ROBUST TELEHEALTHCARE SYSTEM: NFC-BASED APPROACH

By

Divyashikha Sethia

A DISSERTATION

Submitted to
Delhi Technological University
in partial fulfillment of the requirements
for the degree of

Computer Science – Doctor of Philosophy

2019

CERTIFICATE

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of Engineering)
Bawana Road, Delhi - 110042

This is to certify that the thesis entitled *"Robust Telehealthcare System: NFC-Based Approach"* being submitted by **Ms Divyashikha Sethia (Reg. No: 2K12/PhD/CO/04)** for the award of degree of Doctor of Philosophy to the Delhi Technological University is based on the original research work carried out by her. She has worked under our supervision and has fulfilled the requirements that to our knowledge have reached the requisite standard for the submission of this thesis. It is further certified that the work embodied in this thesis has neither partially nor fully submitted to any other university or institution for the award of any degree or diploma

Prof. Daya Gupta
(Supervisor)
Department of Computer Science and Engineering
Delhi Technological University, Delhi

Prof. Huzur Saran
(Co-Supervisor)
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi

DECLARATION

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi - 110042

I, **Divyashikha Sethia**, a part time research student **(Reg. No: 2K12/PhD/CO/04)**, hereby declare that the thesis entitled *"Robust Telehealthcare System: NFC-Based Approach"* which is being submitted for the award of the degree of Doctor of Philosophy in Computer Engineering, is a record of bonafide research work carried out by me in the Department of Computer Science and Engineering, Delhi Technological University. I further declare that the work presented in the thesis has not been submitted to any university or institution for the award of any diploma or degree.

Divyashikha Sethia
(Candidate)
Department of Computer Science and Engineering
Delhi Technological University

ACKNOWLEDGEMENT

I am immensely grateful to the Supreme Lord for inspiring me and His continuous grace and mercy, throughout the journey of my PhD. With His grace, I received all the help from the right persons at the right time.

I would like to express my sincere gratitude to my supervisors **Prof (Dr) Daya Gupta** and **Prof (Dr) Huzur Saran** for providing me with the opportunity to pursue this research work. They helped me in positioning my research topics, resolving research challenges, providing constant selfless, painstaking guidance, support and encouragement. My sincere thanks to **Prof Daya Gupta** for her invaluable inputs and motivation to overcome obstacles. I have no words to express my profound sense of respect and gratitude for her constant encouragement, help and supportive attitude during the entire period of this research. My heartfelt gratitude to **Prof Huzur Saran** for introducing me to the new research area of this study and continuously supporting with the inculcation of new ideas. His scientific temperament enriched my exploration in research and inspired me to aim towards excellence. His expertise, constant support and generosity have impacted my thinking by nourishing my intellectual curiosity. Without his help, I would not have reached this destination.

I am sincerely thankful to **Prof (Dr) Rajni Jindal** for her kind support and guidance at DTU. I am grateful for research discussions with **Dr Shweta Agrawal** from IIT Madras, **Prof (Dr) Kolin Paul** from IIT Delhi, **Dr Amarjeet Singh**, and **Dr Ponnurangam Kumaraguru** from IIIT Delhi and **Dr Shivaprasad** from Applied Mathematics, DTU. The discussions had valuable technical inputs, which helped me evolve the proposed ideas and work in this study.

My sincere thanks to my friends **Smita, Pallavi, Richa, Shruti, and Navriti** and colleagues at DTU for helping me through the tough times of hard work. I must also express gratitude to the other staff members **Rajat, Saurabh and Ashok** in the Department of Computer Science, DTU for their prompt support for all administrative work. I would like to thank all my faculty colleagues at DTU who supported me directly or indirectly.

All my accomplishments do not make sense till I name my better half, **Prashant Sethia** and my daughters **Amika Sethia** and **Paavani Sethia**. Thank you for encouraging me in all of my pursuits and inspiring me to follow my dreams. I had to spend valuable time at home for research. My family supported me to accomplish all the work morally and emotionally. I saw my daughters grow into two beautiful and intelligent young girls. We had constant common time for studying together. My most profound sense of gratitude to my parents, parents-in-law for their full support in my thesis. I am especially thankful to my extended family members to have complete trust in me and patiently wait to see my research reach this destination.

HOWEVER, THE ULTIMATE GRACE WAS HIS, WITHOUT WHICH NOTHING COULD HAVE BEEN ACHIEVED.

ABSTRACT

Patients with dispersed health records face the challenge of securely accessing and maintaining readily available health history. Dispersed health records cause difficulty in mobility across different hospitals and seeking timely diagnosis and treatment. The cloud-based systems have higher challenges for security and privacy and are not 24/7 available. The portable-based systems are restricted to a specific health provider and may have limitations for space and access. There is a growing usage of mobile devices due to their improved computational and storage capabilities. Hence, they may be useful for health record management. However, current mobile-based health record systems are limited for either remote access to the cloud-based repository or to store records for only offline backup. None of the current health record management solutions fulfils patient mobility, with the aggregation of updated health records, secure and direct access for reading and writing, and maintenance of provenance of health records.

This thesis proposes a next-generation smart health record management system with secure NFC-enabled mobile devices to fulfil the requirements for patient mobility across hospitals. First, the thesis proposes a system design for the smart portable mobile-based health record management system to assist patient mobility across hospitals. It retains *Secure Mobility-Assisted PortablE (S-MAPLE)* health folder on the patient's mobile device for storing dispersed health records. It can be accessed as a contactless card by the health professional's mobile device using low energy wireless interfaces, such as Near Field Communication (NFC)-based Host Card Emulation (HCE) or Bluetooth. NFC provides proof-of-locality and makes eavesdropping and man-in-the-middle attacks difficult. The patients can also view their health records on the health folder locally on their mobile devices. A hardware tamper-resistant Secure Element (SE) in the form or a microSD or SIM Card retains cryptographic credentials and also performs cryptographic computations. A cloud-based HealthSecure service helps manage credentials, unique identity and backup of the health data to refurbish the health folder in case of loss or theft of the patient's mobile device. A variation of the Ciphertext-Policy Attribute-Based Encryption (CP-ABE) scheme secures all health records for directly sharing them with multiple health providers using Role-based Access Control (RBAC) over the NFC interface.

Second, this thesis proposes the essential security and threat requirements. The thesis also suggests the security solutions comprising of secure storage, provenance of health data, mutual authentication with trust between devices, and selective access with scalable revocation. We propose two novel protocols for secure healthcare access from portable devices. *NFC SE-based Mutual Authentication and Attestation (NSE-AA)* protocol provides end-to-end symmetric lightweight mutual authentication and remote attestation between the SEs of the two mobile devices. *Scalable Proxy-based Immediate Revocation for CP-ABE (SPIRC)* scheme improves the Bethencourt's CP-ABE scheme for scalable revocation and uninterrupted access to portable devices, without the requirement of any prior revocation list, re-encryption and re-distribution of keys.

Third, this thesis presents a detailed security analysis of the security framework with an emphasis on the two proposed security protocols. We prove that the NSE-AA protocol is secure using protocol simulations on Automated Validation of Internet Security Protocols and Applications (AVISPA) tool and a formal security proof using the Real-Or-Random (ROR) model. We also prove that the SPIRC scheme is secure from CPA (Chosen Plaintext Attacks) in a security game.

Fourth, this thesis presents the details of the implementation and performance comparison of a prototype for the proposed health record system using mid-range Android devices with NFC and Bluetooth. The protocols are evaluated for their performance and compared qualitatively and quantitatively with the related schemes. The results indicate that the overheads of the security framework are acceptable and that the proposed protocols have improved performance.

The contactless S-MAPLE health folder can assist in the patient mobility across different hospitals with updated, secure and readily available health history. It can help improve the quality of healthcare management by providing timely diagnosis and treatment to the patients.

Keywords health record, mobility, NFC, HCE, TPM, Mutual Authentication, Attribute-based Encryption, Attestation, Secure Element

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

KEY TO ABBREVIATIONS

ABE Attribute-Based Encryption

AID Application Identifier

AIK Attestation Identity Key

APDU Application Protocol Data Unit

AVISPA Automated Validation of Internet Security Protocols and Applications

BTG Break the Glass

CAD Card Acceptance Device

CL-AtSe Constraint-Logic-based Attack Searcher

CPA Chosen Plaintext Attack

CP-ABE Ciphertext-Policy Attribute-Based Encryption

CFI Control-Flow Integrity

DAA Direct Anonymous Attestation

E2EE End-to-end encryption

EDI Electronic Data Interchange

EHR Electronic Health Record

EK Endorsement Key

EMR Electronic Medical Records

FIDO Fast Identity Online

FHIR Fast Healthcare Interoperability Resources

FTDI Future Technology Devices International

GPS Global Positioning System

HAMA HCE with Asymmetric Mutual Authentication

HAPI HL7 Application Programming Interface

HCE Host Card Emulation

HIPAA Health Insurance Portability and Accountability Act

HIS Hospital Information Systems

HL7 Health Level Seven

HLPSL High Level Protocol Specification

HRB Health Record Bank

IMEI International Mobile Equipment Identifier

JVCM Java Card Virtual Machine

KP-ABE Key-Policy Attribute-Based Encryption

LLCP Logical Link Control Protocol

JSON JavaScript Object Notation

MAC Message Authentication Codes

MAT Mobile Attestation Token

mfp More Fragment Packet

MITM Man in the Middle

MOONACS Mobile On-Offline NFC-based Physical Access Control System

NDEF NFC Data Exchange Format

NFC Near Field Communication

NFC CLF NFC Contactless Front-end

NFCIP-1 Near Field Communication Interface and Protocol

NFC-WI NFC Wired Interface

NHS National Health Service

NSE-AA NFC SE-based Mutual Authentication and Attestation

OPD Out Patient Department

OBX Observation Section

OFMC On-the-fly Model-Checker

ORU Observation Result

OTA Over The Air

OTP One-Time Password

PAN Primary Account Number

PCR Platform Configuration Registers

PGHD Patient Generated Health Data

PHR Personal Health Record

PHI Personal Health Information

PID Patient Identifier

PIRATTE Proxy-based Immediate Revocation of ATTribute-based Encryption

POS Point Of Sale

RACS Role-based Access Control scheme

RIM Research In Motion

RBAC Role-Based Access Control

RFID Radio Frequency Identification

RFCOMM Radio frequency communication

ROR Real-Or-Random

SATMC SAT-based Model-Checker

SCH Secure Channel service

SE Secure Element

SIM Subscriber Identification Module

SBC Single Board Computer

S-MAPLE Secure Mobility Assisted PortabLE

SMC Secure Memory Card

SML Stored Measurement Log

SNEP Simple NDEF Exchange Protocol

SPAN Security Protocol ANimator for AVISPA

SPIRC Scalable Proxy-based Immediate Revocation for CP-ABE

SPT Secure Portable Token

SSE Shared Secret service

SWP Single Wire Protocol

TA4SP Tree Automata based on Automatic Approximations for the Analysis of Security Protocols

TCA Trusted Certifying Authority

TCG Trusted Computing Group

TEE Trusted Execution Environment

THCE Trusted Host-based Card Emulation

TLS Transport Layer Security

TMT Taiwan Electronic Medical Record Template

TNF Type Name Field

TPM Trusted Platform Module

UICC Universal Integrated Circuit Card

VS Virtual Server

XML Extensible Markup Language

CHAPTER 1

INTRODUCTION

1.1 Health Record Management

Health records must be maintained properly for a complete health history of a patient to seek correct and timely medical diagnosis and treatment. There are various forms of digital health records, such as *Electronic Health Record (EHR)* and *Personal Health Record (PHR)*.

EHR is a comprehensive repository of full clinical information for a patient from various sources, such as physicians, laboratories, and patients. The EHR can be defined as digitally stored healthcare information about an individual's lifetime. As per Eichelberg et al. [41] EHR has the objective of supporting continuity of care, education and research, ensuring confidentiality at all times, and enabling sharing and integration of health records across multiple providers. EHR is traditionally maintained by the health providers, such as Kaiser Permanente and Veterans Health Administration in the United States (US) [138].

PHR is a personal recording of doctor's visits, medications, claims, and other information, which may be useful for quick medical history. Such a record is created and maintained by a patient individually. Since PHRs are personally managed, the potential disadvantage of PHRs is that the correctness and provenance of the data may be unreliable for a medical professional. However, according to Detmer et al. [35], PHRs have several advantages, such as ease of management of chronic illness and maintaining lifelong health history independent of the hospitals. The PHRs are of the following types [35, 138]:

- **Standalone PHR:** These are manually populated records that are personally managed by patients, such as paper-based, computer-based, or web-based. Standalone PHRs assist patients to share information readily with the health providers. A web-based PHR is maintained by a vendor to store the details, and they provide web interfaces to access the information from a browser.

- **Integrated PHRs:** These comprise of patient health information from sources, such as EHRs, insurance claims, pharmacy data, and home diagnostics. Integrated PHRs provide complete health history of a patient. They do not require manual entry and comprise of provider-based information and hence have higher provenance of data and reduced errors.
- **Tethered PHRs:** These are created by a health provider, which consist of claims data (that may include laboratory and pharmacy information). Patients can access the Tethered PHRs through a secure web portal. Some hospitals have made clinical information available via web portals to patients (or parents of young patients) with certain diagnoses. The health provider maintains a Tethered PHR, and patients can access only a portion of their clinical data under certain rules. A portable device cannot be used to store the Tethered PHR, and hence, the patients may not be able to access them until they change job or service provider.

A robust healthcare infrastructure must have sufficient storage and efficient access for the health records to provide timely diagnosis and treatment of patients. The benefits of such a robust healthcare data infrastructure are [124]:

- Flexible access to patient health records.
- Interoperable access to patient health records.
- Reduced errors in patient health records and clinical procedures.
- Reduced redundant information in health records, such as tests and diagnosis.
- Complete and accurate patient health history.
- Better communication between patients and healthcare providers.
- Enable access to records by authorized health professionals and detect frauds.

Researchers have been looking into the issues of dispersed health records and securing them on various platforms. A patient may need to retain health records from various sources including EHRs, insurance claims, pharmacy data, and home diagnostics. For patients visiting various hospitals, an Integrated PHR may have advantages such as:

- Availability of complete patient health history at the point of care.

- Real-time health records from personal health monitoring devices.
- Reduced cost and improved healthcare services by enabling remote monitoring of patient records and efficient time spent by physicians through collaborative health history.

There are various examples of Integrated PHRs, such as Veterans Affairs health record [81], NHS National Service's Scotland [111], and Denmark's health portal [33]. The Health Record Bank (HRB) [58] is a private independent organisation for securely storing and managing health records from multiple sources. HRB assures secure access to health records and protects the patient's privacy. *However, these systems do not support patient mobility across different health providers and geographies.*

1.2 Open Challenges for Patient Mobility Across Hospitals

This thesis looks into the challenges of a patient with dispersed health records for secure, readily available complete health history, timely medical diagnosis, and treatment. The following sections discuss the open challenges.

1.2.1 Dispersed Health Records

Many patients face the challenge of dispersed health records across various hospitals or other personal health monitoring devices. It becomes difficult to aggregate and retain a complete health history for patients due to the lack of interoperability of health record management systems.

In developing countries like India, there is a lack of healthcare policies and infrastructure for a centralized health system. Hence, people visit different hospitals in urban cities to seek specialized consultations and take second opinions for a reliable diagnosis. Hence, there is a burden on the patient to maintain all records, such as lab tests, medications, and bills personally for complete health history. Usually, patients retain paper-based records with them due to lack of digitisation and centralized health records. Paper-based health records may be unreliable due to issues, such as illegible old records, loss of health information in case the paper-based records are not retained properly.

Health management systems in developed countries are well established and have patients registered to particular healthcare or insurance policy, such as *National Health Service (NHS)* system in U.K [108] and *Taiwan Electronic Medical Record Template (TMT)* in Taiwan [29]. However, a patient may seek treatment from different hospitals, such as in the following cases:

- Case of citizen mobility as in European countries, for work and tourism across various states and countries.
- Emergency where patients may land in a hospital that is not under their health policy.

Hence, a patient's health records can be dispersed on different *Hospital Information Systems (HIS)* of health providers. Typically most HIS are cloud-based systems, which have certain challenges as discussed below.

Cloud-based Health Management System- A remote server maintains all health records that are accessible through a web portal by the medical professionals and patients. Spanakis et al. [137] proposed *MyHealthAvatar (MHA)*, which is a personal cloud-based digital health-related collection bag to aggregate heterogeneous health information. It has interfaces for accessing, collecting and sharing long-term multilevel personal health, such as clinical data, genetic data, medical sensor data and devices, human behaviour data, and activity data for clinical data analysis, prediction, and prevention for the patients. Health records can be stored centrally or decentrally as in the *Dutch Electronic Patient Dossier (EPD) [112]* to reduce security attacks. There are some public health servers used by patients for managing PHRs, such as Microsoft HealthVault [101], In Case of Emergency PHR Mobile [119], and Indivo system proposed by Wang et al. [149]. Some countries, such as UK [111], Denmark [112], and Taiwan [29] also store health records on centralized servers. The patient can control health information, emergency data, security and privacy on these portals. *However, there are few challenges for the cloud-based health systems due to which they cannot be used for securely storing dispersed health records and patient mobility across different hospitals. The challenges are listed below*:

- When the records are shared with different medical practitioners or personally updated by a patient, there are higher risks, challenges for errors, and lack of provenance of health records.
- Each of the centralized systems is independent and well-defined. However, due to different standards and policies [78, 45], the health records cannot be shared across hospitals and hence cannot help in patient mobility across different hospitals.
- These systems require high infrastructure.
- In case of lack of connectivity they cannot provide 24/7 availability of health records.
- There are higher challenges for security and privacy threats [1].
- There are higher risks for the usage of health records for data mining without knowledge of patients and hospitals.

Hence, patients must retain their dispersed health records on secure portable devices such as mobile devices for high availability of their health records. *However, the existing portable and mobile-based systems provide only backup of health records and cannot be accessed directly for reading and writing of health records.* Smart cards issued by certain healthcare systems provide secure, portable, and readily available health records. *However, they have limited information, such as emergency details, and specific information for the related healthcare system due to lack of space. Smart cards also have a limitation that they cannot provide instant visualization of health records without an external reader device.*

None of the existing solutions using the cloud, portable devices and smartphones can provide readily available aggregated health history for patient mobility across hospitals. Table 1.1 describes the comparison between different health record management systems.

Issue	**Cloud-based**	**Portable-based**	**Mobile-based**
Availability	Limited	24/7	24/7
Emergency	Limited	Yes	Yes
Storage	Large	Limited	Medium
Security requirements	High	Medium	Medium
Direct access	Yes	No	No

Table 1.1: Existing Health Record Management Techniques

Hence, there is a need for a portable device with sufficient space and ease of access, which can be directly accessed for read and write by a health professional. Current mobile devices have improved computational and storage capabilities, and hence, they can provide a portable health management system for patient mobility across hospitals. A portable health device can aggregate health history from dispersed hospitals and provide the following benefits:

- Updated health information for seeking timely diagnosis and treatment
- High availability even in disconnected networks
- Ease of access

1.2.2 Security and Privacy of Health Records

Portable devices must be protected from security and privacy threats to provide correct treatment to patients and also protect their privacy [48, 62]. Besides securing health records, there may be additional challenges for integration of health records due to different laws for security and privacy of data [156]. The European data protection legislation [31] divides the health records into the following categories [156]:

- Non-personal Data
- Personal but Non-sensitive Data
- Sensitive Personal Data

The healthcare professionals must access different health data based on their roles as discussed later.

The health records in practice are referred as *Personal Health Information (PHI)* and are protected by health laws, such as *Health Insurance Portability and Accountability Act (HIPAA)* [78] in the United States (US) and *European regulations for healthcare data protection* [45]. HIPAA was established in 1996 in the US, and it defines the privacy rights of a patient. HIPAA provides guidelines for who can access the patient's personal healthcare information and security measures for administrative, physical, and technical domains in the information system.

Several survey research papers address various issues for security and privacy of health records [70, 62, 8, 143, 48, 1, 83]. It is essential to preserve the privacy and security of health records on a patient's portable device to prevent any disclosure of information [70]. Security and privacy of health records are vital for patients so that they may disclose their ailments and seek proper care. According to the United States Department of Health and Human Services, around two million Americans do not disclose their mental illnesses due to privacy concerns [164].

The health record system must retain the *confidentiality* of health records. Health professionals who receive a patient's information must respect their *privacy* and keep it undisclosed [18, 1]. The system must also retain the *integrity* of health records. The system must retain the validity and accuracy of health records and protect patient's rights. If the health history of a patient is allowed for direct sharing for reading and writing with multiple stakeholders, it is important to maintain *provenance* of health records. Only then the physicians in different hospitals will be able to rely on the aggregated health history of the patient.

According to Avalanche et al. "Health information *privacy* is an individual's right to control the acquisition uses, or disclosures of his or her identifiable health data" [18]. There must be selective protection of privacy and access to trusted and authorized healthcare professionals. The adversary could be patient himself, insider and outsider. Privacy-related threats can be categorised as identity threats, access threats, and disclosure threats.

The health system must retain the *anonymity* of the user. It is essential that the contents of health records and communication must not be accessible and correlated with patient's identity to an adversary.

The health records and credentials must be stored on *secure storage* so that adversary cannot access them easily. The health records must also have *high availability* to allow physicians to be aware of the past health history; provide timely diagnosis and treatment to the patient.

It is important that the healthcare providers can justify and take responsibility for their actions, such as prescription of medications, diagnosis and treatment [43].

Patient may want to restrict the discloser of health records with a health provider. An insider who is an adversary with valid authorization to the system may peek inside the health records for curiosity. It may lead to embarrassment to a patient. According to Samarati and Vimercati [131], access control policies are classified as follows:

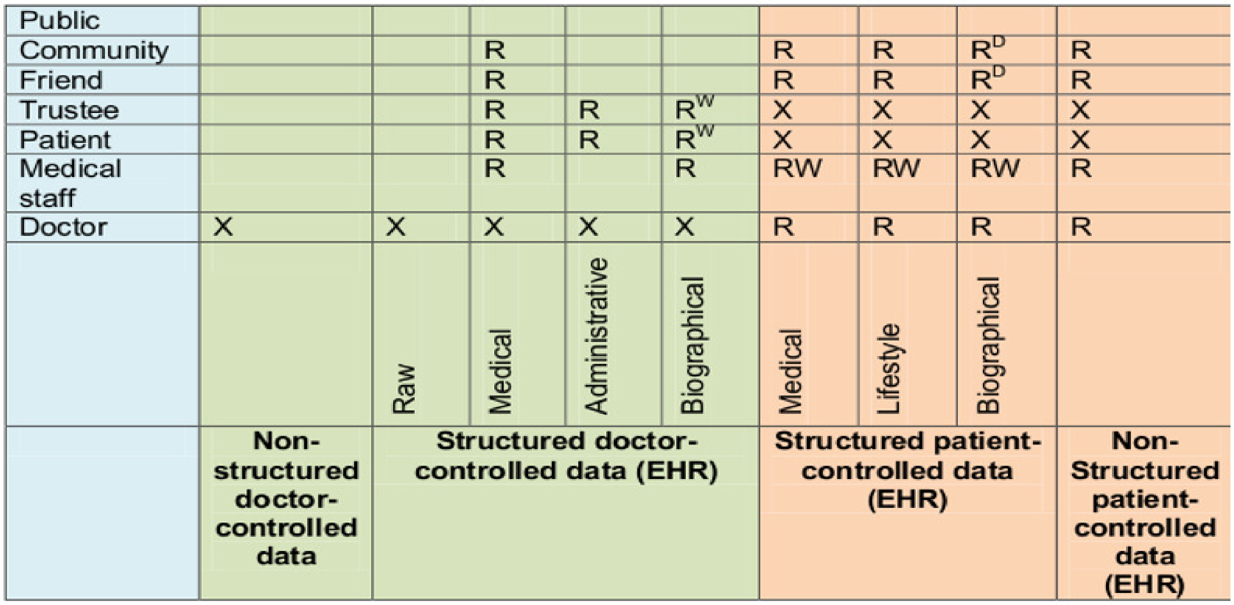
- *Discretionary Access Control:* Grants access based on the user's identity, and access rules, which state if the user is allowed access or not.
- *Mandatory Access Control:* Grants access based on the regulations managed by a central authority.
- *Role-based Access Control (RBAC):* Grants access based on the roles of the users and on rules that control what accesses are allowed to which roles.

There must be selective access control for health record access since there are several stakeholders with different roles. Each stakeholder must access information that is relevant and must protect the patient's privacy. Most research papers recommend using RBAC policy management. It can be overridden in case of an emergency using schemes such as *Ciphertext-Policy Attribute-Based Encryption (CP-ABE)* scheme proposed by Bethencourt et al. [21]. Sharing of health records must require verification of the source and permissions for total or selective sharing.

The medical stakeholders can be categorised into different roles, such as healthcare professionals, Patient, Trustee, Friend, community, and public [156]. When the health records are populated by multiple stakeholders, its reliability decreases and the risk of exposure increases. The health records can be of several types:

- Controlled or generated by a doctor.
- Generated through medical devices and laboratories (EHR).
- Patient-controlled created by the patient or uploaded by medical staff independent from physicians, like dietitians, physiotherapists (PHR).

Figure 1.1 illustrates different users and their rights for accessing different categories of health information. All health professionals must access the information selectively based on their roles and access rights. It is essential to maintain provenance of health records when a patient directly shares them with health professionals for direct read and write. Only then physicians in different hospitals can rely on the patient's aggregated health history. Alshehri and Raj [8] propose a combination of RBAC (Role-Based Access Control) and ABAC (Attribute-Based Access Control) methods for secure access to health information.



Public									
Community			R			R	R	R^D	R
Friend			R			R	R	R^D	R
Trustee			R	R	R^W	X	X	X	X
Patient			R	R	R^W	X	X	X	X
Medical staff			R		R	RW	RW	RW	R
Doctor	X	X	X	X	X	R	R	R	R
		Raw	Medical	Administrative	Biographical	Medical	Lifestyle	Biographical	
	Non-structured doctor-controlled data	**Structured doctor-controlled data (EHR)**				**Structured patient-controlled data (EHR)**			**Non-Structured patient-controlled data (EHR)**

Figure 1.1: User Rights: X-no access, R-read access, W-write access [156]

An *audit log* should store all events of access to the health records. The logs may be examined later to find if access and data transactions were appropriate and can hold an adversary accountable for violating a policy. An audit trail can help patients determine who has accessed their health records, what information has been accessed, for how long, and for what purpose. Patients must have information related to the creation of their health records, specific instances of the usage, events to update and delete parts of the health records.

The health records must be available with consent except for *emergency* personnel to assist the patient in the time of an emergency. Various methods can assist in emergency access to health records, such as the use of private-keys, smart card usage, emergency responder, and break-glass techniques.

There must be a provision to *search* the health information from the encrypted data through techniques, such as proxy encryption and public-key encryption. The queries can be answered in an encrypted format and can be decrypted with specific keys.

It is essential to *authenticate* the patient as well as the health provider, to assure the right treatment is given to the right patient. PIN and hardware tamper-resistant smart cards can be used to authenticate the related devices. There must be authentication of the medical professional or reader devices using a digital signature with PKI technique, or group signatures for signing a health procedure. It is also essential to ensure that devices have *trustful states* and are not prone to malware.

An adversary may access health information from a stolen device. Any disclosure of security keys can make health records vulnerable to manipulation. Proper key management, distribution, storage, and *revocation* techniques must be used. There must be a provision to prevent sharing of health information with healthcare providers who have been removed due to a breach of trust. Patients may wish to *delegate* authority to a friend or family member to access their health records temporarily with revocation. For example, patients may delegate for a secure report collection in case of their absence. There must be a provision to delegate part of cryptographic credentials to a trusted person, such as a family member, friend, or a colleague. Patients/trusted server must revoke the credentials delegated after usage.

The existing portable health record management schemes do not address issues for the direct real-time update of portable health records with selective access, secure storage of keys, and trustful state of the reader devices. They also do not consider the proof-of-locality of a reader due to which there are higher chances of threats, such as the MITM and eavesdropping.

1.3 Requirements for Patient Mobility Across Hospitals

The above open challenges for dispersed health records suggest that a secure portable health record management system must be used to aggregate health history for patient mobility across hospitals. The portable health record system must fulfil the following requirements:

- **R1: Aggregation-** It must store dispersed health records from different hospitals for a complete health history of a patient in a standard health format.
- **R2: Up to date-** Besides maintaining a copy of the health records on the local HIS, the medical professional must also directly update it on the portable device and keep it up to date.
- **R3: Usability across hospitals-** A patient must be able to take the device to different hospitals for direct reading and writing of records and aggregate the health records to form a complete health history with provenance.
- **R4: Availability-** The complete health history must be readily available with patients and directly accessible from their portable health device. The health history must be accessible by an authorized medical professional for timely medical diagnosis and treatment, especially in case of an emergency, chronic ailments, and patients who travel across different places.
- **R5: Easy Accessibility-** The health records must be accessible directly with the proximity of the reader device and the portable device.
- **R6: Selective Access-** The portable device must retain different types of health records that must be accessible directly to read and write by different medical professionals, such as a physician, nurse, lab technician, and pharmacist using selective RBAC as illustrated in Figure 1.2.
- **R7: Security and Privacy-** As per the existing research work, discussed previously in Section 1.2.2, it is essential to secure the health records and maintain patient's privacy. There must be a provision for secure storage and adherence to different laws for security and privacy to aggregate health records from different healthcare providers [156]. It is also essential to maintain provenance of health records so that medical professionals can refer to them reliably. The portable device must also assist in the reliable identification of authorized patients and medical professionals. The security framework must provide confidentiality, integrity and availability.

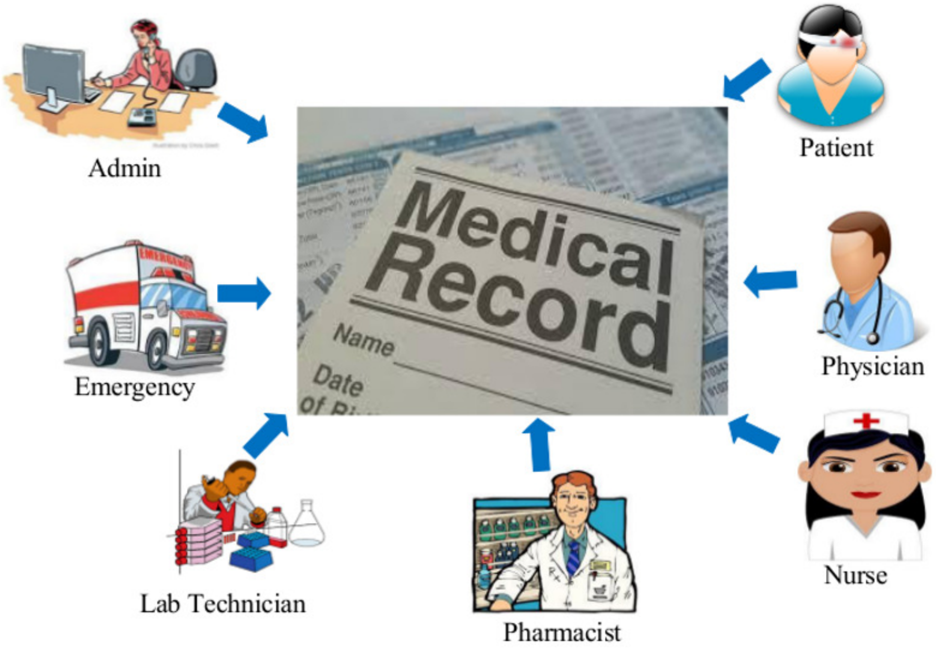


Figure 1.2: Selective Access by Different Health Professionals

1.4 Existing Techniques and Research Gaps

This thesis considers portable health record management systems and communication techniques for ease of access, such as low-energy wireless interfaces like *Near Field Communication (NFC)* [32] for patient mobility across hospitals. This thesis also looks into security issues of secure storage, provenance of health records, selective access and authentication, and trust with NFC for accessing the health records. The following sections present the existing techniques and their limitations.

1.4.1 Portable Health Record Management Systems

As discussed in Section 1.2.1, a patient must have a portable health device for patient mobility in hospitals with easy and readily available health records. However, the existing health record management systems have limitations, as discussed below.

- **Portable Devices for Health Record Management:** These systems can improve health management through readily available information, such as medications, allergies, and adverse reaction. Devices can be a USB stick, such as Personal HealthKey [155], MedicAlert e-HealthKey [99], key chains, bracelets, and smart cards. Emergency personnel may access the USB flash drive by inserting it into a USB port. *However, USB-based devices may be a security threat to hospital computers [155]. Also, the information can be accessed only over the USB interface and hence, it is not readily available on a personal device with a display screen for the patient, such as from a mobile-based hand-held device*.

Certain health providers give smart card-based health card to patients for readily available health records. They retain health records securely and allow only an authorized reader to access information reliably. Some countries use smart cards for health information, such as Sesam Vitale smart cards in France [51], Germany [23], UK [108], Taiwan [29], and Rashtriya Bima Yojna in India [122]. The smart cards help for identification, basic health information, seeking prescriptions, and hospital admissions. *However, the smart cards have limited space and cannot provide visualization of health records due to lack of a display screen.*

MyCareCard proposed by Rybynok et al. [130], is a USB-based that device contains updated health records for only offline access and limitations of a USB device. A Poket Doktor System proposed by Hall et al. [64], is a large spaced Bluetooth-enabled smart card with Radio Frequency Identification (RFID) interface to automate Bluetooth for sharing health records with a medical professional. This system is nearest to our work to provide ease of access through device proximity and availability of health records. *However, it lacks aggregation of health records from different sources and selective RBAC. Secure Portable Token (SPT)* proposed by Anciaux et al. [12], is a Tamper-resistant health folder, which combines the security of a smart card with the storage capacity of a USB key for storing health records. *However, it does not support patient mobility across hospitals and has limitations of a USB device.*

- **Mobile Devices for Health Record Management :**

With the growing use of mobile devices across the globe and improvement in their computational and storage capabilities, they can be used to aggregate health information from body sensors [95] or retain secure portable health records. *However, as per the previous research papers [6, 40, 37, 83]: current mobile-based health records systems have the following limitations:*

- ***Backup of Health Records:*** *Mobile devices have been used only for the backup of health records and offline access. All records are updated directly on the cloud and maybe only backed up later on the patient's mobile device. Hence, records on the mobile device are not up to date, especially in a disconnected network such as, in the remote areas.*
- ***Lack of Provenance:*** *None of the existing mobile-based health record techniques maintain provenance of health records and hence are considered unreliable by physicians.*

The penetration of mobile devices is rising in emerging countries, such as India. They can provide solutions for smart health record management systems, which can provide patient mobility across different hospitals for emergency care and travellers with readily available up to date health history.

Akinyele et al. [6] presented an iPhone-based application *iHealthEMR* to store and backup health records. The health records are encrypted using Bethencourt et al.'s CP-ABE scheme [21] for selective access. *However, the scheme is not suitable for a portable device to support scalable user revocation (protection from several malicious readers) as discussed in Section 1.4.4. The mobile device only provide backup and cannot be updated directly.*

Doukas et al. [40] proposed an Android-based mobile-based application to access the medical images from the cloud and does not provide any health record storage system. Dmitrienko et al. [37] proposed a TruWallet application for a Nokia device. It uses a security kernel, trusted hardware for application isolation and credential storage with a virtual machine environment for securing health records. Ahmed and Ahamad [5] proposed a scheme to secure health applications on mobile devices using Taintdroid on Android-based mobile devices for tracking the flow of sensitive health information. *However, it is difficult to implement these schemes [37, 5] in non-rooted mobile devices since it requires kernel access.*

Table 1.2 describes the comparison of the existing portable health record schemes. *None of the prior schemes satisfies all requirements R1-R7 for patient mobility across different hospitals. Existing personal portable devices and mobile solutions lack support for a direct update to the devices to retain secure health history with provenance of health records. There is a growing penetration of mobile devices for a connected world. They have improved computation capabilities, and support for low energy wireless interfaces for ease of access and direct updates, such as Near Field Communication (NFC) and Bluetooth as discussed in Section 1.4.3. Hence, mobile devices can also assist for a smart health record management system for patient mobility across hospitals.*

Table 1.2: Comparison of Portable Health Record Management Systems

Requirements	**MyCareCard [130]**	**Poket Blue [64]**	**SPT health folder [12]**	**iHealthEMR iPhone app [6]**	**@HealthCloud Android app [40]**
R1:Aggregation	N	N	N	N	N
R2:Up to date	Y	N	N	N	N
R3:Usability across hospitals	N	N	N	N	N
R4:Availability	Y	Y	Y	Y	Y
R5:Easy Accessibility	N	Y	N	Y	N
R6:Selective Access	N	N	N	Y	N
R7:Security and Privacy	N	Y	N	N	Y

1.4.2 Secure Storage

Smart cards provide a secure health record management system [79]. They also have an advantage for direct access to reading and writing and robust security. NFC-enabled mobile devices have a component known as Secure Element (SE) [127]. The NFC-enable mobile devices can be used as a contactless card using either hardware card emulation using SEs or software-based card emulation using the Host Card Emulation (HCE) mode as discussed in Section 2.4.1.1. The SEs can also provide secure storage and cryptographic computations as a smart card. Smart cards have been used to secure health record management systems, such as by Kardas et al. [79]. However, the smart card-based systems are typically for a specific health provider and have limited space, which can retain small health information for a patient. *The secure mobile-based health record management applications can use NFC-based HCE mode and SEs for use smart card-based Secure Elements (SE) [127] in the form factors of microSD cards and SIM cards for a contactless card with sufficient space on a mobile device. We have not come across any SE-based or HCE-based health record management system to the best of our knowledge.*

1.4.3 RFID and NFC based Healthcare Applications

Many healthcare applications use proximity wireless communication interfaces such as *Radio Frequency Identification (RFID)* and *Near Field Communication (NFC)* [32] to improve the healthcare workflow.

Radio Frequency Identification (RFID) is a technology in which radio waves to gather digital information stored in RFID tags at a distance of 30 cm to 2 m [85]. It uses an RFID identifier (a tag, also called a responder), which can be accessed by an RFID reader. Near Field Communication (NFC) is an RFID-based technology that enables short-range wireless information exchange with a distance of 0-20 centimetres [32]. NFC is more secure than RFID because it has a short range for reading as compared to RFID, and thus makes eavesdropping much harder.

RFID has several applications, such as accessing healthcare sensor devices [11], improving healthcare workflow [26], providing an IoT-based medicine system using Bio Sensors [160]. It helps in access control systems for improving hospitals safety and alerting when a patient is leaving the hospital without permission or monitoring temperature in different situations.

NFC has added advantage for security due to the proximity of devices and has been used for several healthcare applications, such as:

- Application for the identification of patients for improved public health [98].
- Help reduce errors in health flow [147, 85].
- Assist for prescription of a drug [148].

- Help in the ease of access for exchange of health records in a distributed health management system [3].

An NFC-based mobile device can operate in various modes, such as Reader Writer mode, Peer-to-Peer mode, Card Emulation and *Host Card Emulation (HCE)* modes [127]. The first two modes use the insecure *NFC Data Exchange Format (NDEF)* messages. The Card Emulation mode has limited space to retain information. The NFC-based HCE mode has several advantages over the other modes, as described in Table 2.1. Hence, HCE mode can be explored on mobile devices for the support of a contactless smart card and provision for bidirectional communication for security handshake.

None of the existing NFC applications have used NFC for accessing health records for direct read and write from a portable health device. Although several financial applications use the HCE mode, it has not been used for healthcare applications to the best of our knowledge. The HCE mode can enhance healthcare applications with support for bidirectional communication and ease of openness to developers.

1.4.3.1 Authentication and Attestation over NFC

Portable health devices must assist secure access to authorized health professionals. NFC can provide proof-of-locality for secure access. NFC has an advantage that due to proximity, it is hard to perform eavesdropping and MITM attacks. However, according to Madlmayr et al. [96] NFC uses an untrusted communication channel and does not ensure the authenticity, authorization, and trustful state of the devices.

More recently researchers have also been looking into the matter of trust with attestation for IoT access [24]. According to the FBI Cyber Bulletin [47], malware can compromise the devices and make them victims to cause a cyber attack. Remote attestation is a mechanism through which a device can prove its trustful state to a remote device. Trusted Computing can be in various forms, such as hardware-based *Trusted Platform Module (TPM)*, and software-based *Trusted Execution*

Environment (TEE) [24]. Although TEE is lightweight, it is insecure and prone to physical attacks as compared to TPM.

Research work on NFC-based security has separately focused on the critical issues of authentication and attestation using NFC modes other than HCE, which has several advantages, as discussed in the earlier sections.

Authentication- Ceipidor et al. [27] proposed a mutual authentication protocol known as Kernees, which is based on the Needham Schroeder protocol. *However, it fails to achieve message authenticity and is prone to Brute Force attacks.* Thammarat et al. [140] proposed a lightweight mutual authentication protocol based on the limited use of the session key and prevents the Brute Force attack by using a set of sessions keys. *However, it lacks support for the device anonymity and attestation. There have been several authentication protocols using the NFC-based Peer-to-Peer mode [113, 158, 55, 97]. However, none of them addresses the issues of mutual attestation, secure storage, user anonymity, and protection to threats like MITM and DoS attacks. Moreover, the NFC Peer-to-Peer mode for bidirectional communication uses the Simple NDEF Exchange Protocol (SNEP) service [94], which, unlike the HCE mode, is not open for developers on unrooted Android devices. Table 2.2 discusses the limitations of the authentication techniques.*

Attestation- Toegl and Hutter [142] proposed a scheme to use an NFC-based mobile device as a *Mobile Attestation Token (MAT)* to access a TPM-based kiosk. The TPM signs its attestation report and sends it to a trusted Virtual Server (*VS*), which validates it and prepares a validation ticket for the user. *However, mobile and kiosk use insecure NDEF messages. There is no validation of the mobile device, which may also be prone to malware. Rooted mobile devices may be prone to malware and cause a breach of trust accordingly [136, 7].*

Aziz et al. [19] provided an extension of Transport Layer Security (TLS) for asymmetric encryption-based mutual authentication and TPM-based mutual attestation over TCP/IP. The TLS session generates a session key K_S. Both devices use nonces in the TLS session and the session for remote attestation. *However, the scheme has drawbacks of being computationally expensive due* *to asymmetric encryption, does not support device anonymity, and does not have proof-of-locality. The host also does not retain the secrecy of registers and logs because it transmits the attestation certificate without any encryption to the remote host.*

The existing schemes for authentication and attestation have limitations. The schemes do not fulfil the security and threat requirements that we have identified and listed later in Section 1.5.3. The existing attestation techniques have limitations, as presented in Table 1.3.

Hence, there is a need for secure mutual authentication and attestation protocol for secure and easy NFC-based access to health records from a portable device.

Table 1.3: Limitations of Existing Attestation Schemes

Requirement	**Toegl and Hutter [142]**	**Aziz et al. [19]**
S3:Mutual Authentication and Attestation	N	Y
S5:User Anonymity	N	N
S6:NFC Proof-of-Locality	Y	N
S7:Secure Storage	N	N
T1:DoS	N	N
T3:Collusion	N	Y
T4:Parallel Session	N	N
T6:Platform Impersonation	Y	N
T7:MITM	N	N
T8:Insider Attack	N	N

1.4.4 Selective Access of Secured Data from a Portable Device

Many cloud-based health record management solutions allow medical professionals to selective access the health records and protect patient's privacy. *Attribute-Based Encryption (ABE) [60]* provides fine-grained access control for sharing of ciphertext with a group of users. ABE comprises of a set of plain text attributes and an access policy to generate the ciphertext, and decryption keys so that each user has a different decryption key. ABE has two main variations *Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [21]* and *Key-Policy Attribute-Based Encryption (KP-ABE) [60]*. Several health record management systems consider CP-ABE for RBAC [107, 90].

According to Ambrosin et al. [9], with the advancement of computational and storage capabilities on mobile devices, Bethencourt et al.'s CP-ABE scheme [21] is practically feasible on mobile and IoT devices. They can be used to store and share critical data using CP-ABE, as suggested in the *iHealthEMR* application by Akinyele et al. [6].

For a portable health device, a patient must have the flexibility to visit numerous hospitals as well as be protected from malicious users. Hence, encryption technique must support ease of retaining ciphertext, mobility for sharing it across multiple users, and have minimal overheads for ciphertext and decryption keys after revocation. For efficient sharing of data from a portable device, the device must support the following requirements for revocation:

- **C1: No prior Knowledge of the Revocation List-** For a portable system that can be shared with multiple users, the revocation list can be dynamic. The ciphertext must be independent of the revocation list so that it must not require re-encryption when the revocation list changes.
- **C2: No Re-encryption of Ciphertext-** Re-encryption of ciphertext after revocation, can interrupt access of valid users.
- **C3: No Re-distribution of Decryption Keys-** The revocation must not affect the non-revoked users. They must be able to access the portable device without any interruption for re-generation of decryption keys.
- **C4: Revoke a scalable Number of Users-** Since the portable device could share information from multiple users, it must be able to revoke multiple adversaries.
- **C5: Independent of the Ciphertext-**: The revocation scheme must not associate any parameter with a ciphertext. In case of such an association for revocation of a user, the scheme will have to update parameters on all ciphertexts that the user accesses and will not be scalable.

CP-ABE with Indirect revocation is suitable for sharing data from a portable device, as discussed later in Section 2.5.2.1. It must fulfil all revocation requirements *C1-C5* for ease of portability, personal access for the owner, and sharing data directly with other external authorized users. In this thesis, we consider the revocation schemes based on Bethencourt et al.'s CP-ABE scheme, which has been implemented and proved feasible on mobile devices and IoT [10, 9].

The CP-ABE techniques that have been used for the cloud-based health record sharing schemes are not suitable for portable devices especially for revocation [115, 107, 90, 154, 17, 157, 73, 103, 72, 149, 91].

Proxy-based Immediate Revocation of ATTribute based Encryption (PIRATTE) scheme by Jahid et al. [76] is a variation of Bethencourt et al.'s CP-ABE scheme [21]. PIRATTE uses Lagrange's interpolation for secret sharing and provides indirect revocation without re-encryption of the ciphertext and key redistribution. *However, it can revoke only a certain number of users based on the degree of the polynomial used for secret sharing. Users receive proxy data from a trusted proxy server to complete decryption and use Lagrange's interpolation secret sharing. PIRATTE fulfils all revocation requirements, except for C4 because it can revoke only limited t number of users.*

A permanent revocation scheme by Dolev et al. [39] modifies the Bethencourt et al.'s CP-ABE scheme [21] and associates a counter *CTR* with the ciphertext and a user state $State_i$ for *ith* user $user_i$. We refer to this scheme as PERMREV in this thesis. The PERMREV scheme considers that the ciphertext resides on a secure cloud-based system. For revocation of $user_i$, the secure server updates *CTR*, re-encrypts the ciphertext, and sends the updated $State_i$ with new *CTR* only to the non-revoked users. Since revoked users do not get any updated state, the decryption fails. For no-ciphertext re-encryption in Modified PERMREV (M-PERMREV) the server can broadcast *State* to all users. For a revoked user, the proxy server updates the *CTR* and updates state of only revoked users, which causes the failure of decryption. *However, M-PERMREV scheme does not fulfill requirement C5 because it associates a CTR with the user's state* $State_i$ *and the ciphertext.*

None of the above research papers based on Bethencourt et al.'s CP-ABE scheme [21] address the issue of scalable revocation of malicious users for accessing shared data from a portable device. Table 1.4 discusses the limitations for the existing revocation schemes based on Bethencourt et al.'s CP-ABE scheme [21]. They do not fulfil all the requirements C1-C5 for revocation from a portable *device. There is a need for a protocol that can provide scalable access for sharing portable devices with multiple stakeholders that can satisfy all the above-listed requirements.*

Table 1.4: Limitations of Existing Revocation Schemes for Portable Devices

Requirments	**PIRATTE [76]**	**M-PERMREV [39]**
C1:Require Prior Revocation List	N	N
C2:Require Re-encryption	N	N
C3:Require Re-distribution of Keys	N	N
C4:Revoke Scalable users	N	Y
C5:Independent of Ciphertext	Y	N

1.5 Research Problem

1.5.1 Problem Statement

In this thesis, we look into an open challenge of dispersed health records, patient mobility across different hospitals, and issues for provenance, security, privacy, and trust of health records. The research problem comprises of proposing the design and architecture for a novel health record management system, which must aggregate dispersed health records on a patient's mobile, securely and directly share with multiple professionals with ease of access and provide readily available up to date complete health history. The system must address the security issues for provenance of health records, secure storage, mutual authentication, validation of trustful states of the devices, and selective access to health records with scalable revocation.

1.5.2 Research Objectives

This thesis proposes a next-generation secure and smart portable mobile-based health wallet to provide patient mobility across different hospitals with updated health records. For mobility and high availability of health history, it must aggregate dispersed health records and be up to date. The health wallet must be accessible directly to multiple authorized stakeholders to support mobility of patient across different health domains. It must provide flexibility and ease of maintaining digital records readily available to the patient.

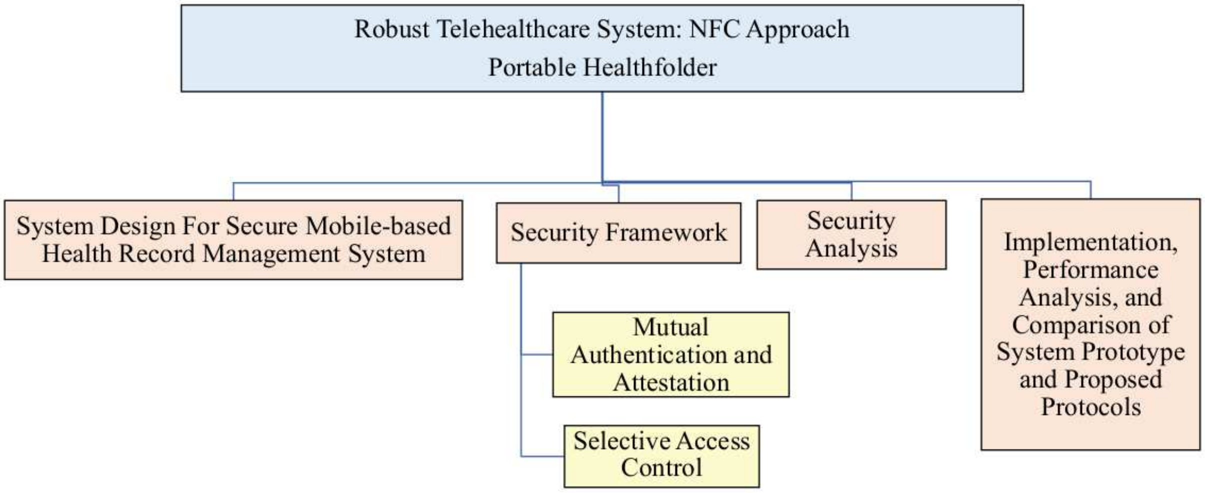

Robust Telehealthcare System: NFC Approach
Portable Healthfolder
System Design For Secure Mobile-based Health Record Management System
Security Framework
Security Analysis
Implementation, Performance Analysis, and Comparison of System Prototype and Proposed Protocols
Mutual Authentication and Attestation
Selective Access Control

Figure 1.3: Research Objectives

The objectives of this thesis are illustrated in Figure 1.3 and discussed below:

1. **Proposal of Healthcare Architecture:** Propose the design and architecture for a portable system to assist patient mobility across hospitals, and support the prime features for R1-R7 as mentioned in Section 1.3.

2. **Security and Privacy of Health Records:**

   Portable devices may be prone to security and privacy threats. Hence, the system must secure the health records as per the requirements identified by [48, 62]. There must be provision for secure storage, mutual authentication, trustful state of a device, selective sharing of records, and maintain provenance of health records.

3. **Security Analysis:** Perform detailed formal and informal security analysis for the proposed security schemes.

4. **Implementation and Performance Evaluation of the Prototype:** Implement a system prototype and perform evaluation and comparison for the prototype and proposed protocols with related systems.

We further elaborate on the key findings of this thesis work.

1.5.3 Summary of Contributions

This section describes the solution approach that this thesis has adopted for achieving the above research objectives.

1. **Proposed Architecture:** This thesis proposes a novel architecture and system design for a next generation smart portable health record management system to retain and securely access dispersed health records from a portable device over an NFC tap. It consists of a patient's mobile device for retaining a contactless *Secure Mobility-Assisted PortabLE (S-MAPLE)* health folder which aggregates the health records. The health folder can be accessed personally as well as shared directly with medical professionals. Hence, the health system can provide timely availability of complete health history for timely medical diagnosis and treatment. We propose the use of an HCE-based card for the first time for health record on a contactless health wallet. NFC provides ease of access and also secures the device with proof-of-locality and makes threats like MITM and eavesdropping difficult. It supports all of the system requirements *R1-R7* in Section 1.3 for patient mobility across different hospitals and provides up to date health history of a patient.

   Published work: Conference 3, Journal 4

2. **Security Framework:** Mobile devices are vulnerable to security threats. Although NFC provides proof-of-locality, it is prone to security threats, as discussed in Section 2.4.1.4. Based on the security issues discussed in earlier Section 1.2.2, this thesis identifies the critical security and threat requirements as illustrated in Figures 1.4 and 1.5 respectively. This thesis also proposes a security framework, which focuses on the following security solutions for securing the proposed smart health record management system:

   - *Secure Storage:* The mobile devices of patients and health professionals use a hardware tamper-resistant Secure Element (SE) [32] based on a microSD card for secure storage

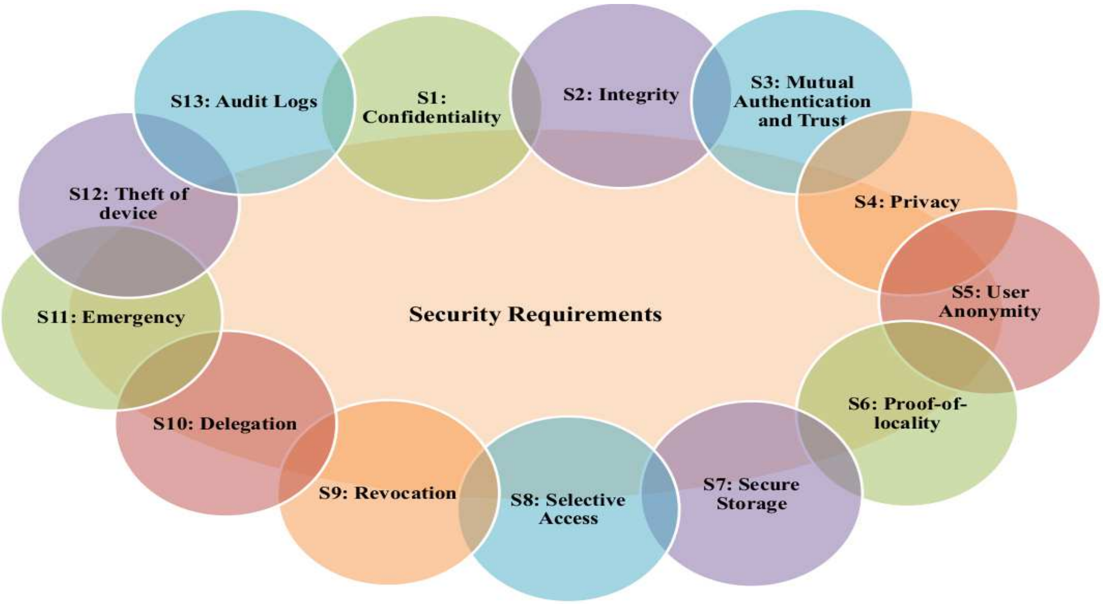


Figure 1.4: Security Requirements

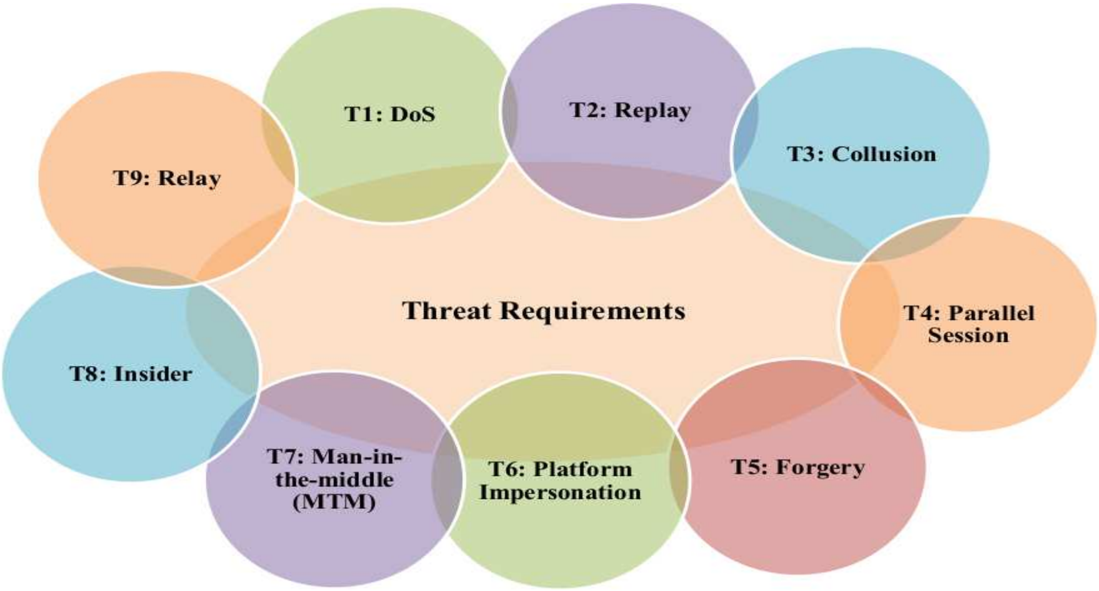


Figure 1.5: Threat Requirements

and cryptographic computations.

Published work: Journal 2, 3 and 4

- *Provenance of health records:* The portable health record system supports direct reading and writing of health records by multiple stakeholders over NFC for proof-of-locality.

The mutual authentication and attestation of the devices assure only authorized medical professionals can access the health records. The encryption with a CP-ABE-based scheme assures confidentiality and integrity of the health records with efficient selective RBAC with multiple stakeholders.

Published work: Journals 1, 2, and 4

- *Mutual Authentication and Attestation:* A cloud-based HealthSecure service supports mutual authentication for backup of health card on a digital vault and management of cryptographic services and unique identities of the devices. This thesis proposes a **NFC SE-based Authentication and Attestation** protocol for proof-of-locality and an end-to-end anonymous mutual authentication between SEs along with an associated remote attestation for the trust of the devices.

  Published work: Journals 2 and 3

- *Selective Access and scalable revocation:*

  Different stakeholders can access various sections on the health card through selective access control with Bethencourt et al.'s CP-ABE scheme [21]. **Scalable Proxy-based Immediate Revocation for CP-ABE (SPIRC)** is an extension of CP-ABE for selective access with scalable revocation of users without the requirement of re-encryption and re-distribution of keys to valid users.

  Published work: Conference 1,2; Journal 1

3. **Security Analysis:** This thesis presents a detailed informal and formal security analysis for the proposed security solutions and the proposed NSE-AA and SPIRC protocols in detail. NSE-AA is secure in the ROR model [2] and is proved secure using the *Automated Validation of Internet Security Protocols and Applications (AVISPA)* simulation tool [14]. The SPIRC protocol is safe for *Chosen Plaintext Attack (CPA)* security game. The construction of SPIRC scheme is secure under the generic bilinear group model.

   Published work: Journals 1, 2, and 4

4. **Implementation and performance evaluation of system prototype:** This thesis presents

the implementation of a prototype and performance evaluation for the health card and reader applications on current mid-range priced Android NFC-based mobile devices. The reader can successfully read and write to the HCE card. It executes the security solutions and protocols proposed in the security framework. This thesis presents a detailed performance evaluation and comparison for the proposed system and proposed security protocols.

Published work: Journals 1, 2 and 4

1.5.4 Who will Benefit from the Proposed Health Record Management System

The next generation smart portable health record system can be used by patients as a health wallet on their mobile devices and can be accessed by authorized health professional reader directly as a contactless card. The advantages for a patient are:

- *Readily available health records:* The patient can access health records readily and seek treatment in case of an emergency. It is important, especially for patients with chronic ailments and for the aged people who need to refer to their health history for the right diagnosis and treatment.
- *Easy access:* The health folder can be accessed directly by an authorized health professional as a contactless card for both read and writing
- *Upto date Records:* Health folder retains most recent health records because the health professionals directly update it.
- *Mobility Across Hospitals:* The patient can use the health folder as a health wallet and visit different hospitals with the flexibility of seeking treatment from various physicians.
- *Secure Health Records:* The security framework can provide security, privacy and trust for access to health records. The S-MAPLE system provides tamper-resistant storage of credentials and unique identity. It can be accessed by any authorized stakeholder as per selective Role-based Access control (RBAC) using fine-grained access control with CP-ABE. A cloud-based system assures management of credentials and backup of health records for refurbishing it in case of theft of patient's mobile device. The system also manages a

revocation list. In the event of a breach of trust by a stakeholder, it protects from multiple malicious users without the requirement of re-encryption or re-distribution of credentials to the patient as well as other valid stakeholders.

- *Aggregation of Data:* The S-MAPLE health folder enables easy collaboration of health information in a profile for the health history of a patient. The health folder can also provide visualization of past health history for a patient, such as the past blood sugar level. The health professional also gets details of the prior health history, which helps them for a quick, accurate analysis and treatment.
- *Assured Identification of Stakeholders:* The S-MAPLE access involves mutual authentication to assure that a valid patient is seeking treatment from an authorized health professional.
- *Provenance of Data:* It assures that a trusted stakeholder updates the health folder with reliable information for future access. Since the cryptographic credentials reside on tamper-resistant storage, it is difficult to hamper the health records.
- *Improved Quality of Healthcare:* Patient's can seek timely treatment with reduced errors because they have readily available secure health records. The patients can get quality healthcare with patient mobility across hospitals.

1.6 Outline of this Thesis

This thesis is structured as follows. In Chapter 2, we present the literature review, which discusses the details of the related techniques and briefly explains various technical concepts used. We discuss various existing schemes for Portable health record management with their limitations and the details HL7 health standard. We also explain the smart card, related standards, applications for healthcare, and related security threats. This chapter further discusses the low-energy wireless communication interfaces for RFID, NFC, and Bluetooth. It elaborates on the details of the NFC architecture, modes, security and privacy issues, and existing schemes for authentication over NFC. We also discuss the details of the existing CP-ABE schemes for revocation and selective access from portable devices by multiple users. We explain the related schemes for revocation and attestation for trust and present details for the simulation tool Automated Validation of Internet Security Protocols and Applications (AVISPA) for verifying the safety of security protocols.

In Chapter 3, we present the System Design for a Smart Health Record Management System. The chapter discusses the design and architecture for the access of the S-MAPLE health folder. It also elaborates on the techniques used for interoperability and the design of communication interface with HCE Bidirectional interface with Bluetooth. Chapter 4 discusses the Security Requirements and Solutions. It proposes the identified security and threat requirements for the health folder and proposed solutions considered in this thesis. We next present the details of the design for the proposed NSE-AA protocol in Chapter 5, followed by details of the SPIRC scheme in Chapter 6. These chapters discuss the protocol details for the two proposed protocols. Chapter 6 also presents the use case of the SPIRC scheme for selective access to the S-MAPLE health folder.

Chapter 7 presents the details of the Security Analysis. The chapter discusses the informal and formal security and threat analysis for the proposed protocols for NSE-AA and SPIRC. It presents the formal proof of NSE-AA protocol using the ROR model and simulation with AVISPA tool. In Chapter 8, we present the implementation details and performance evaluation for a system prototype. The chapter also presents the performance analysis and comparison of the proposed protocols. The thesis finally summarises the contribution of the research work in Chapter 9 for the conclusion and future work. It discusses the summary of contributions, limitations and the future scope of the work. We finally present the details for the Published Work, Bibliography and Annexure, which contains the AVISPA validation script for the NSE-AA protocol.

CHAPTER 2

LITERATURE REVIEW

This chapter presents the details for existing techniques that can assist in patient mobility across hospitals and the related technical background for the research work in this thesis. Section 2.1 presents the details for existing portable health record management systems and their limitations for patient mobility across hospitals. This section is followed by an overview of the HL7 health standard in Section 2.2, which is used to aggregate the health records in the proposed system. The chapter further discusses the details of secure storage with smart cards, their applications in the area of secure healthcare, and the related security threats in Section 2.3. Section 2.4 presents a discussion on the NFC technology for applications in healthcare applications. The section presents the details for the NFC modes and how HCE mode has greater benefits over existing NFC modes for a bidirectional communication for security handshake. NFC makes ease of access with the external device and yet provides secure access with proof-of-locality. The section also presents the security threats for NFC and the existing NFC-based authentication schemes along with their limitations.

Section 2.5 presents an overview of the existing techniques which are based on the Bethencourt et al.'s CP-ABE scheme. The section discusses the existing techniques for revocation and their limitations for supporting all requirements *C1-C5* for revocation from a portable device. The chapter finally presents the requirement for trusted computing with remote attestation and the existing schemes in Section 2.6. It presents the overview of the AVISPA simulation tool in Section 2.6.2.1, which is used to prove the safety of a security protocol under the Dolev-Yao intruder model [38].

2.1 Portable Health Record Management Systems

As discussed in Chapter 1, for patient mobility across hospitals, a patient must retain a portable device for health record system. Portable health systems can provide readily available health records for the timely diagnosis and treatment of a patient. With the recent improvement in the computational and storage capabilities of mobile devices, they can be used to store readily available health records with patients. However, they have been considered for either backup of health records or accessing cloud-based health record systems. Section 1.4.1 discusses the limitations of the existing portable and mobile-based health systems. As discussed in Table 1.2, none of the existing portable health record systems can fulfils all the requirements for patient mobility across hospitals. Although, some of them provide certain security features, however, none of them supports all the security requirements identified in Chapter 4. In the following sections, we discuss the details of related portable and mobile-based health record systems.

2.1.1 Portable MyCareCard System

MyCareCard is a hand-held portable device to provide a medical history of a patient within the United Kingdom, especially for an emergency [130]. The salient features of MyCareCard are:

- The device comprises of a USB-based smart card to retain the medical history and has advantages of both smart card and USB. The MyCareCard can be accessed using a GUI-based MyCareCard Browser.
- The portable device must contain current medication, name, allergies, blood group, long-term conditions, major health problems in the past, and next of kin.
- Healthcare professionals access health information based on their roles.
- The device provides portability and availability of health records to hospitals only within the UK.

The card has the following limitations for patient mobility across hospitals.

- It cannot be used outside the UK and hence does not aggregate all health records.
- The device lacks ease of access due to the disadvantages of the USB access as discussed in Section 1.4.1
- It lacks support for a robust security framework for authentication and selective authorization.

2.1.2 Portable Poket Doktor System

Poket Doktor System is a portable device to retain readily available health records especially useful in case of an emergency for timely healthcare [64]. The salient features of the Poket Doktor System are:

- The device comprises of a Bluetooth enabled smart card that stores health records. It also supports an RFID interface to help automate the Bluetooth pairing hence provides ease of access. A reader device can wirelessly access the portable device and also display the health records.
- Traditional Bluetooth has a lengthy discovery process. Hence the system uses a novel process called *Rendez-Blue*, which utilises RFID technology to speed up the Bluetooth connection establishment process and also reduces power consumption. The system keeps the Bluetooth module in a sleep mode for saving power, and a low-powered RFID interface triggers it on demand.
- In case of an emergency, Rendez-Blue assists medical professionals to modify the search radius to discover and connect to the smart card-based portable devices at distances between inches to several meters.
- The device provides basic security features, such as secure key-exchange, strong data encryption, and multiple levels of access to information using password protection.
- The device retains health records and uses Extensible Markup Language (XML) to allow standardization and efficiency in retrieving and parsing the health records.
- The device provides wireless access to health records on a portable system with ease of access, high availability, speed, reliability, and usability in emergencies.

However, the system has the following limitations:

- It does not provide aggregation of health records across different hospitals.
- There is no provision to write records directly to the device.

- It uses RFID, which provides distance ranging from an inch to meters to search health devices for an emergency in catastrophic conditions. However, RFID is less secure as compared to NFC.
- It does not provide selective access and support for other security requirements identified in Chapter 4.

2.1.3 Portable Secure Portable Token (SPT) with Tamper-resistant Health Folder

SPT comprises of a secure smart card and a USB key interface for storage of medical data on a tamper-resistant folder [12]. The system offers patients control over their health data. There may be less control of health data in the centralized health record systems due to:

- *Guidance of Patient Consent:* Patients may agree to predefined or default access policies, which they may not fully understand.
- *Unbounded Data Retention:* Patient may not be comfortable with certain health records to be retained for a lifetime due to privacy concerns.
- *No Security Guarantee Outside the Server Domain:* Health professionals can extract health records from the server on their mobile devices. If the mobile device is infected, it can cause a security breach.
- *No Disconnected Access to the Folder:* There is a huge disadvantage in case of disconnected access, disrupted networks and patients who cannot afford network access.

The salient features of the SPT device are:

- It stores sensitive information, called *hidden data (HD)*. HD may reveal sensitive information, and hence it is stored only locally on the SPT and is not replicated on the central server. Other non-hidden data is called *regular data (RD)* and is copied both on the SPT and the central server. Later, on the advice of a doctor, the patient can change the category of data from hidden to regular. However, the patient cannot change the category of health data from regular to hidden. If stakeholders query the regular data, they can also copy it and may not be possible to convert it to the hidden form completely. The hidden data preserves privacy but loses the property for durability. The central server retains the regular data in a crypto protected format, such as encrypted and signed format. The encryption keys are protected from the central server and are known only to the patient.

- Any authorized person, can connect to the central server or an SPT local server and retrieve the regular data based on the access control policy. Only people within the trusted circle can get access to the hidden data, irrespective of their role(s) and privilege(s). It stores the crypto protected data on the central server, and the encryption keys are known only to the patient's SPT and the SPTs of people within the trusted circle. The patient's SPT and the SPTs of the trusted circle exchange the encryption keys using a secure protocol (based on symmetric encryption). The synchronized data (regular, hidden, or encryption keys) is never disclosed to anyone except the recipient and is also protected during transmission by a secure protocol.
- The SPT device is tamper-resistant and provides readily available health records.

However, the system has several limitations as discussed below:

- The SPT devices does not support patient mobility and aggregation of data from different hospitals.
- The SPT device lacks ease of access due to the disadvantages of the USB access as discussed in Section 1.4.1.
- The security framework also lacks support for role-based access control and other security requirements.

2.1.4 Mobile-based System for Self-protecting Health Records

Akinyele et al. [6] have developed a iPhone mobile-based application called iHealthEMR, for implementation of self-protecting *Electronic Medical Records (EMRs)*. The application allows storage of EMRs for offline access outside the healthcare organizations on mobile devices and cloud-based systems, such as Google Health. The application encrypts the offline EMRs with their new dual-policy ABE library [60] for fine-grained access control. All stakeholders must access health records based on their roles. Besides Role-based Access Control (RBAC) with ABE, the application also supports content-based access control to authorize for collecting health records to an individual based on certain criteria. For example, a stakeholder may be provided access to only records within a certain period or certain kind of records. The application provides offline storage and offers readily available records, especially for an emergency, such as during a catastrophe like a hurricane. iHealthEMR has the following salient features:

- The policy application and encryption engine automatically generate the ABE policies for patient's health record.
- The ABE master controller manages the ABE decryption keys and securely provides the keys onto the patient's mobile devices.
- Stores encrypted health records on the hospital web servers or directly on third-party servers, such as Google Health for patient download.
- The mobile application allows patients to access and share ABE encrypted health records.
- The patients may store the health records in plaintext or encrypted format on third-party servers.
- The system proposes an ABE library, which implements three distinct ABE schemes: The key-policy scheme proposed by Lewko et al. [89], and two variations of the ciphertext-policy scheme from Waters [150]. The results indicate that they perform better for both encryption and decryption as compared to Bethencourt et al.'s CP-ABE scheme [21].
- The self-protecting scheme provides only offline backup of records on the patient's mobile device with selective access. It is readily available and can provide ease of visualization to the patient on the mobile device.

However, the scheme has the following limitations:

- It does not support patient mobility across hospitals.
- It does not support direct reading and writing to the mobile-based device.

- It lacks support for usability across hospitals.
- It does not provide secure storage, and also does not fulfil the security requirements identified in Chapter 4.

2.1.5 Mobile-based System for Accessing Cloud-based Health Records

Doukas et al. [40] developed an Android-based application called @*HealthCloud* that enables storage, update, and retrieval of patient's electronic healthcare data using cloud computing for the management. The health data may comprise of patient health records and medical images (supporting DICOM format and JPEG2000 coding). The prototype uses SHA1 hashing for message authentication and SSL for encrypted data communication.

The @*HealthCloud* application comprises of a cloud-based server with the following modules:

- The Patient Health Record application retrieves the patient records from the cloud.
- The Medical Imaging module helps in the display of the medical images on the device.

The patients use the proposed application to only access and view their health records from a cloud-based system on their mobile devices. It does not provide any health record management or portability across hospitals.

2.2 Health Standards

When health records distribute across different health systems, it is essential that they are interoperable for efficient patient care. According to Eichelberg et al. [41], *interoperability* of health systems implies that one system can accept data from the other system and perform the task in the required manner without the intervention of extra operators. There are open challenges of syntactic and semantic interoperability and integration of health records among several health systems, due to the differences in health formats, policies and laws.

In this thesis, we do not address the issue of interoperability. However, this thesis proposes storage of health records on the portable system in common standard health formats such, as Health

Level 7 (HL7) using translators as discussed in Section 3.1.3.

2.2.1 Health Level Seven (HL7)

HL7 [68] provides a framework and standards for the exchange and integration of electronic health information for clinical practice and management, delivery, and evaluation of health services. The standard provides interoperability to improve care delivery, optimization of workflow and reduces ambiguity. It also enhances knowledge transfer among all of the stakeholders, including healthcare providers, government agencies, vendor community, and patients.

HL7 standard sends information as a collection of one or more messages. Each message transmits one health record or health-related information. Examples of HL7 messages include patient records, laboratory records, and billing information.

Although many healthcare systems use the HL7 messages widely, many systems do not know how to use it, and they require a translator. The HL7 interface engines work alongside existing applications as an interpreter, speaking the language of HL7. There are currently two versions of HL7 Version 2 (HL7v2) and HL7 Version 3 (HL7v3). HL7v2 is an ANSI certified set of international standards used for medical data exchange and is a popular interoperability standard. Although Version 2 is the most widely implemented health standard, it lacks interoperability. HL7v3 is based on the object-oriented data model, called the *Reference Information Model (RIM)* and uses the Clinical Document Architecture (CDA). HL7v3 is not as widely used as HLv2, due to stringent modelling rules. Hence, in this thesis, we use the HL7v2 health records.

2.2.1.1 HL7 Message

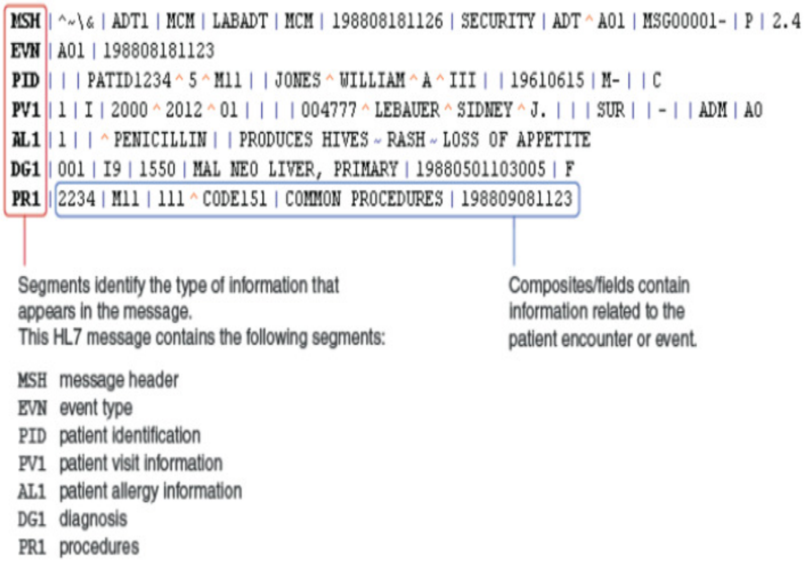
HL7 Messages transfer electronic data between different healthcare applications based on trigger events in the healthcare system, such as an event for patient admission in a hospital. The sender application prepares the HL7 message by collecting appropriate data and passes the *Electronic Data Interchange (EDI)* message to the requestor.

The messages consist of one or more segments. Although the messages are in the human-readable (ASCII) format, it is difficult to interpret them. A carriage return character separates each segment form the other. A different line of text displays each segment.

Each HL7 segment comprises of one or more composites or fields. A pipe (|) character separates one composite from the other. If a composite contains other composites, (?) upper arrow characters separate these sub-composites (or sub-fields).

Figure 2.1 illustrates a sample HL7 Message, where segments represent the following:

- MSH: Message Header
- PID: Patient Identifier
- NK1: Next Kin (First)
- PV1: Patient Visit (First)



```
MSH|^~\&|ADT1|MCM|LABADT|MCM|198808181126|SECURITY|ADT^A01|MSG00001-|P|2.4
EVN|A01|198808181123
PID|||PATID1234^5^M11||JONES^WILLIAM^A^III||19610615|M-||C
PV1|1|I|2000^2012^01||||004777^LEBAUER^SIDNEY^J.|||SUR||-||ADM|A0
AL1|1||^PENICILLIN||PRODUCES HIVES~RASH~LOSS OF APPETITE
DG1|001|I9|1550|MAL NEO LIVER, PRIMARY|19880501103005|F
PR1|2234|M11|111^CODE151|COMMON PROCEDURES|198809081123
```

Segments identify the type of information that appears in the message.
This HL7 message contains the following segments:

MSH	message header
EVN	event type
PID	patient identification
PV1	patient visit information
AL1	patient allergy information
DG1	diagnosis
PR1	procedures

Composites/fields contain information related to the patient encounter or event.

Figure 2.1: Sample HL7 message

Categories of HL7 message: The following are the primary message types in the HL7 standard [69]:

- *Patient Administration (ADT):* This message carries the patient demographic information for HL7 communication. ADT messages are widespread in HL7 processing and are among the most widely used of all message types. ADT messages also provide information about trigger events, such as patient admit, discharge, transfer, and registration. There are certain important segments in an ADT message, such as the PID (Patient Identification) segment, the PV1 (Patient Visit) segment, and the IN1 (Insurance) segment.

- *Orders (ORMs):* This message is a general order message that is used to transmit information about an order.
- *Observation Results (ORUs):* This message transmits observations and results from the source, which fills the information for the healthcare system or physician. It can also transmit results data from one health system to another. ORU messages are also used to register or link to clinical trials, or for medical reporting purposes for drugs and devices. These messages can consist of the following types of observations:

  - Clinical lab results
  - Imaging study reports
  - EKG pulmonary function study results
  - Patient condition or other data, such as vital signs, symptoms, allergies, and notes.

  ORU messages are structured reports of observations. Their format separates each observation into individual entity fields. They cannot carry images, but they use different data types, such as text, numbers, and codes. *Observation Request (OBR)* and *Observation Section (OBX)* are the most significant segments of ORU messages due to the following functions:

  - The OBR segment is used in all ORU messages as a report header and contains important information for filling the form, such as order number, request date/time, observation date/time, and ordering provider. It can be used more than once for each observation result in a message.
  - The OBX segment consists of the main clinical observation results. A message can use multiple OBX segments. One or more NTE segments can follow it to provide additional notes and comments about the observation.

- *Charges (DFTs):* This message describes a financial transaction for patient accounts that are sent to a billing system. It may include messages, such as ancillary charges or patient deposits.

2.3 Secure Storage with Smart Cards

As discussed in Section 1.4.2, a portable health record system must store secure data on tamper-resistant storage, such as a smart card. A smart card can store credentials and identity, which must not be accessible to an adversary. It can also perform cryptographic computations securely. It is useful for applications, such as identification, authorization, payment, and ticketing. The smart cards can comprise only memory or no memory with a small microprocessor to execute tasks. Commercially available smart cards have memory ranging from 2 kb to 64 kb [117]. They are of two types:

- *Contact card:* It is usually a memory card, which a user places in close contact with the reader. The ISO/IEC 7816 standard [74] provides specifications for contact cards.
- *Contactless card:* It communicates with the reader using high-frequency waves similar to RFID. The card receives energy from an electromagnetic field generated by the reader. International standards, such as ISO/IEC14443, ISO/IEC15693, Felica ISO/IEC15408, NFC with ISO/IEC18092, and EZ-PASS Proprietary Ultra-High-Frequency Technology provide specifications for a contactless card.

2.3.1 Application Protocol Data Unit (APDU)

The smart card reader and smart card communicate with each other using APDU packets. The ISO/IEC 7816-4 standard [74] defines the structure of the APDU, security, and commands for interchange. There are two categories of APDUs: command APDUs and response APDUs. A reader sends an APDU command packet to the card, which replies with an APDU response packet.

2.3.2 Java Card

Java Card [77] refers to a software technology that allows Java-based applications (applets) to be run securely on smart cards and small memory resource-constrained devices [25]. It is widely used in the SIM cards (used in GSM mobile phones), ATM cards, and secure microSD cards, such as GoTrust [57] with Java Card.

The Java Card-based smart cards comprise an operating system and ROM that contains a Java Card Virtual Machine (JCVM). The JCVM can support Java card applets, which implement a subset of the Java programming language. A Java Card applet implements the install method to initialize the applet [25]. It also implements a process method for handling command and response APDUs. Although a smart card can have multiple applets at a time, only one applet is active at a time. A standard Java compiler converts the applet source code into Java bytecodes. A converter checks for unsupported features, such as floats and strings. It then converts the bytecode into a more condensed form (CAP format) that gets loaded onto a smart card.

2.3.3 Secure Smart Cards for Healthcare

Smart cards have been used to secure health records and to provide portable health records, as described in the following examples:

- Yang et al. [163] proposed an E-prescription system to store up-to-date PHRs and insurance information on smart cards. It provides instant data access to doctors for crucial diagnosis and prescription. A patient can sign in with the secret signing key on the smart card for accessing prescriptions securely. The system also supports delegation of signing capability to other people.
- A smart card-based health management system proposed by Kardas et al. [79], uses a smart card for personal identification and transfer of health records. Both patient and the healthcare professional retain smart cards. The patient's smart card contains identification, personal, and general health information that can prove useful, especially in case of an emergency. All

prescriptions can be stored on the smart card as well as on the hospital database. The health professional uses the smart card to authenticate the system and access the patient's smart card. The system uses encryption keys and digital signature keys stored on the smart cards for authentication between client and server using a distributed communication protocol. The smart card system proposed by Kardas et al. [79] takes around 1.5s to read data of size 255 bytes and 2 s to write data of size 255 bytes. It takes approximately 9 s to after insertion of the smart card into the *Card Acceptance Device (CAD)*, starting a user session and display of the PIN entry dialogue.

Although smart cards are secure and portable; they have a small memory, which can store limited health data. It can also not provide instant visualization of health records. An external reader is required even for the patient to view their health records.

2.3.4 Smart Card Security Threats

Although smart cards are tamper-resistant, they are prone to the following security threats [161, 120]:

- *Denial of Service:* An adversary sends invalid login requests continuously to the smart card, due to which the card is left unusable for a valid user.
- *Replay Attack:* An intruder intercepts and re-submits messages to impersonate a legitimate user.
- *Parallel Session Attack:* An intruder intercepts the messages transmitted between a valid user and a smart card and sends it back to access the card.
- *Impersonation Attack:* An intruder tries to modify the intercepted messages to masquerade a legal user and accesses the smart card.
- *Man in The Middle (MITM) Attack:* It is a form of active eavesdropping in which an intruder sits between the card and a legitimate user and makes independent connections between them.

- *Relay Attack:* In a relay attack an attacker can cause the reader to communicate with a remotely located victim's smart card [80]. Hence, the attacker can build a virtual pickpocket system to affect the victim's contactless smart card without the victim's knowledge. Section 2.4.1.4 ellaborates on the details of the relay attack and its countermeasures.

2.4 NFC for Portable Healthcare Applications

Section 1.4.3 discusses the existing applications for RFID and NFC. It also discusses the details of the different NFC modes. None of the applications uses the Host Card Emulation (HCE) mode, which has several advantages, as discussed in Table 2.1. In the following subsections, we discuss the technical details for NFC, the novel HCE mode and the existing security threats. The HCE mode has been proposed for a secure smart mobile-based health record system in this thesis for the first time to the best of our knowledge.

2.4.1 Near Field Communication

NFC is a low energy wireless technology, with few centimetres of access distance, 13.56 MHz operating frequency and has a maximum throughput of around 0.4 Mbps. NFC's proximity assures proof-of-locality. A MITM attack comprises an attacker intercepting messages between two NFC devices. However, it becomes difficult for an attacker to come in between the two devices in NFC because of their proximity. Hence, NFC makes such MITM attacks and eavesdropping difficult [50].

2.4.1.1 NFC Architecture

NFC-enabled mobile devices are composed of various integrated circuits, such as Secure Element (SE) and NFC interface as shown in Figure 2.2.

NFC interface consists of a contactless, analogue/digital-based front-end known as *NFC Contactless Front-end (NFC CLF)*. The NFC controller enables the NFC transactions and communicates with the NFC CLF via the NFC antenna.

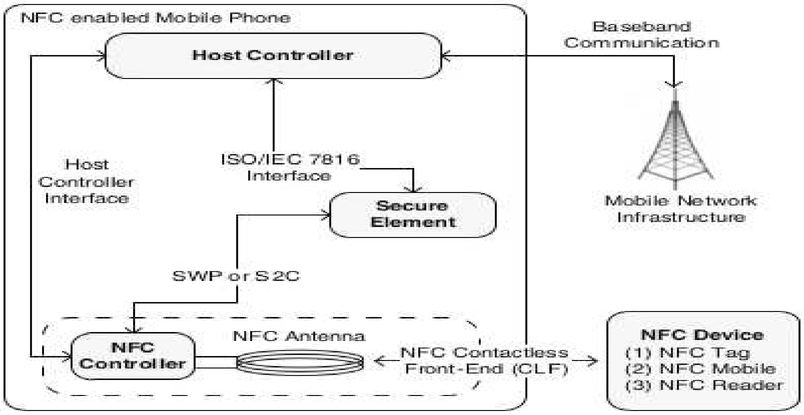


Figure 2.2: NFC-based Mobile Device Architecture [32]

A Secure Element (SE) is a secure smart card with a tamper-resistant microchip, which connects to the NFC controller of the mobile device. It stores sensitive data and also computes secure transactions for the NFC device. The NFC controller is connected to the SE through *Single Wire Protocol (SWP)* or *NFC Wired Interface (NFC-WI)*. Applications compiled with special libraries on the host processor can also access the NFC controller internally. An external card reader can access it externally through the RF field. SEs are of the following form factors:

- *Embedded SE:* It is a smart card that integrates into the mobile devices, such as in the iPhone.
- *Secure Memory Card (SMC):* It provides high-level security as a smart card complies with smart card standards, such as ISO/IEC 7816 and JavaCard [123]. SMC is removable, with larger memory, and can host several applications, such as the microSD card from GoTrust [57].
- *Universal Integrated Circuit Card (UICC):* It is a generic multi-application platform for smart card applications and implements a *Subscriber Identification Module (SIM)* on it. UICC is portable, personal and secure for a specific device, and can be easily managed remotely via *Over The Air (OTA)* technology. There are no UICC smart cards commercially available due to open issues on the UICC card management in NFC-based services.

Most SEs use the Java Card technology, which enables Java-based applets to execute with limited memory and processing capabilities. The UICC form factor of SE is the most secure form[123].

However, since the SMC form factor is independent of the manufacturer, we propose to utilise it in this thesis.

An SE can be used to emulate a hardware card on NFC-enabled mobile devices as well as to store credentials that can be accessed by internal applications compiled with special manufacturer's libraries. An SE can be accessed using the APDU command and response packets based on the ISO 7816-4 standard.

2.4.1.2 NFC Modes

NFC-enabled mobile devices can operate in the following modes: .

- **Reader-Writer mode-** In the reader/writer mode, the device can access NFC tag with *NFC Data Exchange Format (NDEF)* messages [127]. An NDEF is a data exchange format, which is used by NFC devices to exchange data. An NDEF message can contain an array of NDEF records. Each record consists of a header and a payload. The header consists of a 3-bit *Type Name Field (TNF)*, type (detailed typing for the payload), and ID (identifier meta-data). Android mobile devices create an NDEF record for raw byte array payloads. This mode provides a good way of sending and receiving messages on an NFC-enabled device.

  However, according to Roland and Langer [126], NDEF is insecure, even though there is a provision for digital signatures [82]. The tags also cannot be write-protected, and they have limited capabilities to support bidirectional security handshake with the reader device. Tag replacement or insecure access can create a potential for a device compromise.

  NFC Tags: An NFC tag can store various data types, such as a URL (web address) or a telephone number. The actual amount of data depends on the type of NFC tag and its storage capacities. A standard Ultralight NFC tag can store a URL of around 41 characters, whereas the newer NTAG203 NFC tag can store a URL of around 132 characters. Usually, it stores this information in a specific NDEF format, so that most devices can reliably read it.

- **Peer-to-Peer mode-** This mode is used to exchange data from one device to another, such as a file or a digital photo. The devices establish bidirectional communication. The *Near Field Communication Interface and Protocol (NFCIP-1)* provides a standard for an RF communication interface for the request-response model. One of the devices is the initiator, and the other one is the target. NFC divides the object of communication between the initiator and the target. The initiator generates an RF field (Radio Frequency field) and initiates the NFCIP-1 protocol. The target receives signals from the initiator and responds to the initiator using the RF field. In passive mode, the target communicates using the RF field of the initiator. In active mode, the target uses a self-generated RF field. Devices can use either of the modes based on their application.

  The *NFC security standards (NFC-SEC)* [110] defines a protocol stack for encryption functions on data link layer on top of NFCIP-1. It defines *Shared Secret service (SSE)* and *Secure Channel service (SCH)* for NFCIP-1. The SSE service generates a secret key for secure communication between NFC devices and initiates key agreement and key confirmation. The SCH service enables the communication between NFC devices with confidentiality and integrity using a key generated through SSE service. The NFC-SEC defines procedures of key agreement using *Elliptic Curve Secret Value Derivation Primitive, Diffie-Hellman (ECSDVP-DH)* version. The NFC terminal must have a public key and private key based on the Elliptic curve. The SCH service generates three hierarchical keys hierarchically by using the key generated through SSE for providing confidentiality and integrity of the messages.

  Simple NDEF Exchange Protocol (SNEP) on recent Android versions supports exchange of only one NDEF message per NFC session [127]. The bidirectional OPEN-SNEP [94] and security protocols, such as *Logical Link Control Protocol (LLCP)* secured by *Transport Layer Security (TLS)* known as LLPS [145], are not available on Android devices.

- **Card Emulation-** In card emulation, a card is emulated on an SE of the NFC-enabled mobile device. The card interacts with a smart card reader with PC/SC interface [127]. It communicates using an APDU command and requests packets based on the ISO-7816 standard [127]. *An SE-based card has drawbacks of limited computational capability, limited space, complex development process, and requirement of an additional PC/SC reader on traditional mobile devices [106].* Figures 2.3 illustrates how a reader interfaces with an SE-based card.

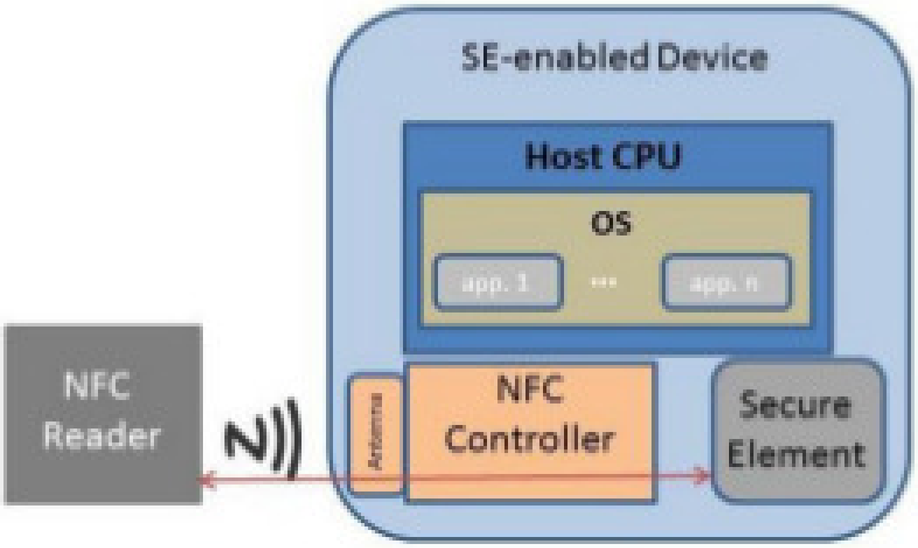


Figure 2.3: NFC enabled Device with SE-based Card Emulation [7]

- **Host Card Emulation (HCE)-** The HCE mode enables emulation of a smart card at a software level [7, 136]. Users can tap the device to initiate transactions with an application without the requirement of an SE in the device. This emulated card can be read by any NFC device, which is working in Read/Write mode. This emulated card can be used to make payments, display tickets, vouchers, and present ID. *Research In Motion (RIM)*, on the Blackberry platform [125], was the first to incorporate the HCE functionality on their mobile devices. Subsequently, Cyanogenmod integrated some patches to the Android OS, which permitted NFC enabled mobile phones to perform card emulation from the host. However, HCE attracted the most attention when Google incorporated it within Android 4.4 (KitKat). Figure 2.4 represents how a reader interfaces with an HCE-based card.

SE vs HCE: The key goal of HCE is to offer simple card emulation and to enable developers and service providers with new easy to introduce NFC services in the market [136]. However,

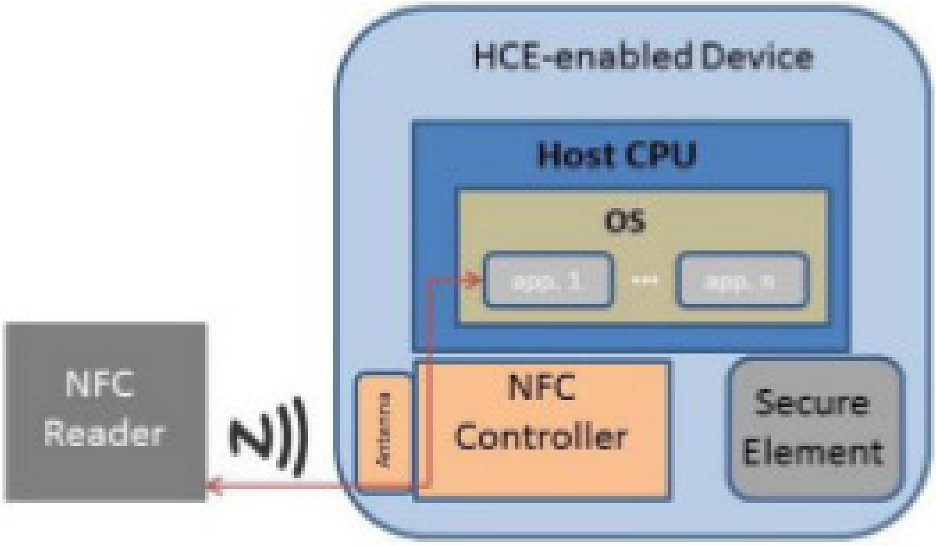


Figure 2.4: NFC enabled Device with HCE-based Card Emulation [7]

HCE has the following limitations as compared to SE:

- *Low Power Mode:* HCE require higher power as compared to SE based card emulation.
- *Roaming and Data Connectivity Scenarios:* Most HCE transactions require cloud-based credential management, which may be interrupted due to lack of connectivity.
- *Transaction Latency:* HCE transactions are slower as compared to SE.

Security threats with HCE: HCE is vulnerable to security threats since it is software based [152]. The absence of a secure environment leaves the system vulnerable. The vulnerable state can easily be exploited by a malware residing in the device's main memory to eavesdrop critical data, such as login credentials or payment transaction [125, 116]. Moreover, interference by other applications is also possible. A compromised cryptographic library can affect the confidentiality and integrity of the exchanged information [15].

Generally, HCE-based solutions use cloud systems for storing and retrieving credentials. Storing sensitive data in a secure remote location offers some protection against this vulnerability. However, to ensure complete protection, a secure environment on the device is necessary. A malware application on a mobile device can monitor the communication between an NFC controller and an HCE card application. It can have the following effects:

- Make the operating system, especially on rooted devices vulnerable to threats.
- Cause a denial of service if it can change the routing table for HCE application.
- Steal credentials stored in the applications that are used to access the cloud storage and backup.

The following measures can secure HCE [153]:

- *HC1: White Box Cryptography*- It embeds the secret in the code and transforms the ciphertext into a form such that it is difficult to derive.
- *HC2: Tamper Proofed Software*- It adds software security so that the attacker cannot modify the software statically or dynamically. It can be in the form of runtime integrity checking. In case the tamper-proofed system detects an attack, it produces a response, which causes the program to fail or record and log the occurrence of the attack.
- *HC3: Biometric Factors*- It can strengthen user authentication for HCE applications along with other means of authentication. It can use biometric factors, such as fingerprints, facial recognition, and voice recognition.
- *HC4: Device Identity Solutions*- These solutions help to authenticate mobile devices to online services and secure HCE-based applications. Fast Identity Online (FIDO) Alliance protocol that uses public key cryptography for online authentication is an example of device identity. A user's device creates a new key pair, retains the private key, and registers the public key with the online service. The user's device authenticates by signing off with the private key. It can only be unlocked locally on the device through biometric authentication or entry of a PIN. It can also support many security techniques, such as tokenization and *One-Time Password (OTP)*.
- *HC5: Security Frameworks/Trusted Execution Environment*- TEE is a secure area in the mobile device processor or coprocessor. It can store and process data safely, and also safely execute authorized security software (trusted applications) in a trusted environment. The TEE consists of software and hardware. It manages the access rights, isolates, and protects critical applications from the *Rich operating system (Rich OS)*.

The TEE can also associate with the SE for tamper resistant storage of credentials. The TEE is not affected by the compromise of a mobile device's operating system because it is isolated. It can also provide additional security features for the HCE-based applications.

- *HC6: Encryption-* It ensures that the data does not transmit as plain text. There can be a card breach if the attacker can intercept the clear text between the card and the reader, such as in the *MITM* attack. Applications can store HCE data in an encrypted format. *End-to-end encryption (E2EE)* ensures that the reader encrypts the data and protects it during transmission. Encryption can be applied in combination with tokenization for payment applications. Critical card holder data, such as *Primary Account Number (PAN)* can be encrypted and later be used for tokenization to replace the PAN.
- *HC7: Tokenization-* It is the process of substituting a random value for a high-value credential (for example, a PAN or Social Security number), thereby creating a correspondingly low-value credential. The substitution masks the identity of a card and secures HCE-based applications.
- *HC8: Additional Security Provided by an SE-* HCE-based applications can be secured in a hybrid model by storing sensitive data either in the cloud or in the SE. The SE can store credentials securely. TEE can further enhance the security of the SE and ensure that only trusted applications can access it.

In this thesis we propose the usage of SE for secure storage of credentials for HCE [42] along with encryption. SE is more secure as compared to TEE and is 24/7 available unlike the Tokenization measures.

Advantages of HCE- Table 2.1 discusses the comparison between the different NFC modes. HCE has several advantages:

- *Bidirectional Communication:* The HCE-based card can communicate with the external card reader using the APDU command and response packets. Hence, it can support

Table 2.1: Comparison of NFC Modes [136, 127, 7]

Mode	**Reader-Write**	**Peer-to-Peer**	**Card Emulation**	**HCE**
Security	Poor	Medium	High	High with SE
Storage	Few bytes	Large	Few Kbytes	Large
Speed	Low	Medium	Fast	Medium
Communication	Unidirectional	Bidirectional	Bidirectional	Bidirectional
Open development	Yes	No	Yes	Yes
Mobile-based reader	Yes	Yes	No	Yes
Power consumption	Low	High	Medium	High

bidirectional communication protocol based on the application requirements.

- *Higher Storage:* The HCE-card has larger storage as compared to the hardware-based smart cards since the latter resides on a host processor of a mobile device.
- *Lower Development Complexity and Cost:* Developers are free to design and implement the HCE card and reader application as per the requirements. They are not dependent on the mobile device manufacturer or root permission on the mobile device.
- *Independent of Service Provider for Deployment-* Unlike applets on SIM cards the HCE applications can be installed with ease by the user [7].
- *Directly Accessible by Another Mobile Device-* An HCE reader can be a traditional card reader or another mobile device with an HCE-based reader application. HCE can also provide a secure proprietary contactless with a customized communication protocol. Hence, there is support for direct end-to-end communication between the mobile devices. There must be a limit to the number of APDUs and size of the data so that a user does not need to hold the device for an extended duration.
- *Computing Power:* The HCE-card resides on the processor of the mobile device, and hence its computing capability is comparable to the Peer-to-Peer mode.

HCE can provide ease of access for tap-based access to health records from a mobile device and provide bidirectional communication for security interactions required to fulfill the security requirements mentioned in Chapter 4.

2.4.1.3 Bluetooth Automation with NFC

The throughput of NFC is less as compared to Bluetooth, and hence, NFC can automate Bluetooth pairing with the proximity tap to exchange data and reduce the communication time. Traditionally, NFC Peer-to-Peer mode is used to automate Bluetooth such as in the application for credit transfer between mobile devices [105]. However, it uses insecure NDEF messages, which can cause pairing with a malicious device. *Hence, there is a need to look into other secure NFC modes, such as HCE to assist in the automation of Bluetooth setup.*

2.4.1.4 NFC Security and Privacy

Although NFC assures proximity of devices and makes MITM attack difficult, it has several security threats [109, 96]. NFC lacks support for authorization and cannot assure trustful states of the devices. It is important to protect a mobile device with secure NFC communication for the following reasons:

- *User's Privacy:* It comprises of the data on the mobile device and cryptographic credentials on the SE.
- *Functionality of Device:* Mobile device communication interfaces and features, such as voice and data access.
- *Information Transferred over NFC Link:* Data communicated over NFC must be secure from an adversary.

Hence, there is a requirement to secure the NFC communication between the two devices. According to [109, 96] the main threats for NFC are as follows:

- *NS1: Untrusted Communication Channel-* There is no support to authenticate the NFC devices and communicate reliably.
- *NS2: Eavesdropping-* An adversary can use special hardware to intercept the messages between the two devices. The proximity between the two devices makes eavesdropping

difficult. However, NFC has no built-in encryption, and when an adversary does eavesdrop using a special hardware, there can be a loss of confidentiality of data [50].

- *NS3: Data Corruption-* An adversary that is in the range of the NFC devices can send radio pulses at 13.56 MHz, which can corrupt and jam the signals.
- *NS4: Data Modification and Insertion-* An attacker can modify and insert data through precise changes in the signal. Since there is no inherent encryption, it imposes a threat to unreliable communication.
- *NS5: Denial of Service-* A malicious tag can launch unwanted or malicious activities or applications on the device and hence, leave the device unusable. There should be some mechanism to turn NFC on and off.
- *NS6: Relay attack-* In this attack, an attacker can eavesdrop messages and relay them to a legitimate prover, get the correct response, and relay it back to the verifier [66]. The attacker relays all the application layer data so that the verifier can exchange all messages with the prover. Figure 2.5 illustrates implementation of a practical relay attack with a Peer-to-Peer NFC system [52].

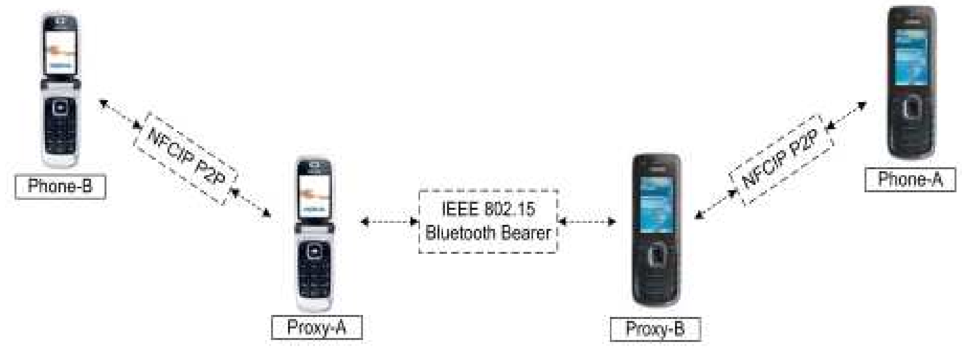


Figure 2.5: Practical Relay Attack with NFC Peer-to-Peer and Bluetooth Setup [52]

Phone-B sets up a Peer-to-Peer setup with Proxy-A, which has a Bluetooth communication with Proxy-B. There is another Peer-to-Peer setup between Proxy-B and Phone-A. Phone-A and Phone-B exchange data through the Proxies. Rolland et al. [128, 129] practically demonstrate relay attacks for card emulation on an NFC-based Google Wallet. Various measures can prevent a relay attack, such as [128]:

- *RA1: Faraday's Cage-* When an NFC-based mobile device operates in the card emulation mode, a Faraday's cage can shield the card's radio frequency interface. *However, shielding is possible only for external card emulation and not for software-based HCE cards.*
- *RA2: Additional Circuitry for Activating Card Emulation-* It is possible to enable and disable external card emulation through software. *However, it is not possible to deactivate the internal card emulation through a physical button.*
- *RA3: Two-factor Authentication-* Two-factor authentication with an entry of a PIN on the reader side can reliably prevent a relay attack.
- *RA4: Distance Bounding Protocol-* A verifier can detect a relay attack by monitoring any additional delay in the propagation time in case an attacker forwards data over a long distance. It measures the round-trip time of a challenge-response. These protocols establish an upper bound on the distance for the device that is preventing the relay attack. It takes into consideration the delay introduced into the channel from the time between sending a challenge and receiving a response and requires a reasonable benchmark for an acceptable delay.

  These protocols may not be feasible on currently available NFC-based mobile devices due to the high sensitivity of time delay and the requirement for the special-purpose hardware [63]. They also require a fast communication channel, which is not available in current NFC-based mobile devices. It is difficult to incorporate distance bounding protocols to detect a relay attack on HCE-enabled mobile devices because the performance of HCE cards depends on the processor of the mobile device [144]. In case of HCE, the tolerance around the benchmark will be extremely high, which makes it very difficult to distinguish between a relay attack and variation due to normal phone operations. The timing variation may also prevent the use of some security measures *to detect fake cards or attacks in progress.*

  - *RA5: Multi-Channel Communication-* User must verify the proximity of the remote device on another audio or visual channel. Although this technique reduces the simplicity of an NFC transaction, the relay process is complicated. An attacker must relay on multiple channels, some of which may require high bandwidth.
  - *RA6: Location-Based-* Nodes can verify the proximity of the location and identify a relay attack. Mobile devices can communicate their location, such as the *Global Positioning System (GPS)* signed with their private keys and communicated to a remote location. The verifier and prover devices must know their locations. The prover signs its location and sends it to the verifier. The verifier compares prover's location with its location and confirms if it is nearby. If the attacker has relayed the information, then the prover's location will be further away, and a relay attack will be detected. However, the location-based techniques, such as GPS have limitations of not being available indoors and some operators not prepared to share the information.
  - *RA7: Ambient Sensor-* Secure proximity detection techniques based on ambient sensors on NFC-based mobile devices can help gather information, such as light and audio. If the information is different at the verifier and the prover, then a relay attack can be detected [63]. The scheme does not allow the user to perform explicit actions and also preserves the user's location privacy.

- *NS7: Skimming of Applications on the SE-* Third-parties may get access to the index of applications stored on the SE and may steal valuable information as the device is swapped for a certain application and may result in a loss of privacy.
- *NS8: Managing In-device Security-* Inbuilt applications running on the host controller that interacts with the SE must validate through certificate-based authentication.
- *NS9: Insecure NFC link-* NFC interface supports the transfer of plain data, which can enable an attacker to eavesdrop communication.

- *NS10: Phishing attack-* An attack can be performed by modifying or replacing tags and misleading the user. Signatures on tags and reader device can prevent this attack.

2.4.2 Authentication schemes over NFC

It is essential to ensure that the devices can mutually authenticate each other at an application level because NFC does not provide encryption or any support for authentication. As discussed in Section 1.4.3.1 it is important to ensure trustful states for the devices through trusted computing mechanisms such as remote attestation. However, none of the existing NFC-based security schemes address the issues for mutual authentication and attestation together to the best of our knowledge. The following sections discuss the details for the related techniques for NFC-based authentication and their limitations. Table 2.2 discusses the limitations of the existing techniques for authentication over NFC. None of the above authentication techniques addresses the issues of mutual attestation, secure storage, and user anonymity.

Table 2.2: Limitations of NFC-based Authentication Schemes

Requirement	**Ceipidor et al. [27]**	**Thammarat et al. [140]**	**Peer-to-Peer [113, 158, 55, 97]**	**Lee et al. [87]**	**Proposed NSE-AA**
Message Integrity	N	Y	Y	Y	Y
Brute Force Attack	N	Y	N	N	Y
Device Anonymity	N	N	Y	N	Y
Device Attestation	N	N	N	N	Y
Secure Storage	N	N	N	N	Y
Mutual Authentication	N	Y	Y	N	Y

2.4.2.1 Mutual Authentication Kernees Protocol

Ceipidor et al. [27] proposed the Kernees protocol for mutual authentication between NFC-based mobile devices and *Point Of Sale (POS)* terminal for secure payment. It uses the contactless card emulation mode and EMV standard for payment. The EMV protocol was developed by Europay, MasterCard, and VISA and it is a global standard for payment between cards and POS. One of the essential phases for a financial transaction is the mutual authentication phase. It is essential because the data is exchanged via OTA and an adversary can intercept it through special hardware. In most cases the payer's card authenticates to the POS and the POS does not authenticate to the card. Unauthorized reader devices could leak information, read from the card, and cause problem, such as outflow of sensitive data, such as the financial data.

The authors suggest that digital certificates have limitations for mutual authentication, because the card cannot check the expiration date of the POS certificate due to lack of a clock. Public key cryptography may also be computationally expensive on resource constrained cards. Hence, they propose a protocol based on Needham Schroeder symmetric protocol.

The Kernees protocol consists of three entities: *P* is the POS, *N* for an NFC Phone, and *AS* for the Authentication Server. The details of the protocol are:

- *Step 1:* *N* connects to *P*; *P* sends message *M1: (E* (K_P, R_1, TS)*)* to *N;* where R_1 is a random number, *TS* is the timestamp, and K_P is a shared key between *P* and *AS*.
- *Step 2:* *N* sends *M2:* $(ID_N,\ E\ (K_N, R_2, M1))$ to *P*, where R_2 is a random number, ID_N is *N*'s identity and K_N is a shared key between *N* and *AS*.
- *Step 3:* *P* sends *M3:* $(ID_P, M2)$ to *AS*, where ID_P is identity of *P*.
- *Step 4:* *AS* extracts *TS* from *M1* and R_2 from *M2*. *AS* creates a session key *K* and sends *M4:* $(E\ (K_P, K, ID_N, TS)\ ||\ E\ (K_N, K, ID_P, R_2))$ to *P*
- *Step 5:* *P* extracts R_2, *TS*, and *K* from *M4*; It compares the received *TS* with the stored *TS*. If they are same, then it sends *M5:* $(E(K_N, K, ID_P, R_2),\ E(K, R_3))$ to *N*.
- *Step 6:* *N* extracts *K* from *M5*, verifies received R_2, and authenticates *P*; sends *M6:* $(E\ (K, R_3\text{-}1, R_4))$ to *P*
- *Step 7:* *P* verifies R_3 and authenticates *N*, sends *M7:* $(E(K, R_4\text{-}1))$ to *N*.
- *Step 8:* *N* verifies R_4.

If the verification is accepted, then the authentication process is successful. The protocol satisfies message authentication, confidentiality, and mutual authentication properties. *However, it lacks* *message integrity. Moreover, the session keys K_N and K_P, are static parameters and hence, the Kernees protocol is prone to Brute Force attacks. The protocol also lacks secure storage, device anonymity and attestation of devices.*

2.4.2.2 Lightweight Mutual Authentication Protocol

Thammarat et al. [140] proposed a lightweight mutual authentication protocol for NFC communication, to prevent replay and MITM attacks. The protocol is based on the limited use of the session key and it can prevent the Brute Force attack by using a set of sessions keys.

This thesis considers the NFCv2 protocol proposed by Thammarat et al. [140]. It consists of an NFC-enabled mobile phone (N) in card emulation mode, POS that is a sales station providing NFC device, and Authentication Server AS. $SK_{A\text{-}Bj}$, j=1, ..., m, is the set of session keys shared between users A and B. Initially user registers the device to exchange key $(K_{N\text{-}AS}, DK_{N\text{-}AS}, m)$ between user and AS, and key $(K_{N\text{-}POS}, DK_{N\text{-}POS}, m)$ between the user and the POS. It uses session generation technique as proposed by Kungpisdan and Metheekul [84] to generate a set of session keys where $j = 1,...,m$. $SK_{N\text{-}POSj}$ is a set of keys between the user and the POS and $\mathrm{SK}_{\mathrm{N\text{-}ASj}}$ is a set of keys between the user and the AS.

Similarly the POS and the authentication server exchange $(K_{POS\text{-}AS}, DK_{POS\text{-}AS}, m)$ and create a set of session keys $SK_{POS\text{-}ASj}$.

The details of the proposed protocol are:

- *M1: N* sends *M1:* $(ID_N, n_1, T_1, E(S_{KN\text{-}POSj}, r1), h(n_1, S_{KN\text{-}POSj}), r2))$ to *POS*; where T_1 is the timestamp when *N* initiates communication, *r1:* $E(S_{KN\text{-}ASj}, Request, T_1)$, *r2:* $MAC\ (Request, T_1, ID_N, S_{KN\text{-}ASj})$.
- *M2: POS* sends *M2:* $(ID_N, ID_{POS}, r1, r2, h(ID_P, E(S_{KN\text{-}POSj}, r1), S_{KPOS\text{-}ASj}))$ to *AS*.
- *M3: AS* verifies *r1* and *r2* and decides to Accept/Reject and sends the result in *M3*: Accept/Reject to *POS*, *r3, r4*; where *r3:* $MAC(Accept/Reject, S_{KPOS\text{-}ASj+1})$, *r4:* $MAC\ (Accept/Reject, T_1, T_2, SK_{N\text{-}ASj+1})$, T_2 is the timestamp when *AS* is sending the result.

- *M4: POS* verifies *r3*, authenticates *N* and sends *M4:* Accept/Reject to *N*, T_2, n_2, *r5, r4; where r5: MAC(*n_1, n_2, $S_{KN\text{-}POSj+1}$*)*.
- *N* verifies *r4* and *r5* and authenticates *POS*

The protocol prevents a Brute Force attack because it is difficult to find the correct session key as session keys change every time a transaction completes. It also prevents a replay attack because it uses unique nonces and limited-use session keys. Use of hash functions provides data integrity. Use of *Message Authentication Codes (MAC)* with session keys provides authentication of the party. It is difficult to perform the MITM and Brute Force attacks because the protocol changes the session keys constantly by using strong encryption techniques. *However, the protocol lacks support for the device anonymity, device attestation for trust and secure storage.*

2.4.2.3 P2P Authentication Protocols

The NFC-SEC for Peer-to-Peer mode helps generate a session key and makes MITM attack and eavesdropping difficult. However, it does not help authenticate users and protect user's privacy and may make the devices vulnerable to spoofing attacks, such as MITM. Eun et al. [46] propose a conditional privacy protocol over NFC-SEC using pseudonyms, which can help protect it from an honest user who can spoof as someone. There are several authentication protocols [113, 158, 55, 97], which use the NFC-based Peer-to-Peer mode and improve the Eun et al.'s scheme [46].

However, the NFC Peer-to-Peer mode for two-way communication uses the SNEP service [127], which unlike the HCE mode, is not open for developers on unrooted Android devices. The authentication protocols also use asymmetric encryption using ECC, which is computationally expensive as compared to the symmetric encryption. The protocols also do not address the issues of secure storage and attestation for trustful states of devices.

2.4.2.4 Digital Signature for Authentication over HCE

Lee et al. [87] proposed an authentication protocol, that uses NFC-based HCE mode the Android-based mobile devices and proposes a simple protocol using digital signatures for authentication. The protocol has the following phases:

- *Registration:* A user registers with the server and generates a set of public and private keys. The mobile device saves server information and the private key, and the server stores the device UUID for device identification and the public key. However, it does not address issues of secure storage, user anonymity.

- *Login:* A reader device contacts the server and receives a nonce comprising of the server info and a time-stamp. It scans for the card and once detected forwards the nonce to the card device. The card extracts the server information and checks its validity. The card signs the nonce with its private key and sends its UUID and the signed nonce to the server via the reader device.

- *Verification:* The server first validates the signed nonce with the public key of the card using its UUID. It authenticates the card and returns the results to the card through the reader.

The protocol has disadvantages since it only authenticates the card and does not authenticate the reader. Also, it does not look into the issues of secure storage, device anonymity, and attestation for the trustful state of devices.

2.5 Selective Access of Secured Data from a Portable Device

2.5.1 Attribute-Based Encryption

Attribute-Based Encryption (ABE) [60] provides fine-grained access control for sharing a ciphertext with a group of users. It comprises of a set of plaintext attributes and an access policy to generate ciphertext and decryption keys so that each user has a different decryption key. ABE has an advantage that users cannot aggregate their attributes together to decrypt the ciphertext and hence, it is collusion-free. It has the following variations [86, 115]:

- *Ciphertext-Policy Attribute-Based Encryption (CP-ABE) [21]*: It associates a set of attributes to the decryption key and an access policy to the ciphertext. A decryption key can decrypt the ciphertext if its associated attributes satisfy the access policy of the ciphertext. Users can be assigned different decryption keys, with each decryption key associated with a subset of attributes that satisfy the ciphertext's access policy. A set of attributes can be used to define the role of a user. *Hence, CP-ABE can provide Role-Based Access Control (RBAC).*
- *Key-Policy Attribute-Based Encryption (KP-ABE) [60]*: It associates attributes to the ciphertext and access policies to the decryption keys.
- *Hierarchical Attribute-based Encryption Scheme:* This scheme generates the keys hierarchically at different levels. It uses the disjunctive normal form policy to generate the keys hierarchically. It also assumes that a single authority administers all attributes in one conjunctive clause. Although it is secure, it is not easy to implement.
- *Multiauthority Attribute-based Encryption Scheme:* Different cooperative and independent authorities authorize a user's secret key. Although it is secure, the scheme requires cooperation and interaction between the different authorities.

2.5.1.1 Bilinear Maps

Bilinear maps associate pairs of elements from two algebraic cyclic groups to an element of a third algebraic cyclic group. The Bilinear Maps can support the following types of pairings:

1. **Symmetric Pairing:** *Definition-* Let G, G_T be cyclic groups of prime order p. Let g be generator of G. A symmetric bilinear pairing or bilinear map e is defined as:

   $e : G * G \rightarrow G_T$.

   It has the following properties:

   - *Bilinearity:* For all u, v elements of G; a, b element of Z_p, $e(u^a, v^b) = e(u,v)^{ab}$.

- *Non-degeneracy:* e(g,g) $\neq$ 1.

2. **Asymmetric Pairing:** *Definition-* Let G_1, G_2, G_T be cyclic groups of prime order p. Let g_1 and g_2 be generators of G_1 and G_2. A asymmetric bilinear pairing or bilinear map e is defined as:

   $e : G_1 * G_2 \rightarrow G_T$.

   It has the following properties:

   - *Bilinearity*: For all u, v elements of G_1 and G_2; a, b element of Z_p, $e(u^a, v^b) = e(u,v)^{ab}$.
   - *Non-degeneracy:* e(g,g) $\neq$ 1.

2.5.1.2 Access Structure

Definition 1 *Let P_1,P_2,... P_n be a set of parties. A collection $A \subseteq 2^{P_1,P_2,...P_n}$ is monotone if $\forall$ B,C if B ϵ A and B $\subseteq$ C then C ϵ A. An access structure is a collection A of non-empty subsets of P_1, P_2, ...P_n. The sets in A are called the authorized sets, and the sets not in A are called the unauthorized sets.*

2.5.2 Ciphertext-Policy Attribute-Based Encryption

The ciphertext-policy attribute based encryption scheme comprises of the following algorithms:

- **Setup:** It takes the implicit security parameter as input and outputs the public parameter *PK* and the master key *MK*.
- **Encrypt (PK;M;A):** It takes public parameters *PK*, a message *M*, and an access structure *A* for the attributes as input. It encrypts *M* and produces a ciphertext *CT* so that only the user who has a set of attributes that satisfy the access structure can decrypt the message. The ciphertext implicitly contains *A*.
- **Key Generation (MK; S):** It takes the master key *MK* and a set of attributes *S* that describes the decryption key as input. It generates a private key *SK*.

- **Decrypt (PK;CT;SK):** It takes the public parameters *PK*, a ciphertext *CT*, which contains the access policy *A*, and a private key *SK* for a set *S* of attributes as input. If attributes in the set *S* satisfy the access structure *A* then it decrypts the ciphertext and returns the message *M*.
- **Delegate (SK; S'):** It takes the secret key *SK* for a set of attributes *S* and a set $S' \subseteq S$ as input. It generates a delegated secret key *SK'* for the associated set of attributes *S'*.

2.5.2.1 CP-ABE Revocation Techniques

It is essential to prevent access of ciphertext from a malicious user through revocation techniques. The revocation techniques can be direct, indirect and hybrid [115]. Unlike the direct schemes, indirect schemes do not require any prior knowledge of the revocation list. Indirect schemes broadcast an intermediate key update so that only non-revoked users can update their keys. Hence, they are suitable for portable devices to provide ease and flexibility to the owner. They also require a key update phase, which can provide bottleneck for interaction with the *Trusted Certified Authority (TCA)*.

Indirect revocation schemes for CP-ABE for portable devices must satisfy all revocation requirements *C1-C5* defined in Section 1.4.4 for sharing data from a portable device securely and directly with external users.

The CP-ABE techniques used for cloud-based schemes are not directly suitable for portable devices. A broadcast variation of CP-ABE proposed by Narayan et al. [107] has a limitation that the length of ciphertext grows proportionally with the number of revoked users. *Hence, the scheme may not be feasible for portable devices with limited storage.* A scalable health record management scheme by Li et al. [90] uses a revocation scheme proposed by Worcester et al. [154]. *However, the scheme requires re-encryption of ciphertext for revocation and hence violates requirement C2.*

Attrapadung et al. [17] provided a hybrid revocation scheme, which supports both direct and indirect modes. *However, it has a drawback of long user secret key length, which can be an overhead for portable mobile devices*. Ibraimi et al. [73] suggested an indirect revocation scheme, which generates two portions of the private key, which are required for decryption. The user and a mediator retain one of the two portions of the key. The mediator sends the right portion of the key to a non-revoked user to assist in decryption. *However, it uses the CP-ABE scheme by Cheung et al. [30], which has a drawback that there is an increase in the size of ciphertext and key with the increase in the total number of attributes in the access policy. Hence, the scheme is not suitable for a mobile device with limited storage*. Modi et al. [103] proposed a revocation scheme for secure file access on the cloud. Hur et al. [72] proposed an indirect revocation scheme to provide fine-grained attribute revocation. Tian et al. [141] propose a CP-ABE scheme known as *Role-based Access Control scheme (RACS)* to provide RBAC. *However the schemes [103, 72, 141] cannot be used for portable devices because they require re-encryption and hence do not fulfil requirement C2.*

The following sections discuss the details of two indirect schemes for revocation using Bethencourt et al.'s CP-ABE scheme [21] as discussed in Section 1.4.4. Table 1.4 discusses the limitations for these schemes, which satisfy requirements *C1-C3* for revocation. Jahid et al. [76] proposed PIRATTE scheme which satisfies all requirements except *C4* for scalable revocation. Dolev et al. [39] proposed a permanent revocation scheme which does not fulfil requirement *C5* and requires a constant term associated with the ciphertext which is updated for revocation of a user.

2.5.3 Proxy-based Immediate Revocation of ATTribute-based Encryption (PIRATTE)

Jahid et al. [76] proposed the PIRATTE scheme, which improves the Bethencourt et al.'s CP-ABE scheme [21] for indirect revocation to satisfy requirements *C1, C2, C3* and *C5*. Users receive proxy data from a proxy server to complete decryption. PIRATTE uses a polynomial P of degree *(t+1)* in the master key. The trusted proxy server divides the secret *P(0)* into portions and provides a share to each user. During decryption, each user seeks a proxy key and t shares of the secret from the proxy server. It uses Lagrange's interpolation to combine the t secret portions with the user portion to generate the secret *P(0)*. If the user is non-revoked, then the proxy server sends valid secret portions. Otherwise, it sends invalid secret portions, so that the user cannot generate the secret *P(0)* and hence decryption fails. PIRATTE fulfils all revocation requirements, except for *C4* because it can revoke only limited t number of users.

Secret Sharing- It generates a random polynomial P of degree t such that P(0) = s, where s is the secret shared among n users. The i-th user gets the share <i; P(i)>. If there are t + 1 shares P(x_0), ...,P(x_t), then from Lagrange's interpolation P(0): $\sum_{i=1}^{t} \lambda_i P(x_i)$ where $\lambda_i = \prod_{j\neq i} \frac{x_j}{(x_j - x_i)}$

PIRATTE supports user and attribute revocation as discussed in the following sections. In user revocation the Proxy server can revoke a specific user completely. In attribute revocation, the proxy server revokes a specific attribute for a user and does revoke the user completely. The other attributes for the user can still assist in decryption for a ciphertext, which they can satisfy. The PIRATTE scheme also presents delegation of a secret key. The following section describes the construction for the user-based revocation scheme and the intuition behind the proxy components using Lagarange's interpolation.

2.5.3.1 Construction of PIRATTE User-based Revocation

Intuition-

The master key MK contains a polynomial *P* of degree *t* with *P(0)* as the secret, which blinds user's secret keys. Each user *u* has a key with a random share *P(u)* of *P(0)*. The proxy key comprises of *t* shares of the key and helps convert a part of the ciphertext for successful decryption. A proxy server maintains a revocation list. When the proxy server has to revoke a user, the user's share becomes a part of the proxy key and the converted ciphertext. A revoked user does not have enough *(t + 1)* points and hence cannot unblind the key and the ciphertext and decrypt it. The decryption is successful only for the non-revoked users because they can succesfully combine their secret keys. The proxy key consists of *t* random *P(u)* points for a non-revoked user. The scheme can revoke maximum *t* revocations because the scheme is based on polynomial secret sharing, and the degree of the polynomial is t.

- **Setup:** *Trusted Computing Authority (TCA)* chooses G_1, G_2, g_1, g_2; sets the broadcast secret P(0), and random elements α and $\beta \epsilon Z_p$ to generate a public key *PK* and a master key *MK* as defined below:

$$PK = G_1, G_1, g_1, g_2, h = {g_1}^{\beta}, e(g_1, g_2)^{\alpha} \tag{2.1}$$

$$MK = \beta, {g_2}^{\alpha}, P \qquad (2.2)$$

- **Encrypt (PK,M,τ):** X is a set of leaf nodes in access tree τ. It encrypts data M to generate ciphertext *CT*. The encryption algorithm as in Bethencourt et al.'s CP-ABE scheme [21], except that it uses asymmetric groups.

  $$CT = \left(\tau, \tilde{C} = Me\left(g_1, g_2\right)^{\alpha \mathrm{s}}, C = h^{\mathrm{s}}\right);$$

  $$\forall x \epsilon X : C_{\mathrm{x}} = {g_1}^{\mathrm{qx(0)}}, C'_{\mathrm{x}} = H\left(att(x)\right)^{\mathrm{qx(0)}} = {g_2}^{\mathrm{hxqx(0)}}\right) \qquad (2.3)$$

  H:0,1→G_2* is a hash function that maps a string attribute to a random element in G_2, and $h_x = log_{g_2}H(att(x))$.

- **KeyGen:** The algorithm outputs a secret key corresponding to the set of attributes *S*, blinded by *P(0)* from *MK*. The algorithm has an extra component D''_j that contains user information in addition to the attribute information. It generates random numbers r and r_j for each attribute *j*. The user u_k receives the secret key *SK* defined as:

  $$SK = (D = {g_2}^{\frac{\alpha+r}{\beta}}; \forall j \epsilon S :$$

  $$D_{\mathrm{j}} = {g_2}^{\mathrm{r}} H\left(j\right)^{\mathrm{r_j P(0)}} = {g_2}^{\mathrm{r + h_j r_j P(0)}},$$

  $$D'_{\mathrm{j}} = {g_1}^{\mathrm{r_j}},$$

  $$D''_{\mathrm{j}} = \left(D'_{\mathrm{j}}\right)^{\mathrm{P(u_k)}} = {g_1}^{\mathrm{r_j P(u_k)}}) \qquad (2.4)$$

- **ProxyRekey:** Whenever the trusted proxy server wants to revoke keys, it creates a list of revoked users RL with their identities u_i, i *ϵ 1,....t*, and evaluates the corresponding $P(u_i)$ using *MK*. The trusted proxy server gives the proxy key *PXK* to the user. In case of no or fewer than *t* revocations, TCA generates random *(x; P(x))* other than the actual user identities, to make RL of length *t*. Proxy key PXK:

  $$\mathrm{PXK} = \forall \mathrm{u_i} \epsilon \mathrm{RL} : (\mathrm{u_i}, \mathrm{P(u_i)})$$

- **Convert(PXK, $\forall$x ϵ X: $\mathbf{C_x}$, $\mathbf{u_k}$):** The proxy server uses its key *PXK* and the decryptor's identity u_k to calculate C''_x as follows:

  $$\forall \mathrm{i}, \mathrm{j} \epsilon\ 1,...\mathrm{t},\ \mathrm{k} \neq 1....\ \mathrm{t};$$

  $$\lambda_{\mathrm{i}} = \frac{u_{\mathrm{k}}}{u_{\mathrm{k}} - u_{\mathrm{i}}} . \prod_{j \neq i} \frac{u_{\mathrm{j}}}{\left(u_{\mathrm{j}} - u_{\mathrm{i}}\right)} \qquad (2.5)$$

$\forall$y ϵ X:

$$C''_{\mathrm{x}} = (C'_{\mathrm{x}})^{\sum_{i=j}^{t} \lambda_{\mathrm{i}} P(u_{\mathrm{i}})} = g_2{}^{\mathrm{h_x q_x(0)} \sum_{i=j}^{t} \lambda_{\mathrm{i}} P(u_{\mathrm{i}})} \tag{2.6}$$

The user secret key *SK* is blinded by *P(0)*, and requires C''_x *and* C_x and C'_x for decryption.The proxy server also calculates λ_k and sends it to the user u_k.

- **Decrypt:** The decryption algorithm is a recursive algorithm is as defined in the Bethencourt et al.'s CP-ABE scheme [21]. The recursive algorithm *DecryptNode(CT,SK, x)* takes ciphertext $CT = (\tau, \tilde{C}, C, \forall\ x \ \epsilon\ X : C_x,\ C'_x)$, private key *SK* associated with a set *S* of attributes, and a node *x* from τ.

    1. *If node x is a leaf node:* Let *i = att(x)* where *att(x)* is the attribute for the leaf node *x* in the tree. If $i \ \epsilon\ S$ then:

    $$\begin{aligned} DecryptNode(CT;SK;x) &= \frac{e(C_{\mathrm{x}}, D_{\mathrm{i}})}{e(D''_{\mathrm{i}}, C'_{\mathrm{x}})^{\lambda_{\mathrm{k}}} e(D'_{\mathrm{i}}, C''_{\mathrm{x}})} \\ &= \frac{e(g_1,g_2)^{\mathrm{rq_x(0)+q_x(0)h_ir_iP(0)}}}{e(g_1,g_2)^{\mathrm{r_ih_iq_x(0)}\lambda_{\mathrm{k}}\mathrm{P(u_k)}} e(g_1,g_2)^{\mathrm{r_ih_iq_x(0)}\sum_{j=1}^{t}\lambda_{\mathrm{j}}P(u_{\mathrm{j}})}} \\ &= \frac{e(g_1,g_2)^{\mathrm{rq_x(0)+q_x(0)h_ir_iP(0)}}}{e(g_1,g_2)^{\mathrm{r_ih_iq_x(0)}(\lambda_{\mathrm{k}}\mathrm{P(u_k)}+\sum_{j=1}^{t}\lambda_{\mathrm{j}}P(u_{\mathrm{j}}))}} \\ &= \frac{e(g_1,g_2)^{\mathrm{rq_x(0)+q_x(0)h_ir_iP(0)}}}{e(g_1,g_2)^{\mathrm{r_ih_iq_x(0)P(0)}}}, k \notin 1,2,\ldots,t \\ &= e(g_1,g_2)^{\mathrm{rq_x(0)}} \end{aligned} \tag{2.7}$$

    2. *If x is a non-leaf node:* For all nodes *z* that are children of x, it invokes DecryptNode(CT; SK; z) and stores the output as $\mathrm{F_z}$. Sx denotes a $\mathrm{k_x}$-sized set of child nodes z such that $\mathrm{F_z} \neq \emptyset$. If there is no such set, it implies that the node is not satisfied and the function

returns $\emptyset$. Otherwise, we compute

$$F_{\mathrm{x}} = \prod_{z \epsilon S_{\mathrm{x}}} {F_{\mathrm{z}}}^{\lambda_{\mathrm{i}}}$$

$$(i = index(z)\ \lambda_{\mathrm{i}}\ calculated\ \forall z \epsilon S_{\mathrm{x}})$$

$$= \prod_{z \epsilon S_{\mathrm{x}}} (e(g_1, g_2)^{\mathrm{rqz}(0)})^{\lambda_{\mathrm{i}}}$$

$$= \prod_{z \epsilon S_{\mathrm{x}}} (e(g_1, g_2)^{\mathrm{rq}_{\mathrm{parent(z)}}{}^{\mathrm{index(z)}}})^{\lambda_{\mathrm{i}}}$$

$$= \prod_{z \epsilon S_{\mathrm{x}}} (e(g_1, g_2)^{\mathrm{rq_x(i)}})^{\lambda_{\mathrm{i}}}$$

$$= e(g_1, g_2)^{\sum_{z \epsilon S_{\mathrm{x}}} \mathrm{rq_x(i)}\lambda_{\mathrm{i}}}$$

$$= e(g_1, g_2)^{\mathrm{rq_x(0)}} \tag{2.8}$$

Algorithm calls DecryptNode recursively starting at root node R of the access tree.

Let $A = DecryptNode(CT;SK;\ r) = e(g_1,\ g_2)^{rq_{R}(0)} = e(g_1,\ g_2)^{rs}$ at root node R. Decryption can be done as follows:

$$P = \frac{\tilde{C}}{\frac{e(C,D)}{A}} = Me(g,g)^{\alpha \mathrm{s}} \frac{e(g,g)^{\mathrm{rs}}}{e(h^{\mathrm{s}}, g^{\frac{\alpha+r}{\beta}})}$$

$$= Me(g,g)^{\alpha \mathrm{s}} \frac{e(g,g)^{\mathrm{rs}}}{e(g^{\beta \mathrm{s}}, g^{\frac{\alpha+r}{\beta}})}$$

$$= Me(g,g)^{\alpha \mathrm{s}} \frac{e(g,g)^{\mathrm{rs}}}{e(g,g)^{(\alpha+\mathrm{r})\mathrm{s}}}$$

$$= M. \tag{2.9}$$

2.5.3.2 Limitations

The PIRATTE scheme improves the Bethencourt et al.'s CP-ABE scheme [21] with scalable revocation without requiring re-encryption and re-distribution of keys. *However, it can revoke only a certain number of users and hence is not scalable.*

2.5.4 Permanent Revocation in Attribute-Based Broadcast Encryption [39]

The permanent revocation scheme (referred to as PERMREV in this thesis) proposed by Dolev et al. [39], modifies Bethencourt et al.'s CP-ABE scheme [21] to satisfy requirements *C1,C3 and C4* with the help of a trusted proxy server. PERMREV associates a counter *CTR* with the ciphertext and a user state $State_i$ for *ith* user $user_i$. It considers ciphertext to reside on a secure cloud-based system. For revocation of $user_i$, the secure server updates *CTR*, re-encrypts the ciphertext, and sends the updated $State_i$ with the new *CTR* only to non-revoked users. The decryption fails for the revoked users because they do not receive any updated state after a change in *CTR*.

To avoid re-encryption of the ciphertext, this thesis denotes the Modified PERMREV scheme as M-PERMREV. It requires a server to broadcast *State* to all users. For a revoked user, the server updates the *CTR* and the user state for only revoked users, which causes failure of decryption. The scheme is collusion resistant. *However, M-PERMREV scheme does not fulfill requirement C5 because it associates a constant CTR with the user's state* $State_i$ *for every ciphertext*.

For PERMREV scheme, each user from a receiver set maintains the state $State_i$ and a secret counter *CTR* such that the $State_i = f_i(CTR)$, where f_i is a function. When a user u_j is revoked from the receiver set, the broadcaster updates the counter variable *CTR* to a new secret value $C\tilde{T}R$, and broadcasts its encrypted value to all non-revoked users. As a result, the state of each non-revoked user is updated. The ciphertext is also associated with the secret value and is updated after the update. Hence, decryption is successful only for non-revoked users and fails for the revoked users.

2.5.4.1 Construction

- **Setup:** The trusted server chooses bilinear group G_0 and random elements α and $\beta \in Z_p$ to generate a public key *PK*, which is similar as in the PIRATTE scheme in equation 2.1. The trusted server also generates a master key *MK* with a secret constant *CTR* as follows:

  $$MK = \beta, g^{\alpha}, CTR \tag{2.10}$$

- **KeyGen (MK,S):** The algorithm generates a secret key *SK* for a set of attributes *S*. For each

user_i the trusted server chooses random numbers r_i and $r_{ij} \epsilon Z_p$ for each attribute $j \epsilon S$. It also uses a component E_i, which represents the unique state of a user u_i. It is a function of *CTR*. The private key D is:

$$D = (g^{\frac{\alpha+r_\text{i}}{\beta}}, E_\text{i} = e(g,g)^{\text{r}_\text{i}.\text{CTR}}$$

$$\forall j \epsilon S : D_\text{j} = g^\text{r} H(j)^{\text{r}_\text{j}}, D'_\text{j} = g^{\text{r}_\text{j}}) \tag{2.11}$$

- **Encrypt (PK,M,τ):** The tree structure τ represents the access policy with attributes at leaves and threshold of k-of-n gates at the interior nodes. q_x is the polynomial at node x with degree $d = k - 1$, where k is the threshold value of the node. For all OR nodes and leaf nodes, the polynomial degree is 0. The algorithm chooses a random secret $s \epsilon Z_p$ for a message M, such that for root node *R*, $q_R(0) = s$. In this algorithm the secret s is modified by the constant *CTR* as $s_2 = (- s - CTR \ mod \ p)$. The secret is distributed from top to bottom for all other nodes, $q_x(0) = q_{parent(x)}(index(x))$, where *index(x)* is a number associated with x between *1* and *num* (number of children of *parent(x)*). Y is the set of leaf nodes in the access tree τ. The ciphertext *CT* is:

  $$CT = (\tau, \tilde{C} = M.e(g,g)^{\alpha \text{s}_2}, C = h^{\text{s}_2})$$

  $$\forall y \epsilon Y : C_\text{y} = g^{\text{q}_\text{y}(0)}, C'_\text{y} = H(att(y))^{\text{q}_\text{y}(0)} = g^{\text{h}_\text{y}\text{q}_\text{y}(0)}) \tag{2.12}$$

  $H{:}0,1^* \rightarrow G$ is a hash function that maps a string attribute to a random element in G and $h_x = log_g H(att(x))$.

- **Decrypt (CT,SK)** The decryption algorithm is according to Bethencourt et al.'s CP-ABE scheme [21].

  $\text{A}_\text{i} = \text{e (g, g)}^{\text{r}_\text{i}\text{s}_2}$.

Decryption of plaintext P can be done as follows:

$$P = \frac{\tilde{C}}{e(C,D).A_{\mathrm{i}}.E_{\mathrm{i}}}$$

$$e(C,D) = e(g^{\beta}s_2, g^{\frac{\alpha+r_{\mathrm{i}}}{\beta}})$$

$$= e(g,g)^{(\alpha+\mathrm{r_i})\mathrm{s}_2}$$

$$= e(g,g)^{\alpha\mathrm{s}_2}.e(g,g)^{\mathrm{r_i(\text{-}s\text{-}CTR)}}$$

Hence,

$$e(C,D).E_{\mathrm{i}} = e(g,g)^{\alpha\mathrm{s}_2}.e(g,g)^{\mathrm{\text{-}r_i s}}$$

$$e(C,D).E_{\mathrm{i}}.A_{\mathrm{i}} = e(g,g)^{\alpha\mathrm{s}_2}$$

Hence,

$$P = \frac{\tilde{C}}{e(C,D).A_{\mathrm{i}}.E_{\mathrm{i}}}$$
$$= \frac{M.e(g,g)^{\alpha\mathrm{s}_2}}{e(g,g)^{\alpha\mathrm{s}_2}}$$
$$= M \tag{2.13}$$

The broadcaster updates *CTR* in *MK* as- *CTR : CTR +s mod p.* The non-revoked user updates the state E_i in its private key as

E_i: E_i. $A_i = e\ (g,g)^{r_iCTR}e(g,g)^{r_is} = e(g,g)^{r_i(CTR+s)}$.

Decryption is successful because the *CTR* has been updated in the ciphertext as well as the non-revoked private key. Other revoked users cannot compute the function for *CTR,* $e(g,g)^{r_iCTR}$ by collusion with other users. Hence, the revocation is permanent.

2.5.4.2 Limitations

The PERMREV and the M-PERMREV schemes provides scalable revocation, no prior knowledge of the revocation list, and no re-distribution of keys. M-PERMREV improves the PERMREV scheme since it fulfills requirement C2. *However, M-PERMREV has a disadvantage that it associates a constant parameter with the ciphertext, which is updated for revocation. Hence, it does not support requirement C5.*

2.6 Remote Attestation for Trust

According to Asokan et al. [16], "The term *Trusted Computing* is used to collectively describe technologies enabling the establishment of trust in local and remote computing systems by using trustworthy components and trust anchors to ensure the integrity of other parts of the system." Trusted Computing Group (TCG) [139] has been leading the standardization efforts in trusted computing. Sherpard et al. [134] refer to the different technologies for trusted computing. According to Asokan et al. [16] the basic security mechanisms in a mobile system are as follows:

- *Platform Integrity:* The integrity of the device OS code is verified either during system boot or device runtime. The platform providers can detect any unauthorized changes made in the OS. In case of a secure boot, if the validation fails, the platform provider aborts the boot process.
- *Secure Storage:* The secure storage device stores data to disallow unauthorized access and store a device-specific key for confidentiality and integrity. The key can be accessed only by authorized code. It must also have necessary cryptographic mechanisms, such as an authenticated encryption algorithm.
- *Isolated Execution:* It provides the ability to run the security-critical code outside the control of the untrusted environment.
- *Device Authentication:* External device authentication can help verify the identity of the mobile device, which may include device manufacturer information. The device identity can be the *International Mobile Equipment Identifier (IMEI)* or link-layer identities, such as Bluetooth and Wi-Fi addresses. The device manufacturers sign a device certificate, and their public key can later assist in its verification. The device identities can be signed using the device public key and verified by an external verifier.
- *Attestation and Provisioning:* Attestation is a process between two devices: a prover and a verifier, which helps the verifier ascertain the trustful software state of the prover [4]. Remote

attestation occurs when the prover and the verifier communicate over an interface, such as Bluetooth, NFC or wired and wireless TCP/IP. It comprises of:

  - The prover obtains evidence of its current state through the measurement process.
  - The prover conveys the result of attestation to the verifier.

  The software and firmware status is signed with a certified device key and verified by a remote device or server. The secrets and code can be sent securely to the target device through a process called provisioning.

This thesis looks into trustful computing with attestation to verify the software state of the devices. It ensures that no malware can compromise the devices and cause a security breach. A mobile device could use either TEE or TPM for attestation as discussed below:

- **Trusted Execution Environment (TEE):** TEE is a trustful execution environment that runs alongside the main operating system, called *Rich Execution Environment (REE)*. Its objective is to provide security services for processing. TEE can enhance the security of a mobile device in collaboration with hardware, for example, dedicated storage, such as SE, and dual mode CPUs, and software, for example, secure kernel, and separated drivers facilities. TEE has dedicated resources that are isolated from REE and its applications and protects the applications residing on the TEE against a range of physical attacks. *However, it is not tamper-resistant like SE*. Each TEE holds its cryptographic resources, such as the private key and certificate that are hard-wired in a read-only memory. It fulfils the following security requirements:

  - *Isolated Execution:* Ensures that applications can execute completely isolated from other applications.
  - *Secure Storage:* Protects persistently stored data, for example, cryptographic keys.
  - *Remote Attestation:* Enables remote parties to ascertain the trustful state of the device.
  - *Secure Provisioning:* Enables communication by remote parties with a specific trusted application, thereby protecting the integrity and confidentiality of transmitted data.

  - *Trusted Path:* Provides a channel that enables data exchange between the user and the TEE and protects against eavesdropping.

  Global Platform [56] specifies the TEE functionality for mobile devices. TrustZon is the most common implementation of TEE. It has been deployed by ARM10 for the ARM Cortex Processor family. Other TEE implementations are Intel Identity Protection Technology and Texas Instrument MShield. Yang et al. [159] proposed a *Direct Anonymous Attestation (DAA)* using Trustzone called DAA-TZ for mobile devices. It preserves device anonymity from remote service providers.

- **Trusted Platform Module (TPM):**

  TPM resides on the motherboard of a computing platform and is resistant to physical attacks. It contains functions for key generation, asymmetric and symmetric encryption, and digital signature. All cryptographic processing is in an isolated manner. TPM is a co-processor, which protects cryptographic keys and records the software state of the device [4] and connects with the software and hardware architecture of a system. TPM can assist in remote attestation to report and ascertain the software state to the remote host over communication interfaces, such as the Internet or low-energy interface, such as NFC [142].

  For attestation, before any component takes control of the CPU, the device saves the measurement of its characteristic code and configuration into TPM's *Platform Configuration Registers (PCR)*. For every measurement of the system state, the device stores it in a *Stored Measurement Log (SML)*. The PCR values are later used with the SML to attest the device state to a remote party with the help of a *Attestation Identity Key (AIK)* to sign and authenticate these values. The remote machine compares the signed values with the reference values to determine if the device is in a trustful state. TPM also has a *Endorsement Key (EK)*, which is generated by the manufacturer to uniquely identify it and obtain an AIK from a trusted server.

Table 2.3 describes the comparison between the two technologies for attestation. Since TPM is hardware-based and more secure than TEE, hence we consider the TPM-based attestation techniques in this thesis.

The following sections provide the details of the attestation techniques as discussed in Section 1.4.3.1. Table 2.4 describes their limitations as compared to the proposed NSE-AA protocol in this thesis.

Table 2.3: Comparison of Trusted Computing Platforms [104, 121]

Issue	**TPM**	**TEE**
Performance	Poor	Medium
Storage	Few bytes	Large
Speed	Low	Fast
Physical attack	No	Yes

Table 2.4: Limitations of Attestation Techniques

Requirement	**Toegl and Hutter [142]**	**Aziz et al. [19]**	**NSE-AA**
Proof-of-locality	Y	N	Y
User anonymity	N	N	Y
Secure Storage	N	N	Y
DoS attack	N	N	Y
Parallel session attack	N	N	Y
MITM attack	N	N	Y
Insider attack	N	N	Y

2.6.1 Locality in Remote Attestation using NFC for Mobile-based Kiosk Access

It is essential to verify that public computer systems are secure, free from malware, or exposed to altered software. Customers may want to assure that the POS device for billing or *Automatic Teller Machines (ATMs)* machines are free from malware. It must not collect the PIN or other relevant information and use it later for fraud. Voters may want to verify that there is no tampering of the electronic voting machines before casting their vote to have a trustful outcome of polls.

Kiosk computing: The public computer systems can be deployed for access to various places, such as shops and cafes. They may have internal storage as well as a software program for operation of the computer system. An attacker could visit the kiosk several times and get access to the kiosk to install malicious software. Hence, the kiosk cannot be trusted. Togle and Hutter [142] proposed a novel scheme to assure trustful states of such public systems. The scheme uses a TPM-based attestation scheme to ascertain the trustful state of the kiosk.

The scheme consists of a user's mobile device, which can help to interact with a local computer system such as a kiosk using NFC. It performs the TPM-based attestation of the kiosk and conveys the report to the user. NFC provides proof-of-locality for the attestation process to ensure that the devices are present physically and reduces the chances of MITM attacks.

MAT protocol- The proposed scheme uses an NFC-based mobile device as a *Mobile Attestation Token (MAT)* to access a TPM-based kiosk. In this scheme, the mobile device generates a random nonce N_A and sends it as part of a challenge to the kiosk. The kiosk passes it internally to its TPM module. TPM signs its attestation report and nonce N_A with the private attestation AIK and sends it to a trusted *Virtual Server(VS)*. The server validates it and prepares a validation ticket for the user's mobile device.

The scheme introduces proof-of-locality in the remote attestation process and makes MITM attacks difficult. *However, the mobile and kiosk use insecure NDEF messages, and there is no validation of the mobile device, which may also be prone to malware, especially on rooted devices and cause a breach of trust [136, 7].*

2.6.2 Extended TLS with Attestation

Transport Layer Security (TLS) is a cryptographic protocol that ensures secure transmission of data and the authenticity of communication between client and server. However, it does not assure the trustful state of devices. Aziz et al. [19] provided an extension of TLS for asymmetric encryption-based mutual authentication and TPM-based mutual attestation over TCP/IP. It is secure against replay and collusion attacks. The safety proof is verified through simulation using the AVISPA tool with *High-Level Protocol Specification (HLPSL)*. It uses the Dolev-Yao intruder model [38] for the proposed protocol. The extended protocol comprises of the following protocols:

1. *P1:Registration Protocol-* The public *Endorsement Key (EK)* uniquely identifies a TPM and is certified by a *Trusted Certifying Authority (TCA)*. Each host must register with TCA, present its public key EK, and receive the EK certificate.
2. *P2:AIK Certificate Creation Protocol-* Each host with identifier id_{host} generates a unique random nonce N_H and exchanges it in the TLS session to generate a session key K_S. It then generates an AIK certificate for the validity of its AIK identifier id_{AIKH}*: h(*id_{host}*,*N_H*)* and public AIK Pk_{AIKH}.
3. *P3: TPM-based Attestation Protocol-* The protocol consists of a TPM challenge-response authentication. Both devices use nonces used in the previous authentication phase for freshness. It uses Medium Authentication Code (MAC) of the TLS session key for encryption of all attestation messages to ensure authenticity. In the attestation phase, the host sends its attestation report consisting of the PCR and SML values signed by its AIK private key Sk_{AIKH} along with its AIK certificate $Cert_{AIKH}$ to the remote host to verify its trustful state.

   An AIK certificate provides anonymity of the device from server and unlinkability from an eavesdropper. Both client and server attest each other to ensure the integrity and to establish trust.

The scheme prevents replay and collusion attack. A compromised host cannot attest because each new attestation phase uses a fresh nonce. Hence, the AK certificate verification fails in case a host tries to replay messages to attest. Similarly, the collusion attack prevents a host to use the trustful device's attestation information for personal attestation. AIK certification fails and prevents collusion due to the use of nonces in an AIK identity. *However, the scheme has drawbacks of being computationally expensive due to asymmetric encryption, no support for device anonymity, no proof-of-locality, and secrecy of PCR and SML because the host sends an unencrypted* $Cert_{AIKH}$.

2.6.2.1 Formal Verification Using the AVISPA Tool

Aziz et al. [19] prove that the Extended TLS protocol is safe through simulation using the *Automated Validation of Internet Security Protocols and Applications (AVISPA)* tool [14]. The protocol is safe against the Dolev Yao attackers [38]. Both the client and the server authenticate each other and generate a session key in the authentication phase. In the attestation phase, they share the SML data for verification and establishment of trust.

The AVISPA tool uses *High-Level Protocols Specification Language HLPSL)* [49] to specify a protocol for simulation. The protocol is simulated using the *Security Protocol ANimator for AVISPA (SPAN)* tool. The HLPSL script comprises of entities with independent roles, number of roles, sessions, and principals. It incorporates an intruder (i) using the Dolev-Yao model [38]. In this model, an intruder who is a valid user can fully control all transmission messages over the network. The tool converts the HLPSL code to the intermediate format (IF) using the HLPSL2F translator. It sends the IF form to one of the backends, which are:

- On-the-fly Model-Checker (OFMC)
- Constraint-Logic-based Attack Searcher (CL-AtSe)
- SAT-based Model-Checker (SATMC)
- Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)

The backends on execution produce an output format (OF). It has a SUMMARY, which indicates whether the protocol is safe or unsafe, or has inconclusive analysis. The DETAILS section explains the conditions under which the protocol is safe, conditions for finding an attack, or the reason for inconclusive analysis. The other sections are PROTOCOL for the name of the protocol, GOAL for the analysis, and BACKEND for the backend name.

The HLPSL supports basic types of data:

- *agent:* principal names

- *symmetric_key:* secret symmetric key
- *public_key:* public key
- *hash_func:* hash function
- *nat:* natural number

If the public key is *pu*, then the private key is the inverse of the public key and is denoted by *inv_ku*. If N is a typed text, then N' is a fresh value.

Specifying protocol- The entities are represented as roles. There are also roles for session and environment. A role can send and receive messages from other roles using SEND() and RECV() operations. Based on the messages received the roles transition from one state to the other. If the protocol must keep information in a variable *V* secret permanently, then there must be a goal *secrecy_of V*.

In the session segment, all basic roles including roles for entities, are instantiated with arguments. The role environment, contains global constants, the composition of one or more sessions, and knowledge of intruder's behaviour.

The goal section defines security properties. HLPSL supports goals for strong and weak authentication and secrecy goals. In a weak authentication role, role B authenticates that role A has sent a message. While in a strong authentication, which is an extension of the weak authentication, role B assures that there is also no replay of messages. The HLPSL script specifies authentication goals using *witness* and *request* command. It specifies the secrecy goal using the *secret* command. For all, constant *id* identifies the goal in the goal section. It has four predefined goal commands are as follows.

- *secret(E,id,S):* Defines information *E* is a shared secret between agents of set *S*.
- *witness(A, B, id, E):* Defines weak authentication property of agent A by agent B on information E. It denotes that agent A is a witness for the information E.
- *request(B, A, id, E):* Defines strong authentication of agent A by agent B on information E. It denotes that the agent B must request a check of the value E.

- *wrequest(B,A,id,E):* It is similar to requesting a weak authentication property.

2.7 Summary

The chapter presents the existing techniques and the key technical background information for the work done in this thesis. The existing portable health record management systems do not fulfill patient mobility across hospitals. NFC can be used for active and secure direct access to health information from a portable health device. The NFC-based HCE mode has been used in traditional financial applications. It has several advantages over the existing modes and can be used for bidirectional support for a robust security handshake. Although NFC provides proof-of-locality, there are several security threats which must be overcome. The existing NFC security schemes either focus on authentication or attestation for trust. There is a need to have mutual authentication and attestation to establish security handshake between the two devices and ensure their trustful states. The existing CP-ABE schemes based on Bethencourt et al.'s scheme [21] lack support for scalable revocation without re-encryption and redistribution of keys for sharing of information securely from a portable device. NFC-based mobile-based devices must prevent misuse of the information that they exchange and forward to an adversary, such as in case of a relay attack. Hence, there is a need for mutual attestation schemes to ensure trustful states of devices that communicate over NFC. None of the existing schemes provide the mutual attestation over NFC to the best of our knowledge.

CHAPTER 3

SYSTEM DESIGN FOR SMART HEALTH RECORD MANAGEMENT SYSTEM

This chapter presents the details for architecture and design for a next-generation portable Smart Health Record Management system with secure NFC-enabled mobile devices to retain dispersed health records. Current mobile devices lack the support for usability across different hospitals, as discussed in Section 2.1. Since mobile devices are widely used and have improved computational and storage capabilities, they can help in a portable health record management system. The proposed system provides secure yet easy access to updated health history and assists patient mobility across hospitals. The patient mobile device can retain health records and be used as a contactless health wallet. Different medical professionals can directly read and write selective records from their mobile device as the reader device. The following sections describe the system design and the architecture for the proposed system.

3.1 Proposed System Design and Architecture

A patient's NFC-enabled mobile device aggregates dispersed health records on a *Secure Mobility-Assisted PortabLE (S-MAPLE)* health folder as an HCE-based contactless card. It can be accessed by the mobile device of an authorized medical professional over low energy wireless interfaces, such as NFC and Bluetooth and locally by the patient as well. NFC-based proof-of-locality, SEs, end-to-end mutual authentication with attestation protocol, and a variation of Bethencourt et al.'s CP-ABE scheme [21] to secure the proposed system. A cloud-based service provides data aggregation, translation of health records, management of credentials, and a secure digital vault for backup. The S-MAPLE health folder stores dispersed health records in a standard format, such as HL7.

We consider a scenario of patients visiting a hospital with dispersed records to seek treatment. They first register to the administrator through a kiosk machine over NFC. It helps in the setup of the current *Out Patient Department (OPD)* session keys on the patient and medical professional's mobile devices. Later, during the OPD session, a physician can tap a mobile-based reader to the patient's mobile-based S-MAPLE health folder. The physician can directly access the health folder over the NFC tap to read the health history, diagnose, and write back a prescription to the health folder. Figure 3.1 illustrates a use case of an OPD session.

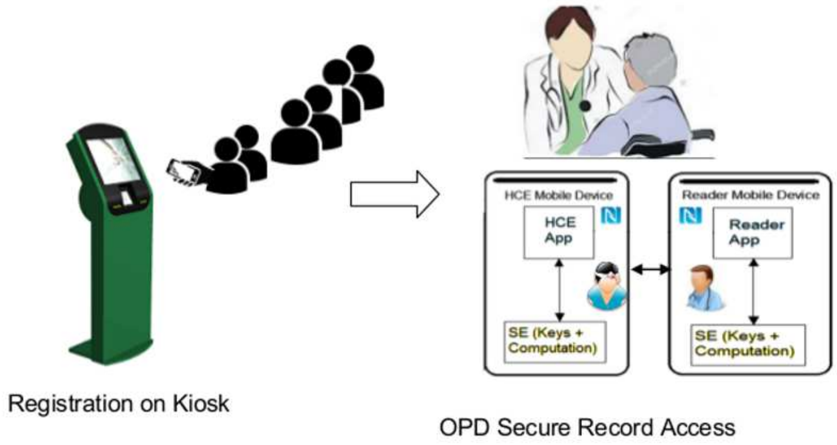


Figure 3.1: Health Management Flow with S-MAPLE Health Card

S-MAPLE health folder assists in patient mobility across hospitals and recent aggregated health history. Table 3.1 summarizes how the S-MAPLE health folder satisfies various requirements *R1-R7* for patient mobility across hospitals.

Table 3.1: Requirements Fulfilled by the S-MAPLE Health folder Architecture

Requirements	**Method**
R1: Aggregation	Storage of health records in a standard HL7 format and data aggregation and translation
R2: Up to date	New records written directly to the S-MAPLE health folder as well as on the local HIS
R3: Usability across hospitals	Direct access to the S-MAPLE health folder in different hospitals
R4: Availability	Storage of recent few years of health records and past health summary is readily available
R5: Easy Accessibility	Ease of access with NFC tap between the devices
R6: Selective Access	Selective RBAC with the SPIRC scheme based on CP-ABE
R7:Security and Privacy	Secure storage on SE; NSE-AA protocol for end-to-end mutual authentication with attestation; SPIRC for confidentiality, selective RBAC and scalable revocation; NFC for proof-of-locality and Digital vault to refurbish lost health folder

In the following sections, we describe the smart health record management system with a secure NFC-enabled mobile device for retaining the S-MAPLE health folder.

3.1.1 S-MAPLE Health Folder Organization

The portable NFC-based health folder retains different health records from various hospitals. It translates the health records to a standard HL7 health format using translation tools such as Mirth connect [102]. Hence, the system fulfils the requirement *R1* for aggregation. It maintains the health folder as a *JavaScript Object Notation (JSON)* file, which is lightweight and fast to access as compared to the traditional XML file format.

The health card application pre-parses the health records with the *HL7 Application Programming Interface (HAPI)* [67] parser because HL7 is cumbersome to parse. The S-MAPLE health folder which is a JSON file maintains two arrays, one for the HL7 data and the other for the pre-parsed non-HL7 data. It uses the parsed non-HL7 data for efficient visualisation of the health records. Table 3.2 describes the layout of a sample health folder for two departments, Oncology, and Cardiology and the access rights for different medical professionals.

Each subsection of a department record is encrypted using the newly proposed *Scalable Proxy-based Immediate Revocation for CP-ABE (SPIRC)* scheme. It encrypts the health folder for confidentiality, selective RBAC, and scalable revocation. The details of the SPIRC scheme are described later in Chapter 6. It is a variation of Bethencourt et al.'s CP-ABE [21] scheme, which fulfills all requirement *C1-C5* in Section 1.4.4 for sharing data from a portable device. Each section of a department is encrypted with a read access policy and a write access policy as discussed later in Section 6.3.

Physicians can access the S-MAPLE health folder by tapping their mobile device over NFC to read the past health records as well as a summary of the old health records. In case, they require much older health records; they may access them from the digital vault using SPIRC-based delegation of cryptographic keys as discussed in Section 6.2.3. A physician can diagnose a patient's health problem with the current symptoms as well as the previous health history and tap to write a new prescription on the health folder.

Table 3.2: JSON Health Folder with HL7 Health Records

Department	**Roles**	**Basic Vitals**	**Allergies / Diseases**	**Advanced Vitals**	**Drugs**	**Lab Tests / Vaccination**	**Emerg./ Admin.**
Oncology	Doctor	RW	RW	RW	RW	RW	R
	Nurse	RW	R	R	R	R	R
	Pharm.	—	—	—	RW	—	R
	Lab Tech	—	—	—	—	RW	R
	Emerg.	RW	RW	RW	RW	RW	R
	Patient	R	R	R	R	R	R
	Admin	R	R	R	R	R	W
	Read Access	ACRB	ACRB	ACRB	ACRM	ACRL	ACRALL
	Write Access	ACWBV	ACWSP	ACWSP	ACWM	ACWL	ACWADM
Cardiology	Doctor	RW	RW	RW	RW	RW	R
	:	:	:	:	:	:	:

With the size of 10 health records on a JSON file, around 57KB, an X-ray report around 2MB and MRI scan report around 200 MB, the S-MAPLE health folder can easily store most recent health records on the current mid-ranged priced mobile devices. Even with 100 records (OPD and lab tests), 10 XRay images, and 2 fMRI scans, the space required is less than 1 GB. We feel that modern mobile devices have a minimum of 16 to 32 GB of RAM, which is adequate to store few years of records and summary information. Hence, it can provide readily available past health information for the patient and satisfies requirement *R4* for availability. It can be presented as a mobile-based health wallet and accessed by various authorized health professionals across different hospitals. Hence, it satisfies requirement *R3* for usability across hospitals.

3.1.2 System Model

Let us consider a scenario where patients with dispersed health records visit a hospital to seek treatment. They must register with the administrator before they can consult a physician for an OPD session.

Figure 3.2 demonstrates a system, which involves a patient's mobile device **P** that retains the S-MAPLE health folder **F** with dispersed health records in HL7 format. **P** maintains a software-based contactless card using NFC-based HCE mode, which can access the health folder **F** internally as well as support the exchange of health information with the reader device over an NFC tap. NFC provides ease of access and thus helps satisfy requirement *R5*.

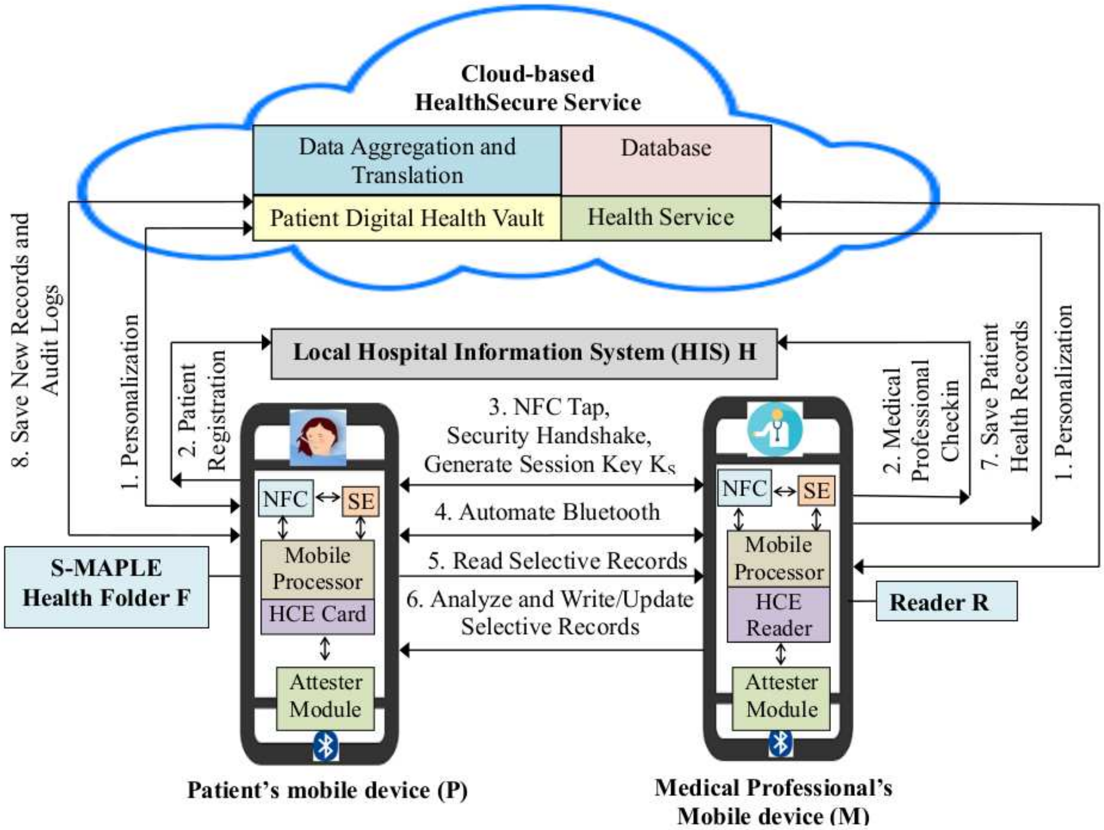

Cloud-based
HealthSecure Service
Data Aggregation and Translation
Database
Patient Digital Health Vault
Health Service
8. Save New Records and Audit Logs
1. Personalization
2. Patient Registration
Local Hospital Information System (HIS) H
2. Medical Professional Checkin
7. Save Patient Health Records
1. Personalization
3. NFC Tap,
Security Handshake,
Generate Session Key K_S
4. Automate Bluetooth
5. Read Selective Records
6. Analyze and Write/Update Selective Records
NFC
SE
Mobile Processor
HCE Card
Attester Module
NFC
SE
Mobile Processor
HCE Reader
Attester Module
S-MAPLE
Health Folder F
Reader R
Patient's mobile device (P)
Medical Professional's
Mobile device (M)

Figure 3.2: System Model for Smart Health Record Management System

A medical professional uses a mobile device **M** with an HCE-based reader application **R** to tap and access the HCE card directly for selective access using the SPIRC scheme. Multiple stakeholders can access each section with selective RBAC. The card and the reader applications support the HL7 health format. A cloud-based HealthSecure service **HSS** provides data aggregation and translation of health records, management of cryptographic credentials, and a secure digital vault for backup of the health folder. All devices store cryptographic credentials on a SE, which provides tamper-resistant storage and performs secure cryptographic computations. Figure 3.2 demonstrates the flow of the interactions between the components and the steps are listed below:

1. The HealthSecure service assists in the personalisation of SEs of valid users to store the

credentials and identity on their mobile devices.

2. The patient and the medical professional register and check-in respectively for an OPD session in the hospital.

3. During an OPD session, the patient and the medical professional tap their devices close for initiating a security handshake with an end-to-end *NFC SE-based Mutual Authentication and Attestation (NSE-AA) protocol* proposed in this thesis to verify that only valid and trustful devices interact. It sets up a unique session key, which encrypts all further communication. Chapter 5 discusses the details of the NSE-AA protocol.

4. The HCE tap further automates Bluetooth pairing over HCE to provide higher throughput for the fast communication of large data, such as medical images. Mobile devices without NFC can alternatively use secure QR-Code to automate Bluetooth pairing using inbuilt cameras [146].

5. The reader application interfaces with the card application using a bidirectional HCE or Bluetooth interface to selectively read old dispersed health records for the last few years and a summary of older health records.

6. The medical practitioner analyzes the health records, and adds the new observations, diagnosis, and prescription as a new health record and writes it on the health folder over an HCE interface using the HL7 format. Hence, the health folder updated health records and satisfies requirement *R2*.

7. The reader device uses existing translation services on the HealthSecure service to translate the HL7 format and store them on the HIS in the required local health format.

8. The health folder stores the health records and the audit logs on the secure digital vault for future reference.

3.1.3 System Architecture

This section describes various software components for the system architecture and illustrates them in Figures 3.3 and 3.4.

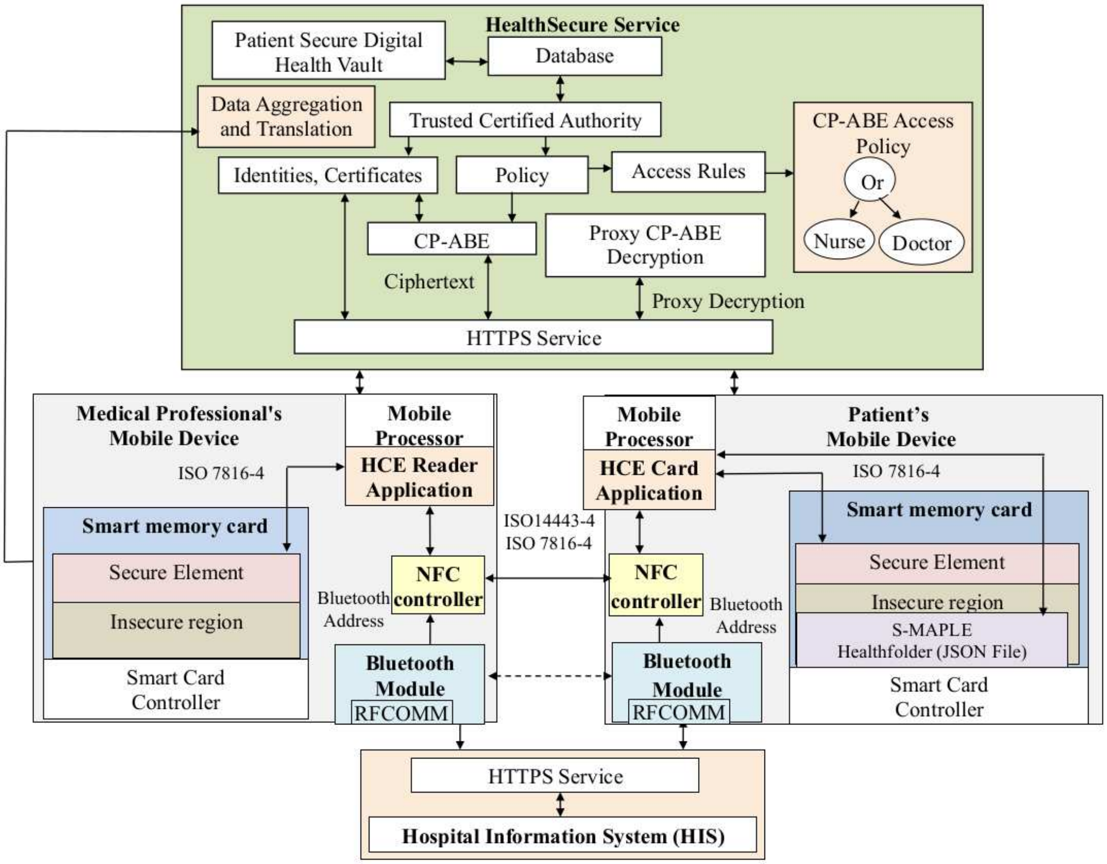


Figure 3.3: System Architecture

- **Patient/Medical professional mobile devices:** The patient's mobile device retains a card application on the mobile processor to emulate an HCE-based contactless card. The encrypted S-MAPLE health folder resides in an insecure region, such as the internal memory or a microSD card. The proposed system uses a special microSD card with insecure storage and a secure embedded tamper-resistant SE.

  The medical professional's mobile device retains an HCE reader application on the mobile processor. It can access the S-MAPLE health folder using NFC-based HCE bidirectional library and Bluetooth.

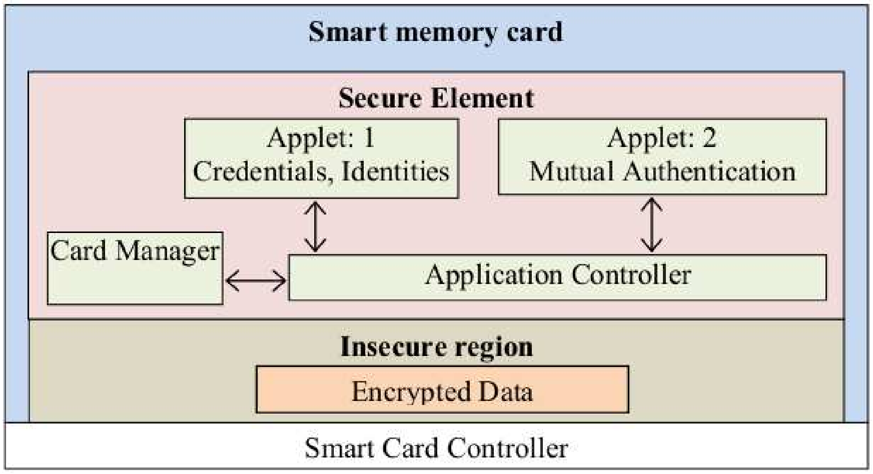


Figure 3.4: Smart Memory Card Architecture

As discussed in Section 2.4.1.1 the SE is based on a Java Card and contains Java Card applets for the following features:

1. Retain identity, certificates, and decryption keys.
2. Perform cryptographic computations for the end-to-end NSE-AA protocol over HCE.

- **HealthSecure Service:** It is a cloud-based service, which provides the following services for the secure portable health system:
  - *Data aggregation and translation:* Both the card and the reader devices exchange health records using the HL7 format. The HealthSecure service uses existing tools, such as Mirth Connect [102] to help translate health records so that the medical professional can store them on the HIS also in the local health format.
  - *Trusted Certified Authority (TCA):* TCA administers cryptographic credentials and identities of registered patients and medical professionals. The Healthsecure service administrator or the patient may define the access policy for the SPIRC scheme and allocate decryption keys to the stakeholders. The S-MAPLE framework outsources the SPIRC encryption to the HealthSecure service, due to the computational overheads of bilinear pairing. TCA also provides a trusted proxy server to support partial decryption

for the SPIRC scheme for managing a revocation list and for providing the proxy components over HTTPS to assist decryption on the mobile devices.

  - *Secure Digital Vault:* The patient's mobile device data syncs all new records on the secure digital vault. It can be used to refurbish the health folder in case of loss or theft of the device and to access old records that are not available in the health folder.

- **Health Information System (HIS):** It maintains EHRs locally on the hospital's database, such as the openMRS [114] system using translations tools from the HealthSecure service.

3.1.4 Bidirectional HCE Communication

Typically payment applications use bidirectional HCE communication [7]. The application of HCE for a mobile-based health wallet presented in this thesis for patient mobility across hospitals does not exist to the best of our knowledge. The HCE-based cards can use proprietary APDU packets for unique communication protocol and enhance data security and trust. However, the APDU command and response packets can carry data upto to 255 bytes [13] on currently available Android-based devices. This thesis proposes an HCE library for the communication of large-sized data between the card and the reader for reading and writing data, which is larger than 255 bytes. Both card and reader devices can send and receive data to each other. The sender device fragments large data and sends the fragments in multiple packets. The receiver device further reassembles them. The HCE-based communication comprises of a bidirectional protocol with error control for reading and writing to the health folder.

The bidirectional HCE tap helps achieve the following:

- Identify mobile devices and ensure their trustful states using the NSE-AA protocol.
- Automate Bluetooth pairing for higher throughput.
- Exchange of data for reading and writing.

Traditionally Peer-to-Peer NFC mode is used to automate Bluetooth pairing without manual intervention using the insecure NDEF messages [126]. The proposed system uses the HCE mode to

Table 3.3: HCE Bidirectional Read and Write Modes (Steps 3-8 Read; 9-16 Write)

S.No	Messages	Description
1.	Card/Reader:	Open Applications
2.	Card⟷Reader:	Select Card
3.	Card←Reader:	Read Command
4.	Card:	Fragment Data
5.	Card→Reader:	Response OK || F_1 || mfp=1
6.	Card←Reader:	Repeat Read command....
7.	Card→Reader:	Response OK || F_N || mfp=1
8.	Reader:	Reassemble fragments F_1-F_N
9.	Reader:	Fragment data to write
10.	Card:←Reader:	Write Command || F_1 || mfp=1
11.	Card:→Reader:	Response OK
12.	Card:←Reader:	Repeat Write command ....
13.	Card:←Reader:	Write Command || FN || mfp=0
14.	Card:	Reassemble fragments F_1-F_N
15.	Card:→Reader:	Response OK
16.	Card:⟷Reader:	Terminate Session

exchange the Bluetooth address and establish a connection over *Radio frequency communication (RFCOMM)* sockets without any manual intervention. The NSE-AA protocol and proof-of-locality wth NFC assure that the Bluetooth pairing is between the two devices in proximity.

The HCE library has a Reader Mode and a Writer Mode. The mobile devices for both card and reader applications, start the respective applications to initiate communication. The reader device taps and selects the card *Application Identifier (AID)* and sends the read or write command to the card. The sender sets the *More Fragment Packet (mfp) flag* to 1 if more fragments are pending and 0 if there are no more fragments. The receiver reassembles all packets when it receives a packet with *mfp* as 0. When the devices lose contact, the interface terminates. Table 3.3 describes the steps for the Bidirectional HCE library for read and write.

3.2 Summary

This chapter proposes the system design and architecture of the next-generation portable smart health record management system. The proposed system fulfils all requirements *R1-R7* to support patient mobility across hospitals and recent aggregated health history. The chapter discusses the organization of the S-MAPLE health folder and how it can be accessed selectively with ease over the NFC tap. The System Model presents the details for the interaction between a patient and a medical professional mobile device for an OPD session. The chapter further specifies the details of the various software components of the system architecture. It also discusses the bidirectional HCE communication requirement for the smart health system.

CHAPTER 4

SECURITY REQUIREMENTS AND SOLUTIONS

This chapter discusses the necessary security and threat requirements for the proposed portable health record system. The chapter also highlights the focus areas for the security framework and the security solutions in brief. Based on the challenges for security and privacy discussed in Section 1.2.2, there is a need to secure the health records on the mobile-based system. The patient mobile-based system must be able to retain the health records securely as a contactless health wallet. Hence, it is essential that it is secure against the threats for a contactless card.

4.1 Security Requirements

The S-MAPLE health folder has a strong security framework to satisfy requirement *R7*. Based on the challenges for security and privacy discussed in Section 1.2.2, we have identified the following security requirements:

- **SR1: Confidentiality-** The health folder must be encrypted and be accessible only to authorized users. The framework must retain the details of the treatment and medications safely for a patient. A patient could be suffering from a disease, such as HIV, which must be confidential and shared with only selective health professionals. The information must be accessible only to the authorized stakeholders.
- **SR2: Integrity-** The health record must be written accurately to the health folder by an authorized health professional. Only valid modifications must be allowed at a later stage such as update in the diagnosis after reviewing the results of a lab report. An intruder must not be able to alter any health records. There must be provenance of dispersed health records so that different health professionals in different hospitals can consider the portable health system as reliable. The integrity of health records is important since it can impact the history of a patient, which may be considered useful especially in case of chronic ailments, such as

diabetes.

- **SR3: Mutual Authentication and Trust-** A valid patient and authorized medical professional must have unique identities and must be able to authenticate each other for a trustful session. Medical professional must ensure that the right patient is seeking treatment and there is no medical fraud. Authentication may also help eliminate errors when a nurse places an injection on the wrong patient. Similarly, patients must ensure that they are visiting and authorized health professional who has not been replaced by a junior helper or an adversary. A junior helper may try to cover the actual physician and may not have the required expertise. An adversary may impose as a physician and harm the patient with the wrong medication.

  Mobile devices may have malware that can risk the health card and reader applications. Both mobile devices must prove their trustful states to each other before they can exchange any data. The device may have malware, which can cause relay attacks and expose the patient at risk with wrong health services or costly medications and misuse of treatments by wrong patients.

- **SR4: Privacy-** It must retain privacy of patient's identity on their devices as well as during communication. Patients may not be comfortable to disclose their ailments, such as mental disorder and depression.

- **SR5: User Anonymity-** Each device must have a unique virtual identity, which is known only to the device owner. An adversary must not be able to use it for replay attacks or to find the actual identity. An intruder must not be able to find the identity of the patient by association with an ailment that must be kept confidential. Patients must not lose their identity so that an adversary can access their health records. An insider can reveal patient identity for medical fraud such as submitting false medical insurance claims, which may financially and medically harm a patient [18].

  Various anonymity techniques may be used, such as anonymity for data, user identity, communication, and unlinkability. The patient can be identifiable for treatment, billing, and health management. It may be generated using a combination of a password known to the user and credentials on the SE.

Records on the mobile can cause location disclosure of a patient. The patient may not want to share the location details with a doctor. The security framework can apply a hash function to the patient's identifier to generate an anonymous patient identifier. It can associate the hashed identity with the health records so that a patient's identity is secured.

- **SR6: Proof-of-locality-** The devices must initiate communication only when they are close and can assure proof-of-locality. The requirement for the proximity of devices must make MITM attack and eavesdropping difficult. Wireless communication such as that over TCP/IP or Bluetooth is prone to a third party device causing these attacks.
- **SR7: Secure Storage-** There must be secure storage on the mobile device to retain health records and cryptographic credentials because mobile devices are prone to security threats [37, 5]. All health records must be stored in the insecure region on the mobile devices in an encrypted format. There must be a provision to access the credentials and identities by special applications from the secure storage to decrypt the health records and view with selective RBAC.
- **SR8: Selective Access-** The health folder must be accessed using selective RBAC by medical professionals based on their roles. It may have a collection of different types of health information that must be accessed by a specific health professional as illustrated previously in Figure 1.1. For example, a pharmacist must be able to selectively access only the drugs that are prescribed and must not be able to access the details of the diagnosis or treatment.
- **SR9: Revocation-** The health folder must revoke a malicious user, such as a patient submitting wrong medical bills or an intruder impersonating as a doctor or involved in medical identity theft. However, it must allow uninterrupted access to the non-revoked users without requiring re-encryption or re-distribution of keys. Revocation is also essential in case the device falls in the hands of an intruder. The proxy server must be able to revoke the credentials so that

an intruder cannot decrypt and access the health records.

- **SR10: Delegation-** Patients must be able to temporarily delegate a decryption key to a family or friend to collect a report or medication on their behalf. Consider the case of a patient who is ill and must be taking rest at home. A series of blood tests are conducted to confirm the problem and offer correct treatment. The patient cannot visit the lab technician with the mobile device to get the reports. Instead of handing the patient's mobile device to a friend/family, the patient can delegate a portion of the decryption key and the encrypted health folder with the related health record for a visit. The helper may take the patient's partial record on the helper's mobile device along with the delegated keys. The helper taps the device to the lab technicians device to gather the lab report, retain on the device in an encrypted format, and be able to view it through decryption for a limited time frame. After the report is synced in on the patient's mobile device, the patient can revoke the delegated key on the helper's mobile device. The helper application allows decryption only in memory and take full precautions that the records are not stored physically for later misutilization by the helper.

- **SR11: Emergency-** Patients must be able to share their emergency information with emergency personnel to indicate the right treatment needed. The following methods can be used to handle emergency cases without harming the confidentiality of the health records [48]:

  - *Private-key storage:* Patients must obtain a private key from a healthcare certification authority to encrypt and store their health records securely. Patients must also store the private on a trusted server. In case of an emergency such as a patient in a coma, it may be essential that the emergency personnel must treat the patient after knowing the allergic and chronic history. In such a situation, the certification authority can recover and present the private key to the emergency personnel to enable timely access to health data.

  - *Smart card:* Patients may securely retain the health records on a smart card or use it to access the remote health records. They may use it along with some additional information with the card owner to enable access. In case of an emergency, there must be a provision to recover the keys to access the necessary information by the emergency personnel.

  - *Emergency responder:* In case of an emergency, the patients may not be able to manage their health records. A trusted person known as the emergency responder must be allowed access to manage the health records for a limited duration of time to provide timely access to health history as well as protect patient's privacy.
  - *Break The Glass:* The security framework must support special access to the health folder using the *Break the Glass (BTG)* key [54, 48]. At the beginning of the creation of encrypted health records, the patient can specify special emergency attributes and a BTG key. In case of an emergency, the health professional can access the BTG key and access the health records. The key can later be revoked. It is important to audit all access to the health records during the time of access of the BTG key.

- **SR12: Theft of Device-** In case of theft or loss of the device, there must be a provision to revoke old credentials and refurbish the health records on a patient's new mobile device.

- **SR13: Audit Logs-** The portable health record management system must record all events of reading and writing in the cloud along with a backup of the transaction for reference in case of improper access. Audit trails can provide proofs when there is a dispute such as abuse of permissions, illegal attempt to access a section, and the disclosure of patient's health information.

4.2 Threat Requirements

The health wallet comprises of an HCE-based card on the patient's mobile device. It follows standards based on ISO 8716 for communication using the APDU commands. Hence the framework must also protect the contactless health folder from the threats that affect smart cards, such as [161]:

- **TR1: Dos Attack-** An intruder can try to access the portable health folder on a contactless card with attempts to initiate an authentication that fails and makes it useless for a valid user. An intruder may try to access the health card to steal health data or insurance information for the health wallet and leave the wallet not usable at the time of need.

- **TR2: Replay Attack-** An intruder can try to replay some of the messages to access the S-MAPLE health folder. An intruder may try to replay and update new treatment plan on the health folder for financial gains from the insurance company. Even though decryption may fail, the intruder may use old data to update on the health folder. Hence, any replay of messages to the health wallet must be prevented using fresh nonces for each session.

- **TR3: Collusion Attack-** An intruder *I* can use another host *V's* information to access unauthorized information. For example, a pharmacist may try to look into the diagnosis of a patient due which the patient may lose privacy.

- **TR4: Parallel Session Attack-** An intruder can eavesdrop and gather messages and replay them to cause a parallel session attack.

- **TR5: Forgery Attack-** An intruder can use a registered stakeholder's identification and access the S-MAPLE health folder.

- **TR6: Platform Impersonation Attack-** A malicious server can replace the actual server for the TCA services.

- **TR7: MITM Attack-** An unregistered user can eavesdrop, spoof, decrypt, and relay a message.

- **TR8: Insider Attack-** An intruder, who is an insider, can impersonate the credentials of a user, such as a medical professional and try to seek health information of a patient during an OPD session. An insider may try to seek health information for celebrity and health records.

- **TR9: Relay Attack-** In a relay attack as discussed in Section 2.4.1.4, the attacker who is a proxy reader can masquerade as a valid reader by relaying the information received from the actual card to a proxy card over Bluetooth or remote access. The proxy card can further communicate it to the actual reader and similarly relay back the response to the actual card. There are two cases for a relay attack with the mobile-based health wallet [132]:

- *Case 1: Fraud Health Professional-* A malicious reader with a fraud health professional, can write a wrong prescription and medication to harm the patient, as shown in Figure 4.1. A patient interacts with a fraud doctor's malicious reader who relays the information to a valid doctor. The valid doctor in the remote location may be forced by an accomplice to give a wrong medication to harm the patient. In the other case, a valid doctor may move to a remote location, missing the duty, and have a junior represent the doctor to relay the information.
- *Case 2: Fraud Patient-* A health professional can interact with a fraud patient to seek treatment on behalf of a valid patient, as illustrated in Figure 4.2. The fraud patient may wish to seek costly treatment but cannot authenticate due to lack of credentials. The fraud patient can relay the information to a valid patient in a remote location. The valid patient without knowledge may authenticate with the actual health professional. After the false verification, the health professional may provide physical treatment or costly medication to the fraud patient.

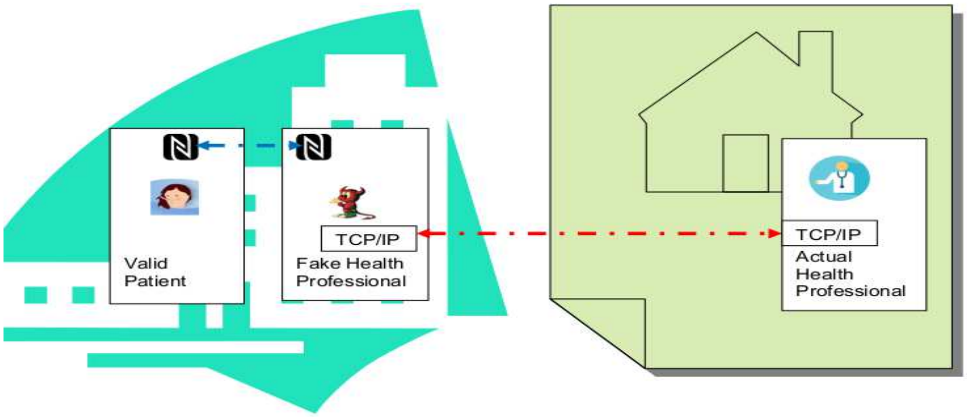


Figure 4.1: Relay Attack by a Fraud Health Professional

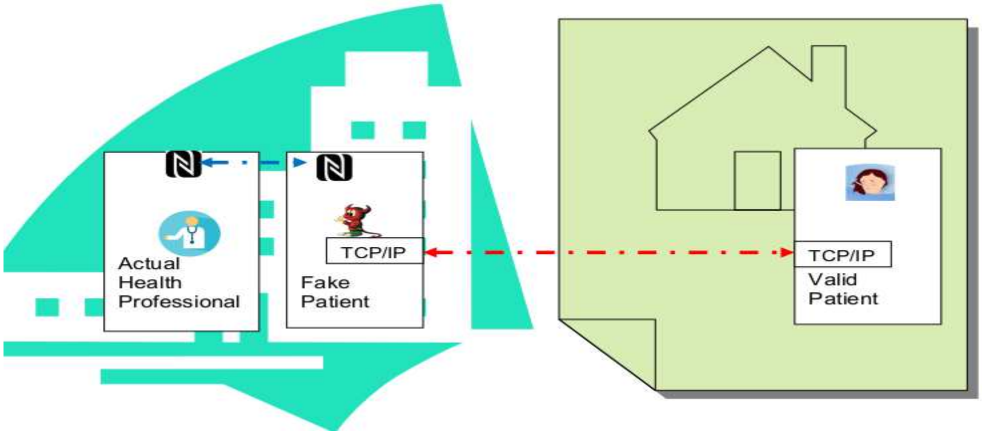


Figure 4.2: Relay Attack by a Fraud Patient

4.3 Proposed Solutions

This thesis proposes the following security solutions to fulfil the security and threat requirements:

1. **S1: Secure Element-** Secure Element (SE) provides tamper-resistant storage and secure computations. As discussed in Section 2.4.1.1, SE is a smart card chip, which connects internally with the NFC controller. This thesis looks into the form case of a secure microSD card such those from GoTrust [57] for both patient and mobile devices. SEs have Java card applets to assist secure storage and computations as discussed in Section 3.1.3. A secure authentication protocol over an HCE interface uses the SEs to perform secure computations.

2. **S2: CP-ABE-** CP-ABE provides fine-grained access control for sharing ciphertext with several users. The proposed health system encrypts all health records with a variation of Bethencourt et al.'s CP-ABE scheme [21], called *Scalable Proxy-based Immediate Revocation for CP-ABE* (SPIRC) proposed in this thesis. SPIRC provides scalable user revocation and satisfies all requirements *C1-C5* for sharing data from a portable device. A proxy server maintains a revocation list and assists in partial decryption. Whenever a user accesses the ciphertext and must decrypt it, the user contacts the proxy server to seek proxy components over HTTPS. The proxy server modifies the proxy components only for the malicious users

so that decryption fails and allows uninterrupted access to the other non-revoked users. It is collusion resistant, satisfies forward secrecy, and is Chosen Plaintext Attacks (CPA) secure. Chapter 6 presents the details of the SPIRC scheme followed by its security analysis in Section 7.3, and performance and comparison in Section 8.4.

3. **S3: Mutual Authentication and Attestation-** This thesis proposes a *NFC SE-based Mutual Authentication and Attestation (NSE-AA)* protocol. It provides proof-of-locality with NFC, end-to-end anonymous mutual authentication between SEs using limited lightweight symmetric encryption and also associates it with a remote attestation phase to ensure trustful states of the devices. This thesis presents a detailed security analysis with formal and informal security proof using the ROR model [2]. It is robust and has less computation and communication overheads as compared to the existing schemes. A simulation of the protocol on the AVISPA tool [14] proves that it is safe. Details are given in the chapter 5.

4. **S4: NFC-** It ensures that the devices that are interacting are in proximity. As discussed in Section 2.4.1.4, NFC makes it difficult to perform eavesdropping and MITM attack difficult. The other proposed solutions S2 and S3 further secure the NFC communication.

5. **S5: Secure Digital vault-** The HealthSecure Service provides a secure digital vault as discussed in Section 3.1.3 to backup the health records, store audit logs, and refurbish the records in case the mobile device is lost or stolen.

4.4 Scenarios for a Secure Health Wallet

The following scenarios, highlight the requirement for a security framework for a health wallet for the S-MAPLE health folder:

Scenario 1: Chronic health treatment- Nirmala suffers from chronic uveitis in her eyes and has been visiting several hospitals for treatment in India. The private hospitals offer treatment with less waiting time but at a higher cost. The root cause of her illness is not known for four years. She has been taking oral steroids and injections in the eye to prevent inflammation, which may cause cataract. She needs to maintain her health history whenever she visits a new health provider. She finally seeks treatment at a renowned public hospital in New Delhi, because the doctors are much experienced and she considers the treatment trustworthy. Her chronic ailment requires maintenance of health records and lab tests such as for Tuberculosis, Sarcoidosis, and Arthritis.

Nirmala wants to retain confidentiality (*SR1*) of the health records. There must be integrity of health records (*SR2*) since she visits various hospitals. Whenever she visits an OPD session, she wishes to see the right senior physician and nurse that administers the eye drops for examination (*SR3*). During one visit she was advised for an advanced CT scan for lungs, which indicated lymph nodes. She was advised to start the rigorous six months oral course for tuberculosis (TB). She wants to keep the privacy of her TB treatment due to the myth that it is incurable and the requirement for isolation. She continues her work along with treatment and wishes to retain privacy. She submits hospitals bills for her health insurance coverage at her work. She wants that the treatment is confidential. The lab bills must indicate the financial information and help retain her anonymity (*SR5*). She wishes that there are no third party attacks between the reader and her health wallet devices (*SR6*). She uses her mobile device along with the S-MAPLE health card application actively throughout the day. There must be secure storage that retains the wallet credentials that must be accessible only by the trusted health card application (*SR7*). She seeks the health wallet to visit her physician regularly every fortnight and has to keep track of her steroid medications (oral and eye drops) precisely. Regular visits to the pharmacist, her illness must be confidential. The pharmacist must tap to fetch the drugs required and must not get any other personal or health details such a chronic illness or treatment Nirmala may be reluctant to disclose. There must be provision for delegation of the health card's decryption key and health folder to a family member to collect reports or drugs on her behalf. However, after usage, there must be revocation of the keys (*SR9, SR10*). In case of an emergency, if she has hurt her eye, the emergency personnel must be able to access eye medications, and chronic illness so that they provide the treatment without causing any loss of eyesight (*SR11*). In case she loses her device, the TCA must revoke old credentials and restore new credentials and backup of the health folder on the new device (*SR12*). She uses the health wallet for the tap-based health services from different stakeholders of the HCE interface. Hence, her wallet must be protected from threats *T1-T9*.

Scenario 2: Emergency care- The health wallet comprises of the aggregated health history of a patient. It is encrypted using the SPIRC scheme. In case of an emergency, the Emergency personnel must validate and access the SPIRC decryption key with privileges to access the details of the health folder to provide the right treatment to the patient (*SR8, SR9, SR10, SR11*). The decryption can be later revoked. Since the health wallet retains the entire health history, which is readily available, the emergency personnel can provide timely treatment.

4.5 Summary

This chapter presents the security and threat requirements for the next generation smart portable health record management system. The health wallet is accessible using NFC, which ensures proof-of-locality and reduces risks for eavesdropping. However, NFC is prone to various security threats, and mobile applications must further secure it. The mobile devices are also vulnerable to various security threats. Hence, it is crucial to have a robust security framework for the proposed smart portable health management system. It is vital to maintain provenance of the health records on the health wallet so that they are considered reliable across different hospitals. The health wallet must allow selective access to various information on the health folder to retain the privacy of the patient. Since the health wallet is a contactless card, it must be secure from the threats for a smart card. The chapter presents the details for the relay attack scenario for a fraud patient as well as a fraud medical professional. Further, the chapter presents the security solutions with secure storage, an improved CP-ABE scheme known as SPIRC for selective RBAC and scalable revocation, a novel NFC-based mutual authentication and attestation protocol, NFC for proof of locality and a Secure Digital Vault to store a backup of records. Further, the chapter describes two use cases for chronic health management and emergency care which can benefit for a secure health wallet on a mobile device. In both cases, a secure health wallet with readily available health history can provide optimized healthcare with security, privacy and trust of the device as well as the patients.

CHAPTER 5

NFC-BASED MUTUAL AUTHENTICATION AND ATTESTATION

This chapter presents the details for a novel *NFC SE-based Authentication and Attestation (NSE-AA)* protocol for an end-to-end anonymous lightweight mutual authentication with limited use of symmetric encryption between two SEs and TPM-based attestation for security and trust over HCE. The protocol further secures the NFC communication and assures valid devices with trustful states interact with each other. The following sections discuss the details for the different phases of the NSE-AA protocol.

5.1 Motivation

NFC has an advantage that due to proximity, it is hard to perform eavesdropping and MITM attacks. However, it uses an untrusted communication channel and does not ensure the authenticity, authorization, and trustful state of devices, as discussed in Section 2.4.1.4.

Bidirectional communication is essential for mutual security handshake and trust between the devices. As described in Table 2.1, the HCE mode has several advantages over existing NFC modes. However, since HCE is software-based, it is vulnerable to threats. It requires mechanisms, such as cloud-based SE, internal SE, or TEE, to secure the interaction [159]. This thesis proposes the usage of an SE to secure an HCE card because the TEE is less secure and cloud-SE may not be 24/7 available.

HCE can provide a practical application of NFC with proof-of-locality for secure access to health records from the S-MAPLE folder on the patient's mobile device. HCE can provide bidirectional communication for mutual authentication and attestation. However, these issues are not addressed together in any NFC-based security scheme in the previous research papers [27, 140, 142, 55, 113, 71, 158] to the best of our knowledge.

The existing NFC authentication schemes [27, 140] have limitations for device anonymity , device attestation , and secure storage as discussed in Table 2.2. There is a requirement for a novel authentication protocol for associating mutual attestation for trust, device anonymity and secure storage.

More recently, researchers have also been looking into the matter of trust with remote attestation, such as access to Internet of Things (IoT) [142, 24]. As discussed in Section 2.6, malware can compromise the devices and make them victims to cause a cyber attack. This thesis looks into the mutual remote attestation of devices for trustful communication between them over NFC. As described in Table 2.3, hardware-based Trusted Platform Module (TPM) [142] is more secure than software-based TEE [24]. Hence, we look into the mechanisms for securing NFC communication with a new protocol to provide proof-of-locality, end-to-end anonymous mutual authentication between SEs, and an associated remote attestation for trust.

The attestation schemes previously proposed by Toegl and Hutter [142] and Aziz et al. [19] have limitations as discussed in Table 2.4. They do not provide user anonymity, and secure storage. They are also prone to DoS, parallel session, MITM, and insider attacks. Hence, there is a need to propose a new protocol over NFC, which can perform lightweight authentication as well as mutually attest both devices, as proposed in the scheme by Aziz et al. [19], over an HCE interface.

5.2 Details for NSE-AA Protocol

This section describes the details of the NSE protocol in this section for mutual authentication with mutual attestation. The objective of the NSE-AA protocol is to provide secure NFC-based communication between two devices. It considers generic communication between an IoT device and a user's mobile device. In this thesis, we consider that a patient's mobile-based S-MAPLE health folder is accessed securely by a medical professional's mobile device over an NFC interface. *The patient's mobile-device represents the user device with the HCE card, and the medical professional's mobile device represents the IoT device with the HCE reader application.* The chapter describes the NSE-AA protocol with a user device accessing an IoT device. Table 5.1 provides notations used for the NSE-AA protocol.

The protocol comprises of the following phases:

Table 5.1: Notations Used for NSE-AA Protocol

Symbol	**Meaning**
Pk_{TCA}	Public verification key of the Trusted CA
A	Adversary
H	Host (IoT device/Medical mobile device *D*/ User device/Patient mobile device *U*)
id_{AH}	Unique actual host identifier
pwd_{H}	Password for host H identity
Id_{VH}	Virtual identity/ Pseudonym for host *H*
N_{H}	Non-predictable nonce of host *H*
KDF	Key Generation function
K_{HS} (K_{US}/K_{DS})	Symmetric key between host and server
K_{UD}	Symmetric key between IoT and user devices
K_{S}	Symmetric session key generated in NSE-AA
id_{AIKH}	AIK identifier for the host
$Cert_{AIKH}$	Certificate AIK for host *H*
Pk_{AIKH}, Sk_{AIKH}	Public/Private AIKs for host *H*
$Cert_{EKH}$	Endorsement certificate for host *H*
S_{PCR}	Selection of PCR values to verify
I_{V}	Request to indicate if TPM version is required
$TPMinfo_{H}$	Version/revision information for host H's TPM
SML_{H}	Stored Measurement Log (SML) on TPM
BT-Addr	Bluetooth MAC Address
sign(M)K	Sign message *M* with private key *K*
E(K,M)	Encrypt message *M* with key *K*
D(K,M)	Decrypt message *M* with key *K*
h(M)	Hash over message *M*
IoT-SE<->User-SE	End-to-end communication between SEs

1. *Registration and Personalisation:* All devices personalise their identities and credentials on tamper-resistant storage.

2. *Mutual authentication:* Both mobile devices authenticate each other and generate a session key K_S for securing further communication.

3. *AIK Certificate generation:* Each device generates an Attestation Identity Key (AIK) certificate by using nonces and K_S in the previous phase.

4. *Attestation:* Each device attests its software and memory status using AIK private keys to the remote device.

5.2.1 Registration and Personalisation

During the personalization phase, each host H (IoT/User device) initializes its SE and TPM with credentials issued by TCA. Each host has a unique actual identity Id_{AH} and a secure symmetric key K_{HS} shared with TCA. Every user device and IoT device also retains a unique symmetric key K_{UD} shared between them. All credentials are stored on the SE of the devices. Each host H generates a password pwd_H and a random number b_H and uses them to generate pwb_H*:* $h\ (pwd_H \| b_H)$ and R_H*:* $h(Id_{AH} \| K_{HS})$. It stores id_{AH}, b_H and R_H on its local SE and TCA. These values are later used to generate a virtual identity id_{VH} as discussed in Section 5.2.2.1. The virtual identity is used in the mutual authentication phase to identify the host to TCA while retaining user anonymity. The Endorsement certificate $Cert_{EKH}$ for attestation credentials resides on the TPM. The Endorsement public key Pk_{EKH} uniquely identifies the host's TPM and presents it to TCA for identification and generation of an AIK certificate.

5.2.2 Mutual Authentication Phase

All valid user and IoT devices must securely identify and mutually authenticate each other to ensure that the registered devices can interact. The SEs compute all the cryptographic operations. Special internal applications on the devices access these computations from the applets on the SE and send them to the SE of the remote device over an HCE interface. Hence, the mutual authentication is end-to-end between the two SEs and does not allow a snooper to gain any valuable information. Table 5.2 describes the steps for the NSE-AA mutual authentication phase.

5.2.2.1 Virtual Identity Generation and Validation

Each device generates its virtual identity id_{VH} using a random nonce N_H. The generation of a virtual identity comprises of the following steps:

Table 5.2: NSE-AA Mutual Authentication

Message	**Description**
IoT<->User	1. M1: Select AID
User-SE:	2.Generate N_U, $T1: xor(N_U,K_{UD})$, Id_{VU}, $O: E(K_{US}, (N_U)), Q: h(N_U \Vert K_{UD})$
IoT-SE<-User-SE:	3. $M2: Id_{VU} \Vert T1 \Vert E(K_{UD},O) \Vert Q$
IoT-SE:	4. N'_U: xor(T1,K_{UD}); Decrypt to extract O; Verify N_U in Q; Generate N_D, Id_{VD}
IoT->Server:	5. $M3: Id_{VD} \Vert Id_{VU} \Vert E(K_{DS}, (N_D)) \Vert O$
Server:	6.Verify Id_{VD},Id_{VU}; Decrypt and extract N'_U, N'_D; Generate $T3: xor(pwb'_U,K_{DS})$, $T4: xor(pwb'_D,K_{US})$, $R: h(pwb'_U \Vert N'_U \Vert N'_D \Vert K_{US})$, $X: h(pwb'_D \Vert N'_U \Vert N'_D \Vert K_{DS})$
IoT<-Server:	7. $M4: T3 \Vert T4 \Vert X \Vert R$
IoT-SE:	8. $pwb'_U: xor(T3,K_{DS})$; Verify X; Generate K_S: $KDF(pwb'_U \Vert pwb_D \Vert N_U \Vert N_D \Vert K_{UD})$, $T2: xor(N_D,K_{UD})$, $Y: h(T2 \Vert R \Vert K_{UD})$
IoT-SE->User-SE:	9. $M5: T2 \Vert T4 \Vert Y$
User-SE:	10. $N'_D: xor(T2, K_{UD})$, $pwb'_D: xor(T4, K_{US})$, Verify nonces in Y and authenticate IoT-SE; Generate K_S: $KDF(pwb_U \Vert pwb'_D \Vert N_U \Vert N'_D \Vert K_{UD})$; $Z: h(N'_D \Vert K_{UD} \Vert K_S)$
IoT-SE<-User-SE:	11. $M6: Z$
IoT-SE:	12. Verify Z and authenticate *U-SE*
IoT<->User:	13. $E(K_S,Data)$

1. Before initiating access, a host H must enter password pwd'_H and the actual identity id'_{AH}.
2. It computes pwb'_H and R'_H and compares them as well as id'_{AH} with the stored values on the SE.
3. It generates id_{VH}: $<C_H, D_H, B_H>$, where

   $C_H: h(pwb_H \Vert N_H \Vert R'_H)$,

   $D_H: xor(h(Id_{AH} \Vert pwb_H),N_H)$,

   $B_H: E(K_{HS}, (Id_{AH} \Vert pwb_H \Vert N_H))$.

Virtual identity verification- It sends the virtual identity to TCA for verification, which comprises of:

- TCA decrypts B_H to extract id'_{AH}, pwb'_H and N'_H.
- TCA computes R'_H: $h(Id'_{AH}||K_{HS})$, N'_H: $xor(D_H,h(Id'_{AH}||pwb'_H))$ *and* C'_H: $h(pwb'_H||N'_H||R'_H)$.
- If N'_H and C'_H are same as the received values, then it validates the host H.

5.2.2.2 Detailed Steps for Mutual Authentication

Step 1: IoT-SE/Reader selects applet on the HCE card using AID.

Step 2-3: User-SE generates a random nonce N_U, parameter $T1$: $xor(N_U,K_{UD})$, virtual identity Id_{VU}: $<C_U,D_U,B_U>$, O: $E(K_{US}, (N_U))$, *and* Q: $h(N_U||K_{UD})$ to assure the integrity of N_U to the IoT device; and sends a message $M2$: $Id_{VU}||T1||E(K_{UD},O)||Q$ to the IoT device over the HCE interface.

Step 4-5: IoT-SE extracts N'_U: $xor(T1,K_{UD})$, decrypts to extract O; verifies nonce N'_U in Q; generates a random nonce N_D, and virtual identity Id_{VD}: $<C_D,D_D,B_D>$; and sends a message $M3$: $Id_{VD}||Id_{VU}||E(K_{DS},N_D)||O$ to the server over the HTTPS interface.

Step 6-7: The server decrypts and extracts N'_U and N'_D; verifies the virtual identities Id_{VU} and Id_{VD}, extracts pwb'_U and pwb'_D, validates the nonces N'_U and N'_D from the virtual identities; generates $T3$: $xor(pwb'_U,K_{DS})$, $T4$: $xor(pwb'_D,K_{US})$, R: $h(pwb'_D||N'_U||N'_D||K_{US})$ to assure the integrity of nonces sent to the User-SE; generates X: $h(pwb'_U||N'_U||N'_D||K_{DS})$ to assure the integrity of nonces sent to the IoT-SE; and sends a message $M4$: $T3||T4||X||R$ to the IoT over the HTTPS interface.

Step 8-9: IoT-SE extracts pwb'_U: $xor(T3,K_{DS})$; verifies nonce N_D and N_U in X; generates $T2$: $xor(N_D,K_{UD})$, Y: $h(T2||R||K_{UD})$ to authenticate itself to the user and session key:

K_S: $KDF(pwb'_U||pwb_D||N'_U||N_D||K_{UD})$;

and sends a message $M5$: $T2||T4||Y$ to the User over the HCE interface.

Step 10-11: User-SE extracts N'_D: $xor(T2,K_{UD})$, pwb'_D: $xor(T4, K_{US})$; validates N'_D by verifying Y; authenticates IoT-SE by validating N_U and K_{US} in Y; generates the session key:

K_S: $KDF(pwb_U||pwb'_D||N_U||N'_D||K_{UD})$

and also generate Z: $h(N'_D||K_{UD}||K_S)$ to authenticate itself to the IoT-SE; and sends a message $M6$: Z to the IoT over the HCE interface.

Step 12-13: IoT-SE verifies N_D in Z and authenticates User-SE. The session key K_S secures all further communication.

5.2.3 Mutual Attestation Phase

Both mobile devices must prove their trustful states. This thesis assumes that the devices have an inbuilt attester module for remote attestation. The Message Authentication Code (MAC) with session key K_S encrypts all attestation messages to maintain confidentiality and integrity. The following are the detailed phases:

AIK certificate generation protocol- Table 5.3 describes the steps for certificate generation for host H (IoT/user device) with a virtual identity id_{VH} and a random nonce N_H generated prior.

Table 5.3: NSE-AA AIK Certificate Generation

Message	**Description**
Host:	1. Get N_H; id_{AIKH}: $h(id_{VH} \Vert N_H)$; TPM Load Key (Sk_{AIKH})
Host:	2. SigTPM: TPMMakeIdentity: $sign(h(id_{AIKH} \Vert Pk_{TCA}))Sk_{EKH}$
Host->TCA:	3. $M7$: $E(K_{HS},(Pk_{AIKH} \Vert N_H \Vert SigTPM \Vert Pk_{EKH})) \Vert id_{VH}$
TCA:	4. Decrypt $M7$ using K_{HS}; Verify Pk_{EKH}, id_{VH}; Generate id'_{AIKH}: $h(id_{VH} \Vert N_H)$; Verify SigTPM;
TCA:	5. Generate $Cert_{AIKH}$: $sign(Pk_{AIKH}, id_{AIKH})Sk_{TCA}$;
TCA:	6. Generate K_C, $C1$: $E(K_{HS}, (K_C \Vert h(Pk_{AIKH}))$, $C2$: $E(K_C, Cert_{AIKH})$
Host<-TCA:	7. $M8$: $C1 \Vert C2$
Host:	8. Decrypt $C1$ to extract K'_C; Generate $h'(Pk_{AIKH})$; Verify it with $h(Pk_{AIKH})$; Decrypt $C2$ to extract $Cert_{AIKH}$.

Details of the steps are given below:

Step 1: Host H generates an AIK identity id_{AIKH}: $h(id_{VH} \Vert N_H)$ to prevent replay and collusion attacks.

Step 2: Host TPM verifies itself to TCA with its Endorsement Key (EK) credentials and generates an asymmetric attestation key pair (Pk_{AIKH}, Sk_{AIKH}) and $SigTPM$: $sign\ (h(id_{AIKH} \Vert Pk_{TCA}))Sk_{EKH}$.

Step 3: Host H sends a message $M7$: $E(K_{HS},(Pk_{AIKH} \Vert N_H \Vert SigTPM \Vert Pk_{EKH})) \Vert id_{VH}$ to TCA to generate an AIK certificate after validating its id_{VH} and Pk_{EKH}.

Step 4: TCA generates *id'_{AIKH}* to verify SigTPM.

Step 5: TCA generates an AIK certificate *$Cert_{AIKH}$*;

Step 6: TCA creates a session key K_C; *C1: $E(K_{HS},(K_C\|\ h(P_{AIKH}))$* and *C2: $E(K_C,Cert_{AIKH})$*;

Step 7: TCA sends a message *M8: $C1\|C2$* to the host *H*.

Step 8: Host *H* decrypts *C1* to extract key *K'_C*, generates the hash of *Pk_{AIKH}*, and compares it with the received hash. It uses *K'_C* to decrypt *C2* and extract and store *$Cert_{AIKH}$*.

This thesis improves the scheme by Aziz et al. [19] by reducing the computations of the certificate generation phase with symmetric encryption in messages *M7* and *M8*.

TPM-based Attestation Protocol: Table 5.4 describes the steps for attestation of an IoT device by a user device.

Table 5.4: NSE-AA Mutual Attestation

Message	Description
IoT<-User:	1. M9: N_U, I_V, S_{PCR} (Send challenge)
IoT:	2. Get SML_D; loadKey(Sk_{AIKD}); *SigTPM:TPM-Quote: sign(PCR_D, SML_D, N_U, TPMInfo)Sk_{AIKD}*; Q: SigTPM||TPMInfo||E(K_S, $Cert_{AIKD}$)
IoT->User:	3. *M10:Q, $MAC_{KS}(Q)$*
User:	4. Decrypt to extract Cer_{AIKD}; Verify $Cert_{AIKD}$ using Pk_{TCA}; Verify SigTPM using Pk_{AIKD}, N_U and TPMInfo; Generate *Id'_{AIKD}: $h(Id_{VD}\|N_D)$* and compare with Id_{AIKD} in $Cert_{AIKD}$

The TPM-based attestation scheme improves the integrity report protocol by Aziz et al. [19]. It comprises of a TPM challenge-response authentication and uses nonces and virtual identities exchanged in the NSE-AA mutual authentication phase to prevent a replay attack. Details of the steps are given below:

Step 1: User device sends nonce N_U as a challenge, I_V to request TPM version number, and S_{PCR} to select and specify the PCR values in the message *M9*.

Step 2: IoT device uses the nonce N_U received and its private AIK to generate a signed quote *$SigTPM_D$: $sign(PCR_D\|SML_D\|N_U\|TPMInfo_D)Sk_{AIKD}$* to preserve the secrecy of *PCR_D* and *SML_D*. It then generates *Q: $SigTPM\|TPMInfo\|E(K_S,Cert_{AIKD})$* and its message authentication code *$MAC_{KS}(Q)$* using the session key K_S.

Step 3: IoT device sends a message *M10:* $Q \,||MAC_{KS}(Q)$ to the user device.

Step 4: User device verifies *$MAC_{KS}(Q)$* and an AIK certificate with TCA's public key Pk_{TCA}. It further verifies the attestation data to ascertain the trustful state of the IoT device.

Similarly, the IoT device also remotely attests the user device. NSE-AA protocol also preserves the secrecy of the PCR and SML for attestation by encrypting $Cert_{AIK}$ with key K_S in parameter Q of message *M10*, which is sent unencrypted in the scheme by Aziz et al. [19].

5.3 Summary

The newly proposed NSE-AA protocol provides secure access to an IoT device such as the S-MAPLE health folder. It initiates an on-demand communication and control of IoT devices with secure tamper-resistant SE and TPM modules. NFC provides proof-of-locality. The protocol establishes an end-to-end security handshake and trust between the SEs of the two devices. HCE provides a feasible and open platform for developers for bidirectional communication as compared to the other NFC modes used in previous NFC-based security schemes. TPM-based attestation assures trust between the two devices and ensures that the devices are free from malware.

CHAPTER 6

SELECTIVE ACCESS WITH SCALABLE REVOCATION FOR PORTABLE DEVICES

This chapter presents the details for the *Scalable Proxy-based Immediate Revocation for CP-ABE (SPIRC)* scheme which extends the Bethencourt et al.'s scheme [21] for scalable revocation. It presents the construction for the user-based revocation, attribute-based revocation and delegation of the secret key.

6.1 Motivation

Bethencourt et al.'s CP-ABE scheme [21] can provide selective RBAC by representing a set of attributes for a specific role. Hence, CP-ABE can provide selective sharing of ciphertext. CP-ABE also supports revocation as well as collusion resistance. It is important to maintain confidentiality and allow selective sharing with authorized users since mobile devices are vulnerable to security and privacy threats. As discussed in Section 1.4.4, the portable S-MAPLE health folder must fulfil requirements for retaining and selectively sharing of data on portables device using Bethencourt et al.'s CP-ABE scheme [21].

Unlike direct schemes, indirect schemes do not require any prior knowledge of a revocation list and support broadcast of an intermediate key update, such that only non-revoked users can update their keys. Hence, they are suitable for portable devices to provide ease and flexibility to the owner. They also require a key update phase, which can provide bottleneck for interaction with Trusted Certified Authority (TCA). Table 1.4 describes the existing revocation schemes PIRATTE [76] and the M-PERMREV scheme [39], which satisfy requirements C1-C3. The Proxy-based Immediate Revocation of Attribute-based encryption (PIRATTE) scheme by Jahid et al. [76] is an indirect revocation scheme for CP-ABE, which satisfies all requirements except *C4* for scalability. Hence, there is a need to improve PIRATTE for scalable revocation for secure storage and sharing of critical data on a portable device. The SPIRC scheme extends the PIRATTE scheme by Jahid et al. [76] for scalable user revocation. It fulfils all revocation requirements *C1-C5* for sharing of secure data from portable devices.

6.2 Details for Scalable Proxy-based Immediate Revocation for CP-ABE (SPIRC) Scheme

The SPIRC scheme considers the assymmetric pairing for bilinear maps as discussed in Section 2.5.1.1.

Intuition- This thesis looks into the issue of storing data securely and sharing it with selective access control from a portable device. It outsources encryption to a trusted proxy server on a Trusted Certified Authority (TCA) as discussed in the system architecture in Section 3.1.3. Mobile devices which access the S-MAPLE health folder locally decrypt the ciphertext to view it selectively. The patient views it on his mobile device. The medical professional taps and accesses the selected data, decrypts and views and provides updates if required.

The TCA retains credentials and identities of registered users, as well as constants related to proxy data. The trusted proxy server helps in partial decryption and revocation. Each $user_i$ registers with a trusted proxy server and is associated with a set of random parameters $S_i = \{\lambda_i, \mathrm{a}_i, b_i\}$. The constants are associated with the decryption keys of the user as well as proxy data. For decryption, a user contacts the trusted server through a secure channel, such as HTTPS to gather proxy data to complete the decryption process. The trusted server also maintains a revocation list RL, which is populated by an authorized owner or administrative personnel to protect a portable device from malicious users from breach of trust or theft of the device. To revoke a user, the proxy server updates S_i so that the proxy data is modified and decryption fails.

The cloud-based service is contacted only for seeking proxy data and not for the actual ciphertext like in the cloud-based sharing applications [90]. The trusted proxy server must comply with all requirements suggested by the *Trusted Computing Group (TCG)*. The trust between authorized users and proxy server can be established through mutual authentication and remote attestation over TLS Aziz et al. [19]. Remote attestation can ensure that they are not compromised with any malicious software. Further, authorized users can commute with the trusted server using separate

CP-ABE access policies for RBAC for allowing trusted revocation and credential configuration by users with administrative roles. Verification of trustful states of devices ensures secure maintenance of credentials as well as revocation list on the proxy server. A detailed design of the trusted proxy server is beyond the scope of this thesis.

6.2.1 Construction of SPIRC User-based Revocation

The SPIRC scheme supports scalable user revocation without requiring re-encryption or re-distribution of keys. This thesis modifies the PIRATTE scheme by Jahid et al. [76] for scalable revocation for infinite users. It comprises of the following algorithms:

- **Setup**: Generates Public key *PK* and Master key *MK*.
- **Encrypt (PK,M,τ)**: Takes data *M*, Public key *PK*, and access policy τ to generate the ciphertext *CT*.
- **KeyGen (MK,S)**: Takes master key *MK* and set of attributes *S* and generates the secret key *SK*.
- **Proxy-Data ($\mathbf{U_k}$,RL)**: Takes user identity $\mathrm{u_k}$ and the revocation list *RL* as input and generates the proxy data *PXD*. The proxy server also invokes CONVERT function to transform portion of the ciphertext C'_x for each attribute x satisfied by users $\mathrm{u_k}$ and generates the converted portion C''_x.
- **Decrypt (CT,SK)**: Decrypts the ciphertext CT to plaintext M if the set of attributes S in SK satisfy the policy τ that is used to generate ciphertext CT.

The details of different phases are given below:

Setup: The trusted proxy server chooses G_1, G_2, g_1, g_2 and random elements α and $\beta \epsilon Z_p$ to generate a public key *PK* and a master key *MK*.

$$PK = G_1, G_1, g_1, g_2, h = {g_1}^{\beta}, e(g_1, g_2)^{\alpha} \tag{6.1}$$

$$MK = \beta, {g_2}^{\alpha} \tag{6.2}$$

Unlike PIRATTE, for master key *MK*, there is no generation of polynomial *P* of degree *t+1*, where t is the number of users that can be revoked. Hence, it provides scalable revocation.

Encrypt (PK,M,τ): The Encryption algorithm is similar as in PIRATTE equation 2.3

KeyGen (MK,S): It generates the Secret key *SK* for $user_i$ for a set of attributes *S*. For each user, it chooses a random numbers *r* along with set $S_i = (\lambda_i, a_i, b_i) \epsilon Z_p$ and for each attribute *j* it chooses random number $r_j \epsilon Z_p$.

$$SK = (D = g_2{}^{(\alpha + \mathrm{r})/\beta}$$

$$\forall j \epsilon S : D_{\mathrm{j}} = g_2{}^{\mathrm{r}} H(j)^{\mathrm{r_j}(\lambda_{\mathrm{i}} \mathrm{a_i} + \mathrm{b_i})} = g_2{}^{\mathrm{r} + \mathrm{h_j r_j}(\lambda_{\mathrm{i}} \mathrm{a_i} + \mathrm{b_i})},$$

$$D'_{\mathrm{j}} = g_1{}^{\mathrm{r_j}},$$

$$D''_{\mathrm{j}} = (D'_{\mathrm{j}})^{\mathrm{a_i}} = g_1{}^{\mathrm{r_j a_i}}) \tag{6.3}$$

The parts of the secret key *SK,* D_j, and D'_j for each attribute *j* contain random number r_j and *D* contains random number *r*, which is specific to a user. Hence, attributes from different users cannot be combined together and it prevents collusion.

Proxy-Data (User$_{\mathbf{i}}$): Proxy server maintains a random set S_i for each user along with a revocation list. For successful decryption, $user_i$ seeks proxy data *PXD* from the proxy server, which is unique to a user. The user also sends C'_x to the proxy server to return Convert C''_x.

$$PXD = \lambda_{\mathrm{i}} \tag{6.4}$$

$$CONVERT(C''_{\mathrm{x}}, b_{\mathrm{i}}) = (C'_{\mathrm{x}})^{\mathrm{b_i}} = g^{\mathrm{h_x q_x(0) b_i}} \tag{6.5}$$

The user secret *SK* is blinded by $(\lambda_i a_i + b_i)$ and needs C''_x along with C_x and C'_x. Proxy can revoke the user by updating the λ_i or b_i for $user_i$ in *PXD* and C''_x.

Decrypt: For a $user_i$, each leaf node *x* of the policy is an attribute, with $j = attr(x)$, if $j \epsilon S$, (*S* is a set of attributes) the $DecrytpNode = A_j$ is as follows:

$$A_{\mathrm{j}} = \frac{e(C_{\mathrm{x}}, D_{\mathrm{j}})}{e(D''_{\mathrm{j}}, C'_{\mathrm{x}})^{\lambda_{\mathrm{i}}} e(D'_{\mathrm{j}}, C''_{\mathrm{x}})}$$

$$e(C_{\mathrm{x}}, D_{\mathrm{j}}) = e(g_1{}^{\mathrm{q_x}(0)}, g_2{}^{\mathrm{r + h_j r_j}(\lambda_{\mathrm{i}} \mathrm{a_i} + \mathrm{b_i}})$$

$$= e(g_1, g_2)^{\mathrm{q_x}(0)\mathrm{r} + \mathrm{q_x}(0)\mathrm{h_j r_j}(\lambda_{\mathrm{i}} \mathrm{a_i} + \mathrm{b_i})} \tag{6.6}$$

$$\begin{aligned} A_{\mathrm{j}} &= \frac{e(g_1, g_2)^{\mathrm{q_x(0)r+q_x(0)h_jr_j(\lambda_ia_i + b_i)}}}{e({g_1}^{\mathrm{r_ja_j}}, {g_2}^{\mathrm{h_jq_x(0)}})^{\lambda_{\mathrm{i}}} e({g_1}^{\mathrm{r_j}}, {g_2}^{\mathrm{h_jq_x(0)b_i}})} \\ &= \frac{e(g_1, g_2)^{\mathrm{q_x(0)r+q_x(0)h_jr_j(\lambda_ia_i + b_i)}}}{e(g_1, g_2)^{\mathrm{r_ja_ih_jq_x(0)\lambda_i}} e(g_1, g_2)^{\mathrm{r_jh_jq_x(0)b_i}}} \\ &= \frac{e(g_1, g_2)^{\mathrm{q_x(0)r+q_x(0)h_jr_j(\lambda_ia_i + b_i)}}}{e(g_1, g_2)^{\mathrm{r_ja_ih_jq_x(0)\lambda_i + r_jh_jq_x(0)b_i}}} \\ &= \frac{e(g_1, g_2)^{\mathrm{q_x(0)r+q_x(0)h_jr_j(\lambda_ia_i + b_i)}}}{e(g_1, g_2)^{\mathrm{r_jh_jq_x(0)(\lambda_ia_i + b_i)}}} \\ &= e(g_1, g_2)^{\mathrm{q_x(0)r}} \end{aligned} \tag{6.7}$$

Each $user_i$ has associated constant values λ_i*,* a_i*, and* b_i, which are maintained on the proxy server. Whenever revocation is required, the proxy server updates λ_i or b_i, which are part of the *PXD* and *C”*$_x$, and cause the DecryptNode function to fail and return ⊥. Rest of the decryption process is the same as in [21, 76] to obtain the original message M.

6.2.2 Construction of SPIRC Attribute-based Revocation

The trusted proxy server can have mechanisms to revoke few attributes from a user rather than revoking the user, such revoking one of the numerous roles of a user. In attribute revocation, the proxy server associates each attribute j of $user_i$ with constant random parameters λ_{ij}, a_{ij}, and b_{ij} . A $user_i$ seeks these values from the proxy server at the time of decryption. The list of attributes revoked for each $user_i$ is maintained by the proxy server and the updated values are provided to the $user_i$ each time the user performs decryption. The other different phases are given below:

The **Setup** scheme is similar as for SPIRC's user revocation in equation 6.2.

The **Encrypt** scheme is similar as for the PIRATTE scheme in equation 2.3.

KeyGen (MK, S): The proxy server generates a secret key SK related to a set of attributes S for $user_i$. It provides each $user_i$ and attribute j with random number r and the set s_{ij} = (λ_{ij}, a_{ij} and b_{ij}) ϵ Z_p. The terms $\lambda_{ij}a_{ij} + b_{ij}$ and a_{ij} are incorporated in D_j and D''_j components of the secret key respectively. The secret key for the $user_i$ over a set of attributes S is given by:

$$SK = (D = g_2^{(\alpha+r)/\beta} \quad ,\forall\, j \,\epsilon S$$
$$D_j = g_2^r H(j)^{r_j(\lambda_{ij}a_{ij}+b_{ij})} = g_2^{r+h_j r_j(\lambda_{ij}a_{ij}+b_{ij})},$$
$$D'_j = g_1^{r_j},$$
$$D''_j = (D'_j)^{a_{ij}} = g_1^{r_j a_{ij}}) \tag{6.8}$$

The parts of secret key SK D_j, D$'_j$, and D$''_j$ are for each attribute j of a specific user $user_i$. Hence, attributes from different users cannot be combined together and prevents collusion.

Proxy-Data (user$_i$): The proxy server maintains a set s_{ij} with random constants for each $user_i$ and attribute j. The $user_i$ seeks proxy data from the server for decrypting the ciphertext.

The $user_i$ sends C$'_y$ to the trusted server for which the trusted server returns C$''_y$ and PXD$_{\text{ij}}$ to the $user_i$.

$$PXD_{\text{ij}} = \lambda_{ij} \tag{6.9}$$
$$Convert(C''_{\text{y}}, b_{\text{ij}}) = (C'_{\text{y}})^{\text{b}_{\text{ij}}}$$
$$= g^{h_y q_y(0) b_{ij}} \tag{6.10}$$

$User_i$ secret key SK is blinded by $(\lambda_{ij}a_{ij} + b_{ij})$ and requires C$''_y$ along with C$_y$ and C$'_y$. The attribute j for $user_i$ can be revoked by the proxy by updating the user specific λ_{ij} or b_{ij}. The decryption in such cases is not possible as the values returned to the user by the trusted server are not be able to cancel the terms in the user's secret key. Hence, the proxy server revokes the user successfully.

Decrypt: For each leaf node x of the policy, j = attr(x) $,\forall j \,\epsilon\, S$, where S is a set of attributes of SK of a $user_i$. Then, the DecryptNode A_j is given by:

$$A_j = \frac{e(C_x, D_j)}{e(D''_j, C'_x)^{\lambda_{ij}}\; e(D'_j, C''_x)}$$
$$e(C_x, D_j) = e(g_1^{q_x(0)}, g_2^{r+h_j r_j(\lambda_{ij}a_{ij}+b_{ij})})$$
$$= e(g_1, g_2)^{q_x(0)r + q_x(0)h_j r_j(\lambda_{ij}a_{ij}+b_{ij})} \tag{6.11}$$

$$\begin{aligned} A_j &= \frac{e(g_1, g_2)^{qx(0)r + qx(0)h_j r_j(\lambda_{ij} a_{ij} + b_{ij})}}{e(g_1^{r_j a_{ij}}, g_2^{h_x qx(0)})^{\lambda_{ij}}\, e(g_1^{r_j}, g_2^{h_x qx(0) b_{ij}})} \\ &= \frac{e(g_1, g_2)^{qx(0)r + qx(0)h_j r_j(\lambda_{ij} a_{ij} + b_{ij})}}{e(g_1, g_2)^{r_j a_{ij} h_j qx(0) \lambda_{ij}}\, e(g_1, g_2)^{r_j h_j qx(0) b_{ij}}} \\ &= \frac{e(g_1, g_2)^{qx(0)r + qx(0)h_j r_j(\lambda_{ij} a_{ij} + b_{ij})}}{e(g_1, g_2)^{r_j a_{ij} h_j qx(0) \lambda_{ij} + r_j h_j qx(0) b_{ij}}} \\ &= \frac{e(g_1, g_2)^{qx(0)r + qx(0)h_j r_j(\lambda_{ij} a_{ij} + b_{ij})}}{e(g_1, g_2)^{r_j h_j qx(0)(\lambda_{ij} a_{ij} + b_{ij})}} \\ &= e(g_1, g_2)^{qx(0)r} \end{aligned} \tag{6.12}$$

The information gathered by the user from the proxy server is used as a denominator in DecryptNode function. Values of λ_{ij} and C''_x are provided to the $user_i$ with j attributes by the server. If an attribute has not been revoked, then the server sends the exact values of attributes that are generated at the time of the secret key generation. The server maintains λ_{ij}, a_{ij}, and b_{ij} random values for each $user_i$. In case server needs to revoke an attribute from the user, it can update λ_{ij} or b_{ij}. The $user_i$ cannot decrypt as numerator pairing has D_j component. The D_j component of the secret key consists of $(\lambda_{ij} a_{ij} + b_{ij})$, which is not cancelled if the denominator term has different value. The DecryptNode function fails and it returns $\perp$. Therefore, the proxy server facilitates the user by providing selective revocation of attributes while maintaining other non-revoked attributes.

6.2.3 SPIRC Delegation

This section presents the details for the delegation support on the SPIRC scheme. Delegation renders authority and power to a trusted entity temporarily. TheSPIRC scheme allows the user to issue a delegated secret key to another user for some or all of its attributes. It supports two different modes for delegation:

- *Single authority:* A single authority is in charge for issuing keys to all the interested parties.
- *Multilevel authority:* A multilevel hierarchy is present for key allotment, that is a user A can issue a key to user B who can further delegate the key to user C.

6.2.3.1 Single Authority Delegation

A $user_i$ can delegate a set $\tilde{S} \subseteq$ S, where S is a set of attributes associated with the key of $user_i$ obtained from the authority. The delegation key $\tilde{SK}$ for $user_k$ for a set of attributes $\tilde{S}$ is generated. The authority generates a set $s_i = \lambda_i, a_i, b_i$ corresponding to each $user_i$. If the server revokes $user_i$, then the delegated $user_k$ will automatically be revoked. The proxy server chooses random numbers $\tilde{r}$, and $\tilde{r}_j \epsilon Z_p \ \forall j \epsilon \tilde{S}$ for a user $user_i$. Single authority delegation makes use of parameter f = $g_2^{\frac{1}{\beta}}$ from the public key PK. The user's secret key SK is as in Equation 6.3. The algorithm generates the delegated key $\tilde{SK}$ for $user_k$ with a set of attributes $\tilde{S}$, and is given by:

$$
\begin{aligned}
&\tilde{SK} = (\, \tilde{D}, \forall j \epsilon \tilde{S}, \tilde{D}_j, D'_j, \tilde{D''}_j)\\
&\tilde{D} = (D) f^{\tilde{r}} = g_2^{\frac{\alpha+r}{\beta}} g_2^{\frac{\tilde{r}}{\beta}} = g_2^{\frac{\alpha+r+\tilde{r}}{\beta}}\\
&\tilde{D}_j = (D_j)\, g_2^{\tilde{r}}\, h(j)^{\tilde{r}_j} = (g_2^{r+h_j r_j(\lambda_i a_i+b_i)}) g_2^{\tilde{r}}\, g_2^{h_j\, \tilde{r}_j} = g_2^{r+h_j r_j(\lambda_i a_i+b_i)+\tilde{r}+h_j \tilde{r}_j}\\
&\tilde{D''}_j = (D''_j) g_1^{\tilde{r}_j/\lambda_i} = (g_1^{r_j a_i})(g_1^{\tilde{r}_j/\lambda_i}) = g_1^{r_j a_i+\tilde{r}_j/\lambda_i}
\end{aligned}
\tag{6.13}
$$

Decryption for delegated user is given as follows:

DecryptNode (CT, $\tilde{SK}$, x): For each leaf node x of the, policy j = attr(x) , $\forall\, j \epsilon \tilde{S}$ for a $user_k$. Then, the DecryptNode A_j is given by:

$$
\begin{aligned}
&A_j = \frac{e(C_x, \tilde{D}_j)}{e(\tilde{D}''_j, C'_x)^{\lambda_i}\, e(D'_j, C''_x)}\\
&e(C_x, \tilde{D}_j) = e(g_1^{qx(0)}, g_2^{r+\tilde{r}+h_j r_j(\lambda_i a_i+b_i)+h_j \tilde{r}_j})\\
&\ = e(g_1, g_2)^{qx(0)r+qx(0)\tilde{r}+qx(0)h_j r_j(\lambda_i a_i+b_i)+qx(0)h_j \tilde{r}_j}\\
&A_j = \frac{e(g_1, g_2)^{qx(0)r+qx(0)\tilde{r}+qx(0)h_j r_j(\lambda_i a_i+b_i)+qx(0)h_j \tilde{r}_j}}{e(g_1^{r_j a_i+\tilde{r}_j/\lambda_i}, g_2^{h_j qx(0)})^{\lambda_i}\, e(g_1^{r_j}, g_2^{h_j qx(0) b_i})}\\
&\ = \frac{e(g_1, g_2)^{qx(0)r+qx(0)\tilde{r}+qx(0)h_j r_j(\lambda_i a_i+b_i)+qx(0)h_j \tilde{r}_j}}{e(g_1, g_2)^{r_j a_i h_j qx(0)\lambda_i+\tilde{r}_j h_j qx(0)}\ e(g_1, g_2)^{r_j h_j qx(0) b_i}}\\
&\ = \frac{e(g_1, g_2)^{qx(0)(r+\tilde{r})+qx(0)h_j r_j(\lambda_i a_i+b_i)+qx(0)h_j \tilde{r}_j}}{e(g_1, g_2)^{qx(0)h_j r_j(\lambda_i a_i+b_i)+qx(0)h_j \tilde{r}_j}}\\
&\ = e(g_1, g_2)^{qx(0)(r+\tilde{r})}
\end{aligned}
\tag{6.14}
$$

else the DecryptNode function returns $\perp$.

The message M is retrieved by the following procedure:

$$\frac{\tilde{C}}{\frac{e(C,D)}{A}} = Me(g_1,g_2)^{\alpha s}\frac{e(g_1,g_2)^{(r+\tilde{r})s}}{e(g_1,g_2)^{\alpha s+(r+\tilde{r})s}} = M$$

If the server revokes user $user_i$, then the delegated $user_k$ is automatically revoked.

6.2.3.2 Multilevel Authority Delegation

In multilevel authority delegation, a user A generates key for user B, and user B further generates the key for user C. User A maintains identity for user B and corresponding set $s_{AB} = \lambda_{AB}, a_{AB}, b_{AB}$ where AB subscript denotes that user A is authority of user B, which delegates keys to user B for decryption. Authority user A will generate key for user B and incorporate the set s_{AB} into the secret key SK of user B. Similarly, user B maintains identity for user C and its corresponding set $s_{BC} = \lambda_{BC}, a_{BC}, b_{BC}$ and delegates it to user C.

KeyGen (MK,S): For a set of attributes S, user A generates a secret key SK for user B. The secret incorporates elements from set s_{AB}. The key generation is given by:

$$SK = (D = g_2^{(\alpha+r)}/\beta, \forall j \epsilon S$$

$$D_j = g_2^r H(j)^{r_j(\lambda_{AB}a_{AB}+b_{AB})} = g_2^{r+h_j r_j(\lambda_{AB}a_{AB}+b_{AB})},$$

$$D'_j = g_1^{r_j},$$

$$D''_j = (D'_j)^{a_{AB}} = g_1^{r_j a_{AB}}) \tag{6.15}$$

User B can generate secret key $\tilde{SK}$ for user C by delegating some set of attributes $\tilde{S} \subseteq$ S to user C using the generated set $s_{BC} = \lambda_{BC}, a_{BC}, b_{BC}$ and secret key obtained from user A. Delegation key is defined as:

$\tilde{SK} = (D, \forall j \epsilon \tilde{S} : (D_j, D'_j, \tilde{D''}_j, \tilde{D'''}_j))$

$$D'_j = g_1^{r_j}$$

$$D''_j = g_1^{r_j a_{AB}}$$

$$\tilde{D}''_j = (D''_j)^{\frac{1}{\lambda_{BC} a_{BC} + b_{BC}}}$$

$$= g_1^{\frac{r_j a_{AB}}{\lambda_{BC} a_{BC} + b_{BC}}}$$

$$\tilde{D}'''_j = (D''_j)^{\frac{a_{BC}}{\lambda_{BC} a_{BC} + b_{BC}}}$$

$$= g_1^{\frac{r_j a_{AB} a_{BC}}{\lambda_{BC} a_{BC} + b_{BC}}} \tag{6.16}$$

$\forall\, j \,\epsilon\, \tilde{S}$, the components D_j, D'_j,$\tilde{D''}_j$, $\tilde{D'''}_j$ are included in the delegated key $\tilde{SK}$. The delegated key possessed by user C contains values from both S_{AB} and S_{BC}. Thus, the delegated key $\tilde{SK}$ will require proxy portions from both user A and user B. The term $f = g_2^{\frac{1}{\beta}}$ is not used in multilevel authority delegation.

Decryption: Decryption for user C is similar to basic decryption in the SPIRC scheme. Decryption on the user C, in addition to having the delegated key $\tilde{SK}$ from user B, will have some new pairings in the DecryptNode function.

DecryptNode(CT,$\tilde{SK}$, x):

$$A_j = \frac{e(C_x, D_j)}{e(\tilde{D''}_j, C''_{x(BC)})^{\lambda_{AB}}\; e(\tilde{D'''}_j, C'_x)^{\lambda_{AB}\lambda_{BC}}\; e(D'_j, C''_{x(AB)})} \tag{6.17}$$

Numerator of A$_j$:

$$A_j = e(g_1^{q_x(0)}, g_2^{r + h_j r_j(\lambda_{AB} a_{AB} + b_{AB})}) \quad = e(g_1, g_2)^{r q_x(0) + q_x(0) h_j r_j(\lambda_{AB} a_{AB} + b_{AB})} \tag{6.18}$$

Denominator of A$_j$:

$$A_j = e(g_1, g_2)^X e(g_1, g_2)^Y e(g_1, g_2)^Z \tag{6.19}$$

where X,Y,Z stands for :

$$e(g_1,g_2)^X = e(g_1,g_2)^{\dfrac{r_j a_{AB} h_j q_x(0) b_{BC} \lambda_{AB}}{\lambda_{BC} a_{BC} + b_{BC}}}$$

$$e(g_1,g_2)^Y = e(g_1,g_2)^{\dfrac{h_j r_j a_{AB} a_{BC} q_x(0) \lambda_{AB} \lambda_{BC}}{\lambda_{BC} a_{BC} + b_{BC}}}$$

$$e(g_1,g_2)^Z = e(g_1,g_2)^{r_j h_j q(0) b_{AB}}$$

$$\begin{aligned} A_j &= \frac{e(g_1,g_2)^{rq_x(0)+q_x(0)h_j r_j(\lambda_{AB} a_{AB}+b_{AB})}}{e(g_1,g_2)^{r_j h_j q_x(0)(\lambda_{AB} a_{AB}+b_{AB})}} \\ &= e(g_1,g_2)^{q_x(0)r} \end{aligned} \tag{6.20}$$

The denominator term contains proxy portions from both users A and B, which are used in pairing. If user B is revoked by user A by updating the values of λ_{AB} or b_{AB}, then user C also gets revoked. User C can also be revoked by user B by updating the values of λ_{BC} or b_{BC} and sending these updated proxy portions $C''_{x(BC)}$ thus, preventing user C from decrypting successfully.

Thus, we can conclude that user C will not be able to decrypt if either user B or user C is revoked as the proxy data of both users A and B are part of decryption process of user C.

6.3 SPIRC for S-MAPLE Health Folder

The Mobile-based health folder retains different health records, such as prescriptions, reports and medication details from various hospitals in standard formats, such as HL7. It organizes records from each department into different sections. Various authorized health professionals access them as per their roles with selective RBAC as described previously in Table 3.2. Table 6.1 describes the main notations for the case study of the S-MAPLE health folder.

6.3.1 Role of Proxy Server

The Proxy server maintains a revocation list. According to the status of a user as revoked or non-revoked, it sends the proxy components to a user such that decryption will either fail or pass. Figure 6.1 illustrates the access to a portable device by another mobile device. The static device shares a symmetric secret key K_{PS} with the proxy server for exchange of challenge and response to prevent replay attacks. Each time a user (reader)accesses the static device (health folder), it sends a challenge, which is a nonce N_S encrypted with K_{PS} along with the ciphertext. The user contacts the proxy server with the challenge and requests for the proxy components. The proxy server sends the proxy components and the response N_S*+1* encrypted with K_{PS}. If the user is a valid user, the decryption is successful, and the user can send information along with the response to the static device. The static device accepts the new information after verification of the response and prevents a user from using any old proxy components to decrypt and try to contact the static device by replaying the invalid update.

Table 6.1: Notations Used in the SPIRC Scheme for S-MAPLE Health Folder

Term	**Description**
H/H’	Unencrypted/Encrypted health folder
U	User (P-Patient/H-Health professional)
KDRUabe	User’s CP-ABE Read decryption key
KDWUabe	User’s CP-ABE Write decryption key
Section_i	Health folder’s ith section, i = 1-7
ri	Random number for Section_i
rei	Encrypted ri for ith section: E (KEWabe, ri)
RW={re1..re6}	Write policy encrypted random numbers
Update_i	Update for Section_i
K_S	Symmetric Session key from prior mutual authentication phase

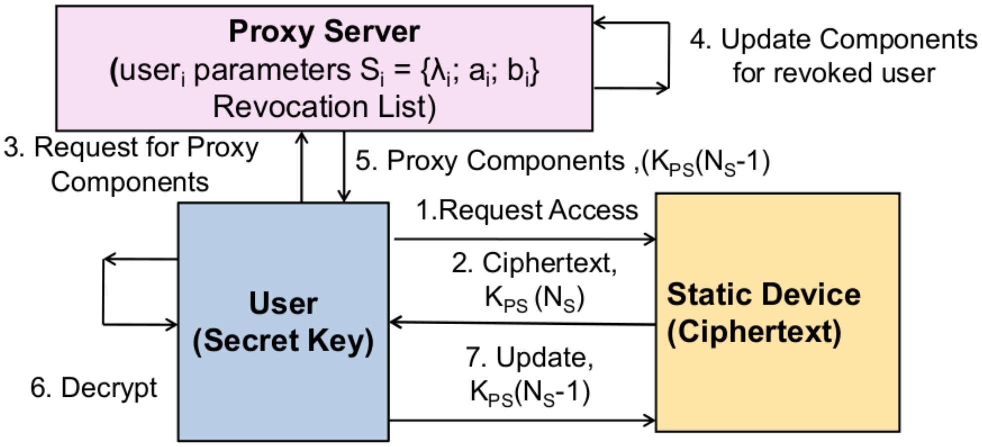

Proxy Server
(user$_i$ parameters $S_i = \{\lambda_i; a_i; b_i\}$
Revocation List)
4. Update Components for revoked user
3. Request for Proxy Components
5. Proxy Components ,($K_{PS}(N_S-1)$
1.Request Access
2. Ciphertext, $K_{PS}(N_S)$
7. Update, $K_{PS}(N_S-1)$
User (Secret Key)
Static Device (Ciphertext)
6. Decrypt

Figure 6.1: Selective Access of a Portableproxyservblock Device using Indirect Revocation

6.3.2 Read Access Policy

For each section, a read access policy encrypts it, and a write access policy encrypts a section specific random number *ri* as *rei*. Table 3.2 described health folder with different departments and sections. Each section has a read access policy and a write access policy. A stakeholder stores two decryption keys: a read decryption key *KDRUabe* and a write decryption key *KDWUabe* to access the authorized sections. A CP-ABE decryption key can decrypt all sections for which the attributes in the key can satisfy the section access policy.

Stakeholders first reads the health folder and obtains the concerned sections by decrypting with their read decryption key *KDRUabe*. However, once they can read a section, they must be able to update it only if they have access according to the write access policy.

Figure 6.2 shows a sample read access policy *ACRALL*, which permits all stakeholders to read.

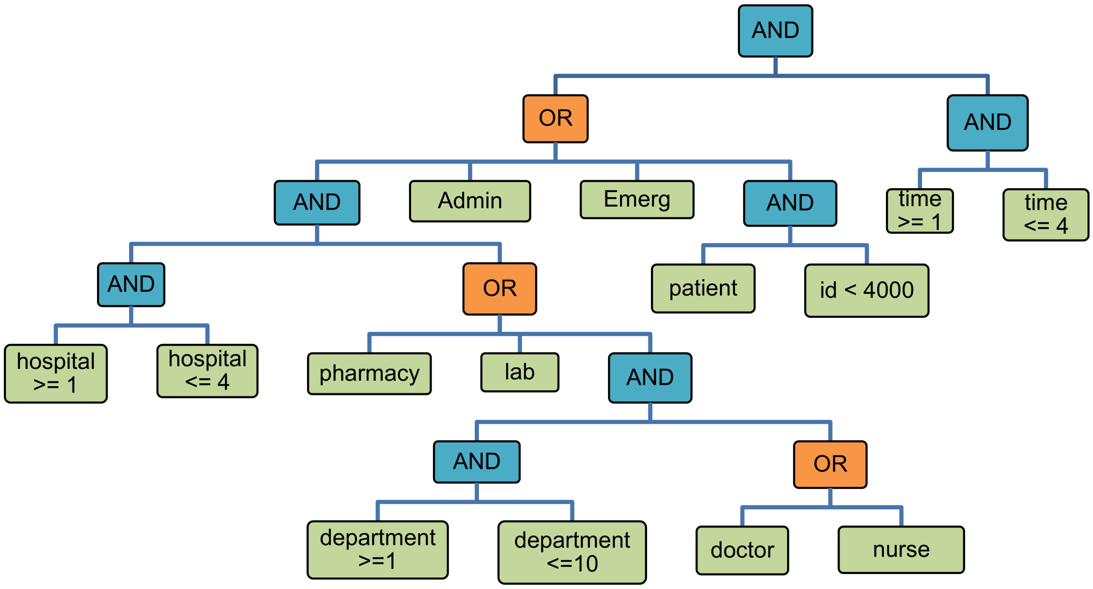
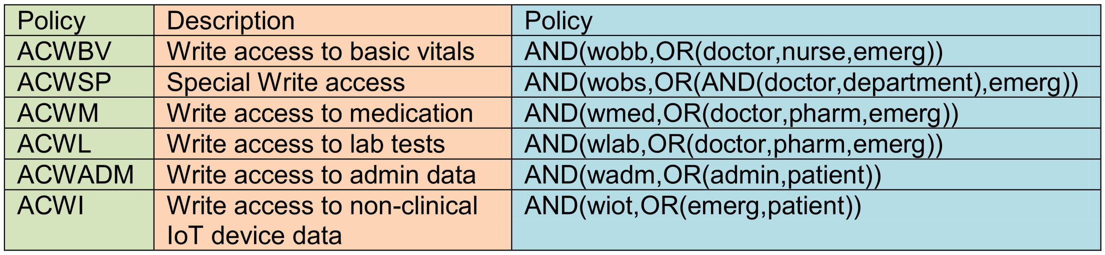


Figure 6.2: CP-ABE Read Access Policy: ACRALL

Each section has a different write access policy with a special set of associated attributes as shown in Figure 6.3. For example, to read sections encrypted with the *ACRALL* read policy, a pharmacist must have a read decryption key with attributes that satisfy the access policy. Similarly, to write to sections encrypted with the *ACWM* write policy, the write decryption key must have attributes, which satisfy the access policy. The attributes *pharmacy*, *time=2* and *wmed* must be present in the decryption key to satisfy these policies.

6.3.3 Write Access Policy

For each section *i* of the health folder, random number *ri* is encrypted with the write CP-ABE policy of the section as *rei*. When stakeholders requests to write to Section *i*, patient challenges it with the encrypted *rei* for the section. If stakeholders have access to write, they can decrypt *rei* using their write decryption key *KDWUabe*. In response, the stakeholder's computes *ri' = ri + 1* and sends it to the Mobile-based health folder along with the update $Update_i$ for the section. The Mobile-based Health folder compares the received *ri'* and the locally computed value of *(ri+1)*. If they match, then the $Update_i$ is written on the health folder, else it is rejected.



Policy	Description	Policy
ACWBV	Write access to basic vitals	AND(wobb,OR(doctor,nurse,emerg))
ACWSP	Special Write access	AND(wobs,OR(AND(doctor,department),emerg))
ACWM	Write access to medication	AND(wmed,OR(doctor,pharm,emerg))
ACWL	Write access to lab tests	AND(wlab,OR(doctor,pharm,emerg))
ACWADM	Write access to admin data	AND(wadm,OR(admin,patient))
ACWI	Write access to non-clinical IoT device data	AND(wiot,OR(emerg,patient))

Figure 6.3: Health folder CP-ABE Write Access Policies

6.3.4 Revocation

Healthsecure service associates time-based attributes with the decryption key, and each stakeholder must renew it periodically. The *ACRALL* policy in Figure 6.2 shows the time-based attributes. Decryption keys with time attributes between *1* and *4* will only satisfy this policy. In case they fail to do so, the time attribute does not match with the time range specified in the policy, and decryption fails. However, during a valid key time, the proxy server must be able to directly revoke a user using the SPIRC scheme and provide fine-grained access control. The revocation can be user or attribute based. The patient must be able to delegate a portion of his key and revoke it at a later stage. Emergency personnel can seek the BTG key for CP-ABE after validation and access the health folder of a patient who is unconscious. The proxy server can later revoke the emergency keys.

Table 6.2: Sequence for SPIRC-based Selective Access and Scalable User-based Revocation

S.No	Message	Description
1'.	Card:	Personalisation: NSE-AA parameters, (KDRPabe,KDWPabe, RW=(re1..re7))
1".	Reader:	Personalisation: NSE-AA parameters, (Non-emergency:KDRMabe, KDWMabe)
2.	Card ⟷ Reader:	NSE-AA Mutual Authentication to generate K_S
3.	Card ← Reader:	Action: write/read, Section$_i$
4.	Card:	MP1=H' || rei
5.	Card → Reader:	E (K_S,MP1)
6.	Reader ⟷ Server:	If emergency personnel obtains BTG keys (KDRMabe, KDWMabe)
7.	Reader ⟷ Server:	Proxy server-based decryption H = D (KDRMabe, H'), ri= D (KDWMabe,rei)
8.	Server:	Revoke users in RL
9.	Server:	ri' = ri+1, Access = hash (Update$_i$), MM1= ri' || Update$_i$
10.	Card ← Reader:	E (K_S,MM1)
11.	Server:	If ri' == ri+1 then accept Update$_i$
12.	Card →: Server	Update$_i$ through HTTPS
13.	Server:	Revoke key if user is an emergency personnel Sync Update$_i$ on digital vault, re-encrypt H as H"
14.	Card ← Server:	H" through HTTPS

6.3.5 Sequence Flow for Read and Write Access

Table 6.2 shows the sequence diagram for the selective access of the S-MAPLE health folder with the SPIRC scheme and related user-based revocation.

The patient and health professionals personalise their device with credentials and identities on SE. After the HCE tap, they mutually authenticate each other and set a secure session key K_S. The reader device requests to read or write to a $Section_i$. The card device sends the encrypted $Section_i$ along with a challenge *rei*. In case of an emergency, the emergency professional obtains the BTG CP-ABE decryption keys *(KDRMabe, KDWMabe)* from the Healthsecure service. The health professional uses the read and write decryption keys to read and write to $Section_i$. After the session terminates, the patient's mobile device sends the update for data sync to the digital vault. The HealthSecure system also re-encrypts the health folder with the new $Update_i$. After the session, the proxy server revokes the *BTG* CP-ABE decryption key for emergency professional the SPIRC scheme.

6.4 Summary

Portable devices, such as mobile devices can retain critical data encrypted with CP-ABE for fine-grained selective access control. This thesis proposes a novel SPIRC scheme, which improves the PIRATTE scheme by Jahid et al. [75] for scalable revocation. It satisfies all the revocation requirements *C1-C5* for ease of maintenance of ciphertext on a portable device.

The protocol proposed in this thesis is the first novel attempt to address secure data on a portable device using Bethencourt et al.'s CP-ABE scheme [21] with scalable user revocation. SPIRC can provide multi-user selective access to IoT devices such as users with different roles in a family access a car with their mobile devices to lock/unlock, configuration setup and access logs using selective RBAC.

The details for the SPIRC protocol are published in conference 1; journal 1 in the Published Work.

CHAPTER 7

SECURITY ANALYSIS

In this chapter, we present a detailed security analysis for the proposed solutions and show how they satisfy and fulfill requirements for security and threat as discussed in Chapter 4. This chapter presents the detailed security analysis for the newly proposed NFC SE-based Mutual Authentication and Attestation (NSE-AA) and Scalable Proxy-based Immediate Revocation for CPABE (SPIRC) protocols.

7.1 Security Analysis for the Proposed Security Solutions

The security solutions fulfill the requirements for security and threat as discussed below:

1. **S1: Secure Element-** The SEs are tamper-resistant and store applets for cryptographic credentials and secure computations for the NSE-AA protocol. The applets can be accessed internally through mobile applications compiled with the SE manufacturer's library. The credentials encrypt the health folder and store it on the insecure region on the mobile device.

2. **S2: CP-ABE-** The proposed SPIRC protocol encrypts all health records and stores them on the insecure region. The SEs on the device store all SPIRC credentials. SPIRC provides fine-grained access control for sharing ciphertext with several users. It provides selective access with scalable revocation without the requirement of re-encryption and re-distribution of the key to the non-revoked users. It is collusion resistant, satisfies forward secrecy, and is Chosen Plaintext Attacks (CPA) secure. Section 7.3 presents the details of the security analysis for the SPIRC scheme.

3. **S3: Mutual Authentication and Attestation-** The proposed NSE-AA protocol provides proof-of-locality with NFC, end-to-end anonymous mutual authentication and attestations between the SEs of the two devices. It helps to overcome the limitations of security for NFC. Section 7.2 presents a detailed security analysis with a formal and informal security proof.

4. **S4: NFC-** It ensures that the devices that are interacting are in proximity with proof-of-locality and makes it difficult to performs eavesdropping and MITM attacks.

5. **S5: Secure Digital vault-** All health records for the S-MAPLE health folder are digitally synced on the Secure Digital Vault on the HealthSecure service as discussed in Section 3.1.3 to backup the health records and refurbish the records in case the mobile device is lost or stolen. The HealthSecure service also stores the audit logs so that it can trace the activities and health professionals involved.

Tables 7.1 and 7.2 discuss how the security solutions S1-S5 fulfill the security and threat requirements respectively. The details of the security and threats requirements analysis with the proposed protocols is presented in the following sections. The fulfillment of security and threat requirements in the framework help cover the challenges for NFC security *NS1-NS10* as discussed in Section 2.4.1.4.

Table 7.1: Fulfillment of Security Rquirements

Solution	**S1:SE**	**S2:CP-ABE**	**S3:Mutual Authen. and Attes.**	**S4:NFC**	**S5:Digital Vault**
SR1:Confidentiality	Y	Y	Y	Y	Y
SR2:Integrity	Y	Y	Y	Y	Y
SR3:Mutual Auth and Trust	Y	-	Y	-	-
SR4:Privacy	Y	Y	Y	-	Y
SR5:User Anonymity	Y	-	Y	-	Y
SR6:Proof-of-locality	-	-	Y	Y	-
SR7:Secure Storage	Y	-	-	-	Y
SR8:Selective Access	-	Y	-	-	-
SR9:Revocation	-	Y	-	-	-
SR10:Delegation	-	Y	-	-	-
SR11:Emergency	Y	Y	-	-	-
SR12:Theft of device	Y	Y	-	-	Y
SR13:Audit Logs	Y	Y	-	-	Y

Table 7.2: Fulfillment of Threat Requirements

Solution	**S1:SE**	**S2:CP-ABE**	**S3:Mutual Authen. and Attest.**	**S4:NFC**	**S5:Digital Vault**
TR1:Dos	Y	-	Y	-	-
TR2:Replay	Y	-	Y	-	-
TR3:Collusion	Y	Y	Y	-	-
TR4:Parallel session	Y	Y	Y	-	-
TR5:Forgery	Y	-	Y	-	-
TR6:Platform impers.	Y	-	Y	-	-
TR7:Man-in-the-middle	Y	-	Y	-	-
TR8:Insider	Y	-	Y	-	-
TR9:Relay	Y	-	Y	-	-

7.2 Security Analysis for NSE-AA Protocol

7.2.1 Informal Security Analysis for NSE-AA

The NSE-AA protocol satisfies the following security requirements identified in Sections 4.1 and 4.2 :

- **SR1: Confidentiality-** All communication between devices is encrypted using the session key generated in the NSE-AA protocol. Also, the devices store encrypted data with cryptographic credentials on the tamper-resistant SEs. Hence, the protocol retains the confidentiality of data on the devices as well as during communication.
- **SR2: Integrity-** The devices encrypt the data as well as send its MAC using the session key in the NSE-AA protocol to ensure the integrity of data.
- **SR3: Mutual Authentication and Attestation-** The NSE-AA protocol provides a lightweight anonymous mutual authentication along with attestation for secure identification and for establishing trust over HCE. The mutual authentication phase improves the authentication scheme proposed by Thammarat et al. [140] with device anonymity using symmetric encryption, random nonces, and passwords known only to the device owners. NSE-AA protocol also ensures the integrity of data by using hash functions. It is also secure based on the Real-Or-Random (ROR) model [2], as discussed in Section 7.2.2. The mutual attestation phase

improves the scheme proposed by Aziz et al. [19] work with secure storage on SE and TPM, an end-to-end lightweight anonymous mutual authentication using symmetric encryption between the SEs, a lightweight AIK certificate generation, secrecy of PCR and SML and proof-of-locality using NFC.

- **SR4: Privacy-** Encryption of data with credentials stored on secure tamper-resistant SEs prevents an intruder from seeking any critical user data.
- **SR5: User Anonymity-** A host H sends its virtual identity id_{VH}:$(C_H \| D_H \| B_H)$ to TCA for verification. The virtual identity is generated using a password known only to the host H and random nonce N_H along with other parameters stored on the SE of the devices. Hence, it is difficult for an adversary to find the actual identities of the devices.
- **SR6: Proof-of-locality-** NFC ensures proof-of-locality for a secure IoT access. The device proximity for NFC makes MITM attack and eavesdropping difficult.
- **SR7: Secure Storage-** The tamper-resistant SE and TPM provide secure storage and cryptographic computations for an end-to-end NSE-AA protocol.

The framework also prevents the following attacks:

- **TR1: Denial of Service (DoS) attack-** An attacker cannot pass through the verification of the tamper-resistant SE and hence prevents a DoS attack.
- **TR2: Replay attack-** The unique nonces used in the NSE-AA protocol prevents an intruder to resend messages and hence prevents a replay attack.
- **TR3: Collusion attack-** For a collusion attack, a host H provides another host V's attestation data: Vs SML and PCR information signed by V's AIK private key Sk_{AIKV} and V's AIK certificate $Cert_{AIKV}$. However, it has previously shared id_{VH} and N_H with the remote host in the mutual authentication phase. The remote device calculates id_{AIKH}:$h(id_{VH} \| N_H)$ and finds it different from the id_{AIKV} in $Cert_{AIKV}$ and hence the attack fails.

- **TR4: Parallel session attack-** Each session uses tamper-resistant SE for cryptographic computations and secure storage and also uses unique nonces to prevent replay for parallel sessions.
- **TR5: Forgery attack-** It is difficult to forge because credentials reside on tamper-resistant SE and TPM. Also, the components of the virtual identity: C_H, D_H, and B_H require pwb_H, id_{AH}, and secret symmetric key K_{HS}, which are known only to the user.
- **TR6: Platform impersonation attack-** Virtual identity and messages exchanged in the communication comprises of symmetric keys shared between devices and TCA or the private component of the asymmetric keys of TCA and AIK. Hence, it is difficult for a malicious server to replace a valid TCA server. The cryptographic credentials are stored on the SEs of the devices and known only to a valid server.
- **TR7: MITM attack-** Both devices register with TCA. Also, the tamper-resistant SE stores unique identities and certificates and executes cryptographic computations for end-to-end security handshake in the NSE-AA protocol. Hence, any unregistered user cannot eavesdrop, spoof, decrypt, and relay messages.
- **TR8: Insider attack impersonation of the device-** End-to-end mutual authentication between the two SEs and unique keys for each user prevents any adversary to capture another user's messages and impersonate it.
- **TR9: Relay Attack-** A fraud patient or medical professional may relay an interaction with the health wallet of a valid patient to a remote location and cause a breach of trust as illustrated in Figures 4.1, 4.2. The NSE-AA protocol ensures that there is an end-to-end mutual authentication between the SEs of the two devices over an HCE interface. Further, even attestation can be prone to relay attacks as suggested in the PROXIMITEE scheme proposed by Dhar et al. [36]. The scheme comprises of an external embedded device to ensure device proximity using distance bounding and secure boot-time initialization. However, distance bounding countermeasure cannot be used with HCE, because HCE is software-based and

HCE card performance is based on the mobile processor [144]. Hence, we look into location-based countermeasures to prevent a relay attack, such as presented in our earlier proposed protocol for mutual authentication known as *HCE with Asymmetric Mutual Authentication (HAMA)* [132]. HAMA uses asymmetric encryption for end-to-end mutual authentication between the SEs. Each device stores signed locations by a trusted server on the SE. The devices exchange signed locations with each other along with other parameters in the mutual authentication. Each device verifies if the signed location of the remote device matches with its actual device location. If the locations do not match, it means there is a relay attack and the mutual authentication fails. Both HAMA and NSE-AA address end-to-end mutual authentication between the SEs over HCE. The latter, however, is costly to implement due to the usage of asymmetric encryption. In the future the NSE-AA protocol can be extended using the location-based countermeasures for relay attack as presented in the HAMA protocol.

7.2.2 Formal Security Proof using ROR Model

This sections proves the security of the session key K_S generated in the mutual authentication phase of the NSE-AAA protocol using the Real-Or-Random (ROR) model [2]. The protocol comprises of entities user U, IoT device D, and TCA.

1. **ROR Model:**

   - **Participants-** Let I_U^{t1}, I_D^{t2}, *and* I_{TCA}^{t3} be the instances *t1, t2*, and *t3* of *U, D*, and *TCA*. The instances are called oracles.
   - **Accepted State** An instance I^t moves to an Accepted state after receiving the final protocol message. The ordered concatenation of all communication messages sent and received by I^t is called the session identification (*sid*) of I^t for the current session.
   - **Partnering** Instances I^{t1} and I^{t2} are called partners if: 1) both are in the Accepted state, 2) both mutually authenticate each other and share the same *sid*, and 3) both are mutual partners of each other.

- **Freshness** If the session key K_S between U and D is not revealed through a reveal query as defined below, I_U^{t1} and I_D^{t2} are considered as fresh.
- **Adversary** An adversary A is modelled using the Dolev-Yao (DY) model [38] and can eavesdrop, modify, delete, or inject the messages transmitted between the entities involved during the communication with help of the following queries:

  Execute(I^{t1}, I^{t2}): The query models the eavesdropping attack and allows an adversary to eavesdrop the messages communicated among *U, D*, and *TCA*.

  Send(I^t, msg): The query models an adversary to transmit a message *msg* to I^t and receives a response from it. It is further modelled as an active attack.

  Reveal(I^t): The query reveals the session key K_S between I^t and its partner to the adversary in the current session.

  CMD(I_U^{t1}): Under this query, an adversary can fetch all the sensitive secret credentials stored from the lost or stolen mobile device's memory and can represent an active attack.

  Test(I^t): At the beginning of a game, an unbiased coin c is flipped. The adversary executes this test query and if *c:1* and the session key K_S, between U and D is fresh, I^t returns K_S else if *c:0* it returns a random number else it returns a null value.

  This thesis assumes that the adversary can access only a limited number of *CMD(I_U^{t1})* queries, while an unlimited number of Test(I^t) queries are accessible.

2. **Security Proof:** This thesis proves that the NSE-AA protocol is safe.

Theorem 1: For an adversary running in polynomial time t against the NSE-AA protocol in the random oracle model, and D a uniformly distributed password dictionary, if no device has been compromised by the adversary then:

$$Adv_{NSE-AA}^{ake} <= \frac{q_h^2}{|Hash|} + \frac{2\,qsend}{|D|}$$

where q_h: number of Hash queries, $qsend$: number of Send queries, $|Hash|$: range of the hash function, and $|D|$: size of D.

Proof: This thesis follows the proof as presented by Change and Le [28]. Let *Gi*, where *i:[0,3]* be the sequence of games and $Succ_i$ be an event when an adversary succeeds in guessing the bit b in the game *Gi*.

Game G0: The game *G0* presents a real attack by an adversary against the NSE-AA protocol. The bit *b* is chosen at random at the beginning of this game, and hence we have:

$$Adv^{ake}_{NSE-AA} = 2Pr[Succ_0] - 1 \tag{7.1}$$

Game G1: This game simulates an adversary's eavesdropping attacks with the *Execute* query. In the end, the adversary sends the *Test* query, whose output decides if the adversary obtains a real session key K_S or a random number. The session key is K_S *: KDF(*$pwb_U||pwb_D||N_U||N_D||K_{UD}$*)*, where pwb_U*:h(*pwd_U*,*b_U*)* and pwb_D*:h(*pwd_D*,*b_D*)*. An adversary cannot compute K_S unless it has access to the SEs, nonces and the device passwords. The SEs are tamper resistant and store the secret key K_{UD} and the random numbers b_U and b_D. It is difficult to obtain the nonces, which are exchanged using *xor* with the symmetric key K_{UD} in the terms *T1* and *T2* from messages *M2* and *M5* respectively. The passwords of the devices are known only to the device owners. Hence, it is difficult to generate the terms pwb_U and pwb_D. It is difficult to obtain these terms by an adversary because the components *T3* and *T4* of message *M4* hide them with the xor operations with the symmetric keys K_{DS} and K_{US}. It is difficult to obtain the keys from tamper-resistant SEs of the IoT and the user devices. Hence, the chances of winning for an adversary do not increase when the adversary eavesdrops.

$$Pr[Succ_0] = Pr[Succ_1]. \tag{7.2}$$

Game G2: The game *G2* modifies the game *G1* by addition of the *Send* and the *Hash* queries to model an active attack. In this attack, an adversary tries to deceive a participant to accept an illegal message. The adversary sends repeated *Hash* queries to find collisions. The *Send* query does not cause any hash collisions because each of the messages is associated with the identity of a participant or a unique session symmetric key between the entities (which are generated based on the actual identities). As per the birthday paradox, we have:

$$|Pr[Succ_1]-Pr[Succ_2]| \leq \frac{q_h^2}{2|Hash|} \tag{7.3}$$

Game G3: This game *G3* modifies the game *G2* by addition of the *CorruptSC* queries to simulate the loss of a smart card or SEs (lost attack). Even though the details of the SE are known, it is important to know the user and IoT device passwords pwd_U and pwd_D to find the session key K_S. The adversary can try to guess the password using the online dictionary attacks becaue the password has a low-entropy. However, if the system limits the number of wrong password guesses, then:

$$|Pr[Succ_2]-Pr[Succ_3]| \leq \frac{q_{\text{send}}}{|D|} \tag{7.4}$$

After the last game *G3*, all queries are simulated. If the adversary has failed to break the security of the NSE-AA protocol, it then finally tries to win the game through the *Test()* query to guess the bit *b*. Hence, *Pr[Succ$_3$]* is the same as the probability of guessing the bit *b*.

$$Pr[Succ_3] \leq \frac{1}{2} \tag{7.5}$$

From equations 7.1 - 7.5, we have:

$$Adv_{NSE-AA}^{ake} <= \frac{q_h^2}{|Hash|} + \frac{2qsend}{|D|}$$

Hence, the NSE-AA protocol is secure when the range of the hash function and the size of the password dictionary are large. An adversary may query *CorruptSC* oracle to simulate a stolen SE and simulate the smart card breach attacks. Still, the adversary cannot generate the session key K_S due to the requirement of the user's password. Hence, the adversary can compromise the NSE-AA protocol only through an online guessing attack. However, since the authentication system normally limits the number of failed logins, which is much smaller than |D|, the online password guessing attack is not possible.

7.2.3 Simulation For Formal Verification Using AVISPA

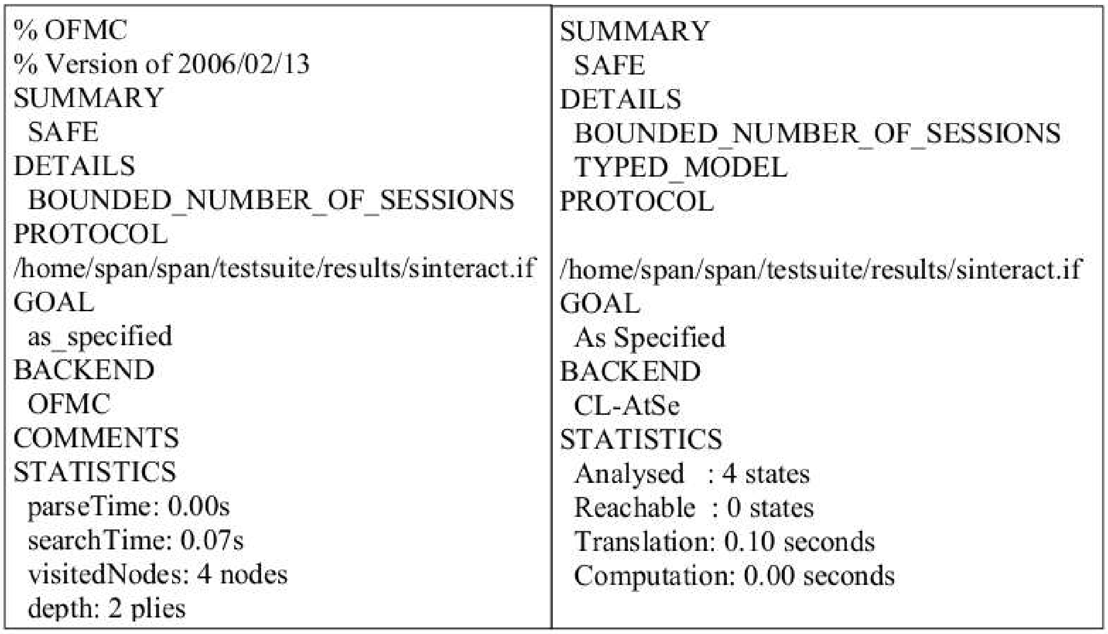
The NSE-AA protocol has been simulated using the AVISPA [14] verification tool as discussed in Section 2.6.2.1. An AVISPA script for NSE-AA is presented in the Annexure. The script provides roles of agents *U, D*, and *S* for the user device, IoT device, and TCA respectively along with definitions for session and environment. The script follows all phases of the NSE-AA protocol as given in Tables 5.2 and 5.4. An intruder (i) and the communication channels are based on the Dolev Yao model [38]. The first phase consists of mutual authentication as per the steps in Table 5.2. The role *U* generates a virtual identity Id_{VU} and a random N_U and sends a challenge to role *D*. Role *D* further generates a virtual identity Id_{VD} and a random nonce N_D, and forwards the challenge to *S*. Role *S* verifies the virtual identities and sends tickets for roles *D* and *U*, which helps them to mutually authenticate each other and generate a session key K_S to encrypt further communication. The generation of all cryptographic operations is on the SE of the devices. The sceript then exchanges messages for the mutual attestation phase as per the steps in Table 5.4. Both devices exchange the PCR and SML logs and validates each others. The script is executed using all the backends of AVISPA. The widely-accepted OFMC and CL-AtSe backends specify if, for a replay attack protection, valid users can execute the security protocol in the presence of a passive intruder. They also check if there is any possibility of an MITM attack by an adversary for the Dolev-Yao model. The protocol script assures the following security goals:

- *secrecy_of sec_ks:* Secret Ks between device U and D.
- *secrecy_of sec_ua:* Actual identity IAu is known to the device U and TCA.
- *secrecy_of sec_da:* Actual identity IdAd is known to the device D and TCA.
- *authentication_on auth_1*: Authentication of role D by role U through parameter Y: h (T2||R||KUD), T2: xor (ND,KUD), R:h (pwbU'||NU'||ND'||KUS).
- *authentication_on auth_2*: Authentication of role U by role D through parameter Z: h (ND'||KUD||KS).
- *secrecy_of sec_smlu*: Secrecy of SML logs of role U between U and D.

- *secrecy_of sec_smld*: Secrecy of SML logs of role D between U and D.
- *secrecy_of sec_pcru*: Secrecy of PCR of role U between U and D.
- *secrecy_of sec_pcrd*: Secrecy of PCR of role D between U and D.
- *authentication_on smlu_verify*: Verify h(SMLU) to attest user.
- *authentication_on smld_verify*: Verify h(SMLD) to attest IoT device.

7.2.3.1 Analysis of Simulation Output

In this work we have successfully simulated and proved the safety of the NSE-AA protocol with the AVISPA tool. Figure 7.1 shows results for the OFMC and CL-AtSe backends to illustrate that the NSE-AA protocol is secure against MITM and replay attacks.



```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
/home/span/span/testsuite/results/sinteract.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 0.07s
 visitedNodes: 4 nodes
 depth: 2 plies
```

```
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL
PROTOCOL

/home/span/span/testsuite/results/sinteract.if
GOAL
 As Specified
BACKEND
 CL-AtSe
STATISTICS
 Analysed   : 4 states
 Reachable  : 0 states
 Translation: 0.10 seconds
 Computation: 0.00 seconds
```

Figure 7.1: AVISPA Output

The output of the SATMC and TA4SP backends are INCONCLUSIVE because they cannot verify the NSE-AA protocol, which uses algebraic properties of XOR. The random nonces for mutual authentication are unique for each session, which prevents a replay attack. The session key K_S: *KeyGen(PwbU.PwbD.Nu.Nd.Kud)* requires knowledge of the secret K_{UD}, nonces and user's virtual identity, which requires the password known only to the user. Moreover, since an intruder cannot replay messages, it cannot generate a session key. The proximity over NFC as well as difficulty for an adversary to guess the session key makes MITM attacks difficult. The protocol prevents the MITM attacks because an intruder does not know the secrets between the devices as well as the session key, which encrypts all further messages including those for attestation. Both devices share their PCR and SML signed with their respective AIKs. The remote device verifies AIK certificate, PCR, and SML values and *IdAikH* for the host *H*.

7.3 Security Analysis for SPIRC Scheme

The definitions for user-based revocation are as per Jahid et al. [76].

7.3.1 Informal Security Analysis for SPIRC

This section presents security analysis for the SPIRC scheme for selective RBAC for S-MAPLE health folder. It satisfies the following requirements, which have been identified earlier in Section 4.1.

- **SR1: Confidentiality-** The S-MAPLE health folder is encrypted by SPIRC and assures selective access to only authorized health professionals to assure confidentiality. SPIRC supports forward secrecy so that on revocation a revoked user cannot access the health folder with his credentials.
- **SR2: Integrity-** The encryption of S-MAPLE with the SPIRC protocol helps to retain the integrity of the health records. An intruder cannot update or replace them. Even if the device is lost, the scalable proxy-based revocation can prevent an intruder from decrypting and accessing the data.
- **SR3: Privacy-** A health professional can access selected health records of the health folder and hence, unwanted information is not disclosed to a health professional.
- **SR8: Selective access-** Authorized stakeholders access various sections through selective RBAC. Each health professional has a separate CP-ABE decryption key to read and write to

different sections and can access them only if the CP-ABE attributes associated with the key that satisfies the corresponding access policy.

- **SR9: Revocation-** The SPIRC scheme satisfies all revocation requirements for portable ciphertext *C1-C5* and provides flexibility to retain a secure S-MAPLE health folder on the patient's mobile device. If an adversary $user_j$ finds proxy data of another $user_i$, then it will not help the adversary with the decryption because each user has a different set of random constants maintained on the proxy server. There are however overheads of maintaining a constant set s_i for each $user_i$ on the proxy server. With scalable revocation, a patient can share health records across various hospitals and hence get mobility.

  The SPIRC scheme can also use attribute-based revocation to revoke few rights from a medical professional. A medical professional can have multiple roles based on the attributes. The SPIRC scheme allows revocation of a specific role, by revoking related attributes.

- **SR10: Delegation-** A patient can delegate a portion of his decryption key to a family member or a friend to collect a lab report, which can be given to him later and can be synced with the health folder. The delegated key can be revoked by the patient through scalable revocation support in the SPIRC scheme. A senior physician can delegate key to a junior doctor to share the load of patients in a crowded public hospital, such as in the developing countries.

- **SR11: Emergency BTG Key-** An emergency person authenticates with the S-MAPLE health folder and gets temporary CP-ABE decryption keys to read and write from the HealthSecure service to provide emergency care. Later the proxy server revokes the emergency keys.

- **SR12: Theft of device-** On the loss or theft of a registered device, the proxy server revokes the old credentials. Hence, an adversary cannot use the old mobile device. It issues new credentials along with the copy of the re-encrypted health folder on the patient's new mobile device. Hence, it allows portability of secure health records by directly sharing with trusted stakeholders.

- **SR13: Audit Logs-** A new health record written to the health folder is data synced on the digital vault on the HealthSecure service. Every event on the health folder is stored on the audit log. The audit logs are stored in an encrypted form using the SPIRC scheme so that they are secure, even on partially insecure servers.

7.3.2 Security Game

In the security game between an adversary and a challenger, the encryption remains secure even when the adversary compromises the proxy and obtains its key after a recent revocation.

Setup- A challenger runs the SETUP and provides public parameters PK to the adversary. Challenger also generates a proxy data PXD.

Phase 1- The adversary performs repeated queries for KEYGEN to obtain keys for multiple users u_1,... u_{q1} with different sets of attributes S_1,...,S_{q1}. The adversary also contacts the proxy server for the CONVERT({C'_1,... C'_r},u_k) for $C'_i \epsilon G_1$. Simultaneously, the challenger also computes CONVERT with the stored values. The adversary contacts proxy server to get the proxy data PXD. In the meanwhile challenger updates the proxy data PXD.

Challenge- The adversary submits messages M_0 and M_1 of equal lengths and an access structure A* such that either u_k is to be revoked or S_k does not satisfy A*.

The challenger flips a coin to obtain a random bit b and returns M_b encrypted with the access policy A*. It also runs Proxy-Data and returns the proxy data PXD to the adversary.

Phase 2- The adversary makes repeated queries to the KEYGEN to obtain keys for users u_{q1+1},... u_{q2} with attributes S_{q1+1},....S_{q2}. The new keys are such that if $u_k \notin$ revocation list RL, then S_k does not satisfy A*.

Guess- The adversary outputs a guess b' of b.

The adversary has an advantage defined as (Pr[b' = b] - $\frac{1}{2}$). As in the PIRATTE scheme, even if an adversary $user_j$ finds proxy portions of another $user_i$, the portions will not help him with the decryption because each user has a different set of random constants values. The SPIRC scheme provides forward secrecy because a revoked user cannot decrypt any previously recorded ciphertext.

7.3.3 Security Proof

Asymmetric Groups- Similar to PIRATTE scheme [76] for user i and attribute j, different groups are used for C'_j and D'_j. The user sends C'_j to convert and receive C''_j, where $C''_j = C'^{b_i}_j$. If both C'_j and D'_j belong to the same group and user sends D'_j to convert, then user will get $D'^{b_j}_j = g_2^{r_j b_j}$. User will also get λ_j and can get $D''^{\lambda_j}_j = g_2^{r_j \lambda_j a_j}$. Combining these two terms by multiplication will provide $g_2^{r_j(\lambda_j a_j + b_j)}$. User can use this to decrypt any ciphertext without using the proxy server for revocation. Hence asymmetric pairing is used with different groups for C_j and D'_j.

Similar to PIRATTE, SPIRC is based on the generic asymmetric bilinear group model, which considers a asymmetric pairing of e: $G_1 * G_2 \rightarrow G_T$, with the assumption that their is no isomorphism from G_1 to G_2. Both are based on Bethencourt et al.'s CP-ABE scheme [21], and hence is CPA secure. Other variations of CP-ABE, such as the CP-ABE scheme by Cheung et al. [30] are secure against *Chosen Ciphertext Attack (CCA)*. However, this thesis focuses on only Bethencourt et al.'s CP-ABE scheme [21], which has been proven feasible on mobile devices and IoT devices [10, 9].

Theorem 1- The construction of SPIRC scheme is secure under the generic bilinear group model. It assumes that there is unexpected collisions between asymmetric groups.

This thesis assumes that in the security game, $A*$ contains single attribute A_j for some attribute j. After phase 2, the adversary has the following elements for each user u_k and A_j from S_k:

G_1: g_1, g_1^{β}, $C = g_1^{\beta s}$, $C_j = g_1^{s}$, $D'_j = g_1^{r_{ukj}}$, $D''_j = g_1^{r_{ukj} a_k}$

Secret s encrypts the message and $H(j) = g_2^{h_j}$

$G2$: g_2, $D = g_2^{(\alpha+r)/\beta}$, $D_j = g_2^{r_u + h_j r_{ukj} (\lambda_k a_k + b_k)}$, $C'_j = g_2^{h_j s}$

G_T : $e(g_1, g_2)^{\alpha}$, $M \,.\, e(g_1, g_2)^{\alpha s}$

An adversary only knows u_k for all revoked users in the revocation list $RL*$. However, secret s occurs only in elements of the ciphertext C, C_j, and C'_j. To guess s, the adversary can compute $e(C, D^{(uk)}) = e(g_1, g_2)^{\alpha s + r_{uk} s}$. To determine $e(g_1, g_2)^{\alpha s}$, the adversary must compute $e(g_1, g_2)^{rs}$. However, it is not feasible to compute it from D_j. Hence, it is difficult for the adversary to determine the secret s in the security game. SPIRC is hence secure under the generic asymmetric bilinear group model.

7.4 Summary

The chapter presents a security analysis of the proposed security framework. It discusses the contribution of different security solutions. The chapter presents the detailed security analysis for the proposed protocols NSE-AA and SPIRC. The security framework fulfills the security and threat requirements that are discussed in Chapter 4. It further secures the NFC interface for secure access to the mobile-based health wallet.

CHAPTER 8

IMPLEMENTATION AND PERFORMANCE EVALUATION

This chapter presents an overview of the research work accomplished and its targeted application areas. Figure 8.1 illustrates research work findings in this thesis.

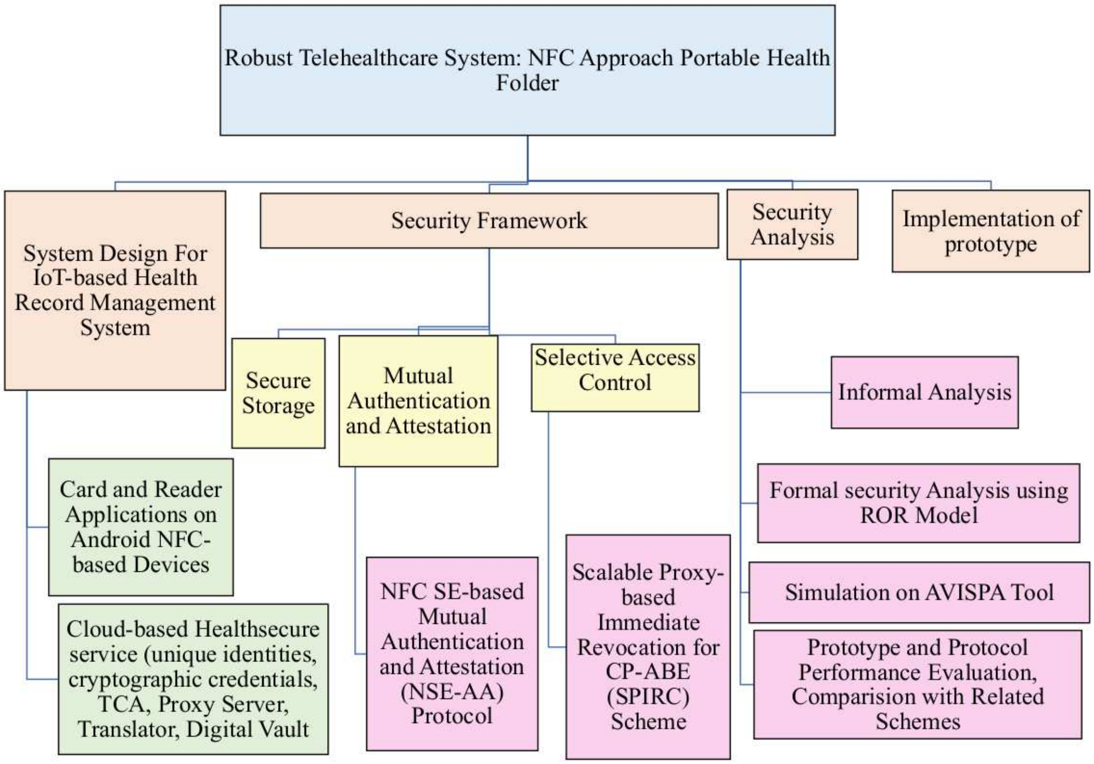

Robust Telehealthcare System: NFC Approach Portable Health Folder
System Design For IoT-based Health Record Management System
Security Framework
Security Analysis
Implementation of prototype
Secure Storage
Mutual Authentication and Attestation
Selective Access Control
Informal Analysis
Card and Reader Applications on Android NFC-based Devices
Formal security Analysis using ROR Model
Simulation on AVISPA Tool
Cloud-based Healthsecure service (unique identities, cryptographic credentials, TCA, Proxy Server, Translator, Digital Vault
NFC SE-based Mutual Authentication and Attestation (NSE-AA) Protocol
Scalable Proxy-based Immediate Revocation for CP-ABE (SPIRC) Scheme
Prototype and Protocol Performance Evaluation, Comparision with Related Schemes

Figure 8.1: Research Work in Thesis

This chapter discusses the implementation and performance details of the prototype for a smart health record management system in the following sections. The chapter also presents the implementation and performance comparison for the NSE-AA and SPIRC protocols with the related scheme. It presents the implementation of a prototype for secure IoT access using the NSE-AA protocol. The SPIRC protocol has been implemented by making changes in the CP-ABE tool kit [22].

8.1 Implementation of Prototype for the Smart Health Record Management System

The proposed system for smart health record management with secure NFC-enabled mobile devices comprises mobile-based applications for the S-MAPLE health folder HCE card application and the HCE reader applications of a health professional. This thesis implements the S-MAPLE health folder as a JSON file with a list of HL7 health records. The patient's mobile device emulates HCE-based card. Both card and reader applications are implemented using:

- 2 mid-range Android mobile devices, such as Sony Xperia M2 running Android 5.0.0 (Lollipop), which supports NFC-based HCE.
- Proxy-based SPIRC scheme for selective access.
- Gotrust based secure microSD card [57], which includes Java card chip for SE on the microSD card to store credentials and identities.
- Android SDK and Android Studio.
- MongoDB and Python interpreter to maintain the HealthSecure service with proxy server for SPIRC.

The following Figures from 8.2-8.11 illustrate a patient application with the S-MAPLE health folder and visualization of the dispersed health records. The steps in the interaction are as follows:

1. Patient and health professional register with TCA. Figure 8.2 shows patient registration and Figure 8.3 shows a confirmation for the registration.
2. The S-MAPLE health folder application synchronizes the HL7 visit health records. Synchronization involves parsing the HL7 records to generate non-HL7 data, which can assist in visualization. The device outsources encryption to the proxy server and receives the encrypted health folder, which it retains on the insecure region of the microSD card on the patient health device. Figure 8.4 illustrates the synchronization.

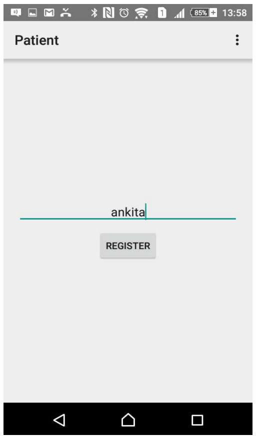


Figure 8.2: Patient Registers

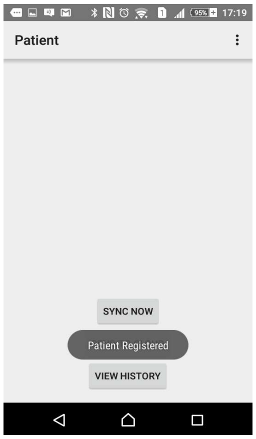
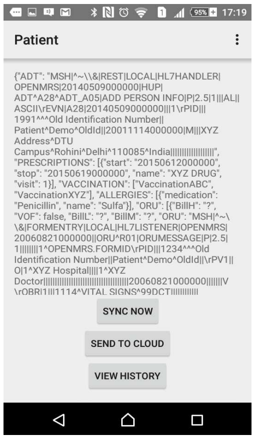


Figure 8.3: Patient Registration Confirmed



```
{"ADT": "MSH|^~\\&|REST|LOCAL|HL7HANDLER|
OPENMRS|20140509000000|HUP|
ADT^A28^ADT_A05|ADD PERSON INFO|P|2.5|1|||AL||
ASCII\rEVN|A28|20140509000000|||1\rPID|||
1991^^^Old Identification Number||
Patient^Demo^OldId||20011114000000|M|||XYZ
Address^DTU
Campus^Rohini^Delhi^110085^India||||||||||||||||||",
"PRESCRIPTIONS": [{"start": "20150612000000",
"stop": "20150619000000", "name": "XYZ DRUG",
"visit": 1}], "VACCINATION": ["VaccinationABC",
"VaccinationXYZ"], "ALLERGIES": [{"medication":
"Penicillin", "name": "Sulfa"}], "ORU": [{"BillH": "?",
"VOF": false, "BillL": "?", "BillM": "?", "ORU": "MSH|^~\
\&|FORMENTRY|LOCAL|HL7LISTENER|OPENMRS|
20060821000000||ORU^R01|ORUMESSAGE|P|2.5|
1||||||||1^OPENMRS.FORMID\rPID|||1234^^^Old
Identification Number||Patient^Demo^OldId||\rPV1||
O|1^XYZ Hospital||||1^XYZ
Doctor||||||||||||||||||||||||||||||||||||||20060821000000|||||||V
\rOBR|1|||1114^VITAL SIGNS^99DCT||||||||||
```

Figure 8.4: Patient Synchronizes Health Folder

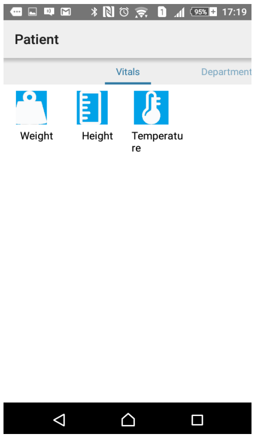


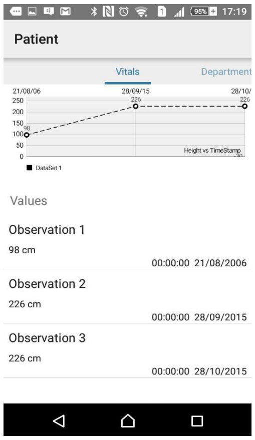
Figure 8.5: Patient's Vital Health Parameters

3. Patient can now visualize different vitals, for example, height, weight, and temperature as shown in Figure 8.5. Each vital comprises of data in a pre-parsed non-HL7 array for visualization. Figure 8.6 illustrates the visualization of height for a infant patient.

4. The health record can also refer to diagnostic images such as an X-Ray as shown in Figure 8.7.

5. Patient health records can be viewed as per the department as shown in Figure 8.8 or based on the visits as shown in Figure 8.9. A visit has two states open or close. An open visit is one in which a physician has prescribed tests. Once the patient visits a lab technician to get the lab tests, the patient taps the health card to the lab technician's device. The lab technician can view only the visit records that are open and require lab tests. The lab technician performs the

lab tests and writes the lab test reports back to the health card. The patient can take the health card back to the doctor to show results. Once the diagnostic is completed and treatment is prescribed, the physician can write the details back to the health card and close the visit.



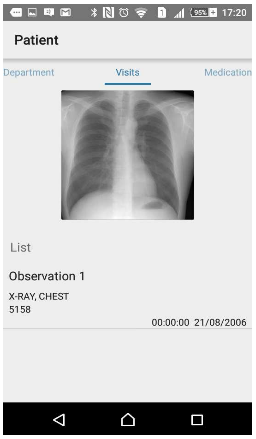
Values

Observation 1
98 cm
00:00:00 21/08/2006

Observation 2
226 cm
00:00:00 28/09/2015

Observation 3
226 cm
00:00:00 28/10/2015

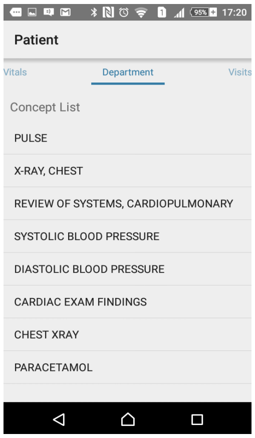
Figure 8.6: Patient's (infant) Height History



List

Observation 1
X-RAY, CHEST
5158
00:00:00 21/08/2006

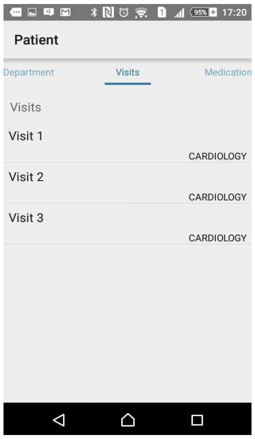
Figure 8.7: Patient X-Ray Health Data



Concept List

PULSE
X-RAY, CHEST
REVIEW OF SYSTEMS, CARDIOPULMONARY
SYSTOLIC BLOOD PRESSURE
DIASTOLIC BLOOD PRESSURE
CARDIAC EXAM FINDINGS
CHEST XRAY
PARACETAMOL

Figure 8.8: Patient Health Record for Cardiology Department



Visits

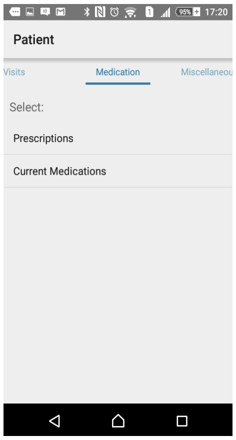
Visit 1
CARDIOLOGY

Visit 2
CARDIOLOGY

Visit 3
CARDIOLOGY

Figure 8.9: Patient Hospital Visits Information

6. The health card can assist the patient in viewing prescribed medications, as shown in Figure 8.10. The application also provides a display of miscellaneous information such as emergency contact, vaccinations, and allergies, as shown in Figure 8.11.



Patient

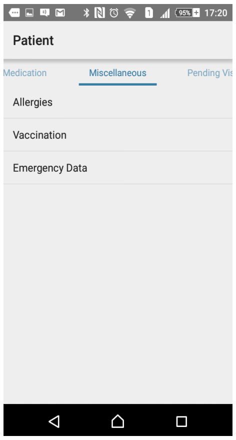
Visits | Medication | Miscellaneou

Select:

Prescriptions

Current Medications

Figure 8.10: Patient Medications From Different Visits



Patient

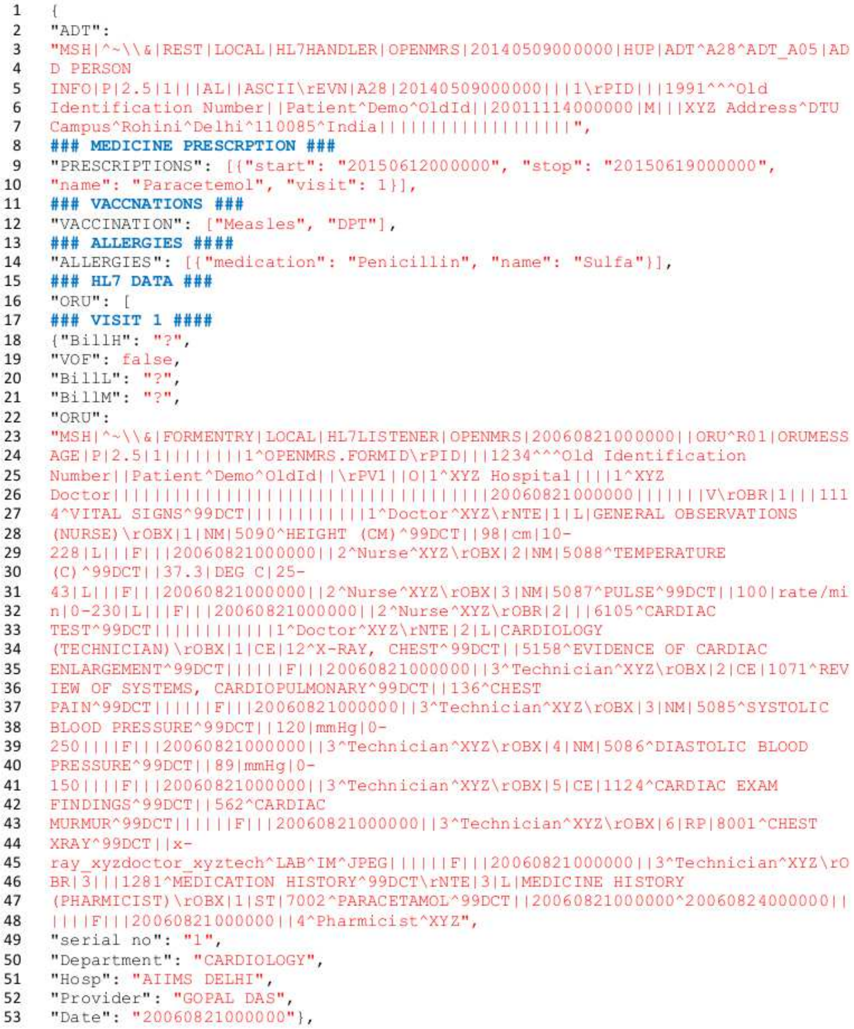
Medication | Miscellaneous | Pending Vis

Allergies

Vaccination

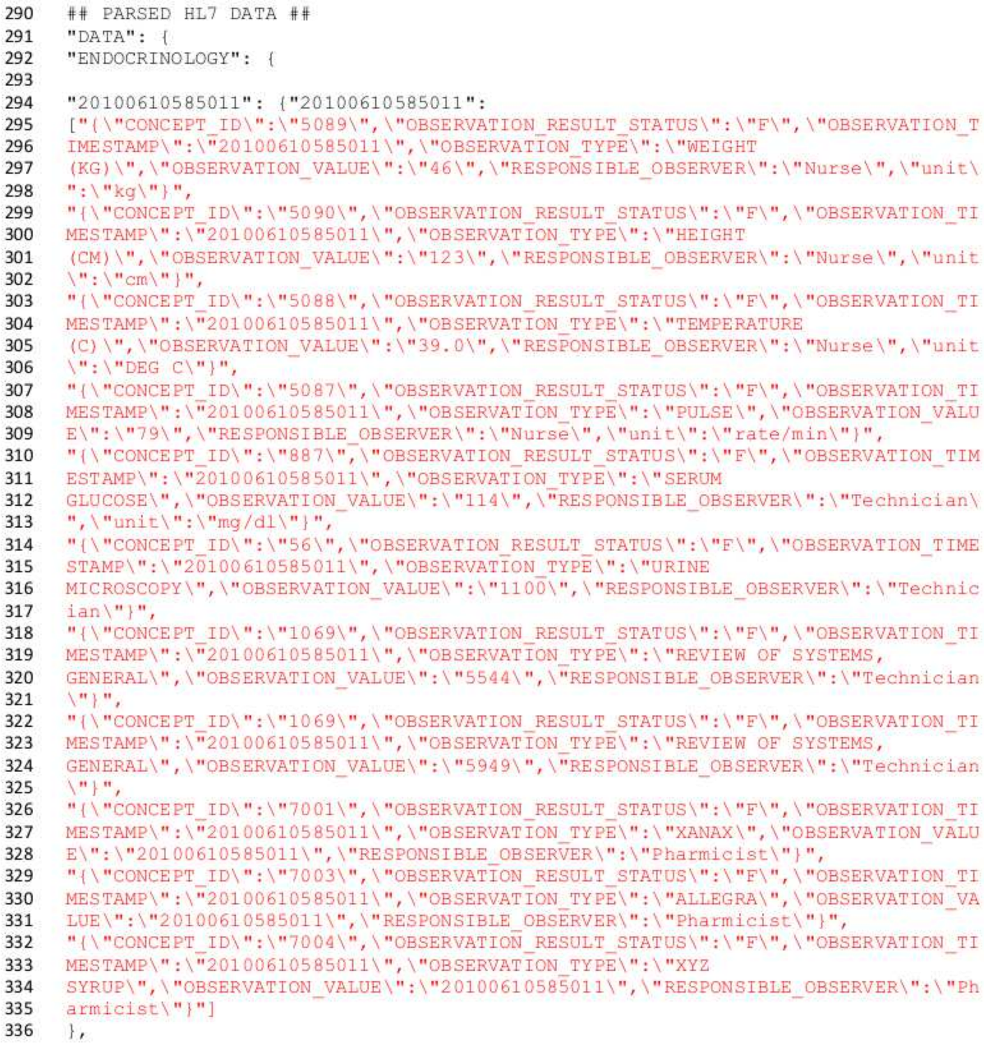
Emergency Data

Figure 8.11: Patient Miscellaneous Health Information



```
{
"ADT":
"MSH|^~\\&|REST|LOCAL|HL7HANDLER|OPENMRS|20140509000000|HUP|ADT^A28^ADT_A05|AD
D PERSON
INFO|P|2.5|1|||AL||ASCII\rEVN|A28|20140509000000|||1\rPID|||1991^^^Old
Identification Number||Patient^Demo^OldId||20011114000000|M|||XYZ Address^DTU
Campus^Rohini^Delhi^110085^India|||||||||||||||||||",
### MEDICINE PRESCRPTION ###
"PRESCRIPTIONS": [{"start": "20150612000000", "stop": "20150619000000",
"name": "Paracetemol", "visit": 1}],
### VACCNATIONS ###
"VACCINATION": ["Measles", "DPT"],
### ALLERGIES ####
"ALLERGIES": [{"medication": "Penicillin", "name": "Sulfa"}],
### HL7 DATA ###
"ORU": [
### VISIT 1 ####
{"BillH": "?",
"VOF": false,
"BillL": "?",
"BillM": "?",
"ORU":
"MSH|^~\\&|FORMENTRY|LOCAL|HL7LISTENER|OPENMRS|20060821000000||ORU^R01|ORUMESS
AGE|P|2.5|1||||||||1^OPENMRS.FORMID\rPID|||1234^^^Old Identification
Number||Patient^Demo^OldId||\rPV1||O|1^XYZ Hospital||||1^XYZ
Doctor||||||||||||||||||||||||||||||||||||||20060821000000||||||V\rOBR|1|||111
4^VITAL SIGNS^99DCT||||||||||||1^Doctor^XYZ\rNTE|1|L|GENERAL OBSERVATIONS
(NURSE)\rOBX|1|NM|5090^HEIGHT (CM)^99DCT||98|cm|10-
228|L|||F|||20060821000000||2^Nurse^XYZ\rOBX|2|NM|5088^TEMPERATURE
(C)^99DCT||37.3|DEG C|25-
43|L|||F|||20060821000000||2^Nurse^XYZ\rOBX|3|NM|5087^PULSE^99DCT||100|rate/mi
n|0-230|L|||F|||20060821000000||2^Nurse^XYZ\rOBR|2|||6105^CARDIAC
TEST^99DCT||||||||||||1^Doctor^XYZ\rNTE|2|L|CARDIOLOGY
(TECHNICIAN)\rOBX|1|CE|12^X-RAY, CHEST^99DCT||5158^EVIDENCE OF CARDIAC
ENLARGEMENT^99DCT||||||F|||20060821000000||3^Technician^XYZ\rOBX|2|CE|1071^REV
IEW OF SYSTEMS, CARDIOPULMONARY^99DCT||136^CHEST
PAIN^99DCT||||||F|||20060821000000||3^Technician^XYZ\rOBX|3|NM|5085^SYSTOLIC
BLOOD PRESSURE^99DCT||120|mmHg|0-
250||||F|||20060821000000||3^Technician^XYZ\rOBX|4|NM|5086^DIASTOLIC BLOOD
PRESSURE^99DCT||89|mmHg|0-
150||||F|||20060821000000||3^Technician^XYZ\rOBX|5|CE|1124^CARDIAC EXAM
FINDINGS^99DCT||562^CARDIAC
MURMUR^99DCT||||||F|||20060821000000||3^Technician^XYZ\rOBX|6|RP|8001^CHEST
XRAY^99DCT||x-
ray_xyzdoctor_xyztech^LAB^IM^JPEG||||||F|||20060821000000||3^Technician^XYZ\rO
BR|3|||1281^MEDICATION HISTORY^99DCT\rNTE|3|L|MEDICINE HISTORY
(PHARMICIST)\rOBX|1|ST|7002^PARACETAMOL^99DCT||20060821000000^20060824000000||
||||F|||20060821000000||4^Pharmicist^XYZ",
"serial no": "1",
"Department": "CARDIOLOGY",
"Hosp": "AIIMS DELHI",
"Provider": "GOPAL DAS",
"Date": "20060821000000"},
```

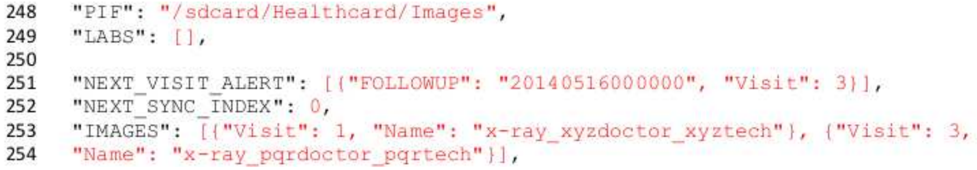
Figure 8.12: S-MAPLE Health Folder: HL7 Visit Data



```
## PARSED HL7 DATA ##
"DATA": {
"ENDOCRINOLOGY": {

"20100610585011": {"20100610585011":
["{\"CONCEPT_ID\":\"5089\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_T
IMESTAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"WEIGHT
(KG)\",\"OBSERVATION_VALUE\":\"46\",\"RESPONSIBLE_OBSERVER\":\"Nurse\",\"unit\
":\"kg\"}",
"{\"CONCEPT_ID\":\"5090\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_TI
MESTAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"HEIGHT
(CM)\",\"OBSERVATION_VALUE\":\"123\",\"RESPONSIBLE_OBSERVER\":\"Nurse\",\"unit
\":\"cm\"}",
"{\"CONCEPT_ID\":\"5088\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_TI
MESTAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"TEMPERATURE
(C)\",\"OBSERVATION_VALUE\":\"39.0\",\"RESPONSIBLE_OBSERVER\":\"Nurse\",\"unit
\":\"DEG C\"}",
"{\"CONCEPT_ID\":\"5087\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_TI
MESTAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"PULSE\",\"OBSERVATION_VALU
E\":\"79\",\"RESPONSIBLE_OBSERVER\":\"Nurse\",\"unit\":\"rate/min\"}",
"{\"CONCEPT_ID\":\"887\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_TIM
ESTAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"SERUM
GLUCOSE\",\"OBSERVATION_VALUE\":\"114\",\"RESPONSIBLE_OBSERVER\":\"Technician\
",\"unit\":\"mg/dl\"}",
"{\"CONCEPT_ID\":\"56\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_TIME
STAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"URINE
MICROSCOPY\",\"OBSERVATION_VALUE\":\"1100\",\"RESPONSIBLE_OBSERVER\":\"Technic
ian\"}",
"{\"CONCEPT_ID\":\"1069\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_TI
MESTAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"REVIEW OF SYSTEMS,
GENERAL\",\"OBSERVATION_VALUE\":\"5544\",\"RESPONSIBLE_OBSERVER\":\"Technician
\"}",
"{\"CONCEPT_ID\":\"1069\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_TI
MESTAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"REVIEW OF SYSTEMS,
GENERAL\",\"OBSERVATION_VALUE\":\"5949\",\"RESPONSIBLE_OBSERVER\":\"Technician
\"}",
"{\"CONCEPT_ID\":\"7001\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_TI
MESTAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"XANAX\",\"OBSERVATION_VALU
E\":\"20100610585011\",\"RESPONSIBLE_OBSERVER\":\"Pharmicist\"}",
"{\"CONCEPT_ID\":\"7003\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_TI
MESTAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"ALLEGRA\",\"OBSERVATION_VA
LUE\":\"20100610585011\",\"RESPONSIBLE_OBSERVER\":\"Pharmicist\"}",
"{\"CONCEPT_ID\":\"7004\",\"OBSERVATION_RESULT_STATUS\":\"F\",\"OBSERVATION_TI
MESTAMP\":\"20100610585011\",\"OBSERVATION_TYPE\":\"XYZ
SYRUP\",\"OBSERVATION_VALUE\":\"20100610585011\",\"RESPONSIBLE_OBSERVER\":\"Ph
armicist\"}"]
},
```

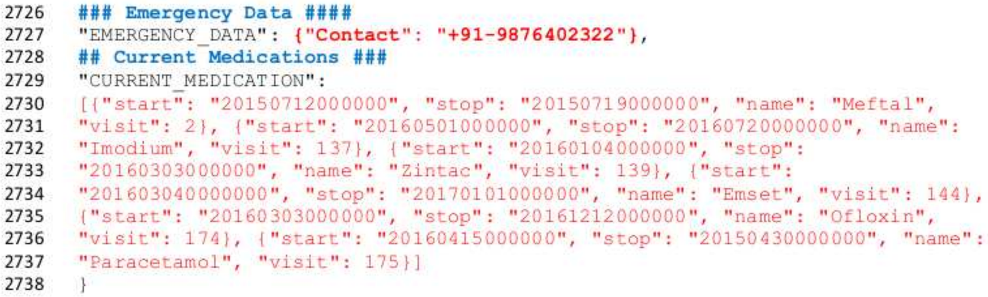
Figure 8.13: S-MAPLE Health Folder: Parsed HL7 Data



```
"PIF": "/sdcard/Healthcard/Images",
"LABS": [],

"NEXT_VISIT_ALERT": [{"FOLLOWUP": "20140516000000", "Visit": 3}],
"NEXT_SYNC_INDEX": 0,
"IMAGES": [{"Visit": 1, "Name": "x-ray_xyzdoctor_xyztech"}, {"Visit": 3,
"Name": "x-ray_pqrdoctor_pqrtech"}],
```

Figure 8.14: S-MAPLE Health Folder: Other Health Information



```
### Emergency Data ####
"EMERGENCY_DATA": {"Contact": "+91-9876402322"},
## Current Medications ###
"CURRENT_MEDICATION":
[{"start": "20150712000000", "stop": "20150719000000", "name": "Meftal",
"visit": 2}, {"start": "20160501000000", "stop": "20160720000000", "name":
"Imodium", "visit": 137}, {"start": "20160104000000", "stop":
"20160303000000", "name": "Zintac", "visit": 139}, {"start":
"20160304000000", "stop": "20170101000000", "name": "Emset", "visit": 144},
{"start": "20160303000000", "stop": "20161212000000", "name": "Ofloxin",
"visit": 174}, {"start": "20160415000000", "stop": "20150430000000", "name":
"Paracetamol", "visit": 175}]
}
```

Figure 8.15: S-MAPLE Health Folder: Miscellaneous Information

Figure 8.12 presents the S-MAPLE health folder in the form of a JSON file. It stores health records from various sources in two arrays. The first array comprises of the HL7 health records and the second array comprises of the parsed HL7 data as discussed in Section 3.1.1. The S-MAPLE health folder also stores other information, as illustrated in Figures 8.14 and 8.15.

- PIF: Locations for images for the health folder.
- Counters for Visit: Counters for iteration of the visit information.
- Image Information: Image names for various visits. PIF points to the location of the folder with the images, such as X-Ray images.
- Emergency: Emergency information, such as the person of contact.
- Current Medications: List of current medications extracted from the visit HL7 information along with the start and stop dates.

8.2 Performance Evaluation for the Smart Health Record Management System

This thesis assumes that a doctor views past 10 records at a visit. For the S-MAPLE health folder, this corresponds to an original folder of a size of 17 KB and encrypted size of 57 KB.

The performance results for accessing an S-MAPLE health folder with 10 text-based health records (size 57 KB) are as follows:

- t_E: Time for Encryption on server: 4197 ms
- t_N: NFC transactions: 207 ms
- t_{MAA}: HCE Mutual Authentication: 3551 ms
- t_B: Transfer over Bluetooth: 5119 ms
- t_{DP}: Proxy decryption support: 450 ms
- t_{DD}: Device decryption: 2143 ms
- t_D: Net decryption time: $t_{DP} + t_{DD}$: 2593 ms
- t_{NR}: $t_N + t_{MAA} + t_B + t_D$: 11470 ms $\approx$ 12 s

The total time to read the card over HCE comprises of the NFC transaction, Bluetooth pairing, HCE mutual authentication, transfer over Bluetooth followed by decryption on the remote device. According to a smart card health system proposed by Kardas et al. [79] it takes around 9 s to start a user session once the card is inserted into the reader device. Hence, the overheads of around 12 s seems acceptable since it provides easy access over the NFC tap along with a robust security handshake. We did not find performance for access time in the other related schemes.

S-MAPLE health folder has an acceptable access time. The health folder can assist in reducing the waiting times in crowded public hosptials as discussed in the our technical report for a simulation study for patient flow management [135].

8.3 Key Findings for NSE-AA protocol

This thesis propose the NSE-AA protocol for mutual authentication and attestation over HCE to establish security handshake and trust between the mobile devices. The NSE-AA protocol is also feasible for an IoT device access with a user-based mobile device over an NFC-based HCE interface.

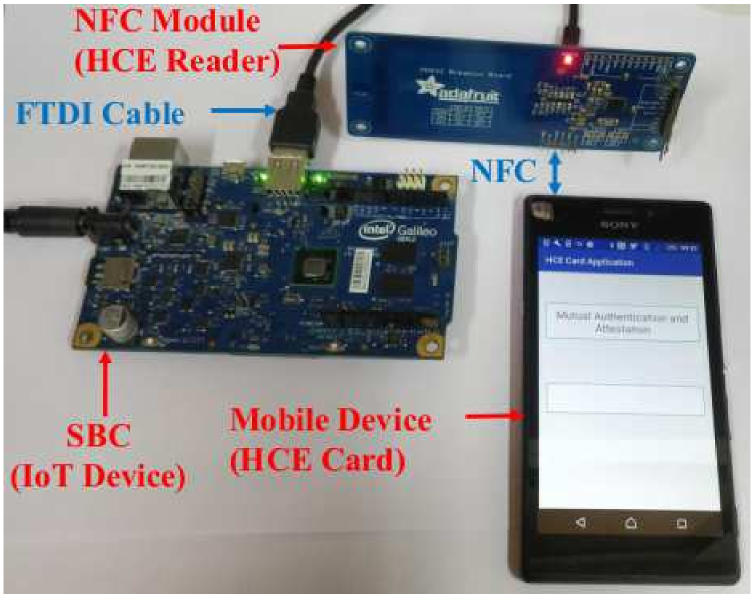


Figure 8.16: System Prototype for IoT Access with NSE-AA Protocol

8.3.1 Implementation of Prototype for IoT Access with NSE-AA

In this thesis, we have developed a prototype using commercial mid-range priced *Single Board Computer (SBC)* for an IoT device with an inbuilt NFC controller that is accessed by a user's mobile device over an NFC tap. Figure 8.16 illustrates the system prototype. The mobile device emulates a software-based HCE card, which interacts with the HCE reader application on the NFC controller. Both devices retain secure commercial microSD card-based SE and TPM-based attestation modules. The SBC operating system resides on an insecure region of the microSD card. It is connected to the NFC controller with a *Future Technology Devices International (FTDI)* cable. Both devices use APDU packets to initiate the phases of the NSE-AA protocol, as discussed in previous sections. The devices use the HCE mode to establish automated Bluetooth pairing through the exchange of the Bluetooth address over HCE and direct communication using the RFCOMM sockets. The computational overheads for the prototype are comparable to the overheads presented in Tables 8.2 and 8.3.

In a practical world, the consumer IoT devices, such as remote surveillance cameras can have a similar inbuilt NFC controller to interface with a user's handheld device over an HCE tap for the secure setup of configurations, key management, and control of logs. A proximity-based IoT access can ensure proof-of-locality so that the devices are not prone to become victims to cause cyber attacks, such as DDoS attacks. The benefits of using HCE with bidirectional communication and open platform for development makes such an application feasible in the practical work. Further details of the prototype are given in our technical report [133].

8.3.2 Performance Evaluation for NSE-AA

Performance of NFC modes- Figure 8.17 shows the evaluation of reading and writing data (3000 bytes) using different NFC modes. The results of transfer for HCE with Bluetooth are comparable to that of tags. Hence, HCE can help in a fast bidirectional interface for accessing the IoT device with security, privacy, and trust.

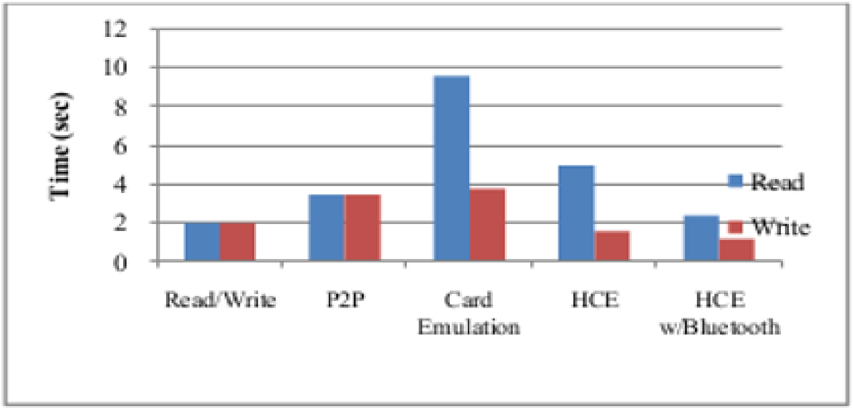


Figure 8.17: Performance Evaluation for NFC Modes

Performance for mutual authentication and attestation- Figure 8.18 illustrates average IoT access time with HCE, HCE with mutual authentication, and HCE with mutual authentication and associated mutual attestation. Overheads of authentication and attestation are higher but they are acceptable for secure and trustful access of an IoT device.

8.3.3 Comparison of NSE-AA with Related Schemes

Table 8.1 compares the NSE-AA protocol with different attestation schemes for security and threat requirements. NSE-AA satisfies all security requirements and protects from various threats. Aziz et al. [19] presents an extension of TLS with TPM-based mutual attestation for remote access but lacks proof-of-locality. The MAT protocol by Toegl and Hutter [142] attest only the remote device. None of these schemes provide user anonymity, secure storage, and protection from DDoS and MITM attacks. NSE-AA is the only protocol that supports bidirectional communication over the NFC interface for security and trust handshake.

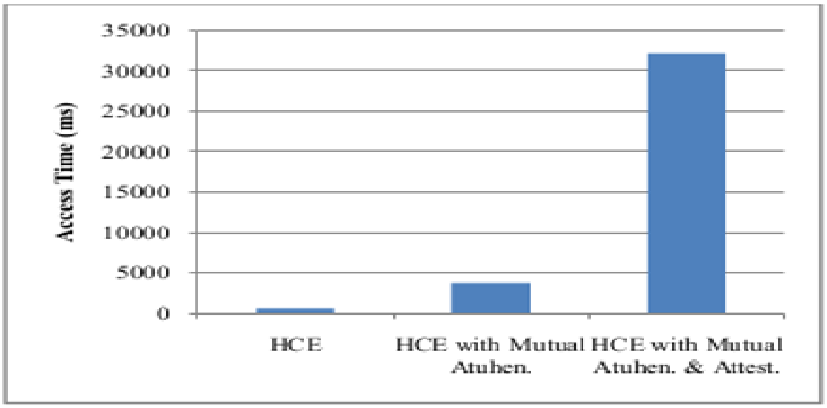


Figure 8.18: Performance for Access Time with HCE

Table 8.1: Comparison of NSE-AA for Security and Threat Requirements

Requirement	**Toegl and Hutter [142]**	**Aziz et al. [19]**	**NSE-AA**
S1:Confidentiality	Y	Y	Y
S2:Integrity	Y	Y	Y
S3:Mutual Authentication and Attestation	N	Y	Y
S4:Privacy	Y	Y	Y
S5:User Anonymity	N	N	Y
S6:NFC Proof-of-Locality	Y	N	Y
S7:Secure Storage	N	N	Y
T1:DoS	N	N	Y
T2:Replay	Y	Y	Y
T3:Collusion	N	Y	Y
T4:Parallel Session	N	N	Y
T5:Forgery	Y	Y	Y
T6:Platform Impersonation	Y	N	Y
T7:MITM	N	N	Y
T8:Insider Attack	N	N	Y

Table 8.2 discusses the comparison of computation overheads for different schemes. This

thesis considers RSA and AES encryption algorithms for asymmetric and symmetric encryption respectively because they are commonly available on commercially available SEs. This thesis considers the following parameter sizes- nonce: 64 bit, Identity: 64 bit, SML: 2k bits, PCR: 160 bits [34], RSA Certificate size: 2048 bit, RSA key length 1024 bits, AES key length 1024 bits.

For comparison, we assume the following cryptographic overheads as considered by Gope et al. [59]:

- SHA operation time T_H: 7.81* 10^{-4} msec
- Symmetric key AES encryption and decryption time T_S: 10.5*10^{-4} msec
- Asymmetric RSA time using Chinese remainder theorem T_{AS} (RSA) 12.06 msec

This thesis assumes that the key derivation function time *Tkdf* is approximately the same as T_H. Time for RSA encryption is incremental as per the input blocks and also pads plaintext, if it is less than 1024 bits. The MAT protocol by Toegl and Hutter [142] has lower overheads as compared to the other schemes. However, it satisfies only limited security requirements. NSE-AA takes around $46\mathrm{T_H}$ + $258\mathrm{T_S}$ + $26\mathrm{T_{AS}}$ computations, which is around 313.87 ms as per timing assumptions considered by Gope and Hwang [59].

Table 8.2: Comparison of NSE-AA for Computational Overheads

Phase	**Toegl and Hutter [142]**	**Aziz et al. [19]**	**NSE-AA**
Mutual Authentication	0	$2\mathrm{T_H}+6\mathrm{T_{AS}}$	$22\mathrm{T_H}+18\mathrm{T_S}$
AIK Certificate	0	$14\mathrm{T_H}+64\mathrm{T_S}+36\mathrm{T_{AS}}$	$20\mathrm{T_H}+176\mathrm{T_S}+14\mathrm{T_{AS}}$
Mutual Attestation	$19\mathrm{T_{AS}}$	$4\mathrm{T_H}+6\mathrm{T_{AS}}$	$4\mathrm{T_H}+64\mathrm{T_S}+12\mathrm{T_{AS}}$
Net Computation	$19\mathrm{T_{AS}}$	$20\mathrm{T_H}+64\mathrm{T_S}+48\mathrm{T_{AS}}$	$46\mathrm{T_H}+ 258\mathrm{T_S}+26\mathrm{T_{AS}}$
Net Time (ms)	229.14	579.00	313.87

Tables 8.3 shows comparison of communication overheads. NSE-AA requires 10 messages with around 27936 bits for communication. The overheads of the NSE-AA protocol are lower as compared to the scheme by Aziz et al. [19] because it uses symmetric encryption. Although, the

overheads of the NSE-AA protocol are higher as compared to the scheme proposed by Toegl and Hutter [142], but it provides better security.

Table 8.3: Comparison of NSE-AA for Communication Overheads

Phase	**Toegl and Hutter [142]**	**Aziz et al. [19]**	**NSE-AA**
Messages	6	10	10
Communication bits	12352	35904	27936

An implementation of NSE-AA for access of an IoT device with in-built NFC control via an external HCE card on user device provides satisfactory results. The IoT device is analogous to a medical device and the user device to a patient S-MAPLE card.

8.4 Key findings for SPIRC Scheme

The SPIRC scheme encrypts the S-MAPLE health folder for confidentiality, selective access, and scalable revocation. A proxy server is used to outsource encryption. Decryption is performed partially on the user's mobile device with the help of a trusted proxy server.

8.4.1 Performance Evaluation for SPIRC

Impact of the number of health records- Figure 8.19 shows the impact of the number of records for encryption, decryption, and access time. It indicates that there is a significant increase in time to read as the number of records increase. However, it does not affect the encryption and decryption timings, because the size of ciphertext does not change. An AES key encrypts the health folder, and the CP-ABE key is used to encrypt the AES key, which remains constant. The read time comprises of communication time and decryption time to view the records. Since the transmission time increases with the number of health records, the read time also increases.

Impact of the number of attributes- Figure 8.20 shows the increase in the time for key generation, encryption, and decryption with the increase in the number of attributes.

Impact of attributes on storage- Figure 8.21 shows that the storage size of encrypted health folder does not get affected significantly by an increase in the number of attributes. However, similar to CP-ABE, the key size increases with the increase in the number of attributes.

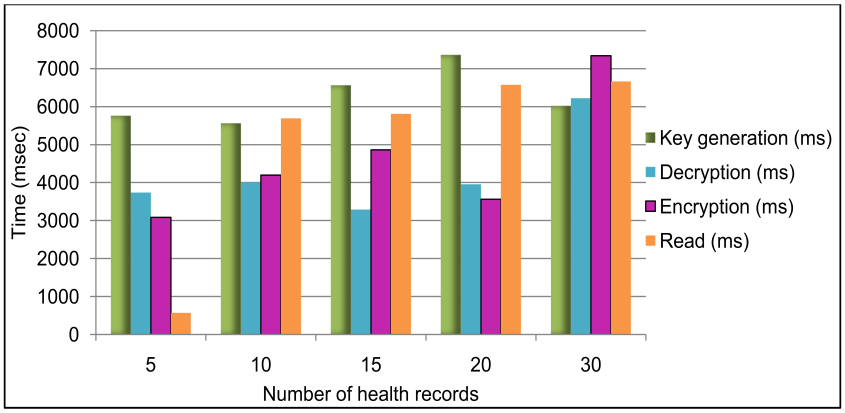


Figure 8.19: Impact of Number of Health Records on Access Time

”

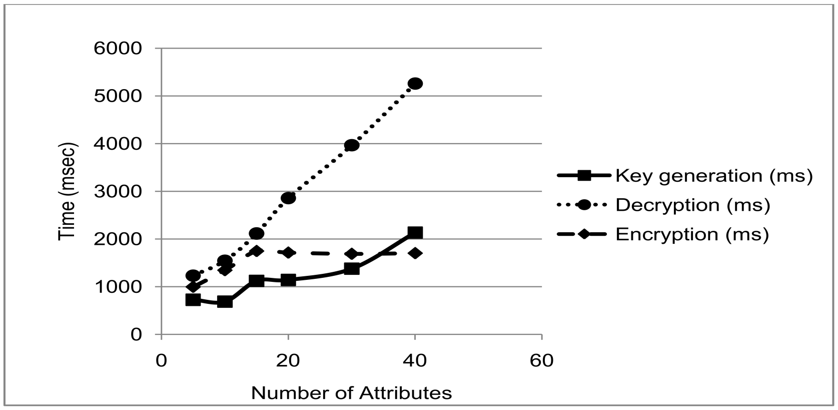


Figure 8.20: Impact of Attributes on Access Time

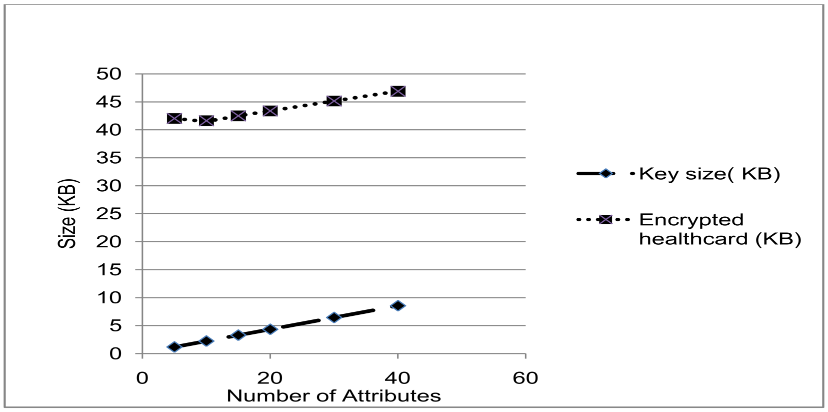


Figure 8.21: Impact of Attributes on Storage

8.4.2 Comparison of SPIRC with Related Schemes

Comparison for Revocation Requirements- Table 8.4 shows the comparison of the different revocation techniques for the revocation requirements. Only the proposed SPIRC protocol fulfils all the requirements *C1-C5* from Section 1.4.4. Hence, it is suitable for secure and selective access of a portable ciphertext and provides ease of use to the owner and other non-revoked users.

Comparison for Timing Overheads- Table 8.5 shows average time for the health folder encryption and decryption using PIRATTE and SPIRC schemes. This thesis finds that the overheads for the security computations for encryption and decryption of health folder for both PIRATTE and SPIRC are similar with acceptable values for usage. SPIRC has lower overheads of proxy decryption as compared to PIRATTE because the proxy data is associated with only random constants, unlike Lagrange-based secret sharing in PIRATTE. Hence, the total decryption time for SPIRC is lower as compared to PIRATTE.

Table 8.4: Comparison of SPIRC for Revocation Requirements

Requirments	**PIRATTE [76]**	**SPIRC (Proposed)**
C1:Require Prior Revocation List	No	No
C2:Require Re-encryption	No	No
C3:Require Re-distribution of Keys	No	No
C4:Revoke Scalable users	No	Yes
C5:Independent of Ciphertext	Yes	Yes

Table 8.5: Comparison of SPIRC for Average Timings

Event	**PIRATTE (ms)**	**SPIRC (ms)**
Server Encryption	4197	4197
Proxy decryption	1864	450
Device decryption	2143	2143

Key Generation- Figure 8.22 illustrates the secret key generation time as a function of the number of attributes for user revocation. Constant random parameters λ_i, a_i, and b_i associated with a $user_i$

are stored in a file and can be retrieved at the time of decryption. Hence, the corresponding secret key generation time is less in SPIRC as compared to PIRATTE.

Figure 8.23 illustrates key generation time for attribute-based revocation. The implementation is available only for the SPIRC scheme. The scheme associates each attribute for user_i with a unique set of constants that can be altered by the proxy server.

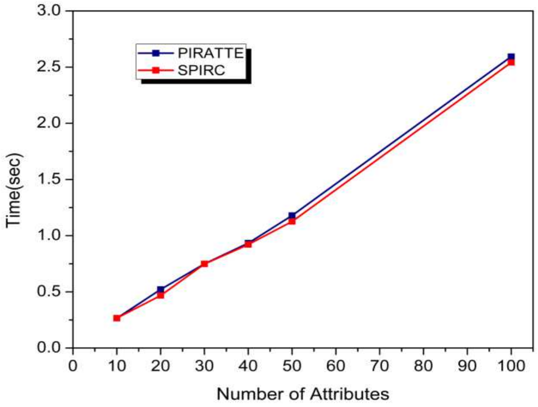


Figure 8.22: Key Generation: User-based Revocation

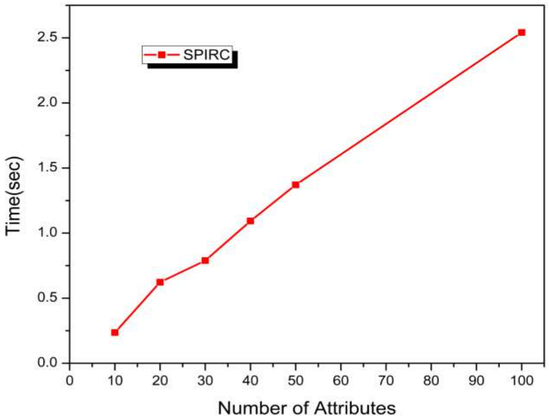


Figure 8.23: Key Generation: Attribute-based Revocation

Encryption- The encryption time is evaluated by generating policies for specified number of attributes (10, 20, 30 . . . . , 50,100). In Figures 8.24 and 8.25 it is observed that the encryption time is linear with the increase in the number of leaf nodes of the access the policy. Since the encryption schemes are similar for both SPIRC and PIRATTE schemes, they both take approximately the same amount of time for encryption.

Proxy Conversion- The PIRATTE scheme involves polynomial computation, whereas SPIRC uses the constant random parameters for each user_i stored in the files generated at the time of secret key generation. This thesis observes that there is a decrease in the conversion time for the SPIRC scheme as compared to the PIRATTE scheme. Figures 8.26 and 8.27 illustrate the impact of attributes on the convert time.

Decryption- The decryption time for both SPIRC and PIRATTE schemes is approximately the same. Figures 8.28 and 8.29 illustrate the impact of attributes on the decryption time.

Delegation of Secret Key- Figure 8.30 illustrates that SPIRC scheme has a lower delegation time as compared to PIRATTE scheme for user-based revocation. SPIRC scheme presents an efficient attribute-based delegation where a user can delegate some or all the attributes to another user temporarily. Figure 8.31 illustrates the impact of the number of attributes on the delegation for attribute-based revocation for SPIRC scheme.

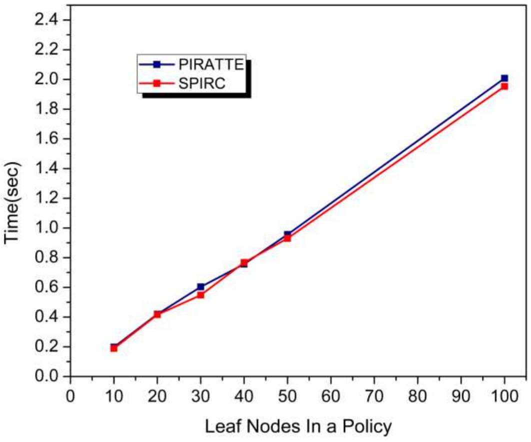


Figure 8.24: Encryption: User-based Revocation

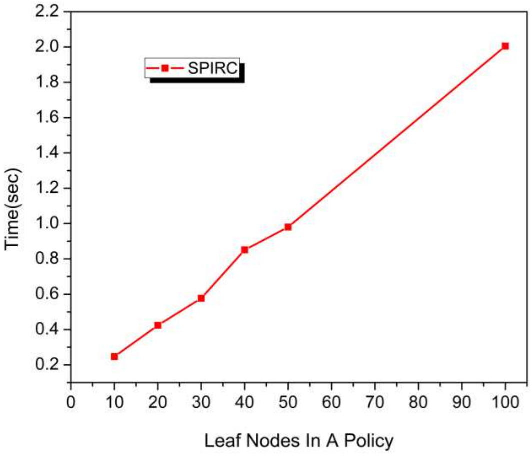


Figure 8.25: Encryption: Attribute-based Revocation

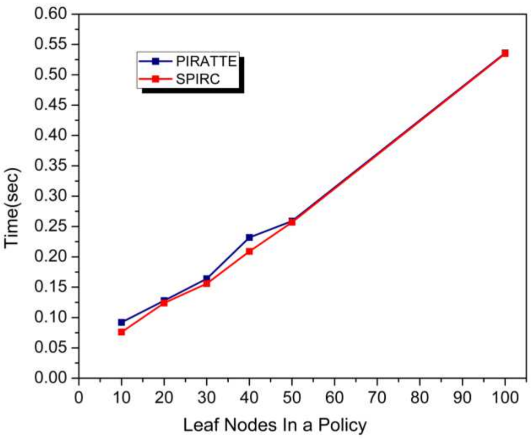


Figure 8.26: Convert: User-based Revocation

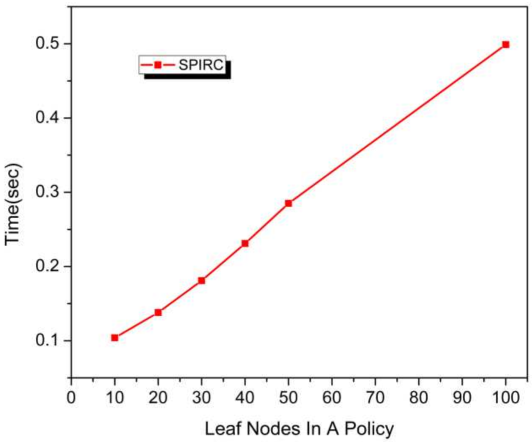


Figure 8.27: Convert: Attribute-based Revocation

Comparison for performance- Table 8.6 shows performance comparison of CP-ABE techniques for overheads of storage and computational overheads of encryption, decryption and revocation for users. The table also describes the terms used for comparison. The comparison assumes

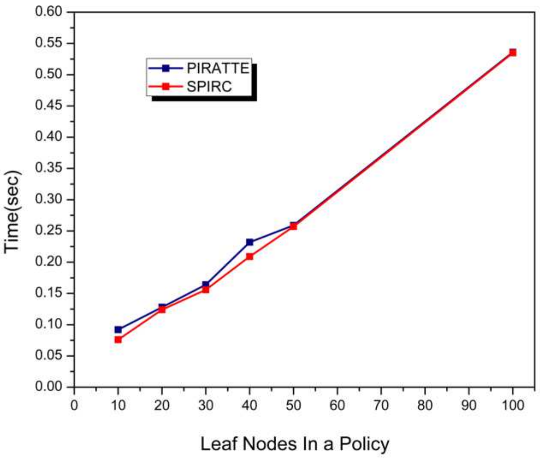


Figure 8.28: Decryption: User-based Revocation

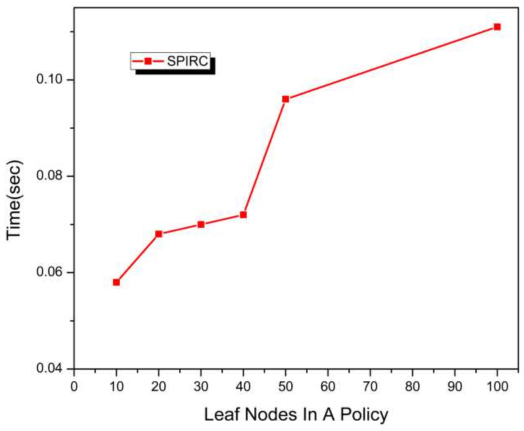


Figure 8.29: Decryption: Attribute-based Revocation

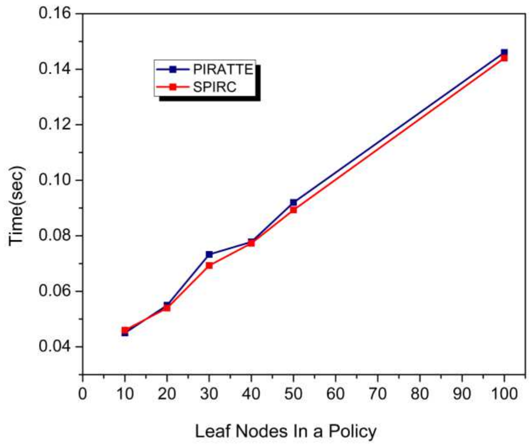


Figure 8.30: Delegation: User-based Revocation

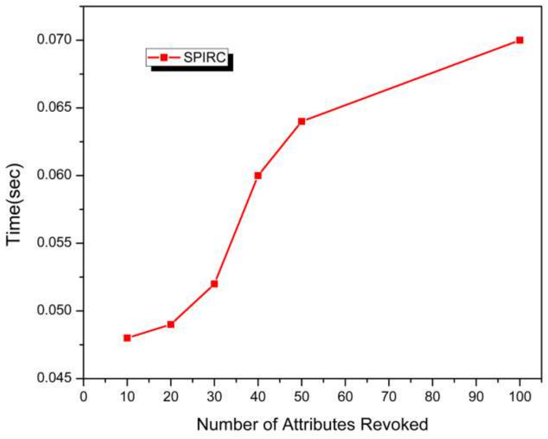


Figure 8.31: Delegation: Attribute-based Revocation

that all CP-ABE schemes use asymmetric group pairing. All schemes have similar lengths for public key *PK*. However, since there is no generation of polynomial *P* in the SPIRC scheme, it has a shorter master key *MK* as compared to the PIRATTE [76] scheme. Both schemes have similar lengths for private key *SK*. However, it is longer as compared to the Bethencourt et al.'s CP-ABE [21] and M-PERMREV[39] schemes (both have same lengths for *SK*). For all schemes, *SK* is dependent on the number of attributes A_U allocated to the user *U*. The ciphertext length is dependent on the number of attributes of ciphertext A_C and is the same for all schemes. There is no broadcast overhead for Bethencourt et al.'s CP-ABE scheme [21]. The Broadcast overhead for PIRATTE is

Table 8.6: Comparison of SPIRC for Storage and Performance Overheads

Scheme	CP-ABE [21]	PIRATTE [76]	M-PERMREV [39]	SPIRC (Proposed scheme)
Size of keys and ciphertext in different schemes				
PK	$2L_{G1} + L_{G2} + L_{GT}$	$2L_{G1} + L_{G2} + L_{GT}$	$2L_{G1} + L_{G2} + L_{GT}$	$2L_{G1} + L_{G2} + L_{GT}$
MK	$L_{G1} + L_{Zp}$	**$L_{G1} + (1+t)L_{Zp}$**	$L_{G1} + 2L_{Zp}$	$L_{G1} + L_{Zp}$
SK	$L_{G2} + (a+L_{G1}+L_{G2})\|A_U\|$	**$L_{G2} + (a+L_{G1}+2L_{G2})\|A_U\|$**	$L_{G2} + (a+L_{G1}+L_{G2})\|A_U\|$	**$L_{G2} + (a+L_{G1}+2L_{G2})\|A_U\|$**
CT	$(2\|A_C\|+1)L_{G1} + L_{G2}$	$(2\|A_C\|+1)L_{G1} + L_{G2}$	$(2\|A_C\|+1)L_{G1} + L_{G2}$	$(2\|A_C\|+1)L_{G1} + L_{G2}$
Broadcast	None	**$tZ_p+\|A_U\|L_{G2}$** $+Z_p$	$L_{G1} + L_{G2}$	**$Z_p+ \|A_U\|L_{G2}$**
Comparison of computational overhead				
Encrypt.	$(2A_C+1)G_1 + G_2$	$(2A_C+1)G_1 + G_2$	$(2A_C+1)G_1 + G_2$	$(2A_C+1)G_1 + G_2$
Decrypt.	$2A_UC_e + (2\|S\|+2)G_2$	**$3A_UC_e + (2\|S\|+2)G_2$**	$2A_UC_e + (2\|S\|+3)G_2$	**$3A_UC_e + (2\|S\|+2)G_2$**

A_C: Attributes of ciphertext C;A_U: Attributes of user U;a: Length of an attribute;C_e: Number of bilinear pairings

G_i: Group or operations in group i, i = 1 or 2;S: Least interior nodes satisfying access structure (including root node);
L*: Bit length of element in *;t number of users to be revoked

dependent on the number of revoked users and the number of attributes of a user A_U. The broadcast overhead of M-PERMREV is constant since it is only a state update for a user. However, it links a separate user state for each ciphertext and does not satisfy revocation requirement *C5*. The SPIRC scheme broadcasts a constant value for proxy data and is independent of the number of revoked users and dependent only on the number of attributes of a user A_U. Also, unlike the M-PERMREV scheme, the proxy data is not be linked with the ciphertext such that there is an overhead of creating separate proxy data for each ciphertext for a user. The encryption time is dependent on the number of attributes in the ciphertext A_C and is similar to all schemes. The decryption for PIRATE and SPIRC schemes use an extra bilinear pairing for proxy-based decryption. Hence, they have a higher decryption time as compared to the decryption time for Bethencourt et al.'s CP-ABE and M-PERMREV schemes. The overall decryption time for SPIRC scheme is smaller as compared to the PIRATTE scheme because it generates a simpler form of proxy data.

8.5 Summary

The chapter discusses the implementation details for the prototype and their performance evaluation. The S-MAPLE health folder has acceptable overheads for access. The chapter also presents the performance evaluation for the proposed protocols NSE-AA and SPIRC. They are compared theoretically as well as practically with the related schemes. Both protocols have better performance as compared to the related schemes and can help accomplish the security requirements with efficiency.

CHAPTER 9

CONCLUSION AND FUTURE WORK

9.1 Thesis Summary

In this thesis, we have designed and implemented the architecture for a smart health record management system with secure NFC-enabled mobile devices. None of the previous portable health record systems can assist in patient mobility across hospitals. The novelty of this thesis is to provide an S-MAPLE health folder on a patient's mobile device using HCE for direct interaction with the mobile device of a health professional. It stores dispersed health records in a standard HL7 format for integration and interoperability between different hospitals. It fulfills all requirements *R1-R7* for mobility of patients to different hospitals as discussed in Table 3.1. The patient's mobile device uses NFC-based HCE card emulation for a contactless health wallet. HCE mode has been used for the first time for a health record management system on mobile devices to the best of our knowledge. NFC provides proof-of-locality and secures access between devices due to proximity. It makes the MITM and eavesdropping attacks difficult. The HCE mode provides bidirectional communication and an open platform for developers to develop a proprietary health wallet. It also provides a secure platform with sufficient storage as compared to the other NFC modes. Previously mobile devices have used Peer-to-Peer NFC mode for Bluetooth pairing. However, it uses the insecure NDEF message format and causes a device to pair with a malicious node over Bluetooth. In this thesis, we use the bidirectional feature of HCE mode to automate Bluetooth pairing for transfer of large data with better throughput.

Current mobile devices have sufficient storage to retain health records from the past few years on the S-MAPLE health folder along with a summary of the older records. Hence, the health folder can provide a complete readily available health history for a patient to seek timely diagnosis and treatment.

This thesis proposes a robust security framework with security solutions for secure storage, provenance of health records, mutual authentication and attestation, and selective access to the health folder with a CP-ABE scheme. Use of Secure Elements for secure storage provides tamper-resistant storage for credentials and performing secure computations. The framework proposes new protocols: NSE-AA for authentication with associated attestation and SPIRC for scalable revocation using CP-ABE for sharing and updating with selective RBAC.

Previously none of the NFC-based security schemes addressed the important issues of authentication and trust together. Both are essential to ensure valid devices with trustful states are allowed to communicate with each other. The NSE-AA protocol proposed in this thesis presents a combined mutual authentication and attestation technique over HCE tap for an end-to-end security handshake and trust between the IoT/S-MAPLE health folder and user devices/mobile devices. Measures have been taken to overcome various attacks, such as the impersonation, replay, and collusion attacks. The tamper-resistant SE and TPM provide secure storage and perform all cryptographic computations. NFC-AA enhances the authentication scheme proposed by Thammarat et al. [140] and the attestation scheme by Aziz et al. [19]. NSE-AA is proved secure under the ROR model [2] as well as through simulation on the AVISPA tool. It provides better security with satisfactory overheads as compared to the previous schemes by Aziz et al. [19] and Toegl and Hutter [142].

It is essential to share the health records with selective RBAC with different stakeholder and also prote ct them from malicious users. In this thesis, we look into the Bethencourt et al.'s CP-ABE scheme [21], which has been proved feasible on mobile devices. However, the previous research schemes do not satisfy all the requirement satisfies all the revocation requirements *C1-C5* for ease of maintenance of ciphertext on a portable device. In this thesis, the proposed novel SPIRC protocol improves the PIRATTE scheme by Jahid et al. [75] for scalable revocation. It satisfies all the revocation requirements *C1-C5*. The overheads for the generation of a master key and broadcast data are as lower as compared to the PIRATTE scheme. Also, SPIRC does not associate any proxy data with the ciphertext as in the M-PERMREV scheme by Dolev et al. [39]. Our work is the first novel attempt to address secure data on a portable device using Bethencourt et al.'s CP-ABE scheme [21] with scalable user revocation. The SPIRC scheme also supports single key authority and multi-key authority delegation as well as attribute-based revocation.

The prototype performance evaluation on mid-range Android devices indicates acceptable delays for access to the S-MAPLE health card. We present details of an HCE reader application and HCE card application for the smart health record management system. The devices tap and enable secure read and write as per the requirement for selective access. It can also assist in reducing hospital wait times in overcrowded hospitals by providing secure and easy support for patient identification and registration. The novel future health folder on the health wallet can provide mobility to patients to visit various hospitals and the availability of a reliable health history for seeking timely medical treatment. It also supports novel selective access to health card as a contactless card by multiple stakeholders.

The performance comparison of the NSE-AA and SPIRC protocol indicates that they have are more secure and have satisfactory overheads as compared to the related schemes. The NSE-AA protocol uses the HCE mode, which has better performance compared to reader mode and peer-to-peer mode. The overheads with mutual authentication and attestation are high but acceptable due to the robust security handshake and trust between devices. NSE-AA protocol satisfies all the security and threat requirements, as discussed in Table 8.1. It has lower overheads for computations and communication as compared to Aziz et al.'s scheme [19]. Although the overheads of NSE-AA are higher as compared to the Toegl and Hutter's scheme [142], it provides better security framework.

The performance evaluation for the SPIRC scheme indicates that with the increase in the number of health records, there is no effect on the encryption and decryption times. However, the access time increases due to the communication overheads. The increase in the attributes for the access structure causes an increase in time for the key generation, encryption, and decryption. The health folder size does not increase with the increase in the number of attributes. However, the length of the decryption key increases.

Both PIRATTE and SPIRC schemes have similar encryption time. However, SPIRC has a lower decryption time, key generation time, proxy key generation and delegation key generation times. Both PIRATTE and SPIRC have similar lengths for the public key. However, PIRATTE has a longer master key since it generates a polynomial for Lagrange's interpolation. Both schemes have similar lengths for the decryption keys. All schemes have ciphertext size dependent on the number of attributes. The broadcast overhead for SPIRC is independent of the number of users to be revoked. The overall decryption time for SPIRC is shorter as compared to PIRATTE due to simpler proxy components.

The proposed smart health record system can assist in patient mobility and improve healthcare for both patients as well as health care providers. It can especially assist in largely populated nations like India, where there are many overheads to maintain a secure, centralized health record system. The proposed health wallet can assist for ease of access and maintenance of complete health history for patients with provenance of data.

9.2 Limitations of the Proposed S-MAPLE Health Folder

The S-MAPLE health folder has certain limitations, which we can improve in future. Some limitations are listed as follows:

- **Aggregation:** Currently, for requirement, *R1* the S-MAPLE folder only aggregates the health records. There are open challenges of semantic interoperability of health records due to lack of common standards across different countries. Various techniques, such as Ontology and cloud-based services [65] can address them in future. Health standards, such as *Fast Healthcare Interoperability Resources (FHIR)* [53] can provide semantic interoperability across different health systems. Hence, the S-MAPLE must support interoperability of health records for ease of patient mobility across different hospitals with different standards.

- **Personal health history and monitoring devices:**

  Recently, there is a trend of *Patient Generated Health Data ([118])* from personal observations, health monitoring, and fitness devices. It is vital to aggregate such personal health information also for a complete patient health history. There are various medically approved devices which are useful to maintain health vitals such as blood pressure, blood sugar levels. We can further enhance the S-MAPLE health folder to aggregate the health readings from such personal health monitoring devices. The secure wallet can maintain their provenance, and the personal readings can provide an up to date health history of patients, especially seeking partial home treatment.

- **Ease of entry of records for medical professionals**r It is also essential to reduce the burden of digital entry for medical professionals from a mobile reader application in future because it is easy for them to write than to type. There must be a provision to scan handwritten health records and translate to standard digital health formats, such as HL7, especially to translate paper-based health records on the S-MAPLE health folder.
- **Improving HCE card access time** Current S-MAPLE health folder uses SEs with slow processing capabilities, which leads to increased time in end-to-end mutual authentication between the two SEs. High-speed SEs and extended APDUs with HCE can help in the access time between the healthcard and the reader devices.

9.3 Future Work

The S-MAPLE health card consists of the translation of health records from different hospitals in the HL7 [68] format for interoperability across different hospitals. However, semantic interoperability is a huge challenge for patients moving among various places with different policies, terminologies, and languages. Semantic Interoperability is defined as integrating resources that were developed using different vocabularies and different perspectives on the data. For semantic interoperability, the systems must exchange data so that meaning of the data is unchanged and the data can be translated into any other format easily. FHIR-based [53] electronic healthcare records can help achieve syntactic and semantic interoperability in future.

We can improve the NSE-AA protocol with biometric-based attributes in the mutual authentication phase [55, 97]. We can use runtime attestation to verify the dynamic device state, such as in the *Control-Flow Integrity (CFI)* scheme [4]. Alternately TEE can provide a lightweight and fast attestation [159] and also secure HCE, such as in the *Trusted Host-based Card Emulation (THCE)* [100].

We can also use modifications of lightweight provably secure mutual authentication schemes, such as suggested by Chang et al. [28]. Alternate high-speed SEs with extended APDU commands can reduce computations time for NSE-AA. We can also look into biometric-based authentication using smart cards, such as suggested by Yang et al. [162]. We can use HCE with extended APDU commands to send larger data and improve the access speed. Credentials with time stamps can provide offline access control, as suggested in Mobile On-Offline NFC-based Physical Access Control System (MOONACS) [61]. With the improving computational capabilities on mobile devices, the framework can be used to securely access IoT devices and other mobile-based applications, such as pickup of a child from school [20].

The SPIRC protocol enhances the Bethencourt et al.'s CP-ABE scheme [21]. In future other advanced CP-ABE can be considered. We can compare SPIRC with other schemes, such as those by Ibraimi et al. [73] using provably secure CP-ABE scheme by Cheung et al. [30] and Lewko et al. [88] based on LSSS. We must also look into other schemes, which are CCA secure as well as optimized for resource-constrained devices, such as the constant ciphertext size scheme by Emura et al. [44].

We can extend the encryption scheme for signature as suggested by Liua et al.'s [93] Ciphertext-Policy Attribute-Based Signcryption scheme. It enhances the CP-ABE scheme by Waters [151] to incorporate encryption along with a digital signature. Liu et al. [92] suggest an efficient mobile computing scheme for using CP-ABE with a combination of offline and online ciphertext generation phases to minimize the computations on the mobile device. Such a scheme may be used on a mobile health folder to minimize computation overheads in future. We can introduce searchability in the encrypted ciphertext to save the efforts of decryption and then viewing the records as in the cloud-based solution proposed by Tong et al. [143].

PUBLISHED WORK-

- **Patent:**

  D. Sethia, D. Gupta, H. Saran, U. Arora, M. Goyal Portable Computing Device Based Secure Medical Records Management: Application number 1313/DEL/2015, Dated: 12/05/2015; Awaiting Examination

- **Journal:**

  1. **D. Sethia, H. Saran, D. Gupta**, CP-ABE for Selective Access with Scalable Revocation: A case study for Mobile-based Healthfolder, Journal of Network System 20 (4) (2018) 689-701, doi: 10.6633/IJNS.201807$_2$0(4).11), *(Scopus Indexed)*
  2. **D. Sethia, D. Gupta, H. Saran**, NFC Secure Element-based Mutual Authentication and Attestation for IoT access, IEEE Transaction on Consumer Electronics 64 (8) 2018, doi: 10.1109/TCE.2018.2873181, *(SCI Indexed)*
  3. **D. Sethia, D. Gupta, Daya, H. Saran, R. Agrawal and A. Gaur,** Mutual Authentication Protocol For Secure NFC Based Mobile Healthcard, IADIS International Journal on Computer Science and Information Systems 11(2) (2016), 195-202, *(SCI Indexed)*
  4. **D. Sethia, D. Gupta, H. Saran**, Smart Health Record Management with Secure NFC-enabled Mobile Devices, Elseiver Journal of Smart Health, Nov 2018, doi: 10.1016/j.smhl.2018.11.001, *Peer reviewed since 2017*

- **Conference:**

  1. **D. Sethia et al.**, Selective IoT Access with Scalable CP-ABE Revocation and Delegation, Proc. Int. Conf. Computational Science and Computational Intelligence, 2017
  2. **D. Sethia, D. Gupta, H. Saran**, Security framework for portable NFC mobile based health record system. Proc. IEEE Int. Conf. WiMob, New York,2016: 1-8
  3. **D. Sethia et al.**, NFC based secure mobile healthcare system, Proc IEEE Int. Conf. COMSNETS, Bangalore, 2014: 1-6

BIBLIOGRAPHY

[1] A. Abbas and S. U. Khan, “A Review on the State-of-the-Art Privacy-Preserving Approaches in the e-Health Clouds,” *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1431–1441, Jul. 2014.

[2] M. Abdalla, P.-A. Fouque, and D. Pointcheval, “Password-Based Authenticated Key Exchange in the Three-Party Setting,” in *Proc. Int. Work. Public Key Cryptography*, Jan. 2005, pp. 65–84.

[3] M. Abdulnabi *et al.*, “A Distributed Framework for Health Information Exchange Using Smartphone Technologies,” *Journal of Biomedical Informatics*, vol. 69, pp. 230–250, 2017.

[4] T. Abera *et al.*, “Things, trouble, trust: On building trust in IoT systems,” in *Proc. IEEE Int. Conf. Design Automation Conference*, Jun. 2016.

[5] M. Ahmed and M. Ahamad, “Protecting health information on mobile devices,” in *Proc. ACM Int. Conf. Data and Application Security and Privacy*, 2012, pp. 229–240.

[6] J. A. Akinyele, M. W. Pagano, and M. D. Green, “Securing Electronic Medical Records Using Attribute-Based Encryption On Mobile Devices,” in *Proc. ACM Work. Security and privacy in smartphones and mobile devices*, 2011, pp. 75–86.

[7] M. Alattar and M. Achemlal, “Host-based Card Emulation: development, security,and ecosystem impact analysis,” in *Proc. IEEE Int. Conf. High Performance Computing and Communications*, Aug. 2014.

[8] S. Alshehri and R. K. Raj, “Secure Access Control for Health Information Sharing Systems,” in *Proc. IEEE Int. Conf. Healthcare Informatics*, 2013.

[9] M. Ambrosin, M. Cont, and T. Dargahi, “On the Feasibility of Attribute-Based Encryption on Smartphone Devices,” in *Proc. ACM Int. Work. IoT challenges in Mobile and Industrial Systems*, 2015, pp. 49–54.

[10] M. Ambrosin *et al.*, “On the Feasibility of Attribute-Based Encryption on Internet of Things Devices,” *IEEE Access*, vol. 36, pp. 25–35, 2016.

[11] S. Amendola *et al.*, “RFID Technology for IoT-Based Personal Healthcare in Smart Spaces,” *IEEE Internet of Things Journal*, vol. 1, pp. 144 – 152, 2014.

[12] N. Anciaux *et al.*, “A Tamper-Resistant and Portable Healthcare Folder,” *Hindawi Journal International Journal of Telemedicine and Applications*, vol. 2008, 2008.

[13] “Android developer IsoDep,” n.d, Last Accessed on Jan 2019. [Online]. Available: https://developer.android.com/reference/android/nfc/tech/IsoDep

[14] A. Armando *et al.*, “The AVISPA Tool for the Automated Validation of Internet Security protocols and Applications,” in *Computer Aided Verification, Lecture Notes in Computer Science, Springer*, vol. 3576, Jul. 2005.

[15] A. Armando, M. Alessio, and V. Luca, “Trusted Host-based Card Emulation,” in *Proc. IEEE Int. Conf. on High Performance Computing and Simulation*, Jul. 2015.

[16] N. Asokan *et al.*, “Mobile Trusted Computing,” *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1189–1206, Aug. 2014.

[17] N. Attrapadung and H. Imai, “Attribute-Based Encryption Supporting Direct/Indirect Revocation Modes,” *Cryptography and Coding Springer Lecture Notes in Computer Science*, vol. 5921, pp. 278–300, 2009.

[18] S. Avancha, A. Baxi, and D. Kotz, “Privacy in mobile technology for personal healthcare,” *ACM Computing Surveys*, vol. 45, pp. 1–54, 2012.

[19] N. Aziz, N. Udzir, and R. Mahmod, “Extending TLS with Mutual Attestation for Platform Integrity Assurance,” *Journal of Communications*, vol. 9, pp. 63–72, 2015.

[20] Y.-W. Bai, C.-N. Fu, and J.-H. Yang, “Using NFC tags and smartphones to design a reliable mechanism to pick a child up from school,” in *Proc. IEEE Int. Conf. Consum. Electr.*, Jan. 2018.

[21] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext Policy Attribute Based Encryption,” in *Proc. IEEE Symposium on Security and Privacy*, 2007, pp. 321–334.

[22] ——, “Ciphertext-policy attribute-based encryption toolkit,” 2011, Last accessed on Jan. 2019. [Online]. Available: http://acsc.cs.utexas.edu/cpabe/

[23] B. Blobel and P. Pharow, “A model driven approach for the German health telematics architectural framework and security infrastructure,” *International Journal of Medical Informatics*, 2007.

[24] M. A. Bouazzouni, E. Conchond, and F. Peyrard, “Trusted mobile computing: An overview of existing solutions,” *Future Gener. Comput. Syst.*, vol. 80, pp. 596–612, 2018.

[25] S. Bouzefrane *et al.*, *Evaluation of Java Card Performance*. Springer, Berlin, Heidelberg, 2008.

[26] L. Catarinucci *et al.*, “An IoT-Aware Architecture for Smart Healthcare Systems,” *IEEE Internet of Things Journal*, vol. 2, pp. 515 – 526, 2015.

[27] U. B. Ceipidor and other, “KerNeeS A protocol for mutual authentication between NFC phones and pos terminals for secure payment transactions,” in *Proc. IEEE Int. Conf. Information Security and Cryptology*, 2012, pp. 331–334.

[28] C.-C. Chang and H.-D. Le, “A Provably Secure, Efficient, and Flexible Authentication Scheme for Adhoc Wireless Sensor Networks,” *IEEE Trans. Wireless Communications*, vol. 15, pp. 357–366, 2016.

[29] W. Chen *et al.*, “Developing electronic health records in Taiwan,” *IEEE IT Professional*, vol. 12, pp. 17–25, 2010.

[30] L. Cheung and C. Newport, “Provably Secure Ciphertext Policy ABE,” in *Proc. ACM Int. Conf. Computer and Communications Security*, 2007, pp. 456–465.

[31] K. Christopher, *European Data Protection Law*. Oxford University Press, 2007.

[32] V. Coskun, B. Ozdenizci, and K. Ok, “A Survey on Near Field Communication (NFC) Technology,” *Springer Wireless Personal Communications*, vol. 71, pp. 2259–2294, 2013.

[33] “Danish Health Portal,” https://www.sundhed.dk, n.d, Last accessed on Jan 2019.

[34] S. Delaune *et al.*, “Formal analysis of protocols based on TPM state registers,” in *Proc. IEEE Int. Conf. Computer Security Foundations Symposium*, Aug. 2011.

[35] D. Detmer, M. Bloomrosen, B. Raymond, and P. Tang, “Integrated Personal Health Records: Transformative Tools for Consumer-Centric Care,” *BMC Medical Informatics and Decision Making*, vol. 8, no. 45, 2008.

[36] A. Dhar *et al.*, “PROXIMITEE: Hardened SGX Attestation and Trusted Path through Proximity Verification : |ryptology ePrint Archive, Report 2018/902,” 2018, Last accessed on Jan. 2019. [Online]. Available: https://eprint.iacr.org/2018/902

[37] A. Dmitrienko *et al.*, “Securing Electronic Medical Records Using Attribute-Based Encryption On Mobile Devices,” *Biomedical Engineering Systems and Technologies of the series Communications in Computer and Information Science*, vol. 273, pp. 365–379, 2010.

[38] D. Dolev and A. Yao, “On the Security of Public Key Protocols,” *IEEE Trans. Information Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[39] S. Dolev, N. Gilboa, and M. Kopeetsky, “Permanent Revocation in Attribute Based Broadcast Encryption,” in *Proc. IEEE Int. Conf. Cyber Security*, Dec. 2012.

[40] C. Doukas, T. Pliakas, and I. Maglogiannis, “Mobile Healthcare Information Management utilizing Cloud Computing and Android OS,” in *Proc. IEEE Int. Conf. Engineering in Medicine and Biology Society*, Aug. 2010.

[41] M. Eichelberg, T. Aden, and J. Riesmeier, “A Survey and Analysis of Electronic Healthcare Record Standards,” *ACM Computing Surveys,*, vol. 20, pp. 1–47, 2005.

[42] J.-E. Ekberg, K. Kostiainen, and N. Asokan, “The Untapped Potential of Trusted Execution Environments on Mobile Devices,” *IEEE Journal Security and Privacy*, vol. 12, pp. 29 – 37, 2014.

[43] E. J. Emanuel and L. L. Emanuel, “What Is Accountability in Health Care?” *Annals of Internal Medicine*, vol. 124, no. 2, pp. 29–39, Feb. 1996.

[44] K. Emura and other, “A ciphertext-policy attribute-based encryption scheme with constant ciphertext length,” *International Journal of Applied Cryptography*, vol. 2, no. 1, pp. 46–59, 2010.

[45] “The new EU Regulation on the protection of personal data: what does it mean for patients?” n.d, Last Accessed on Jan 2019. [Online]. Available: http://www.eu-patient.eu/globalassets/policy/data-protection/data-protection-guide-for-patients-organisations.pdf

[46] H. Eun, H. Lee, and H. Oh, “Conditional privacy preserving security protocol for NFC applications,” *IEEE Trans.Cons. Elec.*, vol. 59, no. 1, pp. 153–160, Feb. 2013.

[47] “FBI Cyber Bulletin: Distributed Denial of Service Attack Against DNS Host Highlights Vulnerability of Internet of Things Devices,” n.d, Last Accessed: Sep. 2018. [Online]. Available: https://info.publicintelligence.net/FBI-IoT-DDoS.pdf

[48] J. L. Fernandez-Aleman *et al.*, “Security and privacy in electronic health records: A systematic literature review,” *Elseiver Journal of Biomedical Informatics*, vol. 46, pp. 541–562, 2013.

[49] S. T. Force, “HLPSL Tutorial A Beginner’s Guide to Modelling and Analysing Internet Security Protocols,” n.d, Last accessed on Jan 2019. [Online]. Available: http://www.avispa-project.org/package/tutorial.pdf

[50] ——, “A Security Analysis of NFC Implementation in the Mobile Proximity Payments Environment,” n.d, Last accessed on 2018-12-15. [Online]. Available: https://docplayer.net/11274084-A-security-analysis-of-nfc-implementation-in-the-mobile-proximity-payments-environment.html

[51] “French Health Card,” http://universaldesign.ie/, n.d, Last Accessed on Jan 2019.

[52] L. Francis *et al.*, “Practical NFC Peer-to-Peer Relay Attack using Mobile Phones,” in *Proc. Springer Int. Work. on Radio Frequency Identification: Security and Privacy Issues*, 2010, pp. 35–49.

[53] B. Franz, A. Schuler, and O. Kraus, “Applying FHIR in an Integrated Health Monitoring System,” *European Journal for Biomedical Informatics*, vol. 11, no. 2, 2015.

[54] R. W. Gardner *et al.*, “Securing medical records on smart phones,” in *Proc. ACM Int. Work. on Security and privacy in medical and home-care systems*, Nov. 2009, pp. 31–40.

[55] S. Ghosh *et al.*, “Swing-pay: One Card Meets All User Payment and Identity Needs: A Digital Card Module using NFC and Biometric Authentication for Peer-to-Peer Payment,” *IEEE Consum. Electron. Mag.*, vol. 6, pp. 82–93, 2017.

[56] “Global Platform, TEE system architecture,” n.d, Last accessed on Jan. 2019. [Online]. Available: https://globalplatform.org/wp-content/uploads/2018/09/GPD_TEE_SystemArch_v1.1.0.10-for-v1.2_PublicReview.pdf

[57] “GO-Trust ID,” n.d, Last Accessed on Jan 2019. [Online]. Available: https://www.gotrustid.com/

[58] J. D. Gold and M. Ball, “The Health Record Banking Imperative: A Conceptual Model,” *IBM Systems Journal*, vol. 46, pp. 43–55, 2007.

[59] P. Gope and T. Hwang, “Lightweight and energy-efficient mutual authentication and key agreement scheme with user anonymity for secure communication in global mobility networks,” *IEEE Syst. Journal*, vol. 10, no. 4, pp. 1370–1379, Apr. 2015.

[60] V. Goyal *et al.*, “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data,” in *Proc. ACM Int. Conf. Computer and communications security*, Oct. 2006, pp. 89–98.

[61] D. Gruntz, C. Arnosti, and M. Hauri, “MOONACS a mobile onoffline NFC-based physical access control system,” *International Journal of Pervasive Computing and Communications*, pp. 2–22, Feb. 2016.

[62] S. Haas *et al.*, “Aspects of privacy for electronic health records,” *International Journal of Medical Informatics*, vol. 80, pp. 26–31, 2011.

[63] T. Halevi, D. Ma, N. Saxena, and T. Xiang, “Secure Proximity Detection for NFC Devices Based on Ambient Sensor Data,” *Computer Security Springer Lecture Notes in Computer Science*, vol. 7459, pp. 379–396, 2012.

[64] E. Hall *et al.*, “Enabling Remote Access to Personal Electronic Medical Records,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, pp. 133 – 139, 2003.

[65] R. Hammami, H. Bellaaj, and A. H. Kacem, “Interoperability for medical information systems: an overview,” *Health and Technology*, vol. 4, no. 3, pp. 261–272, Sep. 2014.

[66] G. P. Hancke, K. E. Mayes, and K. Markantonakis, “Confidence in smart token proximity: Relay attacks revisited,” *Elseiver Journal of Computers and Security*, vol. 28, pp. 615–627, 2009.

[67] "HL7 application programming interface parser," n.d, Last Accessed on Jan 2019. [Online]. Available: http://hl7api.sourceforge.net/

[68] "Introduction to HL7 Standards," n.d, Last accessed on 2017-08-29. [Online]. Available: http://www.hl7.org/implement/standards/

[69] "HL7 Resources," n.d, Last accessed on Jan 2019. [Online]. Available: https://corepointhealth.com/resource-center/hl7-resources/hl7-orm-message/

[70] L.-C. Huang *et al.*, "Privacy preservation and information security protection for patients' portable electronic health records," *Elseiver Journal of Computers in Biology and Medicine*, vol. 39, pp. 743 – 750, 2009.

[71] C.-H. Hung, Y.-W. Bai, and J.-H. Ren, "Design and implementation of a door lock control based on a Near Field Communication of a smartphone," in *Proc. IEEE Int. Conf. Consum. Electr.*, Jun. 2015.

[72] J. Hur and D. K. Noh, "Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems," *IEEE Trans. Parallel and Distributed Systems*, vol. 22, pp. 1214–1221, 2011.

[73] L. Ibraimi *et al.*, "Mediated Ciphertext-Policy Attribute-Based Encryption and Its Application," *Information Security Applications, Springer Lecture Notes in Computer Science*, vol. 5932, pp. 309–323, 2009.

[74] "ISO 7816 part 4: Interindustry Commands for interchange APDUs," n.d, Last accessed on Jan 2019. [Online]. Available: http://www.cardwerk.com/smartcards

[75] S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-based Access Control in Social Networks with Efficient Revocation," in *Proc. ACM Int. Symp. Information, Computer and Communications Security*, 2011, pp. 411–415.

[76] ——, “PIRATTE: Proxy-based Immediate Revocation of ATTribute-based Encryption,” in *arXiv preprint arXiv*, 2012, pp. 1–14.

[77] “Java Card Platform Security,” n.d, Last Accessed on Jan 2019. [Online]. Available: http://www.oracle.com/technetwork/java/javacard/documentation/javacardsecuritywhitepaper-149957.pdf

[78] S. Kahn and V. Sheshadri, “Medical Record Privacy and Security in a Digital Environment,” *IEEE IT Professional*, vol. 10, pp. 46–52, 2008.

[79] G. Kardas and E. T. Tunali, “Design and implementation of a smart card based healthcare information system,” *Journal Computer Methods and Programs in Biomedicine Elseiver*, vol. 81, pp. 66–78, 2006.

[80] Z. Kfir and A. Wool, “Access Control: Policies, Models, and Mechanisms,” in *Proc. IEEE Int. Conf. Security and Privacy for Emerging Areas in Communications Networks*, Sep. 2005.

[81] S. Klein, “The Veterans Health Administration: Implementing Patient-Centered Medical Homes in the Nation’s Largest Integrated Delivery System,” *The Commonwealth Fund*, vol. 1537, 2011.

[82] T. Korak and L. Wilfinger, “Handling the NDEF signature record type in a secure manner,” in *Proc. IEEE Int. Conf. RFID-Technologies and Applications*, 2012.

[83] D. Kotz *et al.*, “Privacy and Security in Mobile Health: A Research Agenda,” *IEEE Computer*, vol. 49, no. 6, pp. 22–30, Jun. 2016.

[84] S. Kungpisdan and S. Metheekul, “A Secure Offline Key Generation With Protection Against Key Compromise,” in *Proc. Int. Conf. World Multi-conference on Systemics*, 2009.

[85] A. Lahtela, M. Hassinen, and V. Jylha, “RFID and NFC in healthcare: Safety of hospitals medication care,” in *Proc. IEEE Int. Conf. Pervasive Computing Technologies for Healthcare*, Feb. 2008.

[86] C.-C. Lee, P.-S. Chung, and M.-S. Hwang, “A Survey on Attribute-based Encryption Schemes of Access Control in Cloud Environments,” *International Journal of Network Security*, vol. 15, pp. 231–240, 2013.

[87] H. Lee *et al.*, “A User-friendly Authentication Solution using NFC Card Emulation on Android,” in *Proc. IEEE Int. Conf. Service-Oriented Computing and Applications*, 2014.

[88] A. Lewko and other, “Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption,” in *Advances in Cryptogrpahy: EUROCRYPT*, Sep. 2010, pp. 62–91.

[89] A. Lewko, A. Sahai, and B. Waters, “Revocation Systems with Very Small Private Keys,” in *Proc. IEEE Symp. Security and Privacy*, May 2010.

[90] M. Li *et al.*, “Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption,” *IEEE Trans. Parallel and Distributed Systems*, vol. 24, pp. 131 – 143, 2012.

[91] C. Liu *et al.*, “A Survey of Attribute-based Access Control with User Revocation in Cloud Data Storage,” *International Journal of Network Security*, vol. 18, pp. 900–916, 2016.

[92] Y. Liu *et al.*, “Secure and fine-grained access control on e-healthcare records in mobile cloud computing,” *Elseiver Future Generation Computer Systems*, vol. 78, pp. 3:1020–3:1026, 2014.

[93] J. Liua, X. Huanga, and J. K. Liuc, “Secure sharing of Personal Health Records in cloud computing: Ciphertext-Policy Attribute-Based Signcryption,” *Elseiver Future Generation Computer Systems*, vol. 52, pp. 67–76, 2014.

[94] A. Lotito and D. Mazzocchi, “OPEN-SNEP Project enabling P2P over NFC using NPP and SNEP,” in *Proc. IEEE Int. Conf. Near Field Communication*, Feb. 2013.

[95] Z. Lv *et al.*, “iCare: A mobile Health Monitoring System for the Elderly,” in *Proc. IEEE Int. Conf. Green Computing and Communications*, 2010.

[96] G. Madlmayr *et al.*, “NFC Devices: Security and Privacy,” in *Proc. IEEE Int. Conf. Availability, Reliability and Security*, Mar. 2008.

[97] A. Majumder *et al.*, “Pay-Cloak: A Biometric Back Cover for Smartphones: Facilitating secure contactless payments and identity virtualization at low cost to end users,” *IEEE Consum. Elect. Mag.*, vol. 6, no. 2, pp. 78–88, Apr. 2017.

[98] A. Marcus *et al.*, “Using NFC-enabled Mobile Phones for Public Health in Developing Countries,” in *Proc. ACM Int. Work. Near Field Communication*, Feb. 2009, pp. 30–35.

[99] “MedicAlert,” n.d, Last Accessed on Jan 2019. [Online]. Available: www.medicalert.org/E-Health

[100] A. Merlo, L. Lorrai, and L. Verderame, “Efficient Trusted Host-based Card Emulation on TEE-enabled Android Services,” in *Proc Int. Conf. Trust and Trustworthy Computing*, Jul. 2016.

[101] “Microsoft Health Vault,” n.d, Last Accessed on Jan 2019. [Online]. Available: http://www.healthvault.com/

[102] “Mirth Connect,” n.d, Last Accessed on Jan 2019. [Online]. Available: http://www.mirth.com/Products-and-Services/Mirth-Connect

[103] J. Modi *et al.*, “A Secure Communication Model for Expressive Access Control Using CP-ABE,” *International Journal of Network Security*, vol. 19, pp. 193–204, 2017.

[104] L. M. V. Monne, “Credential remote management,” 2010, last accessed on Jan. 2019. [Online]. Available: http://www.cse.hut.fi/en/publications/B/11/papers/villalba.pdf

[105] D. M. Monteiro, J. J. P. C. Rodrigues, and J. Lloret, “A Secure NFC Application for Credit Transfer Among Mobile Phones,” in *Proc. IEEE Int. Conf. Computer, Information and Telecommunication Systems*, Jun. 2012.

[106] F. Morgner1 *et al.*, “Mobile smart card reader using NFC-enabled smartphones,” *Security and Privacy in Mobile Information and Communication Systems, Springer Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, vol. 107, pp. 24–37, 2012.

[107] S. Narayan, M. Gagne, and R. Safavi-Naini, “Privacy preserving EHR system using attribute-based infrastructure,” in *Proc. ACM Work. on Cloud computing security workshop*, 2010, pp. 47–52.

[108] “Understanding the New NHS, A guide for everyone working and training within the NHS,” https://www.nhs.uk/NHSEngland/thenhs/about/Documents/simple-nhs-guide.pdf, n.d, Last accessed on Jan 2019.

[109] D. Nelson, M. Qiao, and A. Carpenter, “Security of the near field communication protocol: an overview,” *Journal of Computing Sciences in Colleges*, vol. 29, no. 2, pp. 94–104, Dec. 2013.

[110] “NFCIP-1 Security Services and Protocol Cryptography Standard using ECDH and AES white paper,” 2008, Last accessed on Jan 2019. [Online]. Available: http://www.ecma-international.org/activities/Communications/tc47-2008-089.pdf

[111] “NHS National Services Scotland: Emergency Care Summary,” https://nhsnss.org/services/, n.d, Last Accessed on Jan 2019.

[112] G. Noordende, “A security analysis of the Dutch electronic patient record system,” University of Amsterdam, Amsterdam, technical report UVA-SNE-2010-01, 2010.

[113] V. Odelu, A. K. Das, and A. Goswami, “SEAP: Secure and Efficient Authentication Protocol for NFC Applications Using Pseudonyms,” *IEEE Trans. Consum. Electron.*, vol. 62, no. 1, pp. 30–38, Apr. 2016.

[114] “OpenMrs,” n.d, Last Accessed on Jan 2019. [Online]. Available: openmrs.org/

[115] L. Pang, J. Yang, and Z. Jiang, “A Survey of Research Progress and Development Tendency of Attribute-Based Encryption,” *Hindwai The Scientific World Journal*, vol. 2014, 2014.

[116] M. Pasquet and S. Gerbaix, “Fraud on Host Card Emulation Architecture,” in *Proc. IEEE Int. Conf. Mobile and Secure Services*, Feb. 2016.

[117] M.-P. Pelletier, M. Trepanier, and C. Morency, “Smart card data use in public transit: A literature review,” *Elseiver Transportation Research Part C Emerging Technologies*, vol. 19, no. 4, pp. 557–568, Aug. 2011.

[118] “Patient-Generated Health Data and Health IT,” n.d, Last Accessed on Jan 2019. [Online]. Available: https://www.healthit.gov/policy-researchers-implementers/patient-generated-health-data

[119] “PHR4us: Personal Health Records for Us,” n.d, Last Accessed on Jan 2019. [Online]. Available: https://phr4us.com/put-ice-on-it/

[120] R. S. Pippal, J. C. D., and S. Tapaswi, “Security Issues in Smart Card Authentication Scheme,” *International Journal of Computer Theory and Engineering*, vol. 4, no. 2, pp. 206–2011, Apr. 2012.

[121] E. Poll, “Trusted Execution Environments (TEEs) & Trusted Computing,” 2010, Last accessed on Jan. 2019. [Online]. Available: https://www.cs.ru.nl/E.Poll/hw/slides/TEE.pdf

[122] “Rashtriya Bima Yojana,” n.d, Last Accessed on Jan 2019. [Online]. Available: http://en.wikipedia.org/wiki/Rashtriya_Swasthya_Bima_Yojan

[123] M. Reveilhac and M. Pasquet, “Promising Secure Element Alternatives for NFC Technology,” in *Proc. IEEE Int. Work. Near Field Communication*, Feb. 2009.

[124] “A Robust Health Data Infrastructure, JSR-13-700,” 2014, Last accessed on Jan 2019. [Online]. Available: https://www.healthit.gov/sites/default/files/ptp13-700hhs_white.pdf

[125] M. Roland, “Software card emulation in NFC-enabled mobile phones:great advantage or security nightmare?” in *Proc. Int. Work. Security and Privacy in Spontaneous Interaction and Mobile Phone Use*, 2012.

[126] M. Roland and J. Langer, “Digital Signature Records for the NFC Data Exchange Format,” in *Proc. IEEE Int. Work. Near Field Communication*, Apr. 2010, pp. 71–76.

[127] ——, “Comparison of the usability and security of NFC’s different operating modes in mobile devices,” *Springer J. Elektrotech. Inftech.*, vol. 30, pp. 201–206, 2013.

[128] M. Roland, J. Langer, and J. Scharinger, “Practical Attack Scenarios on Secure Element-Enabled Mobile Devices,” in *Proc. IEEE Int. Work. Near Field Communication*, Mar. 2012.

[129] ——, “Applying Relay Attacks to Google Wallet,” in *Proc. IEEE Int. Cconf. Near Field Communication*, Feb. 2013.

[130] V. O. Rybynok *et al.*, “MyCare Card Development: Portable GUI Framework for the Personal Electronic Health Record Device,” *IEEE Trans. Infor. Tech. In Biomed.*, vol. 15, pp. 66 – 73, 2011.

[131] P. Samarati and S. C. de Vimercati, “Access Control: Policies, Models, and Mechanisms,” in *Foundations of Security Analysis and Design, Lecture Notes in Computer Science*, vol. 2171, Oct. 2001, pp. 137–196.

[132] D. Sethia *et al.*, “Mutual Authentication Protocol For Secure NFC Based Mobile Healthcard,” *IADIS International Journal on Computer Science and Information Systems*, vol. 11, no. 2, pp. 195–202, 2016.

[133] ——, "Technical report for implementation of secure NFC-based IoT prototype using mobile devices," n.d, Last Accessed: Aug. 2018. [Online]. Available: https://sites.google.com/site/divyashikhasethia/home/secure-nfc-based-iot-access/IoTPrototypeTechReport.pdf

[134] C. Shepherd *et al.*, "Secure and Trusted Execution: Past, Present and Future-A Critical Review in the Context of the Internet of Things and Cyber-Physical Systems," in *Proc. IEEE Int. Conf. Trustcom*, Aug. 2016.

[135] S. Sheron and D. Sethia, "Technical project report: Simulation of patient flow management using portable health records," 2017, Last accessed on Jan. 2019. [Online]. Available: https://sites.google.com/site/divyashikhasethia/home/portable-mobile-based-secure-healthcard/techreport-Simulation.pdf

[136] "Secure Element Deployment and Host Card Emulation v10," n.d, Last Accessed on Jan 2019. [Online]. Available: http://simalliance.org/wp-content/uploads/2015/03/Secure-Element-Deployment-Host-Card-Emulation-v1.0.pdf

[137] E. G. Spanakis *et al.*, "MyHealthAvatar personalized and empowerment health services through Internet of Things technologies," in *Wireless Mobile Communication and Healthcare (Mobihealth)*, 2014, pp. 331–334.

[138] L. Sprague, "Personal health records: the people's choice?" Issue Brief George Wash Univ Natl Health Policy Forum, Nov. 2006.

[139] "Trusted Computing Group," n.d, Last accessed on Jan. 2019. [Online]. Available: https://www.trustedcomputinggroup.org/

[140] C. Thammarat *et al.*, "A secure lightweight protocol for NFC communications with mutual authentication based on limited-use of session keys cites," in *Proc. IEEE Int. Conf. Information Networking*, Jan. 2015.

[141] Y. Tian, Y. Peng, G. Gao, and X. Peng, “Role-based Access Control for Body Area Ntworks Using Attribute-based Encryption in Cloud Storage,” *International Journal of Network Security*, vol. 19, pp. 720–726, 2017.

[142] R. Toegl and M. Hutter, “An approach to introducing locality in remote attestation using near field communications,” *Springer The Journal of Supercomputing*, vol. 55, pp. 207 – 227, 2011.

[143] Y. Tong *et al.*, “Cloud-Assisted Mobile-Access of Health Data with Privacy and Auditability,” *IEEE Journal of Biomedical and Health Informatics*, vol. 18, pp. 419–429, 2013.

[144] A. Umar, K. Mayes, and K. Markantonakis, “Performance variation in host-based card emulation compared to a hardware security element,” in *Proc. IEEE Int. Conf. Mobile and Secure Services*, Feb. 2015.

[145] P. Urien, “LLCPS A new security framework based on TLS for NFC P2P applications in the Internet of Things,” in *Proc. IEEE Int. Conf. Consumer Communications and Networking*, Jan. 2013.

[146] M. Vazquez-Briseno *et al.*, “Using RFID/NFC and QR-Code in Mobile Phones to Link the Physical and the Digital World,” *Journal Interactive Multimedia*, pp. 219–242, Mar. 2012.

[147] M. Vergara *et al.*, “Mobile Prescription: An NFC-Based Proposal for AAL,” in *Proc. IEEE Int. Work. on Near Field Communication*, Apr. 2010.

[148] ——, “Using NFC-enabled Mobile Phones for Public Health in Developing Countries,” in *Proc. IEEE Int. Work. Near Field Communication*, Apr. 2010.

[149] C. Wang, X. Liu, and W. Li, “Implementing a Personal Health Record Cloud Platform Using Ciphertext-Policy Attribute-Based Encryption,” in *Proc. IEEE Int. Conf. Intelligent Networking and Collaborative Systems*, Sep. 2012.

[150] B. Waters, "Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions," in *Proc. Springer Advances in Cryptology*, 2009, pp. 619–636.

[151] ——, "Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization," in *Proc. Int. Conf. Public Key Cryptography*, Oct. 2010, pp. 53–70.

[152] "HCE security implications," n.d, Last Accessed: Dec. 2018. [Online]. Available: https://newscience.ul.com/wp-content/uploads/2014/07/hce_security_implications.pdf

[153] "A Smart Card Alliance Mobile And NFC Council White Paper, Host Card Emulation (HCE) 101," 2014, Last accessed on 14-08-2018. [Online]. Available: https://www.securetechalliance.org/wp-content/uploads/HCE-101-WP-FINAL-081114-clean.pdf

[154] S. Y. Worcester, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *Proc. ACM Symp. Information, Computer and Communications Security*, 2010, pp. 261–270.

[155] A. Wright and D. Sittig, "Encryption characteristics of two USB-based personal health record devices," *Journal of the American Medical Informatics Association*, vol. 14, 2007.

[156] K. Wuyts *et al.*, "What electronic health records don't know just yet. A privacy analysis for patient communities and health records interaction," *Springer Health and Technology*, vol. 2, pp. 159–183, 2012.

[157] F. Xhafa *et al.*, "Designing cloud-based electronic health record system with attribute-based encryption," *Springer Multimedia Tools and Applications*, vol. 74, pp. 3441–3458, 2015.

[158] J. Xu, K. Xue, Q. Yang, and P. Hong, "PSAP: Pseudonym-Based Secure Authentication Protocol for NFC Applications," *IEEE Trans. Consum. Electron.*, vol. 64, pp. 83–91, Feb. 2018.

[159] B. Yang *et al.*, "DAA-TZ: An Efficient DAA Scheme for Mobile Devices using ARM Trustzone," in *Proc. Int. Conf. on Trust and Trustworthy Computing*, Aug. 2015.

[160] G. Yang *et al.*, “A Health-IoT Platform Based on the Integration of Intelligent Packaging, Unobtrusive Bio-Sensor, and Intelligent Medicine Box,” *IEEE Trans. Industrial Informatics*, vol. 10, pp. 2180 – 2191, 2014.

[161] L. Yang, J.-F. Ma, and Q. Jiang, “Mutual Authentication Scheme with SmartCards and Password under Trusted Computing,” *International Journal of Network Security*, vol. 14, pp. 155–162, 2012.

[162] W. Yang *et al.*, “Securing Mobile Healthcare Data: A Smart Card Based Cancelable Finger-Vein Bio-Cryptosystem,” *IEEE Access*, vol. 6, pp. 36 939–36 947, Jun. 2018.

[163] Y. Yang, X. Han, F. Bao, and R. Deng, “A smart-card-enabled privacy preserving E-prescription system,” *IEEE Trans. Information Technology in Biomedicine*, vol. 8, pp. 47–58, 2004.

[164] B. Yuksel *et al.*, “Research issues for privacy and security of electronic health services,” *Elseiver Future Generation Computer Systems*, vol. 68, pp. 1–13, 2017.

ANNEXURE-

AVISPA SCRIPT for NSE-AA protocol verification

```
%%%%%%%% AVISP Script%%%%%%%%%%%%%%%5%%%%%%%%%%
%%%% Paper: # Divyashikha Sethia, Daya Gupta and Huzur Saran,"  NFC Secure Element-
based Mutual Authentication and Attestation for IoT access", IEEE Transaction on Consu
( vol 64 no 4), 2018
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% User Device SE %%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role role_U( U:agent,D:agent,S:agent,M:text,H,PRF,KDF:hash_func,V:text,IdAu:text,
Kca,KaikU: public_key, Kud,Kus:symmetric_key,SND_UD,RCV_UD:channel(dy))
played_by U
def=
local
State:nat,Nu,Nd:text,Ks:symmetric_key, SMLu, SMLd:text, PCRu,PCRd:text, TPMInfoU,
TPMInfoD:text, SigTPMu, SigTPMd:text, Qu, QmacU, Qd, QmacD:text,
CertAikU: {agent.public_key.text}_inv(public_key),
% certificates for the private key inv(KaikD)
CertAikD:{agent.public_key.text}_inv(public_key),KaikSetU,KaikSetD:public_key set,
KcaSet:public_key set, KaikD: public_key, O,Q,Z,B,Pwu,Pwbu,Pwbd,Ru,Cu,Du,Bu,T1,T2,
T4:text, IdVu,IdVd,IdAikU,IdAikD: text, M2,M6,M10: text
init State := 0
transition
% Receive start and send M2
```

% Mut Auth Step 2-3 IoT-SE <- User-SE

```
State=0 /\ RCV_UD(start) =|>
```

```
23. State':=6 /\ Nu':=new() /\ B':= new() /\ Pwu':=new() /\ Pwbu' := H(Pwu',B') /\
24. Ru':= H(IdAu.Kus) /\ Cu':= H(Pwbu'.Nu'.Ru') /\ Du':= xor(H(U.Pwbu'),Nu') /\
25. Bu':={IdAu.Pwbu'.Nu'}_Kus /\ secret(IdAu,sec_ua,{U,S}) /\ IdVu':=Cu'.Du'.Bu'
26. /\ O':={Nu'}_Kus /\ Q':=H(Nu'.Kud) /\ T1':= xor(Nu',Kud) /\
27. M2':=IdVu'.T1'.{O'}_Kud.Q' /\ SND_UD(M2')

  28. % Mut Auth Step 10-11 User-SE -> IoT-SE 29. % Receive M5 and send M6

30. State=6 /\RCV_UD(T2'.T4'.H(T2.H(Pwbu'.Nd'.Nu'.Kus).Kud)) =|>
31. %% A checks that he receives the same nonce that he sent at step 1.
32. % User-SE checks that he receives the same nonce Nu and Kus that he used in mesg 2
33. %of step 2
34. % User-SE authentictes itself to IoT-SE through Z:H(Nd'.Kud.Ks')
35. % User authenticate IoT over Y': H(T2.H(Pwbu'.Nd'.Nu'.Kus).Kud))
36. State':=8  /\ Nd':= xor(T2',Kud) /\ Pwbd':= xor(T4',Kus) /\
37. request(U,D,auth_1,H(T2.H(Pwbu'.Nd'.Nu'.Kus).Kud)) /\ Ks':= KDF(Pwbu.Pwbd'.Nu.Nd'.
38. Z':= H(Nd'.Kud.Ks') /\ M6':= Nd'.Z' /\ SND_UD(M6') /\ witness(U,D,auth_2,Z') /\
39. secret(Ks',sec_ks,{U,D})

  40. % Mut Auth Step 12
41. % Receives data using session key

42. State=8 /\ RCV_UD(M'.H(M'.Ks)) =|>

  43. % Attestation
44. % Mut Attestation Step 2 User sends attestation report to IoT Device

45. State':=10 /\ SMLu':=new() /\ PCRu':=new() /\ KaikU' :=new() /\ KaikSetU' :=new()
46. IdAikU':=PRF(IdVu.Nu) /\ CertAikU':={IdAikU'.KaikU'.KaikSetU'}_inv(Kca) /\
47. TPMInfoU':= new() /\ SigTPMu':= {SMLu'.PCRu'.Nd.TPMInfoU'}_inv(KaikU') /\
48. Qu':=SigTPMu'.TPMInfoU'.{CertAikU'}_Ks /\
QmacU':= PRF(Qu'.Ks) /\ M10':= Qu'.QmacU' /\ SND_UD(U.M10') /\
secret(PCRu',sec_pcru,{U,D}) /\ secret(SMLu',sec_smlu,{U,D}) /\
witness(U,D,smlu_verify,SMLu')
% %
```

% Mut Attestation Step 2 User received attestation report from the IoT Device

```
State= 10/\RCV_UD(D.SigTPMd'.TPMInfoD'.{CertAikD'}_Ks.
PRF(SigTPMu'.TPMInfoU'.{CertAikD'}_Ks.Ks))
/\ CertAikD'={PRF(IdVd'.Nd').KaikD'.KaikSetD'}_inv(Kca)/\
SigTPMd'={SMLd'.PCRd'.Nu.TPMInfoD'}_inv(KaikD')
/\ in(Kca,KcaSet) =|>
State' := 12 /\ request(U,D,smld_verify,SMLd')


end role


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Trusted Certified Authority %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role role_S (U, D, S : agent, Kus,Kds: symmetric_key, H: hash_func, IdAu,IdAd:text
T3,T4:text,SND_SD, RCV_SD : channel(dy))
played_by S
def=
local IdU,IdD,Nu,Nd,W,X,Y,R,Cd,Dd,Bd,Cu,Du,Bu,Idu,Pwbu,Idd,Pwbd,Rd,Ru:text,
State : nat, M4:text
init State := 11
transition
```

% Mut Auth Step 6-7

```
State=11 /\ RCV_SD(Cu’.Du’.{IdAu’.Pwbu’.Nu’}_Kus.Cd’.Dd’.{IdAd’.Pwbd’.Nd’}_Kds.
{Nd’}_Kds.{Nu’}_Kus) =|>
% Verify the virtual identities  IoT-SE <- Server
State’ := 13  /\ Nu’:= xor(H(IdAu’.Pwbu’),Du’) /\ Nd’:= xor(H(IdAd’.Pwbd’),Dd’) /\
Ru’:=H(IdAu’.Kus) /\ Rd’:=H(IdAd’.Kds) /\ R’:=H(Pwbu’.Nu’.Nd’.Kus) /\
X’:= H(Pwbd’.Nu’.Nd’.Kds) /\ T3’:=xor(Pwbu’,Kds) /\ T4’:=xor(Pwbd’,Kus) /\
M4’:=T3’.T4’.X’.R’ /\ SND_SD(M4’)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% IoT Device SE %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role role_D(D:agent,U:agent,S:agent,M:text,H,PRF,KDF:hash_func,V:text,IdAd:text,
Kca,KaikD:public_key, Kud,Kds:symmetric_key,SND_DU,RCV_DU,SND_DS,RCV_DS:
channel(dy))
played_by D
def=
local State:nat,Nu,Nd,O,Y,W,R,T1,T2,T3,T4:text,Ks:symmetric_key,SMLu,SMLd:text,
PCRu,PCRd:text,TPMInfoU,TPMInfoD:text,SigTPMu,SigTPMd:text,Qu,QmacU,Qd,
QmacD:text,
CertAikU:{agent.public_key.text}_inv(public_key),
% certificates for the private key inv(KaikD)
CertAikD:{agent.public_key.text}_inv(public_key),KaikSetU,KaikSetD:public_key set,
KcaSet:public_key set,KaikU: public_key, M2,IdVu,B,Pwd,Pwbd,Pwbu,Rd,Cd,Dd,Bd:text,
IdVd,IdAikU,IdAikD: text,M3,M5,M10:text
init State := 1
transition
% Mut Auth Step 4-5 IoT-SE -> Server
```

```
100. 1. State=1   /\ RCV_DU(IdVu'.T1'.{O'}_Kud.H(Nu'.Kud)) =|>
101. State':=3 /\ Nu':=xor(T1,Kud) /\ Nd':=new() /\ B':= new() /\ Pwd':=new() /\
102. Pwbd' := H(Pwd',B') /\ Rd':= H(IdAd.Kds) /\ Cd':= H(Pwbd'.Nd'.Rd') /\
103. Dd':= xor(H(IdAd.Pwbd'),Nd') /\ Bd':={IdAd.Pwbd'.Nd'}_Kds /\
104. secret(IdAd,sec_da,{D,S})
105. % Virtual Identity
106. /\ IdVd':=Cd'.Dd'.Bd' /\ M3':= IdVu'.IdVd'.{Nd'}_Kds.O' /\ SND_DS(M3')
```

107. % Mut Auth Step 8-9 IoT-SE -> User-SE IoT-SE authenticates itself to 108. %User-SE through Y.

```
109. 2.State = 3  /\ RCV_DS(T3'.T4'.H(Pwbu'.Nu'.Nd'.Kds).R') =|>


110. % IoT requests to Authenticate by User over Y'
111. State':= 5 /\ Pwbu':=xor(T3',Kds) /\ Ks':=KDF(Pwbu'.Pwbd.Nu'.Nd'.Kud) /\
112. T2':= xor(Nd,Kud) /\ Y':=H(T2.R'.Kud) /\ M5':=T2'.T4'.Y'/\ SND_DU(M5') /\
113. witness(D,U,auth_1,Y')
```

114. %Mut Auth Step 12-13 IoT-SE authenticates the User-SE through H(Nd'.Kud.Ks')

```
115. State=5 /\ RCV_DU(Nd'.H(Nd'.Kud.Ks')) =|>
116. State':=7 /\ request(D,U,auth_2,H(Nd'.Kud.Ks')) /\ SND_DU(Nd.H(Nd.Ks'))
```

117. % Attestation 118. %Mut Attes Step 2 IoT device receives attesation report from the User device

```
119. State=7 /\ RCV_DU(U.SigTPMu'.TPMInfoU'.{CertAikU'}_Ks.
120. PRF(SigTPMu'.TPMInfoU'.{CertAikU'}_Ks.Ks)) /\
121. CertAikU'={PRF(IdVu'.Nu').KaikU'.KaikSetU'}_inv(Kca) /\
122. SigTPMu'={SMLu'.PCRu'.Nd.TPMInfoU'}_inv(KaikU') /\ in(Kca,KcaSet) =|>
```

123. %Mut Attes Step 2 IoT device sends attestation report to the user device

```
State' := 9  /\ request(D,U,smlu_verify,SMLu') /\ SMLd':= new() /\ PCRd':= new()
/\ KaikD':= new() /\ KaikSetD':= new() /\ IdAikD':=PRF(IdVd.Nd) /\
CertAikD':={IdVd.KaikD'.KaikSetD'}_inv(Kca) /\ TPMInfoD':= new() /\
SigTPMd':={SMLd'.PCRd'.Nu.SigTPMd'}_inv(KaikD') /\
Qd':= SigTPMd'.TPMInfoD'.{CertAikD'}_Ks /\ QmacD':= PRF(Qd'.Ks) /\
M10':=Qd'.QmacD' /\
SND_DU(D.M10') /\ secret(SMLd',sec_smld,{U,D}) /\
secret(PCRd',sec_pcrd,{U,D}) /\ witness(U,D,smld_verify,SMLd')
end role

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Session
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
role session(U:agent,D:agent,S:agent,M:text,T1,T2:text,Kud,Kus,Kds:symmetric_key,
H,PRF,KDF:hash_func,V:text,IdAu,IdAd:text,Kca,KaikU,KaikD: public_key)
def=
local SND_US,RCV_US,SND_SU,RCV_SU,SND_UD,RCV_UD,SND_DU,RCV_DU,SND_DS,
RCV_DS,SND_SD,
RCV_SD:channel(dy)
composition role_D(D,U,S,M,H,PRF,KDF,V,IdAd,Kca,KaikU,Kud,Kds,SND_DU,RCV_DU,
SND_DS,RCV_DS)
/\ role_U(U,D,S,M,H,PRF,KDF,V,IdAu,Kca,KaikD,Kud,Kus,SND_UD,RCV_UD)
/\ role_S(U, D, S, Kus,Kds, H, IdAu,IdAd,T1,T2,SND_SD,RCV_SD)
end role
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% Environment
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

role environment()
def=
const kus,kds,kis,kud,kui,kid:symmetric_key, user,iot,server:agent, s1:text, auth
auth_2,sec_ks,sec_ua,sec_da,sec_smlu,sec_smld,sec_m1,sec_pcrd,sec_pcru:protocol_i
smlu_verify, smld_verify: protocol_id, h,prf,keygen:hash_func, v:text, idua,idda,
idia:text,t1,t2:text, kca,kuiku,kuikd,ki:public_key
intruder_knowledge = {user,iot,kca, kuiku, kuikd, ki, inv(ki), {i.ki}_(inv(kca))
composition
session(user,iot,server,s1,t1,t2,kud,kus,kds,h,prf,keygen,v,idua,idda,kca,kuiku,
kuikd)
/\session(user,i,  server,s1,t1,t2,kui,kus,kis,h,prf,keygen,v,idua,idia,kca,kuiku
/\session(i,iot,server,s1,t1,t2,kid,kis,kds,h,prf,keygen,v,idia,idda,kca,ki,kuikd
end role

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Goal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
goal
secrecy_of sec_ks
secrecy_of sec_ua
secrecy_of sec_da
authentication_on auth_1
authentication_on auth_2
secrecy_of sec_smlu
secrecy_of sec_smld
secrecy_of sec_pcru
secrecy_of sec_pcrd
authentication_on smlu_verify
authentication_on smld_verify
end goal
environment()
```