of health services but their storage of highly sensitive data raises serious concerns regarding patients' privacy and safety [6]. Even though health data are subjected to legal and regulatory restrictions [7], [8], according to [9], in 2015, 39% of all data breaches that occurred within the Services sector were targeted to healthcare.

Traditional solutions for access control are inflexible because the access control policy is hard-coded and pre-set into decision logic or database restrictions. Moreover, they assume uniformity of people's devices, environments and situational and technical conditions, across enterprise/location and time. With the new mobile paradigm of anytime/everywhere, from different mobile devices and Internet wireless connections, there is a need to search for more innovate, flexible, adaptive, dynamic, transparent and more resilient access control models, that are required for more heterogeneous requests.

This paper presents such model. SoTRAACE (Socio-Technical Risk-Adaptable Access Control Model) aggregates various environmental, technical, social and user profile attributes to help performing a risk assessment at the moment of each request. The risk assessment is balanced against the operational need to provide the most accurate and secure access decision possible. As a proof of concept, SoTRAACE is used to model, compare and discuss two different use case scenarios in healthcare.

II. Related Work

One of the most common used access control models in healthcare is the Role-Based Access Control Model (RBAC)[10] which includes the concepts of users, operations, sessions and roles. It allows a more streamlined management of the policies in an organization. Instead of dealing directly with privileges (permissions) per user, the users are associated to roles and each role is associated with labels and privileges. RBAC is policy based and can be adapted to match with

978-1-5386-1585-0/17/$31.00

data legislation's requirements, such as HIPPA [7] privacy guidelines for accessing patient health records and International Organization for Standardization (ISO) norms (e.g, ISO 13606) for healthcare.

Other variations with some traits of flexibility have been defined since. Attribute-Based Access Control (ABAC) [11] is identified as an access control model which is similar to RBAC in the sense that it also adopts a policy driven approach. ABAC is more flexible than RBAC because it uses the attributes of subjects and objects together with environmental attributes to make access decisions, instead of roles.

Situation-Based Access Control (SitBAC) [12], defines a situation as an abstract condition which is composed of user's contexts and related object contexts. SitBAC is a conceptual model, which defines scenarios expressed via situation instances where patients data access is permitted or denied.

Location has also been taken in consideration within some access control models, most of them using Global Positioning System (GPS) technology. An example can be found in [13]. This model uses Geographic Information System (GIS) as a support to make the best evaluation about location and related parameters. With mobile devices and enormous variety of connections which those devices use, these could be useful models for the near future. Location can be used to control access, for instance, if a doctor is outside the range of an hospital, s/he cannot access data regarding their patients.

Recent years have witnessed the growing popularity of Social Network Systems. This concept in healthcare can be reflected in the future with the provision of a more close relationship between patients, health professionals and patients' family. Different levels of friendship for different permissions can be provided. Based on this, the Relationship-Based Access Control (RelBAC) [14] is characterized by the explicit tracking of interpersonal relationships between users, and the expression of access control policies in terms of their relationships. For instance, patients with Alzheimer or other mental illnesses should be provided with a relationship network (e.g., someone they relate and trust - family, friend, health professional) to help manage their medical records.

To improve flexibility in real healthcare settings with the occurrence of errors and/or unanticipated or emergency situations, Break the Glass (BTG) can be used to break or override access controls in a controlled manner, when needed. BTG-RBAC [15] can help in cases such as when a user has not been given the proper roles or permissions and when authentication (e.g, central authentication system failure) or authorization problems (e.g. an emergency situation such in an ambulance with the patient unconscious) may occur.

Finally, there are a few models that try to adapt access control decisions according to the situation and context at the moment of request. Risk-Adaptable Access Control (RAdAC) [16] introduces the idea of balancing security risk against operational need. This is made with the belief that the operational benefits of sharing the information outweigh the potential security risk of sharing it. The basis for making decisions is an understanding of the operational need, the resultant security risk, the policies and operating procedures governing the situation, and the knowledge of the effects of similar decisions from the past [16]. Security policy grants or denies can be reversed according to the operational need and security risk at the moment of the requested access. However neither this model nor a more complete version [17] include social and behavioural factors, trust levels, granularity of the objects, devices with different Operating System (OS), location or even the BTG component, to help make the most accurate and adaptable access control decisions.

To calculate and evaluate risk some methods have been proposed. Cheng et al. have presented a Fuzzy Multi Level Security (MLS) access control model [18] that quantifies the risk associated with an access by calculating it based on a value of information and probability of unauthorized disclosure.

Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) [19] focuses on building an organization wide view of information security risks using three components: riskthreat, asset, and vulnerability.

Common Vulnerability Scoring System (CVSS) [20] provides network risk assessment, using an equation which considers impact from the vulnerability exploitation, and probability of its exploitation, that correlates with definition of risk.

Other rating system for quantifying risk is DREAD (Damage potential, Reproducibility, Exploitability, Affected Users, Discoverability) [21] which rates risk by answering five questions relating to each of those five categories. We will use this method later in the paper to test our use-case scenarios.

III. SoTRAACE Model

A first step in the development of an access control model is the identification of the objects to be protected, the subjects that execute activities and request access to objects, and the actions/operations that can be executed on the objects. The SoTRAACE model is presented in Fig. 1 and its components are described next.

A. SoTRAACE NIST RBAC-BASED Components

Users: a User entity is a human being requesting an access operation to an object, through a mobile device.

Roles: Users can be mapped to different roles, each role with different associated permissions and restrictions. In this paper our use-cases focus on a patient-centric solution only.

Sessions: Each session is associated to a user and their roles, and registers what a user does (tracking logs) for a specific period of time, what resources were accessed and what operations have been made, from which device and with which connections. With this, the system has enough data to learn about user's legitimate and compromised access behaviour and history overtime, enabling the building up of heuristics/profiling data for subsequent access requests. In our model, each session will provide information to decide in which conditions the requested data will be retrieved to

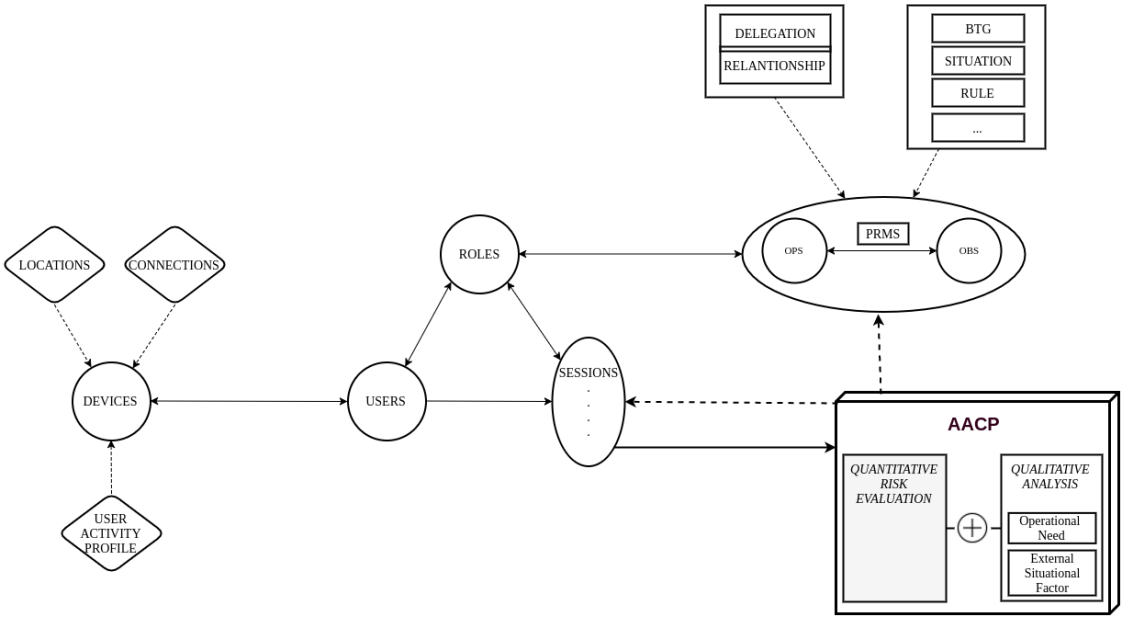

Fig. 1. SoTRAACE - Socio-Technical Risk-Adaptable Access Control Model.

the user and be, afterwards, adaptable according to the risk calculation. More in the description of the AACP component.

Objects (OBS): An object is an entity that contains or receives information.

In Healthcare, an object can have different degrees of granularity representing a complete HIS or just documents (e.g discharge notes), or even a particular set of data within a document (e.g genetic information), each potentially having different degrees of sensibility.

Permissions (PMRS) and Operations (OPS): An operation is an executable of some function for the user.
A permission is an approval to perform an operation on one or more objects.

The next items describe examples of existing constraints that can turn the use of PRMS in SoTRAACE more flexible.

Break The Glass (BTG): BTG [15] is used to override pre-defined access in a controlled manner, with strong auditing measures to make users responsible for their requests and justify them, whenever necessary [15]. In SoTRAACE, if a patient in a hospital is incapable of communicating if s/he is allergic to a specific medication and the nurse treating him/her can have the justified possibility of overriding pre-defined policies and accessing patient's allergy data, before administering it.

Situation and Rule: Situation factors are considered in our model [12]. One example of a situation factor is a request for data access for research purposes. This case comes with different obligations/constraints than when compared to a normal data request. Situation can also be a location, the user or the object, or can be external to all defined parameters. Local situations are directly applied to the access control policy (PRMS: OPS, OBS).

Specific static rules can also be set by the user, health professional or institution and be directly applied to the access control policy.

Relationship/Delegation: Users can establish a relationship with family, friends or healthcare professionals using RelBAC [14]. What differs for delegation is mostly the fact that relationship uses the direct concept of Social Networks Systems. In healthcare: Bob as an obstetrician requests access to Maria's pregnancy data. If she accepts the request, a relation is created in PRMS, to which she can add constraints.

B. SoTRAACE new components

Devices: A Device is defined as an entity that aggregates several contextual, behavioural and situation attributes (e.g., location, connection, user activity profile - UAP) and allows the user to request access to objects in the session, taking into account these previously collected attributes, which are used to evaluate the risk in the Adaptable Access Control Policy (AACP). In our study, the Device is a mobile device. A device can only allow access if the respective IMEI is registered within the user's profile at the server's side otherwise, the user needs to add that device (using a multi-factor authentication). We can define other device attributes to use in the risk evaluation such as the type of OS (e.g., android is proven to be more unsecure than iOS) or the use of older, non-updated

versions, which are more prone to be exploited.

User Activity Profile (UAP): A User Activity Profile is defined as a set of attributes that contains history information regarding all associated user's devices and locations through all the user's sessions, as well as information about user's previous accesses and used connections. To enable audit and learn about the user's behaviour, each user's session is registered in the UAP. This allows an easy search for malicious or legitimate behaviour over different sessions.

Locations: In our model, Locations is defined as a set of attributes. We will use GPS sensors from mobile devices to track the most common user's locations to build a user profile location and check access and calculate risk having into account location history and other current parameters. A profile history of regular access from Portugal may help to raise suspicious of unauthorized access when there is evidence of interleaved accesses from Australia within a reduced time gap. A user may not want to share his/her location due to privacy issues. In this case, the locations set will be empty. If a user wants to add a new device or new location, a multifactor authentication is required.
UAP, Session and Device can have associated/registered none or many Locations.

Connections: A Connection is defined as a set of attributes and is a communication tunnel that binds two end points (e.g, device and server or device and user) to exchange information. In mobile devices, the first evaluation is to determine if the connection is made to a mobile Service Provider (e.g, 3G,4G) or to a wireless network. SoTRAACE can evaluate the encryption algorithm, the length of the encryption key, used protocols, if the connection is password protected, the SSID, and so on.

It also compiles a few questions to help with the risk connection evaluation: *How many users are connected to the wireless network? How many wireless networks are available in the vicinity?*

UAP, Session, Device can have one or more Connections.

The next subsection presents the main engine of SoTRAACE where risk adaptable features can verify and adapt to the environment and user who is requesting access.

C. Adaptable Access Control Policy (AACP)

RAdAC [16] introduced a base definition for the core characteristics of security risk evaluation, operational need, external situation factors and adaptable access control decisions. We will adapt them into our model and add other features.

For quantifying the security risk of each request, the AACP aggregates, in real time, all attributes that are instantiated in the session, namely connection, location and the user activity profile from the device. It also aggregates data from the object descriptive metadata (e.g., type, sensitivity level of the requested resource, owner, institution/company related) as well as the object logs (who/when/where that object was accessed or changed). Each attribute can contain exploitable threats that will be used to perform the quantitative risk evaluation by anticipating how and what is the probability of that security flaw being exploited (e.g. without the use of https, there is a higher probability that user credentials can be stolen).

The quantitative risk evaluation is complemented with more qualitative measures such as the operational need and external situation factors to provide a more accurate, secure and adapted access decision.

This can be understood as the need to access the requested object and can influence more or less the already measured/quantified risk. In our model, the operational need is dynamic as is also defined through other aggregated attributes (connection, location, device, etc), roles, user and situations but evaluated at a different, less objective but more human behavioural light.

As an example of a more qualitative risk evaluation: if a nurse is trying to access a medical record at a different time from her normal working hours, using a different device and connection, the calculated quantitative risk will be higher than usual. However, a more qualitative analysis may attenuate that risk if it confirms that the nurse is accessing data that is customary. In this case, auditing can register some warnings and visual security restrictions can be applied as a preventive measure.

After having calculated the total risk, AACP specifies a set of rules (the decision) that can be applied to PRMS, and under which conditions. These dynamic decisions and rules can: 1) block or allow the access; 2) enforce the fragmentation of the object and just allow access to some fragments; 3) block one or more operations to the object; 4) trigger other hidden security protocols to better avoid the risk; 5) in certain situations different levels of security can be afforded; 6) in situations of extreme operational need and high risk, more secure channels for communications or different cryptography techniques can be used.

However, these high levels of security can be heavy performance wise. For instance, if the risk is low and the operational need is also low, perhaps is not necessary to waste such heavy-cost in security resources but opt for more user interaction security options, which can also empower the use of older devices that some users can still have. To do this, AACP decision rules can also be applied at the Session level where AACP can change the visualization of the requested data, providing dynamic ways to present the object to the user, containing still the requested object but presented, perhaps, in a more categorized/ordered way, not showing all data at once or hide some data that AACP has some degree of certainty that is never useful to that patient.

Transparency is also a must and the user can find out, at all times, what the model is doing, why is doing it, with the option to provide information and tips about risk evaluations and past decisions. Finally, past decisions and respective parameters provided by the AACP will be used to help decide each subsequent decision. This knowledge can be used to improve algorithms that determine the risk, operational need and the

rate of positive access control decisions, to build more accurate UAP and object logs.

IV. Use Cases

ObsCare is an Obstetrical EHR service that stores and manages data related with pregnancy, nutrition, genetic data and general health information. *ObsAPP* is a mobile application to access, upload, share and control *Pregnancy/Nutrition* related records and includes a Pregnancy specialized Nutrition module. Maria, is a pregnant woman registered in the *ObsCare* service and has the *ObsAPP* installed in two devices. Device *A* is registered in her profile, and runs on iOS. Device *B* is running Android Nougat OS (considered unsecure) and is not registered in the system or in her profile. Usually Maria does login between *2pm* and *10pm*, and commonly from *Porto*.

A. Use Case A

Maria wants to access her related pregnancy data at *2am* using *ObsAPP*. She is in a new location, *Braga*, with a new Device *B* so these need to be registered in her profile, using a multi-factor authentication. The session is created with the role *Patient*. SoTRAACE analyses the connection, and recognizes a WiFi with *WEP* and no password, verifies that the SSID is *CoffeMarket*, and that there are more than fifty users connected to this network. Her mobile device also recognizes that there are other nine wireless connections available nearby. Triangulating the SSID with the location and GPS web mapping service (e.g, Google Maps), SoTRAACE finds a Coffee shop nearby named Market. Maria requests her available pregnancy's related document list, which is a regular activity for her and with no previous security issues. SoTRAACE quantitative risk evaluation is high (Table I). However, since Maria is requesting data that she successfully accesses on a regular basis, the operational need minimizes the calculated risk. Due to this evaluation, Maria receives in *B* a list with her requested objects but visually filtered, showing only their date and name and hiding details such as the name of the health institution where they were performed, and so on. This risk evaluation could also imply other access restrictions.

TABLE I
An Example of DREAD Risk Evaluation in Use Case A.

Category	Score	Rational
Damage	10	Unauthorized access, view or change high sensitive information.
Reproducibility	7	The threat can happen in a racing situation (object in the communication channel) or at the end device.
Exploitability	6	A skilled programmer could make an attack to the insecure network or to the vulnerable device. Credentials and device stolen
Affected Users	3	Object owner, his family, related doctor
Discoverability	10	Always assumed to be 10 by default
DREAD SCORE = 7, High Risk		

B. Use Case B

Maria wants to access her related Nutrition records data using *ObsAPP*. She performs login at 3pm in Device *A*, and is inside the location of *Porto* (all these data are previously registered in her profile). Previous accesses are commonly made from that location or with that same device. After successful authentication, the session is created and her role is *Patient*. SoTRAACE analyses the connection, and recognizes a WiFi with *WPA2*. The SSID is *MariaHouseSecure*, and there is no other user connected to this network. Her mobile device also recognizes that there is only one wireless connection available nearby. Triangulating the SSID with the location and GPS web mapping service (e.g, Google Maps) SoTRAACE confirms the area where Maria lives. Maria requests her available Nutrition's related EHR documents. SoTRAACE aggregates all related attributes to perform the risk evaluation for this request whose output is low (TABLE II). The operational need for Maria's request is also low. The requested EHR documents are displayed in *A* without any restrictions.

TABLE II
An Example of DREAD Risk Evaluation in Use Case B.

Category	Score	Rational
Damage	2	Unauthorized access, view or change low sensitive information.
Reproducibility	2	The attack is very difficult to reproduce, even with knowledge of the security hole.
Exploitability	2	The attack requires an extremely skilled programmer, or stolen credentials/device.
Affected Users	1	Object owner
Discoverability	10	Always assumed to be 10 by default
DREAD SCORE = 3.4, Low Risk		

V. Discussion

SoTRAACE is based on standard access control models as well as other peer-reviewed models but also integrates some new features that the authors believe are very important in today's mobile paradigm. This integration enriches the model and at the same time provides easy adaptation to various technical, contextual and user needs. This is the case of the healthcare domain where heterogeneous health professionals, type of data, different security levels, regulations and need of accesses, require an adaptable but still secure model. Although we could only describe two use cases with the patient role, due to space constraints, it would be easy to adapt to different and more complex cases.

SoTRAACE integrates novel features such as the analysis and evaluation of socio-technical risk. It calculates not only a quantitative measure of all the parameters but also a more qualitative one that either minimizes or strengthens the objective value. It provides a more complete analysis without just bluntly denying or providing access to resources independently of other vulnerabilities that should be considered in the access decision. Our model also takes advantage of user and object profiling data - so strong audit features need to be used not only to control but also at the service of the user - to better calculate both types of risk and providing more decision options, as well as better usability.

SoTRAACE access decisions can include access with: 1) extra restrictions; 2) improved security mechanisms without

affecting user's access to the object; 3) warnings to the user, if s/he thus requires; 4) and adaptable security visualization without compromising both security and user's request. This fine tuning is helpful to adopt SoTRAACE to different domains or types of security requirements as for instance, in government or banking where security is highly demanded, AACP can take more restrictive and controlled decisions than, for instance, in research or education where it is more important to have access to more information, faster, but in a secure and trustable way. Secure visualization for common users is still a very incipient field but the authors believe that it is a domain to focus research. A simple, clean and organized view of data can possibly prevent many security hazards, in different situations. We plan to focus more work on this topic. Also, our model can be adopted in systems with other types of devices.

Limitations:

Due to space constraints, is not possible to describe in detail SoTRAACE new features nor the analysis of the two use cases. However, as a preliminary proof of concept, SoTRAACE can easily aggregate and analyze different contextual parameters and user's situations to provide the most secure and usable decision, at every different moment. We also foresee some difficulties in choosing the most adequate risk calculator for every situation, there are no standardized ways to do this, but we can still use the ones at hand and improve them. Similarly, it will be hard to compute the (humanly related) operational need without more information regarding user's tastes, experiences and social interactions beyond the ones that are within the reach of the mobile devices. Also, GPS location can only be used if it is on.

VI. Conclusion and Future Work

This paper presents SoTRAACE, a new adaptable access control model based on quantitative and qualitative risk evaluation, that easily integrates with healthcare domain heterogeneous needs. It is also well set for the new mobile paradigm challenges as well as to nowadays security requirements and human interaction related needs, such as adaptable security visualization. It is an innovative first step in the way to integrate profiling techniques at the service of the patient and not only at the service of business and marketing organizations.

As future work, we intend to: test and improve risk calculation and adopt several types according to security and user requirements; improve human interaction and behavioral algorithms to better calculate operational need metrics; implement and test a prototype in an healthcare institution that uses *ObsCare*; and perform threat analysis and evaluate both SoTRAACE's security, usability and performance requirements.

Acknowledgments

This work is funded by NORTE-01-0145-FEDER-000016 (NanoSTIMA) which is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

References

[1] Park, Yangil and Chen, Jengchung V, *Acceptance and adoption of the innovative use of smartphone*, Industrial Management and Data Systems 107 , no. 9 (2007): 1349-136

[2] Google, *Micro-Moments: Your Guide to Winning the Shift to Mobile* 2015, Available: https://think.storage.googleapis.com/images/micromoments-guide-to-winning-shift-to-mobile-download.pdf

[3] Open Web Application Security Project (OWASP), *Mobile Security Project*, Available : https://www.owasp.org/index.php/OWASP_Mobile_Security_Project

[4] Ozdalga E, Ozdalga A, Ahuja N, *The Smartphone in Medicine: A Review of Current and Potential Use Among Physicians and Students*, Journal Of Medical Internet Research 2012;14(5):e128

[5] M. Plachkinova, S. Andrs and S. Chatterjee, *A Taxonomy of mHealth Apps – Security and Privacy Concerns*, 2015 48th Hawaii International Conference on System Sciences, Kauai, HI, 2015, pp. 3187-3196.

[6] G. N. Samy, R. Ahmad and Z. Ismail, *Threats to Health Information Security," 2009 Fifth International Conference on Information Assurance and Security*, Xi'an, 2009, pp. 540-543.

[7] Health Insurance Portability And Accountability Act (HIPPA), *Public Law 104191- Aug. 21* 1996, Available : https://www.gpo.gov/fdsys/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf

[8] European Union, *Regulamento (UE) 2016/679 do Parlamento Europeu e do Conselho (Portuguese Version)*, 2016, Jornal Oficial da Unio Europeia L 119, 68.

[9] Symantec, *Internet Security Threat Report*, Volume 21, 2016, Available : https://www.symantec.com/content/dam/symantec/docs/reports/istr21-2016en.pdf

[10] Ravi Sandhu, David Ferraiolo, and Richard Kuhn. 2000. *The NIST model for role-based access control: towards a unified standard.* In Proceedings of the fifth ACM workshop on Role-based access control (RBAC '00). ACM, New York, NY, USA, 47-63.

[11] H. S. G. Pussewalage and V. A. Oleshchuk, *An attribute based access control scheme for secure sharing of electronic health records*, 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), Munich, 2016, pp. 1-6.

[12] Mor Peleg, Dizza Beimel, Dov Dori, Yaron Denekamp, *Situation-Based Access Control: Privacy management via modeling of patient data access scenarios*, Journal of Biomedical Informatics, Volume 41, Issue 6, 2008, Pages 1028-1040, ISSN 1532-0464.

[13] Y. Jin, M. Tomoishi and S. Matsuura, *Enhancement of VPN Authentication Using GPS Information with Geo-Privacy Protection*, 2016 25th International Conference on Computer Communication and Networks (ICCCN), Waikoloa, HI, 2016, pp. 1-6.

[14] Philip W.L. Fong. 2011. *Relationship-based access control: protection model and policy language*. In Proceedings of the first ACM conference on Data and application security and privacy (CODASPY '11). ACM, New York, NY, USA, 191-202.

[15] A. Ferreira et al., *How to Securely Break into RBAC: The BTG-RBAC Model*, 2009 Annual Computer Security Applications Conference, Honolulu, HI, 2009, pp. 23-31.

[16] R. McGraw - National Security Agency. *Risk-Adaptable Access Control (RAdAC)*. NIST Privilege (Access) Management Workshop, 2009

[17] S. Kandala, R. Sandhu and V. Bhamidipati, *An Attribute Based Framework for Risk-Adaptive Access Control Models*, 2011 Sixth International Conference on Availability, Reliability and Security, Vienna, 2011, pp. 236-241.

[18] P. C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner and A. S. Reninger, *Fuzzy Multi-Level Security: An Experiment on Quantified Risk-Adaptive Access Control*, 2007 IEEE Symposium on Security and Privacy (SP '07), Berkeley, CA, 2007, pp. 222-230.

[19] Richard A. Caralli, James F. Stevens, Lisa R. Young, William R. Wilson, *Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process*, Software Engineering Institute, CMU/SEI-2007-TR-012, May 2007

[20] FIRST - Improving Security Togheter, *Common Vulnerability Scoring System v3.0: Specification Document*, 2017 by FIRST.org, Inc

[21] J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, *Improving Web Application Security: Threats and Countermeasures*, Microsoft Corporation, June 2003

A.2 Security Risk Evaluation in Health Information Systems - Form sent to the Experts

Security risk evaluation in Health Information Systems

This survey is part of a Delphi study developed in the context of a MSc dissertation in computer science and engineering. Its purpose is to identify and classify risk factors in the access to electronic health records. The objective is to obtain a deeper insight about the effective and perceived risk associated with some of the environmental variables of each request.

Please classify the impact that the referred attribute can have in the overall security of a health information system.

*Required

1. **Type of wireless connection and respective encryption (e.g. WEP with RC4, WPA2 with AES, 3G with Kasumi).** *
   Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

2. **Patterns in the SSID or profile of a wireless connection (e.g. free, open, coffee, guest, hotspot, home, public).** *
   Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

3. **Security mechanisms of the communication protocol (e.g. HTTP, HTTPS, connection via VPNs).** *
   Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

4. **Location where the request is being made (e.g. big cities, public places, schools).** *
   Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

5. **Number of wireless networks reachable in the present location.** *
   Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

6. **Information sensitivity of the requested health record (e.g. public, personal or private data).** *
   Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

7. **Known vulnerabilities associated to a device / OS version (e.g. IOS, Android nougat or marshmallow).** *
   Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

8. **Role of the person trying to access a resource (e.g. physician, nurse, chiropractor, operator of ambulance, chief doctor, patient).** *
Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

9. **Number of mobile devices that the user has registered.** *
Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

10. **Occurrence of recent reports of global security threats and vulnerabilities.** *
Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

11. **Observable behavioural differences regarding time and location of the person who is performing the request (e.g., login in Australia two hours after a successful request in Portugal).** *
Mark only one oval.

	1	2	3	4	5	
Negligible	◯	◯	◯	◯	◯	Critical

12. **Comments or suggestions regarding these or other attributes that, in your opinion, are important for the risk evaluation.**

Bibliography

[1] Symantec, "Internet security threat report, volume 21." https://www.symantec.com/content/dam/symantec/docs/reports/istr-21-2016-en.pdf, April 2016. Online, accessed April 2017. 1, 2

[2] B. M.K and R. K. G.K., "Securing big data over network using md5 algorithm technique," *International Journal of Computer Applications (0975 – 8887)*, vol. 123, August 2015. 1

[3] J. Couzin-Frankel, "Trust me, i'm a medical researcher," *Science*, vol. 347, no. 6221, pp. 501–503, 2015. 1

[4] W. A. Conklin, "It vs. ot security: A time to consider a change in cia to include resilienc," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, pp. 2642–2647, Jan 2016. 1, 8

[5] M. Peleg, D. Beimel, D. Dori, and Y. Denekamp, "Situation-based access control: Privacy management via modeling of patient data access scenarios," *J. of Biomedical Informatics*, vol. 41, pp. 1028–1040, Dec. 2008. 2, 8, 22, 24, 37

[6] L. Grandia, "Healthcare information systems: A look at the past, present, and future," *Health Catalyst*, 2016. 2

[7] G. S. de Souza, R. C. M. Correia, R. E. Garcia, C. Olivete, and B. R. G. Santos, "Health information system for medical survey analysis," in *2016 11th Iberian Conference on Information Systems and Technologies (CISTI)*, pp. 1–6, June 2016. 2

[8] J. Huang, M. Sharaf, and C. T. Huang, "A hierarchical framework for secure and scalable ehr sharing and access control in multi-cloud," in *2012 41st International Conference on Parallel Processing Workshops*, pp. 279–287, Sept 2012. 2

[9] G. S. Ayatollahi H, Bath PA, "Paper-based versus computer-based records in the emergency department: staff preferences, expectations, and concerns.," *Health Informatics Journal*, 2009. 2, 26, 27

[10] Health Insurance Portability And Accountability Act, "Public law 104–191—aug. 21, 1996." https://www.gpo.gov/fdsys/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf. Online, accessed March 2017. 2, 8, 14, 19, 27, 43

[11] U.S. Department of Health & Human Services, "Health information privacy : Hippa." https://www.hhs.gov/hipaa/. Online, accessed March 2017. 2, 27, 43

[12] Committee of Ministers to Member States, "Protection of medical data - recommendation no.r (97) 5, 1997." https://rm.coe.int/CoERMPublicCommonSearchServices/DisplayDCTMContent?documentId=09000016804f0ed0. Online, accessed March 2017. 2, 8, 19, 20, 27, 43

[13] European Union, "Regulamento (UE) 2016/679 do Parlamento Europeu e do Conselho (Portuguese Version)," *Jornal Oficial da União Europeia L 119*, vol. 68, May 2016. 2, 8, 19, 20, 27, 35, 43

[14] C. . C. N. de Proteção de Dados, "Personal genetic information and health information in law nº 12/2005," 2005. 2, 20, 27, 43

[15] C. Wickramage, T. Sahama, and C. Fidge, "Anatomy of log files: Implications for information accountability measures," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 1–6, Sept 2016. 2, 21

[16] M. Luethi and G. F. Knolmayer, "Security in health information systems: An exploratory comparison of u.s. and swiss hospitals," in *2009 42nd Hawaii International Conference on System Sciences*, pp. 1–10, Jan 2009. 2

[17] S. D. Burdette, T. E. Herchline, and R. Oehler, "Practicing medicine in a technological age: Using smartphones in clinical practice," *Clinical Infectious Diseases*, vol. 47, no. 1, p. 117, 2008. 2, 28

[18] G. S. Ayatollahi H, Bath PA, "Acceptance and adoption of the innovative use of smartphone," 2007. 2, 28

[19] P. F. Brennan, S. Downs, and G. Casper, "Project healthdesign: Rethinking the power and potential of personal health records," *Journal of Biomedical Informatics*, vol. 43, no. 5, Supplement, pp. S3 – S5, 2010. Project HealthDesign. 3

[20] W. Huang, Y. Dong, A. Milanova, and J. Dolby, "Scalable and precise taint analysis for android," in *Proceedings of the 2015 International Symposium on Software Testing and Analysis*, ISSTA 2015, (New York, NY, USA), pp. 106–117, ACM, 2015. 3, 29, 98, 100

[21] Check Point: Software Technologies, "Preinstalled malware targeting mobile users." `http://blog.checkpoint.com/2017/03/10/preinstalled-malware-targeting-mobile-users/`, 2017. Online, accessed March 2017. 3

[22] Google, "Micro-moments: Your guide to winning the shift to mobile." `https://think.storage.googleapis.com/images/micromoments-guide-to-winning-shift-to-mobile-download.pdf`, 2015. Online, accessed March 2017. 3

[23] International Data Corporation (IDC), "Smartphone os market share, 2016 q3." `https://www.idc.com/promo/smartphone-market-share`, 2016. Online, accessed March 2017. 3

[24] A. Ferreira and G. Lenzini, *Can Transparency Enhancing Tools Support Patient's Accessing Electronic Health Records?*, pp. 1121–1132. Cham: Springer International Publishing, 2015. 3

[25] H. A. Maw, H. Xiao, and B. Christianson, "An adaptive access control model for medical data in wireless sensor networks," in *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)*, pp. 303–309, Oct 2013. 3, 4

[26] P. Moura, P. Fazendeiro, P. Marques, and A. Ferreira, "Sotraace - socio-technical risk-adaptable access control model," in *2017 International Carnahan Conference on Security Technology (ICCST)*, pp. 1–6, Oct 2017. 6, 35, 42, 99

[27] International Organization for Standardization, “Iso/iec 12207:2008 systems and software engineering – software life cycle processes.” https://www.iso.org/standard/43447.html, 2008. Online, accessed March 2017. 7

[28] ISTQB Exam Certification, “What are the software development models?.” http://istqbexamcertification.com/what-are-the-software-development-models. Online, accessed March 2017. 7

[29] Security Week : Noa Bar-Yosef , “Code wars: Why web application firewalls are not the enemy of the sdlc.” http://www.securityweek.com/code-wars-why-web-application-firewalls-are-not-enemy-sdlc, 2012. Online, accessed March 2017. 7

[30] Oxford Dictionaries, “Definition of security in english.” https://en.oxforddictionaries.com/definition/security. Online, accessed March 2017. 7

[31] Cambridge Dictionaries, “Meaning of “security” in the english dictionary.” https://dictionary.cambridge.org/dictionary/english/security#translations. Online, accessed March 2017. 7

[32] L. G. Pierson and E. L. Witzke, “A security methodology for computer networks,” *AT T Technical Journal*, vol. 67, pp. 28–36, May 1988. 8

[33] J. KS, “Privacy: What’s different now?,” *Interdisciplinary Science Reviews.*, vol. 28, no. 4, pp. 7:1–7:31, 2003. 8

[34] F. SCHAUER, “Free speech and the social construction of privacy,” *Social Research*, vol. 68, no. 1, pp. 221–232, 2001. 8

[35] International Organization for Standardization, “Iso/iec 27000 family - information security management systems.” https://www.iso.org/isoiec-27001-information-security.html, 2008. Online, accessed March 2017. 8

[36] International Organization for Standardization, “Benefits of standards: the iso materials.” https://www.iso.org/benefits-of-standards-the-iso-materials.html. Online, accessed March 2017. 8

[37] Information Systems Audit and Control Association- ISACA, “Glossary of terms.” http://www.isaca.org/Knowledge-Center/Documents/Glossary/glossary.pdf, 2015. Online, accessed March 2017. 8

[38] NIST : National Institute of Standards and Technology, “Computer security resources center - drivers.” http://csrc.nist.gov/drivers/. Online, accessed March 2017. 8

[39] Federal Information Security Management Act of 2002, “United states public law 107- 347 title iii.” http://csrc.nist.gov/drivers/documents/FISMA-final.pdf, 2002. Online, accessed March 2017. 8, 9

[40] NIST : National Institute of Standards and Technology, “Special publication 800-53 revision 4 , security and privacy controls for federal information systems and organizations.” http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf. Online, accessed March 2017. 8, 21, 54, 74, 82, 99

[41] INFOSEC Instituite, "Cia triad." http://resources.infosecinstitute.com/cia-triad/#gref. Online, accessed March 2017. 9

[42] OWASP - Open Web Application Security Project, "Guide to cryptography." https://www.owasp.org/index.php/Guide_to_Cryptography. Online, accessed March 2017. 9

[43] T. Alharbi, A. Aljuhani, and H. Liu, "Holistic ddos mitigation using nfv," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1–4, Jan 2017. 9

[44] OBBC, "Yahoo attack exposes web weakness." http://news.bbc.co.uk/2/hi/science/nature/635444.stm, 200. Online, accessed March 2017. 9

[45] S. Luo, J. Hu, and Z. Chen, "An identity-based one-time password scheme with anonymous authentication," in *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2, pp. 864–867, April 2009. 10

[46] Frost & Sullivan, Booz, Allen, Hamilton, "The 2015 (isc)2 global information security workforce study." https://www.cybercompex.org/fileSendAction/fcType/0/fcOid/445471828686010375/filePointer/445471828686010530/fodoid/445471828686010527/frostsullivan-ISC2-global-information-security-workforce-2015.pdf, 2015. Online, accessed March 2017. 10

[47] Symantec, *INTERNET SECURITY THREAT REPORT*. Volume 20 , 21347933, 2015. 11

[48] D. C. O. I. Ehinome J. Ikhalia, "The need for two factor authentication in social media," *Proceedings of the International Conference on Future Trends in Computing and Communication - FTCC*, vol. 15, no. 1, 2013. 11

[49] Linkdin, "Account security and privacy - best practices." https://www.linkedin.com/help/linkedin/answer/267/account-security-and-privacy-best-practices?lang=en, 2016. Online, accessed March 2017. 11

[50] OWASP - Open Web Application Security Project, "Welcome to owasp." https://www.owasp.org/index.php/Main_Page. Online, accessed March 2017. 11, 65, 82, 99

[51] OWASP - Open Web Application Security Project, "Industry, citations and use." https://www.owasp.org/index.php/Industry:Citations#National_.26_International_Legislation.2C_Standards.2C_Guidelines.2C_Committees_and_Industry_Codes_of_Practice. Online, accessed March 2017. 11

[52] OWASP - Open Web Application Security Project, "Top 10 2013." https://www.owasp.org/index.php/Top_10_2013-Top_10, 2013. Online, accessed March 2017. 11, 73, 75

[53] Symantec, "A new zero-day vulnerability discovered every week in 2015." https://www.symantec.com/content/dam/symantec/docs/infographics/istr-zero-day-en.pdf, 2015. Online, accessed March 2017. 11

[54] StatCounter, "Number of mobile phone users worldwide from 2013 to 2019 (in billions)." https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/, 2015. Online, accessed March 2017. 11

[55] StatCounter, "Top 9 browsers." http://gs.statcounter.com/#mobile_browser-ww-monthly-201501-201512, 2015. Online, accessed March 2017. 12

[56] G. Iosifidis, L. Gao, J. Huang, and L. Tassiulas, "Efficient and fair collaborative mobile internet access," *IEEE/ACM Transactions on Networking*, vol. PP, no. 99, pp. 1–15, 2017. 12

[57] OWASP - Open Web Application Security Project, "Mobile top 10 2016." https://www.owasp.org/index.php/Mobile_Top_10_2016-Top_10, 2016. Online, accessed March 2017. 13

[58] A. R. Sadeghi, "Security and privacy more crucial than ever," *IEEE Security Privacy*, vol. 15, pp. 3–4, Jan 2017. 14

[59] OWASP - Open Web Application Security Project, "Security by design principles." https://www.owasp.org/index.php/Security_by_Design_Principles, 2016. Online, accessed March 2017. 14

[60] International Organization for Standardization, "So/iec 9798-1:2010 preview information technology – security techniques – entity authentication – part 1: General." https://www.iso.org/standard/53634.html, 2010. Online, accessed May 2017. 14

[61] OWASP - Open Web Application Security Project, "Authentication cheat sheet." https://www.owasp.org/index.php/Authentication_Cheat_Sheet#Introduction. Online, accessed May 2017. 14, 15, 17, 54, 56

[62] V. Goyal, A. Abraham, S. Sanyal, and S. Y. Han, "The n/r one time password system," in *International Conference on Information Technology: Coding and Computing (ITCC'05) - Volume II*, vol. 1, pp. 733–738 Vol. 1, April 2005. 15

[63] GFI, "The top 20 free network monitoring and analysis tools for sys admins." http://www.gfi.com/blog/the-top-20-free-network-monitoring-and-analysis-tools-for-sys-admins, 2015. Online, accessed March 2017. 15

[64] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, pp. 770–772, Nov. 1981. 15

[65] R. . . W. Simpson, "N. haller, bellcore, c. metz, a one-time password system." https://tools.ietf.org/html/rfc1938, 1996. Online, accessed May 2017. 15

[66] R. . . W. Simpson, "Ppp challenge handshake authentication protocol (chap)." https://www.ietf.org/rfc/rfc1994.txt, 1996. Online, accessed May 2017. 15

[67] Cisco, "Point-to-point protocol (ppp), understanding and configuring ppp chap authentication." https://www.cisco.com/c/en/us/support/docs/wan/point-to-point-protocol-ppp/10241-ppp-callin-hostname.html. Online, accessed May 2017. 15

[68] K. Mao, J. Chen, J. Liu, and M. Wang, "Security enhancement on an authentication scheme for privacy preservation in ubiquitous healthcare system," in *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, vol. 01, pp. 885–892, Dec 2015. 15, 16

[69] Y. E. Mtonga, K. and H. Kim, “Authenticated privacy preserving pairing-based scheme for remote health monitoring systems.,” in *Journal of Information Security*, pp. 75–90, 2017. 15

[70] S. Lee, H. Kim, and S. W. Lee, “Security concerns of identity authentication and context privacy preservation in uhealthcare system,” in *2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 107–112, July 2013. 15

[71] H. Kim, “Enhanced identity authentication and context privacy preservation in ubiquitous healthcare system,” in *International Journal of Control and Automation*, pp. 391–400, July 2014. 15, 16

[72] Y. Y. Chang, H. B. Zhong, and M. L. Wang, “Implementation of mobile dicom image retrieval application with qr-code authentication,” in *2014 International Symposium on Computer, Consumer and Control*, pp. 372–375, June 2014. 16

[73] A. B. Augusto and M. E. Correia, *OFELIA – A Secure Mobile Attribute Aggregation Infrastructure for User-Centric Identity Management*, pp. 61–74. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. 16, 17, 25, 56

[74] C. Hillen, “The pseudonym broker privacy pattern in medical data collection,” in *2015 IEEE Trustcom/BigDataSE/ISPA*, vol. 1, pp. 999–1005, Aug 2015. 16, 43

[75] P. M. Schwartz, “Property, privacy, and personal data„” *Commun. ACM*, 2004. 16

[76] V. . Schneider, *Information Systems - Enabling Business in a Digital World*. Norwood, MA, USA: Pearson Custom Books, 3rd ed., 2015. 16

[77] M. C. Mont, S. Pearson, and P. Bramhall, *Towards Accountable Management of Privacy and Identity Information*, pp. 146–161. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. 17

[78] D. Recordon and D. Reed, “Openid 2.0: A platform for user-centric identity management,” in *Proceedings of the Second ACM Workshop on Digital Identity Management*, DIM ’06, (New York, NY, USA), pp. 11–16, ACM, 2006. 17

[79] E. D. Hardt, “The OAuth 2.0 Authorization Framework,” RFC 6749, RFC Editor, OCTOBER 2012. 17, 56

[80] C. M. S. M. J. M. B. Campbell, Ping Identity, “Security Assertion Markup Language (SAML) 2.0 Profile for OAuth 2.0 Client Authentication and Authorization Grants,” RFC 7522, RFC Editor, MAY 2015. 17, 25

[81] I. Indu and P. M. R. Anand, “Identity and access management for cloud web services,” in *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, pp. 406–410, Dec 2015. 17

[82] J. Werner and C. M. Westphall, “A model for identity management with privacy in the cloud,” in *2016 IEEE Symposium on Computers and Communication (ISCC)*, pp. 463–468, June 2016. 17

[83] Organization for the Advancement of Structured Information Standards - OASIS, "Security assertion markup language (saml) v2.0 technical overview." https://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-tech-overview-2.0.html, 2008. Online, accessed May 2017. 17, 25

[84] E. E. Hammer-Lahav, "The OAuth 1.0 Protocol," RFC 5849, RFC Editor, APRIL 2010. 17

[85] B. d. M. E. J. M. B. J. Nat Sakimura, John Bradley, "Openid connect standard 1.0 - draft 07." https://openid.net/specs/openid-connect-standard-1_0-07.html. Online, accessed May 2017. 17

[86] M. S. Ferdous and R. Poet, "Formalising identity management protocols," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pp. 137–146, Dec 2016. 17

[87] M. Hansen, A. Schwartz, and A. Cooper, "Privacy and identity management," *IEEE Security Privacy*, vol. 6, pp. 38–45, March 2008. 17

[88] E. Bertino and K. Takahashi, *Identity Management: Concepts, Technologies, and Systems*. Norwood, MA, USA: Artech House, Inc., 2010. 17

[89] A. Bhargav-Spantzel, J. Camenisch, T. Gross, and D. Sommer, "User centricity: A taxonomy and open issues," *J. Comput. Secur.*, vol. 15, pp. 493–527, Oct. 2007. 17, 19

[90] M. C. Domenech, E. Comunello, and M. S. Wangham, "Identity management in e-health: A case study of web of things application using openid connect," in *2014 IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 219–224, Oct 2014. 19

[91] J. Mirkovic, H. Bryhni, and C. M. Ruland, "Secure solution for mobile access to patient's health care record," in *2011 IEEE 13th International Conference on e-Health Networking, Applications and Services*, pp. 296–303, June 2011. 19

[92] M. Deng, R. Scandariato, D. D. Cock, B. Preneel, and W. Joosen, "Identity in federated electronic healthcare," in *2008 1st IFIP Wireless Days*, pp. 1–5, Nov 2008. 20

[93] M. J. Campos, M. E. Correia, and L. Antunes, "Leveraging identity management interoperability in ehealth," in *2011 Carnahan Conference on Security Technology*, pp. 1–8, Oct 2011. 20

[94] D. Slamanig and C. Stingl, "Privacy aspects of ehealth," in *2008 Third International Conference on Availability, Reliability and Security*, pp. 1226–1233, March 2008. 20

[95] D. F. Ferraiolo, D. R. Kuhn, and R. Chandramouli, *Role-Based Access Control*. Norwood, MA, USA: Artech House, Inc., 2nd ed., 2007. 21

[96] OWASP - Open Web Application Security Project, "Access control cheat sheet." https://www.owasp.org/index.php/Access_Control_Cheat_Sheet. Online, accessed May 2017. 21, 22

[97] H. C. A. Tilborg and S. Jajodia, *Encyclopedia of Cryptography and Security*. Springer Publishing Company, Incorporated, 2nd ed., 2011. 21, 22, 80

[98] V. C. Hu and R. Kuhn, "Access control policy verification," *Computer*, vol. 49, pp. 80–83, Dec 2016. 21

[99] L. Røstad, *Access Control in Healthcare Information Systems*. PhD thesis, Norwegian University of Science and Technology, 2009. 21, 23

[100] C. Santos-Pereira, A. B. Augusto, R. Cruz-Correia, and M. E. Correia, "A secure rbac mobile agent access control model for healthcare institutions," in *Proceedings of the 26th IEEE International Symposium on Computer-Based Medical Systems*, pp. 349–354, June 2013. 21

[101] OWASP - Open Web Application Security Project, "Session management." `https://www.owasp.org/index.php/Session_Management_Cheat_Sheet`. Online, accessed May 2017. 21

[102] B. W. Lampson, "Dynamic protection structures," in *Proceedings of the November 18-20, 1969, Fall Joint Computer Conference*, AFIPS '69 (Fall), (New York, NY, USA), pp. 27–38, ACM, 1969. 21

[103] D. E. Bell and L. J. LaPadula, "Secure computer systems: Mathematical foundations," (Springfield, USA), MITRE Technical Report 2547, Volume I, 1973. 22

[104] R. James, Frank Mabry, Kevin Huggins, Michael Miller, Thomas Cook, Florian Tamang, Sam Abbott-McCune, Howard Taylor and William J. Adams, "Secure computer systems: Extensions to the bell-la padula model," (Springfield, USA), MITRE Technical Report 2547, Volume I, 2009.12.1. 22

[105] L. Giuri, "Role-based access control: A natural approach," in *Proceedings of the First ACM Workshop on Role-based Access Control*, RBAC '95, (New York, NY, USA), ACM, 1996. 22

[106] M. Sicuranza and A. Esposito, "An access control model for easy management of patient privacy in ehr systems," in *8th International Conference for Internet Technology and Secured Transactions (ICITST-2013)*, pp. 463–470, Dec 2013. 22, 23, 26

[107] D. R. K. David F. Ferraiolo, Janet A. Cugini, "Role-based access control (rbac): Features and motivations," *National Institute of Standards and Technology U. S. Department of Commerce Gaithersburg MD 20899*, Jan 1995. 22

[108] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The nist model for role-based access control: Towards a unified standard," in *Proceedings of the Fifth ACM Workshop on Role-based Access Control*, RBAC '00, (New York, NY, USA), pp. 47–63, ACM, 2000. 22

[109] A. Ferreira, D. Chadwick, P. Farinha, R. Correia, G. Zao, R. Chilro, and L. Antunes, "How to securely break into rbac: The btg-rbac model," in *2009 Annual Computer Security Applications Conference*, pp. 23–31, Dec 2009. 22, 24, 25, 37

[110] A. Boonyarattaphan, Y. Bai, S. Chung, and R. Poovendran, "Spatial-temporal access control for e-health services," in *2010 IEEE Fifth International Conference on Networking, Architecture, and Storage*, pp. 269–276, July 2010. 22

[111] J. Calvillo-Arbizu, I. Román-Martínez, and L. M. Roa-Romero, "Standardized access control mechanisms for protecting iso 13606-based electronic health record systems," in *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 539–542, June 2014. 22, 23, 25

[112] I. Technology, "Role based access control. ansi/incits 359-2004.," *American National Standard for Information Technology*, February 2004. 22

[113] P. W. Fong, "Relationship-based access control: Protection model and policy language," in *Proceedings of the First ACM Conference on Data and Application Security and Privacy*, CODASPY '11, (New York, NY, USA), pp. 191–202, ACM, 2011. 23, 24, 37

[114] K. Kawagoe and K. Kasai, *Situation, Team and Role based Access Control*, pp. 629–637. Journal of Computer Science 7 (5) ,Science Publications, 2011. 23, 24

[115] M. F. F. Khan and K. Sakamura, "Toward a synergy among discretionary, role-based and context-aware access control models in healthcare information technology," in *World Congress on Internet Security (WorldCIS-2012)*, pp. 66–70, June 2012. 23

[116] M. F. F. Khan and K. Sakamura, "Context-aware access control for clinical information systems," in *2012 International Conference on Innovations in Information Technology (IIT)*, pp. 123–128, March 2012. 23

[117] F. Hansen and V. Oleshchuk, "Location-based security framework for use of handheld devices in medical information systems," in *Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06)*, pp. 5 pp.–569, March 2006. 23

[118] E. Bertino, P. A. Bonatti, and E. Ferrari, "Trbac: A temporal role-based access control model," *ACM Trans. Inf. Syst. Secur.*, vol. 4, pp. 191–233, Aug. 2001. 24

[119] H. S. G. Pussewalage and V. A. Oleshchuk, "An attribute based access control scheme for secure sharing of electronic health records," in *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pp. 1–6, Sept 2016. 24

[120] D. R. Kuhn, E. J. Coyne, and T. R. Weil, "Adding attributes to role-based access control," *Computer*, vol. 43, pp. 79–81, June 2010. 24

[121] M. Sicuranza and M. Ciampi, "A semantic access control for easy management of the privacy for ehr systems," in *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, pp. 400–405, Nov 2014. 24

[122] B. Carminati, E. Ferrari, and A. Perego, *Rule-Based Access Control for Social Networks*, pp. 1734–1744. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. 24

[123] R. K. Thomas, "Team-based access control (tmac): A primitive for applying role-based access controls in collaborative environments," in *Proceedings of the Second ACM Workshop on Role-based Access Control*, RBAC '97, (New York, NY, USA), pp. 13–19, ACM, 1997. 24

[124] Y. Jin, M. Tomoishi, and S. Matsuura, "Enhancement of vpn authentication using gps information with geo-privacy protection," in *2016 25th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–6, Aug 2016. 24, 56

[125] S. Arunkumar, B. Soyluoglu, M. Sensoy, M. Srivatsa, and M. Rajarajan, "Geospatial access control for mobile devices," in *2015 IEEE Region 10 Symposium*, pp. 86–89, May 2015. 24, 25

[126] Yale University : Health Insurance Portability and Accountability Act, "Break glass procedure: Granting emergency access to critical ephi systems." http://hipaa.yale.edu/security/break-glass-procedure-granting-emergency-access-critical-ephi-systems, 2004. Online, accessed May 2017. 24, 25

[127] McGraw R., "Risk-adaptable access control (radac). privilege (access) management workshop.," *NIST - National Institute of Standards and Technology - Information Technology Laboratory.*, 2009. 25, 35, 38

[128] S. Kandala, R. Sandhu, and V. Bhamidipati, "An attribute based framework for risk-adaptive access control models," in *2011 Sixth International Conference on Availability, Reliability and Security*, pp. 236–241, Aug 2011. 25

[129] Organization for the Advancement of Structured Information Standards - OASIS, "Oasis extensible access control markup language (xacml) tc." https://www.oasis-open.org/committees/xacml/. Online, accessed May 2017. 25

[130] R. Nasim and S. Buchegger, "Xacml-based access control for decentralized online social networks," in *2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing*, pp. 671–676, Dec 2014. 25

[131] N. Kahani, K. Elgazzar, and J. R. Cordy, "Authentication and access control in e-health systems in the cloud," in *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, pp. 13–23, April 2016. 25, 26, 56

[132] M. Li, S. Yu, K. Ren, and W. Lou, *Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-owner Settings*, pp. 89–106. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. 25, 29

[133] M. Barua, X. Liang, R. Lu, and X. Shen, "Espac: Enabling security and patient-centric access control for ehealth in cloud computing," *Int. J. Secur. Netw.*, vol. 6, pp. 67–76, Nov. 2011. 25

[134] M. Barua, M. S. Alam, X. Liang, and X. Shen, "Secure and quality of service assurance scheduling scheme for wban with application to ehealth," in *2011 IEEE Wireless Communications and Networking Conference*, pp. 1102–1106, March 2011. 25

[135] H. C. Ossebaard and L. Van Gemert-Pijnen, "ehealth and quality in health care: implementation time," *International Journal for Quality in Health Care*, vol. 28, no. 3, pp. 415–419, 2016. 26, 28

[136] IIA - Internet Innovation Alliance, "10 benefits of health it." https://internetinnovation.org/special-reports/telemed-infographic/. Online, accessed July 2017. 26

[137] U.S. Department of Health & Human Services , "Privacy, security, and electronic health records." https://www.hhs.gov/sites/default/files/ocr/privacy/hipaa/understanding/consumers/privacy-security-electronic-records.pdf. Online, accessed July 2017. 27

[138] U.S. Chamber Of Commerce , "The impact of broadband on senior citizens." Online, accessed July 2017. 27

[139] European Comissione, "Public health." http://ec.europa.eu/health/ehealth/policy/index_en.htm. Online, accessed July 2017. 27

[140] P. Ray and J. Wimalasiri, "The need for technical solutions for maintaining the privacy of ehr," in *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4686–4689, Aug 2006. 27, 29

[141] HealthIT, "What is a personal health record?." https://www.healthit.gov/providers-professionals/faqs/what-personal-health-record. Online, accessed July 2017. 27

[142] HealthIT Buzz : Leon Rodriguer, Former Director, HHS Office for Civil Rights, "Privacy, security, and electronic health records." https://www.healthit.gov/buzz-blog/privacy-and-security-of-ehrs/privacy-security-electronic-health-records/, 2011. Online, accessed March 2017. 27, 43

[143] J. K. J. F. B. M. R. S. E. O. J. Hsu, Huang, "Use of e-health services between 1999 and 2002: a growing digital divide," *Journal of the American Medical Informatics Association*, vol. 12, no. 2, 2005. 28

[144] N. S. Fleming, E. R. Becker, S. D. Culler, D. Cheng, R. McCorkle, B. d. Graca, and D. J. Ballard, "The impact of electronic health records on workflow and financial measures in primary care practices," *Health Services Research*, vol. 49, no. 1pt2, pp. 405–420, 2014. 28

[145] World Health Organization, "mhealth: New horizons for health through mobile technologies, global observatory for ehealth series - volume 3, isbn 978 92 4 156425 0." http://www.who.int/goe/publications/goe_mhealth_web.pdf, 2011. Online, accessed June 2017. 28

[146] E. Ozdalga, A. Ozdalga, and N. Ahuja, "The smartphone in medicine: A review of current and potential use among physicians and students," *J Med Internet Res*, vol. 14, p. e128, Sep 2012. 28

[147] M. Plachkinova, S. Andrés, and S. Chatterjee, "A taxonomy of mhealth apps – security and privacy concerns," in *2015 48th Hawaii International Conference on System Sciences*, pp. 3187–3196, Jan 2015. 28

[148] B. M. Silva, J. J. Rodrigues, I. de la Torre Díez, M. López-Coronado, and K. Saleem, "Mobile-health: A review of current state in 2015," *Journal of Biomedical Informatics*, vol. 56, pp. 265 – 272, 2015. 29

[149] HealthCare Finance - Steff Deschenes, "5 ways telemedicine is reducing the cost of healthcare." http://www.healthcarefinancenews.com/news/5-ways-telemedicine-reducing-cost-healthcare, 2012. Online, accessed June 2017. 29

[150] e. a. Priyanka Agarwal, G. Caleb Alexander, "Patient adoption of mhealth - use, evidence and remaining barriers to mainstream acceptance," *IMS Institute for Healthcare Informatics*, Sept 2015. 29

[151] U. Varshney, "Mobile health: Four emerging themes of research," *Decision Support Systems*, vol. 66, pp. 20 – 35, 2014. 29

[152] S. K. Das, K. Kant, and N. Zhang, *Handbook on Securing Cyber-Physical Critical Infrastructure*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1st ed., 2012. 29

[153] KPMG, "Health care and cyber security – increasing threats require increasing capabilities." http://www.kpmg-institutes.com/institutes/healthcare-life-sciences-institute/articles/2015/08/health-care-and-cyber-security.html. Online, accessed July 2017. 29

[154] Wedi, "Perspectives on cybersecurity in healthcare june 2015." http://www.wedi.org/docs/test/cyber-security-primer.pdf. Online, accessed July 2017. 30

[155] Kroll Fraud Solutions - Healthcare Information and Management Systems Society (HIMSS), "Analytics report: Security of patient data, usa 2008." Online, accessed July 2017. 30

[156] E. Vaast, "Danger is in the eye of the beholders: Social representations of information systems security in healthcare," *The Journal of Strategic Information Systems*, vol. 16, no. 2, pp. 130 – 152, 2007. Security and Privacy. 30

[157] G. N. Samy, R. Ahmad, and Z. Ismail, "Threats to health information security," in *2009 Fifth International Conference on Information Assurance and Security*, vol. 2, pp. 540–543, Aug 2009. 30

[158] S. M. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K. S. Kwak, "The internet of things for health care: A comprehensive survey," *IEEE Access*, vol. 3, pp. 678–708, 2015. 30, 31, 32

[159] Dimitra Liveri, Anna Sarri, Christina Skouloudi, ENISA, "Security and resilience in ehealth : Security challenges and risks , 2015." https://www.enisa.europa.eu/publications/security-and-resilience-in-ehealth-infrastructures-and-services. Online, accessed August 2017. 30, 31, 32, 65

[160] N. C. Dalkey, "The delphi method: An experimental study of group opinion, santa monica, ca: Rand corporation, 1969." https://www.rand.org/pubs/research_memoranda/RM5888.html. Online, accessed December 2017. 40

[161] Android, "Android web page." https://www.android.com/. Online, accessed March 2017. 44

[162] W3C, "Web service architecture." https://www.w3.org/TR/ws-arch/, 2011. Online, accessed July 2017. 58

[163] W3C, "Simple object access protocol (soap) 1.1." https://www.w3.org/TR/2000/NOTE-SOAP-20000508/, 2000. Online, accessed July 2017. 58

[164] Spring, "Understanding rest." https://spring.io/understanding/REST, 2017. Online, accessed July 2017. 59

[165] NIST : National Institute of Standards and Technology, "Recommendation for key management." http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf, 2016. Online, accessed September 2017. 65, 82, 99, 100

[166] International Organization for Standardization, "Iso 18308:2011." https://www.iso.org/standard/52823.html, 2011. Online, accessed March 2017. 65

[167] H. van der Linden, D. Kalra, A. Hasman, and J. Talmon, "Inter-organizational future proof ehr systems," *International Journal of Medical Informatics*, vol. 78, no. 3, pp. 141 – 160, 2009. 65

[168] OWASP - Open Web Application Security Project, "Application threat modeling." https://www.owasp.org/index.php/Application_Threat_Modeling. Online, accessed August 2017. 67, 68, 70, 71, 100

[169] Microsoft, "The stride threat model." https://msdn.microsoft.com/en-us/library/ee823878(v=cs.20).aspx, 2002. Online, accessed September 2017. 67

[170] Microsoft, "Improving web application security: Threats and countermeasures." https://msdn.microsoft.com/en-us/library/ff648644.aspx, 2003. Online, accessed October 2017. 70, 71

[171] NIST : National Institute of Standards and Technology, "Special publication 800-22- a statistical test suite for random and pseudorandom number generators for cryptographic applications." http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-22r1a.pdf, 2010. Online, accessed September 2017. 72

[172] OWASP - Open Web Application Security Project, "Password storage cheat sheet." https://www.owasp.org/index.php/Password_Storage_Cheat_Sheet. Online, accessed April 2017. 73, 74

[173] F. Wiemer and R. Zimmermann, "High-speed implementation of bcrypt password search using special-purpose hardware," in *2014 International Conference on ReConFigurable Computing and FPGAs (ReConFig14)*, pp. 1–6, Dec 2014. 74

[174] OWASP - Open Web Application Security Project, "Cryptographic storage cheat sheet." https://www.owasp.org/index.php/Cryptographic_Storage_Cheat_Sheet. Online, accessed October 2017. 74, 80

[175] E. R. T. Dierks, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, RFC Editor, August 2008. 74

[176] OWASP - Open Web Application Security Project, "Transport layer protection cheat sheet." https://www.owasp.org/index.php/Transport_Layer_Protection_Cheat_Sheet. Online, accessed April 2017. 75, 76, 100

[177] Mozilla, "Transport layer protection cheat sheet." Security/ServerSideTLS. Online, accessed April 2017. 76, 100

[178] D. A. M. J. Viega, "The galois/counter mode of operation (gcm)." http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.694.695&rep=rep1&type=pdf. Online, accessed October 2017. 80

[179] OWASP - Open Web Application Security Project, "Sql injection prevention cheat sheet." https://www.owasp.org/index.php/SQL_Injection_Prevention_Cheat_Sheet. Online, accessed October 2017. 81

[180] James Smith - LoopJ, "Android asynchronous http client." http://loopj.com/android-async-http/, 2017. Online, accessed August 2017. 83

[181] Oracle Corporation, "Jersey - restful web services in java.." https://jersey.github.io/, 2017. Online, accessed August 2017. 84

[182] Advanced Rest Client, "Advanced rest client." https://chrome.google.com/webstore/detail/advanced-rest-client/hgmloofddffdnphfgcellkdfbfbjeloo. Online, accessed December 2017. 97, 100

[183] OWASP - Open Web Application Security Project, "Testing for weak encryption." https://www.owasp.org/index.php/Testing_for_Weak_Encryption_(OTG-CRYPST-004). Online, accessed December 2017. 97, 100

[184] EU General Data Protection Regulation (GDPR), "Gdpr portal." https://www.eugdpr.org/. Online, accessed January 2017. 99

[185] OpenSSL, "Cryptography and ssl/tls toolkit." https://www.openssl.org/, 1999. Online, accessed September 2017. 100

[186] openEHR, "An open domain-driven platform for developing flexible e-health systems." http://www.openehr.org/home. Online, accessed July 2017. 100

[187] Qualis SSL Labs, "Ssl server test." https://www.ssllabs.com/ssltest/. Online, accessed January 2017. 100