Queensland University of Technology

A SECURE FRAMEWORK AND RELATED PROTOCOLS FOR UBIQUITOUS ACCESS TO ELECTRONIC HEALTH RECORDS USING JAVA SIM CARDS

Reza Hassanzadeh (n7064802)

BEng.

Principal Supervisor: Dr. Tony Sahama

Associate Supervisor: Prof. Colin Fidge

Submitted in fulfilment of the requirements for the degree of

IT60- Master of IT (Research)

Computer Science Discipline

Faculty of Science and Technology

Queensland University of Technology

May 2010

Keywords

Electronic Health Records, Information Security, Data Privacy, Java SIM Cards, Next Generation Networks, Smart Phones, Near Field Communication, Biometric Identification, Public Key Infrastructure.

Abstract

Ubiquitous access to patient medical records is an important aspect of caring for patient safety. Unavailability of sufficient medical information at the point-of-care could possibly lead to a fatality. The U.S. Institute of Medicine has reported that between 44,000 and 98,000 people die each year due to medical errors, such as incorrect medication dosages, due to poor legibility in manual records, or delays in consolidating needed information to discern the proper intervention.

In this research we propose employing emergent technologies such as *Java SIM Cards* (JSC), *Smart Phones* (SP), *Next Generation Networks* (NGN), *Near Field Communications* (NFC), *Public Key Infrastructure* (PKI), and *Biometric Identification* to develop a secure framework and related protocols for ubiquitous access to *Electronic Health Records* (EHR). A partial EHR contained within a JSC can be used at the point-of-care in order to help quick diagnosis of a patient's problems. The full EHR can be accessed from an *Electronic Health Records Centre* (EHRC) when time and network availability permit.

Moreover, this framework and related protocols enable patients to give their explicit consent to a doctor to access their personal medical data, by using their *Smart Phone*, when the doctor needs to see or update the patient's medical information during an examination. Also our proposed solution would give the power to patients to modify the *Access Control List* (ACL) related to their EHRs and view their EHRs through their *Smart Phone*.

Currently, very limited research has been done on using JSCs and similar technologies as a portable repository of EHRs or on the specific security issues that are likely to arise when JSCs are used with ubiquitous access to EHRs. Previous research is concerned with using Medicare cards, a kind of *Smart Card*, as a repository of medical information at the patient point-of-care. However, this imposes some limitations on the patient's emergency medical care, including the inability to detect the patient's location, to call and send information to an emergency room automatically, and to interact with the patient in order to get consent.

The aim of our framework and related protocols is to overcome these limitations by taking advantage of the SIM card and the technologies mentioned above. Briefly, our framework and related protocols will offer the full benefits of accessing an up-to-date, precise, and comprehensive medical history of a patient, whilst its mobility will provide ubiquitous access to medical and patient information everywhere it is needed. The objective of our framework and related protocols is to automate interactions between patients, healthcare providers and insurance organisations, increase patient safety, improve quality of care, and reduce the costs.

Table of Contents

List of Figures

List of Abbreviations

ACK	Acknowledge
ACL	Access Control List
AD	Authorised Device
AD-FID	AD-Fresh ID
ADS	Authorised Device Scanner
AL	Access Level
AMIM	Access to Medical Information Menu
AND	Authorised NFC-enabled Device
APF	Authorised Person Fingerprint
AP	Authorised Person
API	Application programming interface
AU-EHRC	Australia-Electronic Health Records Center
AU-TTP	Australia-Trusted Third Party
BI	Biometric Identification
DAC	Discretionary access control
DF	Doctor Fingerprint
ECB	Emergency Call Button
EHR	Electronic Health Records
EHRC	Electronic Health Records Center
EKG	Electrocardiography
E-Mail	Electronic Mail
ER	Emergency Room
FSM	Finite State Machine
FTP	File Transfer Protocol
GP	General Practitioner
GSM	Global System for Mobile Communications
HIPAA	Health Insurance Portability and Accountability Act
HTTP	Hypertext Transfer Protocol
IA	Identification Algorithm
Info	Information
IOM	Institute of Medicine
IP	Internet Protocol

IT	Information Technology
IY	Introduce Yourself
JCRE	Java Card Runtime Environment
JCVM	Java Card Virtual Machine
JSC	Java SIM Card
LDAP	Lightweight Directory Access Protocol
MAC	Mandatory access control
MICT	Mobile Information Communication Technologies
Mng	Management
MSC	Message Sequence Chart
MVCE	Modify View Consent Exit
N-EHRC	National-Electronic Health Records Center
NFC	Near Field Communication
NGN	Next Generation Network
N-TTP	National-Trusted Third Party
OS/HW	Operating System / Hardware
OTA	Over The Air
PC	Personal Computer
PD	Patient Device
PDA	Personal Digital Assistant
PF	Patient Fingerprint
PIN	Personal Identification Number
PKI	Public Key Infrastructure
POC	Point-of-Care
PSTN	Public Switched Telephone Network
QLD-EHRC	Queensland- Electronic Health Records Center
QLD-TTP	Queensland-Trusted Third Party
RBAC	Role-Based Access Control
RFID	Radio Frequency Identification
RFM	Remote File Management
RHIOs	Regional Health Information Organisations
RL	Referral Letter
RSA	Rivest, Shamir and Adleman
SDL	Specification and Description Language
S-EHRC	State-Electronic Health Records Center

SIM	Subscriber Identity Module
SIM ID	Subscriber Identity Module Identification
SM	Setup Message
SMS	Short Message Service
SP	Smart Phone
SP-FID	SP-Fresh ID
SPS	SP Scanner
SSL/TLS	Secure Sockets Layer/Transport Layer Security
STK	SIM Tool Kit
S-TTP	State-Trusted Third Party
TCP/IP	Transmission Control Protocol/ Internet Protocol
TTP	Trusted Third Party
TTP-FID	TTP-Fresh ID
UA	Ubiquitous Access
UK	United Kingdom
URL	Uniform Resource Locator
USA	United States of America
US	United States

Statement of Original Authorship

The work contained in this thesis has not been previously submitted to meet requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made.

Reza Hassanzadeh

Acknowledgments

The writing of this thesis has been one of the most significant academic challenges I have ever had to face. Without the support, patience, and guidance of the following people, this study would not have been completed. It is to them that I owe my deepest gratitude:

My wife Asieh and my daughter Negin, for their love, long-standing support, and constant patience during my study. They spent many hours in a week without me just to allow me to focus. I am so sorry for the time I was away.

Professor Colin Fidge and Dr. Tony Sahama who undertook to act as my supervisors despite their many other academic and professional responsibilities. Their wisdom, knowledge, attitude, and commitment to the highest standards inspired and motivated me.

Professor Acram Taji, Dr. Ricky Tunny and Ms. Agatha Nucifora for offering me a lot of kindly help.

Also I would like to thank Queensland University of technology (QUT) for the financial support (QUT Masters Scholarship) for the development of this research.

This Thesis is dedicated to Asieh and Negin

Chapter 1: Introduction

This thesis concerns designing a secure framework and related protocols to enable an *Authorised Person* (AP) such as a doctor to have Ubiquitous Access (UA) to a patient's medical record on a national scale by using new technologies such as *Java SIM Cards*.

1.1 RESEARCH PROBLEM

Ubiquitous access to a patient's medical records is an important aspect of caring for patient safety. Unavailability of sufficient medical information at the patient point-of-care could possibly lead to a fatality. The US Institute of Medicine (IOM) has reported that between 44,000 to 98,000 people die each year due to medical errors, such as incorrect medication dosages, due to poor legibility in manual records, or delays in consolidating needed medical information to discern the proper intervention (Institute of Medicine, 2000). Also the IOM suggests that 90 percent of medical errors are the result of failed systems and procedures that are poorly designed to accommodate the complexity of health care delivery. If properly designed, these systems and procedures could better prevent inevitable human errors from reaching patients. Most of these medical errors could be avoided with ubiquitous access to patient's medical information at the point of care (Abraham et al., 2008). Ubiquitous access enables Healthcare systems to have access to patient's medical information wherever and whenever it is needed electronically. It has the potential to revolutionise next generation medical applications, based on the *Electronic Health Records* concept. It promises to significantly improve the quality of healthcare services to increase patient safety and reduce medical errors and costs.

1.2 PROBLEM STATEMENT

Ubiquitous access to *Electronic Health Records* by using *Java SIM Cards* is the specific problem which we consider in this research. While progress is being made in the technologies associated with access to patient's medical information

at the point of care, issues related to the development of a secure framework, communication protocols and portable repositories for *Electronic Health Records* using newly-emerging technologies such as *Java SIM Cards* still need to be investigated. A *Java SIM Card* is a GSM compliant SIM card with extra functionality which fits into a *Smart Phone* in place of an existing SIM Card.

1.3 CONTRIBUTION TO THE BODY OF KNOWLEDGE

At this time, very limited research has been done on using *Java SIM Cards* and similar technologies as a portable repository of EHRs or on the specific data security issues that are likely to arise when JSCs are used with ubiquitous access to EHRs. Previous research is concerned with using the *Medicare Card*, which is a kind of *Smart Card*, as a repository of medical information at the patient point-of-care (Bishop et al., 2000; Chan et al., 2001). However, this imposes some limitations on the patient's emergency medical care, including the inability to detect and inform healthcare authorities of the patient's location, to call and send information to an emergency room automatically, and to automate and secure interaction with the patient in order to get consent to access personal details.

In this research we propose employing emergent technologies such as *Java SIM Cards* (JSC), *Smart Phones* (SP), *Next Generation Networks* (NGN), *Near Field Communications* (NFC), *Public Key Infrastructure* (PKI), and *Biometric Identification* to create a secure framework and related communications protocols for ubiquitous access to *Electronic Health Records*. A partial EHR contained within a JSC can be used at the patient point-of-care in order to help quick diagnosis of a patient's problems. The full EHR can be accessed from an *Electronic Health Records Centre* (EHRC) when time and network accessibility permit.

1.4 RESEARCH AIM AND OBJECTIVE

The aim of our framework and related protocols is to overcome existing limitations in electronic healthcare by taking advantage of the new commonly-used *Java SIM Card* and the new technologies mentioned above. Briefly, our framework and related protocols will offer the full benefits of accessing an up-to-date, precise, and

comprehensive medical history of a patient, whilst its mobility will provide ubiquitous access to medical and patient information everywhere it is needed. The objective of our framework is to automate interactions between patients, healthcare providers and insurance organisations, increase patient safety, improve quality of care, and reduce the costs.

From a technical perspective our research aims to show how new technologies such as *Java SIM Cards* (JSC), *Smart Phones* (SP), *Next Generation Networks* (NGN), *Near Field Communications* (NFC), *Public Key Infrastructure* (PKI), and *Biometric Identification* can be combined to create a secure framework and related communications protocols for ubiquitous access to *Electronic Health Records*.

1.5 SCOPE AND LIMITATIONS

The scope of this study is to focus on proposing a secure framework and related protocols to have ubiquitous access to *Electronic Health Records* by using *Java SIM Cards*. Our communication protocols occur between patients' *Smart Phones, Authorised Persons'* electronic devices, and a central management centre. We use simulations to demonstrate the feasibility our approach. Other solutions for ubiquitous access to EHRs than *Java SIM Cards* are possible, such as accessing data through devices such as desktop or laptop computers, but are not explored in this research.

Our framework's components are split into two categories; the first one combines components which already exist, and the second includes new ones which need to be developed. Although our work clearly illustrates the requirements for such an architecture, the development of new components and precise details of the data structures needed in the communication protocols are beyond the scope of this preliminary research project.

1.6 OUTLINE OF THE THESIS

This thesis comprises six chapters. Following this introductory chapter is the literature review, Chapter 2, which provides a summary of related work

in the field of *Electronic Health Records*, *Ubiquitous Healthcare*, *Electronic Healthcare Information Security*, and *Electronic Healthcare Wireless Communication*.

Chapter 3, the proposed framework, gives details of our communications architecture which employs emergent and existing technologies such as *Java SIM Cards* (JSC), *Smart Phones* (SP), *Next Generation Networks* (NGN), *Near Field Communications* (NFC), *Public Key Infrastructure* (PKI), and *Biometric Identification* to develop a secure framework for ubiquitous access to *Electronic Health Records* (EHR).

Chapter 4, the proposed communication protocols, specifies in detail three different protocols needed for communication between the *Smart Phone* (SP), *Authorised Devices* (AD), and the *Trusted Third Party* (TTP), using the Specification and Description Language (SDL).

Chapter 5, the protocols simulation, validates the dynamic behaviour of the entire system using the automatic simulation tool UPPAAL. It shows how our protocols work together correctly in different healthcare scenarios.

Finally, Chapter 6, the conclusion, summarises the research findings and proposes future work.

1.7 SUMMARY

In this chapter, we identified the research problem and summarised our contribution to the body of knowledge in this area. The problem definition will be used together with evidence from the literature review to build proof for the existence of a research gap and justification for undertaking the current research. Also in this chapter we clarified the limitations and scope of the research and outlined the whole of the thesis.

Chapter 2: Literature Review

In Chapter 1, we introduced the research challenge of providing ubiquitous access to patient's medical information by using a *Java SIM Card*. We noted that ubiquitous access to *Electronic Health Records* (EHRs) is an increasingly important aspect of caring for patient safety.

In this chapter we review the literature related to the research problem of ubiquitous access to EHRs at the patient point-of-care. We divide this chapter into four sections: *Electronic Health Records*, *Ubiquitous Healthcare*, *Electronic Healthcare Information Security*, and *Electronic Healthcare and Wireless Communication*. As we will explain in Chapter 3, our framework relies on the state-of-the-art in each of these four areas.

Electronic Health Records (EHRs) refers to an individual patient's medical record in digital format. Digitised health information systems are expected to improve the efficiency and quality of patient care and, ultimately, reduce costs (Whatis, 2009). *Ubiquitous Healthcare* aims to provide an environment in which patients can receive medical treatment regardless of their location and the time. As the quality of life has been improved, we are focusing more on our health and people expect to be treated with quickly and conveniently. *Electronic Healthcare Information Security* focuses on the security and privacy issues related to patient healthcare. It aims to protect each patient's medical information from unauthorised accesses and breaches of patient's privacy. *Electronic Healthcare and Wireless Communication* concentrates on how existing wireless technologies can be employed to provide a ubiquitous environment for better quality of care.

2.1 ELECTRONIC HEALTH RECORD

There is no consensus on the exact definition of an *Electronic Health Record* (EHR); however the ISO/TS 18308 standard does give a definition of the primary purpose of an EHR. This is to provide a documented record of care which supports present and future care by the same or other clinicians. Such documentation provides a means of communication among clinicians contributing to the patient's care (ISO/TS 18308, 2004). A formal definition of the scope and purpose of the Integrated Care EHR has more recently been published as ISO TR 20514. The scope and purpose of an Integrated Care EHR is a repository of information regarding the health status of a subject of care in computer processable form, stored and transmitted securely, and accessible by multiple authorised users (ISO/TR 20514, 2005). It has a standardised or commonly agreed logical information model which is independent of EHR systems. Its primary purpose is the support of continuing, efficient and quality integrated health care and it contains information which is retrospective, concurrent and prospective (Van Der Linden et al., 2009).

Electronic Healthcare Records (EHRs, also called Electronic Health Records), which have been a key research field in medical informatics for many years, can be defined as "digitally stored health care information about an individual's lifetime with the purpose of supporting continuity of care, education and research, and ensuring confidentiality at all times" (Iakovidis, 1998). An EHR includes information such as observations, laboratory tests, diagnostic imaging reports, treatments, therapies, drugs administered, patient identifying information, legal permissions, and allergies. Currently, this information is stored in all kinds of proprietary formats in a multitude of medical information systems available on the market. Typical formats include relational database tables, structured document-based storage in various formats, and unstructured document storage such as digitised hardcopies maintained in a classic document management system. This results in a severe interoperability problem in the healthcare informatics domain (Eichelberg et al., 2005).

In the other words, an EHR refers to an individual patient's medical record in digital format. Digitised health information systems are expected to improve efficiency and quality of care and, ultimately, reduce costs. Internationally, a number of groups are working to overcome the challenges of implementing shared EHRs. Such groups include HL7, openEHR, and CEN.

Among other types of data, an EHR typically includes (Whatis, 2009):

- Contact information.
- Information about visits to health care professionals.
- Allergies.
- Insurance information.
- Family history.
- Immunisation status.
- Information about any conditions or diseases.
- A list of medications.
- Records of hospitalisation.
- Information about any surgeries or procedures performed.

The benefits of using EHRs include:

- The ability to automatically share and update information among different offices and organisations.
- More efficient storage and retrieval.
- The ability to share multimedia information, such as medical imaging results, among locations.
- The ability to link records to sources of relevant and current research.
- Easier standardisation of services and patient care.
- Provision of *Decision Support Systems* (DSS) for healthcare professionals.
- Less redundancy of effort.
- Lower cost to the medical system once implementation is complete.

Interoperability between different EHR standards is one of the most important challenges in e-health systems. It is necessary in order to support better communication and coherence between health care parties.

Healthcare organisations today are still struggling with the integration challenges of heterogeneous information systems. These system interfaces are not always compatible 'out of the box', and when the number of involved systems increases, complexity increases in an unmanageable way. Decentralised proprietary format data largely restricts the value of healthcare information. To overcome this, Chenhui et al. (2008) present a healthcare information system integration approach based on an integration engine to address these integration problems, as shown Figure 1. All the involved systems are connected to the integration engine through different interface adaptors. The integration engine takes the inbound message as the trigger event, and generates outbound messages by workflow drivers according to the workflow configuration file. The integration engine is also responsible for understanding the message context and converting the healthcare information into a unified format, and then storing it in the clinical data repository. Thus, it can facilitate the utilisation and sharing of healthcare information (Chenhui et al., 2008). Our framework needs to use this idea in the central management centre (*Trusted Third Party*) to resolve the integration issues.

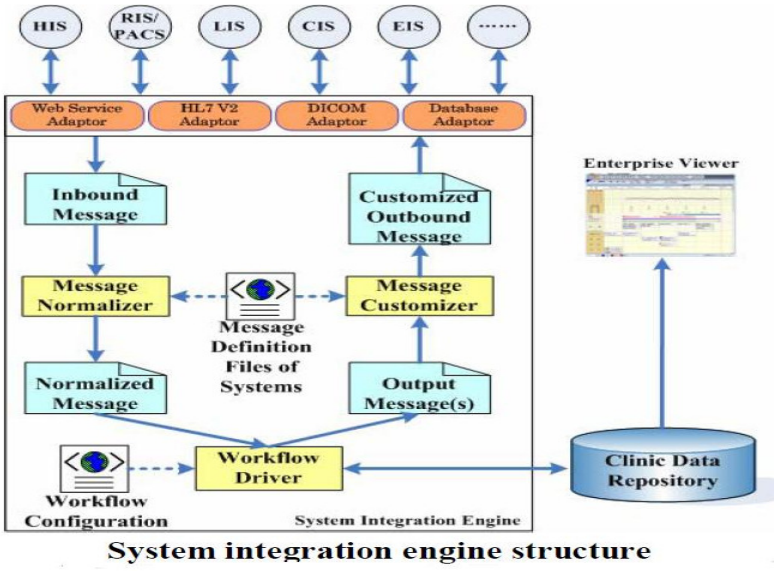


Figure 1. A Possible System Integration Engine Structure[1]

2.2 UBIQUITOUS HEALTHCARE

Ubiquitous Healthcare means an environment in which patients can receive medical treatment regardless of the location and time. As our quality of life has

[1] (Chenhui et al., 2008)

improved, we are focusing more on our health and people want to be treated efficiently and effectively. To support this, interest in remote-treatment has been increasing. So, systems are being developed that can check patient's health status and treat them at a distance from a healthcare facility in real time. Now, we are asking for more services that can detect a patient's location and utilise this information (Ahn et al., 2008).

Ubiquitous Access to patient's medical information has been investigated by many researchers such as (Abraham et al., 2008), (Chan, 2003), (McLaughlin, 2007), (Bishop et al., 2000), (Chan et al., 2001) and others. Finding an efficient solution that can be secure and implemented in existing Medicare systems is a challenging issue. Some researchers have proposed upgrading the existing Medicare card to be a smart card, as a potential solution to access patient's health records anywhere and anytime.

Abraham et al. (2008) examined the use of *Ubiquitous Access* (UA) to medical and patient information via *Mobile Information Communication Technologies* (MICTs) by hospital nurses, because they are at the front lines of care and safety. According to McLaughlin (2007), in the Mount Sinai Medical Centre an encrypted smart card with 64Kb of memory holds not only the patient's name, photo, and insurance information, but also a medical history snapshot, including notes on allergies, medications, recent treatment data, and even in some cases, a compressed EKG test result. Chan et al. (2001) consider a smart card as a potential solution for accessing a patient's health records in all places. The patient's medical history contained within the card can be used to provide quick care to patient problems and allow a patient to have more control over their own medical records. Bishop et al. (2000) combined the network and smart card together for more effectiveness. In this case, the patient's card can be deemed as a medical data storage device which acts as a key to look up medical information on a network and can be used as an identifier for insurance purposes.

However using *Smart Card* imposes some limitations on the patient's emergency medical care and patient's privacy, including the inability to detect and inform healthcare authorities of the patient's location, to call and send information to an emergency room automatically, and to automate and secure interaction with the patient in order to get consent to access personal details.

2.3 ELECTRONIC HEALTH INFORMATION SECURITY

In 1996, the United States' Health Insurance Portability and Accountability Act (HIPAA) offered some general guidelines to enforce the protection of private medical information (US Department of Health & Human Services, 1996). One such guideline stated that patients must be able to view and obtain copies of their records to understand and monitor their health status and the process of diagnosis and therapy. In the real world, patients' health records are distributed around different hospitals and clinics, and the retrieval of this scattered information when a patient visits a doctor in any particular hospital is a major problem. Currently, there are two ways to overcome this problem: either the patient can carry his/her own records on a smart card, or the records can be transmitted through an electronic network (Huang et al., 2009).

Smart cards are often proposed as a security solution for e-health system (Boswell, 2009). They provide a portable, flexible computing platform that is taken to be intrinsically secure. They solve the problems of widely distributing complex cryptographic capabilities to vast numbers of individuals, and of secure key storage to use with that cryptographic capability. Smart cards are often seen as secure because they have limited interfaces. Unlike a *Personal Computer* users can not directly communicate with a smart card. It needs the interface device and an understanding of protocols, file structures, and APIs. They have no apparent peripherals or other removable parts that might be easily attacked, and their interface can be tightly constrained by the developer to limit the scope for an attacker to interact with the card. However, there are security concerns over things such as (Boswell, 2009):

- Protocol errors (e.g. allowing man-in-the-middle attacks; allowing replay attacks; or not protecting the integrity of parameters in critical messages).
- Abuse of the interface to provide unintended functions (e.g. low-level access to a confidential data file; transmitting an unencrypted PIN value for verification over a contactless interface, or returning old data in a communications buffer).
- Internal errors such as buffer overflows.
- Failures in implementation of logic (e.g. conflicts in access control rules; or incorrect state machine transitions).

- Side-channel attacks.

Public key cryptography is the most common technique for securing data exchange over the public Internet (Kavadias et al., 2003). The majority of web sites through which sensitive information is exchanged have adopted the establishment of SSL/TLS (*Secure Sockets Layer/Transport Layer Security*) sessions between communication endpoints for ensuring data privacy and integrity as well as the authenticity of the parties involved. One of the standard procedures that takes place for the establishment of a secure session is the initial exchange of certificates between the two entities. The certificate, containing the public key as well as a set of parameters identifying its owner, is sent and used accordingly to declare to the other end of the communication path – during the protocol's negotiation phase – the presence of a certain physical entity such as a person or organisation wishing to use or provide a service.

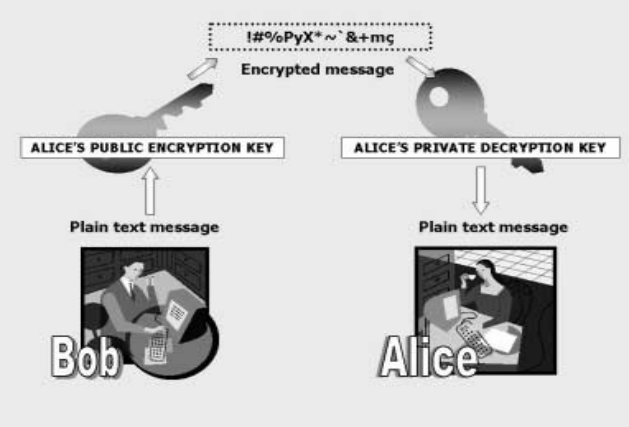


Figure 2. Public Key Infrastructure (PKI)[2]

The SSL/TLS protocols are a set of rules supporting server authentication, client authentication, and encrypted communication between servers and clients (Kavadias et al., 2003). These protocols run above TCP/IP and below higher-level protocols such as the *HyperText Transport Protocol* (HTTP), the *Lightweight Directory Access Protocol* (LDAP), or the *File Transfer Protocol* (FTP). These protocols require at least the certificate of the server. During the initialisation of the session the server sends its certificate to the client in order to authenticate its identity.

[2] Source: http://www.trueb.ch

The authentication process uses *Public-Key Encryption* and *Digital Signatures* to confirm the authenticity of the server. Once the server has been authenticated, the client and server use *Symmetric-Key Encryption* to encrypt all the information they exchange for the remainder of the session and to detect any tampering that may have occurred. SSL and TLS includes three sub-protocols, the handshake, alert and cipherspec, all encapsulated in the record protocol that defines the format used to transmit data. The handshake protocol involves the exchange of a series of messages between the two parties when they first establish a secure connection. This exchange of messages is designed to facilitate actions including authentication of the server to the client, selection of the cryptographic algorithms, optional authentication of the client to the server, generation of shared secrets and establishment of an encrypted connection.

By means of the SSL/TLS protocols which ensure the confidentiality of transmitted data, the highly sensitive information such as *Electronic Health Records* can be transmitted via unsecured network (the Internet). Our framework uses the PKI, and SSL/TLS to establish a secure session between the entities.

An access control model describes at a high level of abstraction a mechanism for governing access to shared resources (Kim et al., 2006). An access control model is an abstraction of an access control mechanism which enforces access control policies specifying who can access what information under what circumstances. There are many access control models which can be categorised into *Discretionary Access Control* (DAC), *Mandatory Access Control* (MAC) and *Role-Based Access Control* (RBAC). DAC models enforce access control based on user identities, object ownership and permission delegation. The owner of an object may delegate the permission of the object to another user. MAC models govern access based on the sensitivity level of subjects and objects. A subject may read an object if the security level of the subject is higher than that of the object. RBAC models enforce access control based on roles. Accessibility is determined by the permissions and users assigned to roles.

Several solutions are available to overcome the security concerns associated with *Electronic Health Records* systems such as cryptographic technology, through the use of *Public Key Infrastructure* (Alhaqbani et al., 2008). However, cryptography merely handles the security of data transmission and does not address the issue of what kind of data is transmitted, or solves the problem of who has access to the data

at the sending and receiving ends. To do this we need to consider Access Control mechanisms such as MAC, DAC, and RBAC that limit who can see Electronic Health Records and how they can manipulate them.

2.4 ELECTRONIC HEALTHCARE WIRELESS COMMUNICATION

Several groups are researching mobile systems for providing access to medical data, mostly in terms of *Telemedicine* and remote patient *Telemonitoring* (Andrade et al., 2003) (Hall et al., 2003) (Hung et al., 2003). In such a mobile architecture, the sensitive and private nature of the medical information renders security in communications among healthcare providers and patients a critical component. However, although a lot of work exists regarding security (in terms of confidentiality, integrity, non-repudiation of receipt/origin, availability) for Intranets (i.e. Hospital Information Systems) or the Internet (i.e. Web Based Telemedicine and Health Record) applications, very little has been done in the field of mobile e-health applications set-up and implementation. The security issues are, in most of these cases, disregarded or not sufficiently handled (Kambourakis et al., 2005).

Mobile communications and wireless public key methods which are easy to use and allow generic secure hybrid communications services, will make an excellent basis for the development of secure electronic communications in a healthcare system. By using a mobile phone as a secure, reliable and cost-effective communications medium, a secure mobile communications service in healthcare gives us the following benefits (Jelekäinen, 2004):

- Identification of both the provider of the healthcare service and the patient;
- The integrity and authenticity of the data transmitted/used;
- Data confidentiality for both parties;
- Non-repudiation of the transaction; and
- Auditability of transaction.

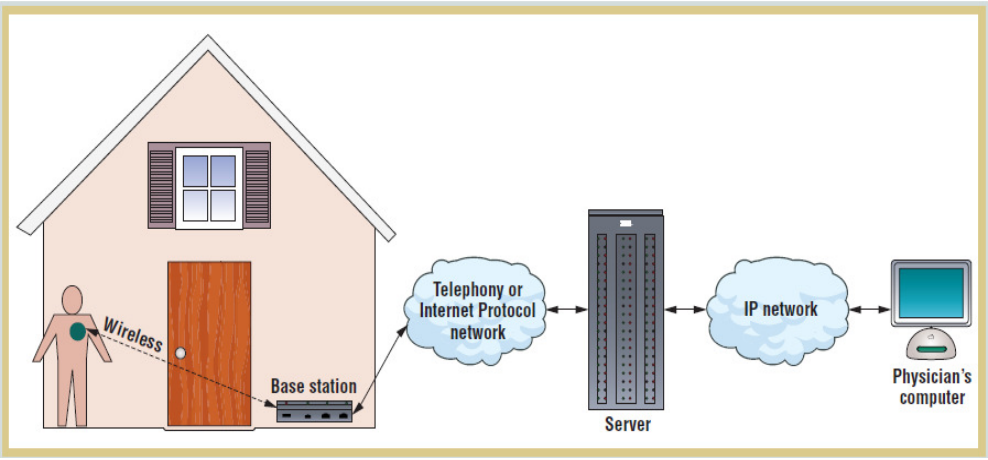


Figure 3. Recent implantable cardiac defibrillators provide home monitoring via wireless base stations that relay data to doctors[3]

There has been for several years discussion of ubiquitous technology; smart, interactive devices are proliferating; and there has been significant speculation about what this means in many domains, including medicine (Katz et al., 2009). Many devices are being developed and tested to provide patients greater access to and control over health information, and to reduce error through better information flow (such as web-based prescription follow-up messaging). Much of the literature on personal and ubiquitous healthcare technology can be considered "technology-driven" because it involves the proposal and evaluation of systems that can address health care needs (Katz et al., 2009).

With respect to application of mobile phones to healthcare, previous researchers (Koskinen et al., 2007) developed a special-purpose mobile phone which can be customised to collect healthcare data depending on user requirements and physician guidance.

Researchers in the UK (Katz et al., 2009) have experimented in obesity treatments using a mobile phone that shared activity information among groups of friends. They found that awareness encouraged reflection on, and increased motivation for, daily activity. However, they also uncovered problems with network reliability related to such applications. A US-based study looked at having users self-monitor caloric balance in real time using a mobile phone. This was done as part of

[3] (Halperin et al., 2008)

an attempt to modify user behavior to reduce obesity (Tsai et al., 2007). The researchers conducted a one-month feasibility study to measure compliance and satisfaction among a sample of 15 participants randomised to one of three groups. They concluded that a mobile phone was as good as or even superior to a paper-based system. Their preliminary results suggest that the mobile phone could be helpful for providing ubiquitous healthcare (Katz et al., 2009).

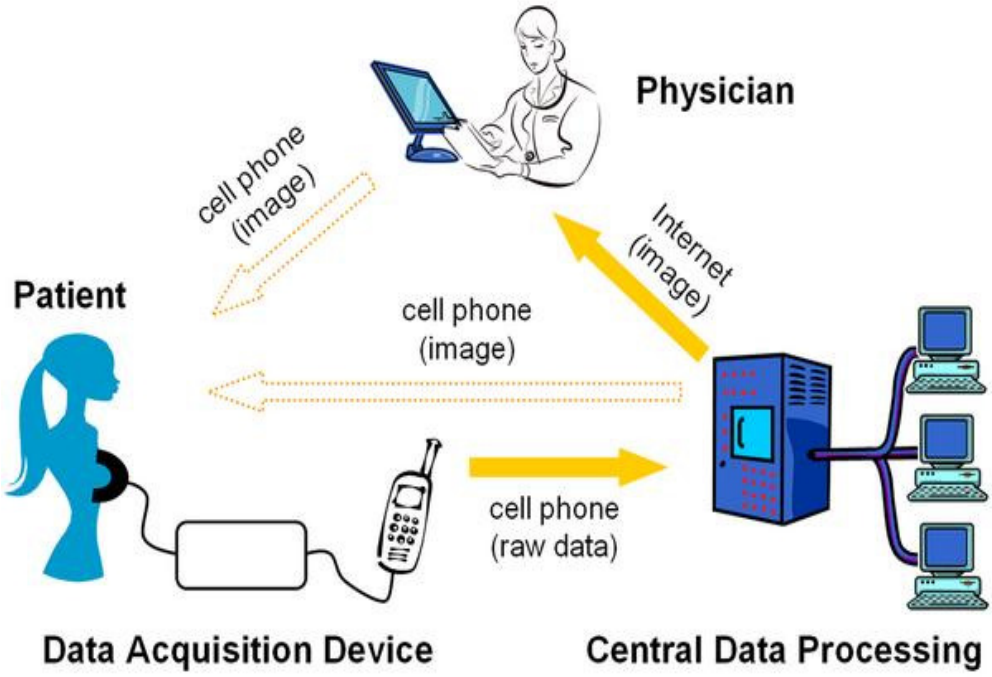


Figure 4. Patient Monitoring[4]

In relation to contactless communication *Near Field Communication* (NFC) is one of the newest wireless networking technologies which can provide intuitive, simple, and safe communication between electronic devices (Grassie, 2007). The NFC-enabled devices allow patient's critical health information contained within patient's device to be seen via an authorised device when the network is not accessible.

NFC is used for the short-range exchange of information via a mobile device (Grassie, 2007). Thanks to its extremely easy handling for users it is opening up new business potential for financial institutions, mobile phone providers, service providers and manufacturers in all sectors. NFC is economically attractive because it is

[4] Source: http://mobilebehavior.com

based on open standards and therefore incurs no license fees, and because it is compatible with all other contactless standards, including Bluetooth. Among NFC's other advantages is the fact that the connection is initialised in less than a second. This compares favourably with *Bluetooth* where users have to wait several seconds before they can transmit data. Another plus is that mobile owners don't have to configure anything for connectivity via NFC. And most important of all, NFC-compatible mobile devices can send data as well as receive it, so that an NFC mobile phone can also act as a reader. The initial markets of interest to NFC service providers are above all the US and Asia, where contactless technologies and their associated devices are already well established (Grassie, 2007).

2.5 SUMMARY

This literature review has surveyed previous work relevant to our research in four categories, *Electronic Health Records*, *Ubiquitous Healthcare*, *Electronic Healthcare Information Security*, and *Electronic Healthcare Wireless Communication*. Our research aims to build on these achievements.

In Section 2.1, we introduced the requirements and content of EHRs. As defined by the literature, an EHR refers to a patient's medical record in a digital format. Digitised health information systems are expected to improve efficiency and quality of care and, ultimately, reduce costs. Moreover, in this section we reviewed the integration challenges of heterogeneous medical records. For the sake of interoperability, we will be considering these challenges in our framework which will be explained in the following chapter.

In Section 2.2, the literature defined *Ubiquitous Healthcare* as an environment in which patients can receive the medical treatment regardless of the location and time. In this area, *Ubiquitous Access* to a patient's medical information has been investigated by many researchers in order to finding an efficient solution that can be secure and implemented in existing Medicare systems. Some researchers have proposed using a *Smart Card*, as a potential solution accessing patient's health records anywhere and anytime, but their limitations include the inability to detect and inform a patient's location, to call and send patient information to an emergency room

automatically, and to computerise and secure interaction with the patient. Overcoming these limitations is our main contribution to the body of knowledge.

In Section 2.3 we mainly focused on the security challenges and solutions such as SSL/TLS and PKI which are currently available or in development for ubiquitous Healthcare.

Finally in Section 2.4 we explored existing wireless communication technologies such as mobile and *Near Field Communication* in order to establish a ubiquitous environment for accessing a patient's medical information.

Chapter 3: A Framework for Ubiquitous Access to Electronic Healthcare Records

In Chapter 2 we reviewed the literature related to ubiquitous access to *Electronic Health Records* (EHRs). This literature review identified that at this time, very limited work has been done on using *Java SIM Cards* and similar technologies as a portable database for EHRs or on the specific security issues that may arise when JSCs are used for this purpose. Also the review showed that how other researchers employed the emergent and existing technologies in e-health area to have secure access to a patient's medical information. Moreover, in Chapter 2 we found out about the limitations which other researchers' works have imposed on the potential for ubiquitous access to EHRs.

In this chapter we give details of our proposed framework which employs emergent and existing technologies such as *Java SIM Cards* (JSC), *Smart Phones* (SP), *Next Generation Networks* (NGN), *Near Field Communications* (NFC), *Public Key Infrastructure* (PKI), and *Biometric Identification* to develop a secure framework for ubiquitous access to *Electronic Health Records* (EHR). A partial EHR contained within a JSC can be used at the patient point-of-care in order to help quick diagnosis of a patient's problems. The full EHR can be accessed from an *Electronic Health Records Centre* (EHRC) when network access is available.

The objective of our framework is to automate interactions between patients, *Authorised Persons*, and a central management centre to access medical information ubiquitously. This framework aims to increase patient safety, improve quality of care, and reduce the costs.

3.1 FRAMEWORK OVERVIEW

Our framework is based on developing a secure conceptual structure to enable an *Authorised Person* (AP) such as a doctor to have Ubiquitous Access (UA) to a patient's medical record on a national scale. As shown in Figure 5, the proposed framework relies on new technologies such as *Java SIM Cards* (JSC), *Smart Phones* (SP), *Near Field Communications* (NFC), *Next Generation Networks* (NGN), *Public Key Infrastructure* (PKI), *Secure Sockets Layer/Transport Layer Security* (SSL/TLS) and *Biometric Identification* (BI).

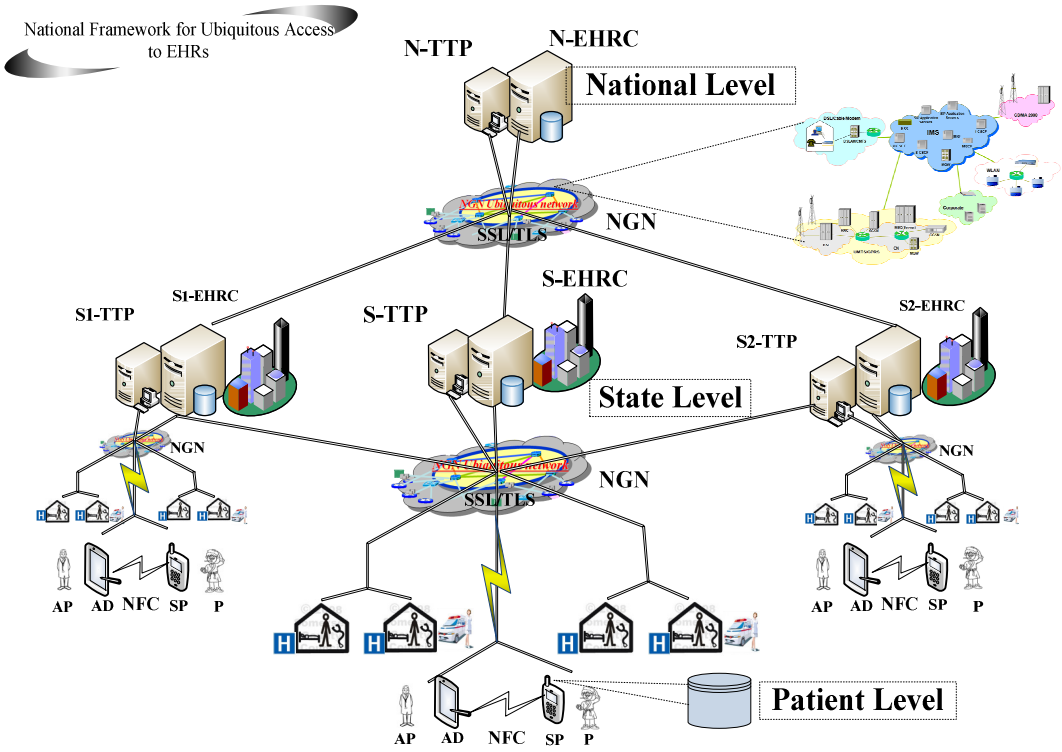


Figure 5. National Framework for Managing Ubiquitous Electronic Health Records

This framework includes six major parts: the *Patient* (P), a *Smart Phone* (SP), an *Authorised Person* (AP), an *Authorised Device* (AD), a *Trusted Third Party* (TTP), and an *Electronic Health Record Centre* (EHRC). The TTP and the EHRC operate at the national (N-TTP, N-EHRC) and state (S-TTP, S-EHRC) levels. The SP is utilised to computerise interaction with a patient. The AD, which is a kind of a *Smart Phone* or *Personal Computer* (PC), is used by an AP such as a doctor to communicate with the SP or TTP. The N-EHRC is a central database containing all pa-

tients' EHRs on a national scale. The N-TTP is employed to manage the whole framework's and S-TTPs' activities. The S-EHRC is a central database containing all patients' EHRs on a state scale. The S-TTP is employed to manage all the SPs and ADs within the particular state.

Mobile phones, in general, fall into three broad categories: basic phones, multimedia phones, and *Smart Phones*. A *Smart Phone* is a handheld device which has both mobile phone and PC-like abilities together. *Smart Phones* have become an emerging phenomenon for personal and business voice, data, e-mail, and Internet access, and could now form the basis of a healthcare network. Power-efficient processors, modern operating systems, broadband Internet access, ease-of-use, convenience, security, reliability, and productivity-enhancing applications will propel the popularity of *Smart Phones*. It is the product of the convergence of regular mobile phones and *Personal Digital Assistants* (PDA) (Chang et al., 2009).

The *Trusted Third Party* must be a powerful server which is able to manage very large amounts data and traffic. We assume it is facilitated with auditing, logging, authorisation and identification, defining the healthcare policies and storage capabilities. Furthermore, the TTP must be connected to a database which contains all details of *Authorised Persons* and patients, including SIM IDs, devices' serial numbers, fingerprint templates, names and national IDs. This information would be provided by an authorized centre when it is needed within relevant legal constraints such as those covered by the *Health Insurance Portability and Accountability Act* in the United States. The HIPAA specifically indicates that patients' privacy should be emphasized (Wei-Bin et al., 2008) and similar legislation is being proposed in other countries introducing EHRs, such as the National E-Health Transition Authority's 'Privacy Blueprints' in Australia. The TTP's technical specifications such as processor's speed and memory would be different for each country and state depending on their population. Such a capability is now possible using current generation network servers from companies such as HP, IBM, or Dell.

The aforementioned characteristics of the SP, AD and TTP make them suitable for the needs of our framework. Our framework needs to provide ubiquitous access to a patient's EHRs, have computerised interaction with a patient and AP, and have a central management system.

3.2 PROPOSED NATIONAL COMMUNICATION FRAMEWORK

As shown in Figure 5, our framework is divided into three levels: the national level, the state level, and the patient level. Each level has its own database which is responsible for storing patient medical records accordingly. Therefore, the framework includes three kinds of databases: a *National-Electronic Health Records Centre* (N-EHRC), *State-Electronic Health Records Centres* (S-EHRC), and patient databases (*Java SIM Cards*).

The patient database stored in a JSC is responsible for storing critical medical information such as their past medical history, blood type, allergies, and the http links (*Uniform Resource Locator*) to the original records and medical images in the central database that we called the *Electronic Health Records Centre* (EHRC).

As the patient's point-of-care location cannot be prearranged, the *Java SIM Card*, due to its intrinsic nature of mobility, can play the role of a portable data repository at the patient's point-of-care in order to help quick diagnosis of a patient's problems. The JSC can make the patient's medical records available across the country or internationally even when the network is not available. It can help doctors who are not familiar with a patient's medical history to access a patient's medical information and ensure proper care is provided. This small database is stored in a *Java SIM Card* which is kind of a *Java Card*.

The *National-Electronic Health Record Centre* (N-EHRC) is a central database which contains all patients' EHRs on a national scale. It is responsible for storing the medical records for all the patients in a particular country. This database must be hosted and maintained by a government authorised organization. The location of this database depends on the network topology in a particular country. The N prefix denotes the country's abbreviation, e.g., 'AU' for Australia and 'IR' for Iran. Therefore, the central database which contains all Australian medical records is designated the AU-EHRC.

The state database or *State-Electronic Health Record Centre* (S-EHRC) is the second tier database and is responsible for storing patients' EHRs, at an intrastate level. The S-EHRC stores a copy of EHRs which belong to patients who reside in a specific state (or province or other relevant division within a country). This database must be hosted and maintained by a local government authority. Again the location

of this database depends on the network topology in a particular country and state. The S-EHRC and N-EHRC work together to provide fault tolerance, better performance, and reliable access to EHRs. The S prefix denotes an abbreviation for the state within a country, e.g., 'QLD' for Queensland or 'NSW' for New South Wales. Hence, the central database which contains all Queensland's patients' medical records is designated the QLD-EHRC.

As shown in Figure 5, the framework also includes two other entities the *National-Trusted Third Party* (N-TTP) and the equivalent state-based *Trusted Third Parties* (S-TTP). The N-TTP operates on a national scale and S-TTPs work on an intrastate scale. The S-TTP is in charge of managing the *Smart Phones* and *Authorised Devices* while the N-TTP is responsible for managing the S-TTPs including authorising, updating, monitoring, enabling, and disabling the S-TTPs.

Having both national and state components is an effective strategy for overcoming both technical and political issues. The technical issues include achieving fault tolerance, better performance, and reliable access to large downloads of data. The political issues include managing different privacy policies between each state related to a healthcare system. This architecture is similar to the Regional Health Information Organisations (RHIOs) in the USA (Search Health IT, 2010), but different to the large healthcare network used in the UK (NHS, 2009). A UK implementation of our model could, for instance, use the countries of England, Northern Ireland**,** Scotland, and Wales instead of states.

In terms of management, each S-TTP is responsible for managing all duties associated with the operation, communication, and maintenance of the SP, AD and S-EHRC within the particular state (Section 4.6). For instance, the S-TTP determines an *Access Level* (AL) for the EHRs based on three factors: the identification of an *Authorised Device* and *Authorised Person* who wants to have access to a patient's medical information, the patient's consent, and the relevant healthcare legislation. Based on these factors the S-TTP maintains three access control lists: *Discretionary Access Control* (DAC), *Mandatory Access Control* (MAC), and *Role Based Access Control* (RBAC) (Kim et al., 2006). The MAC and DAC lists are made by using the patient's consent. The RBAC list is defined by legislators. It has been shown elsewhere (Alhaqbani et al., 2008) that these three constraints can all be used together in a healthcare scenario. The goal is to provide doctors, hospitals, and ambu-

lances with a reasonable level of access to a patient's medical information while still preserving patient privacy. Moreover, in emergency situations only RBAC is valid.

Moreover, interoperability between different medical information systems which cannot communicate with each other is another of the S-TTP's responsibilities. They must be able to recognise a message's context and convert heterogeneous medical information into a unified format (Chenhui et al., 2008). This allows medical information to be exchanged across different healthcare systems.

Accountability is possible only when the S-TTPs are able to provide strong security mechanisms such as access control, audit trails, and authentication of the patient, *Smart Phones, Authorised Persons*, and *Authorised Devices*. For example all access to the medical records must be logged and entered in an audit trail by the S-TTPs.

3.3 DATA SECURITY IN THE FRAMEWORK

In terms of security, this framework relies on PKI, SSL/TLS and *Biometric Identification*. The SSL/TLS protocol is used for implementing secure sessions between a *Smart Phone*, an *Authorised Device* and a *Trusted Third Party*. The PKI is used by the SP and AD to generate *Non-Repudiable* messages (Sections 4.4.6 and 4.5.5). *Biometric Identification* is employed to ensure that only *Authorised Persons* can access a patient's medical information.

PKI is an IT infrastructure which includes a set of procedures, policies, software, hardware, and network services that support security mechanisms such as confidentiality, integrity, authentication, and non-repudiation (Kambourakis et al., 2005). PKI utilises public and private keys for encryption and decryption of sensitive information. If one key is used to encrypt information, then only the related key can decrypt that information. If you know one of the keys, you cannot easily calculate what the other one is. A public key is made public; it is freely distributed and can be seen by all. A private key is kept secret; it is not shared amongst other users. A private key enables you to prove, undeniably, that you are who you claim to be, since only the private key's owner can encrypt data that can be decrypted with the corresponding public key. Others who want to send you a confidential message must encrypt the message with your public key. And if you want to send a confiden-

tial message to a person who has your public key, you must encrypt the message with your own private key (Articsoft, 2009).

Biometrics refers to automated techniques for uniquely recognising a person based on a natural physiological or behavioural feature. Features such as the face, fingerprints, hand geometry, handwriting, iris patterns, retinal patterns, veins, and voice are measured for recognition. Biometric technologies are emerging as a foundation of extremely secure identification and personal verification solutions (The Biometric Consortium, 2009).

3.4 WIRELESS COMMUNICATIONS IN THE FRAMEWORK

In relation to IP-Wireless and contactless communication, the framework uses *Next Generation Network* and *Near Field Communication* technologies respectively. The NGN is utilised to establish *Internet Protocol* (IP) based wireless communication between the *Smart Phone* and *Trusted Third Party*, and the *Authorised Device* and TTP. The NFC is used to facilitate contactless communication between the AD and SP.

As shown in Figure 6, a *Next Generation Network* is an IP-based network that handles multiple types of traffic (such as voice, data, and multimedia). It is the convergence of service provider networks including the *Public Switched Telephone Network* (PSTN), the Internet, and the wireless network. An NGN provides telecom services and can make use of multiple broadband, quality of service-enabled transport technologies, and in which service-related functions are independent from underlying transport-related technologies. It offers unrestricted access by users to different service providers. It supports generalised mobility which allows consistent and ubiquitous provision of services to users. It uses a unique and shared core network for all types of access and services. The core network is divided into three layers: Transport, Control and Services. There are open and standardised interfaces between each layer, and in particular for the Control and Services layers in order to allow third parties to develop and create services independent of the network (ITU-T Next Generation Network, 2009). We contend that it is possible to access a wide range of ubiquitous e-health services through this unified network. *Java SIM Cards* are already used in the NGN for authentication, communication and security; we believe it

is possible to expand those functionalities by using the JSC as a portable repository of EHR data at the patient point-of-care.

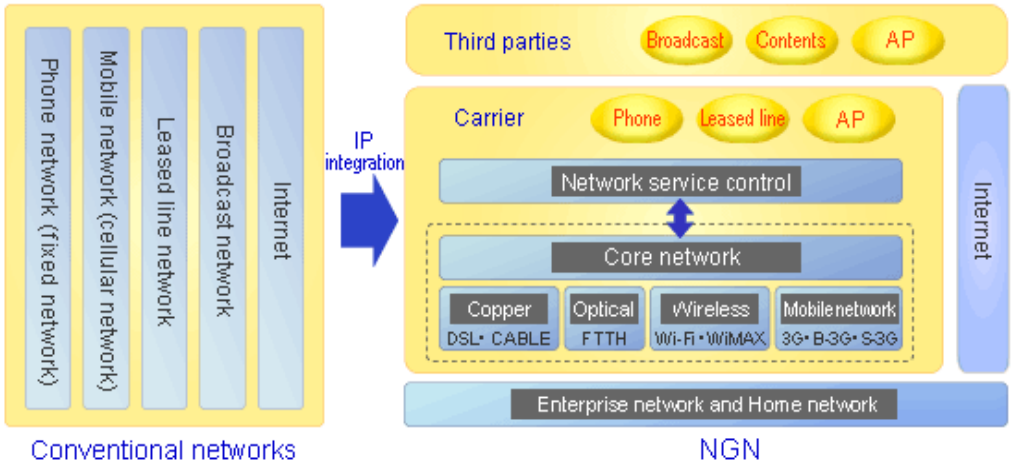


Figure 6. Comparison of NGN and Conventional Networks[5]

Near Field Communication is a wireless connectivity technology evolving from a combination of contactless identification and networking technologies (Morak et al., 2009). It enables convenient short-range communication between electronic devices and smart objects. NFC advanced from Radio Frequency Identification (RFID) technology and is still compatible to certain parts of the existing RFID infrastructure. NFC provides a data transmission rate of up to 424 kbit/s within a short range of typically 5 to 20 centimetres. This short range may not be seen as drawback but, in fact, it is the major feature of this technology because NFC enables rapid and easy communications between two devices which is initiated just by bringing them close together. Due to this usability enhancing feature NFC is bound to be integrated into various types of consumer devices, in particular in handheld devices like PDAs and mobile phones.

In our framework NFC plays the role of a contactless communication protocol between a *Smart Phone* and an *Authorised Device.* Through this technology patients can send their consent to the *Authorised Device* and an *Authorised Person* can see the patient's critical health information contained within a patient's device when it is needed (Section 4.4.5). The communication protocols, in which these devices can work together via NFC, are explained in Chapter 4.

[5] Source: http://www.oki.com/en/ngn/overview/

3.5 DATA STORAGE IN THE FRAMEWORK

In terms of storing medical data our framework use two large databases at the national and state level and a portable small one which is carried by patients on their *Java SIM Cards*. This portable database is responsible for storing critical medical information.

A *Java SIM Card* is a *Subscriber Identity Module* (SIM) card for mobile networks which is made based on a *Java Card*. Due to the OPEN Platform features and strong security aspects of the Java language and Java Card, JSC supports *Remote File Management* (RFM), data security, speed enhancement and anti-cloning functions. The product is highly secure, efficient, easy to manage and provides many possibilities for supporting various applications (Tele Pak, 2009), and is thus well-suited to storing healthcare records.

Java Card is a technology that enables a smart card (chip card, or integrated circuit card) with limited memory to run small Java-based applications (applets). In terms of flexibility, it provides pre-issue and post-issue modifying which allows the developer to fix bugs, and load and delete new data even after a card has been issued. Also Java Cards can load and reuse applications written in the Java programming language from different vendors. As shown in Figure 7, a Java Card consists of three components a *Java Card Virtual Machine* (JCVM), a *Java Card Runtime Environment* (JCRE), and an *Application programming interface* (API) (Sun Microsystems, 2009).

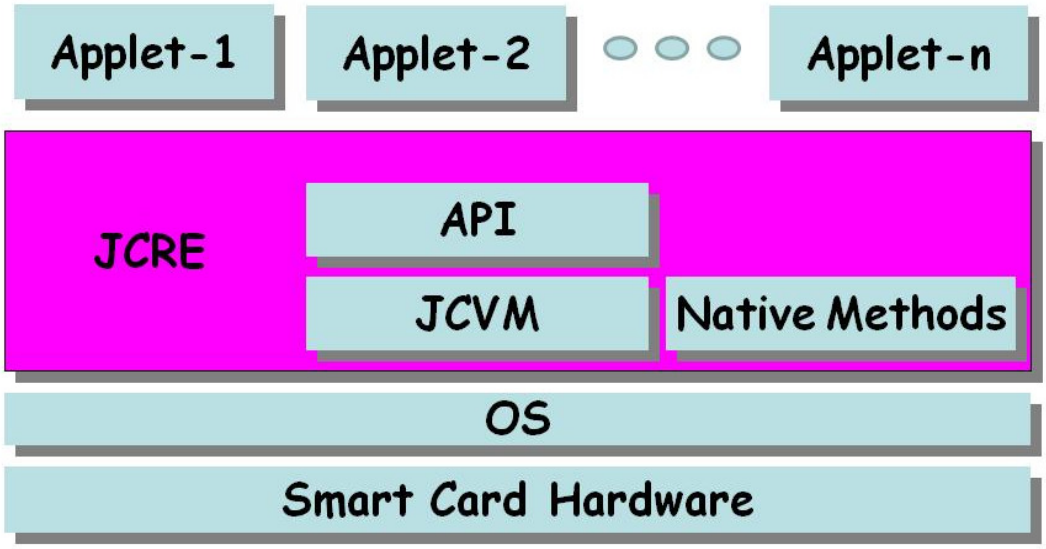


Figure 7. Java Card Architecture

3.6 PROPOSED LOCAL COMMUNICATION FRAMEWORK

As shown in Figure 8, for the purpose of storing an individual's lifetime health information, the local framework contains two central databases called the *Australian-Electronic Health Record Centre* (AU-EHRC) and *Queensland-Electronic Health Record Centre* (QLD-EHRC) in our particular example. In addition, *Java SIM Cards* are used as a patient point-of-care medical information repository. The state-level QLD-EHRC database includes only a copy of records for those patients who already live in Queensland or who temporarily visit the state. If a patient visits or moves to any state in Australia, other than his/her own state and needs medical care, the medical record is automatically fetched from the national database (AU-EHRC) and inserted into the visited state's database such as the QLD-EHRC. This strategy minimises traffic to the national AU-EHRC in a way similar to GSM mobile networks (Peersman et al., 2000).

For the purpose of security, various mechanisms and protocols such as *Biometric Identification*, the *Secure Sockets Layer/Transport Layer Security* (SSL/TLS) protocol and *Public Key Infrastructure* encryption must be utilised to achieve secure access to patients' medical information. As we assume the communication between the *Smart Phone* or *Authorised Device* and *S-TTP* is based on a *Next Generation Network* which uses the Internet as a carrier, our framework must use PKI and SSL/TLS for the purpose of data confidentiality, integrity, authentication, and non-repudiation. These technologies enable the SP, AD, and TTP to exchange sensitive information through the unsecure public network. The patient's *Private Key* is stored in the *Java SIM Card* which has the capability of operating cryptographic algorithms, while the patient's *Public Key* is stored by the AU-TTP and QLD-TTP.

Biometric Identification such as fingerprints must be used for accurately authenticating a patient or an AP (Sections 4.4.1 and 4.5.1). In principle, biometrics cannot be forgotten or lost, and are difficult to duplicate or share among different users (Hu, 2008). A biometrics authentication system requires physical presence of the individual. Among several biometric technologies, fingerprints have been in use for the longest time and have more advantages than others. For instance, there are many devices such as *Smart Phones* on the market that are equipped with a fin-

gerprint scanner. A fingerprint is captured via live scan and then its features are retrieved. The retrieved fingerprint's features are then fed into the JSC, where the fingerprint template is stored, for the matching process. If a match is made, access to the medical information will be granted.

In terms of IP-Wireless and Contactless Communication this framework relies on *Next Generation Network* and *Near Field Communication* technology. While the NGN is used for IP-Wireless communicating, the NFC technology is used for contactless communication between an *Authorised Device* and *Smart Phone*.

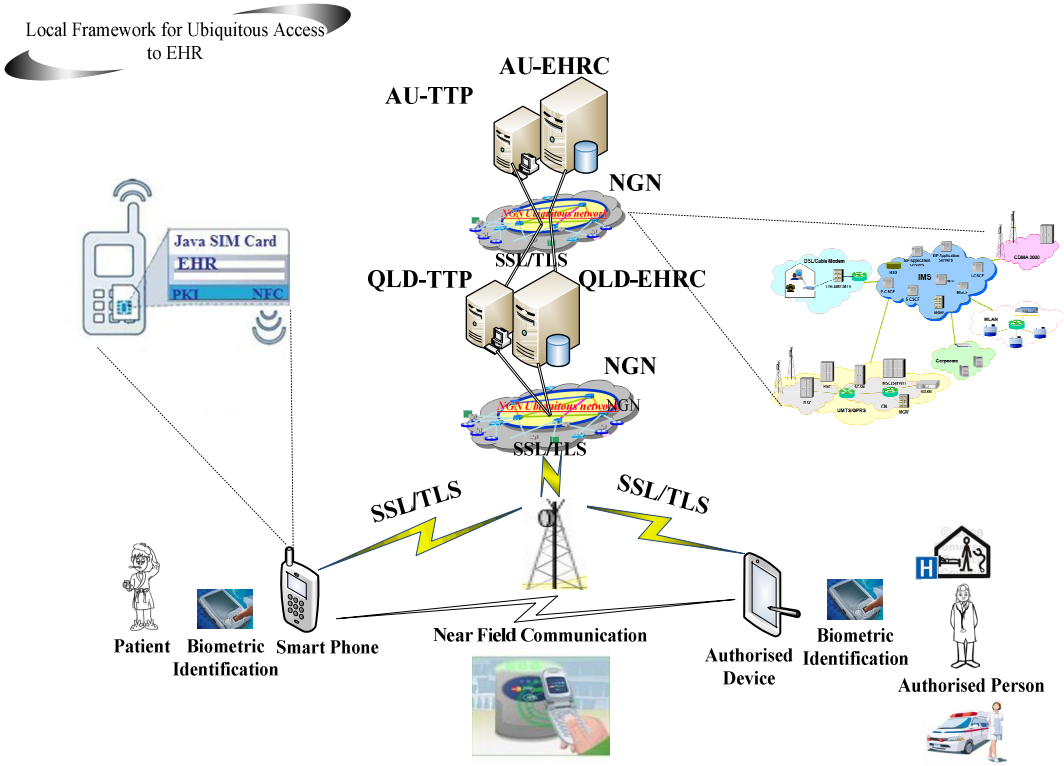


Figure 8. Local Framework for Managing Ubiquitous Electronic Health Records

3.7 CONTENTS OF A JAVA SIM CARD (JSC)

Figure 9 splits the patient's *Java SIM Card* into three layers for managing ubiquitous *Electronic Health Records*: the application and database layer (*Applet Layer*), the *Java Card Runtime Environment* (JCRE) layer, and the *OS/Hardware* layer. While the JCRE and OS/HW layers are specific to a particular computing platform, we require an *Applet Layer* to address the design requirements specific to

the development of *Ubiquitous Access to Electronic Health Records*. Current SIM cards are capable of supporting these features.

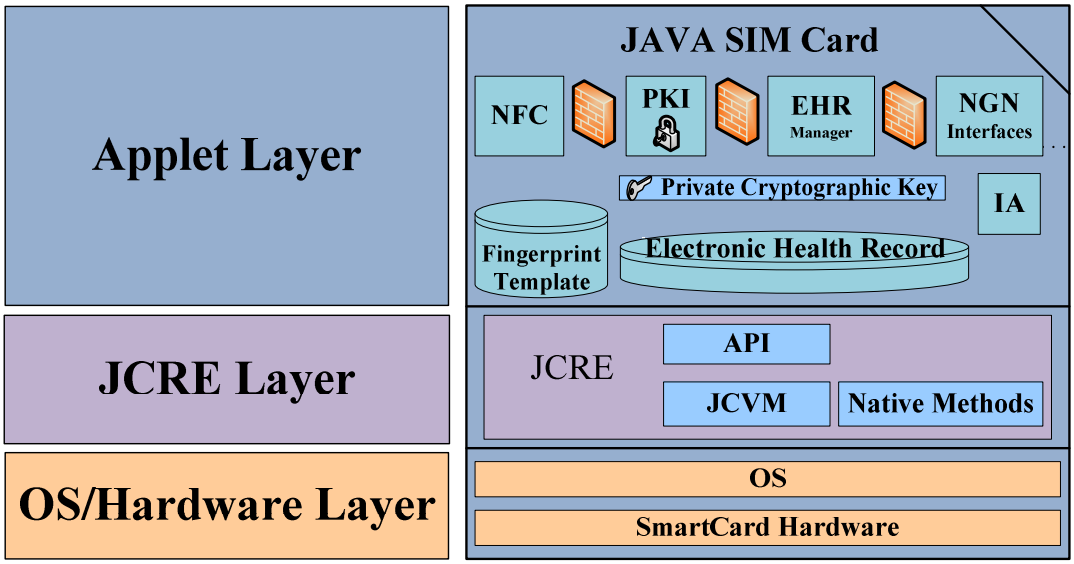


Figure 9. Java SIM Card Architecture for Managing EHRs

The *Applet Layer* must consist of an EHR-Manager, PKI, NFC, *Identification Algorithm* (IA), NGN Interfaces and other applets as necessary. Additionally, this layer includes the *Private Cryptographic Key* and a small database which contains critical medical information such as past medical history, blood type, allergies, and the HTTP links (Uniform Resource Locator) to original records and medical images in the central database.

The EHR-Manager interface is needed to handle communication between the state-level QLD-TTP and the *Java SIM Card* in order to update the EHRs contained within the *Java SIM Card*. After an *Authorised Person* updates a patient's EHRs within the QLD-EHRC, the QLD-TTP automatically updates the EHR within the JSC by using the *EHR-Manager* interface, *Smart Phone*, *Next Generation Network*, SSL/TLS protocol and *Over The Air* (OTA) and *SIM Tool Kit* (STK) technologies. OTA and STK technologies are widely adopted in mobile communication system to read and write the contents of *Java SIM Cards* (Chin et al., 2006). After a patient confirms the acknowledgement SMS message coming from the QLD-TTP by accepting the SMS request, the update will be done. The protocol for interaction be-

tween the patient and the SMS service is handled by the telephone network (Chin et al., 2006), and is not part of our EHR infrastructure.

The JSC must be equipped with a *Near Field Communication Application Programming Interface* (NFC API) in order to exchange information with an *Authorised Device* over about a 10 centimetre distance. The NFC enables the *Smart Phone* and *Authorised Device* to have contactless communication. By using the NFC a *Smart Phone* is able to send a patient's consent to the *Authorised Device* or the *Authorised Person* is able to see the patient's EHR within the JSC. There are many mobile phones such as Nokia 6216 and 5800 in the market that support a SIM-based NFC interface.

The *Identification Algorithm* (IA) is needed for the challenge-response mechanism (Section 4.2) occurring between the *Smart Phone* or *Authorized Device* and the QLD-TTP (Sections 4.4.4, 4.5.2, and 4.6.4). Challenge-response is important in the process of identifying a remote source as either an individual or a device. This algorithm works similarly to GSM authentication (Haverinen et al., 2001). IA uses the challenge and the unique embedded information which is stored within a tamper-resistant *Java SIM Card* to generate a *Fresh ID*. The unique embedded information include the SIM ID, device serial No, scanned fingerprint, name, and national ID.

As shown in Figure 9, the *Java SIM Card* must also be facilitated with a *Public Key Infrastructure* API in order to implement secure sessions between the SP and AD and QLD-TTP (Section 4.6.1) and produce *Non-Repudiable* messages (Sections 4.4.6 and 4.5.5). We assume that all the *Public Keys* of the patients who reside in QLD are stored in the QLD-TTP and it is responsible for managing them. Therefore, if the AD needs to communicate with the SP, it must first connect to the QLD-TTP to get the patient's *Public Key*. Moreover, the AU-TTP must be able to put the patients' *Private Keys* on their *Java SIM Cards* remotely by using *Over The Air* (OTA) and *SIM Tool Kit* (STK) technologies.

For the sake of clarity, we suppose that when the SP needs to send any information to the AD or QLD-TTP, this information must be encrypted by the *Private Key* which is contained within the patient's JSC. An *Authorised Device* or QLD-TTP can decrypt the message received from a *Smart Phone* by using the patient's *Public Key*. If the AD needs to send any information to the SP, first it must get the patient's *Public Key* from the QLD-TTP, and then it must encrypt the information with

the patient's *Public Key*. The SP can decrypt the message received from the AD or QLD-TTP by using its own *Private Key* within the JSC.

The *Java SIM Card* is a multi-application environment. Multiple applets from different vendors can coexist in a single card, and additional applets can be downloaded after card manufacture. An applet often stores highly sensitive information, such as EHRs, fingerprints, private cryptographic keys, and so on. Sharing such sensitive data among applets must be carefully limited (Chen, 2000). Therefore, for further security the JSC must have a firewall between the applets in order to achieve isolation and restrict access to the data of one applet from another as shown in Figure 9. In other words, the firewall confines each of the applets to their allocated area.

Furthermore, in the case of the *Java SIM Card* being lost or stolen, the JSC can protect the information within the card by requiring a fingerprint and PIN code to access the patient's medical information. Also after the QLD-TTT is aware of a card going missing, it can remotely disable the card and delete all personal information inside the card. There are many Smart Phones in the market that equipped with fingerprint scanners such as the Nokia 5800.

In addition, since *Java SIM Cards* are self-contained they are resistant to attack as they do not need to depend upon potentially vulnerable external resources. Also attacking *Java SIM Cards* requires professional equipment and materials which are expensive and not easily available (Bar-El et al., 2006) (Gammel et al., 2005). An average patient could not attack a JSC and change the medical information within it.

3.8 SUMMARY

In this chapter we described our proposed framework. It employs emergent technologies such as *Java SIM Cards* (JSC), *Smart Phones* (SP), *Next Generation Networks* (NGN), *Near Field Communications* (NFC), *Public Key Infrastructure* (PKI), and *Biometric Identification* to produce a secure framework for ubiquitous access to *Electronic Health Records* (EHRs). The framework is divided into three levels: national, state, and the patient's level. Each level has its own database which is responsible for storing patient medical records accordingly. Therefore, this framework includes three databases: a national database (N-EHRC), state databases

(S-EHRC), and patient databases (*Java SIM Cards*). Also this framework includes two kinds of management centre, national management (N-TTP) and state management (S-TTP) which are responsible for supervising the whole of the framework's activities. Moreover, each JSC is equipped with remote management agents which are used by the S-TTP to manage patients' databases.

Our framework will contribute significantly to improving *Ubiquitous Access* to the patient's medical records by simplifying treatment, automating interactions, increasing patient safety, improving quality of care, and reducing the costs.

Our framework's components are divided into two groups; the first one includes the components which already exist, and the second includes new ones need to be developed. In the first group we have JSC, SP, AD, NGN, NFC, PKI, and fingerprint identification all of which are currently available 'off the shelf '.

In the second group we have the new components that must be developed, including the TTPs, N-EHRC, S-EHRCs and the *Applet Layer* of the JSC which consists of the EHR-Manager, PKI, NFC, *Identification Algorithm* (IA), and other necessary applets.

Central to the new components needed to support the framework are its electronic communication protocols. In Chapter 4 we present a set of protocols for communication between the SP, AD, and TTP and in Chapter 5 we demonstrate their feasibility in different healthcare scenario.

Chapter 4: Ubiquitous Access Protocols for Exchanging Electronic Health Records

In Chapter 3, we introduced a framework in which an *Authorised Person* such as a doctor is able to have *Ubiquitous Access* to a patient's medical record on a national scale. We also saw what components the framework needs and how the components fit together in general terms. The chapter concluded by summarising which components of the framework already exist and which new ones need to be developed. Central to the new components needed to support the framework are its electronic communication protocols.

In this chapter we explain in detail three different protocols needed for communication between the *Smart Phone* (SP), *Authorised Device* (AD), and *Trusted Third Party* (TTP). To do this we express our protocols using the *Specification and Description Language* (SDL). All three protocols work on the application layer which means they are independent of the underlying protocol layers for end-to-end communication. The application layer is the 7th layer in the Open Source Interconnection (OSI) reference model (ITU-T X.210, 1993) and is closest to the end user. The SP, AD, and TTP protocols define the processes and procedures that must be followed when these agents want to request data from, respond to and communicate with each other.

4.1 SDL OVERVIEW

The *Specification and Description Language* (SDL) is a formal language defined by the International Telecommunications Union (ITU–T) as the Z.100 recommendation. SDL is used to describe unambiguous specifications and descriptions of the behaviour of real time systems. SDL diagrams are used to model communicating state machines in different industries, especially telecommunications. A specification of a system is the description of its required behaviour and a description of a system is the description of its actual behaviour; that is, its implementation. A system specification, in a broad sense, is the specification of both the behaviour and a set of general parameters of the system (ITU-T Z.100, 2007).

As shown in Figure 10, SDL can be used in all levels of specification, designing, and implementation of a system. It provides a way to describe behaviour to support human communication and understanding. Each process in SDL is an extended finite state machine and the behaviour of a finite state machine is described by states and transitions. A process description is given through a process diagram. In contrast with flowcharts, which can be used to describe a sequence of conditional activities, SDL is based on a state machine concept in which each concurrent state machine is described by a separate flowchart. SDL also uses asynchronous communication channels in order to establish a one-to-one or one-to-many session between processes and entities (ITU-T Z.100, 2007).

The basic theoretical model of an SDL system consists of a set of extended *Finite State Machines* (FSMs) that run in parallel. These machines are independent of each other and communicate with discrete signals. An SDL system includes the following components (Figure 11):

structure—system, block, process, and procedure hierarchy

communication—signals with optional signal parameters and channels (or signal routes)

behaviour—process flowcharts

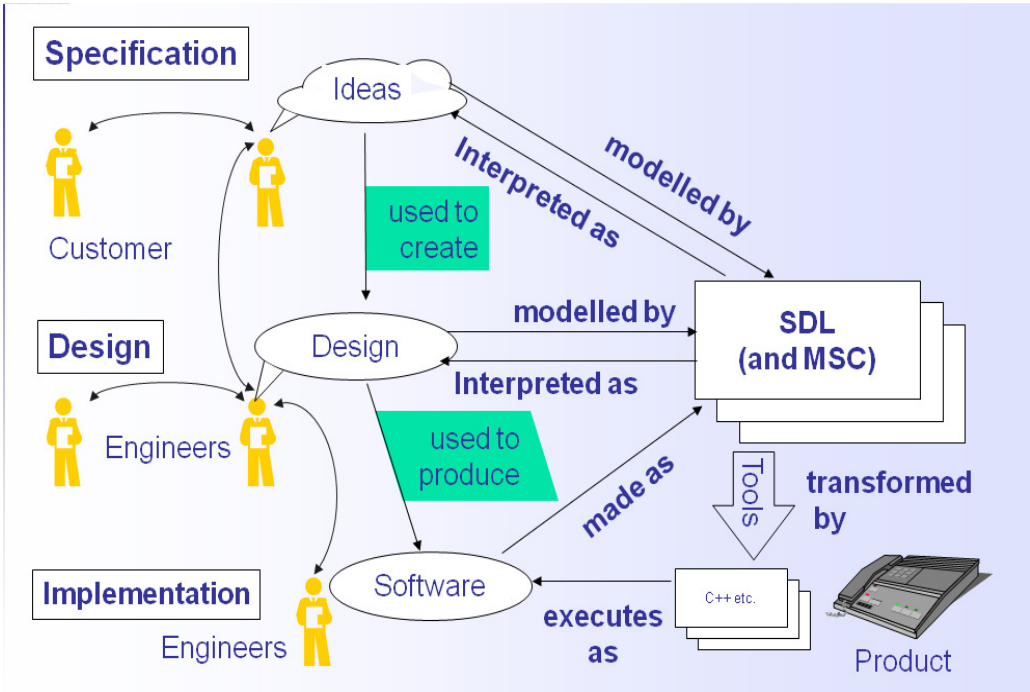


Figure 10. Uses of SDL[6]

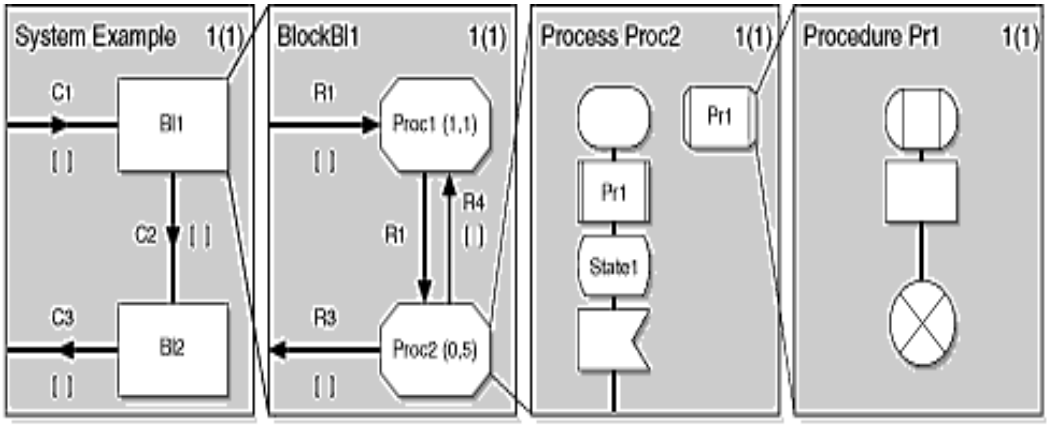


Figure 11. The Structural View of an SDL System[7]

To present our protocols we use the process symbols start, state, input, output, decision, procedure, return, off-page reference, and document in Figure 12. Our protocols are split into several figures which contain SDL symbols, italic labels, and off-page references. The SDL symbols in which there is informal text for describing the procedures are used to describe steps in the protocols. In Chapter 5 we use our

[6] Source: www.disi.unige.it/person/ReggioG/ISII01/SDL.ppt

[7] http://www.iec.org/

SDL processes as the base for automatic simulation of the whole system's behaviour. The italic labels beside the SDL symbols in Figure 13 onwards are not part of the original protocols; we use them to map the SDL figures to our simulation tool model and make our simulation code more readable and traceable. The off-page references enable us to create links between figures.

Another feature of SDL mentioned in Figure 10 is the *Message Sequence Chart* (MSC), as a way of documenting possible dynamic sequences of interaction between processes. *Message Sequence Chart* (MSC) is a standardised notation used for the description of typical or exceptional message exchanges between entities. MSC diagrams provide a clear description of system communication in the form of message flows (ETSI, 2009). MSC and SDL descriptions should be regarded as different but complementary views of a system. SDL provides behaviour descriptions of individual communicating entities, but there is no direct description of communication between several entities. By contrast, MSC provides a clear description of system traces between processes (ETSI, 2009). In Chapter 5 we validate our SDL model by using a simulation tool to automatically generate *Message Sequence Charts* from our protocols.

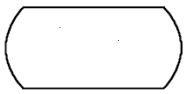
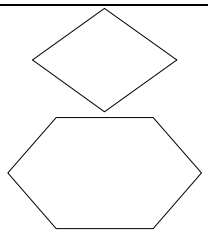
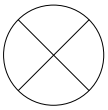
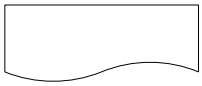
Symbol	Name	Description
	Start	This symbol is specific to a procedure diagram. It indicates the procedure entry point.
	State	This symbol represents a state. The state symbol means the process is waiting for some input to go on.
	Input	An input allows the consumption of the specified input signal instance.
	Output	An output sends the specified signal instance. It is used to exchange information. It puts data in the receiver's message queue in an asynchronous way.
	Decision	A decision transfers the interpretation to the outgoing path whose range condition contains the result given by the interpretation of the question.
	Off-page reference	It is used to split a transition into several pieces so that the diagram stays legible and printable, and to gather different branches to a same point.
	procedure	A procedure box either contains informal text describing an action or a reference to another SDL diagram.
	Return	This symbol is specific to a procedure diagram. It indicates the end of the procedure.
	Document	This symbol is used to add some additional information to the SDL diagram.

Figure 12. SDL Basic Symbols

4.2 CHALLENGE-RESPONSE MECHANISM IN OUR FRAMEWORK

The ability to accurately identify patients, *Authorised Persons* (AP) and their devices, which are interacting with the TTP, is critical for ubiquitous access to EHRs. Also accountability is possible only when strong security mechanisms such as access control, audit trails, and authentication are implemented and the identity of the patients, APs and their devices can be verified. In our protocols the *Smart Phone* (SP) and *Authorised Device* (AD) are identified by the TTP based on a challenge-response mechanism. Challenge-response is important in the process of identifying a remote source as either an individual or a computer. When the SP or AD wants to be identified by the TTP, the TTP sends a challenge (*Random Number* or '*nonce'*) to the SP or AD. The SP or AD sends the challenge to the *Java SIM Card* as input to *Identification Algorithm* (IA). IA uses the challenge and the unique embedded information which is stored within a tamper-resistant *Java SIM Card* to generate a *Fresh ID* (Session ID). The unique embedded information include the SIM ID, device serial No, scanned Fingerprint, name, and National ID. The *Fresh ID* is then relayed to the TTP by SP or AD for validation. As the IA constructs the *Fresh ID* from the two inputs, one of which is constant value (unique embedded information), and one of which is a variable (challenge), it is able to generate a different ID from previous ones for each time the SP or AD wants to be identified. This helps to increase the security of our identification process. To ensure that the challenge is used only once, it should be time-variant, or generated randomly. As long as we use this mechanism and the identification process is protected by cryptography, entities can be identified accurately and malicious users can not access the system. This mechanism is very similar to GSM authentication (Haverinen et al., 2001) and the use of a *Message Authentication Code* (Arazi, 2009).

As mentioned above the challenge-response mechanism occurs between the SP or AD and the TTP. Once the TTP receives a request for identification, it generates a *Fresh ID* (TTP-FID) by using the same information and algorithm which the JSC uses for generating its *Fresh ID*. When the TTP receives the SP or AD's response (SP-FID or AD-FID), it checks to be sure the ID generated by the SP or AD (SP-FID or AD-FID) equals the ID generated by its (TTP-FID). If so, the SP or AD is identified successfully. If not, the identification fails and the SP or AD cannot access

the medical information. This allows the TTP to prevent impersonation of the patient or AP being identified.

4.3 BIOMETRIC AUTHENTICATION IN OUR FRAMEWORK

A personal authentication system is a system that verifies a person's identity usually through a login name or smart card, etc. Traditional authentication is based on the possession of a secret key, that is, once the user possesses the key, his or her authenticity is established. Personal authentication based on PKI is one of the most prevalent authentication methods, which uses a private key to prove the user's identity. Usually cryptographic keys are long and random, (e.g., the key length of RSA is 128 bits), so they are difficult to memorise. As a result, cryptographic keys are stored on a smart card whose access privilege is protected by *Personal Identity Numbers* (PINs), which is a kind of password, or stored on computer and protected by a PIN. Usually PINs are so simple that they can be easily guessed or cracked by dictionary attacks. Simple passwords are easy to break and thus compromise security; complex passwords are difficult to memorise. Finally, passwords cannot provide non-repudiation, that is, when a password is shared with others, it is difficult to know who the actual user is.

Because traditional authentication, based on cryptographic keys, has so many limitations, biometric identification is added to our framework due to its many advantages of security and usability over cryptographic keys. Biometric authentication refers to verifying individuals based on their physiological or behavioural patterns such as fingerprint, face, etc. Biometric characteristics cannot be lost or forgotten, they are extremely difficult to copy, share, and distribute, and require the person being authenticated to be present at the time and point of authentication. It is difficult to forge biometrics, requiring considerable time, money, experience, and access privileges, and it is difficult for a user to repudiate having accessed the digital content using biometrics (Li et al., 2006). As a result, using biometrics in addition to a PIN is strongly recommended for electronic healthcare applications.

4.4 SMART PHONE PROTOCOL

Our *Smart Phone* (SP) protocol specifies the processes and procedures that must be followed when a patient wants to request data from, respond to or communicate with *Authorised Devices* (AD) or a *Trusted Third Party* (TTP). The protocol must work in various emergency and non emergency scenarios. In total the SP protocol includes seven major processes and procedures, each of which is modelled by following a particular path through the protocols presented below. These procedures and processes are: *Authenticating the Patient*, *Modifying the Access Control List*, *Viewing EHRs*, *Identifying the Patient*, *Granting Consent*, *Non-repudiable Consent*, and *Emergency Situations*. They will be explored in detail in the following sections. The entire SP protocol is described by the SDL diagrams in Figures 13 to 18.

In this section we assume the patient has a *Smart Phone* which has a fingerprint scanner, a *Java SIM Card* containing *Near Field Communication* applications, the patient's fingerprint template, the patient's partial *Electronic Health Record*, a PKI handler, the patient's *Private Key*, a *Next Generation Network* interface, an *Identification Algorithm*, and an *EHR manager*. The *Smart Phone* is also able to connect to the web using the SSL/TLS protocol, communicate with the *Java SIM Card*, find and store the three nearest *Emergency Room*'s details, automatically call the *Emergency Room*, automatically send *Short Message Service* (SMS) messages, set a timeout, detect strong vibration, and communicate with other NFC-enable devices.

4.4.1 PROCESS FOR AUTHENTICATING THE PATIENT

Patient authentication is the first significant step towards making patients capable of granting consent or accessing their health data contained within their EHRs. As shown in Figure 13, for authenticating the patient we require two methods, a PIN and fingerprint scanning (Section 4.3). After a patient selects the *Access to Medical Information Menu* (AMIM) option contained in the menu displayed on the *Smart Phone*, the SP asks the patient to enter the PIN. If the PIN is entered incorrectly three times, the SP blocks access to the AMIM option and the patient has to contact the TTP for unblocking. If the PIN code is entered correctly, The SP asks the patient to put his/her fingerprint on the *Smart Phone Scanner* (SPS) in order to verify it with the fingerprint template contained within the *Java SIM Card* (JSC). If the *Patient Fingerprint* (PF) is read incorrectly three times, the SP blocks access to the AMIM

and the patient has to contact the TTP for unblocking. If the PF and the fingerprint template stored in the JSC match then, the SP goes to the *AccessGranted* state (Figure 13).

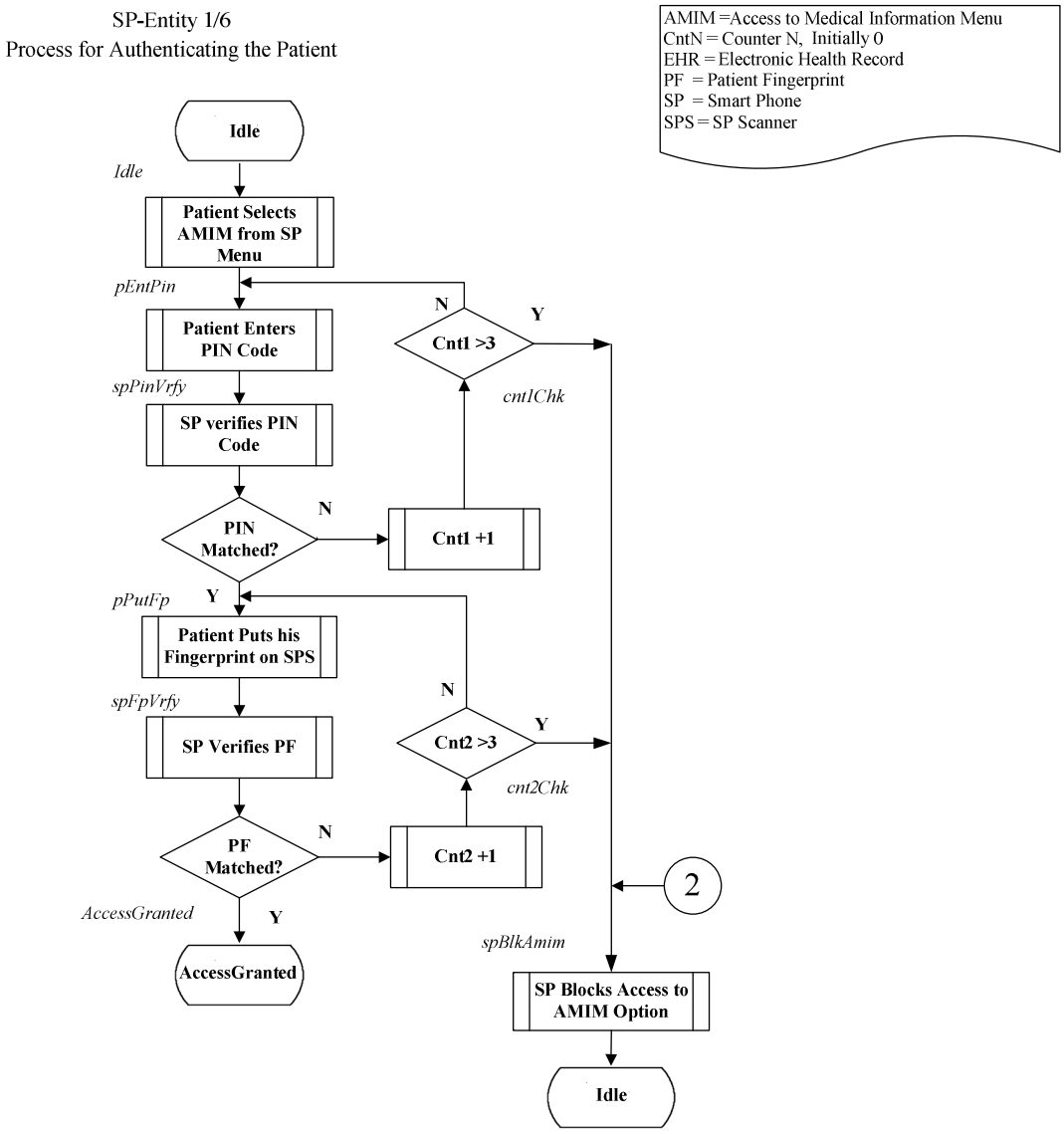


Figure 13. Smart Phone - Process for Authenticating the Patient

4.4.2 PROCESS FOR MODIFYING THE ACCESS CONTROL LIST

Most people consider information about their health to be highly sensitive. They prefer to have rights of access to their medical information and to be entitled to decide who can access their record (Huang et al., 2009). Based on these assumptions, we must have three access control models for *Discretionary Access Control* (DAC),

Role-Based Access Control (RBAC), and *Mandatory Access Control* (MAC) (Kim et al., 2006) which are all stored on the TTP.

Also we propose a *Modifying ACL* procedure which gives patients the power to modify their data's MAC and DAC in order to decide who should access their records and which parts of their EHR can be revealed to others. As shown in Figure 14, the following steps are introduced in the Modifying ACL Procedure:

1) Patient selects the *Modifying ACL* option from his *Smart Phone* menu ($MVCE = 1$).
2) The SP executes the procedure for *Identifying the Patient* to the TTP (Figure 15).
3) If the *Identifying the Patient* procedure returns any errors related to SSL/TLS communication or *Timeout* (these errors are explained in detail in Section 4.4.4):
   a) The SP displays the message "*Communication Error*".
   b) The SP returns to the *AccessGranted* state.
4) Otherwise, if there is no error (*SSLError* or *Timeout*), then the SP goes to Step 5.
5) If the value returned by the *Identifying the Patient* procedure is a "*Confirm"* message:
   a) The patient can modify the ACL according to the *Role-Based Access Control* (RBAC).
   b) The SP returns to the *AccessGranted* state.
6) Otherwise, if the value returned by the *Identifying the Patient* procedure is a "*Non-Confirm"* message:
   a) The SP blocks access to the AMIM option.
   b) The SP goes to the *Idle* state (the normal behaviour of the *Smart Phone*). For unblocking, the patient should contact the TTP.

4.4.3 PROCESS FOR VIEWING EHRS

In 1996, the *Health Insurance Portability and Accountability Act* (HIPAA) offered some general guidelines to enforce the protection of private medical information. One such guideline stated that patients must be able to view and obtain copies of their records, and request amendments to confirm they have the right of accessing their medical records to understand and monitor their health status and the process of diagnosis and therapy (Huang et al., 2009). Looking over the records can help the patient to get rid of the fear factor that a lot of patients have. This option allows

the patient to look at his or her records either as stored locally on their *Java SIM Card* or as stored remotely on the *Electronic Health Records Centre* (EHRC).

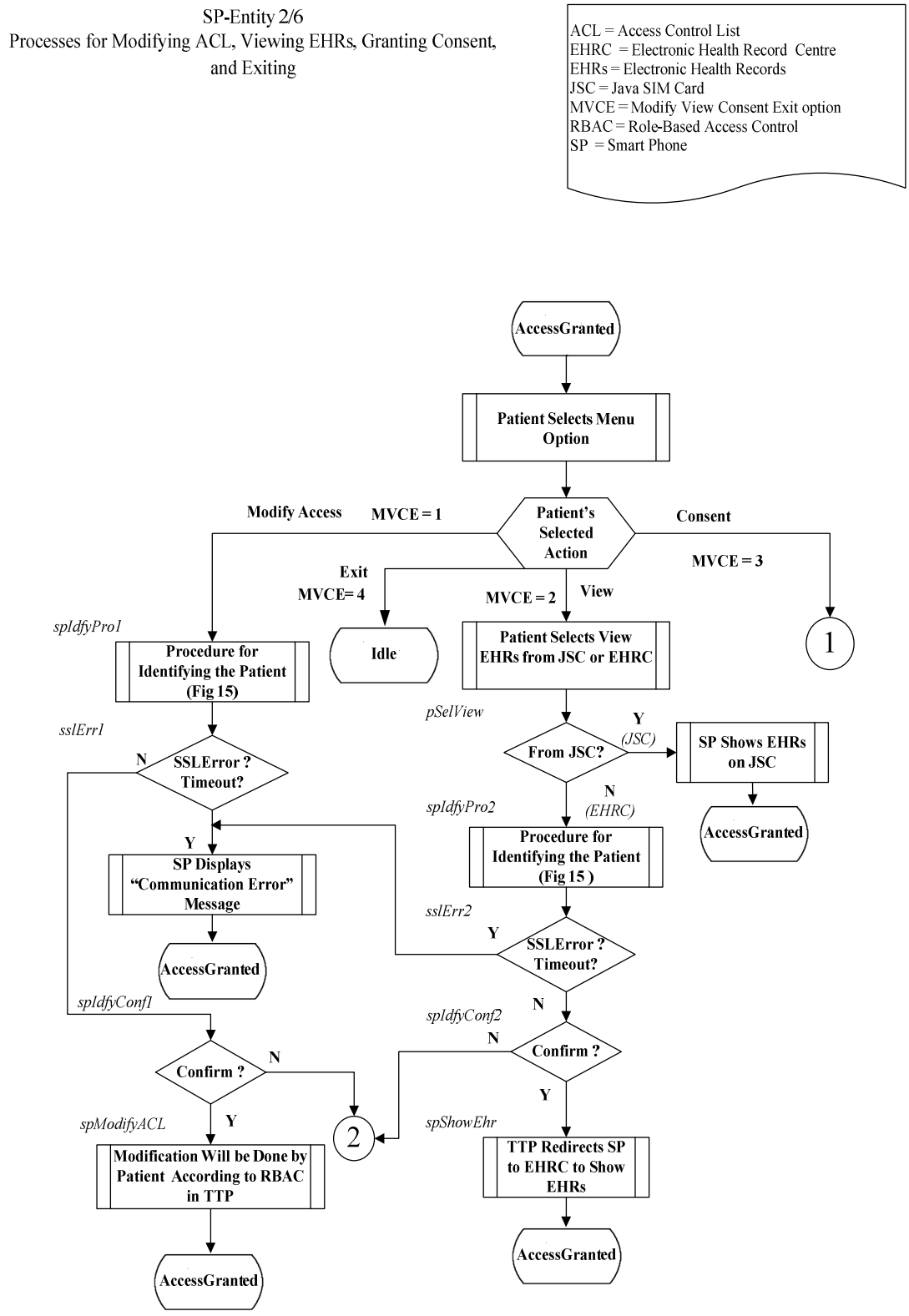


Figure 14. Smart Phone - Processes for Modifying ACL, Viewing EHRs, Granting Consent, and Exiting

The following steps summarise the *viewing EHRs* process, Figure 14:

1) Patient selects the *View* option from his or her *Smart Phone* menu ($MVCE = 2$).
2) Patient selects the *View EHRs from JSC* or *View EHRs from EHRC* option from his or her *Smart Phone* menu.
3) If the patient selects the *View EHRs from JSC* option:
   a) The SP shows the EHRs which are stored within the JSC.
   b) The SP returns to the *AccessGranted* state.
4) Otherwise, if the patient selects the *View EHRs from EHRC* option, then the SP executes the *Identifying the Patient* procedure (Figure 15).
5) If the *Identifying Patient* procedure returns any errors related to SSL/TLS communication or *Timeout* (Section 4.4.4):
   a) The SP displays the message "*Communication Error*".
   b) The SP returns to the *AccessGranted* state.
6) Otherwise, if there is no error (*SSLError* or *Timeout*), then the SP goes to Step 7.
7) If the value returned by the *Identifying Patient* procedure is a "*Confirm*" Message, then the patients can look at their EHR is are within the EHRC. To do this after the patient identification has been confirmed, the TTP redirects (connects) the SP to the EHRC server site through a SSL/TLS session.
8) Otherwise, if the value returned by the *Identifying Patient* procedure is a "*Non-Confirm"* message:
   a) The SP Blocks access to the AMIM option.
   b) The SP returns to the *Idle* state (the normal behaviour of the *Smart Phone*). For unblocking, the patient should contact the TTP.

4.4.4 PROCEDURE FOR IDENTIFYING THE PATIENT

In our *Smart Phone* protocol identification is achieved by using a challenge-response mechanism between the SP and TTP. The challenge-response mechanism is described in detail in Section 4.2. The procedure for *Identifying the Patient* is utilised by the SP specifically when a patient wants to grant consent, modify an ACL, or view his/her health record contained within the EHRC. Some of these processes have already been described and some will be described later in this chapter.

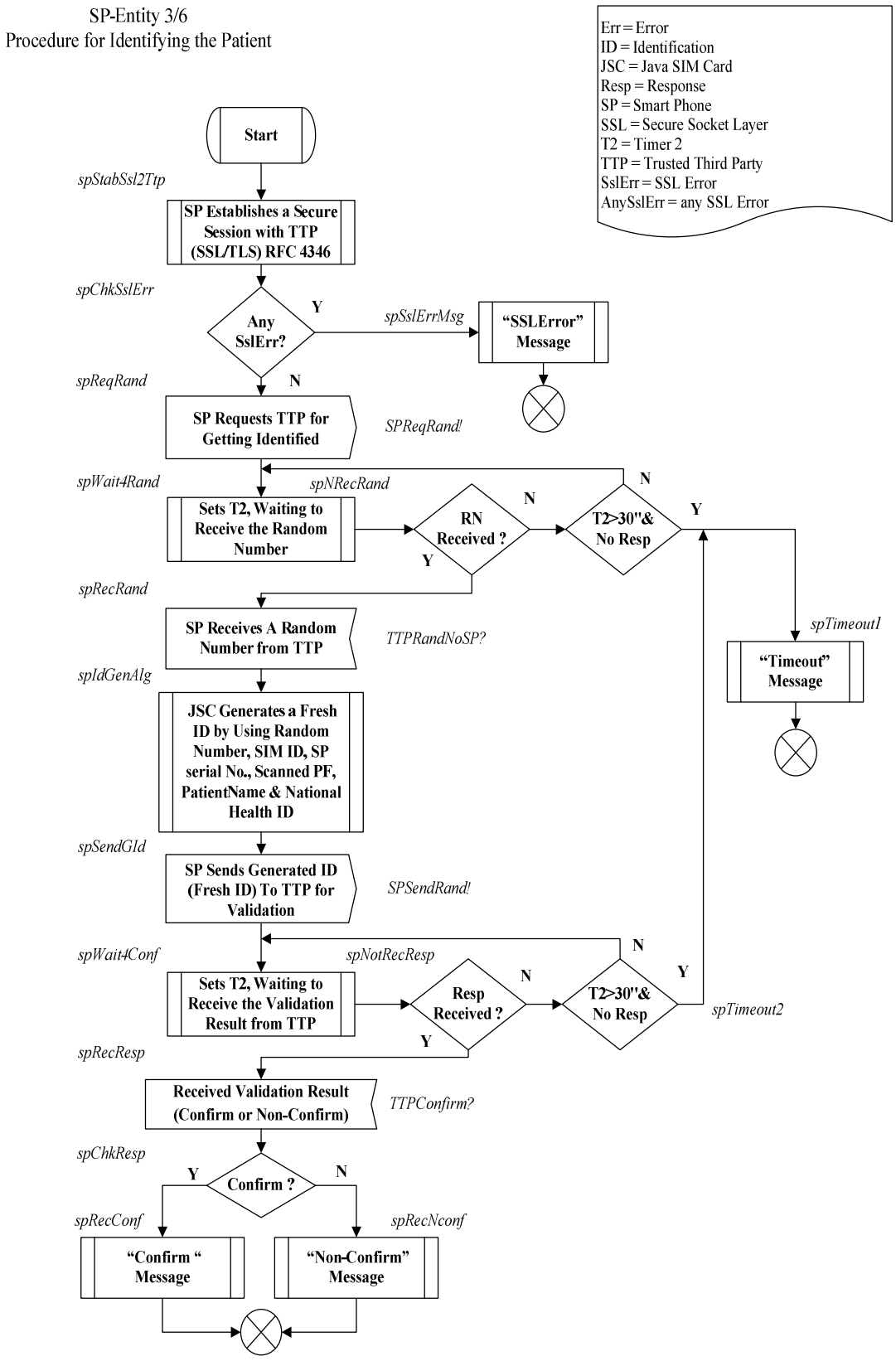


Figure 15. Smart Phone - Procedure for Identifying the Patient

As shown in Figure 15, the procedure for *Identifying the Patient* is outlined in the following steps:

1) The SP establishes a SSL/TLS session with the TTP.

2) If any SSL/TLS error (IETF, 2008) occurs, then the SP returns from the *Identifying Patient* procedure with are "*SSLError*" value.

3) Otherwise, if no SSL/TLS error occurs, then the SP informs the TTP "I need to be identified" and requests the TTP to send a *Random Number*. The *Random Number,* in addition to the other information, is needed by the *Identification Algorithm* within the JSC in order to generate a *Fresh ID*. The SP sends the *Fresh ID* to the TTP for getting identified.

4) If there is no response (*Random Number*) from the TTP within a certain time, the SP returns from the *Identifying the Patient* procedure with a "*Timeout*" value.

5) Otherwise, if the SP receives a *Random Number* from the TTP, the SP gives the *Random Number* to the *Java SIM Card*.

6) The JSC generates a *Fresh ID* and the SP sends it to the TTP for validation.

7) If there is no response (*Confirm* or *Non-Confirm*) from the TTP within a certain time, then the SP returns from the *Identifying the Patient* procedure with a "*Timeout*" value.

8) Otherwise, if the SP receives a "*Confirm"* message from the TTP (if the TTP's generated ID equals the SP's generated ID), then the SP returns from the *Identifying the Patient* procedure with a "*Confirm"* message value.

9) Otherwise, if the SP receives the "*Non-Confirm"* message from the TTP (if the TTP's generated ID does not equal the SP's generated ID), then the SP returns from the *Identifying the Patient* procedure with a "*Non-Confirm*" value.

4.4.5 PROCESS FOR GRANTING CONSENT

Patient consent is essential prior to access or release of any information related to the patient's medical information. Patients can allow or deny sharing their information with healthcare workers. Access to the patient's medical information is prohibited unless the patient has given consent or the patient is in an emergency situation or there is legislative permission (Van Der Linden et al., 2009). Therefore the *Granting Consent* procedure is one of the important parts of this protocol. It is necessary for the purpose of computerised determination of whether sharing of information is legally allowed. As shown in Figures 16 and 17, the *Granting Consent* process is used when a patient wants to give his or her consent to an *Authorised Person* such as a doctor and the doctor needs to view or update the patient's medical information during the examination. The communication between the patient's device and the doctor's device is made by NFC technology, using the SP and AD protocols.

There is a key issue underpinning this procedure about whether the patient trusts the doctor with or without recourse to the *Trusted Third Party* (TTP). As there is not any record in the *Java SIM Card* for the *Authorised Person* it means this is the first time that the patient has gone to the AP, so the AP must be identified by the *Trusted Third Party*. And based on the identification, the TTP authorises the AP and determines the *Access Level* for the patient medical information. Therefore, the SP asks the AD to introduce itself by sending its information which includes the AD's ID, the AP's name, and ID, and the AP's digital signature. This information must be encrypted by the AP's Private Key. The AD will respond to the "*Introduce Yourself*" with a "*Setup Message*". Then, the SP sends the *Setup Message* to TTP for confirmation. If the TTP confirms the SM, the SP stores a record in the JSC for the AD and AP for future use. If the TTP does not confirm the SM, the SP stores a record for the AD and AP in a black list within the JSC. As shown in Figures 16 and 17, depending on the patient's decision, the *Granting Consent* process is summarised in the following steps:

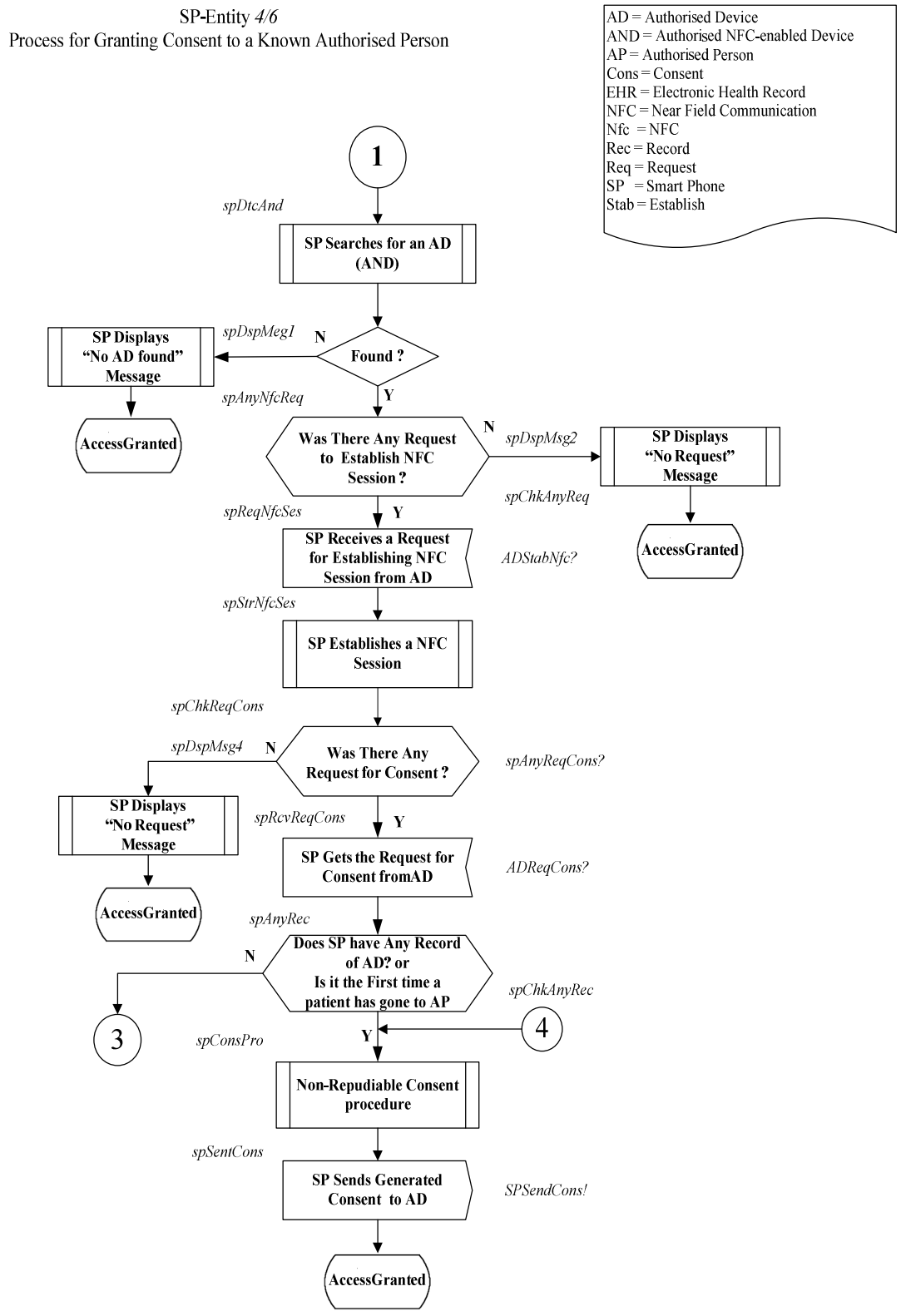


Figure 16. Smart Phone - Process for Granting Consent to a Known Authorised Person

1) The patient selects the *Consent* option from his or her *Smart Phone* menu ($MVCE$ equals 3) (Figure 14).

2) The patient's *Smart Phone* searches for an *Authorised NFC-enabled Device* (AND), which we also call an *Authorised Device* (AD). The AD is a kind of an AND which can communicate with the SP via NFC (Figure 16).

3) If no AD is detected:

   a) The SP displays a message "*No AD found*".

   b) The SP goes to the *AccessGranted* state.

4) Otherwise, if any AD is detected, then the SP checks if there is any request for establishing an NFC session from AD.

5) If there is no request for establishing an NFC session from AD:

   a) The SP displays a message "*No Request*".

   b) The SP goes to the *AccessGranted* state.

6) Otherwise, if there is a request for establishing an NFC session from an AD, then the SP establishes an NFC session.

7) If there is not a request for consent from the AD:

   a) The SP displays the "*No Request*" message.

   b) The SP goes to the *AccessGranted* state.

8) Otherwise, if there is any request for consent, then the SP goes to Step 9.

9) If the SP has an up to date record on the JSC for the AD and AP which shows that ,at least once in the past, the AP and AD's identity has been confirmed by the TTP:

   a) The SP executes *Non- Repudiable Consent* procedure (Section 4.4.6).

   b) The SP sends the *Non- Repudiable* consent to the AD.

   c) The SP returns to the *AccessGranted* state.

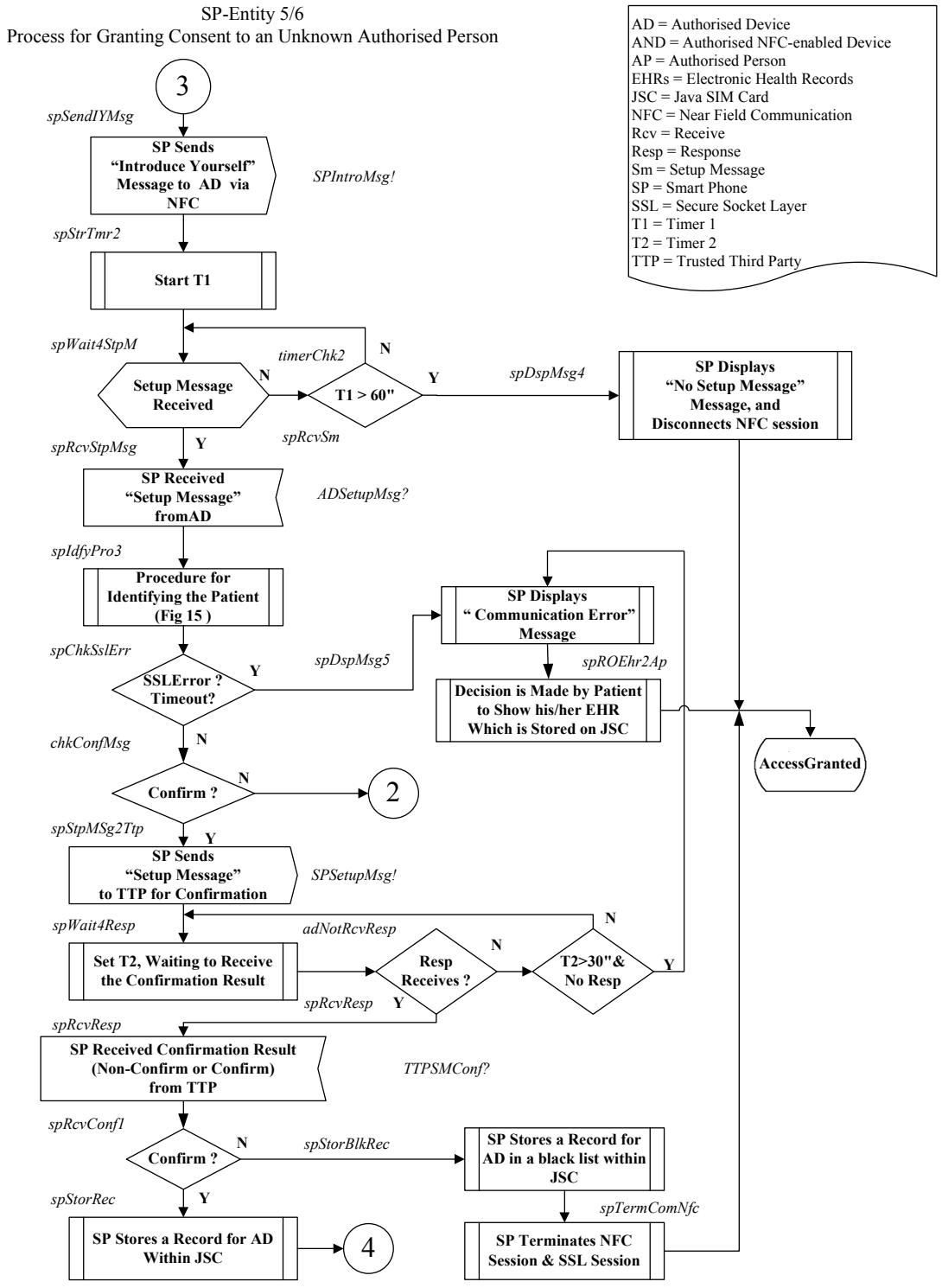


Figure 17. Smart Phone - Process for Granting Consent to an Unknown Authorised Person

10) Otherwise, if there is no record of the AD and AP, then the SP sends the "*Introduce Yourself*" message to the AD via the NFC, as in Figure 17.

11) If the SP does not receive the *Setup Message* from the AD within a certain time:

    a) The SP displays a message "*No Setup Message*", and disconnects the NFC session.

    b) The SP goes to the *AccessGranted* state.

12) Otherwise, if the SP receives the *Setup Message* from the AD, then the SP executes the procedure for *Identifying the Patient* (Figure 15).

13) If the value returned by the procedure for *Identifying the Patient* equals "*SSLError*" or "*Timeout*":

    a) The SP displays "*Communication Error*".

    b) The SP goes to the *AccessGranted* state. In this situation the patient is entitled to decide whether the AP can access his/her EHR which is stored on his/her JSC. In absence of network, the SP and AD log all communications between themselves and once they are connected to the TTP, they send the logged information.

14) Otherwise, if no "*SSLError*" or "*Timeout*" occurred, then the SP checks the value returned by the procedure for *Identifying the Patient*.

15) If the returned value equals "*Non-Confirm*":

    a) The SP blocks access to the AMIM option.

    b) The SP goes to the *Idle* state (the normal behaviour of *Smart Phone*). For unblocking, the patient should contact the *Trusted Third Party*.

16) Otherwise, if the returned value equals "*Confirm"*, then the SP sends "*Setup Message*" (SM) to the TTP for confirming the *Authorised Person* and *Authorised Device* identity. This helps to prevent impersonation.

17) If there is no response (*Confirm or Non-Confirm* for SM) from the TTP within a certain time:

    a) The SP displays "*Communication Error*".

b) The SP goes to the *AccessGranted* state. In this situation the patient is entitled to decide whether the doctor can access to his/her EHRs which is stored on his/her *Java SIM Card*.

18) If the SP receives "*Non-Confirm*" message for the SM from the *Trusted Third Party*:

a) The SP stores a record for the AD and AP in a black list within the JSC.

b) The SP terminates the NFC session and goes to the *AccessGranted* state.

19) Otherwise, if the SP receives a "*Confirm*" message for the SM from the TTP:

a) The SP stores a record for AD within JSC.

b) The SP goes to Step 9a.

4.4.6 PROCEDURE FOR NON-REPUDIABLE CONSENT

As we discussed in Section 4.4.5 the patient's consent stems from the legal and ethical right that entitles the patient to decide who can access his or her medical information and in which level of acceptance. Therefore, patient consent is more than simply getting a patient to give his or her consent. It should be undeniable from the patient especially when it has been given electronically. To put the onus of granting consent on patient we proposed a *Non-Repudiable Consent* procedure which produces a consent by using the SIM's ID, SP's serial No, scanned patient's fingerprint, patient's name and National Health ID, and patient digital signature. The consent is encrypted by the patient's *Private Key*.

4.4.7 PROCESS FOR EMERGENCY SITUATIONS

Smart Phones can save a patient's life in emergency situations, and can be used by those considered to be most at risk such as sufferers of age-related and chronic diseases. In emergency situations *Smart Phones* give users the benefits of instant wireless communication, combined with detecting and informing a patient's location, and automatically calling and sending patient details to an Emergency Room. The details of the three nearest ERs are constantly stored and updated by the SP when a patient changes his or her location. It is especially advantageous when the patient collapses where no one else is around. We assume a patient's SP is always

nearby the patient and the network is available. After the SP senses a strong vibration from the patient falling down it start continuous ringing and vibrating to alert the patient to the fact that his or her phone is going to call the emergency room. If no cancelation action is performed by the patient it automatically calls the ER and sends the location and details of the patient. When the ER gets the information from the SP, it can access the patient's full EHR and can identify where the patient is and what the problem was. Now the ER can send an ambulance to the patient's location with the necessary information including the patient's photo in order to identify the patient, speed up, and improve the quality of treatment. The Emergency procedure is useful for those considered to be most at risk such as elderly people. In the case of a major multiparty accident other factors such as patients becoming separated from their phones are considered out of the scope of this research. As shown in Figure 18, the *Emergency* procedure runs through the following steps:

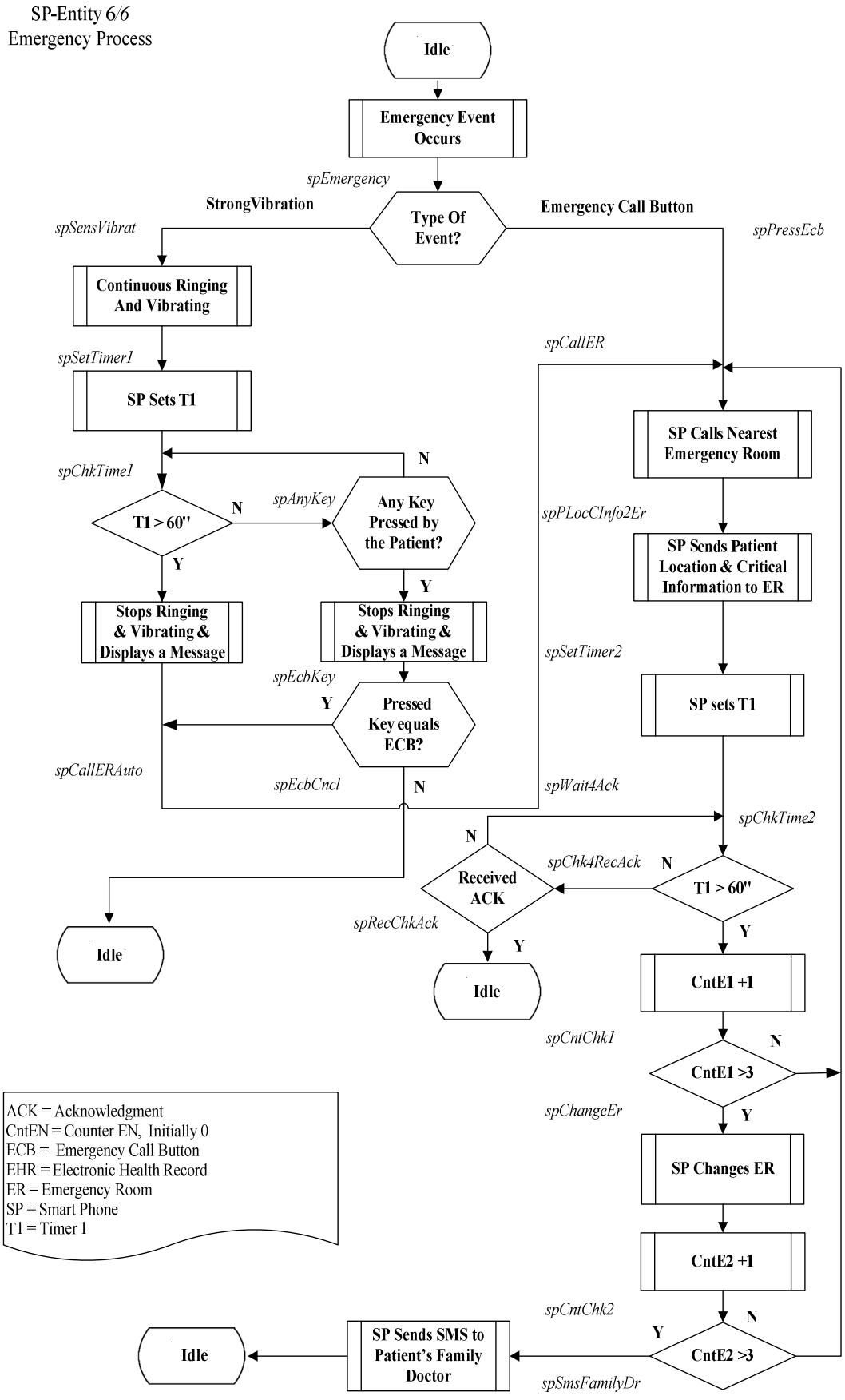


Figure 18. Smart Phone - Process for Emergency Situation

First scenario:

1) Patient presses the *Emergency Call Button* (ECB) for 3 seconds.
2) The SP automatically calls the nearest *Emergency Room* (ER).The SP always has the details of the three nearest ERs, which we call ER1, ER2, and ER3.
3) The SP sends the patient's location and critical information to ER1.
4) If the SP doesn't receive an ACK from ER1for a certain timeout period (*T1*):
    a) It tries again up to a maximum of three attempts.
5) Otherwise, if the SP doesn't receive any ACK form ER1 after three times:
    a) It changes ER1 to ER2.
    b) It goes to Step 2.
6) If the SP doesn't receive any ACK from ER2 after three times:
    a) It changes ER2 to ER3, then It goes to Step 2.
7) Otherwise, if the SP couldn't communicate with any of ER1, ER2 and ER3:
    a) It sends an SMS message to the patient's family doctor (or other emergency contact programmed into the phone), then It goes to the *Idle* state.
8) Otherwise, if the SP gets the ACK, then it goes to the *Idle* state.

Second scenario:

1) The SP senses a strong vibration from the patient falling down.
2) The SP starts continuous ringing and vibrating to alert the patient to the fact that his or her phone is going to call the emergency room.
3) The SP waits for a certain time (*T1*), to decide whether it must call the ER or not.
4) If no key is pressed by the patient within the certain time*:*
    a) The SP stops ringing and vibrating.
    b) The SP goes to the first scenario step 2.
5) Otherwise, if the patient presses any key within a certain time, then the SP stops ringing and vibrating and goes to Step 6.

6) If the pressed key equals to the *Emergency Call Button*, then the SP goes to the first scenario step 2.

7) Otherwise, if the pressed key does not equal the EBC, then the SP goes to the *Idle* state.

4.5 AUTHORISED DEVICE PROTOCOL

Our AD protocol specifies the processes and procedures that must be followed when an *Authorised Person* (AP) such as doctor wants to request data from, respond to or communicate with patient's *Smart Phone* (SP) or *Trusted Third Party* (TTP). The protocol must work in diverse scenarios. In total the AD protocol includes seven major processes and procedures, each of which is modeled by following a particular path through the protocols presented below. These processes and procedures include *Authenticating an Authorised Person, Identifying an Authorised Person, Getting Consent, Generating a Referral Letter*, *Non-Repudiable Setup Message*, *NFC Communication,* and *Verifying a Referral letter.* They will be explored in detail in the following sections. The entire AD protocol is described by the SDL diagrams in Figures 19 to 25.

In this section we assume the *Authorised Person* has an *Authorised Device* such as a *Smart Phone* or Laptop which has a fingerprint scanner, a *Java SIM Card* containing NFC applications, the AP's fingerprint template, a PKI handler, the AP's *Private Key*, a *Next Generation Network* interface, and an *Identification Algorithm* (IA). The AD is also able to connect to the web using the SSL/TLS protocol, communicate with the *Java SIM Card*, produce a digital signature, set a timeout, and communicate with other NFC-enable devices.

4.5.1 PROCESS FOR AUTHENTICATING AN AUTHORISED PERSON

Authorised Person authentication is the first major step towards making the AP accountable for accessing a patient's *Electronic Health Records*. As shown in Figure 19, for authenticating the AP we require two methods, a *Personal Identification Number* (PIN) and fingerprint (Section 4.3). After a doctor selects *Access to Medical Information Menu* (AMIM) option contained in the menu displayed on an *Authorised*

Device (AD), the AD asks the AP (such as a doctor) to enter the PIN. If the PIN is entered incorrectly three times, the AD blocks access to the AMIM option and the AP has to contact the TTP for unblocking. If the PIN code is entered correctly, the AD asks the AP to put his/her fingerprint on the *Authorised Device Scanner* (ADS) in order to verify it with the fingerprint template stored in the AD. If the *Authorised Person Fingerprint* (APF) is read incorrectly three times, the AD blocks access to the AMIM option and the AP has to contact the TTP for unblocking. If the APF and the fingerprint template match, then the AD goes to the *GetAccess* state (Figure 19).

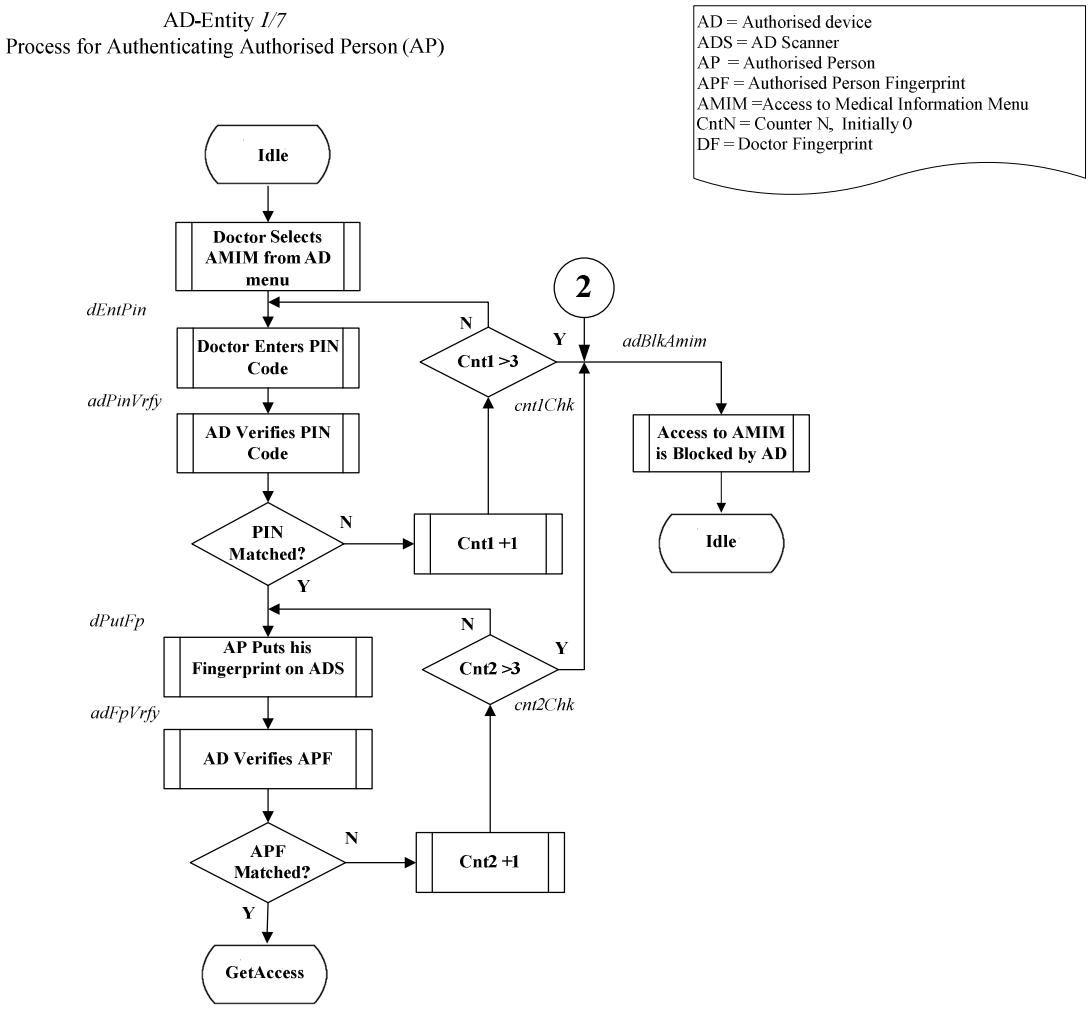


Figure 19. Authorised Device - Process for Authenticating Authorised Person (AP)

4.5.2 PROCEDURE FOR IDENTIFYING AN AUTHORISED PERSON

The ability to accurately identify *Authorised Devices* and *Persons* interacting with the *Trusted Third Party* (TTP) is also vital for ubiquitous access to EHRs. The *Authorised Device* protocol is the same as the *Smart Phone*'s protocol and uses the *Challenge-Response* mechanism for identification (see Section 4.2 for more detail). In the AD protocol the *Challenge-Response* is configured between the *Authorised Device* and *Trusted Third Party* protocols. The identification procedure is utilised by the AD specifically when an AP wants to request consent or verify a referral letter. Each of these procedures will be described in detail later in this chapter.

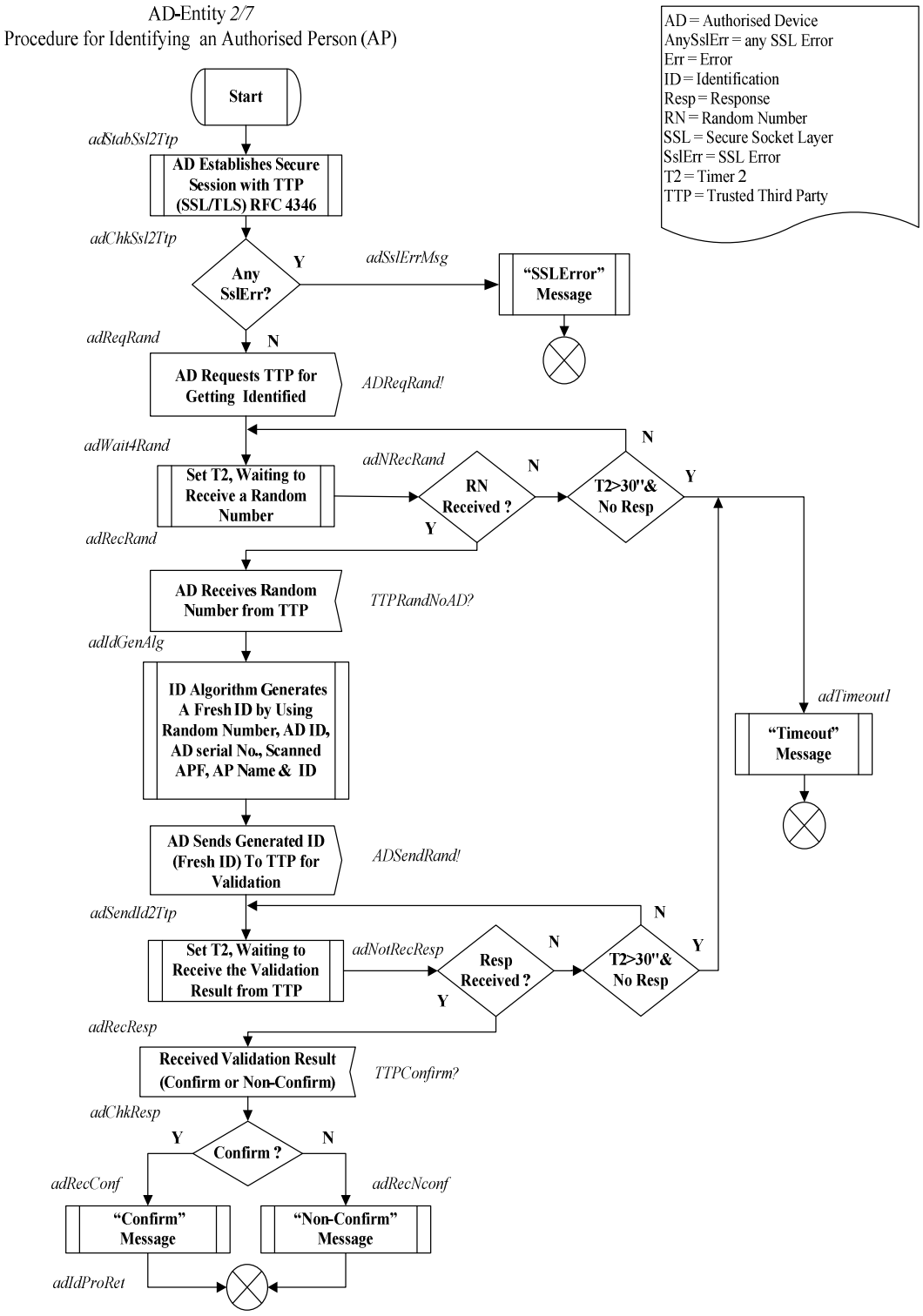


Figure 20. Authorised Device - Procedure for Identifying an Authorised Person

As shown in Figure 20, the procedure for *Identifying the Authorised Person* consists of the following steps:

1) The AD establishes a SSL/TLS session with the TTP.

2) If any SSL/TLS error (IETF, 2008) occurs, the AD returns from *Identifying the Authorised Person* procedure with an "*SSLError*" value.

3) Otherwise, if no SSL/TLS error occurs, then the AD requests the TTP to send a *Random Number* and informs the TTP "I need to be identified". The *Random Number,* in addition to the other information, is needed by the *Identification Algorithm* within the AD in order to generate a *Fresh ID*. The AD sends the *Fresh ID* to the TTP for getting identified.

4) If there is no response (*Random Number*) from the TTP within a certain time, the AD returns from the *Identifying the Authorised Person* procedure with a "*Timeout*" value.

5) Otherwise, if the AD receives a *Random Number* from the TTP, the AD goes to Step 6.

6) The AD generates a *Fresh ID,* sends it to the TTP for validation and waits for a response *(Confirm, Non-Confirm*).

7) If there is no response (*Confirm, Non-Confirm*) from the TTP within a certain time, then the AD returns from the *Identifying the Authorised Person* procedure with the "*Timeout*" value.

8) Otherwise, if the AD receives the response (*Confirm, Non-Confirm*) from the TTP within a certain time, it goes to Step 9.

9) If the AD receives the "*Confirm"* message from the TTP (if the TTP's generated ID equals the AD's generated ID), then the AD returns from *Identifying the Authorised Person* procedure with a "*Confirm"* message value.

10) Otherwise, if the AD receives the "*Non-Confirm"* message from the TTP (if the TTP's generated ID does not equal to the AD's generated ID), then the AD returns from the *Identifying the Authorised Person* procedure with a "*Non-Confirm*" value.

4.5.3 PROCESS FOR GETTING CONSENT

Obtaining patient consent before accessing or sharing medical information is the first phase towards patient privacy. In our protocols, the initial step of getting the patient's consent begins with the contact between the AD and the SP through

the NFC. As shown in Figures 21 to 23 the Consent Getting Procedure is used when an Authorised Person wants to get the patient's consent and the AP needs to see or update the patient's medical information during their examination. The communication between the patient's device and the AP's device is made by NFC technology. The main point behind this procedure is that the level of access to the patient's medical information is determined by the TTP based on the patient's consent and the doctor's identification.

Once the TTP receives the consent, first, it verifies if the consent comes from the real patient, second, it determines the *Access Level* (AL) for accessing the patient's EHR. The goal is to provide the *Authorised Persons* with a reasonable level of access to a patient's medical information. The AL is defined based on the *Discretionary Access Control* (DAC) and *Mandatory Access Control* (MAC), *Role Based Access Control* (RBAC), and the identification of the AD. The TTP makes *Discretionary Access Control* (DAC) and *Mandatory Access Control* (MAC) lists from the patient's consent. The RBAC list is defined by healthcare authorities.

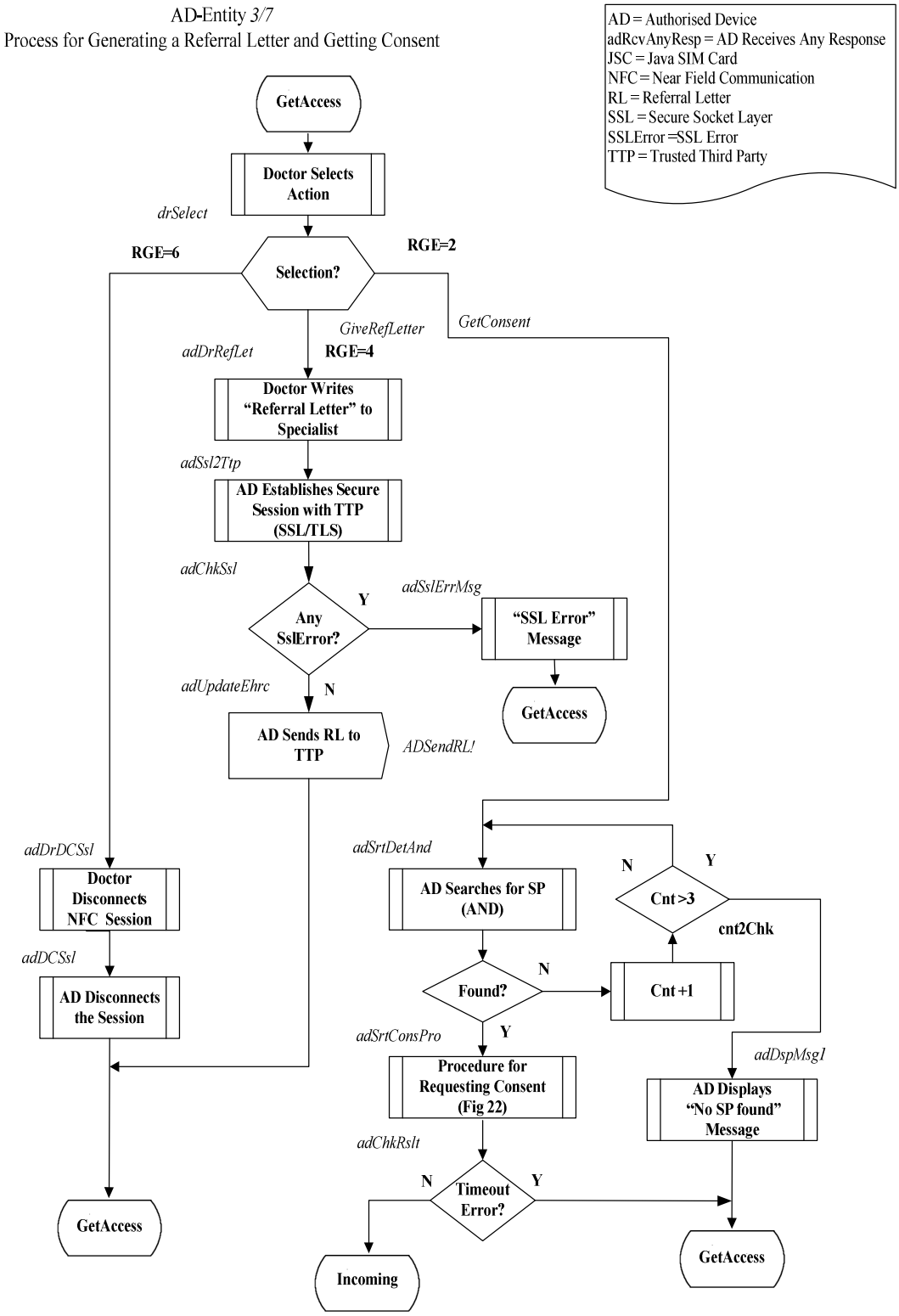


Figure 21. Authorised Device - Process for Generating a Referral Letter and Getting Consent

As shown in Figures 21 to 23, the *Getting Consent* process is encapsulated in the following steps:

Figure 21:

1) A doctor selects the *GetConsent* option from his or her device's menu ($RGE = 2$).
2) The AD searches for an *Authorised NFC-enabled Device* (AND) such as a patient's device (*Smart Phone)* which wants to communicate via NFC and tries up to a maximum of three attempts. In the rest of this process the SP represents the AND.
3) If no SP is detected:
   a) The AD displays a message "*No SP found*".
   b) The AD returns to the *GetAccess* state.
4) Otherwise, If the SP is detected:

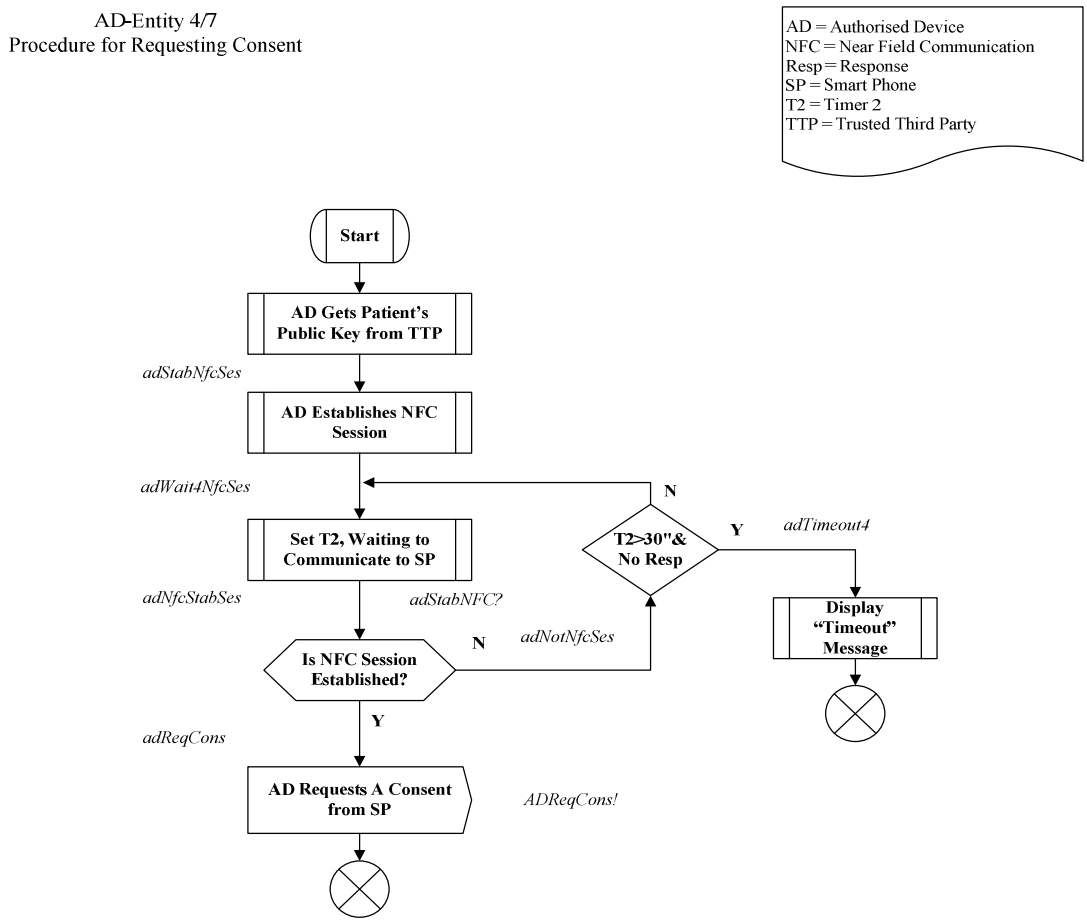


Figure 22. Authorised Device - Procedure for Requesting Consent

Figure 22:

a) The AD executes the procedure for *Requesting Consent* as follows.

   i) The AD gets the patient's *Public Key* from the TTP.

   ii) The AD establishes a NFC session with the SP.

   iii) If there is no response (when establishing the NFC session) from the SP, within a certain time, then the AD returns from the *Requesting Consent* procedure with a "*Timeout*" value.

   iv) Otherwise, if the AD receives a response (establishing a NFC session) from the SP:

      (1) The AD sends a request for consent to the SP.

      (2) The AD returns from the *Requesting Consent* procedure with a "*No Error*" value.

Figure 21:

5) If the value returned by the *Requesting Consent* procedure is a "*Timeout*" error, the AD goes to the *GetAccess* state,

6) Otherwise, if value returned by the *Requesting Consent* procedure is "*No Error*", the AD goes to the *Incoming* state.

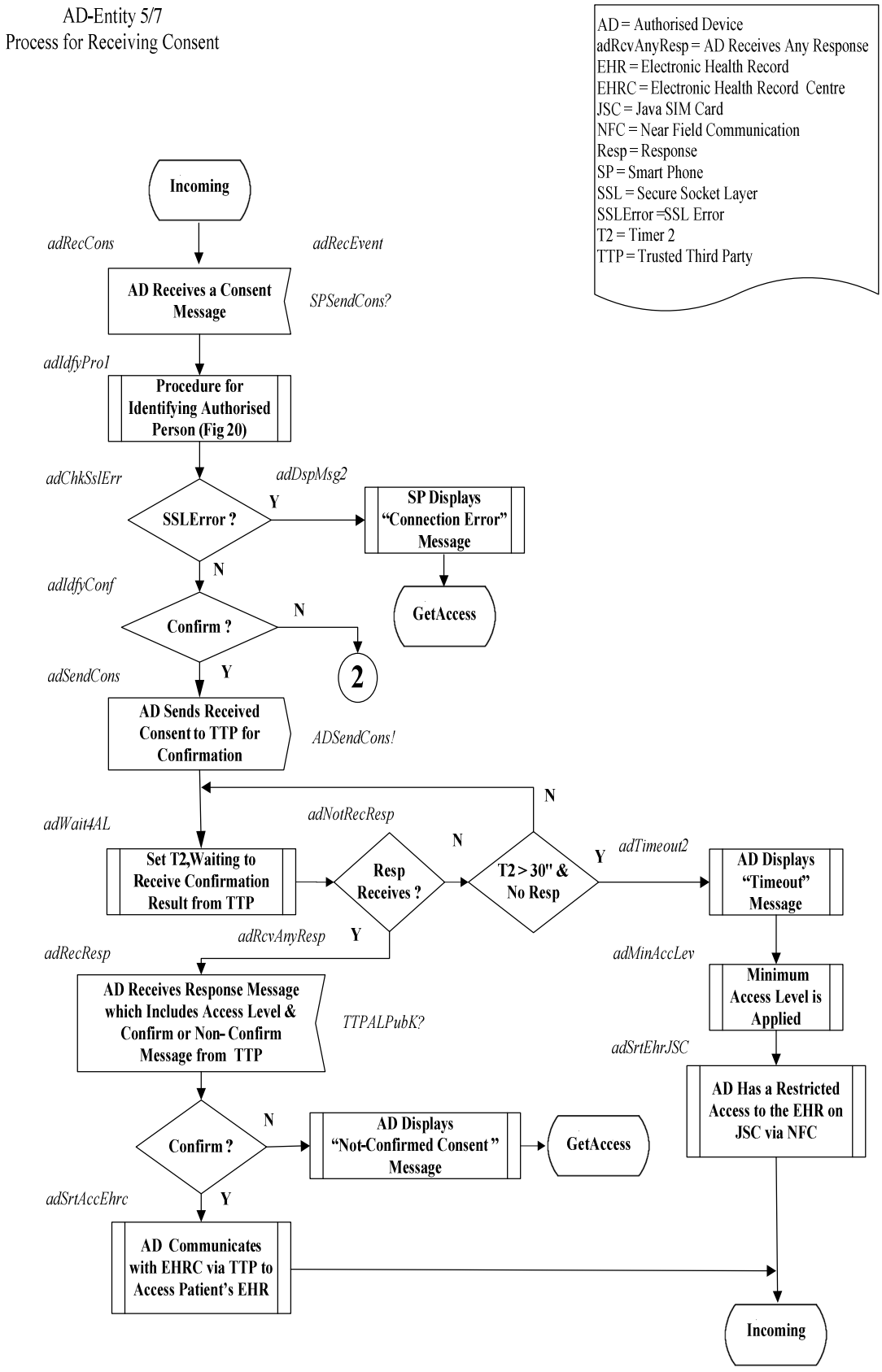


Figure 23. Authorised Device - Process for Receiving Consent

Figure 23:

7) If the AD receives a *consent message* from the SP, the AD goes to Step 8.

8) The AD executes the procedure for *Identifying the Authorised Person* (Figure 20).

9) If the *Identifying the Authorised Person* procedure returns any error related to *SSL/TLS* communication or *Timeout* (these errors explained in detail in Section 4.5.2):

   a) The AD displays the message "*Communication Error*".

   b) The AD goes to the *GetAccess* state.

10) Otherwise, if the *Identifying the Authorised Person* procedure returns no error (*SSL/TLS* or *Timeout*), the AD goes to Step 11.

11) If the value returned by the *Identifying the Authorised Person* procedure is a "*Non-Confirm"* message:

   a) The AD Blocks access to the AMIM option.

   b) The AD goes to the *Idle* state (the normal behaviour of the device). For unblocking, the *Authorised Person* should contact the TTP.

12) Otherwise, if the value returned by the *Identifying the Authorised Person* procedure is a "*Confirm*" message, the AD sends the received consent to the TTP and waits for a response (the AD goes to Step 13).

13) If the AD does not receive a response (Consent: "*Confirm" or "Non-Confirm"*) from the TTP within a certain time:

   a) The AD displays the message "*Timeout*".

   b) Access to the EHR is given with a minimum access level which restricts the AD to have access only to the EHR contained within the JSC.

   c) The AD goes to the *Incoming* state.

14) Otherwise, if the AD receives a response (Consent: "*Confirm" or "Non-Confirm"*) from the TTP within a certain time: the AD goes to Step 15.

15) If the response i*s* a *"Confirm*", the TTP connects (redirects) the AD to the EHRC site for accessing the patient medical records.

16) If the response is a *"Non-Confirm*":

a) The AD displays a "*Not-Confirmed Consent*" message.

b) The AD goes to the *GetAccess* state.

4.5.4 PROCESS FOR GENERATING A REFERRAL LETTER

Referral letters are a flexible means of transferring information between health care professionals in order to provide proper patient care. The *Referral Letter* (RL) is usually written following a consultation between the patient and a *General Practitioner* (GP). A typed referral letter sent by conventional postal services or an internal hospital mailing system is the traditional method of conveying information from the referring doctor to the receiving doctor. It may be that, with emerging technologies, other forms of communication will become acceptable, e.g. fax and electronic mail. Using electronic communication, there may be no requirement for information entered by primary care staff to be re-entered by staff in the secondary care setting. This is the principle of single data entry (SIGN, 2002). In our approach which is described in Chapter 3, we use emerging technologies to develop a secure and electronic communication between a patient and health care providers. In terms of transferring a RL, we use a central method in which the transfer of RLs is done by the TTP. It means a GP sends a RL to the TTP and the specialist can get it from the TTP as well. As shown in Figure 21, the following steps are introduced in the *Giving Referral Letter* process:

1) A doctor writes a "*Referral Letter*" for a patient.

2) The AD establishes a secure session with the TTP (via SSL/TLS) in order to send the RL.

3) If any SSL error occurs:

a) The AD displays a message "*SSL Error*".

b) The AD goes to the *GetAccess* state.

4) Otherwise, If no SSL error occurs:

a) The AD sends the RL to the TTP.

b) The AD goes to the *GetAccess* state.

4.5.5 PROCESS FOR NON-REPUDIABLE SETUP MESSAGE

As we mentioned in Section 4.4.5, the process for Granting Consent, when there is no record in the *Java SIM Card* for the *Authorised NFC-enabled Device* or *Authorised Person* who wants to access the patient's EHR it means this is the first time that the patient has gone to the AP, and the patient needs the AP to be identified by the TTP. Therefore, the patient's SP sends an "*Introduce Yourself*" message to the *Authorised Device* to request it to introduce itself.

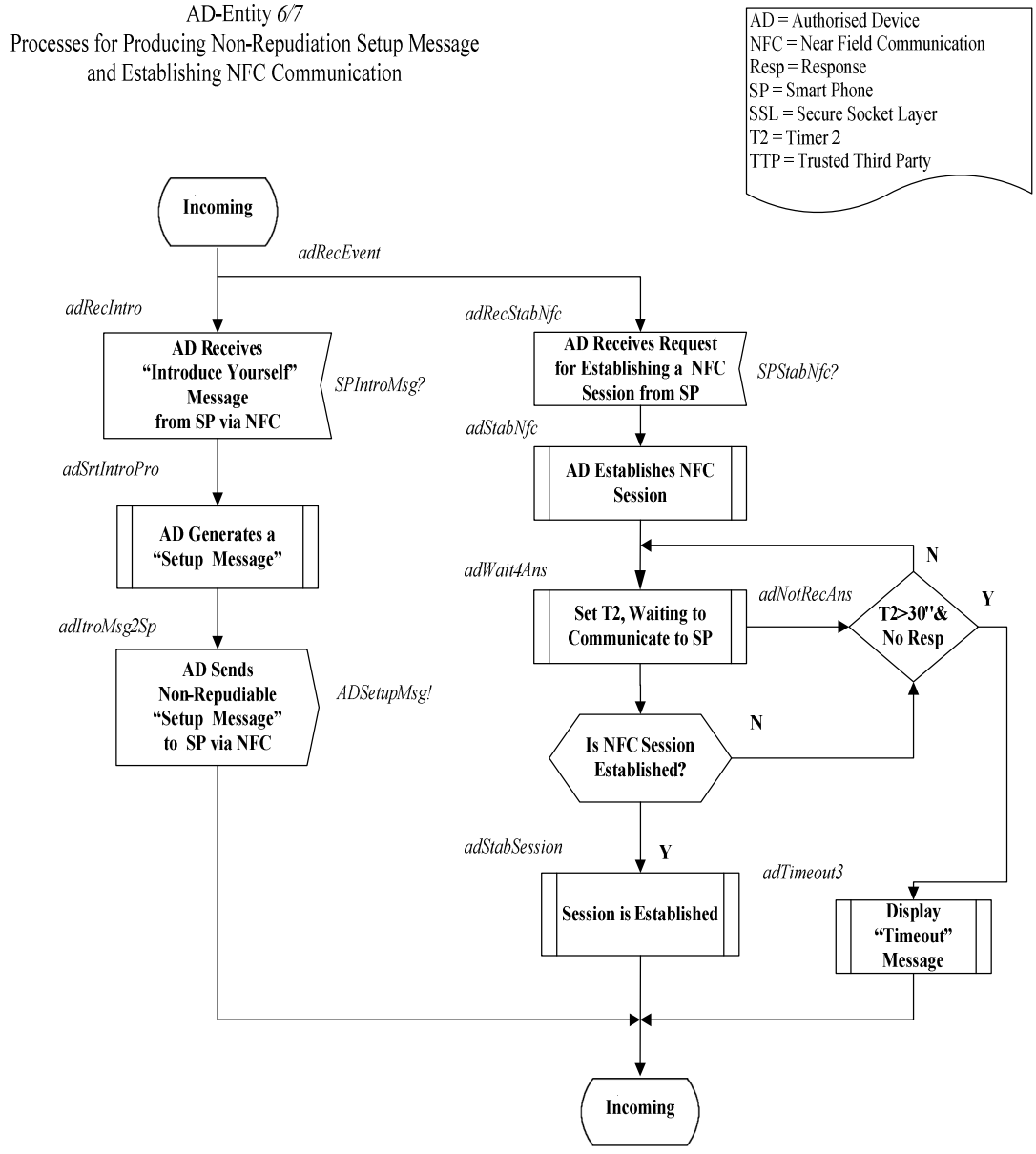


Figure 24. Authorised Device - Processes for Non- Repudiable Setup Message and Establishing NFC Communication

As shown in, the following steps are required for the *Non- Repudiable Setup Message* process:

1) The AD receives an "*Introduce Yourself*" message from the SP.

2) The AD generates a *Non-Repudiable Setup Message*. The *Setup Message* (SM) consists of the AD and AP's details, scanned AP's fingerprint, time, date and the AP's digital signature. The *Setup Message* must be encrypted by the *Authorised Person*'s *Private Key*.

3) The AD sends the generated *Non-Repudiable Setup Message* to the SP via NFC, and goes to the *Incoming* state

4.5.6 PROCESS FOR NFC COMMUNICATION

The NFC is used by the AD protocol to facilitate contactless communication between the *Authorised Device* and the *Smart Phone* when the *Authorised Person* needs to communicate with the SP or have access to medical information within the *Java SIM Card*. As shown in

Figure 24, the *NFC Communication* process is outlined in the following steps:

1) The AD receives a request for establishing a NFC session from the SP.

2) The AD establishes the NFC Session.

3) If there is no response (to establish the NFC Session) from the SP within a certain time:

   a) The AD displays a "*Timeout*" message.

   b) The AD goes to the *Incoming* state.

4) Otherwise, If the AD receives a response (to establish the NFC Session) from the SP:

   a) The AD establishes a NFC session.

   b) The AD goes to the *Incoming* state.

4.5.7 PROCESS FOR VERIFYING A REFERRAL LETTER

As we described in Section 4.5.4, *Process for Referral Letter*, the specialist can get a *Referral Letter* from the TTP based on the patient's details.

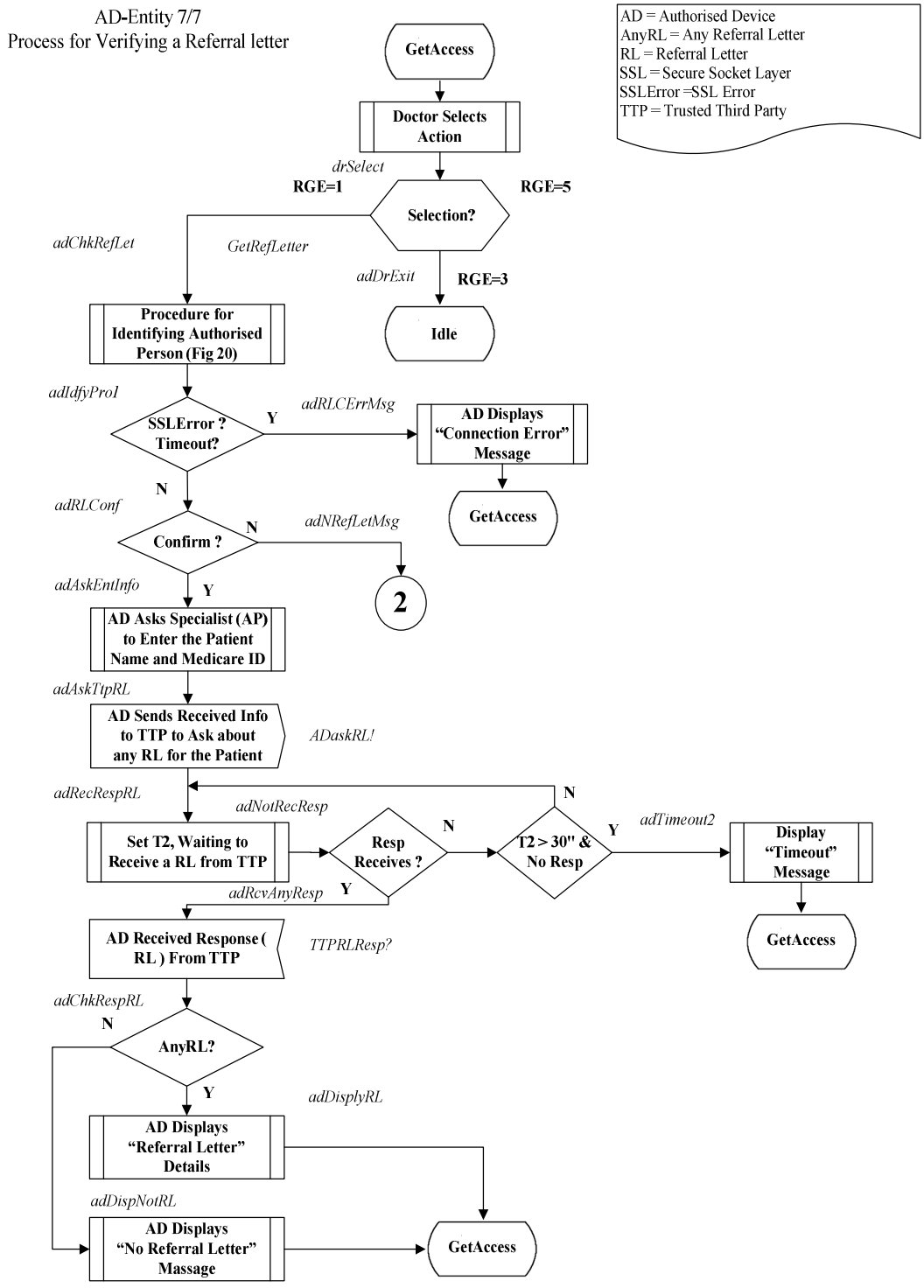


Figure 25. Authorised Device - Process for Verifying a Referral Letter

As shown in Figure 25, the process for *Verifying* a *Referral Letter* goes through the following steps:

1) The AP selects the "*Check RL*" option from his or her AD's menu.
2) The AD executes the procedure for *Identifying the Authorised Person* (Figure 20).
3) If the *Identifying the Authorised Person* procedure returns any error related to *SSL/TLS* communication or *Timeout* (Section 4.5.2):
   a) The AD displays the message "*Communication Error*", then it
   b) The AD goes to the *GetAccess* state.
4) Otherwise, if there is no error (*SSL/TLS* or *Timeout*), then the AD checks the value returned by the *Identifying the Authorised Person* procedure, Step 5.
5) If the value returned by the *Identifying the Authorised Person* procedure is a "*Non-Confirm"* message:
   a) The AD Blocks access to the AMIM option.
   b) The AD goes to the *Idle* state (the normal behaviour of device). For unblocking, the *Authorised Person* should contact the TTP.
6) Otherwise, if the value returned by the *Identifying the Authorised Person* procedure is a "*Confirm*" message, the AD asks the AP to enter the patient's name and Medicare ID.
7) The AD sends the entered information to the TTP and asks if there is any *Referral Letter* for the patient.
8) If there is no response (*Referral Letter*) from the TTP within a certain time :
   a) The AD displays the message "*Timeout*", then it goes to the *GetAccess* state.
9) Otherwise, if the AD receives a response (*Referral Letter*) from the TTP
   a) The AD displays the RL, then it goes to the *GetAccess* state.
10) Otherwise, If there is no RL related to the patient in the response:
   a) The AD displays the message "*No RL Found*".
   b) The AD goes to the *GetAccess* state.

4.6 TRUSTED THIRD PARTY PROTOCOL

Our Trusted Third Party (TTP) protocol specifies the processes and procedures that must be followed when the TTP wants to request data from, respond to or communicate with the *Smart Phone* (SP) and *Authorised Device* (AD). The protocol must work in different emergency and non emergency scenarios. In total the TTP protocol includes five major processes and procedures, each of which is modelled by following a particular path through the protocols presented below. These processes include *Process for Establishing an SSL/TLS Session, Process for Verifying a Setup Message, Process for Verifying Patient Consent*, *Process for Identifying the SP or AD,* and *Process for Storing or Retrieving a Referral Letter.* They are explored in detail in the following sections. The entire TTP protocol is described by the SDL diagrams in Figures 26 to 29.

In this section we assume the TTP is a powerful server which is able to manage very large amounts data and traffic. Also we assume it is facilitated with auditing, logging, authorising, identifying, verifying, and storage capabilities. Moreover, it must be able to determine an *Access Level* (AL) for the patient's EHR based on verifying the identity of the *Authorised Device* and *Authorised Person* who wants to have access to a patient's medical information and the *Role Based Access Control, Discretionary Access Control* and *Mandatory Access Control* settings (Kim et al., 2006). In addition, it must be able to Store-Retrieve *Referral Letter* (RL) from a database and handle the PKI authentication process. Furthermore,, the TTP must be capable to enable and disable access to a patient's Electronic Health Record and redirect (connect) the SP or AD to the *Electronic Health Records Centre* (EHRC) server site after authentication. Finally we assume the TTP is connected to a database which contains all *Authorised Persons'* and patients' details that include their SIM ID, device's serial number, fingerprint template, name and national ID. All these details are used by the *Identification Algorithm* (IA) in order to identify the SP and AD. The TTP also must be equipped with an SSL/TLS protocol for establishing a secure communication.

4.6.1 PROCESS FOR ESTABLISHING AN SSL/TLS SESSION

Establishing an SSL/TLS session for secure communication between an SP or AD and a TTP is the major part of our protocol. SSL/TLS is a cryptographic protocol that is based on the principle of PKI, and uses digital certificates to provide secure services such as confidentiality, data integrity and identity authentication for communications over the Internet. Because many applications are now based on the structure of a *Browser* and *Server*, SSL is in wide spread use, and provides security using the HTTPS protocol (Huawei et al., 2009).

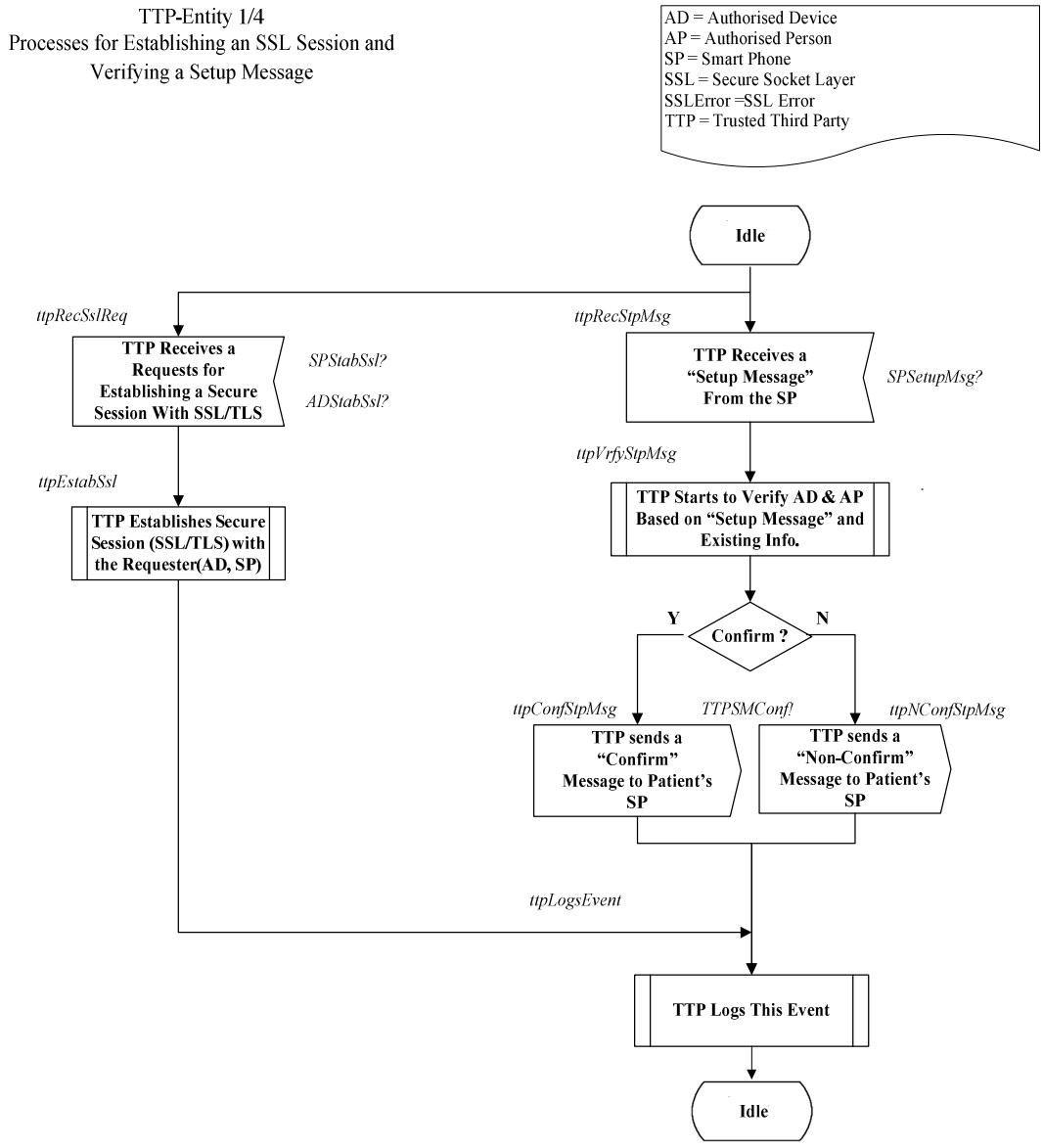


Figure 26. Trusted Third Party - Processes for Establishing an SSL/TLS Session and Verifying a Setup Message

As shown in Figure 26, the *Establishing SSL/TLS an Session* process is going through the following steps:

1) The TTP receives a request for establishing a secure session with SSL/TLS from the SP or AD.
2) The TTP establishes a secure session (SSL/TLS) with the requester.
3) If any SSL error occurs:
    a) The TTP displays the message "*SSL Error*".
    b) The TTP goes to the *Idle* state.
4) Otherwise, If no SSL error occurs:
    a) The TTP starts to communicate.
    b) The TTP logs this event.
    c) After finishing the SSL/TLS session it goes to the *Idle* state.

4.6.2 PROCESS FOR VERIFYING A SETUP MESSAGE

As we mentioned in Section 4.4.5, if there is not any record in the SP for the AD and AP, it means this is the first time that the patient has gone to the AP, so that the AP must be authorised by the TTP. To do this, the SP sends an "*Introduce Yourself*" (IY) message to the AD via the NFC, as shown in Figure 17, and asks the AD to introduce itself by sending its information which includes the *Authorised Device*'s details, the AP's name, and ID, and AP's digital signature. Once the AD receives the IY it replies to the SP with the *Setup Message* (SM) which includes all requested information. Then the SP sends the SM to the TTP for verifying. The TTP verifies the originality of the SM based on the information such as the AD and AP's details, which already exists in the TTP's database and responds to the SP. As shown in Figure 26, the *Verifying Setup Message* process is outlined in the following steps:

1) The TTP receives a "*Setup Message*" from the SP.
2) The TTP verifies the originality of "*Setup Message*".
3) If the TTP confirms the AP and AD based on the SM and existing information:
    a) It sends a "*Confirm*" message to the SP, and goes to Step 5.

4) Otherwise, If the TTP does not confirms the AP and AD based on the SM and existing information:

   a) It sends a "*Non-Confirm*" message to the SP, and goes to Step 5.

5) The TTP logs this event, and goes to the *Idle* state.

4.6.3 PROCESS FOR VERIFYING PATIENT CONSENT

As we mentioned in Section 4.5.3, Figure 23, the level of access (AL) to the patient's medical information is determined by the TTP based on the patient's consent, and the AD's and AP's identities. To do this, the AD sends the received consent to the TTP in order to, first, get the consent verified and, second, get the level of access to the patient's EHR. The TTP verifies the originality of the patient's consent based on the information, such as the patient and SP's details, which already exists in the TTP's database. As shown in Figure 27, the *Verifying Patient Consent* process is summarised in the following steps:

1) The TTP receives a "*Patient's Consent*" message from the AD.

2) The TTP verifies the patient's consent originality based on the existing information such as a patient's and SP's details, patient's *Public Key*, etc.

3) If the TTP does not confirms the "*Patient's Consent*":

   a) It sends a "*Non-Confirm"* message to the AD.

   b) It logs this event, goes to the *Idle* state.

4) Otherwise, if the TTP confirms the "*Patient's Consent*",

   a) The TTP determines an "*Access Level*" based on the AP's and AD's identities, the patient's consent (DAC, MAC), and RBAC.

   b) The TTP sends the AL to the AD.

   c) The TTP connects the AD to the EHRC.

   d) The TTP logs this event, and goes to the *Idle* state.

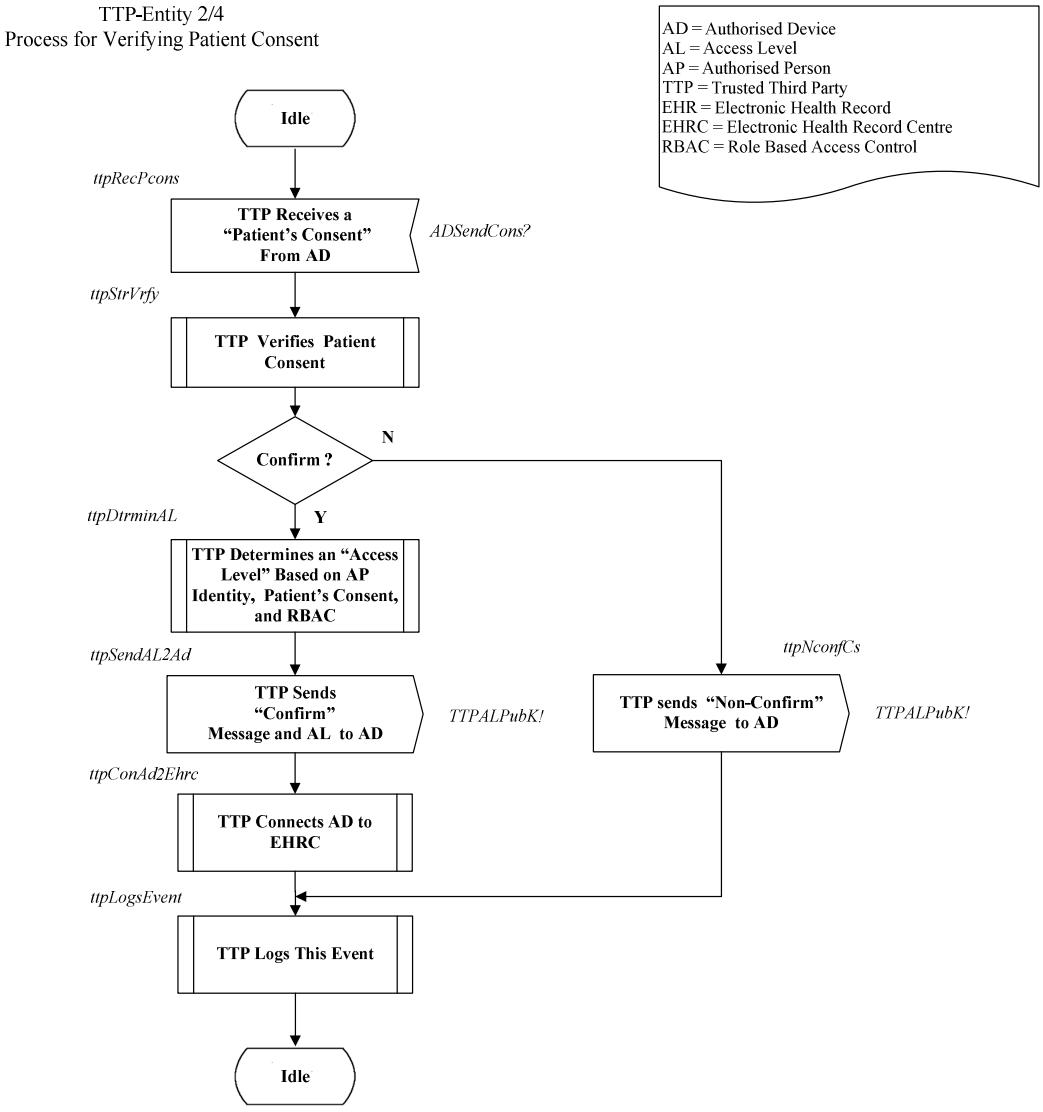


Figure 27. Trusted Third Party - Process for Verifying Patient Consent

4.6.4 PROCESS FOR IDENTIFYING THE SP OR AD

As we described in Sections 4.6.4 the ability to accurately identify the SP and AD which are interacting with the TTP is crucial for implementing ubiquitous access to EHRs. Also, as we pointed out, our protocols use a basic *Challenge-Response* mechanism to identify the entities (Section 4.2). The *Challenge-Response* protocol is configured between the TTP and the other entities such as the AD and SP.

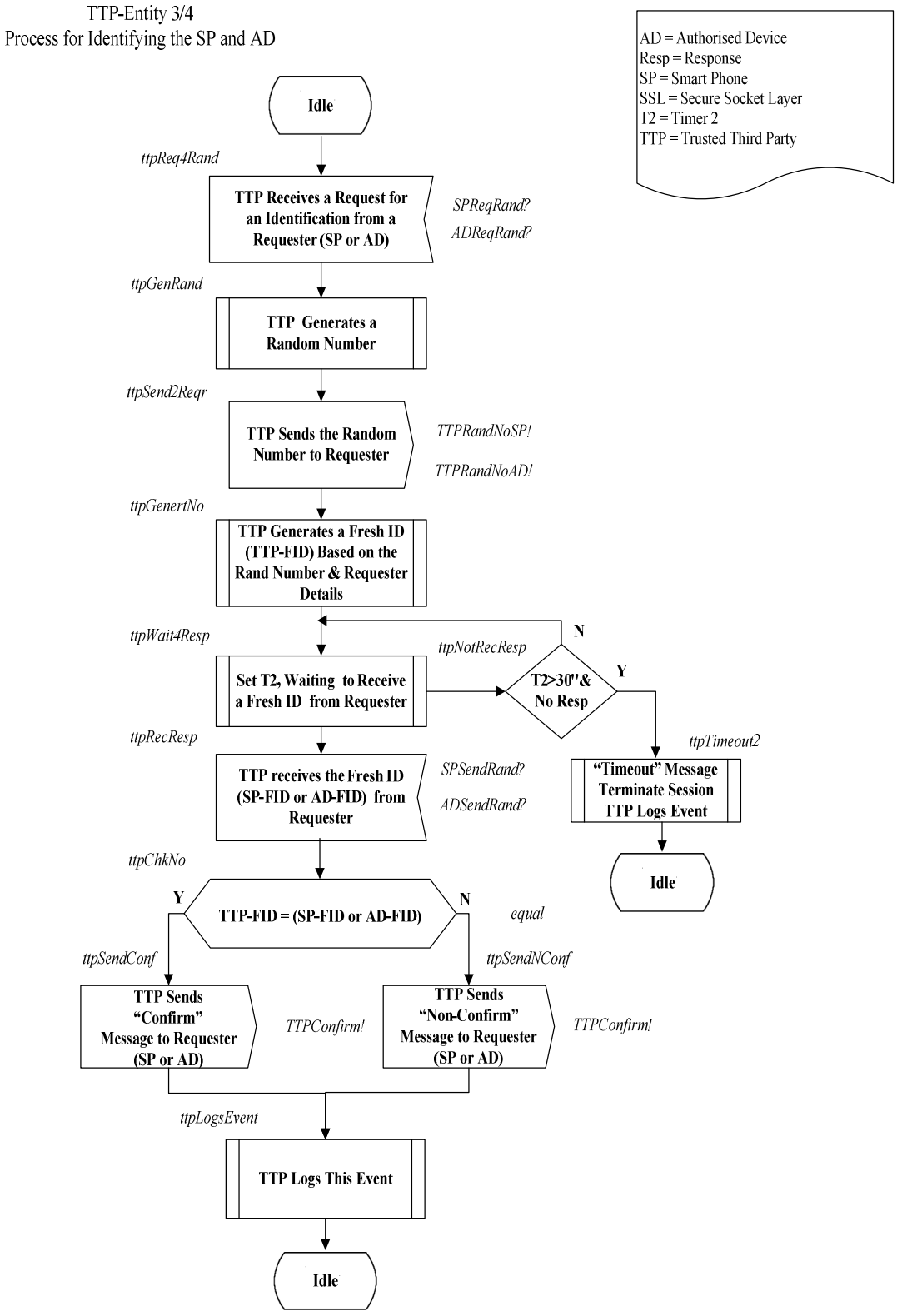


Figure 28. Trusted Third Party - Process for Identifying the SP and AD

As shown in Figure 28, the process for *Identifying the SP and AD* is summed up in the following steps:

1) The TTP receives a request for identification from the SP or the AD.
2) The TTP generates a *Random Number* and sends it to the requester (SP or AD).
3) The TTP generates a *Fresh ID* (TTP-FID) based on the *Rand Number* and the requester's details which already are in the TTP's database.
4) The TTP waits to receive a *Fresh ID* (SP-FID or AD-FID) from the requester within a certain time.
5) If there is no response (SP-FID or AD-FID) from the requester within the certain time:
   a) The TTP displays a "*Timeout*" message, terminates the session, and logs the event.
   b) The TTP goes to the *Idle* state.
6) Otherwise, if the TTP receives a response (SP-FID or AD-FID) which is generated by the requester, it goes to Step 7.
7) If the received ID (SP-FID or AD-FID) equals the generated ID (TTP-FID):
   a) The TTP sends a "*Confirm"* message to the requester (SP or AD).
   b) The TTP goes to Step 9.
8) Otherwise, if the received ID (SP-FID or AD-FID) is not equal to the generated ID (TTP-FID):
   a) The TTP sends a "*Non-Confirm"* message to the requester (SP or AD) and goes to Step 9.
9) The TTP logs this event, and goes to the *Idle* state.

4.6.5 PROCESS FOR STORING OR RETRIEVING A REFERRAL LETTER

As we discussed in Sections 4.6.5, we develop a secure and electronic communication between a patient and health care providers for transferring a *Referral Letter* (RL). In this approach, we use a central method in which all the transferring and

receiving of RLs is done by the TTP. This means a GP sends an RL to the TTP and the specialist can get it from the TTP as well.

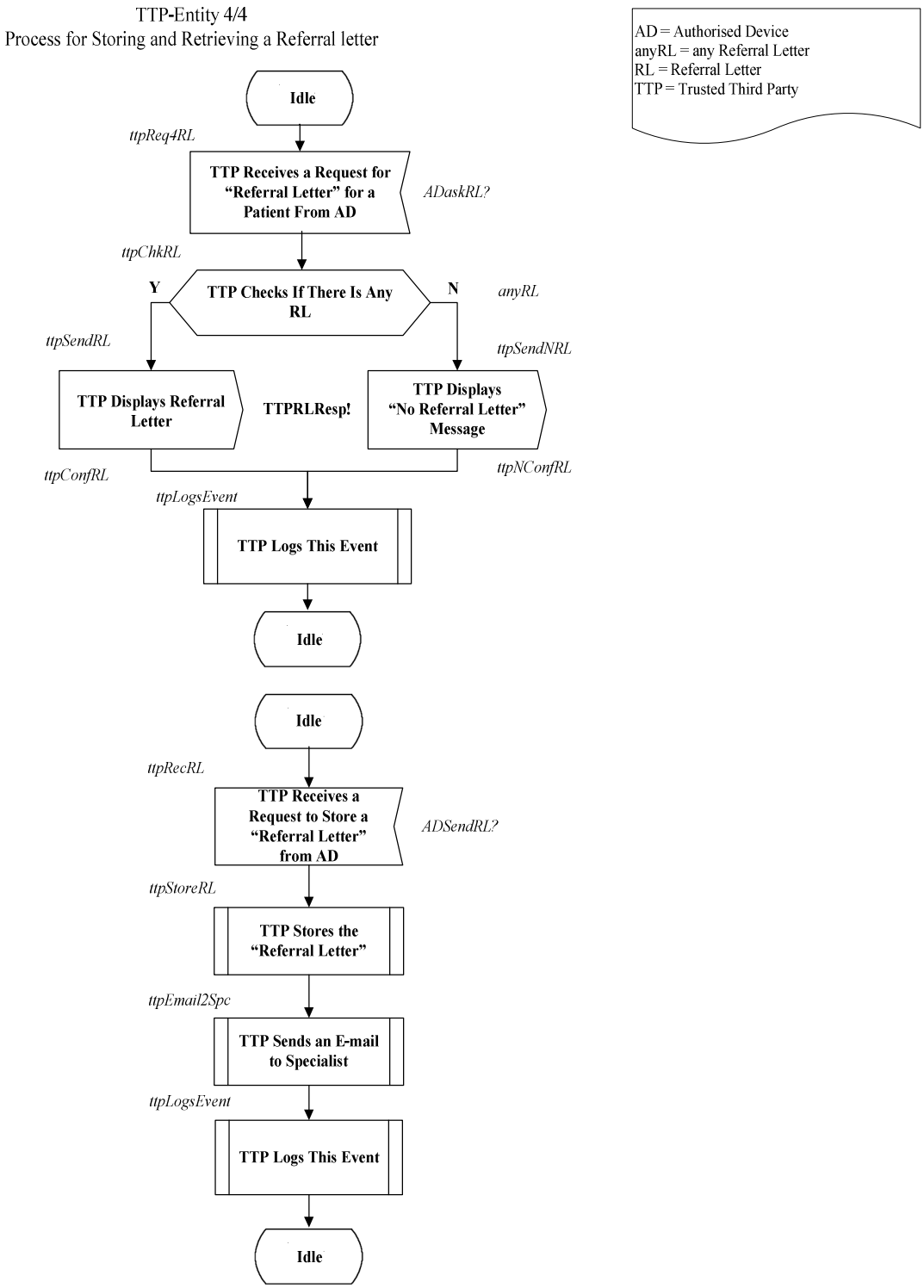


Figure 29. Trusted Third Party - Process for Storing and Retrieving a Referral Letter

As shown in Figure 29, the *Storing and Retrieving Referral Letter* process goes through the following steps:

Storing a Referral letter:

1) The TTP receives a request for storing a "*Referral Letter*" from the AD.
2) The TTP stores the *Referral Letter* in a database.
3) The TTP sends to the relevant specialist an e-mail which includes the RL and all the information needed.
4) The TTP logs this event, and goes to the *Idle* state.

Retrieving a Referral letter:

1) The TTP receives a request for any existing *Referral Letter* related to a patient from the AD.
2) If the TTP finds *Referral Letter* :
   a) The TTP displays the *Referral Letter*, and goes to Step 4.
3) Otherwise, if the TTP finds no *Referral Letter:*
   a) TTP displays a massage "*No Referral Letter*", and goes to Step 4.
4) The TTP logs this event, and goes to the *Idle* state.

4.7 SUMMARY

In this chapter we used the *Specification and Description Language* (SDL) to describe in detail the three essential protocols needed to support our framework for ubiquitous access to *Electronic Health Records*, the *Smart Phone* (SP), the *Authorised Device* (AD), and the *Trusted Third Party* (TTP). All those protocols work on the application layer which means they are independent of the underlying protocol layers for end-to-end communication. The SP, AD, and TTP protocols define the processes and procedures that must be followed when patients and health care providers want to communicate with each other in order to provide ubiquitous access to the patient's *Electronic Health Records* using a *Java SIM Card*. The SP, AD, and TTP protocols work on behalf of a patient's device, a doctor's device, and health care management respectively. We described the identified protocols separately and in the next chapter we show how they work together by simulating the behaviour of an entire system, using an automatic simulation tool.

Chapter 5: Protocol Simulation in Different Scenarios

In Chapter 4, we used the *Specification and Description Language* (SDL) to describe in detail our proposed protocols needed to support our framework for ubiquitous access to *Electronic Health Records*. The protocols define the processes and procedures that must be followed when patients and health care providers want to communicate with each other. The protocols work on behalf of a patient's device, a doctor's device, and health care management respectively.

Although the behaviour of the individual protocols was clearly defined in Chapter 4, analysis of the protocols' combined behaviour is necessary to confirm that the system as a whole behaves correctly. To do this, in this chapter we simulate the behaviour of the entire system by using an automatic simulation tool, UPPAAL. The basis of the UPPAAL model is the notion of timed automata developed as an extension of classical Finite State Automata with clock variables (Hessel, 2001).

In this chapter, we show how our protocols work together in different scenarios such as *Access Granting*, *Modifying ACLs*, *Viewing EHRs*, *Unauthorised Access*, *Consultations,* and *Emergencies* by using UPPAAL to generate *Message Sequence Charts* (MSCs). MSCs are a standardised notation used for the description of typical or exceptional message exchanges between entities. MSC diagrams provide a clear description of system communication in the form of message flows (ETSI, 2009). MSCs and SDL descriptions should be regarded as different but complementary views of a system. SDL provides behaviour descriptions of individual communicating entities, but there is no direct description of communication between several entities. By contrast, MSCs provide a clear description of system traces between processes (ETSI, 2009).

5.1 UPPAAL OVERVIEW

UPPAAL is a tool box for symbolic simulation and automatic verification of real-time systems modeled as networks of timed automata extended with integer variables. More precisely, an UPPAAL model consists of a collection of non-deterministic processes with finite control structure and real-valued clocks communicating through channels and shared integer variables. The current version of UPPAAL includes the following main features (Havelund et al., 1997):

- A graphical interface allowing graphical descriptions of systems.
- A simulator, which provides a graphical visualisation and recording of the possible dynamic behaviours of a system description.
- A model checker for automatic verification of safety and bounded-Iiveness properties by on-the-fly reachability analysis.

The user interface consists of three parts: a system editor, simulator and verifier. The system editor enables the user to model a real time system as a network of timed finite state automata, global or local variables and clocks. The automata templates are entered by means of a graphical notation that resembles the standard notation for timed automata. The user specifies the instances of the templates that are in the model and can pass parameters to them.

The user can declare global variables, clocks and synchronisation channels. Global variables are variables that can be modified by any instance of any automaton in the model. Global clocks can be reset or assigned a natural number by any automaton. A synchronisation channel is used to guarantee that two transitions of two different automata are executed together in the system composed of them. The transitions to be synchronised have to be labelled by complementary outputs *ch*! and inputs *ch*?, where "*ch*" is the synchronisation channel name (Brandozzi, 1995).

A system description (or model) in UPPAAL consists of a collection of automata modelling the finite control structures of the system. In addition the model uses a finite set of (global) real-valued clocks and integer variables. Consider the model of Figure 30 and Figure 31. The model consists of two components A and B with states [A0, Al, A2, A3] and [B0, B1, B2, B3] respectively. In addition to these discrete control structures, the model uses two clocks *x* and *y,* one integer variable *n* and a channel *a* for communication. The edges of the automata are decorated with

three types of labels: a *guard,* expressing a condition on the values of clocks and integer variables that must be satisfied in order for the edge to be taken; a synchronisation action which is performed when the edge is taken forcing synchronisation with another component on a complementary action, and finally a number of clock resets and assignments to integer variables.

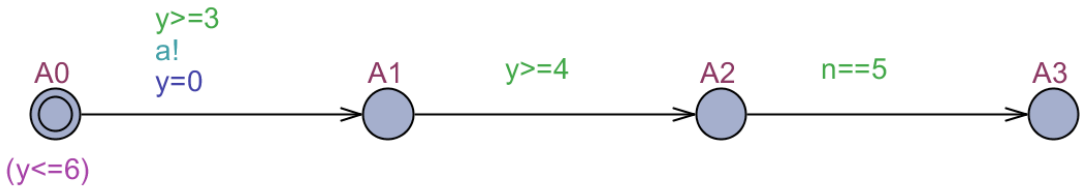


Figure 30. Example UPPAAL model A

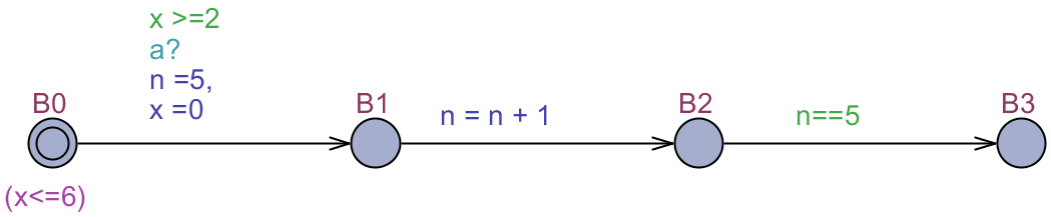


Figure 31. Example UPPAAL model B

The *simulator* then offers the ability to interactively run a system and check for mistakes in its design. The simulator shows a graphic representation of all the automata that compose the system, with their current control nodes and enabled transitions highlighted. The simulator lets the user decide which one of the enabled transitions to execute next. In the simulator the user can also see the values of all the global and local variables and clocks (Brandozzi, 1995). In the case of the example in Figure 30 and 31, the two processes must synchronise on their first transition via channel *a*, after which they will execute independently.

5.2 HOW TO TRANSLATE SDL TO UPPAAL

An SDL model as used herein includes five main parts, being the start symbol, the state symbol, the input symbol, the output symbol and the decision symbol. An example of an SDL model for *Authenticating the Patient* which was explained in Section 4.4.1 can be seen in Figure 32 and its UPPAAL counterpart can be seen in Figure 33. The initial state of the UPPAAL model is defined by the state symbol

labelled *SP_Start*. The other states include *Idle*, *pEntPin*, *spPinVrfy*, *pPutFp*, *spFpVrfy*, *cnt1Chk*, *cnt2Chk*, *AccessGranted*, and *spBlkAmim*. These states can be found in italic format beside the SDL symbols in Figure 32. The decision symbol in the SDL diagram is modelled as a guard in the UPPAAL. The decision symbol divides the path in two ways, one for true, when the condition is met, and one for false, so we have two separate transitions in the UPPAAL model. For example decision "PIN Matched?" in Figure 32 corresponds to state *spPinVrfy* in Figure 33 and its two guarded outgoing transitions. In Figure 33 if the condition "*spPin* equals 0" evaluates to true in the *spPinVrfy* state, variable *Cnt1* will be updated, and the process goes to the *cnt1Chk* state. Otherwise (*spPin* equals 1) it will go to the *pPutFp* state and update the variable *Cnt2*. The SDL variables, such as *Cnt1,* correspond directly to the UPPAAL variables. Similarly, input and output symbols in SDL are translated to channel input (?) and output (!) labels on the corresponding transitions in UPPAAL.

To make the simulation more complete, our UPPAAL models sometimes contain more detail then the SDL models in Chapter 4. For instance, the *ACL Modification* task in Figure 14 has been expanded in the corresponding UPPAAL model to show the "*Modify*" and "*Modification is done*" messages exchanged between the SP and the TTP (Figure 43).

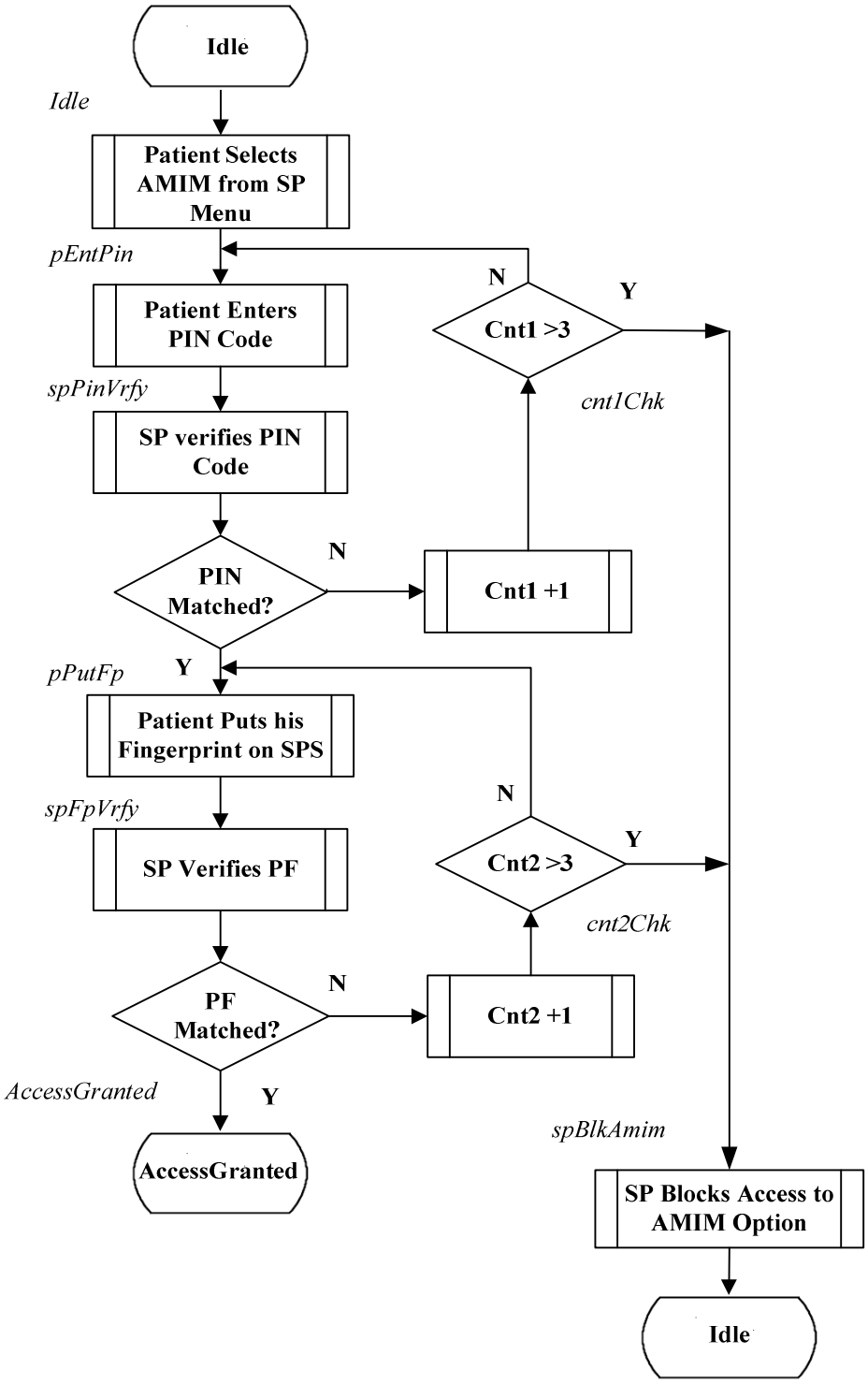


Figure 32. SDL Model for Authenticating a Patient

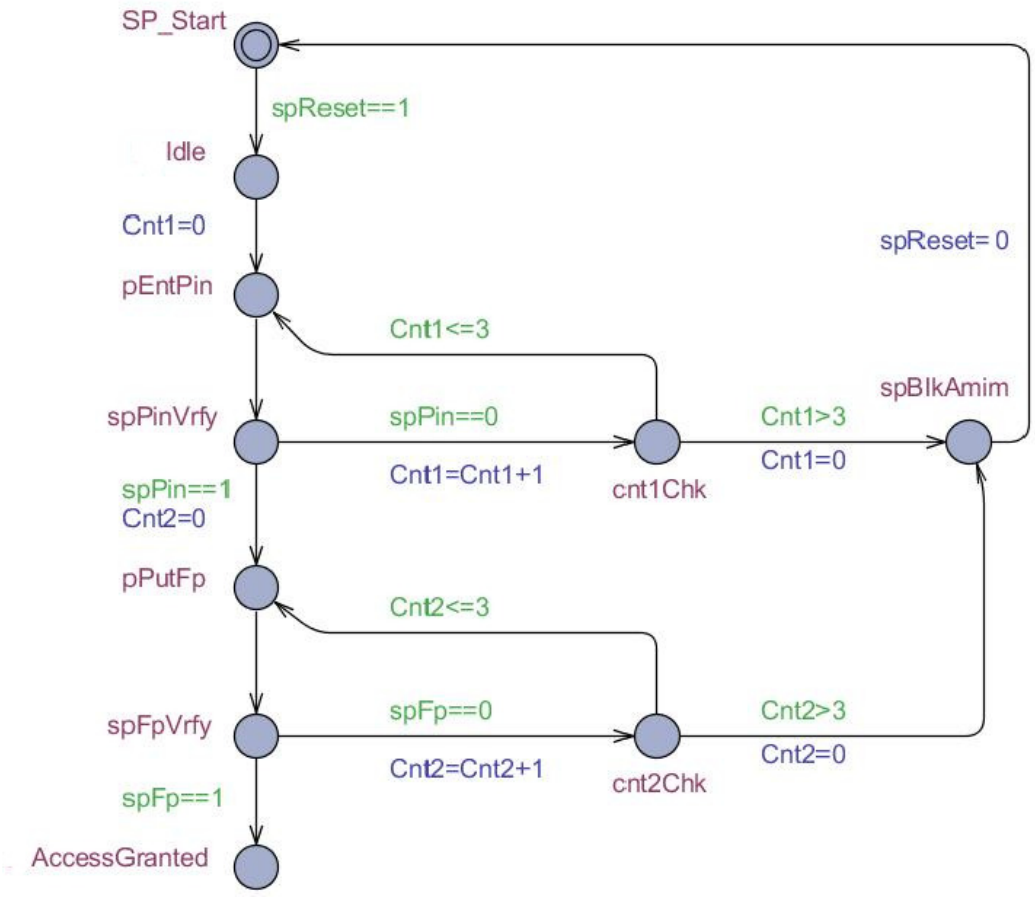


Figure 33. The Model in UPPAAL for Authenticating a Patient

5.3 PROTOCOL SIMULATION IN THE DIFFERENT SCENARIOS

Simulation is a very important element in all development projects. It helps to save development time, costs, and better understand the functions, and provide early tests of the system to find critical errors. In this section we undertake simulation-based studies of our protocols by considering six different healthcare scenarios where our protocols can be applied: *Unauthorised Access*, *Access Granting*, *Viewing EHR*s, *Modifying ACLs*, *Consultations*, and *Emergencies*.

As shown in Figure 34 to Figure 36 we model our *Smart Phone* (SP), *Authorised Device* (AD), and *Trusted Third Party* (TTP) protocols by using UPPAAL's graphical language. Each of these protocols are modeled as a process in UPPAAL, thus we have three processes called SP, AD, and TTP for the protocols respectively. These processes communicate with each other through the channels which model SDL's input and output messages. Since our models of protocols cannot fit in

one page we just expand one process or procedure for each of them. To simulate the actions of people who interact with the system, such as entering a PIN, we used UPPAAL's variables which were set manually. For instance, in Figure 32 we need the patient to enter a PIN which is then checked for correctness. To simulate the patient entering the wrong PIN we would, for example, set variable *spPin* to zero in the corresponding UPPAAL model in Figure 33.

As shown in Figure 34, we modelled all processes and procedures of the *Smart Phone* protocol (Figures 13 to 18) using a single UPPAAL automaton. In this figure we expand the process for an *Emergency Situation* (Section 4.4.7). This process is called when the patient presses the *Emergency Call Button* or the patient's *Smart Phone* senses a strong vibration from the patient collapsing.

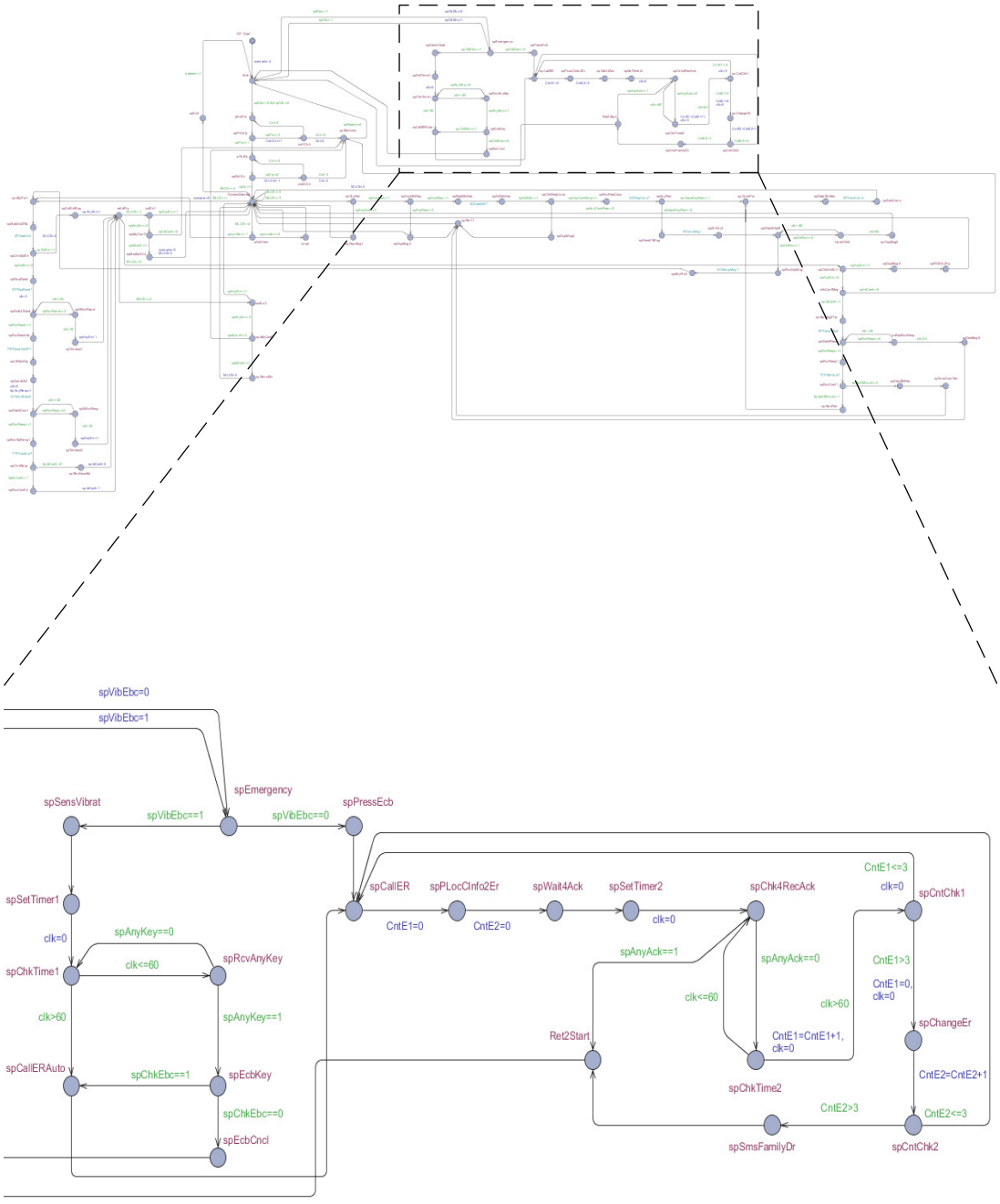


Figure 34. UPPAAL Model of the Smart Phone Protocol

Figure 35 shows the UPPAAL model for all the processes and procedures of the Authorised Device's protocol (Figures 19 to 25). In this figure we expand the process for *Authenticating the Authorised Person* (Section 4.5.1).

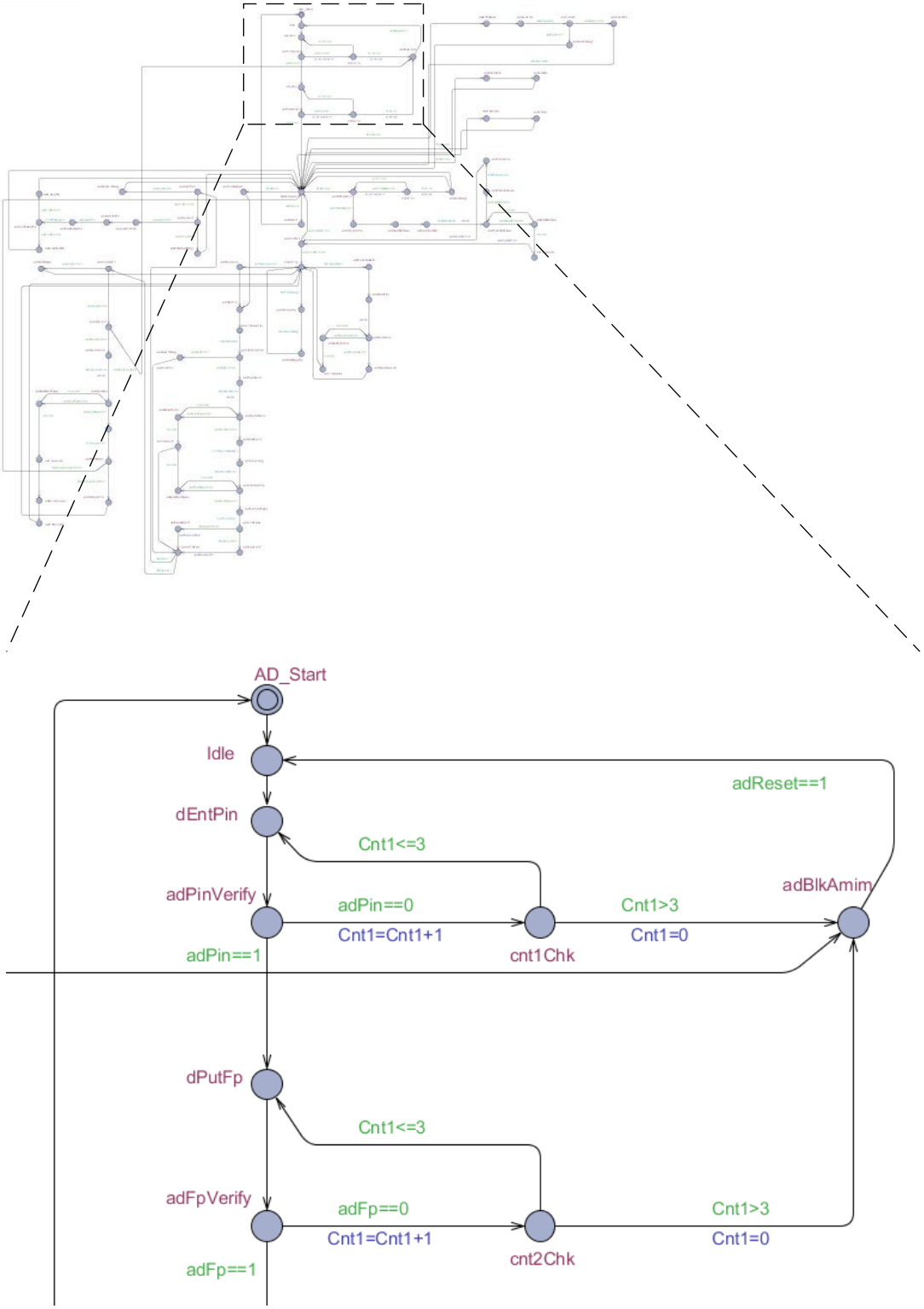


Figure 35. Uppaal Model of the Authorised Device Protocol

Figure 36 shows the UPPAAL model for all the processes and procedures of the *Trusted Third Party*'s protocol (Figures 26 to 29). In this figure we expand the process for Identifying the SP or AD (Section 4.6.4).

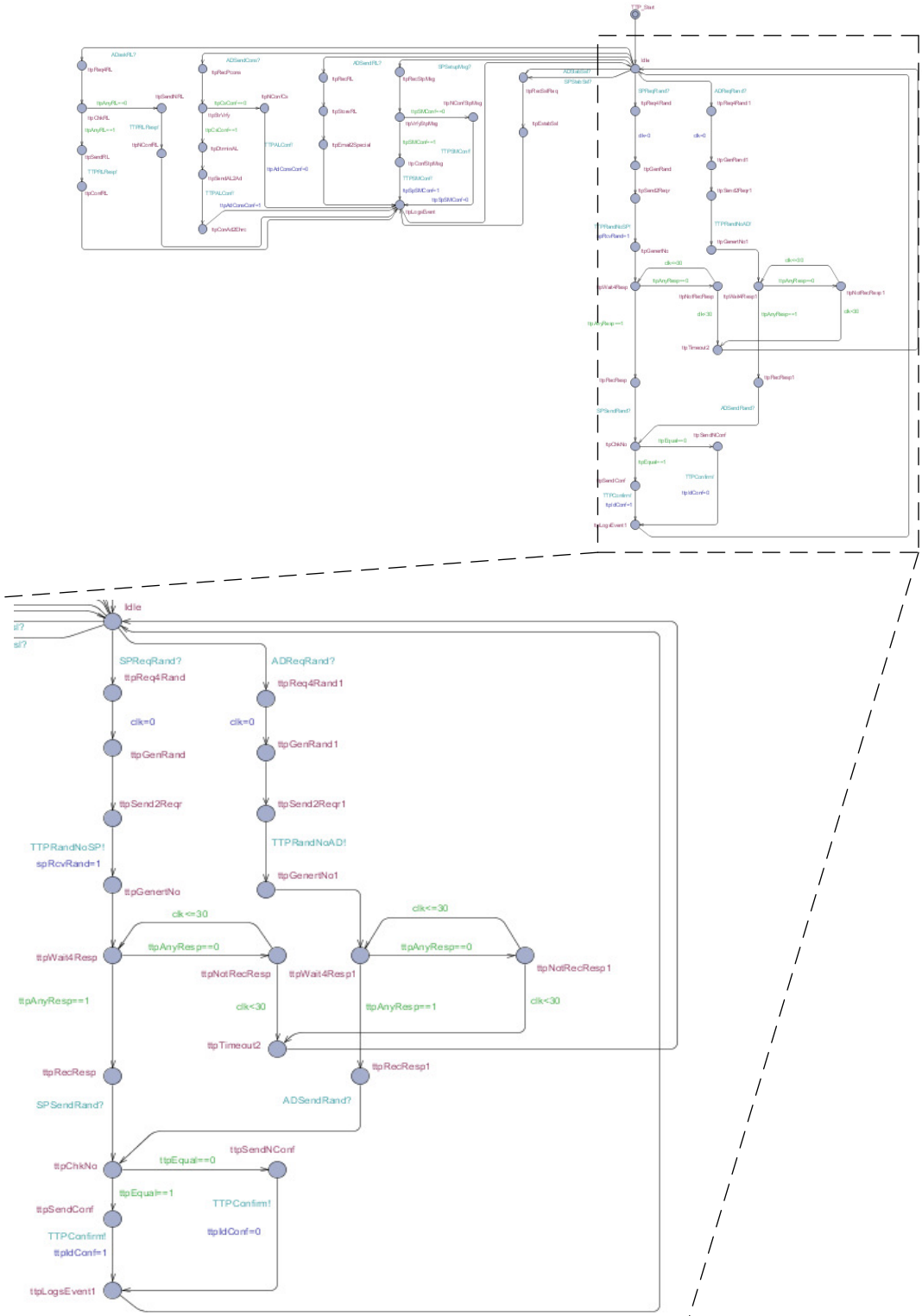


Figure 36. UPPAAL Model of the Trusted Third Party Protocol

In each of the following scenarios we present a (manually-drawn) *Message Sequence Chart* (MSC) showing how the various agents (SP, AD, and TTP) are intended to interact. This is then followed with the MSC produced by UPPAAL's simulation confirming that the protocols do indeed behave as expected. For all scenarios, there are some labels (A, B, C, and D) beside each of the MSCs which are used to map the manually-drawn MSCs to UPPAAL's automatically-generated MSCs. However, the human agents (patient, Authorised Person, etc), Ambulance, Emergency Room and EHRC processes are not part of the UPPAAL model, so no trace is produced for them. Instead we show only the transitions between the states which must be followed by the SP, AD, or the TTP in order to interact with those agents.

As shown in Figure 37, the *Message Sequence Chart* uses the following symbols to describe the system behaviours. The parallel vertical lines are used for presenting the entities' process and the vertical dashed line is used for the *Idle* state. A rectangle containing an identifier is utilised to show an entity's name. To display interaction between the entities the MSC uses horizontal arrows with the message name written above. The solid arrow is used for sending a message and the dashed arrow is utilised for showing a return message. Also to display an internal routine the MSC employs looped arrows with the routine name within.

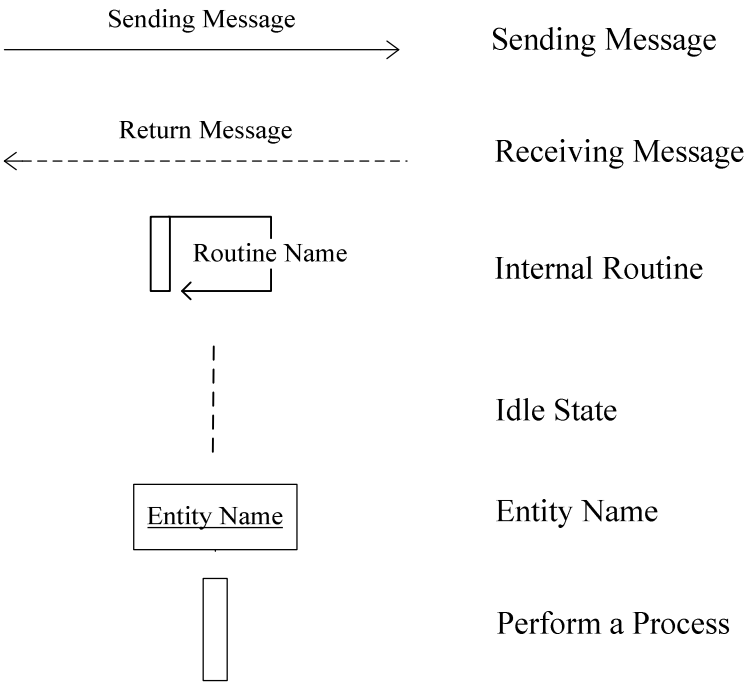


Figure 37. Message Sequence Chart Symbol Interpretation

In UPPAAL's simulation output, Figure 38, a solid arrow with the message name written above is used to show message exchange between the entities and a rounded rectangle, which contains an identifier, is employed for showing a state. The transition between the states which must be followed by the entity is shown by a vertical arrow.

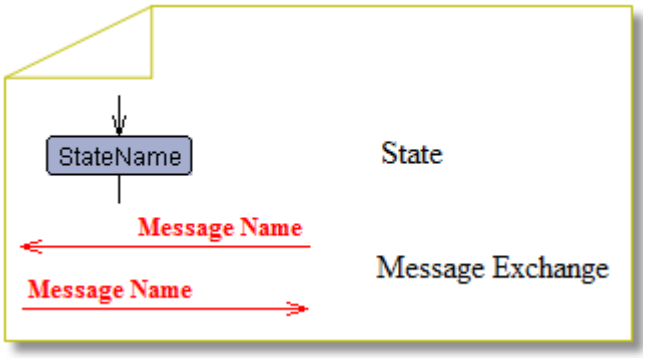


Figure 38. UPPAAL's Simulation Output Interpretation

5.3.1 GRANTING ACCESS TO AN UNKNOWN AUTHORISED PERSON

As we explained in Section 4.4.5, patients can allow or deny sharing their information with other healthcare workers. Access to the patient's medical information is prohibited unless the patient has given consent or the patient is in an emergency situation or there is legislative permission. In this scenario we show how a patient gives his or her consent to an unknown *Authorised Person*. An *unknown Authorised Person* could be a doctor who examines the patient for the first time, so the doctor needs to be confirmed by the *Trusted Third Party*. Figure 39 presents this scenario in *Message Sequence Chart* format and shows our expectation. Figure 40 shows the simulation output of the *Granting Access to an Unknown Authorised Person* scenario generated when the simulation is run in UPPAAL. As can be seen in the UPPAAL output, Figure 40, excluding internal procedures and the patient's behaviours, the results agreed completely with our expectations.

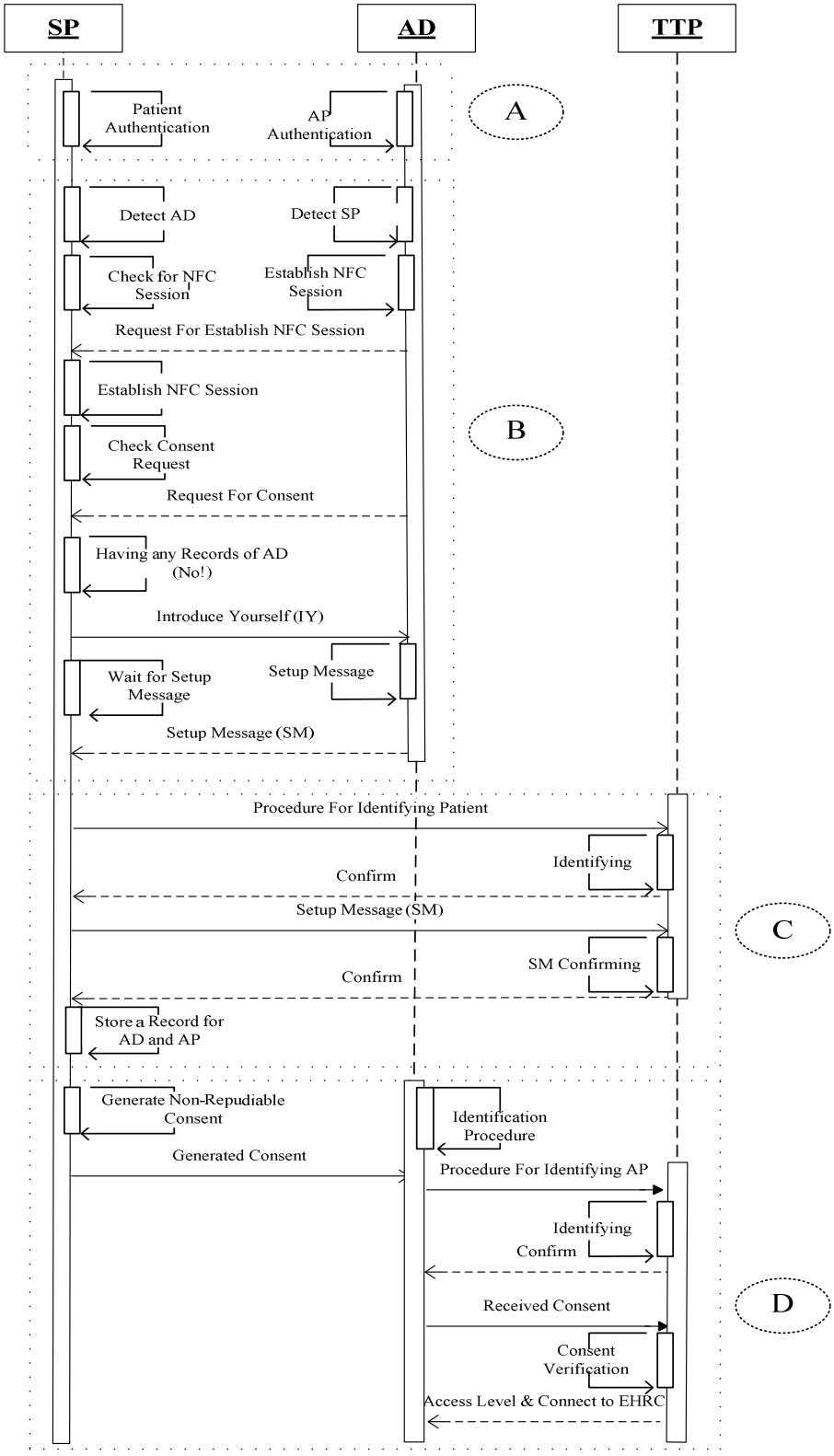


Figure 39. Message Sequence Chart for Granting Access to an Unknown Authorised Person

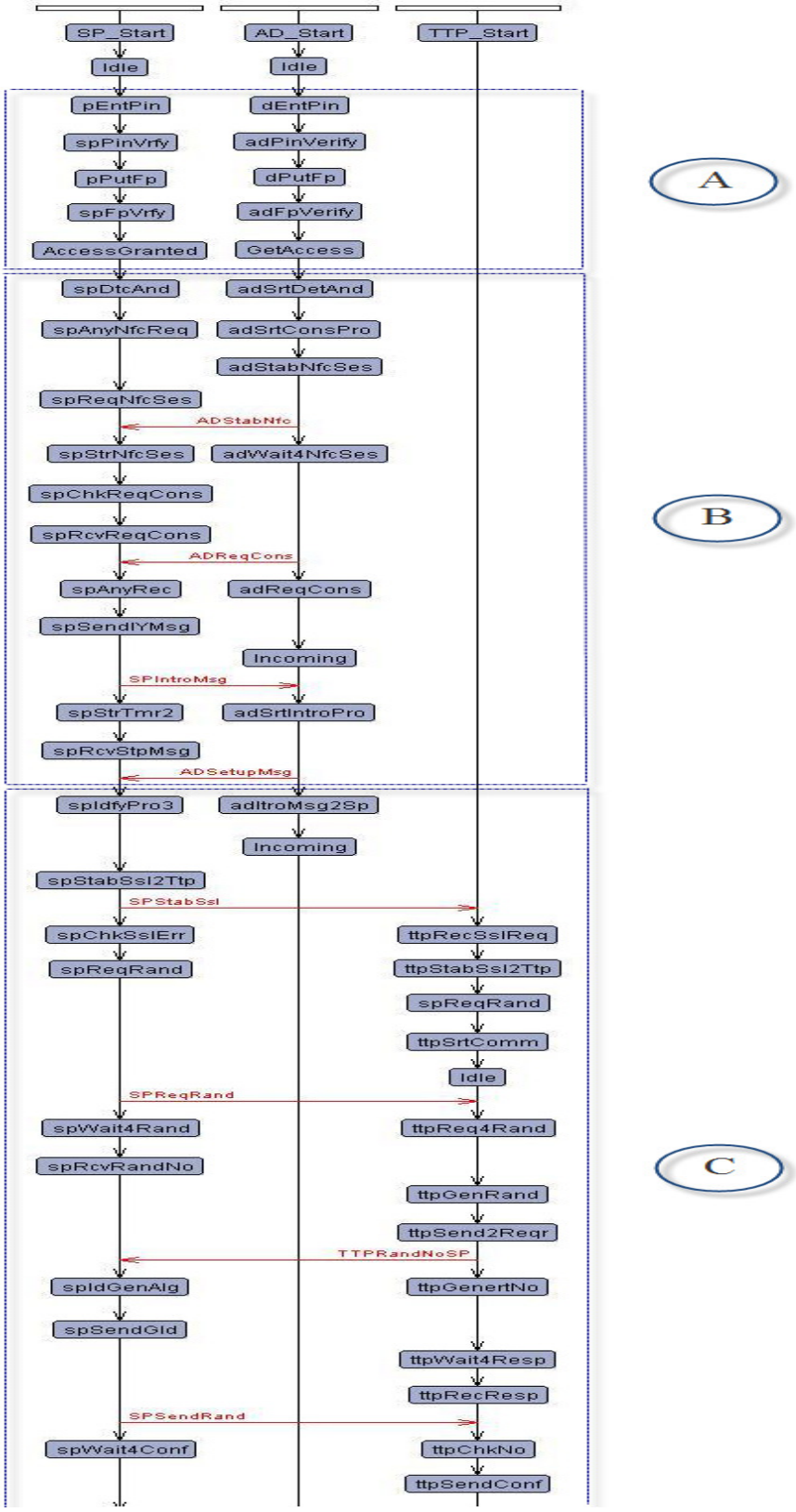


Figure 40 (a). UPPAAL Output for Granting Access to an Unknown Authorised Person (the patient's behaviour is not shown)

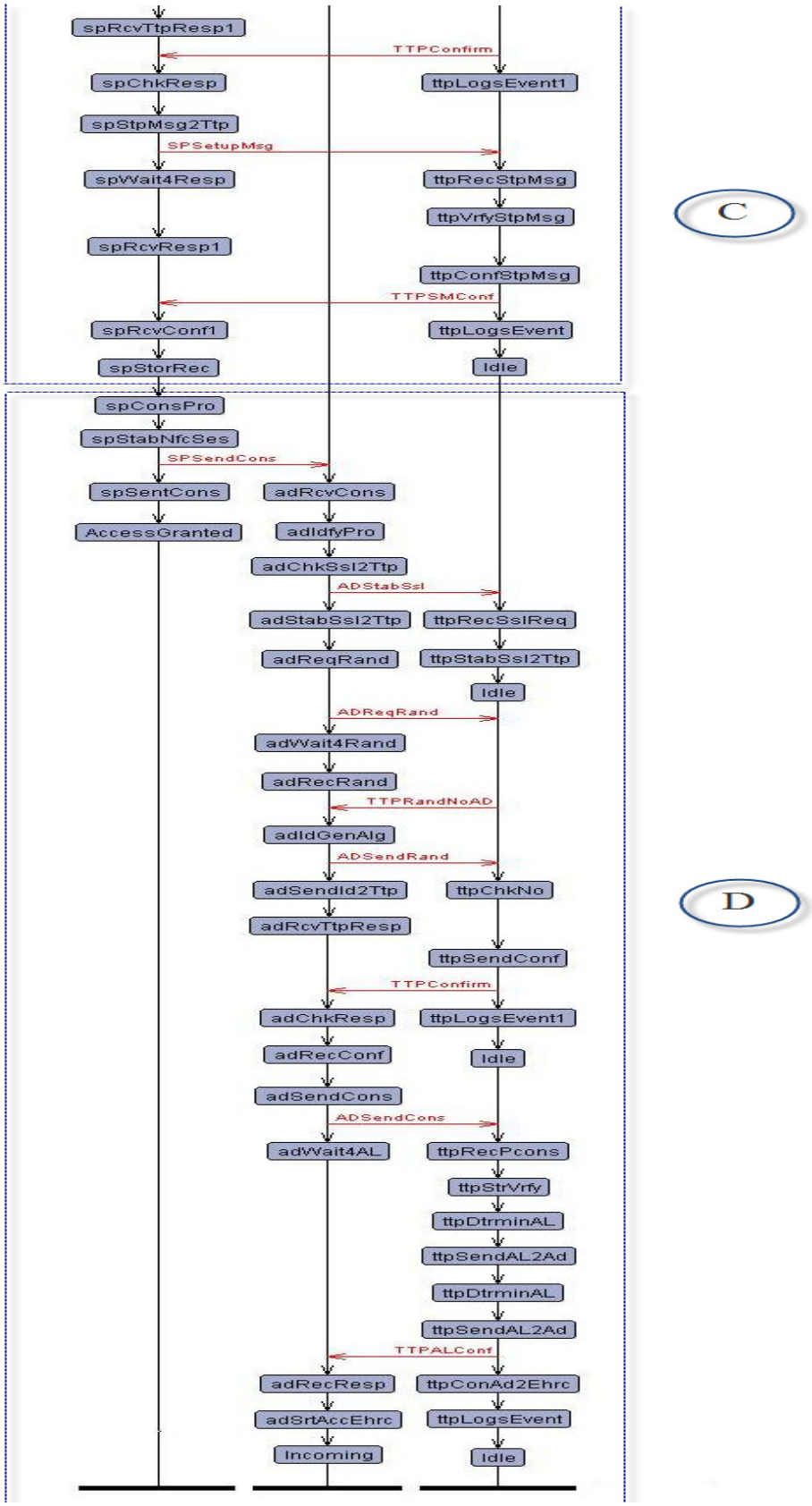


Figure 40 (b). UPPAAL Output for Granting Access to an Unknown Authorised Person (the patient's behaviour is not shown)

5.3.2 VIEWING ELECTRONIC HEALTH RECORDS

As we mentioned in Section 4.4.3, a patient must be able to view and obtain copies of their records, and request amendments to confirm they have the right to access their medical records to understand and monitor their health status and the process of diagnosis and therapy. In this scenario we show how a patient can view his or her centralised *Electronic Health Record* via a *Smart Phone*. We did not model the patient's and *Electronic Health Record Centre*'s behaviours, so no trace can be found in the simulation result (Figure 42) for them. To simulate these behaviours, we used UPPAAL's variables which were set manually. While Figure 41 presents our expectation of this scenario by using a *Message Sequence Chart*, Figure 42 shows the MSC generated by UPPAAL's simulation. As shown in the following figures the simulation result is in agreement with our prediction.

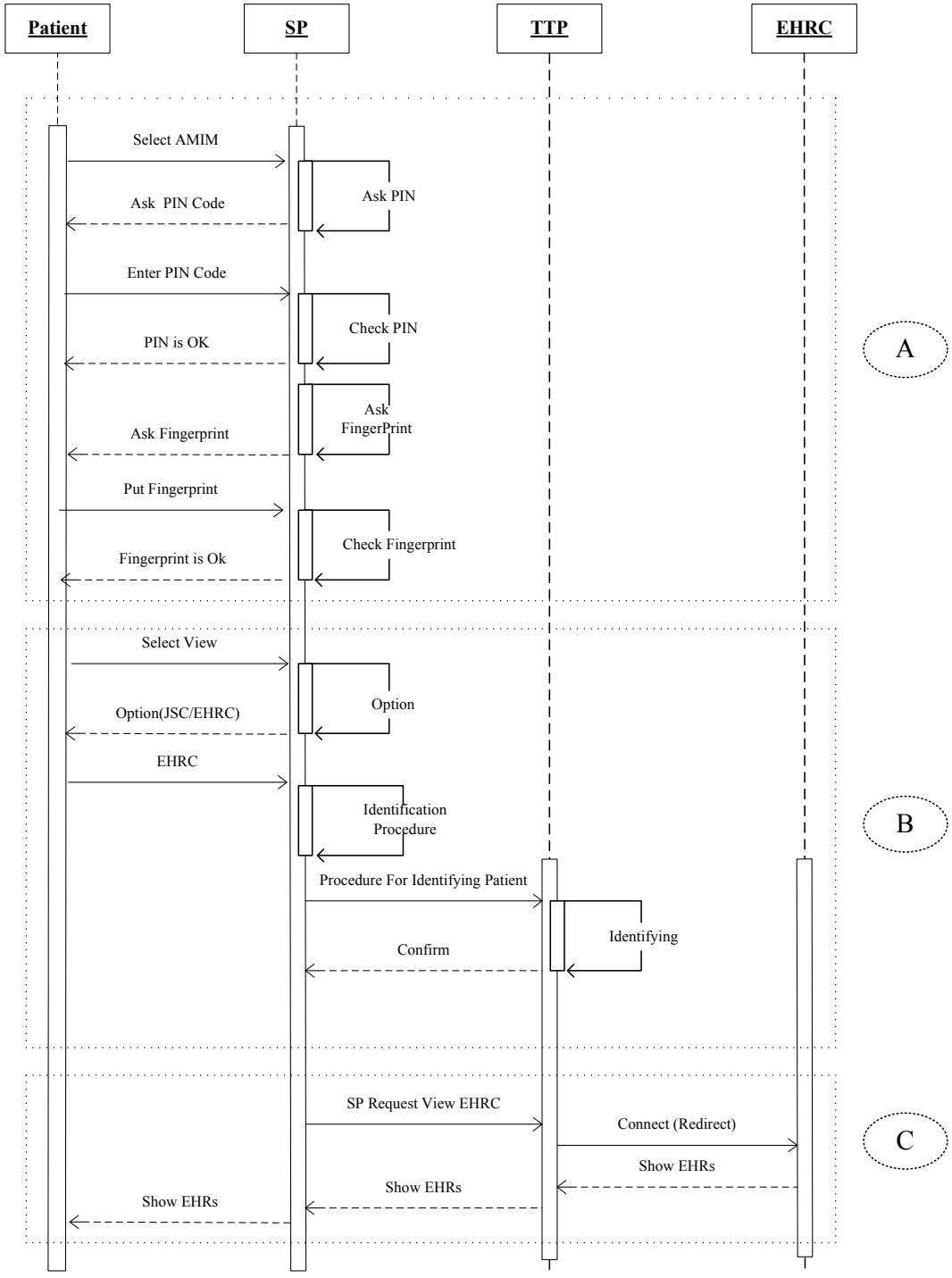


Figure 41. Message Sequence Chart for Viewing Electronic Health Records on the EHRC

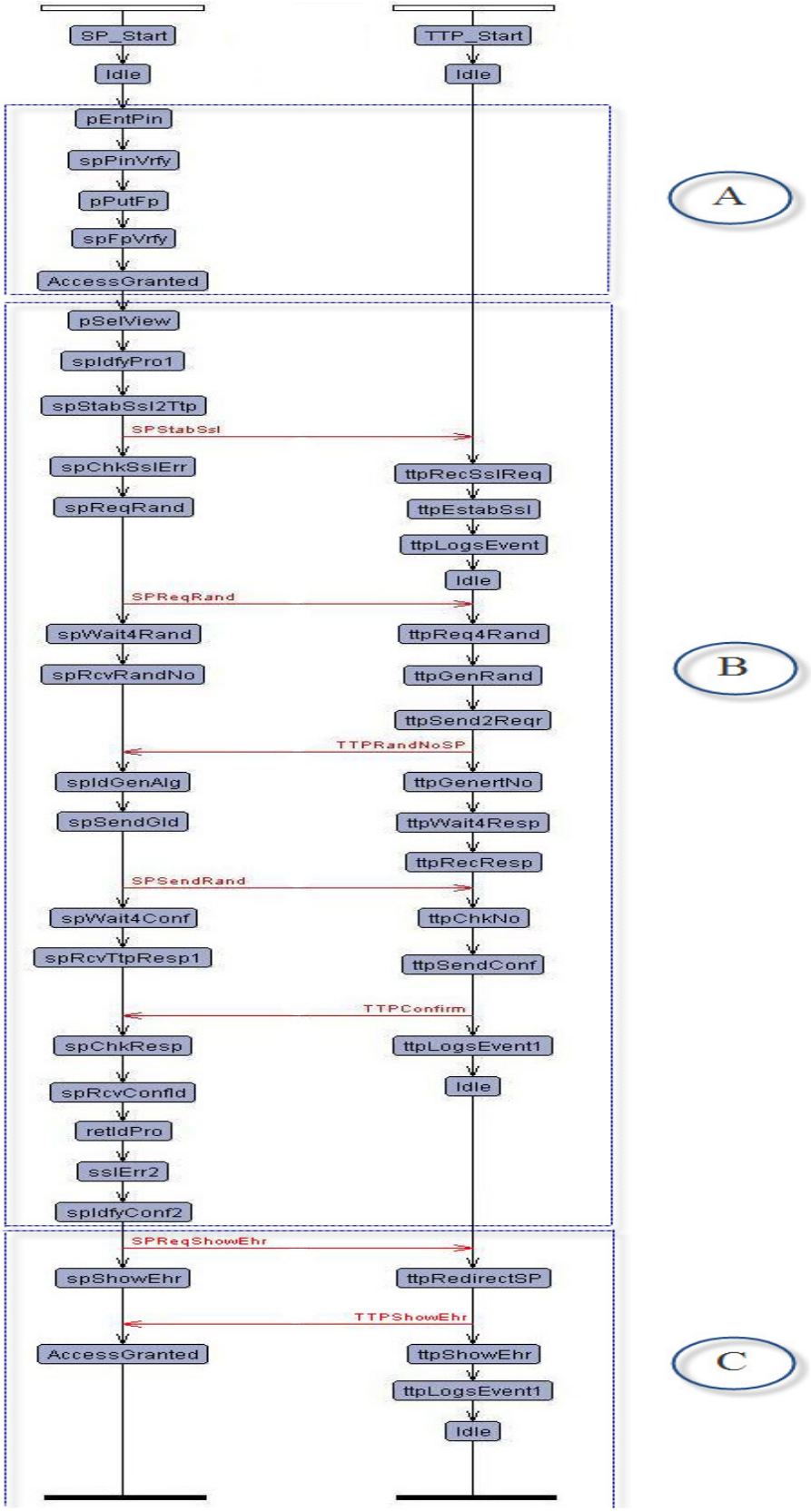


Figure 42. UPPAAL Output for Viewing Electronic Health Records on the EHRC (the patient's behaviour and the EHRC are not shown)

5.3.3 MODIFYING AN ACCESS CONTROL LIST

As we stated in Section 4.4.2, many people consider information about their health to be highly sensitive. They prefer to have rights of access to their medical information and to be entitled to decide who can access their record. Based on these assumptions, we considered three *Access Control Lists* which are stored on the TTP and can be modified by the patient. In this scenario we show how a patient can modify his or her *Access Control Lists* via a *Smart Phone*. We did not model the patient's behaviours, so no trace can be found in the simulation result (Figure 44) for the patient. To simulate these kinds of behaviours, we used UPPAAL's variables which were set manually. As shown in Figure 43, we use a *Message Sequence Chart* to show our expectation of this scenario. The simulation's output from UPPAAL is presented in Figure 44.

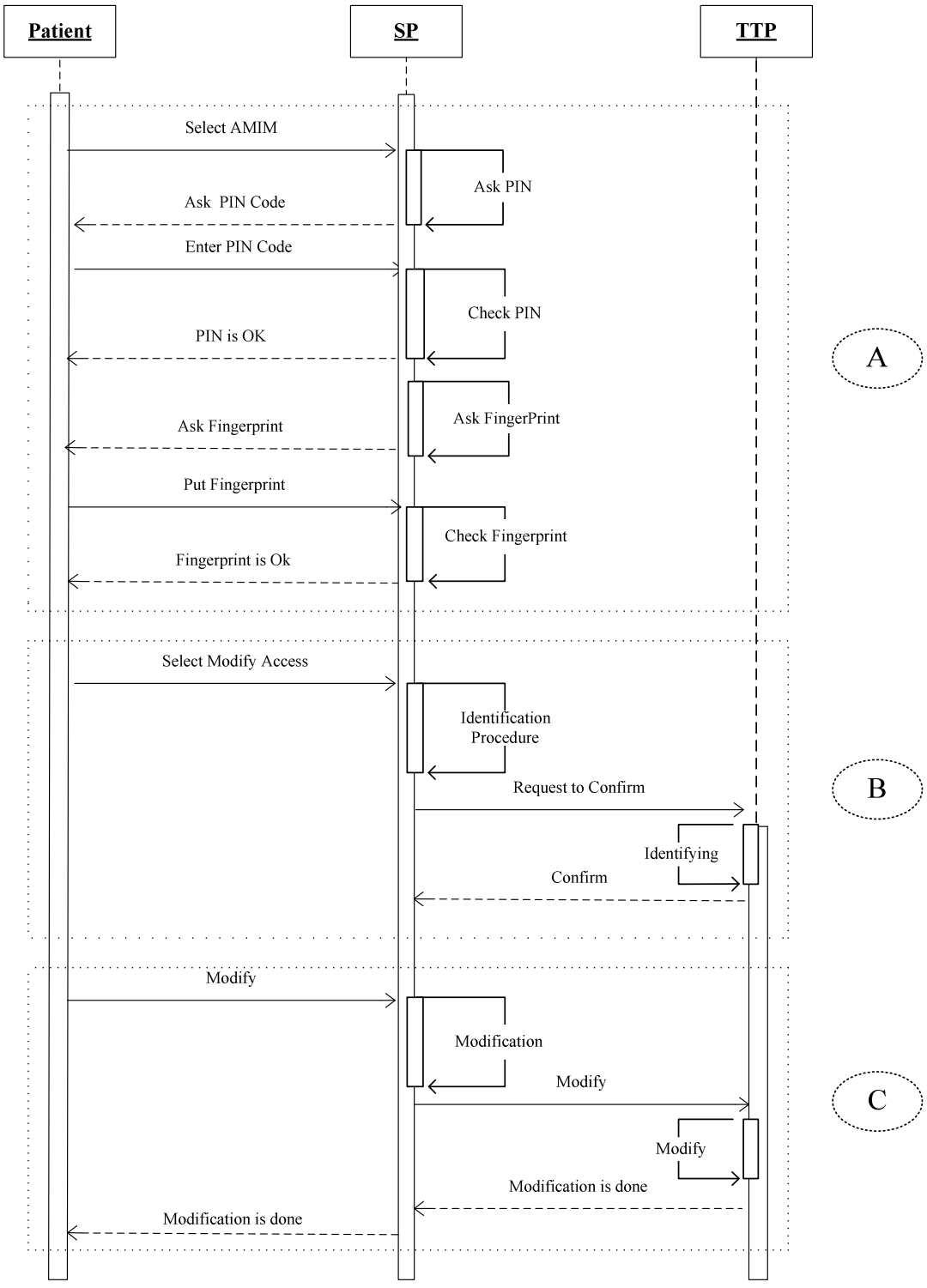


Figure 43. Modifying an Access Control List

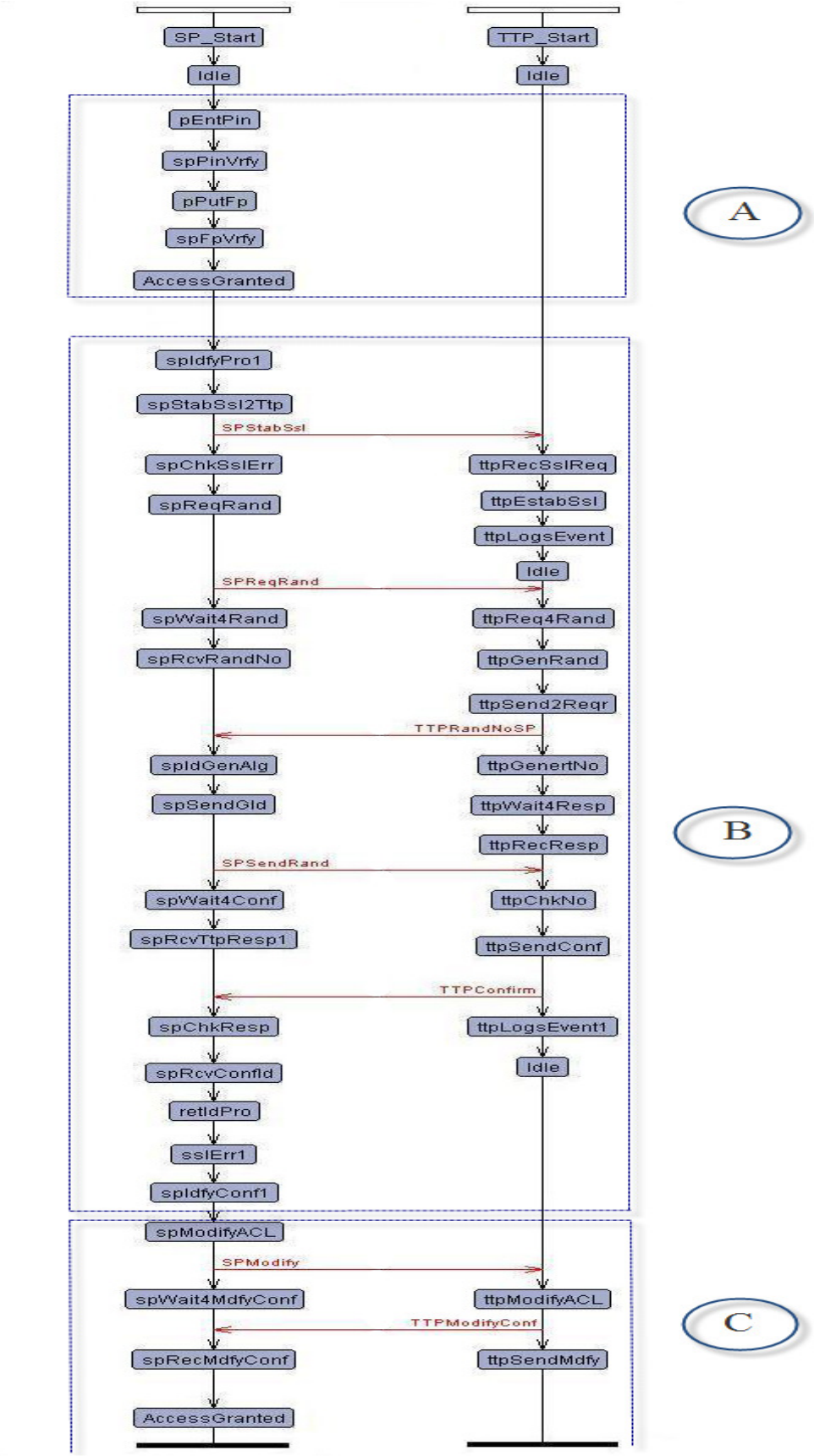


Figure 44. UPPAAL Output for Modifying an Access Control List Scenario (the patient's behaviour is not shown)

5.3.4 UNAUTHORISED ACCESS

In this scenario we show how our protocols can prevent an unauthorised person from accessing the patient's *Electronic Health Records* by simulating the process for *Authenticating the Patient* (Section 4.4.1) which is same as the process for *Authenticating an Authorised Person* (Section 4.5.1). We expect this scenario will follow the sequence illustrated by the *Message Sequence Chart* (MSC) in Figure 45. After a patient (or the AP) enters the PIN three times incorrectly, the SP (or the AD) blocks access to the AMIM option. Figure 46 presents the simulation output of the *Authenticating the Patient* scenario from UPPAAL. This figure shows only the SP behaviours by presenting the transitions between the states which must be followed by the SP in order to authenticate the patient. As we did not model the patient's behaviours, such as entering a PIN code, no trace can be found in the simulation result (Figure 46) for the patient. As the following two figures show, the simulation result agrees with our prediction, with the SP completing its tasks and returning to the *Idle* state.

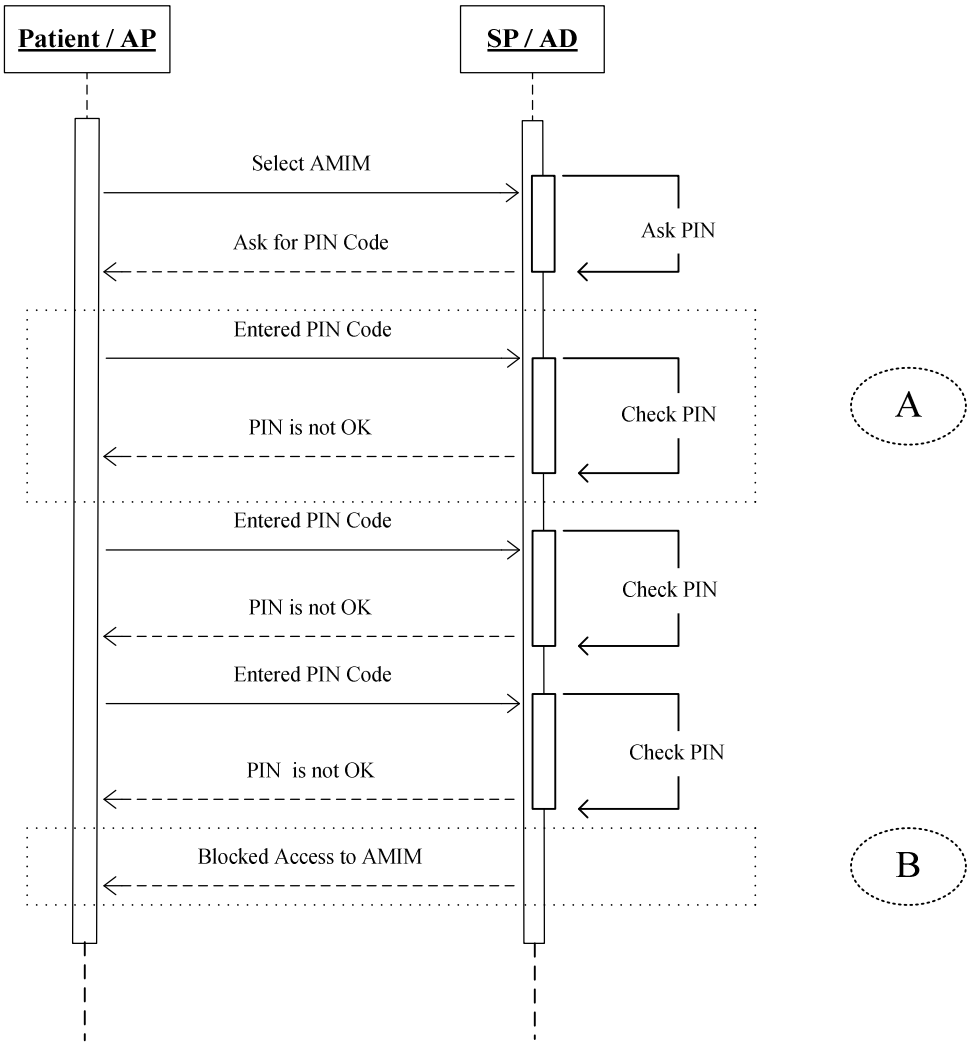


Figure 45. Message Sequence Chart for the Unauthorised Access Scenario

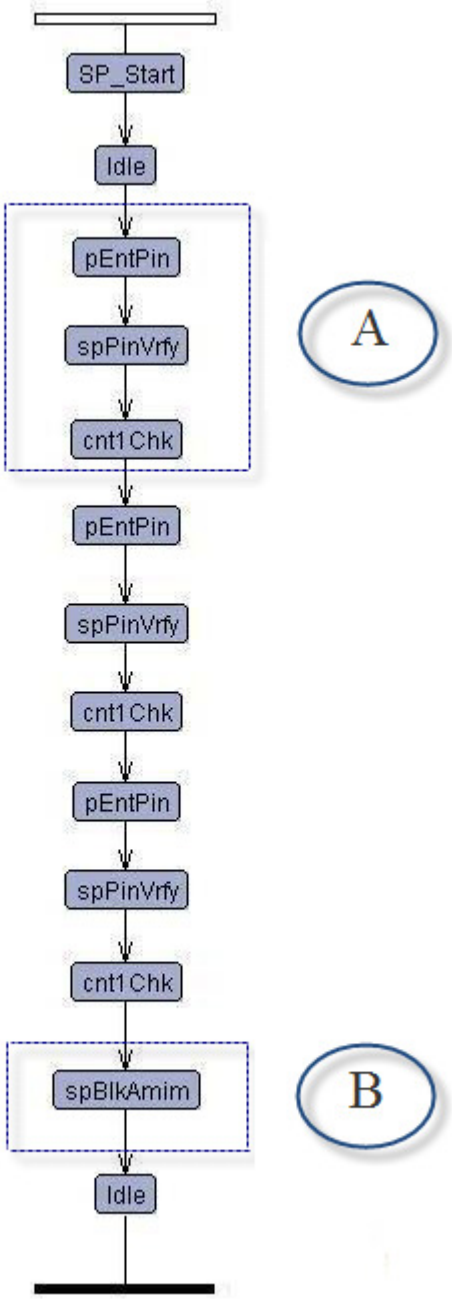


Figure 46. UPPAAL Output for Unauthorised Access Scenario
(only the SP's behaviour is shown)

5.3.5 CONSULTATION PROCESS

As we described in Section 4.5.4 and 4.5.7 a *Referral Letter* is used for transferring information between health care professionals in order to provide proper patient care. In this scenario we show how a referring doctor (General Practitioner) can send a *Referral Letter* and how a receiving doctor (specialist) can access it. In Fig-

ure 47 we present this scenario using *Message Sequence Chart* format. Figure 48 and Figure 49 demonstrate the simulation's results for storing and retrieving a referral letter respectively. As we did not model the GP's and specialist's behaviours, no trace can be found in simulation result (Figures 48 and 49) for them. Both the GP and specialist use their own AD for communication with the TTP. As shown in the following figures the results agree with our expectation.

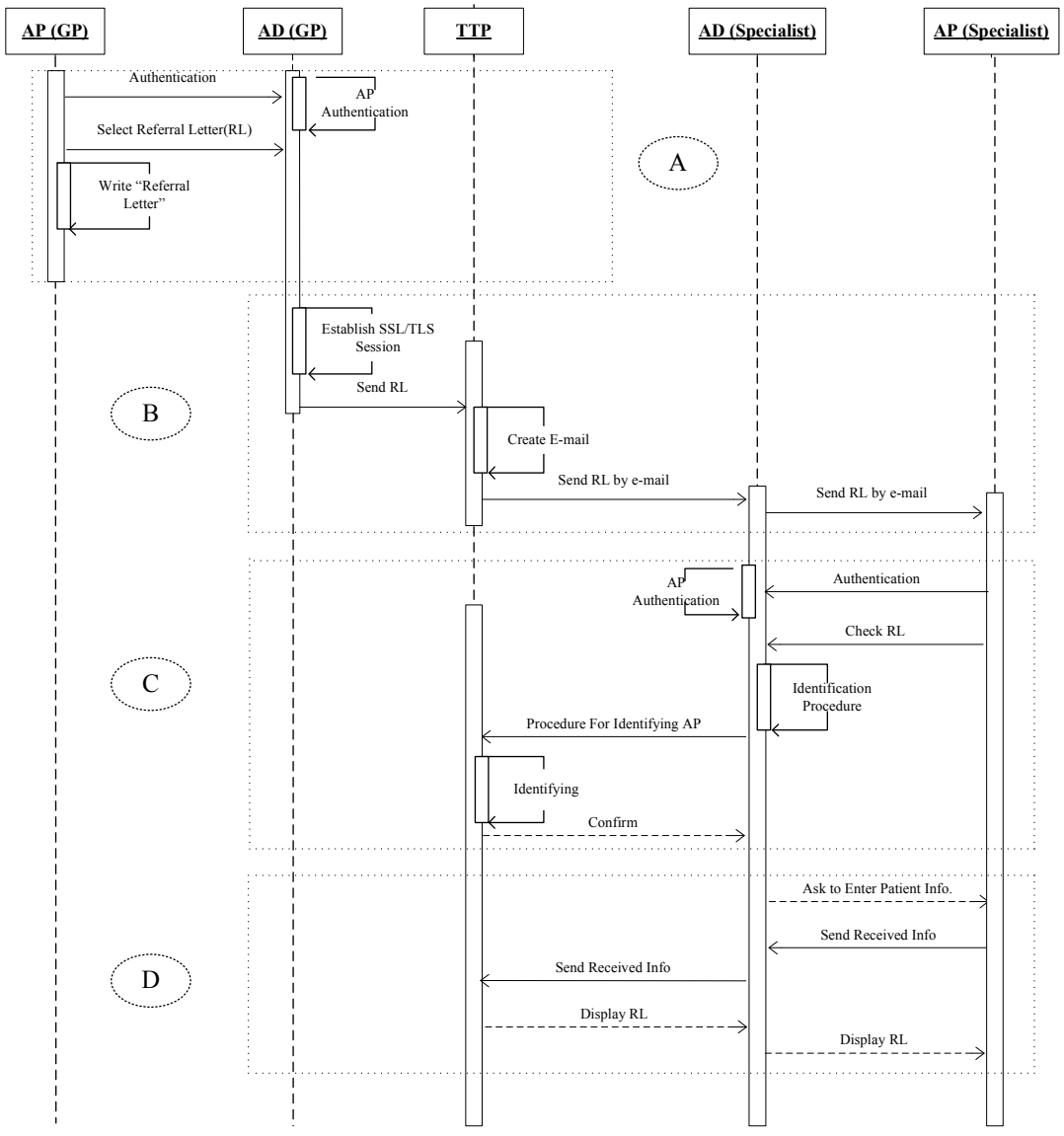


Figure 47. Referral Letter Storage and Retrieval

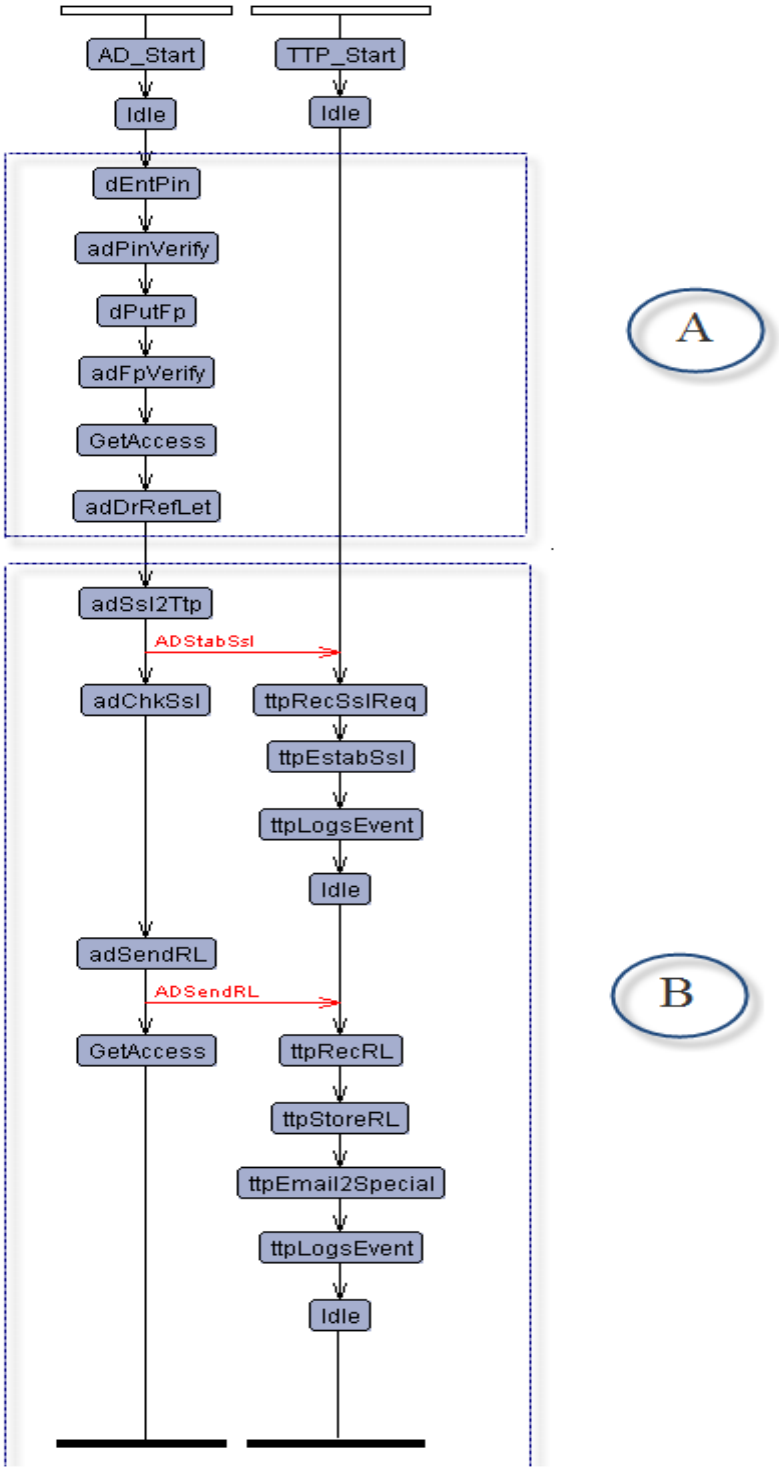


Figure 48. UPPAAL Output for Storing a Referral Letter Scenario
(the AP's behaviour is not shown)

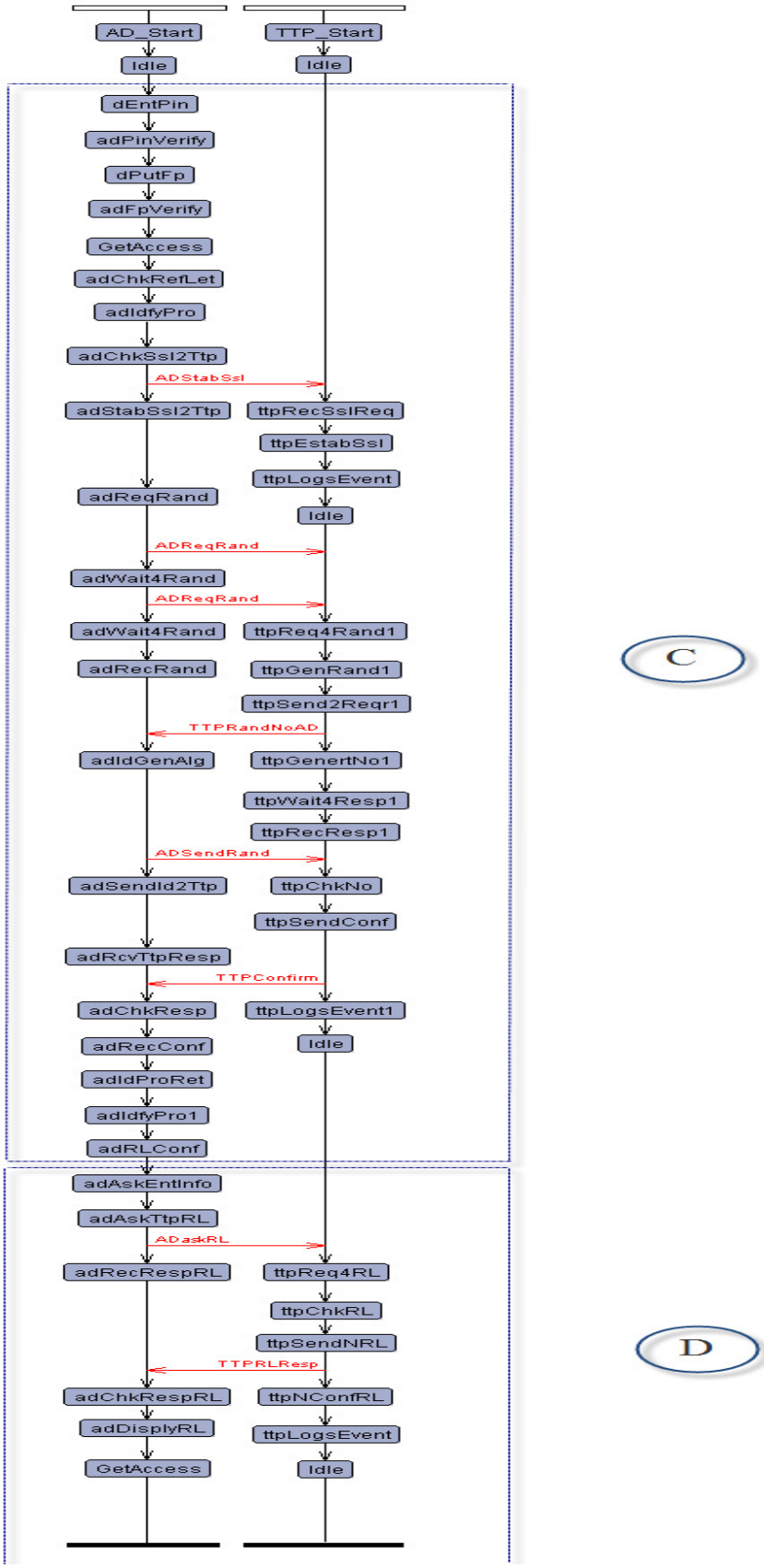


Figure 49. UPPAAL Output for Retrieving a Referral Letter Scenario
(the AP's behaviour is not shown)

5.3.6 EMERGENCIES

As we described in Section 4.4.7, in emergency situations *Smart Phones* give users the benefit of instant wireless communication, combined with detecting and informing a patient's location, and automatically calling and sending patient details to an Emergency Room. After the SP senses a strong vibration from the patient falling down or the patient presses the Emergency Call Button, the SP automatically calls the ER and sends the location and details of the patient to the ER. When the ER gets the information from the SP, it can identify where the patient is and what the problem is. Now the ER can send an ambulance armed with this useful information including patient's photo to the patient's location in order to speed up and improve the quality of treatment. Different emergency scenarios are shown in Figures 50, 52, and 54. As we did not model the patient's or an *Emergency Room*'s actions for simulation, no trace can be found in simulation output (Figures 51, 53, and 55). As shown in Figures 50, and 52, we present scenarios in which a patient needs emergency care and presses the Emergency Call Button from his or her SP. In the first scenario, Figure 50, we assume sending the patient's details and location to the *Emergency Room* (ER) is successful and in the second one, Figure 52, we assume sending is not successful. In the second scenario, if the SP doesn't receive any ACK from the first nearest Emergency Room after three attempts, it will try the second and third nearest ER with the same procedure. UPPAAL's simulations of these scenarios are shown in Figure 51, Figure 53, and Figure 55 respectively. These figures show only the SP behaviours by presenting the transitions between the states which must be followed by the SP in order to handle emergency situations. In Figure 54, we show a scenario in which a patient collapses when no one else is around and the SP senses a strong vibration from the patient falling down. It starts continuous ringing and vibrating to alert the patient to the fact that his or her phone is going to call the emergency room. If no cancelation action is performed by the patient it automatically calls the ER and sends the location and details of the patient. Figure 55 presents the simulation result for this scenario from UPPAAL. As the following figures show, the simulation results once again agree with our expectations.

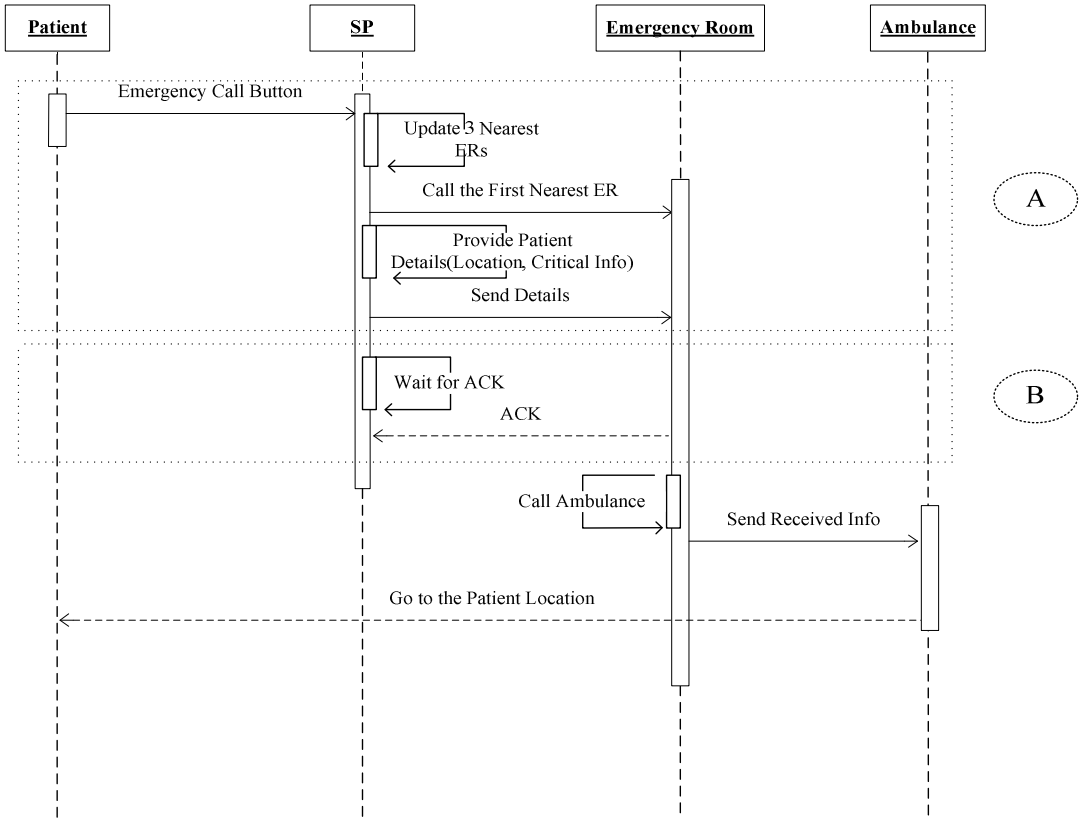


Figure 50. Patient Calls the Emergency Room for an Ambulance Scenario

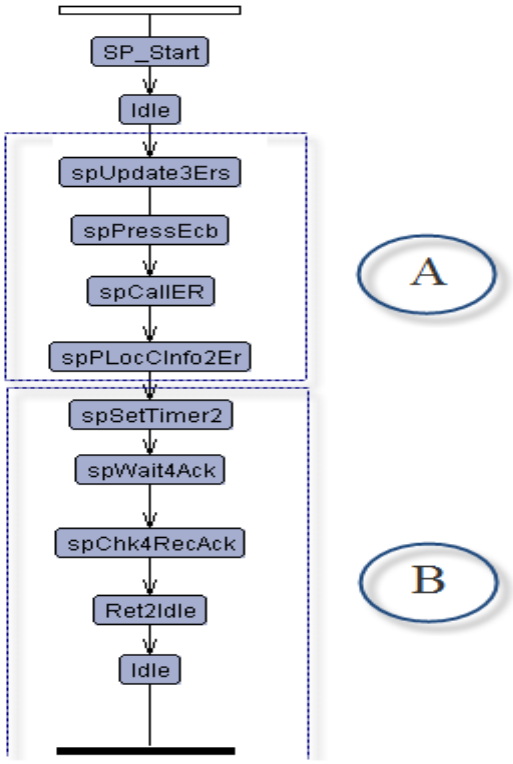


Figure 51. UPPAAL Output for the Patient Calls the Emergency Room for an Ambulance Scenario (only the SP's behaviour is shown)

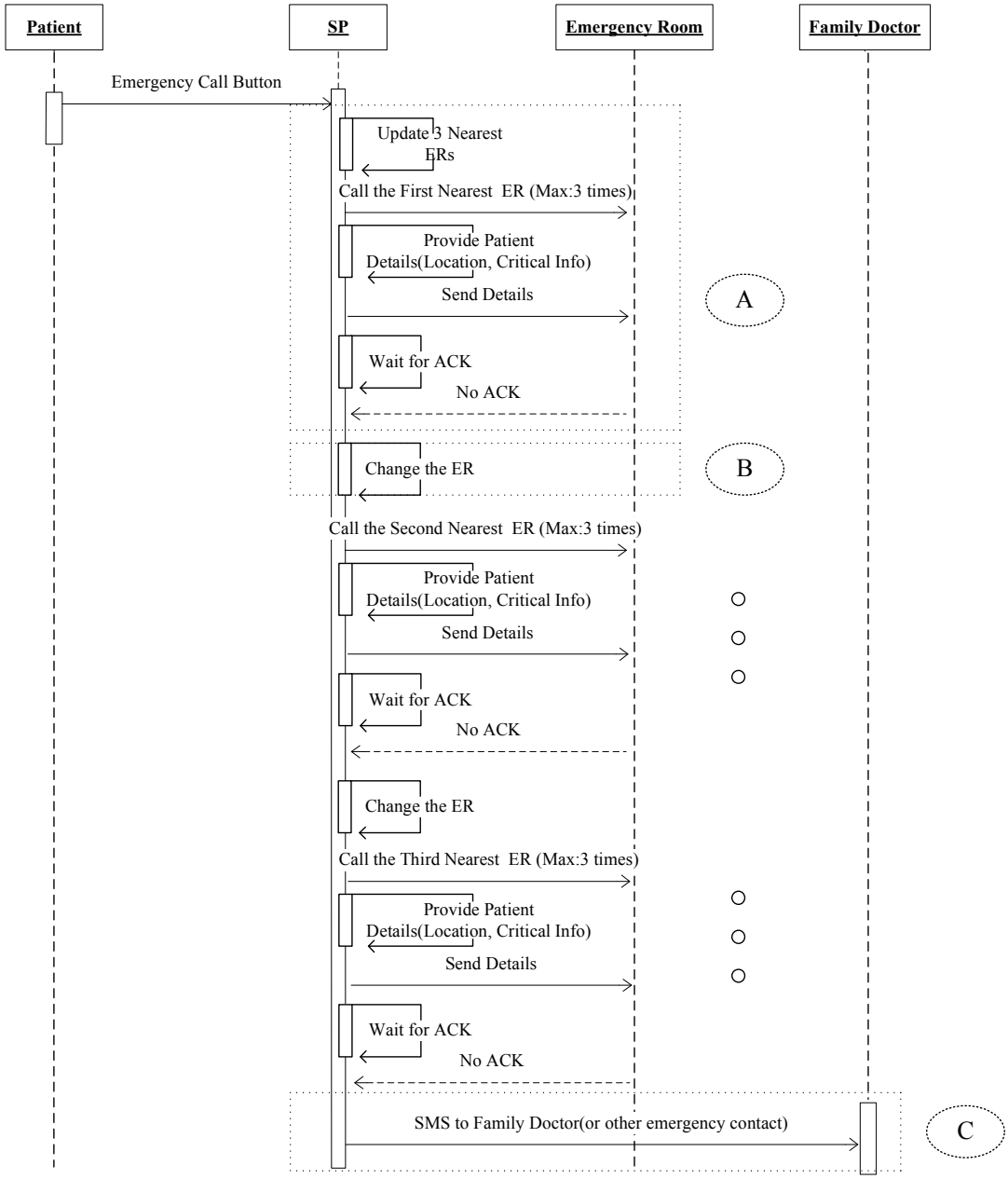


Figure 52. Failure to Contact an Emergency Room Scenario

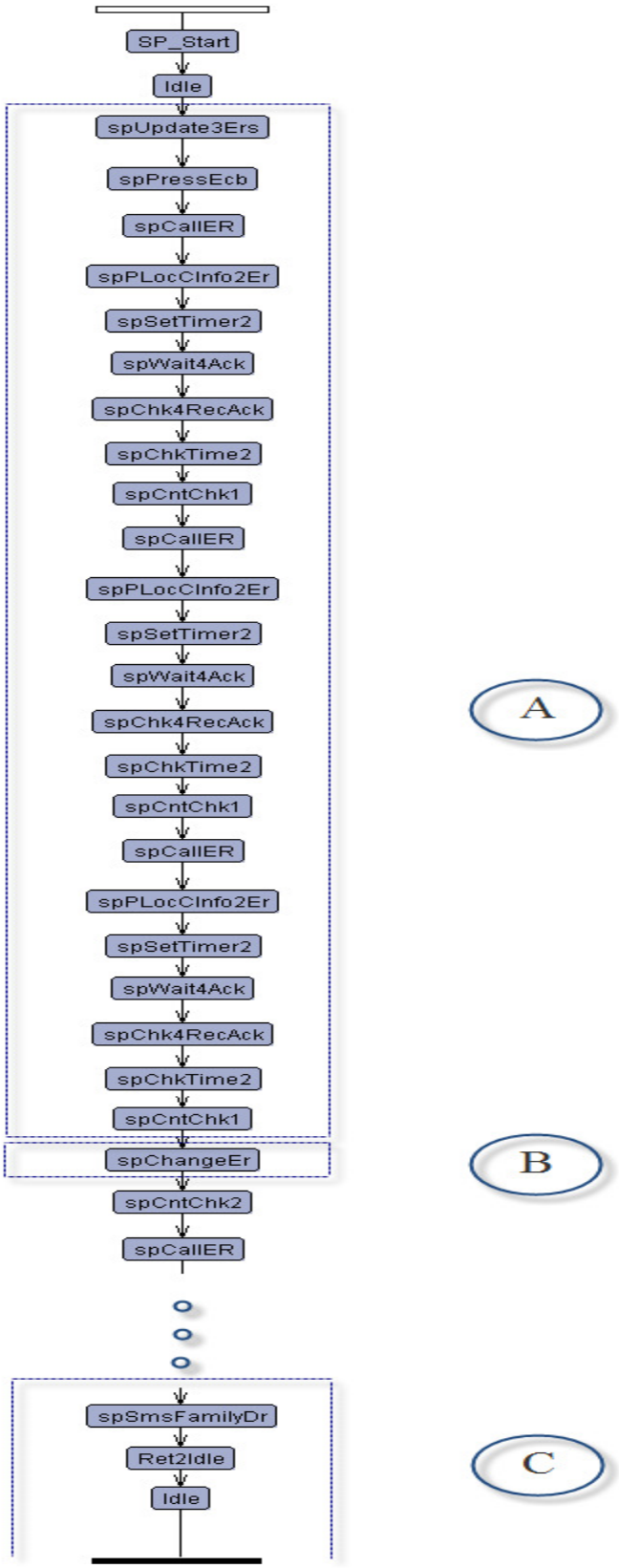


Figure 53. UPPAAL Output for the Failure to Contact an Emergency Room Scenario (only the SP's behaviour is shown)

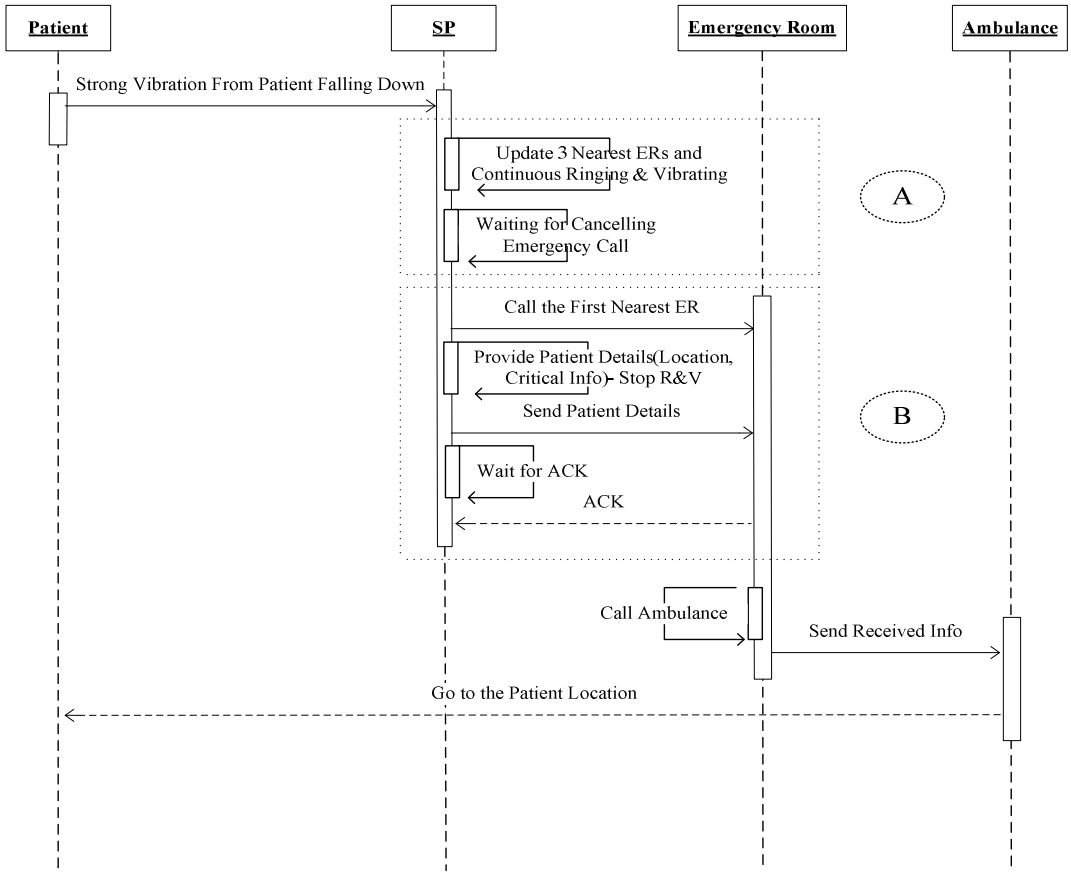


Figure 54. Automatic Call to Emergency Room for an Ambulance Scenario

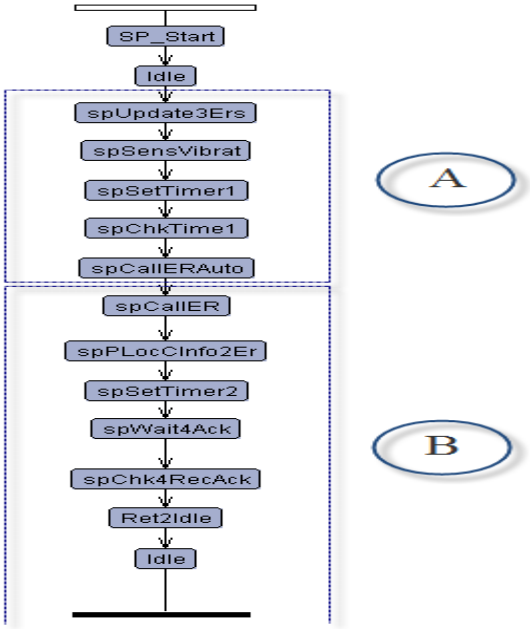


Figure 55. UPPAAL Output for the Automatic Call to an Emergency Room for an Ambulance Scenario (only the SP's behaviour is shown)

5.4 SUMMARY

In this chapter, we used UPPAAL to simulate our protocols in different health-care scenarios. We showed our expectations from the different scenarios by using *Message Sequence Charts* and compared them with the simulation results. All the results agreed with our expectations (though we did not model the human's behaviours or those of external entities such as ambulances and medical centre). This confirms that when fully implemented the SDL protocols from Chapter 4 are capable of supporting our proposed EHR framework.

Chapter 6: Conclusions

Ubiquitous access to a patient's medical records is an important aspect of caring for a patient. Poor patient data, incomplete records and unclear quality of presentation can impact on treatment and could possibly lead to a fatality. Using new technologies such as *Java SIM Cards* to address the challenges of ubiquitous access to EHRs is vital for next generation healthcare systems. While a patient's full EHR can be accessed from an *Electronic Health Records Centre* (EHRC), a partial EHR contained within a *Java SIM Card* can be used at the patient point-of-care to help quick diagnosis of a patient's problems.

By taking advantage of the *Java SIM Card* and related technologies such as *Smart Phones*, *Next Generation Networks*, *Near Field Communications*, *Public Key Infrastructure,* and *Biometric Identification*, we proposed a secure framework and communication protocols which provide a solution for ubiquitous access to EHRs without imposing major changes on existing telecommunication hardware infrastructure. Our framework and protocols make a healthcare system able to overcome weaknesses in previous proposals. Previous research in this field was concerned with using Medicare cards, a kind of *Smart Card*, as a repository of medical information at the patient point-of-care. However, this imposes some limitations on the patient's emergency medical care, including the inability to detect and inform the patient's location, call and send information to an emergency room automatically, and automate and secure interaction with the patient in order to get consent.

Once implemented, our framework and protocols will offer the full benefits of accessing an up-to-date, precise, and comprehensive medical history of a patient, whilst its mobility will provide ubiquitous access to medical and patient information everywhere it is needed. Our framework and protocols will contribute significantly to improving ubiquitous access to medical data by automating interactions between patients and healthcare providers, thus increasing patient safety, increasing quality of care, and reducing costs.

As an outline for future work, we can focus on improving, completing, and implementing the proposed protocols and the *Applet Layer* of the patient's JSC. Also

a pilot study of the whole system is another area for future work which would help to find and resolve possible problems.

Bibliography

Abraham, C., Watson, R. T., & Boudreau, M.-C. (2008). Ubiquitous access: on the front lines of patient care and safety. *Communications of the ACM, 51*(6), 95-99.

Ahn, J., Heo, J., Lim, S., & Kim, W. (2008). A Study on the Application of Patient Location Data for Ubiquitous Healthcare System based on LBS. *Proceeding of the 10th International Conference on Advanced Communication Technology.* (pp. 2140-2143).

Alhaqbani, B., & Fidge, C. (2008). Access Control Requirements for Processing Electronic Health Records. In *Business Process Management Workshops* (pp. 371-382).

Andrade, R., Wangenheim, A. v., & Bortoluzzi, M. K. (2003). Wireless and PDA: a novel strategy to access DICOM-compliant medical data on mobile devices. *International Journal of Medical Informatics, 71*(2-3), 157-163.

Arazi, B. (2009). Message Authentication in Computationally Constrained Environments. *IEEE Transactions on Mobile Computing, 8* (7), 968-974.

Articsoft. (2009). Introduction to Public Key Infrastructure. Retrieved May 10, 2009, from http://www.governmentsecurity.org/forum/index.php?showtopic=1630

Bar-El, H., Choukri, H., Naccache, D., Tunstall, M., & Whelan, C. (2006). The sorcerer¹s apprentice guide to fault attacks. *Proceedings of the IEEE, 94*(2), 370-382.

Bishop, B., Maloney, D., Wilson, P., Nader, N., Sembritzki, J., Meazzini, G., et al. (2000). US cards hold medical records. *Card Technology Today, 12*(6), 14-15.

Boswell, T. (2009). Smart card security evaluation: Community solutions to intractable problems. *Information Security Technical Report, 14*(2), 57-69.

Brandozzi, M. (1995). Model Checking of Real-time Systems and the Uppaal Tool. Retrieved 27 December, 2009, from www.dcs.warwick.ac.uk/~doron/mreport.doc

Chan, A. T. S. (2003). Integrating smart card access to Web-based medical information systems. *Proceedings of the 2003 ACM symposium on Applied computing* (pp. 246-250), Melbourne, Florida.

Chan, A. T. S., Cao , J., Chan , H., & Young , G. (2001). A web-enabled framework for smart card applications in health services. *Communications of the ACM, 44*(9), 76-82.

Chang, Y. F., Chen, C. S., & Zhou, H. (2009). Smart phone for mobile commerce. *Computer Standards & Interfaces, 31*(4), 740-747.

Chen, Z. (2000). *Technology for Smart Cards: Architecture and Programmer's Guide* Pearson.

Chenhui, Z., Huilong, D., & Xudong, L. (2008). An Integration Approach of Healthcare Information System. *International Conference on BioMedical Engineering and Informatics* (pp. 606-609).

Chin, L.-P., & Chen, J.-Y. (2006). SIM card based e-cash applications in the mobile communication system using OTA and STK technology. *IET International Conference on Wireless, Mobile and Multimedia Networks,* (pp. 1-3).

Eichelberg, M., Aden, T., Riesmeier, J., Dogac, A., & Laleci, G. B. (2005). A survey and analysis of Electronic Healthcare Record standards. *ACM Computer Survey., 37*(4), 277-315.

ETSI. (2009). Message Sequence Charts (MSC). Retrieved 05 January, 2009, from http://www.etsi.org/WebSite/Technologies/MSC.aspx

Gammel, B. M., & Ruping, J. (2005). Smart cards inside. *Proceedings of the 31st European Solid-State Circuits Conference.* (pp. 69-74).

Grassie, K. (2007). Easy handling and security make NFC a success. *Card Technology Today, 19*(10), 12-13.

Hall, E. S., Vawdrey, D. K., Knutson, C. D., & Archibald, J. K. (2003). Enabling remote access to personal electronic medical records. *Engineering in Medicine and Biology Magazine, IEEE, 22*(3), 133-139.

Halperin, D., Kohno, T., Heydt-Benjamin, T. S., Fu, K., & Maisel, W. H. (2008). Security and Privacy for Implantable Medical Devices. *IEEE Pervasive Computing, 7*(1), 30-39.

Havelund, K., Skou, A., Larsen, K. G., & Lund, K. (1997). Formal modeling and analysis of an audio/video protocol: an industrial case study using UPPAAL. *The 18th IEEE Proceedings of Real-Time Systems Symposium,* (pp. 2-13).

Haverinen, H., Asokan, N., & Maattanen, T. (2001). Authentication and key generation for mobile IP using GSM authentication and roaming. *IEEE International Conference on Communications,* (pp. 2453-2457 vol.2458).

Hessel, A. (2001). Timing Analysis of an SDL subset in Uppaal. Retrieved 5 January, 2010, from http://www.hessel.nu/sdl2xta/abstract.html

Hu, J. (2008). Mobile fingerprint template protection: Progress and open issues. *The 3rd IEEE Conference on Industrial Electronics and Applications,* (pp. 2133-2138).

Huang, L.-C., Chu, H.-C., Lien, C.-Y., Hsiao, C.-H., & Kao, T. (2009). Privacy preservation and information security protection for patients' portable electronic health records. *Computers in Biology and Medicine, 39*(9), 743-750.

Huawei, Z., & Ruixia, L. (2009). A Scheme to Improve Security of SSL. *Pacific-Asia Conference on Circuits, Communications and Systems,* (pp. 401-404).

Hung, K., & Yuan-Ting, Z. (2003). Implementation of a WAP-based telemedicine system for patient monitoring. *IEEE Transactions on Information Technology in Biomedicine, 7*(2), 101-107.

Iakovidis, I. (1998). Towards personal health record: current situation, obstacles and trends in implementation of electronic healthcare record in Europe. *International Journal of Medical Informatics, 52*(1-3), 105-115.

IETF. (2008). TLS Protocol V 1.2. Retrieved 24 November, 2009, from http://tools.ietf.org/html/rfc5246

Institute of Medicine. (2000). To err is human: Building a safer health system. Retrieved September 23, 2008, from www.nap.edu/books/0309068371/html

ISO/TR 20514. (2005). *Health informatics -- Electronic health record -- Definition, scope and context*.

ISO/TS 18308. (2004). *Health informatics -- Requirements for an electronic health record architecture*.

ITU-T Next Generation Network. (2009). Definition of Next Generation Network. Retrieved 10 May, 2009, from http://www.itu.int/ITUT/studygroups/com13/ngn2004/working_definition.html

ITU-T X.210. (1993). *ITU-T Recommendation X.210*: Open Systems Interconnection - Basic Reference Model: Conventions for the Definition of OSI Services.

ITU-T Z.100. (2007). Specification and Description Language (SDL). Recommendation Z.100. Retrieved 12 November, 2009, from http://www.itu.int/rec/T-REC-Z.100-200711-I/en

Jelekäinen, P. (2004). GSM-PKI solution enabling secure mobile communications. *International Journal of Medical Informatics, 73*(3), 317-320.

Kambourakis, G., Maglogiannis, I., & Rouskas, A. (2005). PKI-based secure mobile access to electronic health services and data. *Technology & Health Care, 13*(6), 511-526.

Katz, J. E., & Rice, R. E. (2009). Public views of mobile medical devices and services: A US national survey of consumer sentiments towards RFID healthcare technology. *International Journal of Medical Informatics, 78*(2), 104-114.

Kavadias, C. D., Koutsopoulos, K. A., Vlachos, M. P., Bourka, A., Kollias, V., & Stassinopoulos, G. (2003). A monitoring/auditing mechanism for SSL/TLS secured service sessions in Health Care Applications. *Technology & Health Care, 11*(1), 1.

Kim, D.-K., Mehta, P., & Gokhale, P. (2006). Describing access control models as design patterns using roles. *Proceedings of the 2006 conference on Pattern languages of programs, article N 11* (pp. 1-10), Portland, Oregon.

Li, C., Yang, Y.-x., & Niu, X.-x. (2006). Biometric-based personal identity-authentication system and security analysis. *The Journal of China Universities of Posts and Telecommunications, 13*(4), 43-47.

McLaughlin, L. (2007). Hospital puts medical records snapshot on smart cards. *Networkworld* Retrieved May 10, 2009, from http://www.networkworld.com/news/2007/101807-hospital-puts-medical-records-snapshot.html?page=2

Morak, J., Hayn, D., Kastner, P., Drobics, M., & Schreier, G. (2009). Near Field Communication Technology as the Key for Data Acquisition in Clinical Research. *First International Workshop on Near Field Communication,* (pp. 15-19).

NHS. (2009). UK National Health Service. Retrieved March 1, 2010, from http://www.nhs.uk/NHSEngland/thenhs/about/Pages/overview.aspx

Peersman, C., Cvetkovic, S., Griffiths, P., & Spear, H. (2000). The Global System for Mobile Communications Short Message Service. *Personal Communications, IEEE, 7*(3), 15-23.

Search Health IT. (2010). Regional Health Information Organization (RHIO). Retrieved March 1 , 2010, from http://searchhealthit.techtarget.com/definition/Regional-Health-Information-Organization-RHIO

SIGN. (2002). Report on a Recommended Referral Document. Retrieved 1 December, 2009, from http://www.sign.ac.uk/pdf/sign31.pdf

Sun Microsystems. (2009). Java Card Technology Overview. Retrieved May 10, 2009, from http://java.sun.com/javacard/overview.jsp

Tele Pak. (2009). JAVA SIM card. Retrieved 23 December, 2009, from http://www.tele-pak.com/plastic-cards/javacards.html

The Biometric Consortium. (2009). An Introduction to Biometric. Retrieved May 10, 2009, from http://www.biometrics.org/html/introduction.html

Tsai, C., Lee, G., Raab, F., Norman, G., Sohn, T., Griswold, W., et al. (2007). Usability and Feasibility of PmEB: A Mobile Phone Application for Monitoring Real Time Caloric Balance. *Mobile Networks and Applications, 12*(2), 173-184.

US Department of Health & Human Services. (1996). *Health Insurance Portability and Accountability Act*

Van Der Linden, H., Kalra, D., Hasman, A., & Talmon, J. (2009). Inter-organizational future proof EHR systems: A review of the security and privacy related issues. *International Journal of Medical Informatics, 78*(3), 141-160.

Wei-Bin, L., & Chien-Ding, L. (2008). A Cryptographic Key Management Solution for HIPAA Privacy/Security Regulations. *Information Technology in Biomedicine, IEEE Transactions on, 12*(1), 34-41.

Whatis. (2009). Electronic-Health-Record. Retrieved May 10, 2009, from http://whatis.techtarget.com/definition/electronic-health-record--ehr-.html