Incorporating Security into Electronic Health Records Based Healthcare Systems with Wireless Sensor Networks

A dissertation of an investigation submitted in fulfilment
for the award of Doctor of Philosophy (DPHD),
at the University of Southern Queensland

A dissertation is submitted by
Mishall Al-Zubaidie

February 2020

Abstract

The potential for applying electronic health record (EHR), electronic medical record (EMR) and healthcare wireless sensor network (HWSN) in the healthcare (HC) industry is tremendous and boundless. However, the security and privacy issues of HC data must be addressed with great caution and as must accessibility. EHR system security and privacy are related to the confidentiality of protected health information (PHI), the integrity of EMR data and the authentication/authorisation of users. Furthermore, EHR data is collected and communicated between different users and patients' health data is shared within a system.

Working towards a solution, we have designed a secure and efficient HC application that integrates wireless sensor network (WSN) technology with EMR/EHR technology. First, the general architecture of the HC application system is proposed and then an EMR/EHR repository is described. The novelties of our approach include the introduction of the WSN's application to automatically collect patients' physiological information/data and securely store them as EMR/EHR records in the repository, making the EMR/EHR system more efficient. Second, a number of efficient security technologies including authentication and users' authorisation, and security protocols such as Elliptic Curve Cryptography (ECC), eXtensible Access Control Markup Language (XACML), have been adopted, modified or designed in our proposed HC application. Thus, security of HC application has been significantly improved and, as a consequence, the patients' privacy has been addressed.

Throughout this study, we have deepened our understanding of the security requirements in HC applications and appreciated the important role played by the latest wireless networking and sensing technology in achieving the security objectives in the modern HC industry. The results of this study include a framework for building secure and efficient HC applications, accompanied by a set of protocols which enable the auto-collection and secure transmission of patients' health and medical information. In addition, we offer two schemes: an authentication scheme for protecting users' identities and privacy and an authorisation scheme for user's differentiated access control (AC) or privileges to the patients' health and/or medical records. These schemes have been theoretically verified.

Certification of Thesis

This thesis is entirely the work of Mishall Al-Zubaidie except where otherwise acknowledged. The work is original and has not previously been submitted for any other award, except where acknowledged.

Signed ______________
Mishall Al-Zubaidie
Date ______________

Signed ______________
Principle Supervisor
Dr. Zhongwei Zhang
Date ______________

Signed ______________
Associate Supervisor
Associate Professor Ji Zhang
Date ______________

Acknowledgements

I would like to acknowledge the support of many beloved parties.

First, I would like to thank Almighty God for his protection, inspiration, and guidance, and for giving me the strength, sanity and wellness necessary to conduct my study.

Second, I would like to thank Thi-Qar University and the Iraqi Ministry of Higher Education and Scientific Research which have given me the opportunity to complete my doctoral study.

Third, I would like to thank the supervision team of Dr. Zhongwei Zhang and Dr. Ji Zhang. During my study, they offered me advice, patience, motivation, support, and the help needed to overcome obstacles. Their critical assessment has been a major factor in my PhD dream coming true.

Fourth, I would like to thank Dr. Barbara Harmes for her continuous revision of of research and thesis chapters. I also would like to express my appreciation and gratitude to Sandra for proofreading my thesis.

Fifth, I would like to express my appreciation for the support of the University of Southern Queensland, the Faculty of Health, Engineering and Sciences, and the Australian Commonwealth Government through the Research Training Program (RTP) Fees Offset scheme. I am grateful to them for enlightening and supporting me during the research period.

Finally, I would like to acknowledge all the support and encouragement given by my family and friends. Each of you has contributed in your own special way and helped me during difficult times. The work presented in this thesis would not have been possible if it were not for the help, support, guidance, and friendship of a number of individuals. For those who are not specifically mentioned, I also thank you.

Dedication

I would like to dedicate this thesis:

To my dear father ...
Hammed Al-Zubaidie who planted in my heart the love of knowledge and the shining light that taught me the meaning of sacrifice and fulfillment.

To my dear mother ...
Eeda Mutar, Paradise in this world and the Hereafter, my love and all my being, the symbol of purity and chastity, the beacon of peace and tenderness, the truth of eternal fulfillment, the inexhaustible river of tenderness.

To my dear wife ...
Hawa Bahedh, a city of roses and a sincere passion for the taste of honey, who helped me through my difficult years of study, and gave me love, respect, attention, trust and patience.

To my dear friend ...
Loretta Gomez, who eased the difficulties of alienation, removed the darkness, and planted a smile on my face.

To my dear brothers and sisters ...
Who taught me the meaning of virtuous life and true brotherhood, who accompanied me and helped me in all the different forums of science.

To my dear country ...
Great Iraq, the country of civilizations of science and knowledge, my nostalgia for all my moments, who helped me in my studies.

To all my teachers, to all who helped me, to every reader of this thesis, who were the inspiration in achieving my dream. To all of them, I dedicate my humble thesis.

Table of Contents

Abstract i

Certification of Thesis ii

Acknowledgements iii

Dedication iv

List of Figures x

List of Tables xiii

List of Abbreviations xiv

List of Publications xviii

Chapter 1 – Introduction 1

1.1 Overview of HC in Some Developed Countries 1
1.2 Information and Communications Technology 2
1.3 Security Concerns in HC Services 2
1.3.1 Encryption for Preserving Users' Privacy 3
1.3.2 Signature for Users' Authenticity 3
1.4 Significance of the Project 4
1.5 Research Objectives and Questions 4
1.5.1 Research Objectives 4
1.5.2 Specific Research Questions 5
1.6 Organisation of this Thesis 6
1.7 Summary of the Chapter 7

Chapter 2 – Literature Review 8

2.1 Healthcare Standards 8
2.1.1 On the Electronic Medical Record 9
2.1.2 On the Electronic Health Record 10
2.1.3 Pros and Cons of Electronic Medical/Health Record 11

2.1.4 Integration of EMR and EHR 12
2.2 Acquiring EMR/EHR in HC Applications 13
2.2.1 Main Applications of HWSN 13
2.2.2 Security and Privacy in HWSN 14
2.3 Security of HC Applications 16
2.3.1 Specific Security Concerns of HC Services 16
2.3.2 Possible Attacks on HC Applications 17
2.3.3 Security Requirements of HC Applications 18
2.3.4 Security Protocols in HC Applications 20
2.4 Some Studies Related to our Research 23
2.4.1 Storage of User Data 24
2.4.2 User Authentication to HC Applications 26
2.4.3 User Authorisation to HC Application 30
2.5 Summary of the Chapter 33

Chapter 3 – Designing a Secure and WSN Based Healthcare System Application **34**

3.1 General Architecture of Proposed HC Application 34
3.2 EMR/EHR Repository 35
3.3 Storage Scheme in the Proposed HC Application 38
3.3.1 Threat to Storage Scheme 38
3.3.2 SHA Hash Function 39
3.3.3 BLAKE Hash Function 39
3.3.4 De-identification Mechanism 40
3.3.5 Efficient HWSN Data Management Using XML 41
3.3.6 Homomorphic Scheme 41
3.4 Elliptic Curve Cryptography for Authentication 42
3.4.1 Threat to Authentication Scheme 42
3.4.2 Elliptic Curve Integrated Encryption Scheme (ECIES) 43
3.4.3 Lightweight Hash-Function Algorithm 43
3.4.4 One Time Password (OTP) 44
3.4.5 Mutual Authentication 44
3.4.6 Media Access Control (MAC) Address 45
3.5 XACML Techniques for our Authorisation Scheme 46
3.5.1 Threat to Authorisation Scheme 46
3.5.2 Elliptic Curve Digital Signature Algorithm (ECDSA) 47
3.5.3 Models of Access Control to the EHR Repository 47
3.5.4 Distributed AC Implementation Technology 48
3.5.5 Shamir Scheme 50

3.6 Development of EMR/EHR System 51
3.6.1 Patient's Confidence in HC Services 51
3.6.2 HC and EMR/EHR Users 51
3.6.3 Administration/Management of Health Organisations 52
3.7 Integrating WSN with HC Application 52
3.8 Protocols to Improve the Security of WSN and EMR 53
3.8.1 HC Data Management and the EMR's Storing/Exchanging . . 55
3.8.2 Integrity and Authentication of EMR 55
3.9 Users' Authentication in EHR 56
3.9.1 Information Confidentiality in EHR 56
3.9.2 Ensure User and Device Authenticity 57
3.10 Privacy of Users' Authorisation in EHR 58
3.10.1 Access Control with EHR Datasets 58
3.10.2 Using Pseudonym and Anonymity with EHR to Hide the Medical Records 59
3.11 Summary of the Chapter 60

Chapter 4 – A More Efficient and Secure EMR Storage and Repository **61**
4.1 Data Collection by HWSNs 61
4.1.1 A Reliable and Efficient Scheme for Data Collection 62
4.2 Our Proposed Data Storage Model 62
4.2.1 Network Model 62
4.2.2 Design Goals of REISCH 63
4.3 REISCH's Scheme 65
4.3.1 Entities Preparation 65
4.3.2 Integrity of EMR being Transmitted 65
4.3.3 Applying Camouflage Signature 66
4.3.4 Implementing Homomorphic 66
4.3.5 REISCH's Protocols 67
4.4 Summary of the Chapter 71

Chapter 5 – Robust Security to Authenticate Users Identity to the EHR Repository **72**
5.1 Information Security in EHR Systems 72
5.1.1 A Robust Model of Authentication for the Proposed HC Application 73
5.2 The Proposed Authentication Scheme 74
5.2.1 Network Model 74
5.2.2 Design Goals of RAMHU 75

5.2.3 Proposed Protocols for the Authentication Scheme 77
5.3 Summary of the Chapter 89

Chapter 6 – Authorisation of HC Users with Differentiated Access Control 90
6.1 Data Security in EHR Systems 90
6.2 Overview of Requirements of Access Control 91
6.2.1 Access Control for the EHR Repository 92
6.3 Our Proposed Authorisation Model 93
6.3.1 Users Access Control Model 93
6.3.2 Design Goals of PAX 94
6.3.3 Implementation of PAX 95
6.3.4 PAX Authorisation Protocols 100
6.4 Summary of the Chapter 109

Chapter 7 – Verification of Protocols for the Security and Performance 110
7.1 Security Testing Tool 110
7.1.1 AVISPA 110
7.2 Analysis of Storage Scheme 111
7.2.1 Possible Attacks on REISCH Scheme 111
7.2.2 Performance Analysis 120
7.3 Analysis of RAMHU Scheme 127
7.3.1 Possible Attacks on the RAMHU Scheme 127
7.3.2 Performance Analysis 141
7.4 Analysis of PAX Scheme 145
7.4.1 Direct and Indirect Users Scenarios in PAX 145
7.4.2 Possible Attacks on PAX Scheme 147
7.4.3 Performance Analysis 157
7.5 Summary of the Chapter 163

Chapter 8 – Conclusions and Future Directions 164
8.1 Conclusions of the Thesis 164
8.2 Future Directions 165
8.3 Summary of the Chapter 167

References 168

Appendices 188

Appendix A –REISCH 189

A.1 SHA Hash Function 189
A.2 BLAKE Hash Function 190
A.3 ECDSA with BLAKE Hash 192

Appendix B –RAMHU **194**
B.1 Elliptic Curve Integrated Encryption Scheme (ECIES) 194
B.2 Lightweight Hash-Function Algorithm 196

Appendix C –PAX **200**
C.1 Elliptic Curve Digital Signature Algorithm (ECDSA) 200

Appendix D –Security Test Tool **202**
D.1 AVISPA 202

List of Figures

2.1 Integration of EMRs with EHR repository 13
2.2 Architecture of healthcare based WSN (Zhang et al. 2014) . . . 14
2.3 An attack on information security (Intruder 1) and device security (Intruder 2) 15

3.1 General architecture of proposed HC application 35
3.2 The proposed project system 36
3.3 Information repositories 37
3.4 Data repository 38
3.5 Types of authentication schemes 45
3.6 Scheme of RBAC model 48
3.7 Scheme of ABAC model 49
3.8 Scheme of XACML 50
3.9 Taxonomy of HC users 52
3.10 The proposed project model in HC environment 54
3.11 Homomorphic and 1 scalar multiplication with signatures 56

4.1 General REISCH model 64
4.2 Camouflage signature 66
4.3 Data collection protocol 68
4.4 Data aggregation protocol 70
4.5 Data storage protocol 71

5.1 General network model 75
5.2 Registration and login protocol 78
5.3 Login protocol 79
5.4 NetworkAddress path in system registry 80
5.5 Authentication protocol 82
5.6 Password update protocol 85
5.7 Revocation protocol 88

6.1 PAX model 94
6.2 Authorisation of direct and indirect users 94
6.3 PAX policy 99

6.4 C_i's request 99
6.5 Authorisation of direct users 101
6.6 Protocol of PAX model between C_i and CS 102
6.7 Protocol of PAX model between CS and AS 103
6.8 Protocol of PAX model between AS and DS 104
6.9 Protocol of PAX model between AS, CS and C_i 106
6.10 Authorisation of indirect users 107
6.11 Protocol of PAX model for indirect users 108

7.1 REISCH's framework in AVISPA 114
7.2 SN_i role of REISCH in HLPSL 115
7.3 CH_i role of REISCH in HLPSL 115
7.4 LS role of REISCH in HLPSL 117
7.5 CS role of REISCH in HLPSL 118
7.6 Session, environment and goal roles of REISCH in HLPSL . . . 119
7.7 Simulation result of REISCH using OFMC backend 119
7.8 Simulation result of REISCH using CL-AtSe backend 119
7.9 Comparison of SHA and BLAKE2 with 1MB data 122
7.10 Execution time of ECDSA-SHA1 and ECDSA-BLAKE2bp with 1MB data 122
7.11 Minimum execution time of hash functions with 1MB data . . . 123
7.12 Maximum execution time of hash functions with 1MB data . . . 123
7.13 Average execution time of hash functions with 1MB data 123
7.14 Minimum execution time of ECDSA algorithms with 1MB data 124
7.15 Maximum execution time of ECDSA algorithms with 1MB data 124
7.16 Average execution time of ECDSA algorithms with 1MB data . 124
7.17 Comparison of alive SNs 125
7.18 RAMHU's framework in AVISPA 135
7.19 C_i role of RAMHU in HLPSL 135
7.20 CS role of RAMHU in HLPSL 137
7.21 AS role of RAMHU in HLPSL 137
7.22 Session, environment, and goal roles of RAMHU in HLPSL . . . 138
7.23 Simulation result of RAMHU using OFMC backend 138
7.24 Simulation result of RAMHU using CL-AtSe backend 138
7.25 Implementations of PHOTON 256-bit and ECIES 256-bit . . . 143
7.26 Users' scenarios in PAX 147
7.27 Part of Sarah's data 147
7.28 Part of a group of patients' medical records 148
7.29 PAX's framework in AVISPA 151

7.30 C_i role of PAX in HLPSL 152
7.31 CS role of PAX in HLPSL 153
7.32 AS role of PAX in HLPSL 154
7.33 DS role of PAX in HLPSL 155
7.34 Session, environment, and goal roles of PAX in HLPSL 156
7.35 Simulation result of PAX using CL-AtSe backend 157
7.36 Implementations of ECDSA 256-bit 161

A.1 The Merkle-Damgard construction of SHA (0, 1 and 2) hash functions 189
A.2 Architecture of BLAKE hash function 191
A.3 Comparison of hash functions speed 193

B.1 Arithmetic operations in ECC hierarchy 195
B.2 The PHOTON hash function 198

D.1 AVISPA's interface 203
D.2 AVISPA's architecture (The AVISPA Team 2006) 204

List of Tables

2.1 Comparison between EMR and EHR 11
2.2 Keys sizes and some information for public key algorithms . . . 20

6.1 Internal and external pseudonyms of users 97
6.2 Parts of SP and OP 97

7.1 Comparison of security features between REISCH and other data collection schemes 120
7.2 REISCH simulation parameters 121
7.3 REISCH computational processes 121
7.4 Comparison of ECDSA's procedures 127
7.5 Comparison of resistance in repelling the various threats between RAMHU and other authentication schemes 140
7.6 Comparison of computation cost between RAMHU and existing authentication schemes 145
7.7 Comparison of security features between PAX and other authorisation schemes 158
7.8 Comparison of performance between PAX and existing authorisation schemes 163

A.1 Comparison of SHA family 190
A.2 Versions of BLAKE hash function 191

B.1 Comparison of lightweight hash function algorithms 199

D.1 Some HLSPL's symbols and statements 204

List of abbreviations

$\oplus$	Exclusive or operation
$\|$	Concatenation operation
A	Aggregation function
AS	Attributes server entity
ASK_{pu_i}, ASK_{pr_i}	AS public and private keys
$ASS_j/ASSig_j$	Signature generated by AS and j is signature number
C_i	Client entity
$C_iK_{pu_i}, C_iK_{pr_i}$	C_i public and private keys
C_iS_j/C_iSig_j	Signature generated by C_i and j is signature number
CH	Cluster head
CM	Check MAC address
CN_i	Client's number
CS	Central server entity
CSK_{pu_i}, CSK_{pr_i}	CS public and private keys
$CSS_j/CSSig_j$	Signature generated by CS and j is signature number
Dec_i	Decryption operation
Dif	Value proves SN_i in the HWSN's area
DS	Data server entity
DSS_j	Signature generated by DS and j is signature number
EI	External intruder

Enc_i	Encryption operation
GM	Get MAC address
$h(.)$	One-way hash function
I	Internal, or external intruder
II	Internal intruder
$K_{pu_i},\ K_{pr_i}$	Public and private keys
LS	Local server
m	Message sent by entity
MID_i	Medical centre identity
MS	Master secret/Master signature
N, SN	Random nonces and random secret nonce
N_{C_i}, N_{CS}, N_{AS}	Nonce random generated by $C_i,\ CS,\ AS$
OP, SP	Object 's pseudonym, subject's pseudonym
OTP_i	One time password to authenticate first time
P	Entity parameters
$Parity$	The value specifies the signature of even/odd
$Pseud$	Pseudonym generated by entities ($SN,\ CH,\ LS,\ CS$)
$PW_i, tmpPW_i$	C_i's password, temporary password
R_i	Role of patient, patient relative or provider
RN	The random number generated by entities
RN_{op}, UN_{op}	Role's number, user's number for OP
RN_{sp}, UN_{sp}	Role's number, user's number for SP
RR_i	Revocation reason
S_{ID}, O_{ID}	Subject ID, object ID
$S_N C_H D$	Distance between SN and CH

$S_N L_S D$	Distance between SN and LS
S_R, O_R	Subject role, object role
S_{sp}, S_{op}	Signature of SP, Signature of OP
$SigLS,\ SigCS$	Signatures generated by $LS,\ CS$
$SigSN,\ SigCH$	Signatures generated by $SN,\ CH$
$SigSnEi,\ SigLsEi$	Random ephemeral value the same length as the signature generated by SN, LS
SL	Sensor location
SN	Sensor
SS	One secret sharing
tm	Temporary
TS	Timestamp generated by entities
TS_{AS}	Timestamp generated by AS
TS_{C_i}	Timestamp generated by C_i
TS_{CS}	Timestamp generated by CS
U	User
UID_i	C_i's identity
UN_i	User's number
$UP_{CS}^{AS}, MP_{CS}^{AS}$	U and MC pseudonyms sent by CS and verified by AS
$UP_{CS}^{C_i}, MP_{CS}^{C_i}$	U and MC pseudonyms sent by CS and verified by C_i
$UP_{AS}^{CS}, MP_{AS}^{CS}$	U and MC pseudonyms sent by AS and verified by CS
$UP_{C_i}^{CS}, MP_{C_i}^{CS}$	U and MC pseudonyms sent by C_i and verified by CS
UR_i	User's role (patient, patient relative or provider)
dy	Dolev-Yao model
ABAC	Attribute-based access control
AC	Access control

CR	Collision resistant
DoS	Denial of service
ECC	Elliptic curve cryptography
ECDSA	Elliptic curve digital signature algorithm
ECIES	Elliptic curve integrated encryption scheme
EHR	Electronic health record
EMR	Electronic medical record
FPR	First preimage resistant
HC	Healthcare
HWSN	Healthcare wireless sensor network
MC	Medical centre
MITM	Man in the middle
PAX	Pseudonymization and anonymization with the XACML
PHI	Protected health information
RAMHU	Robust authentication model for healthcare users
RBAC	Role-based access control
REISCH	Reliable and efficient integrity scheme of data collection in HWSN
SPR	Second preimage resistant
XACML	Extensible access control markup language
XML	Extensible markup language

List of Publications

These publications based on thesis.

- Al-Zubaidie, M., Zhang, Z. & Zhang, J. 2019, 'Effcient and secure ecdsa algorithm and its applications: A survey', *International Journal of Communication Networks and Information Security*, vol. 11, no. 1, pp. 7-35, 2019.
- Al-Zubaidie, M., Zhang, Z. & Zhang, J. 2019, 'RAMHU: A new robust lightweight scheme for mutual users authentication in healthcare applications', *Security and Communication Networks*, doi:10.1155/2019/3263902, vol. 2019, pp. 1-26, 2019.
- Al-Zubaidie, M., Zhang, Z. & Zhang, J. 2019, 'PAX: Using pseudonymization and anonymization to protect patients' identities and data in the healthcare system', *International Journal of Environmental Research and Public Health*, doi:10.3390/ijerph16091490, vol. 16, no. 9, pp. 1-36, 2019.
- Al-Zubaidie, M., Zhang, Z. & Zhang, J. 2020, 'REISCH: Incorporating Lightweight and Reliable Algorithms into Healthcare Applications of WSNs', submitted to the *Journal of "Applied Sciences"*, 2020.
- Al-Zubaidie, M., Zhang, Z. & Zhang, J. 2020, 'User Authentication into Electronic Health Record Based on Reliable Lightweight Algorithms', submitted to the *IGI-Global "Handbook of Research on Cyber Crime and Information Privacy"*, 2020.

Chapter 1: Introduction

In previous decades there have been many advances and research projects related to the novel technology for healthcare (HC) and innovative applications of the latest wireless sensing networking technology. In this thesis, we describe our research project which integrates wireless sensor network (WSN) technology with HC systems for electronic health record (EHR). In particular, this chapter will introduce the significance of the proposed research, and the specific research questions and objectives.

1.1 Overview of HC in Some Developed Countries

The health sector in the past has suffered from several problems such as management, access, complexity, storage and the protection of medical records (Meri et al. 2019). These problems have reflected negatively on the provision of quality care. Many governments, such as those in Australia, UK, USA and Canada, are concerned that the health sector provides citizens with adequate healthcare and alleviates the suffering of patients. For decades, researchers have continued to support the health sector through the development of healthcare applications that operate on electronic systems, such as EHR and electronic medical record (EMR) that provide stakeholders with accurate data in quick time from anywhere.

These systems are a huge development in the health sector and health services. This accurate data helps providers review data, diagnose diseases and prescribe medications to patients. For example, the European Union is actively pursuing research and development in the health sector that has contributed to disease prevention and healthy living (García-Holgado et al. 2019). Given the importance of the health sector, the European Union supports this sector with large sums of up to €449.4 million for the development of the Third Health Program. Therefore, in recent years, health sector institutions have relied on several technologies, such as eXtensible Markup Language (XML)/eXtensible Access Control Markup Language (XACML) to facilitate the handling and management of patient data and information. These technologies allow users in the health sector to share medical records. However, this sharing requires strict precautions and security measures to prevent data from being destroyed or modified by intruders.

1.2 Information and Communications Technology

The integration of Information and Communications Technology (ICT) with HC is extremely important to increase the effectiveness of healthcare applications and to improve peoples' perceptions and hopefully the willingness to accept the service. ICT has revolutionised the health sector; supporting the provision of quality medical services for patients. ICT contributes to the HC of communities in terms of productivity, reduction of costs and facilitation of information sharing (Haftu 2019).

An analytical study on the use of ICT in the health sector indicated that technology is having a positive impact on healthcare development. This study covered 184 countries including Australia, Canada and Germany. ICT was found to offer technological breakthroughs in dealing with medical records via digital devices such as computers, telephones and WSNs (Carvalho et al. 2019). These technologies provided services to patients and healthcare providers, such as allowing patients access to medical reports, results of operations or even obtaining a medical history from a remote server. They have also contributed to the improvement of the quality of service, but more sophisticated techniques are still required to address security and privacy issues in support of the quality of service.

1.3 Security Concerns in HC Services

Security and privacy are two critical aspects in both the development of HC services and the delivery of HC applications. Security is the concept used to indicate that user authentication is necessary before access to the network services is granted (Hamidi 2019). For instance, a provider such as a doctor, has to authenticate before accessing patients' data. Privacy indicates the level of access to network data and services. This level depends on polices and decision engines. Privacy divides users into roles or privileges that specify access by legitimate users to specific data. For instance, a provider such as an emergency doctor, can access a specific patient's data, but he/she cannot access patient's personal information. Security is more comprehensive than privacy since security can be privacy and the opposite is not true. These concepts are critical in the health sector. Consequently, the health sector needs to apply security and privacy in three respects to ensure the protection of patient secrecy:

- Datasets storage: Using techniques such as hash functions and camouflage, to hide users' information on servers
- Authentication: Using techniques such as username and password, to allow user communication within the network
- Authorisation: Using techniques such as XACML and access models, to allow users to access a specific level in the server's repository.

To ensure that the aforementioned three aspects are achieved in healthcare projects, they should be supported with encryption and signature mechanisms.

1.3.1 Encryption for Preserving Users' Privacy

Encryption has been used to preserve patient record secrecy and users' identities. For example, encryption protects information, such as usernames and passwords, within the authentication request when moving from the user's device to the server. Encryption algorithms are divided into symmetric encryption, such as Advanced Encryption Standard (AES) and Data Encryption Standard (DES), and asymmetric encryption, such as Elliptic Curve Cryptography (ECC) and Rivest Shamir Adleman (RSA). In symmetric encryption algorithms, all network members use the same secret key, while with asymmetric encryption, each network member has a unique secret key (Dwivedi et al. 2019). Security in healthcare projects depends, not only on encryption itself, but also on factors that ensure that the encryption algorithm is effective in protecting patients' secrecy. Examples of such factors are key length, security level, randomness, support for privacy mechanisms and recommendations of prestigious encryption institutions.

1.3.2 Signature for Users' Authenticity

The signature performs mathematical operations on the data/information within the authorisation/authentication request in a way that allows the legitimate user to obtain complete, accurate and change-free data/information. For instance, the WSN collects and signs patients' data by the Elliptic Curve Digital Signature Algorithm (ECDSA) to prevent data from being changed as it travels from sensors to the server. There are several ways to perform digital data signature operations. For example, public key algorithms, such as ECDSA, RSA and Elgamal can be used. It is also possible to use hash functions such as Secure Hash Algorithm (SHA1), PHOTON, BLAKE and QUARK to perform signature processes (Heigl et al. 2019). Public key algorithms are more secure than hash functions, but the latter is the best performing for some algorithms. As mentioned earlier, security is the primary and first condition for accepting healthcare systems.

1.4 Significance of the Project

ICT applications in healthcare have offered great benefits to the society by providing care for patients and improving their health. However, collecting modified or inaccurate data (because of attacks) and storing these on servers as well as unauthorised access to a server's database can cause significant harm to patients' health. Also, data exchange between different devices in the network requires data security management to deal with medical records in a flexible and accurate manner. The lack of protection of patients' information adversely affects the treatment of patients, which leads to harm, reduced dignity, stigma, discrimination, embarrassment or even death.

Many research studies have shown that patients and professionals in hospitals or clinics have been interested in issues of information security (personal and health information) and protecting the rights of patients from data tampering. As far as we can see, however, the concerns with the security of patients' data/information have never been addressed effectively. The research described in this thesis addresses the security and privacy necessary to provide a safe environment for the storage and transfer of patients' medical records through the:

- Protecting of the patients' data collected continuously by WSN to prevent data modification
- Identification of legitimate network users (security)
- The determination of authorised users (privacy) of the database stored on servers, as well as secure data management.

1.5 Research Objectives and Questions

This section lists the objectives and problems for our research.

1.5.1 Research Objectives

The following objectives are laid out to achieve in our project research:

1. **Automate healthcare data collection using EMR, WSN and XML**
   Manually collecting healthcare data is expensive and prone to error. Using WSN to automatically collect and convert to electronic records can significantly improve quality and reduce the cost of the healthcare. As providers such as nurses and doctors need to continuously monitor patients,

WSNs help them collect data easily. WSN can be deployed to continuously monitor a patient's condition and gather data. Patients' data transmitted between sensors (nodes and cluster head) and network devices (such as a nurse and a server device) need data management algorithms to maintain both performance and security at the same time. The EMR which includes the patient's confidential data and private information, needs to be accessed by healthcare professionals. Thus, sharing such EMR without breaching a patient's privacy requires EMR management in an efficient and secure manner. XML technology has begun showing its superiority in the exchange of complex data over different systems

2. **Ensure communication of legitimate users using EHR and signencryption**
   Preventing illegal users from connecting to the network and sending fake requests to the EHR server is a prerequisite for accepting healthcare systems. EHR requests include confidential information, such as usernames and passwords. The disclosure of this confidential information can lead to data change or even destruction of the network. Therefore, an efficient signencryption technique is required to protect users' information. In this objective, we focus on encrypting information only because encrypting the entirety of data will be expensive for users' devices and even more expensive for servers

3. **Improve healthcare privacy and security using XACML and signature**
   When all patient information is stored on a server, that server becomes attractive to attackers. Therefore, the use of security mechanisms to determine access to the server is an immensely important issue. All protected health information (PHI) stored in the EHR repository should be been anonymous, and XACML with signature technology would be applied to achieve maximum security. We have used signatures to prevent changing data and XACML to apply policies in making decisions and accepting legitimate requests.

1.5.2 Specific Research Questions

In this section, we will describe some problems that might threaten the privacy and security of information/data in the proposed healthcare system. That is, we intend to investigate the following research problems to complete our study:

1. *How can WSNs help collect a patient's healthcare data within an EMR system efficiently?*

2. *How to transfer healthcare data for WSN to the EMR repository securely?*

3. *How to authenticate users (providers and patients) when communicating EHR services?*

4. *How to authorise healthcare professionals when accessing the healthcare information and data from the EHR repository?*

5. *How to update the EHR information stored in the repository while maintaining privacy?*

1.6 Organisation of this Thesis

This thesis is divided into eight chapters: Introduction, Literature review, General methodology, Storage scheme, Authentication scheme, Authorisation scheme, Discussion and results, and Conclusion and future research. All these chapters are interlinked to build a robust HC application project protecting against security and privacy threats. The chapters structure are as follows:

- Chapter 1: This chapter provides an introduction to the significance, objectives and questions of our project in protecting the security and privacy of HC users

- Chapter 2: This chapter introduces a review of the literature in security and privacy schemes in healthcare applications. It describes security problems and weaknesses in data storage, authentication and authorisation schemes in existing projects

- Chapter 3: This chapter presents details about architecture, repositories and cryptography techniques used in our project: Elliptic Curve Cryptography (ECC) and eXtensible Access Control Markup Language (XACML). In addition, it provides a general methodology for our whole project

- Chapter 4: This chapter focuses on our contributions and the methodology used to build the data collection scheme in HC application. This chapter was submitted as a journal paper

- Chapter 5: This chapter focuses on our contributions and the methodology used to build the authentication scheme in HC application. This chapter was published as a journal paper

- Chapter 6: This chapter focuses on our contributions and the methodology used to build the authorisation scheme in HC application. This chapter was published as a journal paper

- Chapter 7: This chapter discusses the theoretical and experimental analyses for our project's schemes. It describes the results of the implementation of our protocols in each scheme and proves its feasibility in securing medical records
- Chapter 8: This chapter summarises the conclusions and future research for our project.

1.7 Summary of the Chapter

In this chapter, we briefly described the importance of security and privacy in HC applications. A set of questions asked to identify the problems of access to medical records (information and data) is listed. We have also provided the project objectives to solve security and privacy issues in data collection, authentication and authorisation. Finally, the structure of the thesis is outlined to clarify the interconnections of the thesis chapters in the construction of an integrated and solid HC project.

Chapter 2: Literature Review

Both ICT and E-technology have been used in medical and HC systems for decades. Wireless and sensing technology, in particular, have been adopted by medical staff and healthcare professionals. However, in addition to the many ethical problems, there are security concerns such as patient privacy and chronic disease history records. The ICT and E-technology make it easier to breach the security of such systems, rather than harder if the security of these systems was well addressed. This chapter surveys the challenges, opportunities and technologies with an emphasis on three important aspects of security: service availability, authentication and authorisation. It introduces the following themes:

- History of HC applications and WSN integration into HC applications
- Security and privacy issues associated with HC applications
- Security protocols of HC applications
- Investigation of drawbacks in existing HC application systems.

2.1 Healthcare Standards

Medical and healthcare technology and systems have human ethics' implications, and ethical standards must be applied in all countries. Many standards for healthcare applications, such as Health Level Seven (HL7), Health Insurance Portability and Accountability Act (HIPAA) (Rezaeibagha et al. 2015) and Personal Information Protection and Electronic Documents Action (PIPEDA) have been developed in countries like Australia, UK, USA, Canada and Germany. These standards have an important role to play in creating healthcare applications. For instance, many countries have reprioritized the healthcare industry after defence and the military in terms of budget and strategy (Consultants to Government and Industries 2015), and it is becoming increasingly evident that advanced ICT holds the key to the success of providing better healthcare quality at lower cost.

Digital HC or e-health includes many systems, such as EHR, EMR and personal health record (PHR). These systems are used efficiently to share medical records either globally (EHR) or locally (EMR), and are administered by the authority service provider (EHR and EMR) or by the patient (PHR) (Heart et al. 2017). Digital healthcare services have become diversified and demanding. They are used to deliver efficient but affordable services to individual healthcare users and the broader community (Asan 2017). Furthermore, WSNs promise to significantly enhance the quality of care over a wide range of client populations. At the same time, e-health systems provide services that allow providers and patients to share medical records across various health centers such as hospitals, clinics, and even the home. These services provide facilities to help improve the health of patients. Because of the efficiency of electronically sharing patient data (rather than traditional paper-based methods), patient health data is available anywhere at anytime to healthcare providers and patients. HC institutions and researchers are seeking to develop these applications to improve the quality of care, diagnosis, and remote medical surveillance (Khatoun & Zeadally 2017).

A growing number of people have begun to gradually accept e-health services. However, the main problem that threatens the acceptance of these systems for patients and providers is the security and privacy of patients' information and data. This issue should be addressed whether collecting data, or in the authentication and authorisation processes. In the following subsections, we will explain the concepts of EMR and EHR, the advantages and disadvantages of the electronic record, and the integration of EMR and EHR.

2.1.1 On the Electronic Medical Record

An EMR is important compared to a paper medical record for many different features. A medical record is a communication tool used to record and review patients' health status for members of the medical staff and patients themselves. This tool is divided into two categories: paper and electronic record (Harman et al. 2012). The paper record is a traditional method used to check and record patient information. This type of medical record suffered from many problems when dealing with patient data. These problems include accessibility, availability, updating, delays, review, errors, data transfer, lack of coordination of care equality at different levels, management of health information and data, integration of scientific evidence into HC services, and decision-making practices (Beglaryan et al. 2017) and security issues. The second type of record is the electronic medical record. It rapidly processes and transmits data across digital devices. It is designed to provide HC services continuously and accurately. It has attracted the attention of both the HC industry and researchers because it provides advantages in efficiency and effectiveness.

EMR efficiency is achieved through many features, and it supports the use of WSN. EMR is normally a one-organisation system. It provides quality of care and facilitates monitoring, evaluates health conditions, provides reliability, data quality, minimizes human error, facilitates access to information, and reduces the cost of information and communication technology (Najaftorkaman et al. 2015, Muthee et al. 2018). EMR is rich with patient data and provides timely collection and retrieval of data (Muthee et al. 2018, Osmani et al. 2018). Currently, most of Australian professionals use EMR, and it is rated similarly in several countries such as Germany, New Zealand and the Netherlands (Heart et al. 2017).

EMR stores patient health data within a single institution and uses WSN to store patient data in a local repository for use in reports, diagnosis, and treatment. But, an EMR only contains a partial patient medical history (Heart et al. 2017). For example, doctors may use an EMR to identify a patient's prescription and avoid errors, and nurses may use an EMR to monitor tests and reports for a patient. But if the doctor needs complete data about a patient's medical history, he/she needs to send a request to the central server.

2.1.2 On the Electronic Health Record

An EHR is an efficient system supporting large health enterprises by sharing data and providing HC services. Unlike the EMR, the EHR is defined as an inter-organisational system. It includes complete medical history data for each patient that can be provided to professionals. Moreover, it shares patient data among health centres across providers (Heart et al. 2017). EHR is widely deployed in healthcare (hospitals and medical clinics) (Alkureishi et al. 2018) due to its services to providers and patients. Such services are error reduction, increased efficiency, improved care, a rich source of data for researchers, health details including diagnosis and treatment, (Chiu & Hripcsak 2017, Levine et al. 2018), laboratory tests, clinical prescriptions and observations (Shickel et al. 2018).

The acceptance of the EHR system in health institutions has become increasingly important in recent decades. Although EHR was used as a tool for archiving patient data, it has recently become a basic communication tool for patient care and service provision (Asan 2017, Shickel et al. 2018, Czaja et al. 2018). According

Table 2.1: Comparison between EMR and EHR

No	Aspect	EMR	EHR
1	Database	One	Many
2	Cost	Less	More
3	Patients' history	Partial	Complete
4	Access data	Anytime	Anytime and anywhere
5	Communication technology	Wireless	Wireless and Internet
6	Support	Hospitals or clinics	Community, State or national organisation
7	Sharing data	One organisation	Inter-organisation

to the latest report from the Office of the National Coordinator (ONC) for health information technology, approximately 84% of hospitals have adopted the EHR primal system. The Health Information Technology for Economic and Clinical Health (HITECH) in the USA has provided $30 billion as incentives for hospitals to adopt EHR (Shickel et al. 2018). EHR provides benefits for both individuals and institutions (Beglaryan et al. 2017). For example, physician researchers may use an EHR to determine a patient's history for use in a study developing a treatment.

EHRs store medical records for patients in a digital central database and it manage these records between medical centres. EHR provides patients risk assessment depending on the medical reports. In addition, it uses the Internet to transfer patient information/data. This information sharing between medical institutions makes it easier for doctors to diagnose and treat patients at any medical centre (Chen et al. 2012). However, EHRs also suffer from the problem of security weakness during the transfer of data over the Internet or when accessing data in the server database. Therefore, security mechanisms are considered the cornerstone of EHR systems. Table 2.1 presents a comparison between EHR and EMR.

2.1.3 Pros and Cons of Electronic Medical/Health Record

A number of projects developing healthcare systems have utilized EHR and/or EMR technology (Asan 2017, Alkureishi et al. 2018, Beglaryan et al. 2017). The prototype systems developed in these studies suffer the following drawbacks:

1. Increased burden on health professionals, negative impact on communication between patient and health professionals, the patient can connect with the doctor to ask him about a report and also can connect with the health centre to get information or data history.

2. The attendance of a small number of doctors treating a larger number of patients in a particular medical institution (Asan 2017)

3. Lack of transparency in patient information usage (Alkureishi et al. 2018)

4. On an individual level: impact on functionality, external control limits, resistance to change

5. On an organisational level: lack of legal framework, structural, financial, technical and lack of confidence in electronic communication (Beglaryan et al. 2017)

6. The problem of security and privacy when accessing patient data and information.

Recent studies (Asan 2017, Beglaryan et al. 2017, Gold et al. 2017, Alkureishi et al. 2018, Senteio et al. 2018) have also provided us with the following advantages:

1. Ease of reviewing the patient to his/her information and data

2. Many users can review the same medical record simultaneously

3. Auto-updating and search speed in information retrieval

4. Improved patient understanding of care services

5. Facilitation of patient participation and cooperation in decision-making

6. Reduced errors in documents

7. Reduced embarrassment for the patient when interacting with a health professional

8. Transparency of cooperation, and improvement in the interactions between the patient and providers

9. The use and quality of health information, quality of care, efficiency and cost of care

10. Facilitation of data collection, retrieval and use of patient data.

2.1.4 Integration of EMR and EHR

There is no doubt that the integration of EMR and EHR provides many benefits to healthcare providers and patients. These include providing an overview of the concept of patient care, providing the opportunity to study a patient's disease across several health centres (Osmani et al. 2018), providing a better understanding of chronic diseases, and the abundance of data in a central repository that can be an important source for researchers to develop and analyse treatments. One of the largest HC information technology suppliers, Actien-Gesellschaft für Anilin

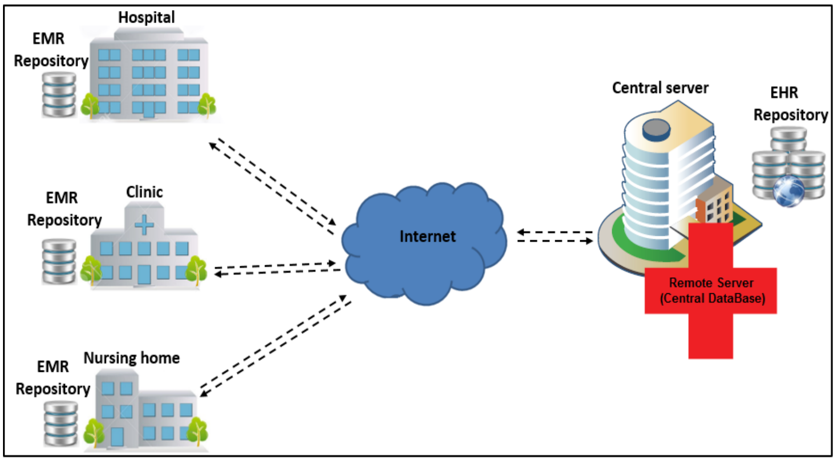


Figure 2.1: Integration of EMRs with EHR repository

Fabrikationen (AGFA), has enhanced the concept of integrating healthcare systems, such as EMR and EHR. This organisation claims that integration generates the best medical decisions and improve the quality of patient care at the individual level and the public health level (Heart et al. 2017). Figure 2.1 shows the integration of EMRs' local repositories with a central repository in EHR on the central server.

2.2 Acquiring EMR/EHR in HC Applications

The wireless sensor network (WSN) is one of the most promising technologies developed in recent times, and has been used in many areas (Al Ameen et al. 2012, El Barachi & Alfandi 2013, El-Semary & Abdel-Azim 2013). WSN technology is at the very early stage of adoption by the medical and healthcare sector. To our best of knowledge, there are just a few research projects being undertaken to automate the process of acquiring patient's medical records in hospitals or elderly healthcare records in the nursing homes.

2.2.1 Main Applications of HWSN

One such system is known as healthcare wireless sensor networks (HWSN) (Ayyildiz et al. 2019). The architecture of HWSNs is shown in Figure 2.2. The HWSN has primarily been used for the following purposes:

- **HC data collection**
  In the first aspect, sensors continuously collect data about the patient and send it to the server (EMR repository). The data collected properly helps doctors diagnose diseases accurately. Furthermore, patient data collection needs protection from intrusion

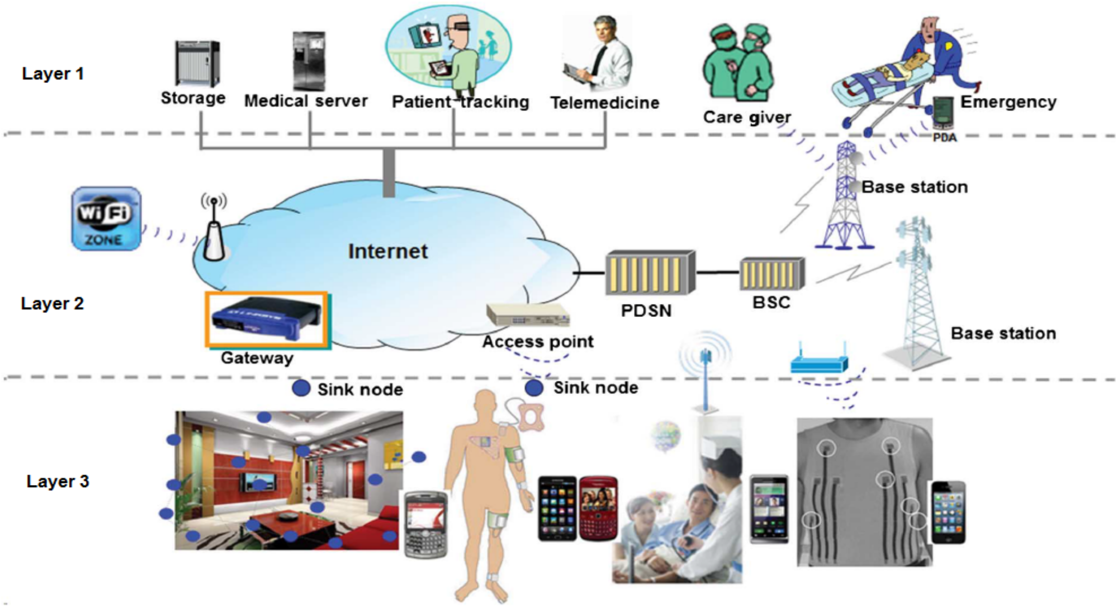


Figure 2.2: Architecture of healthcare based WSN (Zhang et al. 2014)

- **HC data structure**
  The second aspect is the data storage in the form of the datasets in a server's database. The data structure for an EMR/EHR repository should be able to facilitate the sharing of a patients' health information among the HC professionals

- **HC data access and privacy**
  The third aspect is to determine who has the right of users (doctor, nurse, general practitioner, pharmacist and government officer) in the access to these datasets in the data server. All these stages require security and privacy mechanisms to protect the EMR and EHR.

2.2.2 Security and Privacy in HWSN

There are some problems inherent in HWSNs. Figure 2.3 describes the attacks on the HWSNs which are used to infiltrate data collected/stored. Many attacks on, and threats to, the HWSN's security are relevant to the collection of patient data and the privacy of the EMR repository. These attacks have been classified into passive and active attacks (Aceto et al. 2018, Gao et al. 2018). In all types of passive attacks, an adversary eavesdrops on the transmitted packets between network nodes and the server. Also, the attacker analyses these packets to reveal information without changing it (i.e. trying to break the confidentiality), such as eavesdropping and traffic analysis attacks (Al Ameen & Kwak 2011). The active attack is extremely harmful to the networks of healthcare applications. An attacker

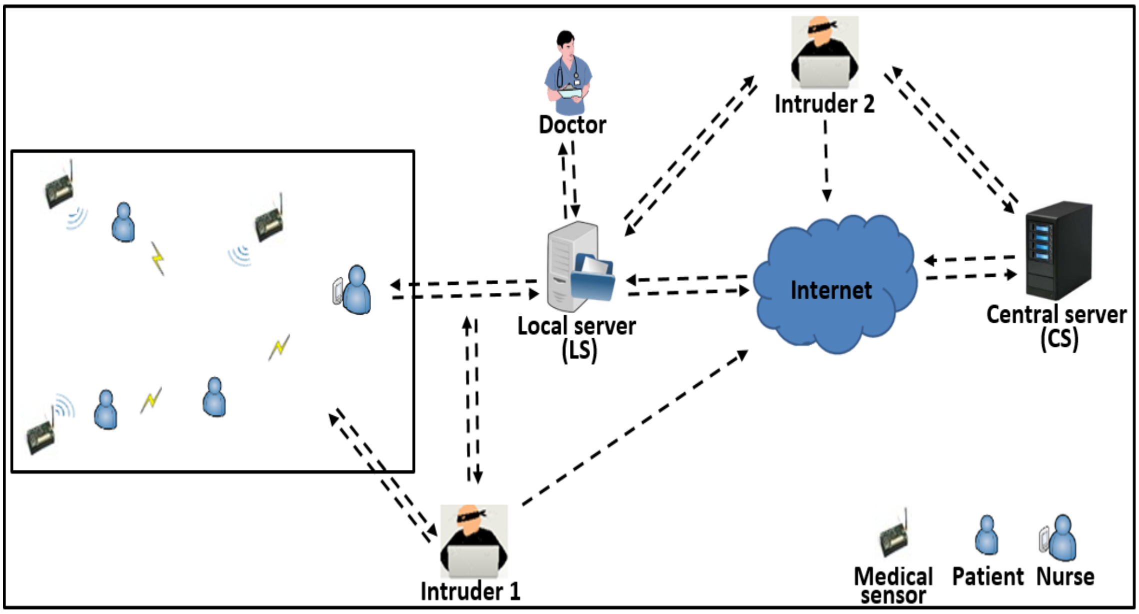


Figure 2.3: An attack on information security (Intruder 1) and device security (Intruder 2)

tampers with data packets and sends them back to their destination, such as masquerading, reply, modification and denial of service (DoS) attacks (Li et al. 2018). Another classification of attacks on healthcare applications is the internal and external attack. In the internal attack, the attacker is a member of the network. This type of attack is more dangerous than external attack because the internal attacker has the authority to send and receive messages/requests, which means an attack from the inside is easier.

Potential attacks on data transferred or stored in an EMR repository by WSN are a serious risk to HC systems. As we can see in Figure 2.3, Intruder 1 can listen to information as it is transferred from the patients' sensors to the server (local server or base station). When the attacker intercepts the message, he/she can obtain information about the physical location of the patient, identifier (ID), timestamps, source address, target address and the medical report sent by the sensors or directed by medical staff. The patient data transmitted through the sensor networks requires complete security and privacy, especially when movement through the network does not require the consent of the patient, such as moving the data of an emergency case. In addition, Intruder 2 can perform an attack on the local/remote server to penetrate the database to obtain patient information. Also, an attacker can get information from the database, such as the patient's name, age, address, type of disease, and the seriousness of the disease. This information allows the attacker to harm the patient in different ways, such as changing or destroying data (Kumar & Lee 2011, Pawar et al. 2018). Therefore, privacy and security issues are extremely important in healthcare applications. If these applications do not provide adequate security for patient information, they become useless and unusable because the disclosure of patient information may affect patient health or even result in death.

2.3 Security of HC Applications

HC applications have their own special security requirements. For instance, these applications require security and privacy mechanisms, such as authorisation policies, encryption algorithms, and robust signatures. The task of these essential mechanisms is to protect medical repositories in EMR and EHR from malicious attacks. According to the Vormetric report on data security 2016, healthcare is one of the sectors that is most vulnerable to hackers' attacks and thus requires increased efforts to secure health data by 64% (Garrett 2016). On 21 January 2016, the report stated that 91% of enterprises suffer from vulnerability threatening data security (internal and external attacks). The study of data security included several countries including Australia, USA and Germany. Therefore, many systems, such as National E-Health Transition Authority (NEHTA) in Australia and the HIPAA in the USA, recommend the correct and accurate application of security and privacy in HC applications to prevent security threats (Gajanayake et al. 2014).

In the following subsections, we will describe HC applications in terms of security issues, possible attacks, security requirements and security protocols.

2.3.1 Specific Security Concerns of HC Services

Security concerns with HC applications include two aspects: security and privacy while acquiring EMR/EHR using WSNs; and the confidentiality of patient EMR/EHR, authenticity and/or authorisation of health professionals. These issues are critical to the acceptance and success of HC applications in the health sector. These issues are represented by the transfer of information and data between network entities (sensors, users' devices and EMR/EHR servers), the storage of databases on a server and the performance of computation processes in resource-constrained devices:

- **Communication Security**
  To protect data and information between source and destination, security mechanisms, such as encryption and signature should be applied to prevent an attacker from accessing records transferred between network entities. These mechanisms resist attacks such as disclosure, alteration, replication and impersonation of medical records transmitted. The communication channel (wireless and Internet) should be protected end-to-end both at the wireless

level and the Internet through the integration of a set of security mechanisms and privacy (Manogaran et al. 2018, Bruland et al. 2018).

- **Dataset Security**
  Data and information stored on the server repository become the target of malicious attacks. In particular, if a HC application is based on a single server, the process of hacking this server results in both data and information being detected (Chuang & Chen 2014). In addition, access to databases without pseudonym and anonymity mechanisms makes it easy for attackers to detect users' real identities. Therefore, HC applications should include separate servers with different tasks. To protect users' medical records, each server has separate duties for user information and data as well as the implementation of pseudonym and anonymity mechanisms to access patient data. For instance, one server contains only users' identities and another contains only users' data. Furthermore, the database should be available to legitimate users at any time and from anywhere, as should support authorisation policies for access to the repository (Griggs et al. 2018). Authorisation policies determine the level of access granted to legitimate users.

- **WSN Security**
  WSN requires efficient security algorithms to work efficiently. EMR systems use WSN to collect patient data. However, a WSN is source-constrained in terms of energy, computing and memory. Therefore, when using encryption and signature mechanisms, security and performance should be efficient. The efficiency of these algorithms is a major challenge in HC applications. It means that sensor nodes continue to collect patient data accurately and for a long time while protecting the data collected from penetration (Al-Turjman & Alturjman 2018, Verma et al. 2018).

2.3.2 Possible Attacks on HC Applications

HC applications are vulnerable to a number of attacks. Here we present some real examples of possible attacks:

- **Authentication attacks** represent many examples of real-world security threats implemented against user authentication in HC applications:
  - In 2016, according to a cybersecurity firm's layer 8 security report analysis, attacks revealed a huge number of passwords and keys during the authentication processes of remote devices in HC organisations (Siwicki 2016)

  - In 2017, a patient's personal information was compromised at the AU Medical Centre, Children's Hospital and Clinics of Georgia. However, information dataset attacks were not detected until 2018 (Donovan 2018)
  - In 2018, according to the proofpoint report, more than 100 million authentication attacks were carried out around the world against clinics, hospitals and insurance companies (proofpoint 2018)
  - In 2019, the Oregon Department of Human Services pointed out that cyber-attacks targeted and breached users' credentials (625000 patients' records) (Jessica Davis 2019).

- **Authorisation attacks** represent many examples of real-world security threats implemented against users' authorisation in HC applications:
  - In 2013, penetration attacks were on healthcare data in US hospitals. These attacks revealed 85.4% of the medical records (protected health information (PHI)) and they are considered to be one of the five largest penetration incidents for patient data (Paganini 2014)
  - In 2016, Apple Health (Medicaid) was exposed to a data breach. This attack revealed 370,000 records for clients at Apple Health (Washington state) (Washington Health Care Authority 2016)
  - In 2017, an unauthorised individual penetrated the EHR at the New Jersey Diamond Institute for Fertility and Menopause. The hacker revealed the PHI of 14633 records containing patient information such as names, birth dates, social security numbers, and sonograms (Davis 2017)
  - In 2018, the US Department of Health and Human Services pointed out that unauthorised access/disclosure attacks targeted many health institutions and penetrated huge numbers of health records (U.S. Department of Health and Human Services 2018).

2.3.3 Security Requirements of HC Applications

In this section, we summarise the security requirements of HC applications. The security requirements of HC applications include:

- **Confidentiality** data encrypted through an encryption algorithm to prevent the attacker from seeing explicit data. When the attacker obtains the encrypted data, the attacker will not benefit from this data because it is incomprehensible (Kumar et al. 2018)

- **Authentication** an authentication service is used to authenticate legitimate users or data in the network to prevent anyone else from accessing data. This means that if the data is a trusted source in the network it is accepted but, if it is an unknown source, it is ignored (Rantos et al. 2018)

- **Authorisation** in the authorisation service, each node (sensor or user's device) in the network has specific sources that can access it. This security requirement is tremendously important in preventing unauthorised persons' access to the sources, for example, providing various privileges among users (doctors, nurses, practitioners and pharmacists) in access to the sources (Javadi & Razzaque 2013)

- **Integrity** an integrity service is used to ensure that the transmitted data is not tampered with or edited by the adversary (Sun et al. 2018).

In addition, the following concerns are closely related to patient's EMR/EHR:

- **Availability** some attacks attempt to disrupt network services by sending a large number of messages to the server, and thus, destroy the network. Network services should be available upon request (Di Pietro et al. 2014)

- **Anonymity** this service hides or distorts the network information and data as it transfers from the sender to the receiver or vice versa. When using anonymity with information and data, the attacker cannot distinguish this information and data to a specific patient (Shen et al. 2018)

- **Unlinkability** the attacker cannot reveal the identity of HC users when linking real information with random pseudonyms. However, the single pseudonym will expose user information to detection (Mehmood et al. 2018)

- **Scalability, Forward Secrecy and Backward Secrecy** environments of healthcare applications require a continuous expansion of the network size. But when a node leaves or joins the network, it does not have the right to access and decrypt the encrypted messages in the future after leaving the network or previous messages before entering the network (Dhillon & Kalra 2018)

- **Freshness and Non-repudiation** the message should be recent to prevent a replay attack, as the sender cannot deny his message to detect the compromised nodes (Di Pietro et al. 2014, Kale & Bhagwat 2018).

Table 2.2: Keys sizes and some information for public key algorithms

Algorithm	Keys sizes					Ratio	Author(s)	Year	Mathematical problem	Other algorithms
RSA	1024	2048	3072	7680	15360	1:6-30	Rivest, Shamirand, and Adleman	1978	Integer factorization	Rabin
Elgamal							Taher Elgamal	1985	Multiplicative group	Schnorr, Nyberg-Rueppel
DSA							David W. Kravitz	1991		
ECC/ECDSA	160-223	224-255	256-383	384-511	512-more		Scott Vanstone	1992	Elliptic curve discrete log. (ECDLP)	ECDH

2.3.4 Security Protocols in HC Applications

In this section, we present an overview of security protocols likely to be used in the HC applications. HC applications require robust protocols in terms of performance and security. Since HC applications normally rely on the traditional cryptographic protocols and/or primitives to achieve their security goals, and traditional encryption/decryption algorithm, public key-based authentication used very long keys. This has been identified as a major obstacle to efficiency.

ECC/ECDSA has been proven to be efficient in its performance because it uses small keys. This makes the cost of computation small compared with traditional public key cryptography algorithms such as RSA, traditional digital signature algorithm (DSA) and ElGamal. For example, ECC/ECDSA with a 256-bit key offers the same level of security as the RSA algorithm with a 3072-bit key (Varchola et al. 2015, Ever 2018). Table 2.2 (Barker & Dang 2016, Harkanson & Kim 2017, Tiwari & Kim 2018, Mi et al. 2018) shows a comparison of key sizes for public key signature algorithms in addition to some information about these algorithms. Security requirements are considered from three perspectives:

- **Performance and efficiency**
  The preservation of efficiency in the ECC/ECDSA is critical in HC applications. Many approaches have been developed to improve the efficiency of the ECC/ECDSA algorithm, reducing the cost of computation, energy, memory, and consumption of processor capabilities. The ECC/ECDSA uses point multiplication (PM) or scalar multiplication (SM) where ECC uses PM for encryption and decryption, while ECDSA uses this operation to generate and verify signatures (Pan et al. 2017). One can improve PM efficiency by improving finite field arithmetic (such as inversion, multiplication, and squaring), elliptic curve model (such as Hessian and Weierstrass), point representation (such as Projective and Jacobian), the methods of PM (such as Comb and Window method) (Joye 2008) or improve hash function.

  Several researchers have made improvements to increase the performance of the ECC/ECDSA. First, in De Dormale & Quisquater (2007) performance and flexibility were investigated in the ECC algorithm with accelerators through hardware implementations. Many issues associated with hardware implementations for ECC were discussed, such as selecting curves, group law, PM algorithms, and the selection of coordinates. In addition, it was pointed out that the architecture of multiple PM in ECDSA verification should be supported because this architecture leads to efficiency in hardware implementation. Much research has pointed out that using hardware accelerators leads to high performance. But, accelerators sacrifice flexibility where reduction circuits should be used to retrieve the flexibility feature. Similarly, Driessen et al. (2008) compared many different signature schemes (ECDSA, XTR-DSA, and NTRUSign) in terms of energy consumption, memory, key length and signature, and performance. Through implementation, the authors found that the NTRUSign algorithm is the best in terms of performance and memory. However, the NTRUSign algorithmis weak when under attack.

Zhong et al. (2016) investigated the ECDSA algorithm and found that it contains some problems that make it inefficient because of the inversion processes used to generate and verify the signature. They removed the inversion process and the results demonstrated that this scheme is more efficient than the previous scheme as it had less running time. Unfortunately, there is no proof of ECDSA security without inversion processes for repelling attacks. Liu et al. (2017) adopted the Montgomery method with the lightweight elliptic curve (twisted Edwards curve (p159, p191, p223, and p255)) to improve speed and balance between memory and performance (cost of communication, execution time, memory). During the implementation, they noted that their scheme offers better memory efficiency than the traditional Montgomery scheme. They recommended the exact selection of ECDSA's parameters curves and a balance between security and efficiency requirements.

- **Security and countermeasures**
  The security improvement in ECC/ECDSA is no less important than its efficiency because this algorithm is designed primarily for the application of security properties. ECC/ECDSA, like previous algorithms, may possibly suffer from some of security vulnerabilities. Random weak, bad random source (Bos et al. 2014), collision, preimage and second preimage in hash value or leaking bits of the private key are all considered weaknesses in ECC/ECDSA. Also, several researchers have made improvements that close security gaps in

the ECC/ECDSA algorithm. The aim of these improvements is to provide countermeasures against various attacks. But when selecting countermeasures, there should be a balance between security and efficiency (Danger et al. 2013). To maintain the security of these algorithms, it is important to use finite fields (either prime or binary) recommended by credible institutions. For instance, the Federal Information Processing Standard (FIPS) or National Institute of Standards and Technology (NIST) are considered reliable. Choosing appropriate curves and finite fields according to authoritative organisations' standards leads to secure ECC/ECDSA's implementations (Mathur et al. 2017). Therefore, we note from the above, that any encryption algorithm or signature should possess a security level appropriate for use in HC applications. Many schemes have measured ECC/ECDSA in terms of security weakness issues and countermeasures.

Many studies support appropriate countermeasures in ECC/ECDSA. Fan et al. (2010) presented a detailed study on attacks and countermeasures in ECC algorithm. They divided attacks into passive and active attacks. They explained that the countermeasure for a specific attack may be vulnerable to other attacks. In addition, countermeasures should be selected carefully. Therefore, the authors have made some recommendations for selecting countermeasures.

Some surveys have studied public cryptography algorithms in terms of the computation of hard problems (integer factorization problem (IFP), discrete logarithm problem (DLP), lattices and error correcting codes) in quantum and classical computers (Abdouli et al. 2011). These authors described RSA, Rabin, ECC, ECDH, ECDSA, ElGamal, lattices (NTRU) and error-correcting code (McEliece cryptography). They pointed out that ECC provides a higher security level than other cryptosystems. In addition, it presents advantages such as high speed, less storage, and smaller keys sizes. But they did not discuss the use of ECC/ECDSA in applications and implementations of different technologies.

Meanwhile, Fan & Verbauwhede (2012), Danger et al. (2013) explained physical attacks on ECC algorithms. They focused on two known physical attacks: side channel analysis (SCA), and fault attacks. They also described many attacks, including these two, as they presented countermeasures against these attacks. The countermeasures included simple power analysis (SPA), differential power analysis (DPA) and fault attack (FA) countermeasures. Also, some recommendations were presented for countermeasures that add randomness, countermeasures selection, and implementation issues.

Bhatia & Verma (2017) discussed security issues in terms of network security, identity theft, and insider threats in preventing many attacks, such as man-in-the-middle (MITM), address resolution protocol (ARP) poisoning, insider privileges, reply, DoS, guessing and impersonation. They noted that ECC/ECDSA is the best of the public encryption algorithms. It provides a security solution for data protection through the integration of cryptography mechanisms in ensuring remote authentication and authorisation.

- **Implementation and applications**
  ECC/ECDSA is suitable for implementation in resource-constrained environments and various applications. A study on security techniques investigated WSNs (Yang et al. 2015). It focused on three features in WSN security: key management, authentication, and secure routing. It pointed out that the ECC algorithm was convenient for resource-constrained devices. In addition, a survey of attack strategies was given in relation to ECC/ECDSA in Bitcoin and Ethereum applications (Mayer 2016). The author pointed out that different standards for curves (such as ANSI X9.63, IEEE P1363, and safecurves). This survey focused on safecurves with SECP256k1 through using ECDSA. Also, it referred to safecurves as one of the strongest curves standards. The author suggested many basic points to prevent attacks on ECDSA or ECC.
  Finally, Harkanson & Kim (2017) compared RSA and ECC/ECDSA. They pointed out that ECC/ECDSA exhibited the highest performance with the same level of security from RSA. They noted that 69% of websites applications use ECC/ECDSA, 3% used RSA and the rest used other algorithms. They also described ECC with some applications (such as vehicular communication, e-health and iris pattern recognition). However, they had a duplication between implementation and application. For example, RFID is a technology that can be used to implement a particular application.

2.4 Some Studies Related to our Research

This section discusses a few recent studies, related to our research, in terms of storage, authentication and authorisation.

2.4.1 Storage of User Data

This section briefly discusses the existing storage schemes (Wander et al. 2005, Wang & Li 2006, Trakadas et al. 2008, De Meulenaer et al. 2008, Fan & Gong 2012, Kodali 2013, Lavanya & Natarajan 2017*a*,*b*, Staudemeyer et al. 2018, Malathy et al. 2018, Sharavanan et al. 2018, Sui & de Meer 2019, Hathaliya et al. 2019, Furtak et al. 2019) designed to secure patient data in the EMR, and highlights their shortcomings.

The performance and security presented by the ECC/ECDSA algorithm make it suitable for use in implementation on WSN. Digital signatures in ECDSA have better efficiency in resource-constrained devices than DSA and RSA. Many authors have pointed to the possibility of using ECC/ECDSA with resource-constrained environments (memory, energy, and CPU capability). Public key cryptographic algorithms have been investigated in many schemes to test their applicability in WSN. For example, Wander et al. (2005) presented a study in energy for the public key cryptography (ECC/ECDSA, RSA) on sensor node Mica2dot with Atmel ATmegal 128L (8-bit). They found that transmission cost is double the receiving cost. They analysed signatures in ECDSA with a key of 160-bit and RSA with a key of 1024-bit. Where signature verification cost in ECDSA is larger than a signature generation while RSA verification is smaller than a signature generation. They noted that ECDSA has less energy cost than RSA. They concluded that ECC/ECDSA is more effective and feasible than RSA in constrained-source devices (WSN) because it generates small keys and certificates with the same security level as RSA. 160-bit ECC/ECDSA and 1024-bit RSA algorithms were also applied on a sensor node MICAz (Wang & Li 2006). The authors used hybrid multiplication to reduce access memory. ECDSA results on MICAz are signature generation=1.3s and signature verification = 2.8s. For the purpose of comparison, the authors also implemented ECDSA on TelosB. MICAz results were slightly less than TelosB's results. The authors proved the possibility of using public key cryptography on WSN. None of the aforementioned researchers addressed the optimal use of public key cryptography whether verification cost in Cluster Head (CH) or computational operations. In our project, we are addressed these problems to obtain lightweight signatures. Consequently, our protocols perform computations with more speed and less time.

Costs of computation and communication in WSN should be efficient when applying security protocols, such as ECC/ECDSA. ECDSA (SHA1) and RSA (AES) were analysed in several types of sensor nodes in terms of energy (communication and computation) and time (Trakadas et al. 2008). ECDSA uses short keys

(160-bit), which reduces memory, computation, energy and data size transmitted, and thus is better than the RSA. De Meulenaer et al. (2008) discussed the cost evaluation of energy (communication and computation) on the WSN through the symmetric key distribution protocol and public key agreement protocol (asymmetric encryption). They noted that symmetric encryption performs faster than asymmetric cryptography. However, their research did not address security issues in symmetric cryptography. Moreover, Fan & Gong (2012) implemented ECDSA on WSN with the binary field (163-bit). They improved signature verification via cooperation of the adjacent nodes. Also, ECDSA's implementation was presented in the sensor node (Kodali 2013). But, because this node supported 8-bit of the microcontroller, the author modified the SHA1 code from the 32 bits original to 8 bits. Through implementation, the original algorithm is better in size and time than the modified algorithm. The author explained the possibility of using the ECDSA algorithm with the sensor node held 8-bit microcontroller. However, these papers applied public key cryptography with a traditional hash function that increased the computational and communication costs of servers.

To store patient data accurately, data collection schemes should rely on reliable and fast hash algorithms in ECDSA. Lavanya & Natarajan (2017*a*,*b*) have applied the ECDSA algorithm as a light-weight authentication scheme in the WSN. This demonstrates the effectiveness and efficiency of using ECDSA in WSN in terms of security and performance. Staudemeyer et al. (2018) designed an ECC/ECDSA-based scheme to provide privacy in WSN. However, they did not provide a performance analysis of the security algorithms during exchange of data in the WSN. Malathy et al. (2018) focused on the efficiency of transmission in WSN to extend the lifetime of sensor nodes with the use of ECDSA and generated message digest (MD) with data. Their scheme relied on a colony optimization scheme to save energy in the WSN. But, it did not support privacy parameters during data transfer. Sharavanan et al. (2018) proposed a scheme to monitor the heterogeneous network environments in WSN and protect the medical information of patients using ECDSA. Unfortunately, their scheme addressed only the computation processes of transport. It did not address the complicated computation processes that generate and verify the signature in ECDSA. The main problem with these studies is that they do not provide mechanisms to prevent localisation attack such as Sybil. Our project includes a mechanism to hide sensor location. As a result, the server receives accurate data from sensors without illegitimate changing.

Recently, Sui & de Meer (2019) designed a data aggregation scheme that focused on computation in demand-response management to improve performance and

security efficiency. Their scheme was based on the identity signature (Bilinear Map) to protect information and data aggregated by integration and authentication. Hathaliya et al. (2019) have proposed an elliptic curve cryptography (160 bits) scheme to encrypt and authenticate patients' biometric properties. They used wearable sensors to collect patient data and used a mobile device to send and store this data in the medical repository (cloud server). Finally, Furtak et al. (2019) designed a framework based on RSA-2048 bits and trusted modules to secure the sensors' domain and prevent unauthorised threats. They categorised sensors into the master, replica and gateway categories in the network area and data structure in the sensor memory. In their framework, security procedures for the domain and sensor were used to support both integrity and authentication. Moreover, many researchers (Kittur & Pais 2019, Kuang et al. 2019, Marino et al. 2019, Zhao et al. 2019, Liu et al. 2019) have pointed out that ECDSA is particularly appropriate for authentication and authorisation schemes because it performs the lightweight processes during security procedures.
Many recent studies (Xia et al. 2016, Rasjid et al. 2017, Chiriaco et al. 2017, Merrill 2017, Yang et al. 2017, Giechaskiel et al. 2018, Brockmann 2018, Park & Kim 2018) have also pointed out that SHA1 suffers from collision, preimage and second preimage attacks. However, no schemes addressed SHA1 performance and security (collision, preimage and second preimage) problems in ECDSA. Furthermore, these schemes did not support the protection of signatures transferred between sensors and server. Our storage scheme uses a lightweight hash function and signature camouflage to secure security parameters.

2.4.2 User Authentication to HC Applications

This section briefly discusses the design of authentication schemes to secure healthcare users in the health network as suggested by existing studies, (He & Zeadally 2015, Giri et al. 2015, Li et al. 2016, Farash et al. 2016, Kumar et al. 2016, Jiang et al. 2016, Rajput, Abbas, Wang, Eun & Oh 2016, Das et al. 2017, Chandrakar & Om 2017, Nizzi et al. 2019, El-Tawab et al. 2019), and highlights their shortcomings.

Authentication scheme becomes insecure when using the same IDs with unreliable key size in all authentication phases. He & Zeadally (2015) proposed an authentication scheme based on ECC and advanced encryption standards (AES) algorithms. Their scheme uses three entities: user, server, controller. Their scheme accomplishes registration, login, and authentication phases. The authors claimed that their scheme fulfills many requirements, such as mutual authentication, anonymity, and forward confidentiality. The problem with their scheme is that user and controller identities are statically sent to all three entities. If the attacker can penetrate the encryption, he/she can see that the user and controller identities related. The attacker can then generate a random number, temporary key, timestamp, message authentication code. After that, he/she encrypts the user and controller identities and obtains the message authentication code. Then, the attacker sends a message to the network to become a legitimate and authenticated user. Their scheme also used a 160-bit key with ECC, which is considered unreliable by trusted institutions, such as NIST. Our authentication scheme does not exchange real information for user/device between client and server.

The storage of medical records (data and information) on a single server represents a serious risk in terms of performance and security. Farash et al. (2016) proposed an authentication scheme based on ECC for healthcare environments. They claimed that their scheme provides forward secrecy. Their scheme accomplishes two stages: setup, and authentication. It provides authentication during the exchange of messages between the server and RFID's tag. However, their scheme shows that the information (identities) and data are stored on a single server. When the server is hacked, the users' information and data are exposed to the detection, tampering, and modification.

Jiang et al. (2016) also designed a three-factor (biometric, smart card and password) authentication protocol to protect e-health clouds. Their scheme protects authentication requests from impersonation attacks and off-line password guessing if a mobile device is lost or stolen. Their scheme relies on ECC to support the confidentiality and authentication of healthcare users. They used a fuzzy extractor to keep the biometric secret. But, this scheme relied on a single server to authenticate users, which is an attacker's target. In addition, it performs seven hash operations that can exhaust the single server capabilities if the network has a huge number of healthcare users, especially if it is not using a lightweight hash. Our authentication scheme uses a lightweight hash that reduces running time and increases the speed of computation operations. Consequently, our scheme reduces the burden on servers.

Leakage of information of legitimate users or using unnecessarily large keys destroys the authentication process. An authentication protocol was proposed to protect patients' passwords by RSA against off-line password guessing attacks (Giri et al. 2015). Their scheme consists of five phases: initial, registration, login, authentication, and password change. The main problem in their scheme is that the authors used RSA with the 1024 key. This algorithm affects the performance of

huge healthcare networks. Many schemes recommend using ECC (Timpner et al. 2016, Sojka-Piotrowska & Langendoerfer 2017, Meddah et al. 2017, AbdAllah et al. 2018) as the ECC-160 is equivalent to RSA-1024 with the same security level. Also, their protocol suffers from sending an ID clearly from client to server at the registration phase. This case causes authentication information to be detected for analysis attacks, leaking any information that an attacker could use to disclose authentication information.
Furthermore, the ECC and a Petri Nets model was proposed to achieve an authentication requirement to protect healthcare applications through the mobile cloud (Kumar et al. 2016). This scheme consists of two phases initial setup and authentication. The authors claimed that their scheme is resistant to attacks of eavesdropping, tracking, replay, spoofing, and cloning. However, their scheme did not address the issue of steal/loss of tag or device and internal attacks that are more serious than external attacks in accessing patient data. They gave no indication of the signature algorithm used to ensure integrity. The other problem is that the tag's ID explicitly sends from server to tag, which makes it easier for the attacker to parse the authentication request. Our authentication scheme adopts lightweight public key cryptography, fast hash function and anti-leakage information instead of tradition public key cryptography that is vulnerable to penetration.

HC applications are continuously expanding and become vulnerable to failing if they are not able to offer scalability. Using shared-key to implement authentication mechanism was designed to prevent known attacks, especially DoS attacks (Li et al. 2016). They used the wrong password detection mechanism to reduce the risk of DoS attacks. However, the registration phase of this scheme is not reliable if the ID of patients is sent in an unsafe channel. This research also suffers from scalability because of the use of the single shared-key mechanism that needs protection from all parties.
Das et al. (2017) provided a user authentication scheme for healthcare applications based on AES and secure hash algorithm (SHA1). They used user biometrics and the anonymity feature to repel attacks, such as replay, MITM, and privileged insider. However, symmetric encryption suffers from the problem of scalability, as well as the difficulty of managing the single secret key. Their scheme will suffer from key management problems if applied to a large health institution with hundreds of users whose accounts must be managed during data changes such as additions and deletions. Furthermore, the attacker can submit a forgery attack on the login message if it detects the single secret key. Our protocols depend on public key encryption to solve the scalability problem and key management. Also, we hide the private key with security parameters to prevent attackers from detecting the

private key even if they can penetrate the client's device.

Pseudonym and mutual authentication are important mechanisms to build robust authentication. Cloud-assisted conditional privacy preserving authentication (CACPPA) (Rajput, Abbas, Wang, Eun & Oh 2016) have been proposed to authenticate a network's nodes. This scheme used the elliptic curve integrated encryption scheme (ECIES) and ECDSA algorithms with a timestamp and pseudonym integration to perform the authentication process. The problem with this scheme is that it does not provide mutual authentication to prevent an attack from a counterfeit party. Neither did the authors explain the size of the keys in the algorithms to make sure their scheme was able to repel various attacks. Furthermore, a single pseudonym cannot separate the link to real information to prevent analysis and tracking attacks for authentication requests. Chandrakar & Om (2017) provided an authentication scheme based on ECC and hash. Their scheme was supported on several servers in user authentication with three factors (biometric, smart card, and password). The user could connect to any server to perform the authentication process. In this scheme, the authors did not specify which hash algorithm was used and the size of the message digest (MD). These procedures are essential to repel attacks such as collision, preimage and second preimage. Using multiple servers means that the same user information is stored on more than one server. As a result, penetrating any server can cause user information to be detected or modified. Moreover, this scheme did not use a mechanism to prevent the association of real user information with an authentication request, such as pseudonyms. Our authentication scheme uses a reliable multi pseudonym and mutual authentication to ensure only legitimate users are connected to the network.

Recently, Nizzi et al. (2019) proposed the address shuffling algorithm (AShA) method to protect devices' MACs when these addresses transferred from sender to recipient. They used keys and hashes to shuffle MAC addresses in the network and prevent intruders from executing collisions or privacy breaches. However, their method is vulnerable to security threats especially as it does not provide resistance against the second preimage. Similarly, El-Tawab et al. (2019) relied on MAC addresses to protect users' privacy. They used techniques such as MAC randomisation and a hash function (SHA-256) to prevent the penetration of users' devices. But, these techniques are still weak in ensuring the authenticity of MAC addresses. Our authentication scheme does not need additional computation operations by shuffling/randomisation for MAC because it uses the check MAC mechanism. It simultaneously provides high-speed computations and reliable security.

2.4.3 User Authorisation to HC Application

This section discusses related works about authorisation (Chadwick et al. 2006, Riedl et al. 2008, Quantin et al. 2011, Gajanayake et al. 2014, Sun et al. 2011, Jo & Chung 2015, Seol et al. 2018, Wang et al. 2019, Shafeeq et al. 2019), and highlights their shortcomings.

Authorisation schemes require a unique signature in authorisation policy for each user to prevent the disclosure of user IDs to unauthorised medical staff. As early as 2006, the PERMIS project was proposed by (Chadwick et al. 2006) with the role-based access control (RBAC) model. It described the conceptual authorisation of the credential validation service (CVS) before the approval stage of access decisions for the resource as well as the distributed management of the credentials. However, the PERMIS system does inadequately protect the CVS. PERMIS also suffers from the problem of inheriting managers for all the attributes of their followers (hospital department managers or specialist doctors who inherit all their practitioners' attributes and thus have access to patient data, which can lead to significant internal attacks). In addition, their project uses one signature of a public key cryptography (PKC#12) file for policies and attributes. Our authorisation scheme provides a secure decision engine and flexibility to create/manage policies with different signatures for each user.

HC server resources, such as memory and processing speed, are crucial in implementing the authorisation scheme in large health environments. The pseudonymization of information for privacy in an e-health (PIPE) project is designed to protect health data in the EHR through a layered system. PIPE includes many keys, such as an external key pair, an internal key pair, a symmetric key pair, and a shared key. It relied on RBAC to protect the keys (Riedl et al. 2008). This scheme used the Shamir scheme as a backup mechanism to retrieve patient keys in the case of the loss of the smart card. But, this scheme did not explain the symmetric and asymmetric encryption algorithms used to generate pseudonyms for users. Also, the scheme increases the complexity of the server system with the use of many keys, especially if the scheme is used by a large health institution. In addition, the server must use the keystore to store the keys, and this requires protection and storage space on the server. Our authorisation scheme does not suffer from memory and speed problems because it does not depend on different complex algorithms and many keys.

A robust authorisation scheme requires uniform structures and contexts in requests and responses securely without complex operations on the HC server. Quantin et al. (2011) suggested using non-central medical records to eliminate issues of standardization and structure in data access requests. They used two medical record search engines (MRSEs) and one data aggregator. The first search engine was used to authorise healthcare providers. The second search engine was used to authorise patients. In addition, a hash was used to increase the pseudonym of patients' IDs. However, this scheme suffered from the use of a single aggregator that was similar to the dataset on the central server, which is vulnerable to attacks. Also, patient data comes from different sources and have different structures and standards. This difference causes a burden on the aggregator. Moreover, the authors used the RSA's encryption algorithm, and this algorithm uses a large key size of 1024 bits, which causes a burden on the server. Also, the aggregator needs time and storage to convert the data into a single context. Furthermore, their scheme suffered from collision and doubloon problems due to the transference and transformation of patient data contexts. Our authorisation scheme does not suffer from the costs of complex operations because it separates duties between three servers which reduces performance overheads on each server.

Internal attacks on the repository or transfer of medical records clearly during the network is a serious risk to patients' health. Gajanayake et al. (2014) integrated four access control models (discretionary [DAC], mandatory [MAC], role-based [RBAC], and purpose-based access control [PBAC]). Their target is to obtain a single model that limits illegal user access control of the medical record. They relied on the sensitivity label for data in the hierarchical structure of the database. They also suggested defining the purpose of accessing patient data. However, their scheme addressed only the doctor and the patient and did not address different classes of healthcare providers. Furthermore, data and requests are clearly transmitted between client and server. In addition, Jo & Chung (2015) proposed an extensible markup language (XML) access control system (XACS) that enables users to access specific elements in an XML document. This system relies on removing certain parts of the XML document to allow users who are authorised to see certain parts of an XML document. However, requester information is transmitted explicitly over the Internet to a server, which makes it easier for an attacker to penetrate the privacy of users. In addition, it does not address internal attacks that are applied by legitimate users even though certain parts of the XML document have been removed. Our authorisation scheme provides robust privacy by ensuring that request, policies, and data do not contain the real personal information of users.

Cryptographic processes for patient data are extremely expensive on the server compared with the mechanisms of authorisation policies and pseudonyms in the authorisation scheme. The healthcare system for patient privacy (HCPP) project was designed for the EHR to protect the privacy of patient data (Sun et al. 2011). Researchers focused on an emergency scenario regarding the protection of patient data. They used a backup mechanism that allows the doctor to access patients' health information without access to confidential parameters. However, this search relies on encrypting all patient data. When a client wants to access patient data, the server uses a keyword (searchable symmetric encryption (SSE)) to perform an encrypted data-mining operation. This process is exceedingly expensive for the server for two reasons. First, the server must encrypt the entire massive database with the continuous addition of new records and, second, the server must continuously mine each access request. In addition, their system does not support levels of authorisation and privileges (roles and attributes) that are more secure in providing privacy to patient records. Also, researchers have reported that the patient has not been exposed to collusion because the patient does not attack himself. But this is not true because there are impersonation attacks that do the job without the theft or loss of the patient's device. Moreover, this research did not specify the type of encryption algorithm used, which is greatly important for security and server performance, and it addressed only emergency cases.

Seol et al. (2018) proposed an access control model based on partial encryption and XML signing in EHR's documents within a cloud environment. Their model is supported in two phases: the first phase is access control using extensible access control markup language (XACML) and the second is to encrypt and sign data with XML. However, the cloud environment presents multiple security and privacy problems in the EHR system because of the distributed exchange of data between the various health centres. In addition, their scheme uses encryption in XML requests and responses. Encryption processes will be extremely costly for legitimate entities exchanges in healthcare systems. Also, in the first phase, requests and responses are clearly sent between legitimate parties and, therefore, will be exposed to attack. Neither did they address the pseudonym mechanism that prevents access to real user information. These researches suffered from encryption costs while our authorisation scheme uses a fast mechanism such as random pseudonyms instead of encryption overheads.

Recently, Wang et al. (2019) relied on attribute based encryption (ABE) and searchable encryption to authorise users' access to patient data. Their scheme focuses on supporting privacy and efficiency in hidden policy with search keywords. In addition, their scheme is based on constant expenses for encryption and the secret key. However, their scheme suffers from internal attacks in addition to the heavy costs to access the data. In addition, Shafeeq et al. (2019) has proposed a decentralized authorisation scheme to grant users' access to patient data. They used XACML, Merkle tree-based signature and symmetric encryption to prevent unauthorised users from penetrating the repository. Although their scheme provides efficient management by XACML and message integrity support by multi-signature, it is vulnerable to threats of single symmetric key penetration as well as the storage expenses required for the Merkle tree signatures. Our authorisation scheme uses public key cryptography and nonce that mean the generation of different signatures for each authorisation request. In addition, our authorisation scheme does not require the storage costs of the Merkle tree and expenses associated with accessing the dataset by search keywords.

2.5 Summary of the Chapter

In this chapter, we have presented details about HC application history, and HC systems such as EMR and EHR. We have also discussed security and privacy issues when medical records are transferred by a communication channel or saved on an EMR/EHR server. Security protocols for HC applications have been investigated. Finally, existing schemes' gaps have been described as a basis for building new data collecting, authentication and authorisation schemes.

Chapter 3: Designing a Secure and WSN Based Healthcare System Application

The development of efficient and low-cost HC application systems involves three categories of issues: ethical issues, medical/health issues, system design and security issues. In this chapter, we will focus on the last one; that is, on the technical and security considerations for the development of HC applications. More specifically these developments are:

- General architecture and EMR/EHR repositories of the proposed HC application
- Techniques proposed for data collection, authentication and authorisation of the proposed HC application
- General proposed network model of HC application
- General methodology for the proposed HC application.

3.1 General Architecture of Proposed HC Application

Figure 3.1 shows our project architecture. The proposed HC application system is shown in Figure 3.2. There are three major components: data collection by WSNs, EHR/EMR storage or repositories, and an online system of HC application for professionals. Our study will carry out the investigations into three aspects as follows:

1. Secure and efficient application using algorithms/protocols to integrate HWSN and EMR in the HC industry

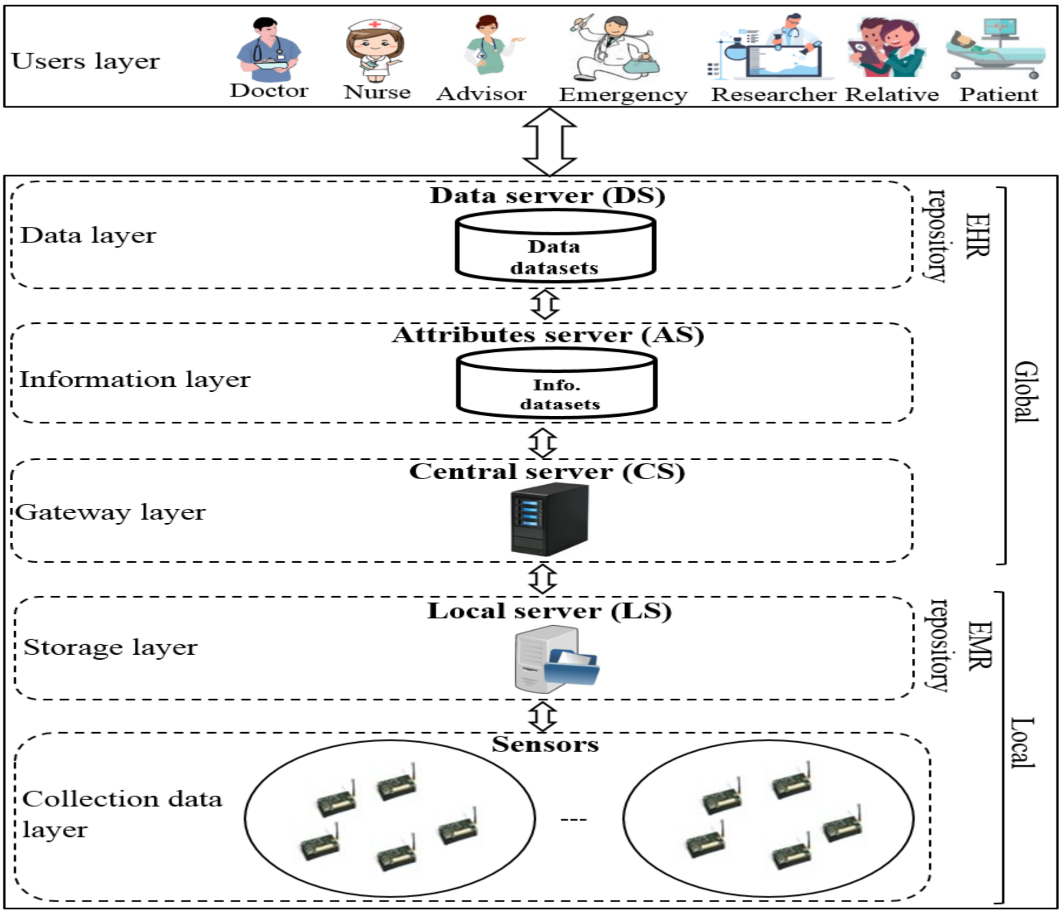


Figure 3.1: General architecture of proposed HC application

2. Techniques to provide an authentication mechanism to legitimate users in the HC application

3. Techniques protecting users' privacy, and increasing the users' confidence in the advanced HC systems in access EHR repository.

3.2 EMR/EHR Repository

Disclosure of medical records in the EMR/EHR's repository is a result of weak security in these systems. For many years, electronic health record (EHR) and electronic medical record (EMR) systems have been extremely useful for managing patients' data. These systems are widely disseminated in the health sector (Sarkar 2017). The main problem with these systems is how to maintain the security and privacy of sensitive patient data and information. Furthermore, protecting EMRs/EHRs is essential for ensuring the stability of a patient's condition. Due to their inability to protect the records from unauthenticated and unauthorised users, some of current EMR/EHR systems fail to provide security and privacy for PHI

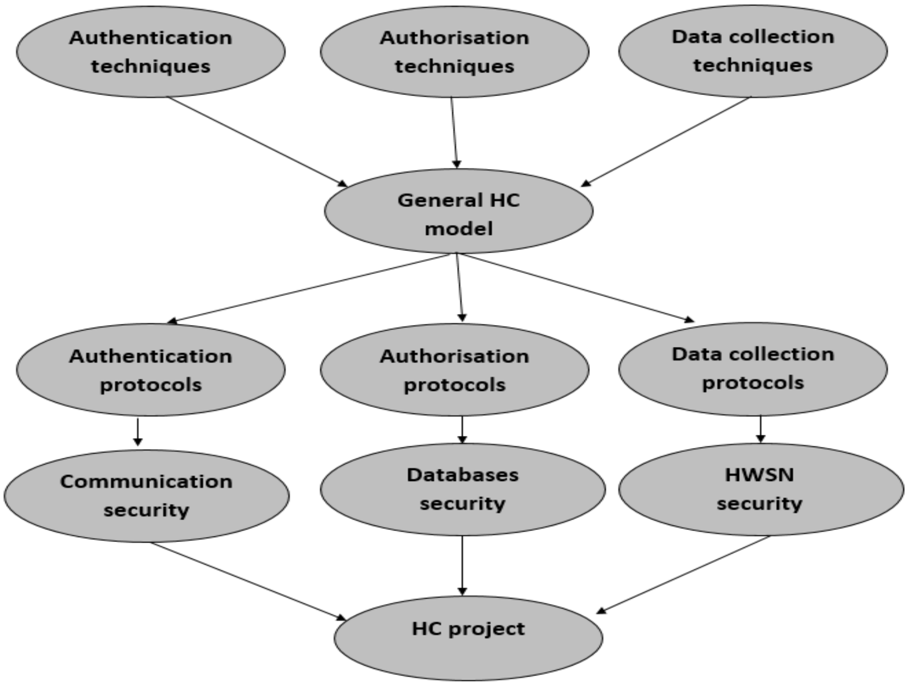
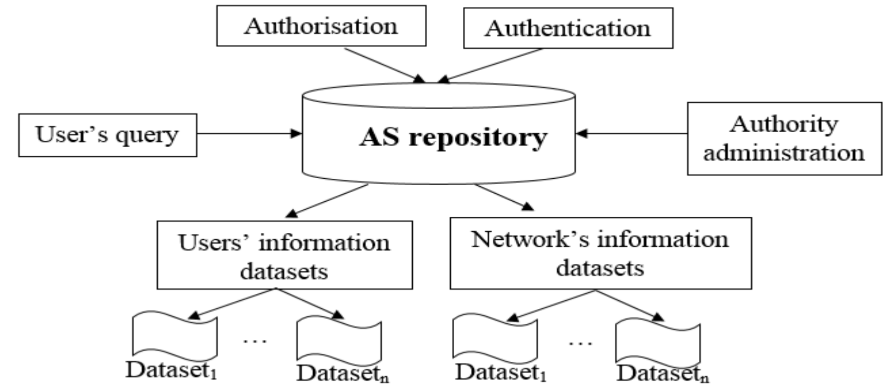


Figure 3.2: The proposed project system

(Entzeridou et al. 2018, Sittig et al. 2018). They also possibly allow authorised users to exceed their specific privileges. Thus, those systems are not a trustworthy source and are undesirable for patients and healthcare providers. Various attacks expose patients' data to malicious tampering or destruction. For example, access to the EMR/EHR repository (data and information) without authentication mechanisms and authorisation policies exposes them to penetration by adversaries (Liu, Xia, Yang & Yang 2018, Manogaran et al. 2018).

Penetration of the EMR/EHR repository may be prevented by improved system security. Many medical records transfer schemes have been developed with data collecting, authentication and authorisation mechanisms. These schemes ensure that only legitimate users are authenticated and authorised to connect. Nonetheless, these schemes still suffer from vulnerable security as discussed in Section 2.4. Specifically, security and privacy precautions should be raised for specific categories of users, doctor advisors, physician-researchers, emergency doctors, and patients' relatives. Presently, these users can break into electronic systems and even violate patients' privacy. The main reason for these breaches is the privileges granted to them or the inadequate security and privacy mechanisms of these systems (Madhavi & Lincke 2018). Providing mechanisms to collect data, authenticate and authorise the users is an essential security requirement to prevent both external and internal attackers. This requirement acts as a barrier against penetrating patients' identities and revealing their sensitive data. Therefore, to address EMR/EHR repository



User's information

Names	Roles	Address	Phone	Gender	Email	DoB	Postal	Primary medical degree
Sara	Patient	-	-	F	-	-	-	-
Simon	Doctor	-	-	M	-	-	-	-
Hawa	Researcher	-	-	F	-	-	-	-
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.

Network's information

Pseudonyms	OTP	Passwords	Signatures	Policies	Requests	Responses	Shamir-policies	Private/public keys
p1	-	-	-	-	-	-	-	-
p2	-	-	-	-	-	-	-	-
.	-	-	-	-	-	-	-	-
.								
.								

Figure 3.3: Information repositories

issues, we develop data collecting, authentication and authorisation schemes that provide security and privacy when collecting or accessing the EMR/EHR repository.

The repository is an important component of HC applications because it provides management and regulation. It is an efficient way to handle EMR/EHR user data and information smoothly and flexibly. It enables patients and providers to obtain health data without incurring management and storage problems. We use an EMR/EHR repository in our project because it offers several advantages such as ease of handling datasets, alleviation of technical details in EMR/EHR systems and efficient storage/management of datasets.
Our project includes several repositories, for instance, Attributes Server (AS) includes information repositories such as user information and network information as shown in Figure 3.3. Also, the Data Server (DS) includes a data repository as shown in Figure 3.4. DS stores data previously received from the HWSN in the repository. When AS receives an authorisation request from users, AS uses authorisation policies to verify a user's information such as name, and network's information such as signature. AS sends an authentication request to the DS to receive data from the policy retrieval point (RPR) repository.

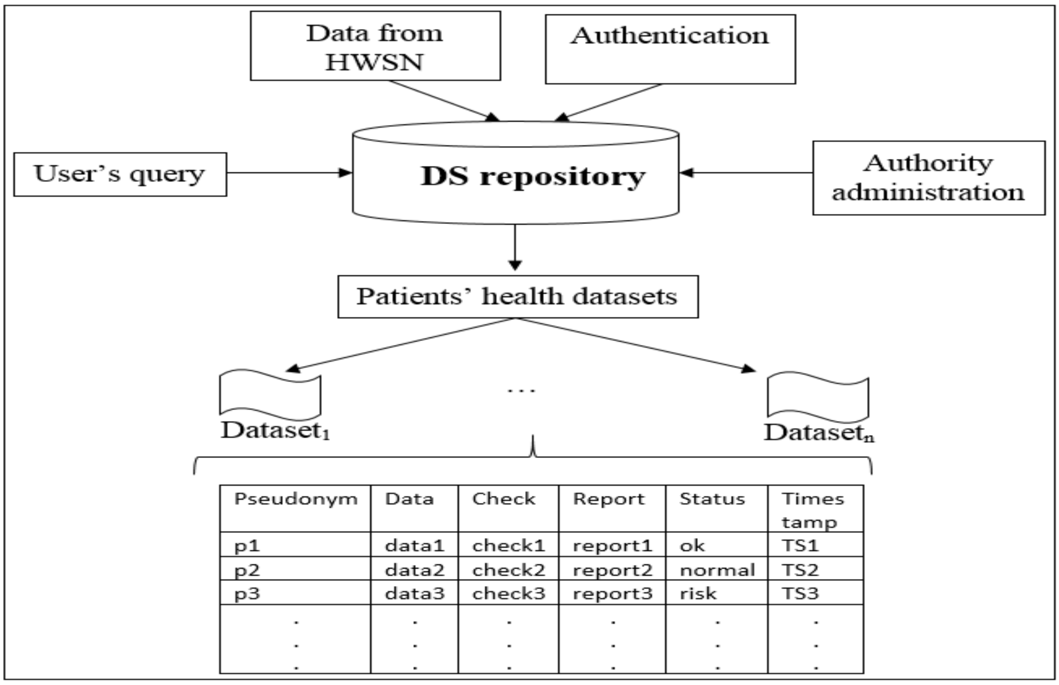


Figure 3.4: Data repository

3.3 Storage Scheme in the Proposed HC Application

In this section, we will illustrate the storage scheme in the proposed HC application. To the best of our knowledge, there are no other studies addressing this issue in the field.

3.3.1 Threat to Storage Scheme

Building a threat model in HWSN is important to identify serious attacks on patients' data and subsequent disclosure. HWSN provides important services to the health sector compared to traditional networks, such as local area network (LAN) and metropolitan area network (MAN), but they are more vulnerable than the latter. These networks rely on self-organisation and synchronization to increase the flexible communication of sensor nodes, but HWSN suffers from a security vulnerability. Due to wireless radio signals in WSN, it is easy for an attacker to access data transmitted among sensors, Cluster Heads (CHs) and Local Server (LS). These networks are targeted for many attacks that exploit resource-constrained, untrusty communication and unattended processes. HWSN threats are as follows:

- The attacker performs a MITM attack to modify or replay attack to resend the data to the CH/LS. The attacker's aim is to use his/her device as legitimate sensors in the network

- The attacker can execute a DoS attack on the CH/LS. This attack exploits a heavy transmission of duplicate or counterfeit data to destroy the HWSN
- The attacker can apply several types of localisation attacks such as:
  - The attacker uses several IDs of legitimate sensors with incorrect data (such as wrong and duplicate IDs) and sends them to the network (Sybil attack)
  - The attacker uses more than a fake node to transfer data between different locations via a tunnel that connects counterfeit sensors. This attack leads to a disguise of communication between the legitimate sensors (Wormhole attack)
  - The attacker uses a malicious sensor node to attract all data sent from the sensors. Then, he/she prevents the correct data from accessing to the LS (Sinkhole attack)
- The attacker performs an attack to penetrate the EMR repository in the LS, to access the patient's data and reveal their identities
- The attacker can launch an eavesdropping attack to obtain patients' data, and then perform an analysis of these data to detect the linkability between data, information, and pseudonyms
- The attacker can copy a legitimate sensor ID in more than one counterfeit sensor. These counterfeit nodes send modified data to the network (node replication attack)
- Collision, preimage and second preimage attacks can be implemented to change signatures and data transferred between network's devices.

3.3.2 SHA Hash Function

During the implementation of the proposed storage scheme in the LS, another technique is used: the traditional hashing algorithms in ECDSA, such as SHA1. For more details of this traditional hashing function, see Appendix A.

3.3.3 BLAKE Hash Function

During the implementation of the proposed storage scheme in the LS, another technique to speed up the procedures of the storage scheme is to use one of the lightweight hashing algorithms such as BLAKE. This hash function provides features, such as simplicity, speed, memory availability and parallel procedures in hardware/software implementations. These features qualify it for efficient use in restricted source devices. It is capable of preventing hash function attacks. For more details of these lightweight hashing functions, see Appendix A.

3.3.4 De-identification Mechanism

Encryption and k-anonymity mechanisms are applied to hide patients' data but, these mechanisms suffer from serious drawbacks. For instance, encryption of collected data (Neubauer & Heurix 2011) has the following drawbacks:

1. Temporary HC providers such as a researcher doctor, will not get benefit from the encrypted data, and if he/she is able to get the collected data by the decryption process, this is a security weakness in the HWSN system
2. Huge datasets' encryption burdens the LS system, causing complex operations and processor power consumption (Zhou et al. 2017)
3. The datasets of collected data perform intensive and continual operations on medical records, such as add, delete and edit. If the records are encrypted, this will multiply the burden on the LS (Vatsalan et al. 2017)
4. Encryption can contain implicitly direct information about the patients. A breach of this encryption will expose patients' information to the intruders (Bogos et al. 2018).

The k-anonymity of collected data suffers from the following:

1. The removal process of all the patients' information obstructs the HC provider from dealing with linked patient data (Neubauer & Heurix 2011)
2. Inserting a large set of false medical records, namely, greatly reduplication the dataset size. This process consumes LS resources, particularly the intensive and continual access of the datasets by HC providers.

To address these disadvantages, we use random pseudonyms in REISCH's requests to hide the correlation of patients' information with data. The medical records transmitted/stored among the sensors, CHs and LS do not contain any of the patients' real information. This mechanism prevents intruders from identifying patients' IDs. In addition, this mechanism is fast and does not need complex operations. When the EMR system wants to add a new HC provider/patient, the REISCH sends a request to the remote servers (Central Server (CS) and AS) which provides the LS with the required information for updating random pseudonyms. These random pseudonyms are linked with the users' IDs. This mechanism enables sensors to access and store a specific patient's data without exceeding granted privileges.

3.3.5 Efficient HWSN Data Management Using XML

The other important part of the proposed EMR system is the repository. Repositories contain data in various contexts since these systems have difficulties dealing with the different coordinates for data. The extensible access control (XML) is considered convenient for the exchange of various data via different environments. XML is the symbolic, simple and flexible language designed to manage, describe and exchange data across the Internet. It divides data into a useful form of information through data organisation, the purpose of sharing data across different systems and stored in the dataset. Also, XML has several features that make it suitable for data management, such as support for unicode, the representation of computer data structures (trees, records and lists), use a formula read by both human and computer. However, XML should support the security mechanisms to provide different levels of protection of sensitive data in the whole or part of the XML document (Jo & Chung 2015).

But this information needs mechanisms to identify the arrival of unauthorised users to protect patients' data. Patient data transmitted between sensors (nodes and CHs) and network devices (such as a nurse and a LS device) need data management algorithms to maintain both performance and security at the same time. EMR, including patients' confidential data and private information, needs to be accessed by HC professionals. Thus, sharing such EMR without breaching a patient's privacy requires EMR management in an efficient and secure manner. XML technology has begun showing its superiority in the exchange of complex data over different systems.

3.3.6 Homomorphic Scheme

Homomorphic is a mechanism for merging all messages and signatures together to improve both performance efficiency and security. This mechanism consists of many types such as linearly, polynomial, fully and aggregate signature (Emmanuel et al. 2018). In this study, we focus on the aggregate signature because it deals with multi-sensors signatures, messages and different private keys depending on different devices, such as sensors. Furthermore, this process is extremely suitable for multihop-based networks during the integration of signatures in a single signature. We assume that we have a range of messages $M = \{m_1, ..., m_n\}$ and a range signatures $S = \{s_1, ..., s_n\}$, M contains all of a group's messages, S is one signature for all signatures, A is an aggregate function and V is a verification function. The process of homomorphic signatures is as follows:

- Each device generates K_{pr} and K_{pu} keys and broadcasts the K_{pu} keys to the network members

- Each device signs the m by the signature algorithm, which includes the device's ID, message and private key $s(K_{pr}, m_i, ID)$
- The aggregation procedure in the intermediate nodes, such as CH relies on A to collect all public keys, messages and signatures $A(K_{pu_1}, ..., K_{pu_n}; m_1, ..., m_n; s_1, ...s_n)$
- The verification procedure will be in the final entity, such as LS, which uses V to validate the signatures $V(K_{pu_1}, ..., K_{pu_n}; m_1, ..., m_n; s_1, ...s_n)$. If the verification process fails, it means that the data integrity operation is incorrect.

The homomorphic aggregate signature scheme is important to support the performance of network devices by making the intermediate nodes, such as CH performs a single signature process for all members' signatures of the group rather than the signature verification process (the ECDSA verification process consumes more time and energy than the signature process) (Luo et al. 2019). In addition, homomorphic increases security measures in preventing the tracking of patients' information and data or changing signatures of legitimate network devices (Kapusta et al. 2019).

3.4 Elliptic Curve Cryptography for Authentication

First, we focus on the security aspect of authenticating legitimate users. To ensure only legitimate users are associated with the HC application network, our scheme includes a set of techniques (Elliptic Curve Integrated Encryption Scheme (ECIES), PHOTON, one-time password, mutual authentication, and media access control (MAC) address) to validate the authentication request. Efficiency and security are important determinants of health applications. Sensitive patient records require protection from intruders and at the same time require efficiency because health applications involve frequent and continuous exchanges between a large number of users (patients and professionals). Our scheme relies on algorithms that provide lightweight operations and a high-security level for encryption and signature operations. This section describes the threat model and the basic concepts of these techniques.

3.4.1 Threat to Authentication Scheme

In the architecture of our proposed HC application, the AS is responsible for users' authenticity. The AS must be authentication proof against a number of attacks. In this section, we list possible attacks against the authentication scheme used in the AS of the proposed HC system. AS is trustworthy. It resists repository penetration attacks. This server contains only users' information datasets. We impose threats in our authentication model as follows:

- The attacker can steal the client application and its files to analyse the data, retrieve the parameters, and reveal the secret key of the user. Then, the attacker can use this application on different devices
- The attacker can listen for authentication requests in the insecure environment and execute interception, replay, MITM, and modification attacks. The attacker's goal is to become a legitimate user in the network
- The attacker can execute a forgery or masquerading attack in an attempt to penetrate the authentication process
- A legitimate user (such as a doctor, nurse, or patient) can perform a privileged insider attack based on his/her legitimacy in the network
- The attacker can successfully guess the real username, password, role and pseudonym associated with it. This guess can be accomplished during an intensive analysis of many authentication requests.

3.4.2 Elliptic Curve Integrated Encryption Scheme (ECIES)

To implement the authentication scheme in the AS, Elliptic Curve Cryptography (ECC) will be adopted. As ECC is no longer dependent on the discrete logarithm problem (DLP); ECC can be perform very quickly. In this section, we will provide an introduction to ECC and its applications. For more details of ECC cryptography, see Appendix B.

3.4.3 Lightweight Hash-Function Algorithm

During the implementation of the proposed authentication scheme in the AS, another technique to speed up the authentication is to use the best of the lightweight hashing algorithms which is PHOTON-256. This algorithm is extremely beneficial in information signature processes, providing high-performance efficiency and satisfying the security principles of signature compared to traditional hash function algorithms. This lightweight function is exceedingly convenient for our scheme as a signing mechanism in our authentication protocols to provide information integrity by generating a robust and reliable signature. For more details of these lightweight hashing functions, see Appendix B.

3.4.4 One Time Password (OTP)

OTP is a way to authenticate legitimate users by generating a passcode or nonce only once in a specified time. It becomes not applicable in the next times for authentication. OTP is an effective method for authenticating users in HC applications if used with robust encryption and signature technologies. Using a static password, or nonce, without other authentication mechanisms makes a system weak and prone to attack. Therefore, OTP significantly provides support to the authentication process. This mechanism prevents many attacks, such as replay, MITM, and guessing (Thiranant et al. 2015, Chen et al. 2017). The attacker cannot use this passcode or nonce to connect to the network later. Each client receives a random OTP from the authorities provider via a secure channel, and this is used for the login session. The client sends the OTP as part of an authentication request. If the authentication process is valid, the server will delete the OTP from the dataset and will not accept it in the future. OTP is a powerful mechanism to mitigate the risk of hackers' communication in the network.
In this scheme, we apply the OTP to generate a random password (to prevent hackers from expecting the OTP) with the first link to users. This procedure is to ensure only legitimate users are connected to the network. In our authentication scheme, users (patient or provider) only use the OTP once and will not need to use the OTP with name and password in future authentication operations. This mechanism allows the server to record the original physical address in the dataset. Even if the attacker detects the OTP, he will not benefit from it in the next times. We also use other mechanisms (such as timestamp, checking the original physical address and multi pseudonyms) with the OTP.

3.4.5 Mutual Authentication

The network parties (clients and server) exchange requests for authentication remotely (Internet) or locally (WLAN). This process plays an important role in connecting users to HC applications. The authentication request can include many security parameters such as name, password, OTP, nonce, pseudonym, signature, and encryption. A robust authentication protocol should include security requirements that are critical to protecting patients' information and data. In addition, traditional methods of authentication (password and name) are not suitable for health applications (Liu & Chung 2017). In general, two kinds of authentication have been applied in the various projects (as described in Figure 3.5); simple and mutual.

In simple authentication, one party performs the authentication process, such as

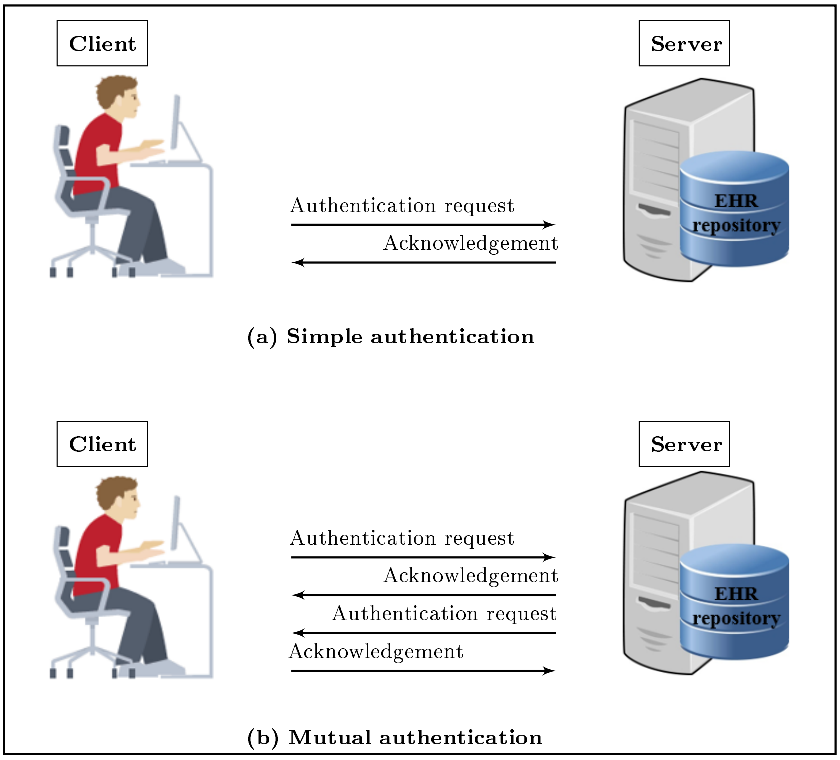


Figure 3.5: Types of authentication schemes

server verifying the authentication request for the client. This type of authentication is vulnerable to attacks such as masquerading, spoofing, and impersonation. The attacker can use his/her device as a fake server to receive all clients requests. Mutual authentication provides a security solution to prevent known attacks. In this type of authentication, each party authenticates the other party and thus prevents counterfeit attacks by both client and server. Our scheme adopts the mechanism of mutual authentication in the preservation of users' information and data.

3.4.6 Media Access Control (MAC) Address

Network devices of all types should contain a hardware card (interface) to connect to the local or global network. Each wire or wireless interface has a media access control (MAC) or physical address that consists of 48-bit. It is divided into six octets and written in hexadecimal, such as "8C: 70: 5A: 41: 49: BC". This address is a unique identifier (no duplication of addresses) for devices that are defined globally and are persistent (Martin et al. 2017). MAC address is the identifier of the device that can connect to the network. This address is used in WLAN networks because it offers advantages, such as reducing costs and speed in access control procedures (Mattos & Duarte 2016). It can be changed programmatically in various operating systems, such as Linux and Windows. In addition, anyone can use Address Resolution Protocol (ARP) to detect the MAC address of another user

in the network (after entering his IP address) (Dallaglio et al. 2015).
The main problem with this address is that the attacker can execute an eavesdropping attack to access the MAC addresses of legitimate devices in the network. Then, he/she selects a legitimate MAC address to use it. For example, an attacker could execute an ARP poisoning or spoofing attack by using a fake identifier of the MAC address (for a legitimate client or server). Consequently, the attacker gains illegal privileges that would enable it to perform other attacks, such as MITM and DOS (Xiao et al. 2016, Masoud et al. 2015). Moreover, randomization operations for the MAC address have become useless in the protection against tracking attacks (Matte et al. 2016, Vanhoef et al. 2016). Therefore, if the server does not have a mechanism to detect MAC address change, the attacker becomes a legitimate user in the network and has access to network resources.

3.5 XACML Techniques for our Authorisation Scheme

The second major component in the proposed HC application, as shown in Figure 3.8, is the EHR repository. Its primary purpose is to store patients' health records or medical records. All kinds of users, including the patients themselves and HC professionals, need to access the repository legally and ethically. In this section, we will introduce the techniques used to authorise the legitimate users.

3.5.1 Threat to Authorisation Scheme

Many serious risks to HC authorisation systems that require the building of a threat model to detect weaknesses in these systems. We assume that attacks can be internal, external, active, and passive. In summary, the existing HC applications are vulnerable to the following attacks:

- The attacker can flood the server with intensive authorisation requests which is to stop the service from legitimate users and destroys the network
- To access patient data and reveal patient identities, the attacker performs an attack to penetrate the repository in the CS
- The attacker performs a MITM attack to modify data and to become a legitimate user in the network
- The attacker sends a fake authorisation request during the execution of a forgery/impersonation attack to gain access to patient data

- The attacker can launch an eavesdropping attack to obtain authorisation requests, and then perform an analysis of these requests to detect the correlation between data, information, and pseudonyms

- The attacker can execute timing attacks by using the time period to reveal user authorisation information.

3.5.2 Elliptic Curve Digital Signature Algorithm (ECDSA)

To implement the authorisation scheme in the AS, the ECDSA will be adopted. It depends on the use of the points on the curve to integrate and sign data. ECDSA uses small parameters which expedites performance of computations, thus reducing time and storage (Sojka-Piotrowska & Langendoerfer 2017). These features are very important for large organisations and constrained-source devices, such as WSN, because these networks require processing power, memory, bandwidth, or power consumption (Dou et al. 2017). In this section, we will provide an introduction to ECDSA and its applications. For more details of the ECDSA algorithm, see Appendix C.

3.5.3 Models of Access Control to the EHR Repository

Any system needs access control models to determine users' access to the data repository. Many access control models, and each one depends on a particular method and set of rules. One of the most distinct access control models is Role-Based Access Control (RBAC). This model relies on the classification of users into roles, and each role has privileges and rights regarding data access (Gajanayake et al. 2014). With RBAC, the security of the system is based on the structure of the system's roles assigned to users (Sánchez et al. 2017). Each role in the system is assigned according to the job of the user in the organisation (Alturki 2017). RBAC was introduced to solve problems with previous access models, such as Mandatory Access Control (MAC) and Discretionary Access Control (DAC). As shown in Figure 3.6, the RBAC model divides users into roles (such as patient, doctor, and researcher). For example, the researcher role can access the data assigned to that role, which enables him/her to develop medical research to find a cure for a medical condition.

In recent years, there has been significant interest in using the Attribute-Based Access Control (ABAC) model for the protection of data privacy. This model is designed to access data more accurately (fine-grained) and securely. It handles user attributes (such as name, address, age, mobile, location, time) to allow users to

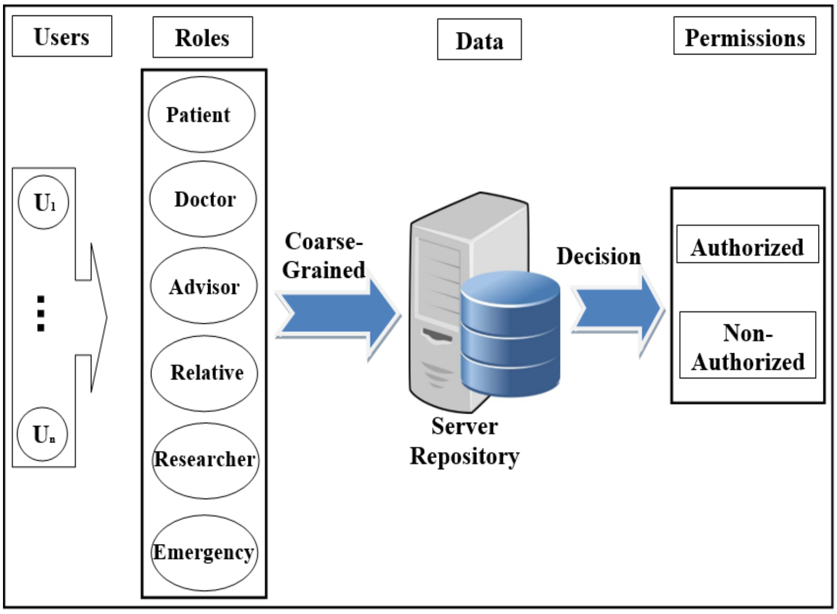


Figure 3.6: Scheme of RBAC model

access the server's repository. ABAC is proposed to go beyond the limits of the rules and design of the most well-known control access models (DAC, MAC, and RBAC) (Jin et al. 2012, Zhang & Zhang 2017). ABAC is a rich model because it deals with a wide range of user attributes. ABAC supports administration, authorisation of context-aware, risk-intelligence, and scalability in various applications, such as the Internet, IoT, Big Data, cloud computing, and VANET (Brossard et al. 2017). The attributes in ABAC are categorized into subject, object, action, and environment. As shown in Figure 3.7, each user has a set of attributes that allows him/her to access data in the server. In our authorisation scheme, we combine ABAC and RBAC to obtain a model that depends on both roles and attributes.

3.5.4 Distributed AC Implementation Technology

The most important component in the proposed EHR system is the EHR repository. Access to data is a major challenge in big data management systems (EHR) that use different techniques. In addition, the exchange of information over the Internet has become essential and requires access authorisation, particularly in HC applications. XACML standards include both access control (authorisation) and data management based on XML in the different systems (Lu & Sinnott 2018). Effectively, XACML offers features for data access and authorisation for the users at the fine-grained level.

Many techniques, such as an Open Authentication (OAuth), XML Access Control Language (XACL), Enterprise Privacy Authorisation Language (EPAL), Open

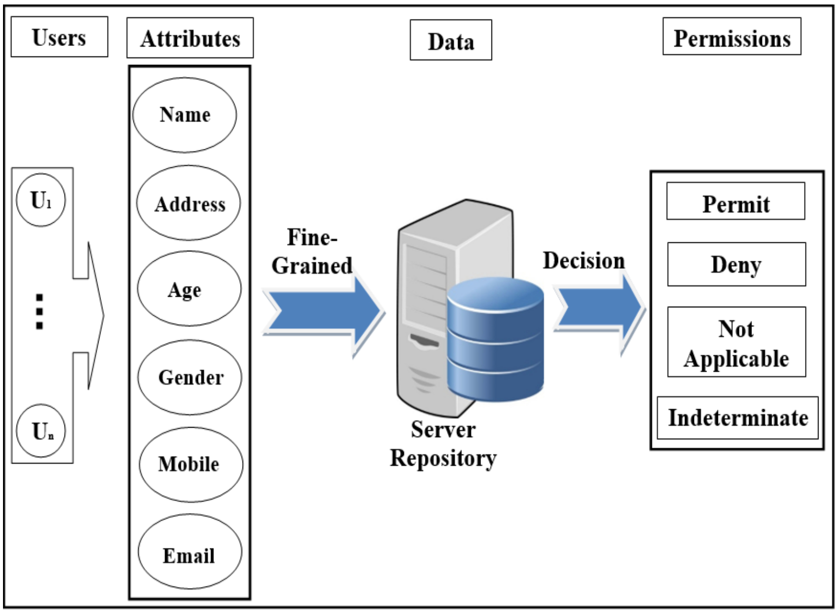


Figure 3.7: Scheme of ABAC model

Digital Rights Language (ODRL), and Security Assertion Markup Language (SAML) have used in authorisation projects. However, OAuth suffers from password management; XACL suffers from few functions and is non-extensible; EPAL does not support multiple subjects, error handling and a hierarchical role; ODRL suffers from common vocabulary, conflict between serialization and does not support combining algorithms; SAML is extremely expensive, difficult to use and uses a single sign-on (SSO), namely, it uses one password for all applications which may be reason for penetration. XACML is the most flexible and effective technique (Grace & Surridge 2017, Beltran et al. 2017, Turkmen et al. 2017). This technology is presented by the organisation for the advancement of structured information standards (OASIS). This standard has many of the features that qualify it for use on the Internet, such as combining policy, combining algorithm, attribute, multiple subjects, policy distribution, implementation independency and obligations (Zhang & Zhang 2017, Turkmen et al. 2017, Deng et al. 2018).

This technique is based on the specific policies first and then on many modules, such as policy enforcement point (PEP), policy decision point (PDP), policy administration point (PAP), policy information point (PIP), and policy retrieval point (PRP) to evaluate the request for access (Calvillo-Arbizu et al. 2014). As shown in Figure 3.8, PEP sends and receive requests and access responses to the repository; PDP evaluates the decision; PAP creates policies based on users' attributes; PIP retrieves users' attributes; and PRP retrieves users' data from the repository. The result of the decision (permit, deny, not applicable, indeterminate) is sent to the subject via PEP (Zhang & Zhang 2017).

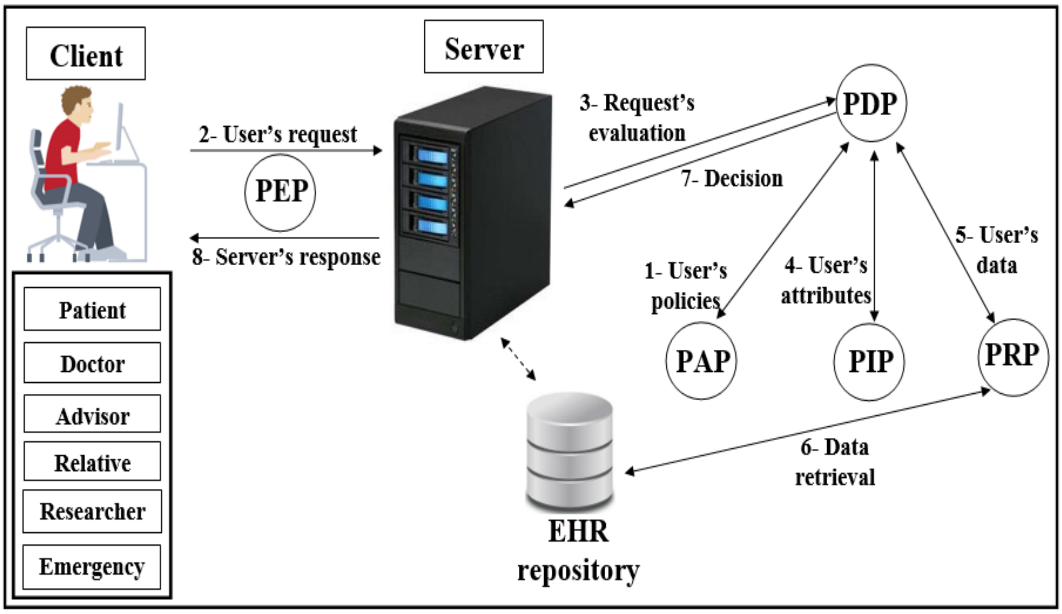


Figure 3.8: Scheme of XACML

3.5.5 Shamir Scheme

The secret sharing or the Shamir scheme depends on a set of keys/secrets sharing (SS_s) and threshold (TH) to produce a master key/secret (MS). The master secret can be created from some or all of the SS_s (Liu, Yang, Wang, Zhu & Ji 2018). In this scheme, TH specifies the minimum number of keys/secrets that allow reconfiguring MS (Ahmadian & Jamshidpour 2018, Stinson & Wei 2018). This scheme consists of two phases: generation and reconstruction. In the generation phase, the server divides MS into a set of secrets sharing (SS_1, SS_2, ..., SS_n), and each Client (C_i) securely receives one secret sharing (SS) that is part of MS. In the reconstruction phase, C_i needs to achieve any set of secrets (SS_s) required by relying on the value of TH to construct MS (correctness and homomorphism properties). If C_i has TH-1 from SS_s, C_i fails to obtain information from server (secrecy property).

Calculating the MS is an immensely difficult operation for the attacker. In addition, the secrets configured for the MS are anonymous users; the attacker does not know if these secrets belong to any of the users (Huang et al. 2016). The Shamir scheme provides an anonymity solution to generate a MS with several features. It provides full security in hiding C_is' SS_s, a MS size equal to C_is' SS_s sizes, easy creation of a MS from a set of keys/secrets, and creation of a new key/secret for one-time use (Dikshit & Singh 2017). Our authorisation scheme uses the Shamir scheme ($TH = 3$) with ECDSA. This security procedure prevents attackers from accessing or modifying patients' data during the implementation of security objectives (authorisation, authentication, authenticity, and integrity).

3.6 Development of EMR/EHR System

3.6.1 Patient's Confidence in HC Services

HC services include not only patients' health records, but also their personal information such as age, chronic disease history, and even sexual orientation. Therefore, security and privacy issues should be addressed carefully by developers of the health applications. Information privacy in HC applications is of interest to HC providers on the one hand, because privacy issues affect the legal and operational environments, and on the other hand, they affect the patient's confidence in the use of HC applications (Rathert et al. 2017). Therefore, earning the trust of patients in health applications depends on two factors. Those are the understanding of providers for security weaknesses and the development of health applications facing these threats.

3.6.2 HC and EMR/EHR Users

Patients need to trust their HC providers such as doctors, nurses and have confidence in the HC services and HC instruments. Patients in HC institutions need services that are efficient, fast and continuous, and at the same time prevent disclosure of their information (ie. restrict access to only authorised persons) (Ganiga et al. 2018). Access to HC networks has several challenges in security and privacy, such as validation of medical devices to prevent members using unauthorised devices, restrict the access of guests and visitors to the HC applications (such as researchers doctors and specialists) and control access to medical records for patients by identifying the role of each of the medical staff members. For instance, the nurse can access only data of mental health that to apply the doctor's directives, the doctor can access identity data and information of mental health, the pharmacist can access only health data to specify prescription as well as the method of taking medication and so on. In addition, it is important to be compliant with medical standards for HC organisations (such as HIPAA standards to ensure data confidentiality) (Bradford Networks 2012). All of these challenges require a great effort and can be a burden on IT staff.

Figure 3.9 shows the taxonomy of HC users. This Figure is divided into users (such as patient, nurse and doctor) and datasets or medical records (information and data). Also, information includes personal information (such as name, age and address) and network information (such as password, OTP, MAC and signature). Patient data includes specific disease data (such as sexual, mental and dermatological).

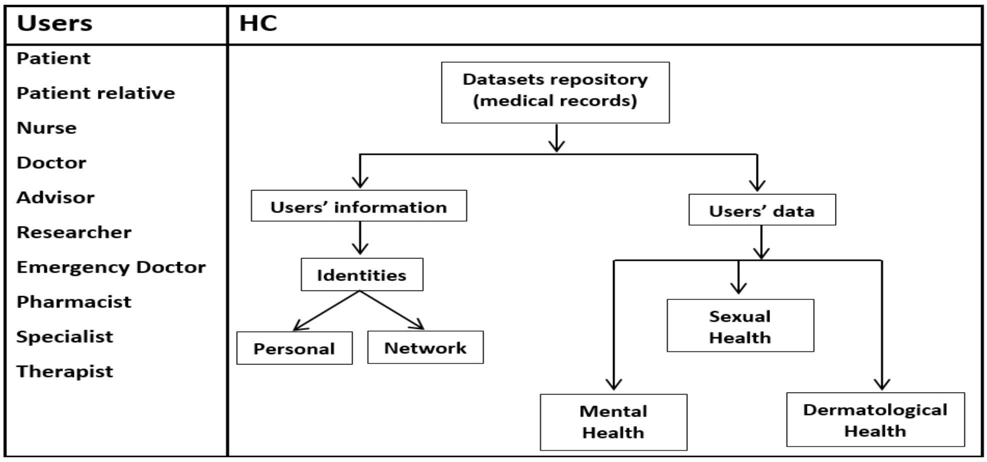


Figure 3.9: Taxonomy of HC users

3.6.3 Administration/Management of Health Organisations

Patients must also be confident that their medical records have been stored where security has been absolutely well addressed. Because of the increasing challenges to health organisations, these organisations require another feature, such as data management in addition to the privacy and security issues (Consultants to Government and Industries 2015, Lokshina & Lanting 2019). The different data contexts in network exchanges cause a heavy burden on authorisation systems. Also, the network security in the HC systems requires the integration of all the different network units in the security system. In addition, the authorisation system should support the connection with the proper adoption of the security context of each device in the network (such as a computer, sensor and phone). Furthermore, providing the vision of information with access control in real-time helps to build an efficient security system for HC (Bradford Networks 2012).

3.7 Integrating WSN with HC Application

In this section, we will focus on how to address the protection issue in our HC application system. This proposal includes medical records protection in three areas: data collection (WSN), transferring authentication and authorisation information, and accessing server datasets (EMR/EHR repository) as shown in Figure 3.10. The components used in the proposed scheme are sensor nodes, a Local Server (LS), communication devices (laptop and desktop) and a remote server (Central Server (CS), Attribute Server (AS) and Data Server (DS)). LS contains data collections by WSN, CS is considered to be as a portal to authenticate and authorise users' requests, AS contains users' real identities (IDs) and DS stores patients' data.

Sensor nodes are used to gather patients' data in an environment (hospital or clinic) and stored it in the LS (EMR repository). Collecting patients' data through sensor nodes requires protection from risks and threats such as internal and external attack. Also, the transfer of patient's data from and to the local or remote server datasets requires the application of critical security policies to prevent the various attacks. Communication devices are used to send and receive medical records, such as medical reports. These devices are used by patients or providers (medical staff) and that are authenticated and authorised access to confidential data for patients. Servers are used to store patients' medical records and control access to datasets. For example, a provider, such as a doctor in a hospital can send authentication and authorisation request by his/her communication device to earn access to a specific patient's data. Also, the patient can obtain his/her medical record history by sending a request directly to the remote server (CS) over the Internet. This situation obtains if the patient is authenticated and authorised to grant access to his/her information and data.

Our framework offers the following capabilities to ensure security and privacy requirements in the HC application:

- WSN data management and use of the proper context (XML) for the exchange of data on different devices, as well as protection of the WSN's data through signature (ECDSA) with the fast hash function and MD anonymity
- Use of lightweight algorithms, such as ECIES and PHOTON (signencryption) to address the performance and security in authentication protocols
- Controlled access for authorised users and datasets management via the Internet by authorisation protocols.

3.8 Protocols to Improve the Security of WSN and EMR

In this section, we will present a set of protocols used in the HC application based on the WSN. This scheme will be detailed further in Chapter 4 to explain our security protocols in the HWSN among sensors, CH and LS.

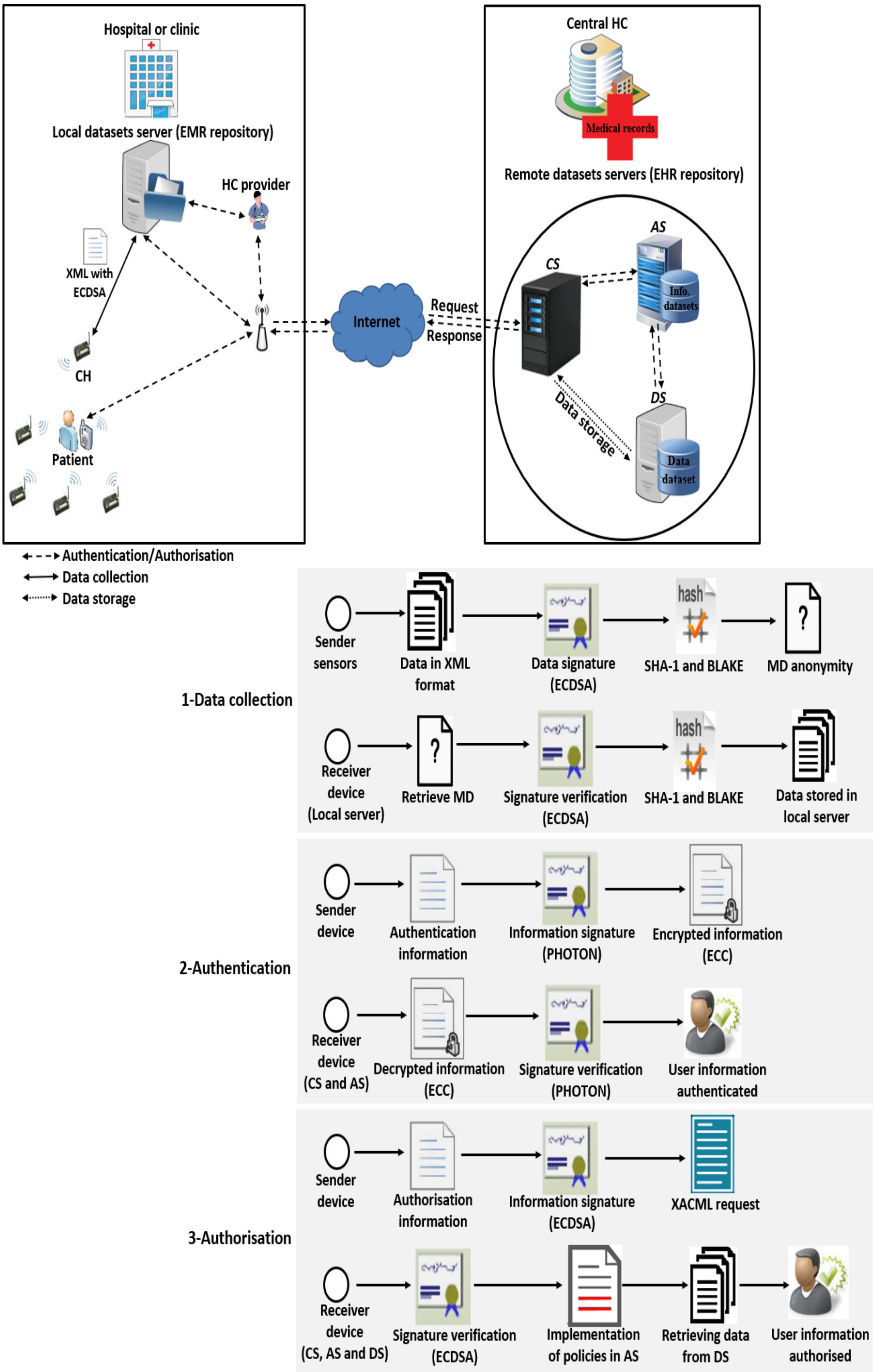


Figure 3.10: The proposed project model in HC environment

3.8.1 HC Data Management and the EMR's Storing/Exchanging

We propose to use of XML to manage the wireless local area network (WLAN) data. This descriptive language enables all devices to read data in a simple and meaningful way. In addition, we propose using the mechanism of signature (ECDSA) with XML file for the management and protection of patient data at the same time.

The process of verifying collected signatures from sensors consumes CH energy and time quickly. The ECDSA algorithm accomplishes the operations the public key, signature generation and signature verification. The operation of signature verification in the ECDSA algorithm consumes more time and energy than a signature generation (Jariwala & Jinwala 2012, Othman et al. 2013). We assume that we have a clinic or a hospital care scheme as shown in Figure 3.11. We propose using homomorphic property with the signature in all sensor nodes. Homomorphic property means the direct computation on signed data. Each sensor node signs the message, and then sends it to the CH. Thereafter, the CH combines all signed messages without verifying the messages. Afterwards, the CH sends signed messages to a LS (unrestricted resources). As the transport operations consume more energy from the computation operations, the collection of signed messages without verify in the CH reduces the transport operations and thus saves time and energy.

3.8.2 Integrity and Authentication of EMR

We propose to use the BLAKE and anonymity with the ECDSA algorithm in the authentication of EMR. For more details about the BLAKE and ECDSA algorithm, refer to Appendix A.3.

We propose adding an anonymity property to ECDSA's signature to hide information of a signature's parameters r and s (each of r and s have the same number of bits) from the attackers. We accomplish this property in the original signature (r, s) depending on the value sent from the server (LS). If the value is odd, we divide the signature into three parts. If the value is even, we divide the signature into four parts and exchange parts. In both odd and even cases, a counterfeit signature and padding are added to the original signature to support signature anonymity. After the completion of this operation, the value is sent with the message to specify the odd or even division in the receiver. Afterwards, the receiver can verify that the message and return the same parts of the signature to

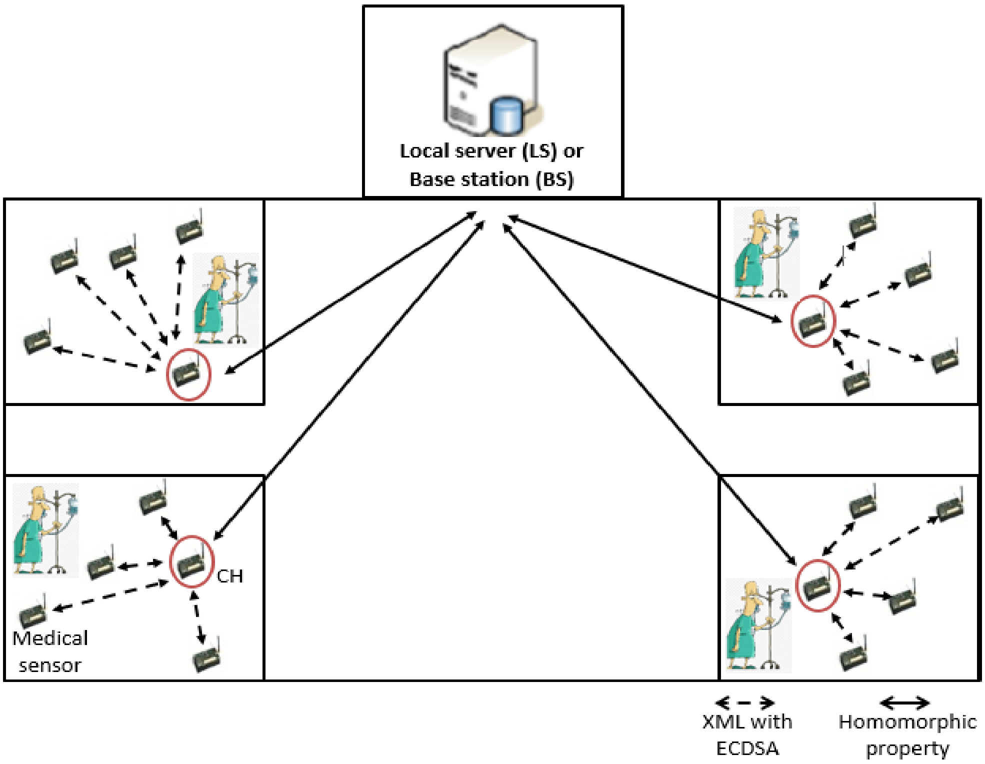


Figure 3.11: Homomorphic and 1 scalar multiplication with signatures

its original place. This mechanism prevents attacks from penetrating the ECDSA signatures because the attacker, when he/she gets on the value of r and s, he/she does not have the original value for r and s. As a result, the attacker cannot derive the private key and the patient's data is protected from the change.

3.9 Users' Authentication in EHR

In this section, we will briefly explain the design of the authentication scheme. This scheme will be discussed in more detail in Chapter 5 to explain authentication protocols among network entities.

3.9.1 Information Confidentiality in EHR

Users' information is a sensitive part of the definition of users in the HC application. This information proves their legitimacy in accessing the network services. The authentication process is the first security procedure when connecting to a HC application network. Therefore, user authentication information needs mechanisms that apply the requirements of confidentiality and integrity to prevent disclosure or changes to information. Signcryption (encryption and signature) for users' small information is critical to repressing attacks, especially, external. However, EHR systems require efficient security protocols because of the large volume of communications exchanged between users and the EHR repository. We propose using ECIES 256-bit for encryption and PHOTON 256-bit for signature. These algorithms perform efficient operations as specified in the Appendix B.

Although, encryption is extremely expensive for server resources when used with large data in HC environments, it is suitable when used with small authentication information. Therefore, we use efficient ECIES to encrypt only users' information in our authentication scheme. In this scheme, we rely on three entities (communication devices, CS and AS) to complete the authentication process. The communication device sends the authentication request after signing and encrypting the user information to the CS and AS. These servers validate the authentication request by decrypting the authentication request, verifying signatures, and checking user security parameters such as pseudonym, password and timestamp. Failure of the authentication request prevents the user from accessing the network services, such as the authorisation process of data access is not allowed when the authentication process fails.

3.9.2 Ensure User and Device Authenticity

The design of authentication protocols in the HC application requires robust mechanisms to demonstrate legitimate users and their devices. There are several reasons for penetration of the authentication process in health applications. For instance, the theft or loss of the legitimate user's device, the analysis of requests for login, registration and authentication transferred among network entities, the use of real information for a legitimate user (internal attack) or disguise as a legitimate user (external attack) are serious risks of penetration of the authentication process.

The process of separating real information from authentication protocols and security parameters (such as password and private key) from user communication devices greatly limits the penetration of the authentication process. Therefore, we propose using techniques, such as OTP, mutual authentication and MAC with encryption and signing to prevent the aforementioned attacks on the authentication process. OTP is used to establish the user's legitimacy in the HC network registration process. Mutual authentication is used to prevent imitation attacks either from the communication device or server. MAC is used to prove the legitimacy of the device. Additionally, we propose hiding the private key of the ECIES algorithm in a computation process that includes the signature and password, while preventing the password from being stored explicitly on the user's device. Consequently, the use of these techniques within authentication protocols provides authenticity to the legitimate user and the device.

3.10 Privacy of Users' Authorisation in EHR

In the previous section, we addressed the authentication issues; while we will briefly address the authorisation issues in this section. This scheme will be detailed further in Chapter 6 to explain authorisation protocols among network entities.

3.10.1 Access Control with EHR Datasets

The other important part of HC applications is the protection of patient's data stored in the server datasets (DS). These data need protection mechanisms from internal and external attackers. Therefore, patients' datasets in the server require defining the access privileges of each user (such as patient, patient relative, nurse, doctor, advisor, researcher, emergency doctor, pharmacist) wants access to this data. The access control (AC) property only allows authorised users access to certain data. AC provides confidentiality in the EHR by restricting access rights for authorised users (Rezaeibagha et al. 2015).

AC models are the cornerstone of robust privacy in authorisation systems. We propose to merge two models' RBAC and ABAC to specify the role of each user with a specific task and attributes that grant more privileges. In addition, we propose the use of features XACML to protect patient's data from illegal and unauthorised access. Initially, users (providers or patients) should be authenticated to the HC network. Our authentication scheme prevents external attackers through the use of signencryption (ECC/PHOTON).

The next stage is the use of XACML to determine the medical records that the user can access. At this stage, our application involves identifying the capabilities of each user to determine the access of each user to certain data and define the tasks of each user. We propose signing of the attributes and information in an XACML request. XACML defines the policies and the semantic to apply those policies. XACML formats are used for request and response between the entities PEP and PDP to determine AC of the source (Sartoli & Namin 2019). Furthermore, XACML is composed of policies and rules. Policies define the applied of request, while the rules govern limitations for the application of XACML. The request consists of attributes associated with the request sender, and the response contains the decision (permit, deny, not applicable and indeterminate).

3.10.2 Using Pseudonym and Anonymity with EHR to Hide the Medical Records

Pseudonym and anonymity are privacy properties in unlinking, hiding and disguising the medical records of patients. The pseudonym property is applied to the patients' data by unlinking real information with data about users who are not required to know the data identity. For example, the nurse is not required to know the information about patient identities (IDs). But it is possible to know some issues, such as the medical name (pseudonym), the range of age and medical reports. The doctor can know some of the personal information of the patient. But it is not important to know the criminal status of the patient information. In addition, if a doctor asked to consult another doctor, the latter is allowed to access only health data and medical reports. Suppose the patient's information and data are stored in the datasets and only the central HC administrator has the authority to access all medical records. We assume the use of four datasets (users' attributes (patients and HC providers), pseudonyms, policies (on AS) and patients' data (on DS)). Each patient has a number and pseudonym in the central database in addition to the use of the concept of multiple pseudonyms between the communication device and servers (CS, AS and DS). For instance, if the patient or provider sends an authorisation request to the network, the user's pseudonym will be different when the request is transferred between the communication device and network entities (CS, AS and DS). The use of a multi pseudonyms mechanism prevents an attacker from tracking requests and responses when they are transferred between network entities.

Users' information requires the anonymity property to disguise and prevent disclosure of their IDs. We propose the use of information anonymity with XACML to protect patients' data from illegitimate and unauthorised access. The anonymity property is applied to the users' information by concealing signatures (ECDSA) and using the Shamir scheme. The addition of the anonymity property with XACML leads the increased privacy of data by hiding data and limiting access for authorised users.

Authorisation protocols become more robust when implementing a Shamir scheme with signatures because a set of secrets is created to produce the master signature. These secrets provide anonymity (depending on the number of secrets or threshold) in each authorisation request. We use the anonymity property with information rather than data for two reasons. First, to disguise and protect the users' information, and second, to maintain network efficiency. Because the use of anonymity with large patient data is extremely expensive for the network's resources compared to the small size of user information being interrogated during the user authorisation process. In addition, we use multiple pseudonyms with both information and data to prevent the association of real information with data.

3.11 Summary of the Chapter

In this chapter, we have described the general architecture, EMR/EHR repository and methodology of our proposed project. In addition, a detailed explanation of cryptography techniques used in the authentication, authorisation and data storage schemes is provided. Also, we have described the general framework of the proposed project that includes our general network model that connects/links three security schemes in one system.. In Chapters 4, 5 and 6, we will provide a detailed explanation of our authentication, authorisation, and data collection schemes.

Chapter 4: A More Efficient and Secure EMR Storage and Repository

In traditional HC, the collecting and storing of patients' health and medical records has been slow and insecure; a costly and 'privacy-intrusive' practice. ICT advances have made it possible to use electronic health and medical records which can be collected by sensor networks and securely stored in a data server. In this chapter, we will put forward to taking advantage of the wireless sensor network for collecting EMR records in a local data server or a data repository.

4.1 Data Collection by HWSNs

EMRs are widely applied in the health sector (Sarkar 2017). Moreover, EMR needs patients' data collection technology such as HWSN. In terms of sensor devices, HWSN consumes its resources through computations, communications and transportation. Because these networks have limited memory, energy and bandwidth, sensors suffer from many issues that limit their ability to accomplish necessary data security. For instance, routing protocols are important to reduce energy consumption when sending authentication requests. Also, the heavy computations of complex signatures are a serious problem for the energy consumption of sensors. In addition, remote sensors consume considerable energy when connected to a network cluster. Furthermore, signatures verification in each CH_i (per round) is a huge burden on the energy consumption and time of CHs. In addition, joining malicious sensors to wireless clusters quickly destroys CHs. Therefore, in addition to location security, protecting information sensors is essential for data security.

Unauthenticated communications and transfers cause complex computations that waste the time of the EMR server. In each round, the EMR server requires verification of signatures for both CHs and SNs, in addition to the complex operations of storing the collected data. Furthermore, privacy breach issues in access to the repository are a very serious problem in EMR systems. The loss of data/information due to repository penetrations takes a long time to retrieve accurately. Additionally, continual and systematic updating of medical records and the difficulty of classifying data, provide opportunities to hack an EMR server. Therefore, preventing attackers from breaking into the EMR server is important to the provision of data security. As a result, data security issues in both sensors and the EMR server should be addressed to build a reliable data collection scheme.

4.1.1 A Reliable and Efficient Scheme for Data Collection

We propose a **R**eliable and **E**fficient **I**ntegrity **S**cheme for Data **C**ollection in **H**WSN (REISCH) to ensure that patient data is transferred/stored to the LS/BS securely and efficiently. The REISCH has been characterized as follows:

- Applying the Elliptic Curve Digital Signature Algorithm (ECDSA) with BLAK2bp instead of ECDSA with a Secure Hash Algorithm 1 (SHA1) to improve HWSN lifetime and prevent intruders from altering/changing patients' data
- Using the homomorphic mechanism with CHs to reduce energy consumption when aggregating patient data from sensors
- Hiding the sensor's identification (SID) and location (SL) by using random pseudonyms. This mechanism prevents intruders from detecting sensor information transmitted between network terminals.

4.2 Our Proposed Data Storage Model

Within any HC application system, the security of the EMR/EHR repository is important but, privacy is more important because if a hacker penetrates the repository and detects user identities, then his/her harm will be greater. In this section, we introduce the model and develop protocols for data collection and storage.

4.2.1 Network Model

The REISCH scheme includes a set of entities, as shown in Figure 4.1:

1. Sensor (SN): This entity collects raw data related to a specific patient. It sends this data to the CH

2. Cluster head (CH): This entity aggregates data from the sensors that followed it. Then, it sends this data to the LS

3. Local server (LS): This entity receives data from all CHs in each round and stored it in the EMR's repository. This data is subsequently sent periodically to the central server (CS)

4. Central server (CS): This entity is gateway access remote servers such as the attributes server (AS) and the data server (DS). It receives data from the LS and sends data to the DS after being authenticated by the AS. Security procedures in AS and DS are left for the Future Directions chapter.

Our network model works with a low-energy adaptive clustering hierarchy (LEACH) protocol for WSN. LEACH uses clustering architecture to improve WSN lifetime. More details about this protocol are available in Awaad & Jebbar (2015). Each group of SNs collects raw data for a specific patient. These SNs sign data before sending them to CHs. Each CH aggregates data and signatures from his followers. Then, each CH uses the homomorphic property with all data and signatures without verifying the signatures to reduce energy consumption on the CH and send them to the LS. As the LS has unlimited resources, it verifies and validates collected data from SNs. The LS sends data stored on the EMR's repository to the central repository to allow HC users (patients and providers) to access them by sending authentication/authorisation requests to the central server (CS), attributes server (AS) and data server (DS). This chapter focuses on performance and security issues in SNs, CHs, LS and CS. Security issues for datasets and transferred data in CS, AS and DS are left for future works.

4.2.2 Design Goals of REISCH

To develop a reliable data collection scheme, REISCH adopts the following security requirements:

- **Information confidentiality**: This requirement is achieved to hide SNs/patients' identities and to protect patients' secrecy from disclosure by intruders.

- **Data integrity**: This is to protect the patient data from intruders' tampering. The collected data should arrive at the intended target without alteration to provide a reliable communication channel between the SNs, CHs, LS and CS Al-Zubaidie et al. (2019*b*)

- **Non-repudiation**: This is to prove that the message is sent by a particular SN in the HWSN. If a legitimate entity in HWSN performs internal attacks,

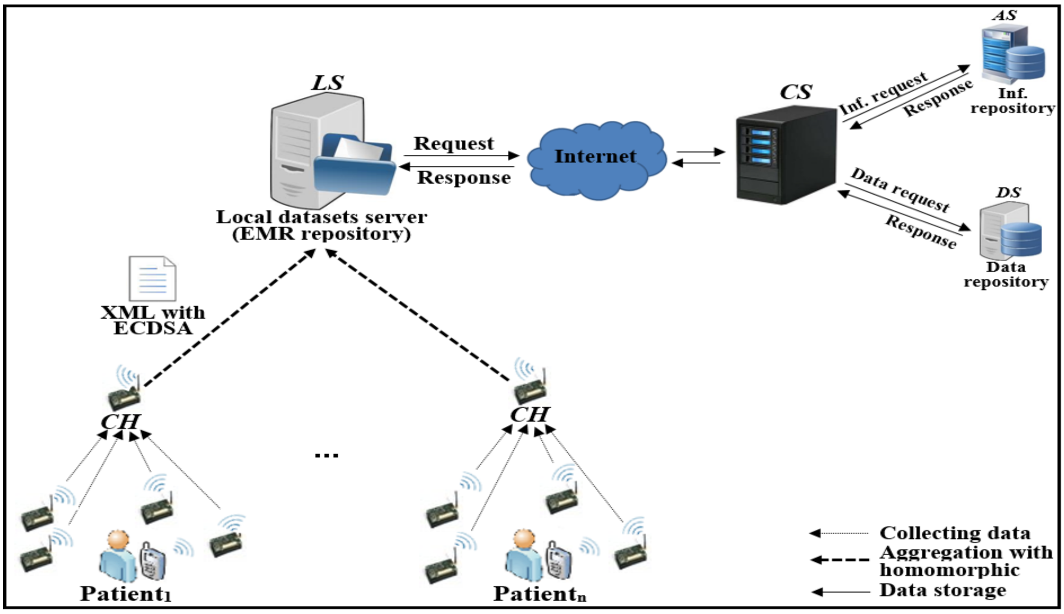


Figure 4.1: General REISCH model

he/she cannot deny his/her messages while using the privileges granted to him/her

- **Freshness**: It indicates that the data collection message is new and updated to guarantee that the intruder cannot replay the previous message at a later time. This goal is accomplished by a checking of time, a random passcode, and random signatures within each data collection round to counteract spoofing risks such as replay, MITM, and impersonation

- **Security of Localisation**: This feature ensures that the patient/sensor's real location is protected from detection, or sends error messages to the LS by an intruder

- **Scalability**: HWSN applications elaborate in a scalable environment in both data and devices. Thus, these applications need data collection schemes capable of processing and adapting to the ever-increasing number of devices of the HWSN. This feature indicates the ability of the data collection scheme to properly handle huge HWSN devices. Public key signature schemes are ideal for providing this requirement Kumar et al. (2016)

- **Survivability**: It provides a certain level of services in patient data collection or network capability to withstand failure/threats in an appropriate manner and continue to provide services between the SNs and LS for as long as possible

- **Accountability**: This property means tracking the behaviour of malicious threats/suspicious activities by legitimate users/counterfeiting attacks in accessing the EMR repository

- **Efficiency**: HWSN sources such as energy, storage, and processor should be within the design objectives of security protocols in HWSN.

4.3 REISCH's Scheme

In this subsection, we will explain REISCH details in terms of entities preparation, using ECDSA-BLAKE2bp, applying a camouflage signature, implementing homomorphic and REISCH protocols.

4.3.1 Entities Preparation

To start collection and storage processes, it should prepare the HWSN network with the following points:

- Each sensor (SN_i) and LS server is provided SN_i pseudonym (SN_{Pseud}), SN_i pseudonym signature ($SigLS_i(SN_{Pseud})$) and SN_i location (SN_{SL})
- All entities (SN_i, CH_i, LS and CS) generate K_{pu_i} and K_{pr_i} to apply asymmetric cryptographic
- Each entity broadcasts K_{pu_i} to network members
- Each SN_i uses ECDSA signatures ($SigSN$ and $SigCH$) to achieve collected data integrity
- Each server (LS and CS) uses ECDSA signatures ($SigLS$ and $SigCS$) to achieve storage data integrity.

4.3.2 Integrity of EMR being Transmitted

For the integrity of patient records being transmitted between the HWSNs and the LS, the REISCH signs all EMR/EHR records using the ECDSA algorithm, in which its hash function will be BLAKE2bp, rather than the traditional SHA1. In REISCH, we used ECDSA-BLAKE2bp to ensure data integrity as well as adding SN_{Pseud} within $SigSN$ to prevent changing data. The LS and CS accept only a valid signature after verification. The high performance and security of the ECDSA-BLAKE2bp algorithm makes it an appropriate choice for protecting EMR health records. Also, using ECDSA-BLAKE2bp with XML adds the feature of medical records management in HWSN.

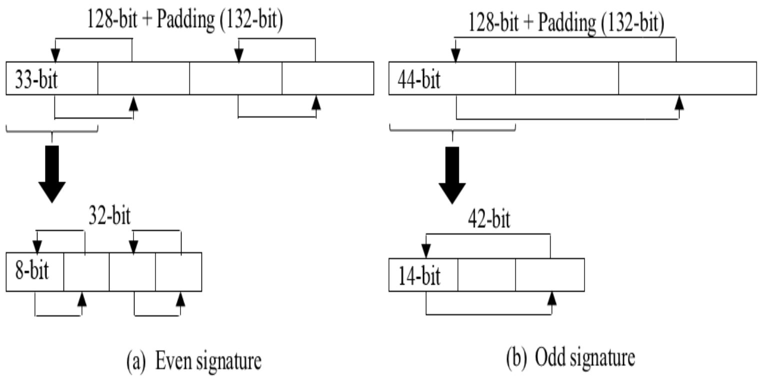


Figure 4.2: Camouflage signature

4.3.3 Applying Camouflage Signature

REISCH uses the camouflage process to hide the data signature completely and prevent traceability, analysis or alteration of data. The camouflage process starts by signing the data to obtain a 64-bit MD and then adding a 64-bit counterfeit signature to a total length of 64-bit + 64-bit = 128-bit. In addition, each SN_i adds padding (0000) to become the total length of the 132-bit signature as shown in Figure 4.2. SN_i performs the process of exchanging data signature segments based on Parity (even / odd) value. It receives this value invisibly from the LS because this value is included in the ephemeral random value ($SigLsE_i$). SN_i tests $SigLsE_i$, if "even" it divides 132-bit into four segments (each segment to 33-bit) and exchanges the segments. Then, SN_i truncates 32-bit from the first segment and divides it into four segments (each segment to 8-bit). If $SigLsEi$ is "odd", it divides 132-bit into three segments (each segment to 44-bit) and then exchanges the segments. It then truncates 42-bit and divides the first segment into three segments (each segment to 14-bit). Because the exchanging operation is based on Parity sent from the LS, this process prevents the detection of the original signature of the data and prevents the data from being changed. Thus, this process protects patient data from tampering or alteration.

4.3.4 Implementing Homomorphic

REISCH uses the homomorphic property with the ECDSA-BLAKE2bp algorithm to increase network performance. Because the verification process in ECDSA consumes more time and processing than the signature process, it is very convenient to use the homomorphic property in HWSN to support both performance and security. The LEACH protocol is based on the principle of clustering to reduce energy consumption, thus REISCH uses the aggregate signature to allow CH_i to aggregate signatures and data without using verification. To double security in REISCH, CH_i

performs the process of aggregating temporary signatures such as $SigSnT_{3s}$ and $SigSnT_{4s}$ in addition to random numbers (SN_{RNs}) and data. Temporary signatures contain unclear original signatures that prevent an intruder from penetrating patient data. The homomorphic procedure reduces energy consumption and thus increases the possibility of using the ECDSA algorithm with the HWSN for as long as possible.

4.3.5 REISCH's Protocols

REISCH scheme consists of three protocols. During these protocols, REISCH provides reliable data collection processes to protect collected patients' data.

- **Protocol 1 between SNs and CHs**
  This protocol performs the data collection process (Figure 4.3 shows the first protocol processes between SN_i and CH_i in the data collection). At the beginning of each round, each SN_i receives a one-time passcode (LS_{OTP_i}) and a random number (SN_{RN_i}). This LS_{OTP_i} contains an ephemeral random value ($SigLsE_i$) of the same length as the signature. SN_i extracts $SigLS_i(SN_{Pseud})$ from the dataset and executes $\oplus$ to extract the secret value $SigLsEi$. Then, SN_i executes the $Parity$ (as shown in Section 4.3.3) process based on $SigLsEi$ to get the temporary signature ($SigSnT_1$). After that, SN_i generates an ephemeral random value ($SigSnE_i$) with the same signature length and uses it with $SigSnT_1$ to compute the $SigSnT_2$ value. Next, SN_i computes the Dif value that represents the subtraction value of the distance between CH_i and SN_i (S_NC_HD) and the distance between LS and SN_i (S_NL_SD). Dif specifies that SN_i is within the HWSN framework (1000m * 1000m). Additionally, SN_i computes a new timestamp (SN_{TS}) and one time passcode (SN_{OTP}). Thereafter, SN_i performs a hidden process for SN_{TS} and SN_{OTP} at a temporary value (SN_{TS_t}) with the addition of a value of only seconds (SS) at the end of the SN_{TS_t}. Furthermore, SN_i uses SN_P to concatenate secret parameters such as SN_{TS_t}, SN_{OTP}, SN_{RN_i}, SN_{Pseud} and SN_{SL} to match them at the LS. To protect both SN_P and $SigSnE_i$, SN_i uses the $\oplus$ operation to hide them by calculating the temporary values of $SigSnT_3$ and $SigSnT_4$. At this point, SN_i computes the message (SN_m) and sends it to CH_i which is a sequence of $SigSnT_3$, $SigSnT_4$, SN_{RN_i}, Dif, SN_{TS_t} and data collection.

  In the CH_i side, it also receives LS_{OTP_i} of the LS and SN_m of SN_i. Afterwards, CH_i truncates Dif_i and tests its value within the HWSN framework by computation $Dif_i \leq$ Maximum value, where the Maximum value should be less than or equal to 707.1068. Then, CH_i computes the timestamp (CH_{TS_1})

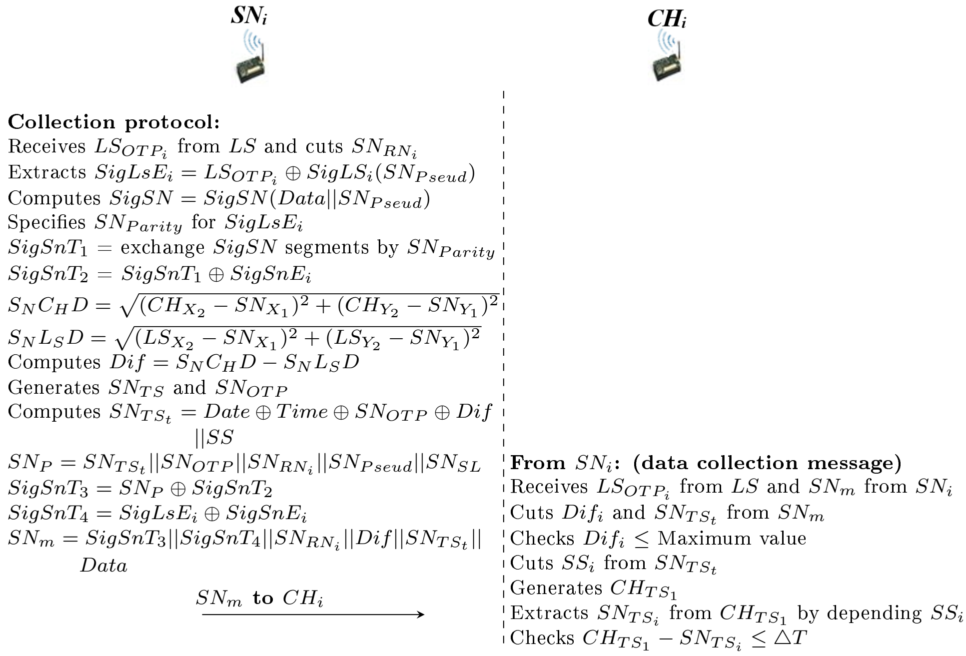


Figure 4.3: Data collection protocol

to prevent late messages. CH_i truncates SS from SN_{TS_t} to obtain SN_{TS_i}. If the difference between CH_{TS_1} and SN_{TS_i} is less than the $\triangle T$ delay rate (we assumed that $\triangle T = 3$), namely, that the message is fresh.

- **Protocol 2 between** CH**s and** LS

  This protocol performs the data aggregation process (Figure 4.4 shows the second protocol processes between the CH_i and LS in the data aggregation). Each CH_i receives temporary signatures, random numbers and collected data from its SN_i followers. Then, CH_i executes the signature process $SigCH$ for the temporary signatures received ($SigSnT_{3_s}$) from its SN_i followers. Thereafter, CH_i extracts the $SigLsE_i$ unique value from LS_{OTP_i} similar to the first protocol based on $SigLS_i(CH_{Pseud})$ stored. Next, CH_i performs the CH_{Parity} process based on $SigLsE_i$ extracted (as described in Section 4.3.3) to compute $SigChT_1$. Moreover, CH_i computes $SigChT_2$ depending on the $SigChT_1 \oplus SigLsE_i$ operation. After that, CH_i generates CH_{TS_2} and CH_{OTP} to prevent the problem of replaying messages later. CH_i calculates CH_P which represents the sequence of secret parameters. Also, CH_i computes CH_A to complete the process of aggregating temporary signatures ($SigSnT_{3s}$ and $SigSnT_{4s}$), random numbers (SN_{RNs}) and collected data ($Data_s$). Finally, CH_i computes CH_m and sends it to the LS.

In the LS side, after the LS sends LS_{OTP_i} for all SN_i, it waits to receive CH_m of all CH_i per round. The LS truncates CH_{RN_i}, SS_i and CH_A from each CH_m received. It uses SS_i to reconfigure CH_{TS_2}, the LS also generates a timestamp (LS_{TS_1}) and tests $\triangle T$ between LS_{TS_1} and CH_{TS_2} to confirm the freshness of the message. Then, it tests whether CH_{RN_i} matches the value previously sent. If CH_{RN_i} is correct, it is used to determine CH_{Pseudi} and the latter is used to determine CH_i location (CH_{SL_i}). Then, the LS retrieves the temporary signatures and random numbers ($SigSnT_{3_s}$, $SigSnT_{4_s}$ and SN_{RN_s}) from CH_A. The LS uses the $SigLsE_i$ value to specify a $Parity$ (even/odd) value for all SN_i and CH_i. It computes a signature ($SigLS_{1_i}$) for all SN_i signatures that followed a specific CH_i ($SigSnT_{3_s}$) and exchanges the $SigLS_{1_i}$ segments based on CH_{Parity}. After that, the LS calculates $SigLsT_{1_i}$ which equals $SigChT_2$ in CH_i based on $SigLS_{1_i} \oplus SigLsE_i$. To ensure the legitimacy of CH_i, the LS extracts the secret parameters at CH_{P_i} and tests the match CH_{Pseudi} and CH_{SL_i} in the datasets. At this point, the LS checks for data integrity collected by SN_i. Similarly, the LS uses SN_{RN_i} to determine SN_{Pseudi}, and performs data signature ($SigLS_{2_i}$) that equals the $SigSN$ in SN_i and exchanges the $SigLS_{2_i}$ segments based on SN_{Parity_i}. Next, the LS uses $SigSnT_{4_i}$ and $SigLsE_i$ to extract $SigSnE_i$. Thereafter, the LS uses $SigSnT_{3_i}$ and $SigSnE_i$ to compute $SigLsT_{2_i}$. Finally, the LS extracts the secret parameters for SN_i from SN_{P_i} and tests matching SN_{Pseudi} and SN_{SL_i} in datasets. If all signatures and parameters are validated correctly, then the data collected by SN_i is legitimate and correct, and has not been tampered with by the intruder.

- **Protocol 3 between LS and CS**

  This protocol performs the data storage process (Figure 4.5 shows the third protocol processes between the LS and CS in the data storage). Initially, the LS generates a new pseudonym (LS_{Pseud_n}) and timestamp (LS_{TS_2}) to prepare for the process of sending data to the CS. Then, the LS computes the $SigLS$ signature based on the CS 's old pseudonym (CS_{Pseud_o}). After that, the LS generates and sends LS_{OTP} to the CS, which is based on the $SigLS$, LS_{Pseud_n}, LS_{TS_2} as well as append SS at the end of LS_{OTP}.

  On the CS side, it generates CS_{TS_1}, CS_{Pseud_n}, CS_{OTP} and CS_{RN}. CS uses CS_{TS_1} to test the message arrival time of the LS. Depending on generated secret parameters, such as CS_{OTP}, the CS computes CsT_1 and CsT_2 temporarily. In addition, the CS generates a $SigCS$ signature that includes the temporary value (CsT_2). At this point, the CS computes and sends CS_m to LS containing the sequence of parameters such as $SigCS$, CsT_1, CsT_2, SS and CS_{RN}.

  On the LS side, it truncates parameters embedded within CS_m. Thereafter,

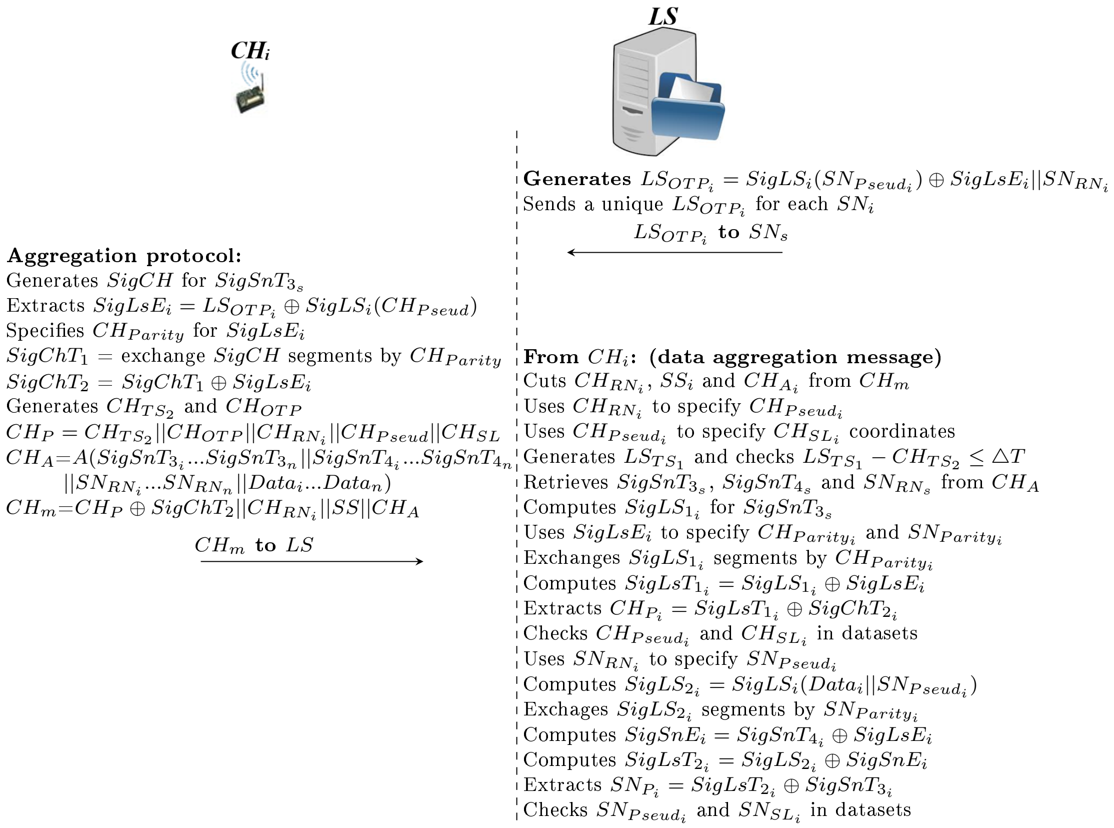


Figure 4.4: Data aggregation protocol

the LS generates LS_{TS_3} to check the arrival time of CS_m. Furthermore, the LS computes CS_{OTP} that relying mainly on LS_{Pseud_n}. Afterwards, the LS extracts CS_{Pseud_n} to calculate $SigLS_3$. The LS tests matching $SigLS_3$ and $SigCS$, and if the result is identical, this means that the mutual authentication process between the LS and CS is performed correctly and legitimately. After this stage, the LS prepares the data storage request to CS. First, the LS generates LS_{TS_4} and LS_{RN} to ensure randomness and freshness. After that, the LS computes the $SigLS_4$ signature that depends on the LsT_1 temporary parameters. Then, the LS computes the $SigLS_5$ data signature that depends on temporary parameters such as LsT_2, LsT_3, and $SigLS_4$ as well as the $Data$. Finally, the LS sends LS_m which includes $SigLS_5$, SS, LS_{RN} and $Data$ to CS.

On the CS side, it receives LS_m of LS. The CS generates CS_{TS_2} new to test access time LS_m. The CS calculates $SigCS_1$ and $SigCS_2$ similarly to $SigLS_4$ and $SigLS_5$ respectively. At this point, CS checks matching $SigCS_2$ and $SigLS_5$, and if the result is identical, it means that the CS received patients' data from the LS correctly and integrated it without any changes by malicious attacks.

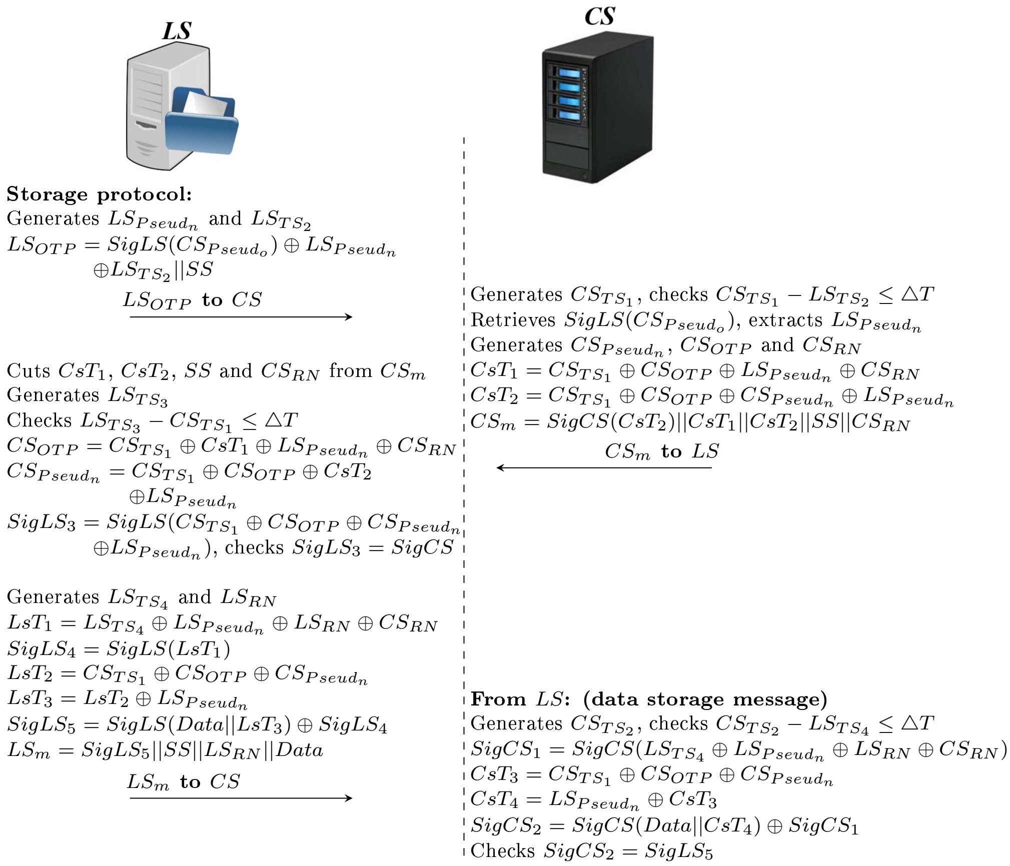


Figure 4.5: Data storage protocol

4.4 Summary of the Chapter

In this chapter, we have provided details of the security of data collection and our contributions to build a new data collection scheme. After that, a network model for the data collection scheme explains the general structure for securely storing patient data in the EMR repository. Finally, the data collection scheme methodology has been explained in detail.

Chapter 5: Robust Security to Authenticate Users Identity to the EHR Repository

In the preceding chapters, we identified the limitations and/necessary improvements to existing HC applications. Chapter 4 proposed a solution integrating WSN technology with EMR repository in a HC application. From this chapter on, we will explore the security strategy developed for the proposed HC application. This chapter describes an authentication scheme developed to protect user/network information.

5.1 Information Security in EHR Systems

As we know, user information (personal such as name and network such as password) defines and proves user legitimacy in EHR systems. This information requires a procedure to prevent disclosure of users' secrets while at the same time proving their legitimacy in the network. Authentication is the most critical security requirement that plays a key role in building correct security before the exchange of patient data in EHR (Arshad et al. 2015, Das et al. 2017, Li et al. 2016, Rajput, Abbas, Wang, Eun & Oh 2016). First, it reduces malicious or fatal errors caused by penetration attacks on the authentication information. Second, it alleviates errors in specifying drug, dose, timing, or procedure (Yuehong et al. 2016). Therefore, access to these data should be controlled to prevent unauthenticated access. As a result, authentication protocols are a critical requirement to repel various attacks.

Typically, the server application should prevent all fake and illegal authentication requests. It should protect personal information, health records, and physiological parameters (such as sugar, and heart rate) (Arshad et al. 2015, Wazid et al.

2016). However, authentication information may be easier to compromise if EHR data and information are stored on a single server. Furthermore, the transfer of authentication information in an insecure environment (WLAN or Internet), may expose patients' data for detection or modification (Li et al. 2016, Shrestha et al. 2016).

Unfortunately, traditional cryptography (such as RSA) and signature (such as SHA1) schemes require complex computations that consume server resources, such as processing power and memory, when used with EHR systems that may deal with large amounts of health data and thus, render them unusable. The electronic signature is an important way to check the integrity of user information in the authentication request (Giri et al. 2015). Many algorithms, especially lightweight algorithms, such as PHOTON, QUARK, and SPONGENT, are used to implement signature algorithms that perform lightweight operations to reduce high overheads on servers. HC applications require high-speed cryptography and signature, and secure algorithms (Liu & Chung 2017). There is a misconception about using a symmetric encryption algorithm instead of public encryption algorithms. Using symmetric encryption instead of asymmetric encryption, where symmetric is faster but sacrifices security. Recent research has proved that using asymmetric encryption is better because it provides strong security and supports applicability in large systems (Farash et al. 2016). In addition, symmetric encryption algorithms have problems with scalability and single secret key detection. To implement an authentication scheme, many algorithms, such as ECC, RSA, hash function, bilinear pairing, fuzzy extractor, and XOR operation (Chandrakar & Om 2017), are used to design HC projects. Many recent HC applications are based on ECC and RSA; both of which provide the same security level, although ECC is more efficient than RSA. The design of an authentication scheme in HC applications should provide mutual authentication, resistance to known attacks such as MITM, eavesdropping, tracing, replay, impersonation, guessing, DoS, protection of information and reduced cost and high-efficiency (Yeh 2016, Shankar et al. 2015).

5.1.1 A Robust Model of Authentication for the Proposed HC Application

We propose a **R**obust **A**uthentication **M**odel for **H**ealthcare **U**sers (RAMHU) that uses lightweight algorithms and security techniques for HC applications that perform massive and continuous authentication processes while simultaneously protecting against various attacks. Our contributions to this scheme are summarised as follows:

- RAMHU uses lightweight algorithms for encryption (ECIES) and signature

(PHOTON). These algorithms provide efficient and secure authentication for users in HC applications compared to other algorithms

- RAMHU applies an OTP mechanism to authenticate users in their first registration in the HC network with timestamp verification and random nonce generation to repel different types of external attacks
- RAMHU uses a multi pseudonyms mechanism to prevent any association between the real information, pseudonyms, and user data. This mechanism prevents attackers from identifying HC users (providers and patients)
- RAMHU integrates the login request with the MAC address in addition to verifying that this address is original and not fake for authentication of legitimate devices. This prevents attackers from using different devices to compromise the network information
- RAMHU improves the mutual authentication between the server and clients to prevent spoofing and impersonation attacks by either fake server or client. This prevents external attacks intended to deceive trusted parties.

5.2 The Proposed Authentication Scheme

In this section, we will detail our authentication scheme that provides security and efficiency features in HC applications. This section is divided into the network model, model goals and proposed authentication scheme protocols.

5.2.1 Network Model

The RAMHU model consists of four entities as shown in Figure 5.1:

1. Client (C_i) or user: This entity includes patients, relatives of patients, and HC providers such as doctors, researchers, emergency practitioners, advisors, and nurses
2. Central server (CS): This entity is a gateway to authenticate users with the attributes server and to authorise the data server
3. Attributes server (AS): This entity contains real user information as well as multi pseudonyms. The authentication process requires verifying the association of the actual information with the multi pseudonyms in this entity
4. Data server (DS): This entity contains user data as well as multi pseudonyms. This entity is not implemented in our authentication scheme. Our scheme focuses only on the process of user authentication in the HC network.

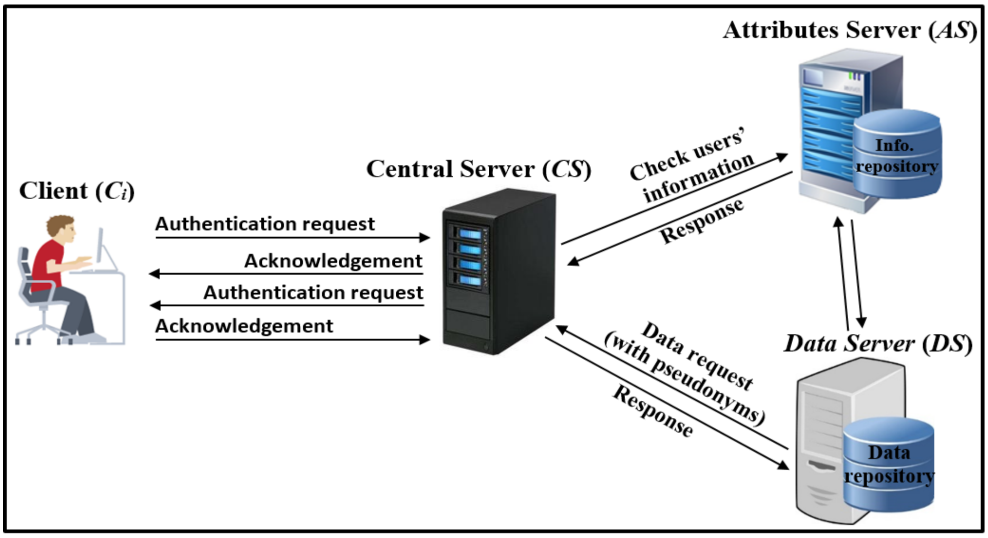


Figure 5.1: General network model

Generally, C_i creates an authentication request mainly based on ECIES and PHOTON. C_i sends a request to the CS to verify encryption, signature, and security parameters (such as MAC address, pseudonyms, and OTP_i). Then, the CS sends a request to the AS to verify the link between the pseudonyms, the real information, the signatures, and PW_i. After that, the AS sends the response to the CS that verifies the signature and the parameters and then the CS sends the response (authenticated or not) to the C_i. If the user is authenticated, the user can then send an authorisation request to access the data in the repository (DS) and obtain the authorisation response from the CS, AS, and DS. Our authorisation protocols will be detailed in Chapter 6.

5.2.2 Design Goals of RAMHU

To build a robust authentication scheme, RAMHU must meet the following security requirements:

- **Confidentiality**: This requirement is performed to hide authentication information and to preserve user secrecy from detection by intruders. To fulfil this requirement, a high-security cryptographic algorithm (Al-Janabi et al. 2017) should be used. RAMHU executes ECIES to hide authentication information from intruders

- **Integrity**: This protects the authentication request information from modification by intruders. The authentication request should reach the intended destination without modification to provide a reliable communication channel for legitimate users (Rajput, Abbas & Oh 2016). RAMHU performs a PHOTON to prevent any process of altering or modifying the user authentication information in HC applications

- **Non-repudiation**: This requirement prevents both clients and server from denying their authentication requests. This is a way to prove that the message is sent by a particular sender in the HC applications network. If a legitimate user in the network performs internal attacks, he/she cannot deny his/her activities while exploiting the privileges granted to him/her. Our scheme uses PHOTON signatures and a MAC address to meet this requirement and detect malicious attacks

- **Anonymity**: This requirement is extremely important in supporting the confidentiality of the authentication request. The purpose of this requirement is to disguise the source and destination of the authentication request. If the authentication scheme applies anonymity with encryption, the attacker finds it exceedingly difficult to analyse authentication requests for a particular user at different times because the authentication request is different each time the user is connected to the network (Rajput, Abbas, Wang, Eun & Oh 2016). RAMHU applies this requirement through the use of random nonces among entities

- **Pseudonym**: This requirement denotes the provision of a mechanism to connect non-real attributes (such as terms and symbols) with the real attributes of the user (such as name, address, and phone number). The use of this mechanism in HC applications is an extremely important way of protecting the personal information of users and prevent the detection of their identities. RAMHU uses a multi pseudonyms mechanism to prevent and separate association with real information

- **Forward secrecy**: This requirement is accomplished when network users use new keys and parameters temporarily without relying on old ones. This requirement prevents attackers from exploiting users' keys and passwords in decrypting authentication requests. Using the temporary random password, private key, and MAC, RAMHU prevents users from accessing previous authentication information

- **Mutual authentication**: This requirement is used in healthcare applications to mitigate the risks of external fraud. With this feature, each party ensures that it deals with a legitimate party. The server authenticates the client by checking encryptions and signatures and vice versa to establish a secure communication channel. In RAMHU, CS and C_i authenticate each other to prevent masquerading and impersonating attacks

- **Scalability**: HC applications operate in a scalable environment in terms of data and users. Therefore, these applications require authentication schemes capable of handling and adapting to the ever-increasing number of users of HC applications. This requirement refers to the ability of the authentication scheme to appropriately handle large HC systems. Public key encryption schemes are efficient in supporting this requirement (Kumar et al. 2016)

- **Freshness**: This requirement indicates that the authentication request is new; updated to ensure that the attacker cannot replay the authentication request at a later time. This requirement is achieved through the provision of time checking, a random nonce, and change of signatures in each authentication process to counteract counterfeit attacks, such as MITM, replay, and impersonation (Al-Janabi et al. 2017), which ensures that the authentication request is unaltered or not tampered with.

5.2.3 Proposed Protocols for the Authentication Scheme

The RAMHU scheme consists of five protocols: initial setup, registration/login, authentication, password update, and revocation. During these protocols, RAMHU provides reliable authentication processes to protect users' information.

- **Initial Setup Protocol**
  In this protocol, all entities are ready to start communicating with each other while configuring all security parameters and ECIES's keys with the following steps:

  - Each legitimate user receives a client application from the authorised system provider
  - Each legitimate user receives a password (that can be changed later) and a random OTP_i to be used in the first registration
  - All entities (C_i, CS, and AS) should create public and private keys (C_i (CK_{pu_i}, CK_{pr_i}), CS (CSK_{pu_i}, CSK_{pr_i}), and AS (ASK_{pu_i}, ASK_{pr_i})) to be used to validate the authentication request. All entities choose an elliptic curve $Ep(a, b)$ over a prime field F_P (where, $P = 256$) and base point G on the curve. Each entity selects a private key K_{pr_i} randomly and generates the public key K_{pu_i} during the implementation of scalar multiplication ($K_{pu_i} = K_{pr_i} * G$)
  - All entities broadcast public key (CK_{pu_i}, CSK_{pu_i}, and ASK_{pu_i}) to use in ECIES's encryption operations.

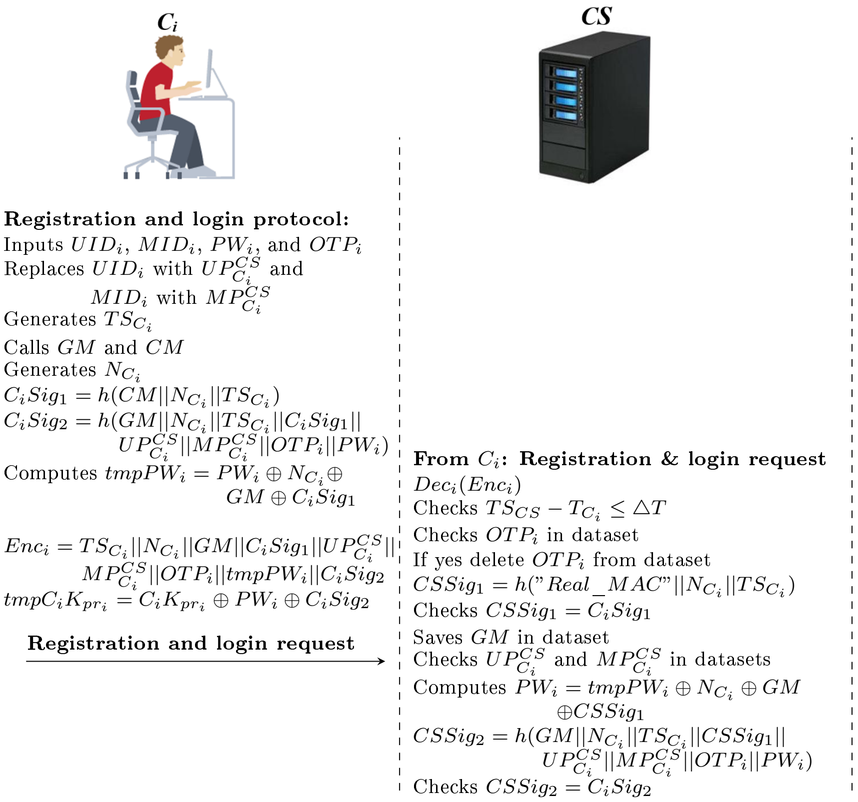


Figure 5.2: Registration and login protocol

- **Registration and Login Protocol**

  Patients and healthcare providers (C_i) should complete the registration and login protocol with CS to become legitimate users of HC applications. Without this protocol, the user cannot complete the authentication process in RAMHU. User registration is performed once, namely, the user does not need to complete the registration protocol subsequently (only the login protocol), the registration information is kept in the servers until the revocation protocol and deletion of user security parameters, such as pseudonyms, MAC address and PW_i. This protocol accomplishes the following steps (Figure 5.2 shows registration and login protocol, and Figure 5.3 shows login protocol):

  C_i side:

  - The user enters UID_i (such as his/her name), MID_i (such as medical centre name), PW_i and OTP_i for registration and login while only entering UID_i, MID_i, and PW_i for the login protocol at subsequent logins to the client (C_i) application. C_i replaces UID_i with user's pseudonym ($UP_{C_i}^{CS}$), and MID_i with medical centre pseudonym ($MP_{C_i}^{CS}$) to protect the authentication information when moving from C_i to CS. C_i generates timestamp (TS_{C_i}) to be used to verify the sending time of the authentication request in the CS

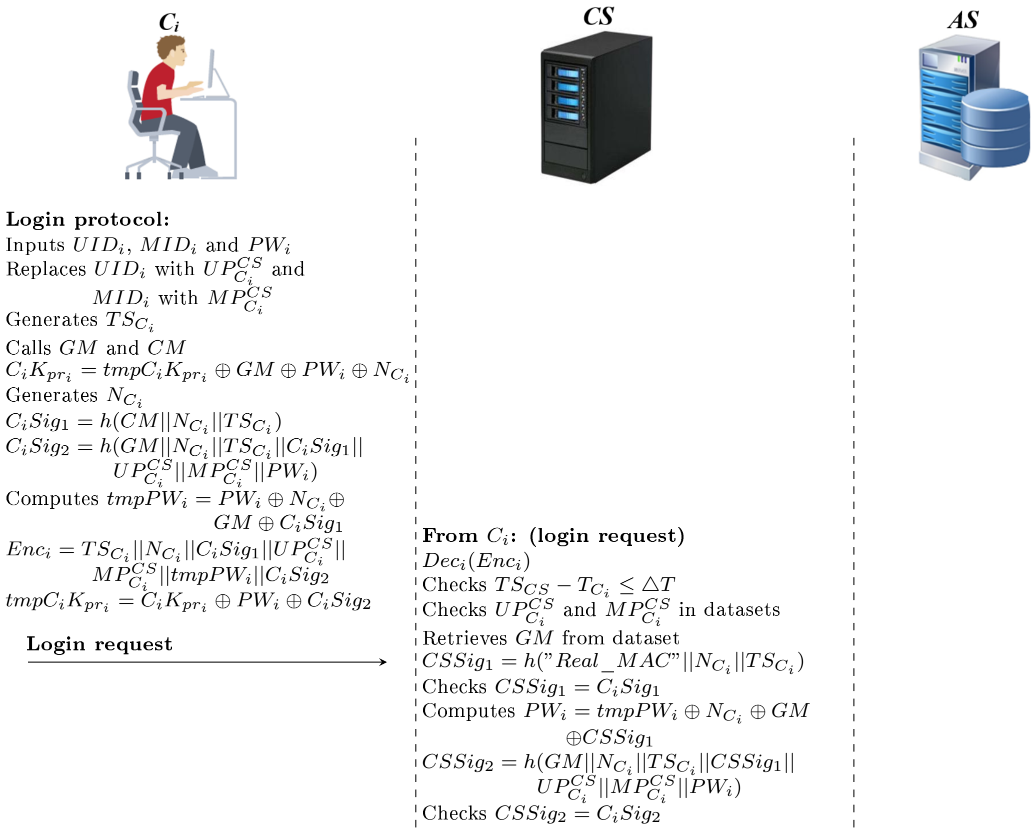


Figure 5.3: Login protocol

– C_i gets a MAC address (GM) by entering its IP (Internet protocol) address. The process of checking MAC (CM) is performed by C_i to test the credibility of the MAC address. In the Linux system, we used the command "ethtool -P interface name" (such as "ethtool -P wlo1") in the C_i application. If the result is identical to GM, it means that the MAC address is native (CM = "$Real\_MAC$"). In the Windows system, C_i searches for string value "NetworkAddress" in the path of the system registry "$HKEY\_LOCAL\_MACHINE \setminus SYSTEM \setminus CurrentControlSet \setminus Control \setminus Class \setminus \{4D36E972 - E325 - 11CE - BFC1 - 08002BE10318\} \setminus$" (as shown in Figure 5.4). If NetworkAddress = null, CM = "$Real\_MAC$"; otherwise CM = "$Fake\_MAC$". The GM sent is encrypted with an authentication request while CM is implicitly sent with the signature value (C_iSig_1). If only login protocol, C_i needs to extract the private key from the temporary key, MAC address, password, and random nonce through $tmpC_iK_{pr_i} \oplus GM \oplus PW_i \oplus N_{C_i}$. C_i generates a random nonce (N_{C_i}) to change signature and encryption data and add anonymity to the authentication request

– C_i performs two signatures using the PHOTON-256 algorithm to protect information from modification. The first signature

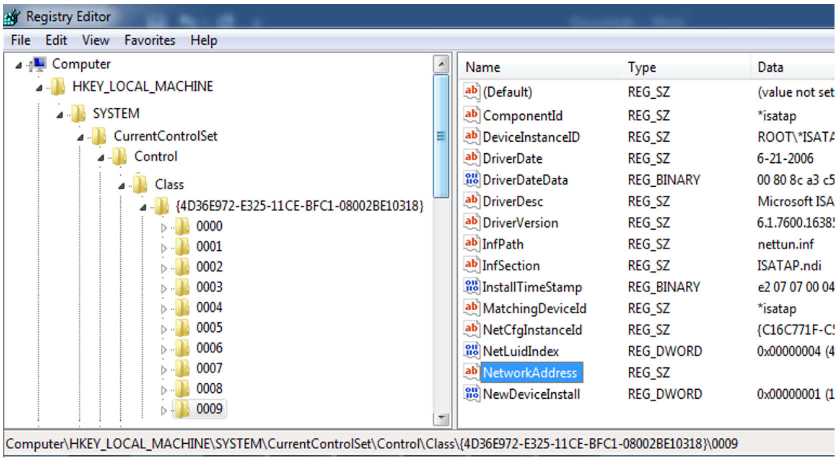


Figure 5.4: NetworkAddress path in system registry

includes the parameters check MAC, nonce, and timestamp ($CiSig_1 = h(CM \| N_{C_i} \| TS_{C_i})$. The second signature includes all the authentication parameters of the get MAC, nonce, timestamp, first signature, pseudonyms, one time password, and password ($CiSig_2 = h(GM \| N_{C_i} \| TS_{C_i} \| \; CiSig_1 \| UP_{C_i}^{CS} \| MP_{C_i}^{CS} \; \| OTP \| PW_i)$. In the login protocol, OTP is not added to the signature

- C_i computes a temporary value ($tmpPW_i = PW_i \oplus N_{C_i} \oplus GM \oplus C_iSig_1$) of PW_i when moving from C_i to CS

- C_i uses ECIES to encrypt and hide all the data of this request ($Enc_i = TS_{C_i} \| N_{C_i} \| GM \| \; CiSig_1 \| UP_{C_i}^{CS} \; \| MP_{C_i}^{CS} \| \; OTP_i \| tmpPW_i \| CiSig_2$). In the login protocol, OTP_i and GM are not added to the encryption as in Figure 5.3. After that, C_i sends the registration and login request or login to CS to complete the authentication protocol. Then C_i hides the private key by $tmpC_iK_{pr_i} = C_iK_{pr_i} \oplus PW_i \oplus C_iSig_2$.

CS side:

- Upon receiving a registration and login request or login request, CS decrypts (Enc_i) this request using ECIES's $C_iK_{pu_i}$ and CSK_{pr_i}. It checks timestamp (TS_{CS}-$TS_{C_i} \leq \triangle T$) to make sure that this request arrived at an appropriate time and without delay

- In the registration and login protocol, the CS examines the random OTP_i in the dataset. If OTP_i exists, the user is considered legitimate for the registration process. After that, the CS deletes OTP_i from the dataset to prevent it from being used at subsequent logins. If OTP_i is not found, it discards the connection. In the login protocol, CS examines the $UP_{C_i}^{CS}$ and $MP_{C_i}^{CS}$ and then tests their association with the MAC address in the dataset. If GM is found, the CS completes the steps of this protocol; otherwise, it cancels the connection

- The CS needs to ensure that the user's device is legitimate within the network. The CS computes the signature value to make sure that the MAC address is native and non-modified ($CSSig_1 = h(\text{"}Real\_MAC\text{"} \| N_{C_i} \| TS_{C_i})$. It examines the result of the computed signature ($CSSig_1$) with the C_i (C_iSig_1) signature. If the result signatures are identical, then the device is legitimate and the MAC address did not change. In the registration and login protocol, the CS stores this address in dataset for use and checks the next times in the login protocol
- The CS performs the computation operation the $tmpPW_i \oplus N_{C_i} \oplus GM \oplus CSSig_1$ to extract the PW_i value and then uses this value to produce a second signature ($CSSig_2$)
- The CS computes a second signature operation ($CSSig2= h(GM \| N_{C_i} \| TS_{C_i} \| CSSig_1 \| UP_{C_i}^{CS} \| MP_{C_i}^{CS} \| OTP_i \| PW_i)$ to guarantee that all the encrypted information is not changed. Then, it compares the computed signature ($CSSig_2$) with the received signature in the request ($CiSig_2$). If the signatures are identical, then the information for this request is unchanged or not tampered. In the login protocol, OTP_i and GM are not added to the signature. At this point, the CS prepares to send the user's authentication request to the AS.

- **Authentication Protocol**

  In this protocol, RAMHU needs to link pseudonyms and passwords with real information for users in the AS's datasets. Note that, the user information (such as name, age, address, mobile number, and passwords) are stored in a separate server (AS) and multi pseudonyms are used to prevent detection and tracking of user information. This protocol is illustrated in the following steps (Figure 5.5 shows the authentication protocol in RAMHU):

  CS side:

  - The CS computes a new timestamp (TS_{CS}) to ensure the fresh authentication request
  - The CS replaces C_i's pseudonyms ($UP_{C_i}^{CS}$ and $MP_{C_i}^{CS}$) with CS's pseudonyms (UP_{CS}^{AS} and MP_{CS}^{AS}) to prevent attackers from tracking the authentication request. It generates random nonce (N_{CS}) to ensure an anonymous authentication request

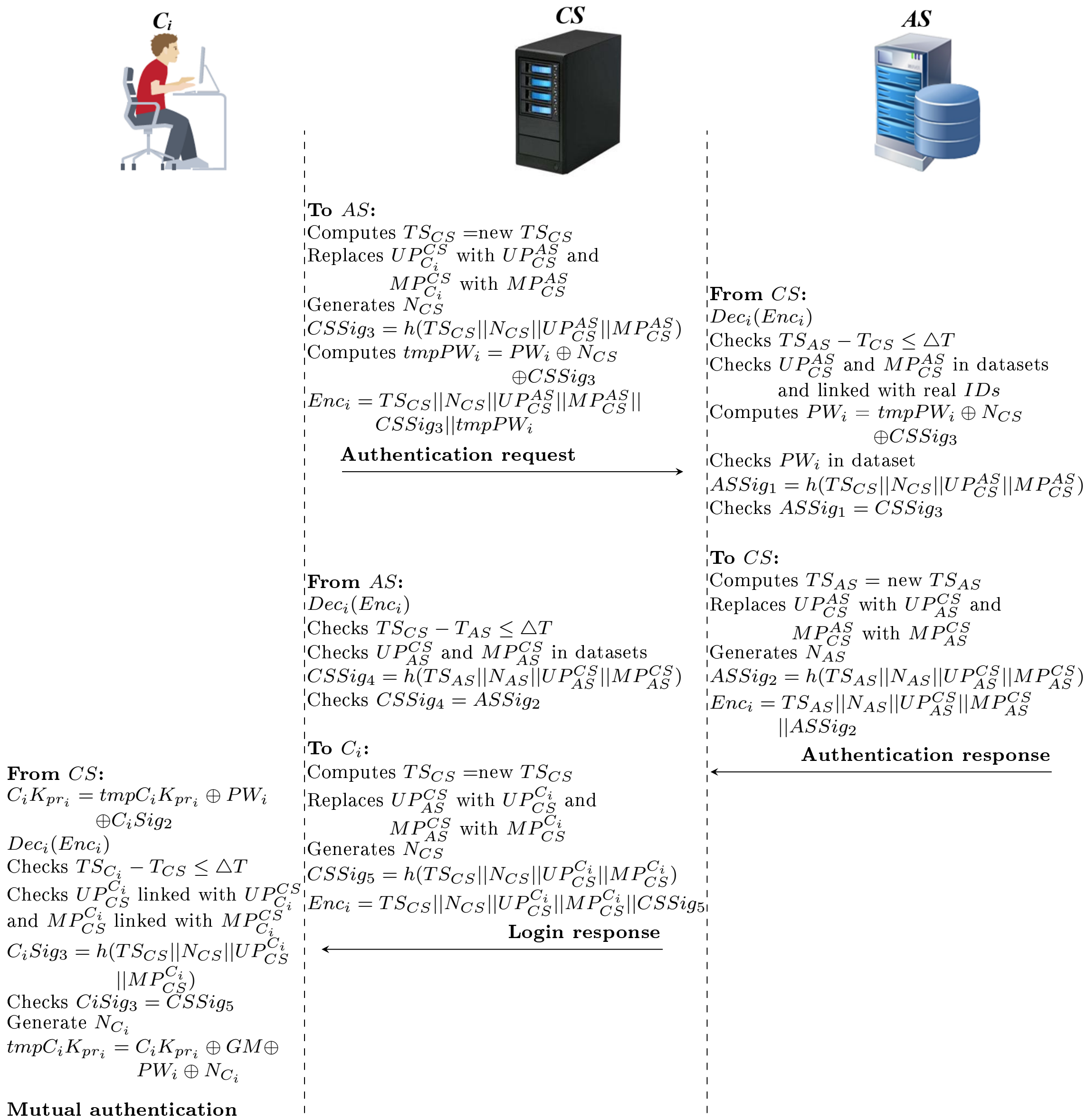


Figure 5.5: Authentication protocol

- The CS signs security parameters by PHOTON-256 ($CSSig_3 = h(TS_{CS} \| N_{CS} \quad \| UP_{CS}^{AS} \| MP_{CS}^{AS})$ to prevent the modification of authentication request data
- The CS computes temporary PW_i by the computation of $PW_i \oplus N_{CS} \oplus CSSig3$
- The CS encrypts the authentication request information ($Enc_i = TS_{CS} \| N_{CS} \| UP_{CS}^{AS} \quad \| MP_{CS}^{AS} \| CSSig_3 \quad \| tmpPW_i$). It sends an authentication request to verify the user information in the AS.

AS side:

- Upon receiving the authentication request, the AS decrypts that request

(Enc_i) using ECIES's ASK_{pr_i} and CSK_{pu_i} to obtain the authentication information clearly

- It checks the timestamp by ($TS_{AS} - TS_{CS} \leq \triangle T$) to ensure that the authentication request is not delayed. The AS checks the pseudonyms (UP_{CS}^{AS} and MP_{CS}^{AS}) sent from the CS and correlates it with the user's real identifier (UID_i and MID_i) in the datasets. The AS extracts the user's password from the equation $PW_i = tmpPW_i \oplus N_{CS} \oplus CSSig_3$. After that, it checks matching PW_i in the dataset. The AS computes the value of the signature based on the authentication information ($ASSig_1 = h(TS_{CS} \| N_{CS} \| UP_{CS}^{AS} \| MP_{CS}^{AS})$) by PHOTON-256. The AS compares the computed value of the signature ($ASSig_1$) with the value of the received signature ($CSSig_3$). If the signature values are identical, the user information in the request for the signature is unmodified. At this point, the AS considers this user to be legitimate and reliable
- The AS prepares a response to authenticate the request of that user. It computes a new timestamp (TS_{AS} = new TS_{AS}) to prevent delayed or replayed requests at later times. The AS replaces the CS's pseudonyms (UP_{CS}^{AS} and UP_{CS}^{AS}) received with the AS's pseudonyms (UP_{AS}^{CS} and MP_{AS}^{CS}) to hide user information. It generates a new random nonce (N_{AS}) to add anonymity and prevent attacks from encryption and signature analysis
- The AS computes a signature ($ASSig_2 = h(TS_{AS} \| N_{AS} \| UP_{AS}^{CS} \| MP_{AS}^{CS})$) to prevent modifications of the authentication response information
- The AS encrypts the authentication information ($Enc_i = TS_{AS} \| N_{AS} \| UP_{AS}^{CS} \| MP_{AS}^{CS} \| ASSig_2$) and sends the authentication response to the CS to complete the authentication process.

CS side:

- The CS decrypts the authentication response (Enc_i) received from the AS. It checks the timestamp value ($TS_{CS} - TS_{AS} \leq \triangle T$) to prevent late authentication responses. It examines the UP_{CS}^{AS} and MP_{CS}^{AS} in datasets to complete the process of linking multi pseudonyms to the user
- The CS computes the signature $CSSig4 = h(TS_{AS} \| N_{AS} \| UP_{AS}^{CS} \| MP_{AS}^{CS})$ for authentication response information. It compares the computed result of the signature ($CSSig_4$) with the value of the received signature ($ASSig_2$). If the signature values match, then the authentication response information is unchanged

- After this point, the CS initiates a login response request. It computes the value of the new timestamp ($TS_{CS} = newTS_{CS}$). It replaces the AS's pseudonyms (UP_{AS}^{CS} and MP_{AS}^{CS}) received with $UP_{CS}^{C_i}$ and $MP_{CS}^{C_i}$. It generates a new random nonce (N_{CS}) to hide the encryption and signature information
- The CS computes a new signature value ($CSSig_5 = h(TS_{CS}\|N_{CS}\|UP_{CS}^{C_i}\|MP_{CS}^{C_i})$) to protect login response information from modification
- The CS encrypts the login response information ($Enc_i = TS_{CS}\|N_{CS}\|UP_{CS}^{C_i}\|MP_{CS}^{C_i}\|\ CSSig_5$) and sends this response to C_i.

C_i side:

- C_i extracts the private key ($C_iK_{pr_i}$) from the $tmpC_iK_{pr_i} \oplus PW_i \oplus C_iSig_2$, and decrypts the login request (Enc_i) received from the CS
- C_i checks timestamp ($TS_{C_i} - TS_{CS} \leq \triangle T$) to ensure that the login response is not late or replayed
- C_i checks that CS's pseudonyms ($UP_{CS}^{C_i}$ and $MP_{CS}^{C_i}$) are received in the dataset and associated with $UP_{C_i}^{CS}$ and $MP_{C_i}^{CS}$. At this point, RAMHU applies multi pseudonyms among model entities (Ci, CS, and AS) to prevent traceability in linking the real information of the user with pseudonyms
- C_i calculates the value of the signature $CiSig_3 = h(TS_{CS}\|N_{CS}\|UP_{CS}^{C_i}\|MP_{CS}^{C_i})$ for login response information. It compares the result of the computed signature ($CiSig_3$) and the value of the received signature ($CSSig_5$). If the signature values are identical, namely, the login response information is unmodified or not tampered, C_i accepts the login response; otherwise C_i discards the login response. Then, C_i hides its private key by $C_iK_{pr_i} \oplus GM \oplus PW_i \oplus N_{C_i}$ after generating a random nonce to prevent the detection of the private key if the device is hacked. At this point, if all processes are achieved correctly, then all requests are considered legitimate and reliable through the implementation of mutual authentication.

- **Password Update Protocol**
  The protocol for changing PW_i is important in any HC system for two reasons. First, preventing the use of PW_i fixed for a long time, reducing the guessing attacks. Second, changing PW_i gives users more flexibility in choosing the appropriate PW_i. This process requires strict security measures to protect

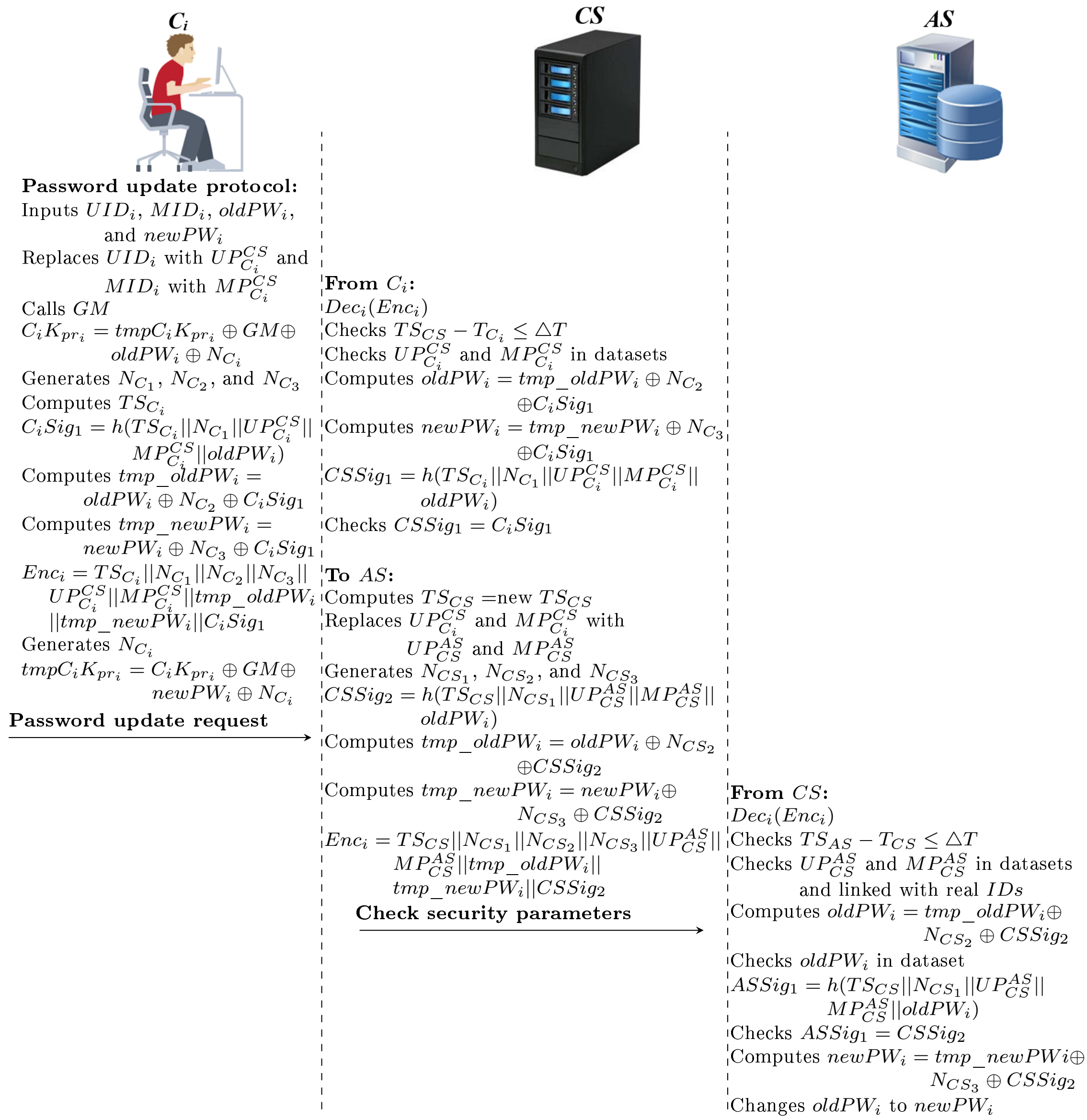


Figure 5.6: Password update protocol

new PW_i. RAMHU provides the legitimate user with a mechanism to change his/her password at any time. If the user wants to change his/her PW_i, the following illustration describes the new PW_i protect procedures in a secure manner (Figure 5.6 shows the password update protocol in RAMHU):

– C_i side: Ci enters UID_i, MID_i, old PW_i, and new PW_i. It replaces UID_i and MID_i with pseudonyms to hide the real user information. C_i calls MAC address, then extracts the private key from the $tmpC_iK_{pr_i} \oplus GM \oplus oldPW_i \oplus N_{C_i}$ to use in the encryption process of the password update request. C_i generates three random nonces (N_{C_1}, N_{C_2}, and N_{C_3}) and computes a new timestamp (TS_{C_i}). C_i computes the signature value ($C_iSig_1 = h(TS_{C_i}\|N_{C_1}\|UP_{C_i}^{CS}\|MP_{C_i}^{CS}\|oldPW_i$) by PHOTON-256 based on the parameters of the password change request. It applies an anonymity mechanism to old PW_i and new PW_i ($tmp\_oldPW_i = oldPW_i \oplus N_{C_2} \oplus CiSig_1$ and $tmp\_newPW_i = newPW_i \oplus N_{C_3} \oplus CiSig_1$) to hide passwords and not explicitly send it in the PW_i change request. It encrypts the PW_i change request ($Enc_i = TS_{C_i} \| N_{C_1} \| N_{C_2} \| N_{C_3} \| \ UP_{C_i}^{CS} \| MP_{C_i}^{CS} \| \ tmp\_oldPW_i \| \ tmp\_newPW_i \ \| C_iSig_1$) and sends it to CS, C_i then hides its private key by $C_iK_{pr_i} \oplus GM \oplus newPW_i \oplus N_{C_i}$

– CS side: The CS receives and decrypts a password update request (Enc_i). It examines the timestamp to prevent delayed requests, and examines $UP_{C_i}^{CS}$ and $MP_{C_i}^{CS}$ in datasets. The CS extracts old PW_i and new PW_i from $tmp\_oldPW_i \oplus N_{C_2} \oplus CiSig_1$ and $tmp\_newPW_i \oplus N_{C_3} \oplus CiSig_1$. Then, it computes signature value ($CSSig_1$) depending on $TS_{C_i}, N_{C_1}, UP_{C_i}^{CS}, MP_{C_i}^{CS}$, and $oldPW_i$ to check the matching between $CSSig_1$ and C_iSig_1. Similarly, in the authentication protocol, the CS computes TS_{CS} and replaces the pseudonyms. The CS generates three nonces N_{CS_1}, N_{CS_2}, and N_{CS_3} and then it computes signature value ($h(TS_{C_i} \| N_{C_1} \| UP_{C_i}^{CS} \| MP_{C_i}^{CS} \| oldPW_i$) and hides old PW_i and new PW_i by $oldPW_i \oplus N_{CS_2} \oplus CSSig_2$ and $newPW_i \oplus N_{CS_3} \oplus CSSig_2$. It encrypts the password update request with security parameters (TS_{CS}, N_{CS_1}, N_{CS_2}, N_{CS_3}, UP_{CS}^{AS}, MP_{CS}^{AS}, tmp_oldPW_i, tmp_newPW_i, and $CSSig2$) and sends it to the AS

– AS side: The AS receives the password update request and decrypts (Enc_i) this request with ASK_{pr_i}, and CSK_{pu_i}. It checks the time delay, and then it checks link pseudonyms with real user information. It extracts old PW_i from $tmp\_oldPW_i \oplus N_{CS_2} \oplus CSSig_2$ and then it checks old PW_i in dataset. It computes the signature value $ASSig_1 = h(TS_{CS} \| N_{CS_1} \| UP_{CS}^{AS} \| MP_{CS}^{AS} \| \ oldPW_i)$, and then, it compares the calculated result ($ASSig_1$) with the result of the received signature ($CSSig_2$). If identical, the signature is true; otherwise, the AS rejects the PW_i change request. The AS performs the calculation $tmp\_newPW_i \oplus NCS_3 \oplus CSSig_2$ to obtain the new PW_i value. If all previous checks are validated, the AS changes old PW_i to new PW_i.

- **Revocation Protocol**

This protocol can be completed by C_i, or the AS. If C_i wants to revoke his account from the HC system after completing his/her duties, such as a research doctor who uses the system for a limited period and then cancels his account after the completion of his duties. Additionally, the AS can revoke the account of any user who performs suspicious activities (internal attacks) that are not within his/her privileges, such as a nurse who wants to access the personal information of a particular doctor, or patient. Furthermore, the user can ask the authorities provider (AS) to cancel his/her account information that is associated with his/her data (note that the patients' data history remains stored in the DS even after the completion of the revocation protocol). The protocol of revocation is extremely important in restricting the malicious activities of any healthcare system. RAMHU includes a revocation protocol to provide strict security procedures in protecting users' authentication information (Figure 5.7 shows revocation protocol in C_i side in RAMHU):

Revocation from C_i side:

- C_i enters UID_i, MID_i, and PW_i and then replaces UID_i with $UP_{C_i}^{CS}$ and MID_i with $MP_{C_i}^{CS}$. It chooses the revocation reason (RR_i) from the drop-down list (such as ending the researcher's study, ending a satisfactory condition, resigning a professional, changing a health institution, and the unwillingness of a patient to use the system). These reasons have converted to signatures using PHOTON to get MDs with a 256-bit in the dataset. C_i computes new TS_{C_i}, N_{C_1}, N_{C_2}, and N_{C_3}. Then, C_i performs the process of $RR_i \oplus N_{C_1}$ to add randomness for RR_i. Using this procedure is tremendously useful in tightening security and distinguishing the roles of users (patients or professionals) in the CS. It computes the signature $C_iSig_1 = h(TS_{C_i} \| N_{C_1} \| UP_{C_i}^{CS} \| MP_{C_i}^{CS} \| RR_i \| \ PW_i \| \ "delete")$. C_i performs a computation to hide the RR_i ($tmpRR_i = RR_i \oplus N_{C_2} \oplus C_iSig_1$). C_i computes the temporary PW_i value of $PW_i \oplus N_{C_3} \oplus C_iSig_1$ to hide the PW_i value. Additionally, it computes encryption ($Enc_i = TS_{C_i} \| N_{C_1} \| N_{C_2} \| N_{C_3} \| UP_{C_i}^{CS} \| MP_{C_i}^{CS} \| tmpRR_i \| \ tmpPW_i \ \| C_iSig_1$) and sends a revocation request to the CS. Then, it hides a private key, in the same way, in the password update protocol
- The CS decrypts (Enc_i) and computes the timestamp and examines $UP_{C_i}^{CS}$ and $MP_{C_i}^{CS}$ in the datasets. It obtains the RR_i from the computation equation $RR_i = tmpRR_i \oplus N_{C_2} \oplus CSSig_1$. It extracts PW_i from $tmpPW_i \oplus N_{C_3} \oplus C_iSig_1$ and checks PW_i matching in the dataset. CS computes the signature ($CSSig_1 = h(TS_{C_i} \| N_{C_1} \| UP_{C_i}^{CS} \| MP_{C_i}^{CS} \| RR_i \| PW_i \| \ "delete")$ and then compares the result of the signatures. The CS computes operation $RR_i \oplus N_{C_1}$ to use RR_i's signature in order to compare the RR_i with the user's R_i. If all operations are achieved and validated correctly, the CS sends a request to the AS to check the

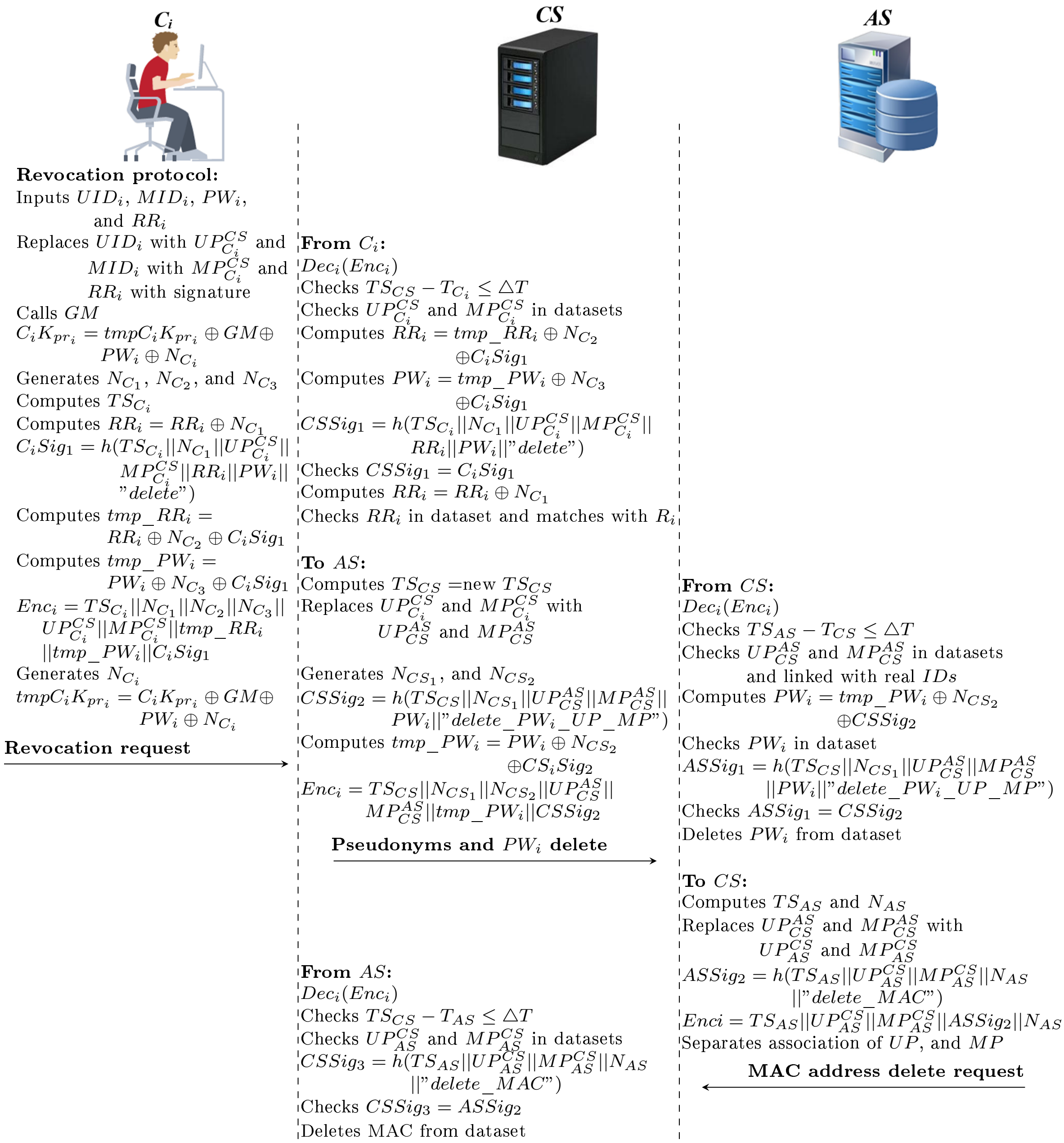


Figure 5.7: Revocation protocol

pseudonym's association with real information and check PW_i. The AS deletes association between pseudonyms and information, and deletes the user's PW_i from dataset. The AS sends MAC delete request to the CS. Upon receiving the MAC delete request, the CS decrypts this request, then checks the security parameters and signature. If all operations are achieved and validated correctly, the CS deletes the MAC address from the dataset. At this point, this user cannot perform an authentication process in RAMHU.

Revocation from AS side:

– The AS deletes the association between the user information and pseudonyms and PW_i in the datasets. It sends an encrypted request (Enc_i) to the CS to delete the device's MAC address for this user. After the completion of this protocol, the user is considered illegal and cannot access the healthcare services.

5.3 Summary of the Chapter

In this chapter, we have provided details of the new authentication scheme and described our network model in the HC application. Additionally, RAMHU's goals to provide authentication requirements was detailed. Following this, we presented a set of protocols. Finally, the RAMHU scheme methodology was explained in detail.

Chapter 6: Authorisation of HC Users with Differentiated Access Control

In Chapter 5, we elaborated on a scheme that authenticates HC user identity and the EHR repository. In this chapter, we will explain another integral component of our proposed HC application. This scheme is responsible for authorising various HC users, including patients, with differentiated access control to the EHR repository. The objective of this authorisation method is to provide HC users with more efficient and secure services, as well as preserving patient privacy.

6.1 Data Security in EHR Systems

The EHR/EMR technology is a recent advance in the HC industry that digitizes patients' health or medical data and stores these records in a database or repository. In this way, patients' health records and history can be accessed much more accurately and quickly. In emergency, this may save lives. EHR systems include identifications and patients' data that require authorisation privileges to determine access control for authorised users (Calvillo-Arbizu et al. 2014). Accurate medical data is essential for diagnosing diseases and determining the condition of patients(Alhaqbani & Fidge 2008, Riedl et al. 2008). Any unauthorised change to this data causes health problems for patients. In addition, the penetration of medical records of patients with diseases such as HIV infection or dermatological conditions can lead to discrimination, harassment, or even death of the patient if the diagnostic data changes during the transition from client to server (Neubauer & Heurix 2011, Riedl et al. 2008). In a broad sense, a terrorist may cause national instability by disclosing patients' data, changing the data, destroying the data, or impersonating some patients (Quantin et al. 2011). HC systems, in particular EHR systems, should provide end-to-end privacy for patients' data. Also, data storage and authorisation policies for patients in a central server yield data management gains but are an attractive target for hackers (Quantin et al. 2011). Therefore, there should be security mechanisms to protect the privacy of the patient as well as to prevent the penetration of policies on the server.

The use of patient data for various purposes, such as consultations, access by a relative or caregiver, research, and emergency (secondary or indirect use) are a major challenge for authorisation systems. For example, a researcher should not exceed the limits of privacy granted to him/her (Calvillo-Arbizu et al. 2014). In an emergency, when the patients' doctor is unavailable or the patient does not have the capacity to give consent to another doctor, the patient's privacy is seriously compromised (Sun et al. 2011). Also, if the patient is incapacitated, a relative is responsible for receiving the patient's data (Riedl et al. 2007). Sometimes, the doctor also needs to consult another doctor to treat a patient's condition. All these cases can result in the intrusion and penetration of data. The sharing of medical records among users of the EHR system allows patient data to be misused or abused by malicious breaches (Rezaeibagha et al. 2015). There are many examples of the penetration of patients' medical records for patients by intruders, as mentioned in Section 2.3.2 (Koczkodaj et al. 2018, U.S. Department of Health and Human Services 2018). These penetrations show that the HC system requires a high level of security. Furthermore, an internal attacker penetrates medical records more easily than an external attacker because each practitioner has a privilege that allows him/her to access the server system.

6.2 Overview of Requirements of Access Control

Many access control models have been used in EHR, such as mandatory access control (MAC), discretionary access control (DAC), role based access control (RBAC), and attribute based access control (ABAC), and each model has specific authorisation mechanisms for data access (Gajanayake et al. 2014). In this project, we adopted the integration of the RBAC and ABAC to support a security level based on both role and user attributes. Therefore, EHR systems require mechanisms to ensure the privacy of patients' data while protecting authorisation policies and HC provider requests (Fernández-Alemán et al. 2013). In order to develop a successful project, privacy must be provided to the patient via the following measures:

1. Preventing attackers from accessing patient data and making data anonymous in case attackers do gain access to the data (i.e., external attacks)

2. Preventing legitimate users from exceeding their privileges (i.e., internal attacks)

3. Securing all requests, policies, and data of the change on the server or during the transfer between the clients and server to ensure the accuracy and reliability of patient data

4. Applying anonymity to requests and policies to hide users' identities

5. Applying a random pseudonym to requests, policies, and data to separate data associated with the real attributes of patients.

6.2.1 Access Control for the EHR Repository

We developed a **P**seudonymization and **A**nonymization with the **X**ACML (PAX) modular system, which depends on client and server applications. The characteristics of the proposed authorisation scheme can be summarised as follows:

- Combining ABAC and RBAC
  In this scheme, we integrate two existing models (ABAC and RBAC) to develop a system that provides handling of patients' information at the coarse-grained and fine-grained levels. Our model fits the privacy and security requirements for medical records in the EHR by merging a user's ID with the role as a single attribute entered in the signature to identify subjects and objects

- Separating users into two sets
  We have proposed separating users into direct and indirect sets for patients' records to allow the server to distinguish between users' requests. This significantly reduces the penetration rate of internal attacks

- Using ECDSA's signatures with XACML
  The anonymity property has been applied to the requests and policies of subjects. This feature was used during the implementation of the ECDSA signature algorithm with XACML to prevent attackers from determining the identity of HC providers (to prevent knowledge of the relation between a physician with a particular patient)

- Using Shamir scheme with signatures
  We used the Shamir scheme with the ECDSA signatures in the third protocol for authorising indirect users. This procedure is necessary to verify unauthorised users of patient data who could be conducting serious attacks on the EHR system

- Using random pseudonym with patient data
  The pseudonym property has been applied to the requests and policies of subjects and resources. This feature prevents hackers from knowing that the data belongs to a particular patient (separating data from real attributes).

6.3 Our Proposed Authorisation Model

In this section, we provide details of our new authorisation scheme that supports security and privacy mechanisms to ensure legitimate user authorisation in HC applications. This section will be divided into the network model, application of privacy concepts and PAX authorisation protocols for users.

6.3.1 Users Access Control Model

As shown in Figure 6.1, Pseudonymization and Anonymization with the XACML (PAX) is an authorisation system that works with EHR. The network model consists of four entities: client (C_i), central server (CS), attributes server (AS) and data server (DS). These entities communicate with each other in the PAX framework to accomplish authorisation and privacy preservation of users when accessing datasets. The CS is the portal that prevents users from directly accessing to both the AS and DS. Patients' data are stored on the data server (DS) and are fully separated from the attributes of the users (patients and HC providers) that are stored on the attributes server (AS). Each C_i creates an access request and sends it to the CS. Then, the CS verifies the authorisation information for the user's request. If this request is valid, the CS sends the authorisation request to the AS for an evaluation; otherwise, the CS sends the "deny" response to C_i. When the AS receives the authorisation request from the CS, the AS evaluates the access request by PDPs modules, verifies signatures, pseudonyms, and other security parameters. If all evaluates and tests are valid, the AS sends a request to the DS to retrieve the patient data; otherwise, the AS sends the "deny" response to the CS. After that, the DS checks for signatures (Sigs) and privacy parameters (PP), if all operations are correctly performed, the DS sends the required data with pseudonyms and Sigs to AS which in turn sends the "permit" response to C_i by the CS to allow access to the dataset. The authorised user will receive the "permit" response and the copy of the required data.
The PAX system uses two PDPs (PDP1 and PDP2) to implement the user authorisation process, as shown in Figure 6.1. In this scheme, we focus on securing requests and policies to provide a high level of user privacy. PAX depends on the Balana Project, which is the only open source project that implements XACML v3.0.

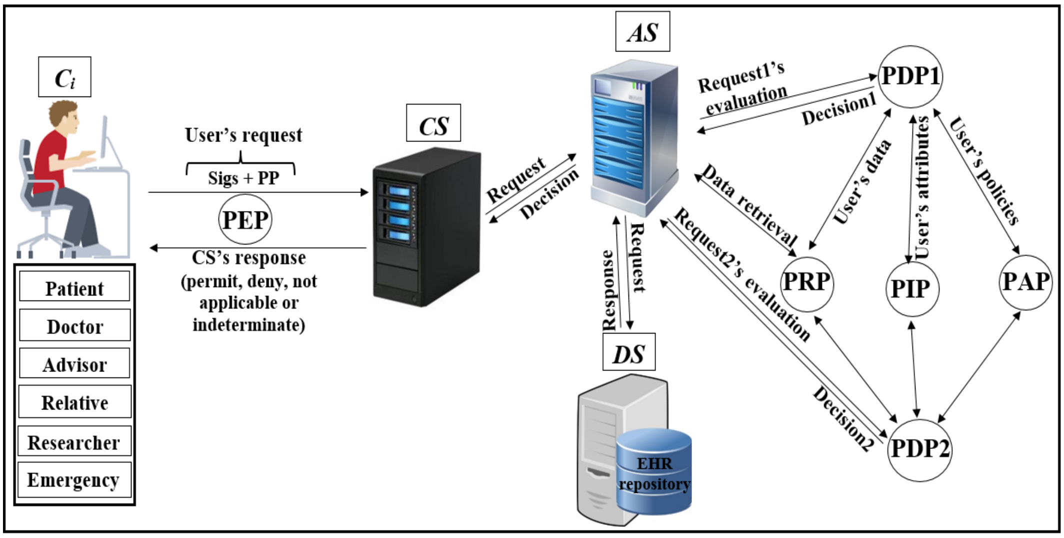


Figure 6.1: PAX model

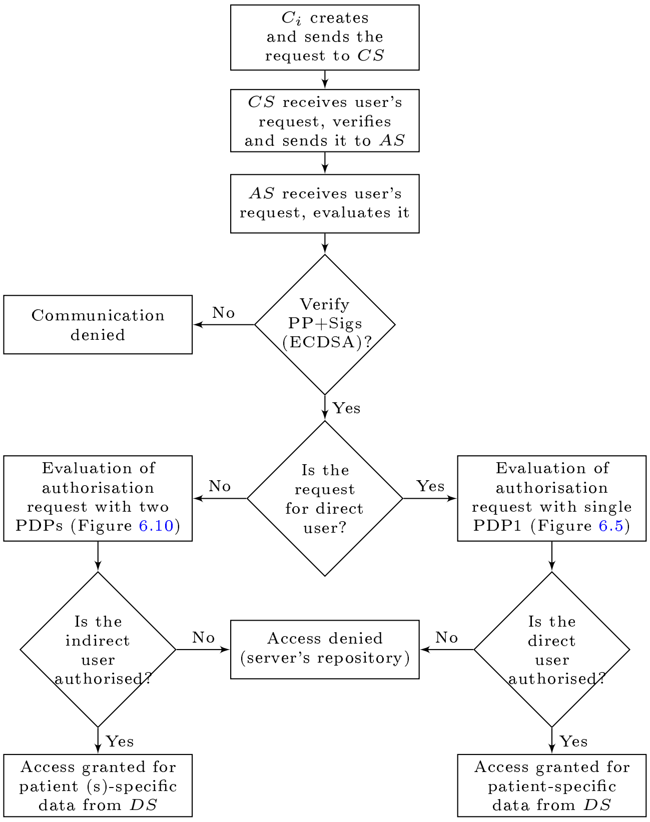


Figure 6.2: Authorisation of direct and indirect users

Privacy and security are essential requirements of EHR. In PAX, we are careful to provide high-level privacy and security mechanisms to authorise users.

6.3.2 Design Goals of PAX

There is a set of security features included in PAX:

- **Integrity and non-repudiation of requests**: User requests and policies need protection from change or repudiation. We used the ECDSA algorithm to sign user attributes. Any change in the signatures will be detected in the server because the server checks the user requests before authorising access to the data. In addition, the signatory party cannot deny its signature. These features make the system immune against to internal changing attacks

- **Authentication and authorisation of requests**: Each EHR requires authentication and authorisation properties to protect medical records from unauthorised access. We applied ECDSA to the XACML v3.0 to support these properties in PAX. The use of signatures in XACML between the C_i and the CS, AS and DS support user authentication in addition to the use of policies and rules to identify authorised users and the level of access granted to them

- **Confidentiality and anonymization**: One of the security features of hiding information is confidentiality. We applied ECDSA to add confidential requests to subjects and objects, and we added a Shamir scheme (backup or fail-open mechanism) to provide anonymity of SS_s to users of the EHR system. This process prevents the attacker from seeing explicit attributes and does not allow a hacker to know the user-configured SS for any healthcare provider. A Shamir scheme ensures the anonymity of the signature. This backup mechanism enables indirect users to access protected health information (PHI) with privacy and security

- **Pseudonymization**: A patient's privacy requires the separation of personal information from their data. Pseudonymization prevents the intruder from knowing the data belonging to any of the patients. PAX supports pseudonym in both subjects' and objects' attributes using pseudonyms for real attributes. This feature supports the privacy of patients' data

- **Audit and activities**: PAX records all user activities (requests and responses) accessing medical records. It monitors user activity, including the number of access times, the result of the decision, and the amount of data required. The audit process is important for any healthcare system in determining users' activities. PAX stores and organizes requests and responses for each user (patient, doctor, advisor, relative, researcher, and emergency doctor) separately to facilitate the management of these activities.

6.3.3 Implementation of PAX

In this section, we will introduce the privacy concepts in PAX.

- **EHR's Users in PAX**
  Security and privacy address where, when, and why data is available and who can access the data repository. Patients and HC providers require services that are efficient, fast, and continuous and at the same time incorporate strict restrictions to determine data access. Therefore, AC to medical records has several challenges in terms of security and privacy:

  1. Legitimate users should not exceed their privileges
  2. Users' roles in the EHR system should be defined. For example, a doctor can have several roles, such as an emergency doctor and a researcher doctor
  3. Data should be anonymous when it reaches the wrong user due to misuse or attacks
  4. Compliance with medical standards for EHR (such as HIPAA) is essential.

  In PAX, we divide users into two categories:

  – Direct users: These users include those who are directly associated with the data, such as the patient and the doctor
  – Indirect users: These users include those who are not directly and continuously associated with the data, such as advisors, patients' relatives, researchers, and emergency doctors.

  Although PAX includes both categories of users, this project focuses on indirect users (Figure 6.2 shows a flow chart for authorisation of direct and indirect users in PAX). Any HC system can be exposed to an internal attack by indirect users if there are no security and privacy mechanisms to prevent them.

- **User Pseudonym in PAX**
  Authorisation systems require a lightweight mechanism to prevent intruders from distinguishing specific patients' data without the complications of encryption and anonymous mechanisms. To address this problem, we apply random pseudonyms with PAX to separate the association between patients' attributes and their data. The medical records transmitted between the client and server do not contain any patient attributes. This prevents attackers from identifying patients.

  In PAX, we propose the use of four datasets: the first for user attributes (patients and HC providers), the second for applying pseudonyms to users,

Table 6.1: Internal and external pseudonyms of users

Users	UR	CN	Internal pseudonym	RN	UN	External pseudonym
patient	p		p_1 ... p_n			
doctor	d		d_1 ... d_n			
advisor	a		a_1 ... a_n			
relative	pr	1 ... n	pr_1 ... pr_n	1 ... n	1 ... n	1 ... n
researcher	r		r_1 ... r_n			(48-bit)
emergency	e		e_1 ... e_n			
Shamir	-		-			

Table 6.2: Parts of SP and OP

SP						OP					
RN_{sp}			UN_{sp}			RN_{op}			UN_{op}		
RN_{sp_l}	RN_{sp_m}	RN_{sp_h}	UN_{sp_l}	UN_{sp_m}	UN_{sp_h}	RN_{op_l}	RN_{op_m}	RN_{op_h}	UN_{op_l}	UN_{op_m}	UN_{op_h}

the third for user policies (on the AS) and the fourth for patient data (on the DS).

When the EHR system wants to add a new HC provider or patient, the PAX randomly generates a pseudonym for that user and adds it to the second dataset. Suppose that we have a dataset for random pseudonyms, as in Table 6.1. PAX generates pseudonyms (such as $p429$ or $d761$) for patients or HC providers during the addition of a letter representing the user's role (UR) such as p or d plus a random client's number (CN). Each subject's pseudonym (SP) and object's pseudonym (OP) consists of UR and CN (internal pseudonym), which are not transferred between entities and are used for policy verification at the AS. XACML's request in PAX depends on the SP and OP (external pseudonym), and both SP and OP are divided into the role's number (RN) and user's number (UN) (after replacing UR with RN and CN with RN) and the latter are segmented into three parts (low (l), medium (m), and high (h)) with length 8 bits per part as in Table 6.2. These pseudonyms are associated with the user IDs. It enables users to access a specific patient's data without exceeding granted privileges and rights.

- **Using ECDSA's Signatures**

  PAX uses ECDSA (NIST prime-256) with requests and policies to ensure that security requirements apply to the privacy of patients' data. We have applied ECDSA signatures with subjects' and objects' attributes to ensure integrity property to prevent changing attributes in requests and policies, authentication property to prevent external attackers and non-repudiation property to prevent authorised users from denying their requests to receive medical records. The application of security requirements is very important in systems, such as HC systems, that use sensitive data. In PAX, the C_i signs the request with pseudonyms (RN and UN), and the servers (CS and AS) verify the request's

Sigs. If valid, the AS assigns the request to the PDPs engines (after replacing Sigs (external pseudonym) with Sigs (internal pseudonym)) in XACML v3.0; otherwise, the request is rejected. PAX uses ECDSA Sigs to hide parts of SP and OP when exchanging XACML requests between PAX entities. The high performance and security level make this algorithm suitable for application in large systems (such as EHR).

- **Policies Administration in PAX**
  System Administrator is responsible for creating policies for HC providers and patients in the AS by PAP. Policy in PAX consists of the policy ID, subject, object, and rules for policy implementation. The first process in the PAX system is to create datasets for pseudonyms and attributes for all users. The process of creating policies depends on previous datasets. PAX uses ECDSA to generate a signature of SP (S_{sp}) and a signature of OP (S_{op}) based on the pseudonyms (UR and CN) for both SP and OP. Creating signature-based policies and pseudonyms protects policies on the server in a way that is immune to internal and external attacks (policies do not depend on users' real attributes). For example, the system administrator creates a user policy by entering the doctor's name and UR and patient's name, and PAX creates this policy as shown in Figure 6.3. The policy parameters are highlighted in green: d20 represents the SP and uses as the policy's ID; the first long 128-bit hexadecimal number represents the S_{op} and the second long 128-bit hexadecimal number represents the S_{sp}. This policy can include a set of rules such as determining the date of data access, the time specified on a given day, or the number of access times.

- **Client Requests and Server Responses**
  PAX's users must create an authorisation request to access medical records. This request consists of the subjects' and objects' attributes. The C_i application in PAX uses the parts of RN and UN as a single attribute to generate the ECDSA's Sig for the subjects and the objects. Figure 6.4 shows the client's request to access patient data (where the request parameters are highlighted in green; $C_iS_{2_{tm}}||RN_{op_{tm}}||UN_{op_{tm}}||N_C\ ||C_iS_{4_{tm}}$ in resource segment represents the object's attributes, and the $C_iS_{1_{tm}}||RN_{sp_{tm}}\ \ ||UN_{sp_{tm}}||N_C\ ||TS_{C_{tm}}||SN_{C_{tm}}$ in access-subject segment represents the subject's attributes). Also, the C_i application uses a part of RN_{sp} to explain to the AS the user's role to determine the desired policy after verifying the Sigs. Then, the C_i sends the request to the AS by CS for evaluation. The AS evaluates the request in the PDP engines, and the response (permit or deny) returns to the C_i by CS.

```xml
<Policy xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" PolicyId="permitPolicyFord20"
RuleCombiningAlgId="urn:oasis:names:tc:xacml:3.0:rule-combining-algorithm:first-applicable" Version="1.0">
  <Target>
    <AnyOf>
      <AllOf>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">36b6e95b598c43eae10c6e2369d29a3df8
            263e6a75336e20bf8cac2a791ab269bc2664e625636e70389546de67135074c851b6d15751cd29b0a76df47eec2887
           </AttributeValue>
          <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
           Category="urn:oasis:names:tc:xacml:3.0:
           attribute-category:resource" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
        </Match>
        <Match MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue  DataType="http://www.w3.org/2001/XMLSchema#string">0284894f6327dcca8ebd16a2e8a783ad7
            1f911f9d4b8af86b826048eb9c88846075a37b87654ac0ec1f58c35dbc06587d0b6a77d4303de6e19d1bf1fd715ab07
           </AttributeValue>
          <AttributeDesignator AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" Category="urn:oasis:names:tc:xacml:1.0:
           subject-category:access-subject" DataType="http://www.w3.org/2001/XMLSchema#string" MustBePresent="true"/>
        </Match>
      </AllOf>
    </AnyOf>
  </Target>
  <Rule Effect="Permit" RuleId="permit1"/>
</Policy>
```

Figure 6.3: PAX policy

```xml
<Request xmlns="urn:oasis:names:tc:xacml:3.0:core:schema:wd-17" CombinedDecision="false" ReturnPolicyIdList="false">
  <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:action">
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id" IncludeInResult="false">
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">access</AttributeValue>
  </Attribute>
  </Attributes>
  <Attributes Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject">
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id" IncludeInResult="false">
        <AttributeValue DataType="http://www.w3.org/2001/
           XMLSchema#string">C_iS_{1_{tm}}||RN_{sp_{tm}}||UN_{sp_{tm}}||N_C||TS_{C_{tm}}||SN_{C_{tm}}
        </AttributeValue>
        </Attribute>
  </Attributes>
      <Attributes Category="urn:oasis:names:tc:xacml:3.0:attribute-category:resource">
      <Attribute AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id" IncludeInResult="false">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">C_iS_{2_{tm}}||RN_{op_{tm}}||UN_{op_{tm}}||N_C||C_iS_{4_{tm}}
        </AttributeValue>
      </Attribute>
  </Attributes>
</Request>
```

Figure 6.4: C_i's request

- **Using Shamir Scheme**

  In PAX, we implemented the Shamir scheme to increase the level of security for indirect users (advisors, patients' relatives, researchers, and emergency). Indirect users are legitimate users who can perform an internal attack because of the rights granted to them. PAX uses ECDSA to sign all signatures of HC users to create a master signature (MS). Then, PAX uses the Shamir scheme to generate secrets sharing (SS_s) from a MS. Each indirect user receives SS via a secure communication channel. C_i needs a set of SS_s to reproduce the MS. PAX uses $TH = 3$, which means that the randomly selected SS_s require at least three SS_s to generate the MS. Also, depending on RN_{sp}, the AS specifies that the user's role is indirect and uses the Shamir scheme with ECDSA Sig to verify the original MS and then evaluate the request by PDP2. Using Shamir's scheme with XACML adds the property of authenticity as an indirect user cannot access data with the same SS_s. This operation enables PAX to secure the privacy of patient data and protect patient data from internal and external attacks. When an indirect user wants access to medical records, he/she does not know whether the SS_s used to generate the MS belongs to any specific HC providers.

6.3.4 PAX Authorisation Protocols

In this section, we will provide, in detail, PAX's protocols framework to authorise direct and indirect users:

- **Authorisation Protocols for Direct Subjects and Objects**
  To run through the authorisation process for direct users of PAX, the security techniques mentioned in Section 3.5 will be the basis for building the PAX authorisation system. In this section, we will explain the protocols for authorising direct users such as doctors and patients to access medical records (EHR).
  - **Prerequisite procedures**
    There is a set of steps that must be taken before authorisation can begin:
    1. Create two datasets (attributes, pseudonym) on the AS. If datasets are established, the processes are to add new users or delete direct users
    2. Create policies (dataset 3) for all direct users based on anonymity and pseudonym
    3. Storage of medical records (dataset 4) for patients in the DS's repository (after collecting them from patients using wireless medical devices, this process requires security mechanisms, but the process of storing medical records safely detailed in Chapter 4). We assume that patient data is located on the DS.
  - **Authorisation protocols**
    The following protocols detail how the direct user is associated with the EHR in DS. Figure 6.5 depicts the general authorisation process, while Figures 6.6, 6.7, 6.8, and 6.9 show the authorisation protocols of direct users with PAX entities.
    1. First protocol as shown in Figure 6.6:
       * PAX user enters the subject ID (S_{ID}), object ID (O_{ID}), subject role (S_R) and object role (O_R) to the C_i application. C_i

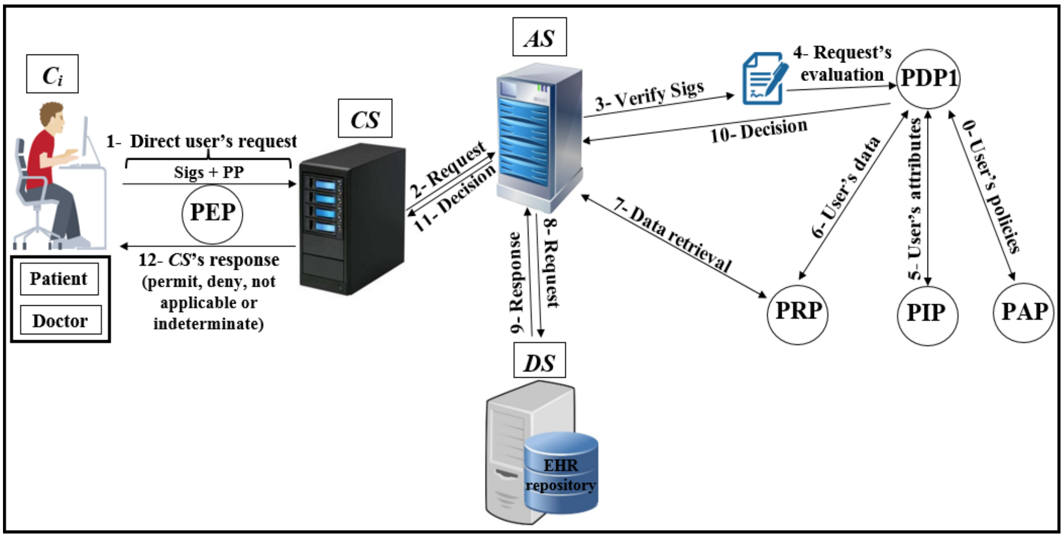


Figure 6.5: Authorisation of direct users

replaces S_{ID}, O_{ID}, S_R and O_R with CN_{sp}, CN_{op}, UR_{sp} and UR_{op} respectively. Next, internal pseudonyms are replaced with UN_{sp}, UN_{op}, RN_{sp} RN_{op} respectively. Then, C_i generates random nonces (N_C and SN_C) and new timestamp (TS_{C_i}). SN_C is a random secret between C_i and the CS. C_i computes 4 Sigs (C_iS_1, C_iS_2, C_iS_3 and C_iS_4). C_iS_1 and C_iS_2 is used to ensure the legitimacy of C_i in the CS. C_iS_3 is used to protect SN_C between C_i and CS. C_iS_4 is used to validate C_i in both the AS and DS (depending on RN_{op_h} and UN_{op_h}). C_i hides all Sigs such as C_iS_1 temporary ($C_iS_{1_{tm}}$) and PP such as $TS_{C_{tm}}$ and $SN_{C_{tm}}$. At this point, C_i sends XACML's request to the CS including the subject's information ($C_iS_{1_{tm}}||RN_{sp_{tm}}||UN_{sp_{tm}}||N_C||TS_{C_{tm}}||$ $SN_{C_{tm}}$) and object's information ($C_iS_{2_{tm}}||$ $RN_{op_{tm}}||UN_{op_{tm}}||N_C||C_iS_{4_{tm}}$)

* CS receives XACML's request from C_i, cuts Sigs and PP from access-subject ($C_iS_{1_{tm}}$, $RN_{sp_{tm}}$, $UN_{sp_{tm}}$, N_C, $TS_{C_{tm}}$ and $SN_{C_{tm}}$) and resource ($C_iS_{2_{tm}}$, $RN_{op_{tm}}$, $UN_{op_{tm}}$, N_C and $C_iS_{4_{tm}}$). Then, the CS extracts RN_{sp_l}, UN_{sp_l}, RN_{op_l} and UN_{op_l} from receiving parameters (such as $RN_{sp_{tm}}$). UN_{sp_l} and UN_{op_l} are used to retrieve UN_{sp_m} and UN_{op_m} from the datasets. the CS extracts C_iS_4, SN_C, TS_{C_i} and checks timestamp. Then, the CS computes Sigs (CSS_1, CSS_2 and CSS_3), and uses CSS_1 to extract original C_iS_1 and C_iS_2. After that, the CS checks CSS_2=C_iS_1 and CSS_3=C_iS_2. If the Sigs are not identical, the CS cancels the connection; otherwise, it moves to the next protocol.

C_i

CS

C_i:

Enters S_{ID}, O_{ID}, S_R and O_R
Replaces S_{ID} with CN_{sp}, O_{ID} with CN_{op}
S_R with UR_{sp} and O_R with UR_{op}
Replaces CN_{sp}, CN_{op}, UR_{sp} and UR_{op} with
UN_{sp}, UN_{op}, RN_{sp} and RN_{op}
$SP{=}RN_{sp}||UN_{sp}$, $OP{=}RN_{op}||UN_{op}$
$UN_{sp} = UN_{sp_l}||UN_{sp_m}||UN_{sp_h}$
$UN_{op} = UN_{op_l}||UN_{op_m}||UN_{op_h}$
$RN_{sp} = RN_{sp_l}||RN_{sp_m}||RN_{sp_h}$
$RN_{op} = RN_{op_l}||RN_{op_m}||RN_{op_h}$
Generates new N_C, SN_C and TS_{C_i}
$C_iS_1 = ECDSA(RN_{sp_m}||UN_{sp_m}||N_C||TS_{C_i})$
$C_iS_2 = ECDSA(RN_{op_m}||UN_{op_m}||N_C||TS_{C_i})$
$C_iS_3 = ECDSA(SN_C)$
$C_iS_4 = ECDSA(RN_{op_h}||UN_{op_h}||TS_{C_i})$
$C_iS_{1_{tm}} = C_iS_1 \oplus C_iS_3$, $C_iS_{2_{tm}} = C_iS_2 \oplus C_iS_3$
$TS_{C_{tm}} = TS_{C_i} \oplus SN_C \oplus RN_{sp_m} \oplus UN_{sp_m}$
$SN_{C_{tm}} = UN_{sp_m} \oplus UN_{op_m} \oplus SN_C \oplus C_iS_4 \oplus C_iS_{1_{tm}}$
$C_iS_{4_{tm}} = C_iS_4 \oplus UN_{sp_m} \oplus UN_{op_m} \oplus C_iS_{2_{tm}}$
$RN_{sp_{tm}} = RN_{sp_l} \oplus C_iS_{1_{tm}} \oplus SN_{C_{tm}}$
$RN_{op_{tm}} = RN_{op_l} \oplus C_iS_{2_{tm}} \oplus SN_{C_{tm}}$
$UN_{sp_{tm}} = UN_{sp_l} \oplus C_iS_{1_{tm}} \oplus SN_{C_{tm}}$
$UN_{op_{tm}} = UN_{op_l} \oplus C_iS_{2_{tm}} \oplus SN_{C_{tm}}$
Request$=(C_iS_{1_{tm}}||RN_{sp_{tm}}||UN_{sp_{tm}}||N_C||TS_{C_{tm}}||$
$SN_{C_{tm}}$,$C_iS_{2_{tm}}||RN_{op_{tm}}||UN_{op_{tm}}||N_C||$
$C_iS_{4_{tm}})$

Sends XACML's request $\longrightarrow$

CS:

Receive XACML's request
From access-subject:
Cuts $C_iS_{1_{tm}}$, $RN_{sp_{tm}}$, $UN_{sp_{tm}}$, N_C, $TS_{C_{tm}}$ and $SN_{C_{tm}}$
From resource:
Cuts $C_iS_{2_{tm}}$, $RN_{op_{tm}}$, $UN_{op_{tm}}$, N_C and $C_iS_{4_{tm}}$
Extracts $RN_{sp_l} = RN_{sp_{tm}} \oplus C_iS_{1_{tm}} \oplus SN_{C_{tm}}$
Similarly, extracts RN_{op_l}, UN_{sp_l} and UN_{op_l}
Retrieves UN_{sp_m} and UN_{op_m} from datasets depending on
UN_{sp_l} and UN_{op_l}
$C_iS_4 = C_iS_{4_{tm}} \oplus UN_{sp_m} \oplus UN_{op_m} \oplus C_iS_{2_{tm}}$
$SN_C = UN_{sp_m} \oplus UN_{op_m} \oplus SN_{C_{tm}} \oplus C_iS_4 \oplus C_iS_{1_{tm}}$
$TS_{C_i} = TS_{C_{tm}} \oplus SN_C \oplus RN_{sp_m} \oplus UN_{sp_m}$
Checks $TS_{CS} - TS_{C_i} \leq \Delta T$, $CSS_1 = ECDSA(SN_C)$
$C_iS_1 = C_iS_{1_{tm}} \oplus CSS_1$, $C_iS_2 = C_iS_{2_{tm}} \oplus CSS_1$
$CSS_2 = ECDSA(RN_{sp_m}||UN_{sp_m}||N_C||TS_{C_i})$
$CSS_3 = ECDSA(RN_{op_m}||UN_{op_m}||N_C||TS_{C_i})$
Checks $CSS_2 = C_iS_1$ and $CSS_3 = C_iS_2$

Figure 6.6: Protocol of PAX model between C_i and CS

2. Second protocol as shown in Figure 6.7:

   * The CS generates random secret (SN_{CS}) and new timestamp (TS_{CS}) between the CS and AS. Then, the CS computes the secret signature (CSS_4) to protect SN_{CS}. Also, the CS hides C_i's parameters such as N_C and TS_{C_i} to use them with validation operations in the AS and DS. In addition, all Sigs (such as $CSS_{2_{tm}}$) and PP (such as N_{CS} and $TS_{CS_{tm}}$) are anonymously hidden by the CS. At this point, the CS sends XACML's request to the AS

   * The AS receives the request, cuts Sigs and PP. After that, the AS extracts the original parameters (such as C_iS_4 and TS_{CS}) and checks the timestamp. The AS computes ASS_1 (to extract CSS_2 and CSS_3) and computes ASS_2 and ASS_3 (to check $ASS_2{=}CSS_2$ and $ASS_3{=}CSS_3$). The AS retrieves RN_{op_h} and UN_{op_h} from the dataset (depending on RN_{op_m} and UN_{op_m}) and computes ASS_4 to ensure C_i request is legitimate after checks $ASS_4 = C_iS_4$. The AS uses the parts of external pseudonyms to specify UR_{sp},

CS

AS

Creates new XACML's request

Generates new SN_{CS} and TS_{CS}

$CSS_4 = ECDSA(SN_{CS})$

$CSS2_{tm} = CSS_2 \oplus CSS_4$, $CSS3_{tm} = CSS_3 \oplus CSS_4$

$N_{CS} = N_C \oplus TS_{C_i} \oplus TS_{CS} \oplus SN_{CS}$

$TS_{C_{tm}} = TS_{C_i} \oplus SN_{CS} \oplus RN_{sp_m} \oplus UN_{sp_m}$

$TS_{CS_{tm}} = TS_{CS} \oplus SN_{CS} \oplus RN_{op_m} \oplus UN_{op_m}$

$SN_{CS_{tm}} = UN_{sp_m} \oplus UN_{op_m} \oplus SN_{CS} \oplus C_iS_4 \oplus CSS2_{tm}$

$C_iS4_{tm} = C_iS_4 \oplus UN_{sp_m} \oplus UN_{op_m} \oplus CSS3_{tm}$

$RN_{sp_{tm}} = RN_{sp_l} \oplus CSS2_{tm} \oplus SN_{CS_{tm}}$

$RN_{op_{tm}} = RN_{op_l} \oplus CSS3_{tm} \oplus SN_{CS_{tm}}$

$UN_{sp_{tm}} = UN_{sp_l} \oplus CSS2_{tm} \oplus SN_{CS_{tm}}$

$UN_{op_{tm}} = UN_{op_l} \oplus CSS3_{tm} \oplus SN_{CS_{tm}}$

Request=$(CSS2_{tm}||RN_{sp_{tm}}||UN_{sp_{tm}}||N_{CS}||TS_{C_{tm}}||TS_{CS_{tm}}||SN_{CS_{tm}}, CSS3_{tm}||RN_{op_{tm}}||UN_{op_{tm}}||C_iS4_{tm})$

Sends XACML's request $\longrightarrow$

Receive XACML's request

Similarly for CS:

Cuts security parameters

Extracts RN_{sp_l}, RN_{op_l}, UN_{sp_l} and UN_{op_l}

Retrieves UN_{sp_m} and UN_{op_m}

Extracts C_iS_4, SN_{CS}, TS_{C_i}, TS_{CS} and N_C, checks TS_{AS}

$ASS_1 = ECDSA(SN_{CS})$

$CSS_2 = CSS2_{tm} \oplus ASS_1$, $CSS_3 = C_iS3_{tm} \oplus ASS_1$

$ASS_2 = ECDSA(RN_{sp_m}||UN_{sp_m}||N_C||TS_{C_i})$

$ASS_3 = ECDSA(RN_{op_m}||UN_{op_m}||N_C||TS_{C_i})$

Checks $ASS_2 = CSS_2$ and $ASS_3 = CSS_3$

Retrieves RN_{op_h} and UN_{op_h}

$ASS_4 = ECDSA(RN_{op_h}||UN_{op_h}||TS_{C_i})$, $ASS_4 = C_iS_4$

$SP = RN_{sp}||UN_{sp}$, $OP = RN_{op}||UN_{op}$

Specifies UR_{sp}, UR_{op}, CN_{sp} and CN_{sp} depending on RN_{sp} and RN_{op}, UN_{sp} and UN_{sp}

Retrieves S_{sp} and S_{op} depending on SP and OP

Uses PDP1 to evaluate request (S_{sp}, S_{op})

Specifies policy depend on policy's ID (SP)

Checks attributes (S_{ID},O_{ID},S_R,O_R) in PIP, applies policy in PAP

If decision ="permit", then

If UR_{sp}= direct user, then

Sends data retrieval request by PRP to DS

If UR_{sp}= indirect user, then

Sends request with Shamir scheme to C_i by CS

Figure 6.7: Protocol of PAX model between CS and AS

UR_{op}, CN_{sp} and CN_{sp}. AS retrieves Sigs of SP and OP (S_{sp} and S_{op}) depending on the internal SP and OP. The AS uses PDP1 engine to evaluate XACML's request after adding S_{sp} and S_{op} to that request. The AS specifies the user's policy in PAP and checks the user's attributes in PIP. PDP1 applies policy to get a decision (permit, deny, not applicable and indeterminate). If decision="permit", the AS uses UR_{sp} to specify the user's role (direct/indirect). If UR_{sp}=direct, the AS sends the data retrieval request by PRP to the DS; if UR_{sp}=indirect, the AS sends the Shamir request that contain at least two SS_s to ensure legitimate indirect users. Otherwise the AS sends a reject response to C_i by CS.

3. Third protocol as shown in Figure 6.8:

  * Similarly, the AS generates random secret (SN_{AS}) and timestamp (TS_{AS}) between the AS and DS. The AS computes ASS_5 to protect secret (SN_{AS}) between the AS and DS. Additionally, the AS computes ASS_6 to ensure legitimate PP

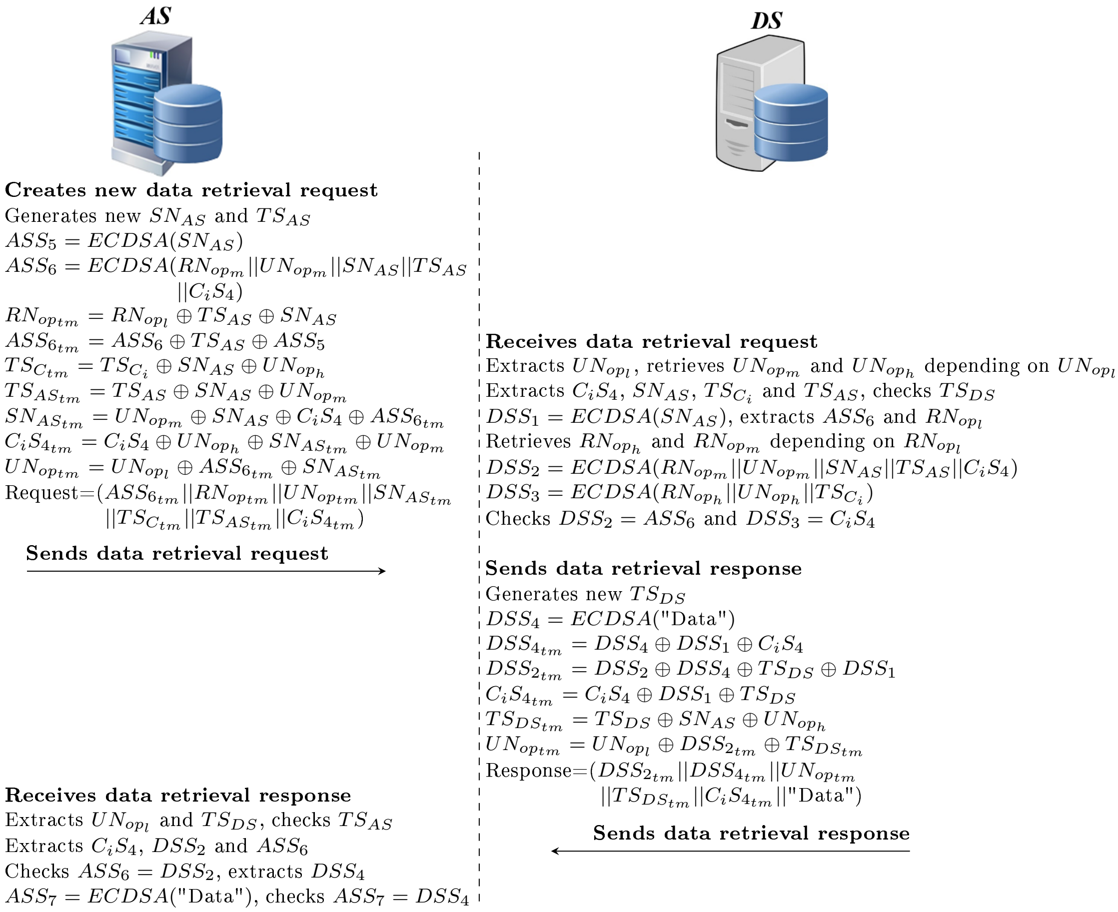


Figure 6.8: Protocol of PAX model between AS and DS

(RN_{op_m} and UN_{op_m}) in the DS. All Sigs (such as $ASS_{6_{tm}}$) and PP (such as $TS_{AS_{tm}}$ and $SN_{AS_{tm}}$) are anonymously hidden by the AS. Then, the AS sends XACML's request to the DS

* The DS receives the request, cuts Sigs and PP. After that, the DS extracts original parameters (such as C_iS_4 and SN_{AS}) and checks the timestamp. The DS computes DSS_1 (to extract ASS_6) and retrieves RN_{op_h} and RN_{op_m} depending on RN_{op_l}. Then, the DS computes DSS_2 and DSS_3 to check $DSS_2 = ASS_6$ and $DSS_3 = C_iS_4$. If the AS's parameters are validated in the DS correctly, the DS computes a timestamp (TS_{DS}) and signs the patient's data (DSS_4). All Sigs (such as $DSS_{4_{tm}}$) and PP (such as $TS_{DS_{tm}}$) are anonymously hidden by the DS. At this point, the DS sends the response to the AS
* The AS receives the response, extracts PP (such as TS_{DS}) and checks timestamp. The AS tests the Sigs checking (such as $ASS_6 = DSS_2$). Then, the AS computes data signature (ASS_7) to check data integrity by $ASS_7 = DSS_4$.

4. Fourth protocol as shown in Figure 6.9:
   * The AS prepares the response to the CS by generating a new timestamp (TS_{AS}), hides data signature (ASS_7) with ASS_2, ASS_3, C_iS_4 and secret signature (ASS_1). The AS hides PP and sends the response that contains the decision and patient's data to the CS
   * The CS receives the response and extracts Sigs and PP. The CS computes data signature (DSS_5) to check data integrity ($CSS_5 = ASS_7$). Then, the CS checks other Sigs (CSS_2, CSS_3 and CSS_4) with received Sigs (ASS_2, ASS_3 and C_iS_4) to ensure legitimacy of the AS. The CS prepares the response to C_i by generating a new timestamp and hides data signature (CSS_5) with CSS_2, CSS_3, C_iS_4 and secret signature (CSS_1). The CS sends the response to C_i
   * C_i receives the response, extracts PP and checks timestamp. C_i computes data signature (C_iS_5) to check data integrity by $C_iS_5 = CSS_5$. Then, C_i extracts signatures (CSS_2, CSS_3, CSS_1 and C_iS_4) and checks them with original signatures (C_iS_1, C_iS_2, C_iS_3 and C_iS_4) respectively. C_i uses CSS_2, CSS_3 and CSS_1 (secret signature between C_i and CS) to check legitimacy of the CS while using C_iS_4 to check legitimacy of the AS and DS. If all Sigs are validated, namely, authorised C_i received securely correct data.

- **Authorisation Protocols for Indirect Subjects and Objects**

  Indirect user authorisation is an important process to secure sensitive patient data in the EHR stored in DS. PAX offers additional procedures to prevent the abuse of indirect user privileges.

  – **Prerequisite procedures**

  There is a set of steps that must be performed before authorisations are applied:

  1. Steps from 1 to 3 are similar to those for direct users
  2. The Shamir scheme is used to generate the SS_s from the MS for the number of users, each C_i has unique SS the same length as the MS, and authorised with two policies for each indirect user on the AS. The policy evaluation process is also done with two, PDP1 and PDP2, evaluation engines. The use of the two evaluation engines is very important in separating direct and indirect users and increasing

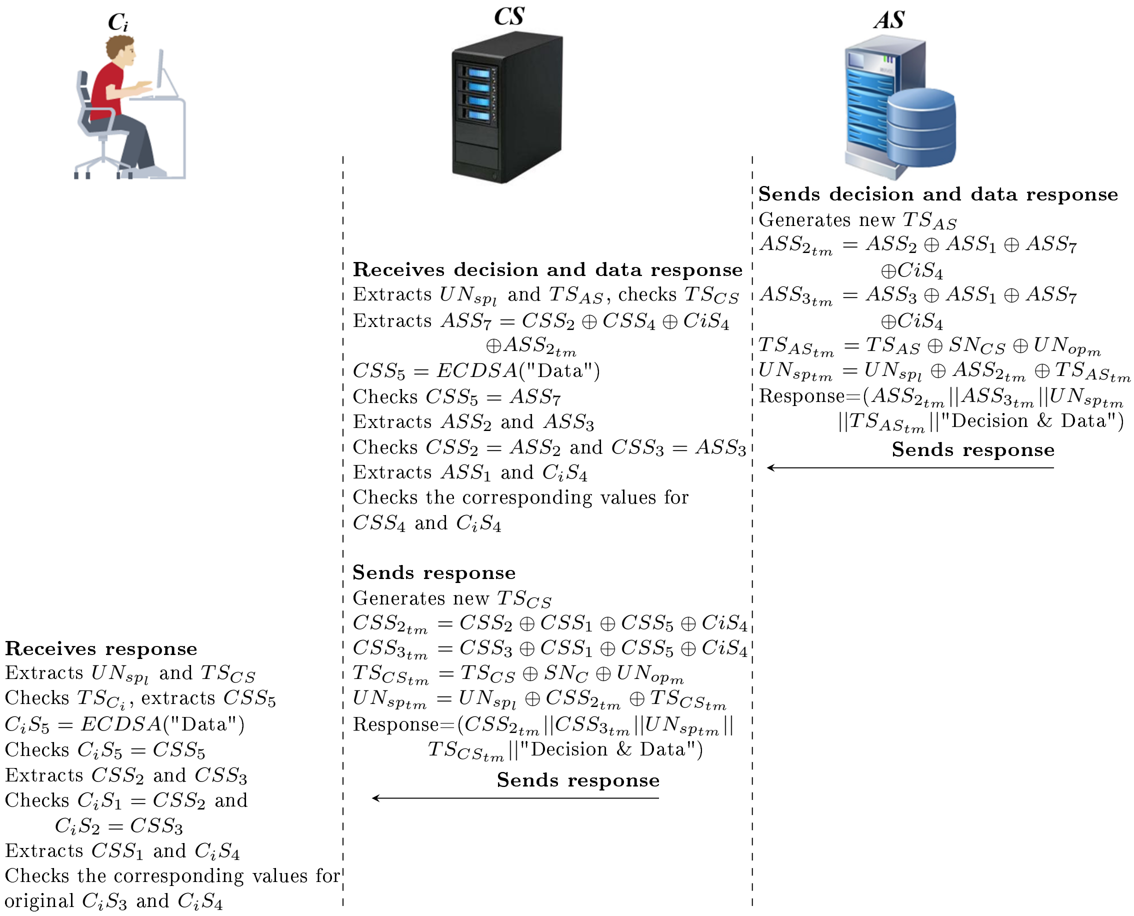


Figure 6.9: Protocol of PAX model between AS, CS and C_i

security in the privacy of medical records

3. The PAX authorisation system identifies certain medical records (the patients' history at a given time such as a year or more ago) for indirect users who can access them, as shown in Figure 7.28 (researcher case).

– **Authorisation protocols**

The following protocols detail how the indirect user obtains medical records in PAX. Figure 6.10 illustrates generally the authorisation of indirect users, while Figures 6.6, 6.7,6.11, 6.8, and 6.9 show the authorisation protocols of indirect users in PAX:

1. The steps of the first and second protocols are similar to the ones of the direct users authorisation

2. Third protocol as shown in Figure 6.11:

   * The AS computed the MS previously by signing all users' signatures. Then, the AS computes the Shamir scheme to generate SS_s with the same number of users (each C_i has one unique SS). In PAX, C_i needs at least three SS_s to generate

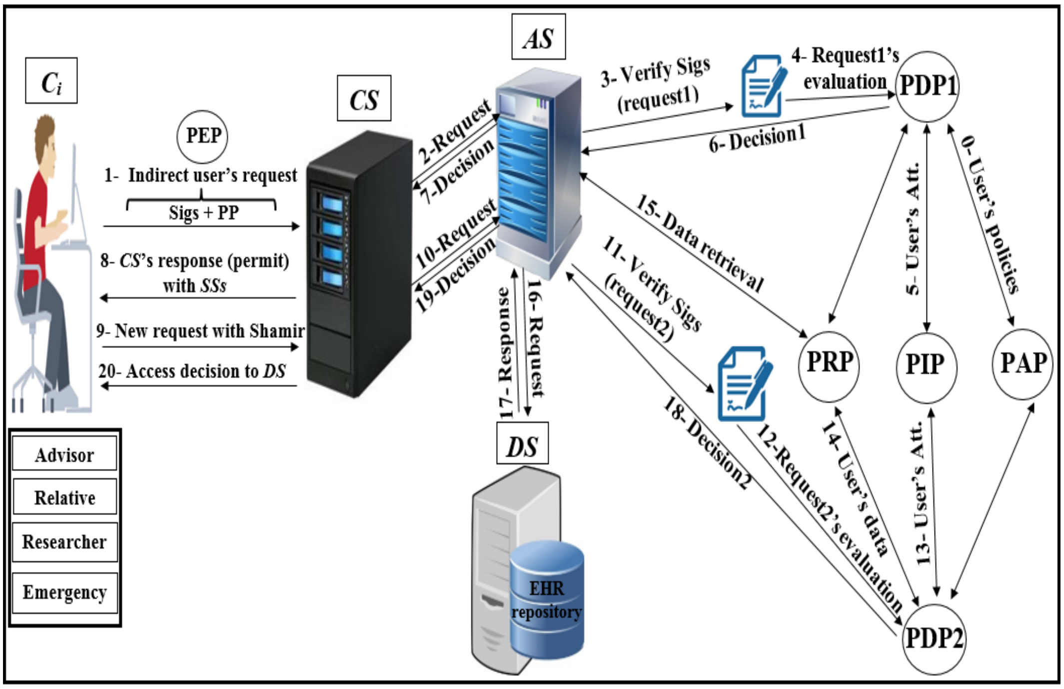


Figure 6.10: Authorisation of indirect users

the original MS. In this protocol, the AS generates a new timestamp and retrieves at least two SS_s. After that, the AS hides SS_s with ASS_2, C_iS_4, S_{sp} and secret signature (ASS_1) as well as parameters (such as $TS_{AS_{tm}}$ and $UN_{sp_{tm}}$) are anonymously hidden. At this point, the AS sends request to the CS

* The CS receives the request, extracts PP and checks the timestamp. Then, the CS removes the secret signature (CSS_4) and adds the secret signature (CSS_1) in $CSS_{2_{tm}}$. The CS generates a new timestamp (TS_{CS}), hides PP and sends the request to C_i

* C_i receives Shamir's request, extracts PP and checks the timestamp. Then, C_i computes C_iS_6 to extract SS_s and retrieves his SS. At the moment, C_i can generate the MS from Shamir (C_i's $SS||SS_s$), hides the MS with C_iS_6 and C_iS_3, generates timestamp and hides PP. At this point, C_i sends the response to the CS

* The CS receives the response, extracts PP and checks timestamp. Also, the CS removes CSS_1 and adds CSS_4 in $C_iS_{6_{tm}}$. The CS generates a new timestamp, hides PP and sends the response to the AS

* The AS receives Shamir response, extracts PP and checks timestamp. Then, the AS extracts the received the MS and

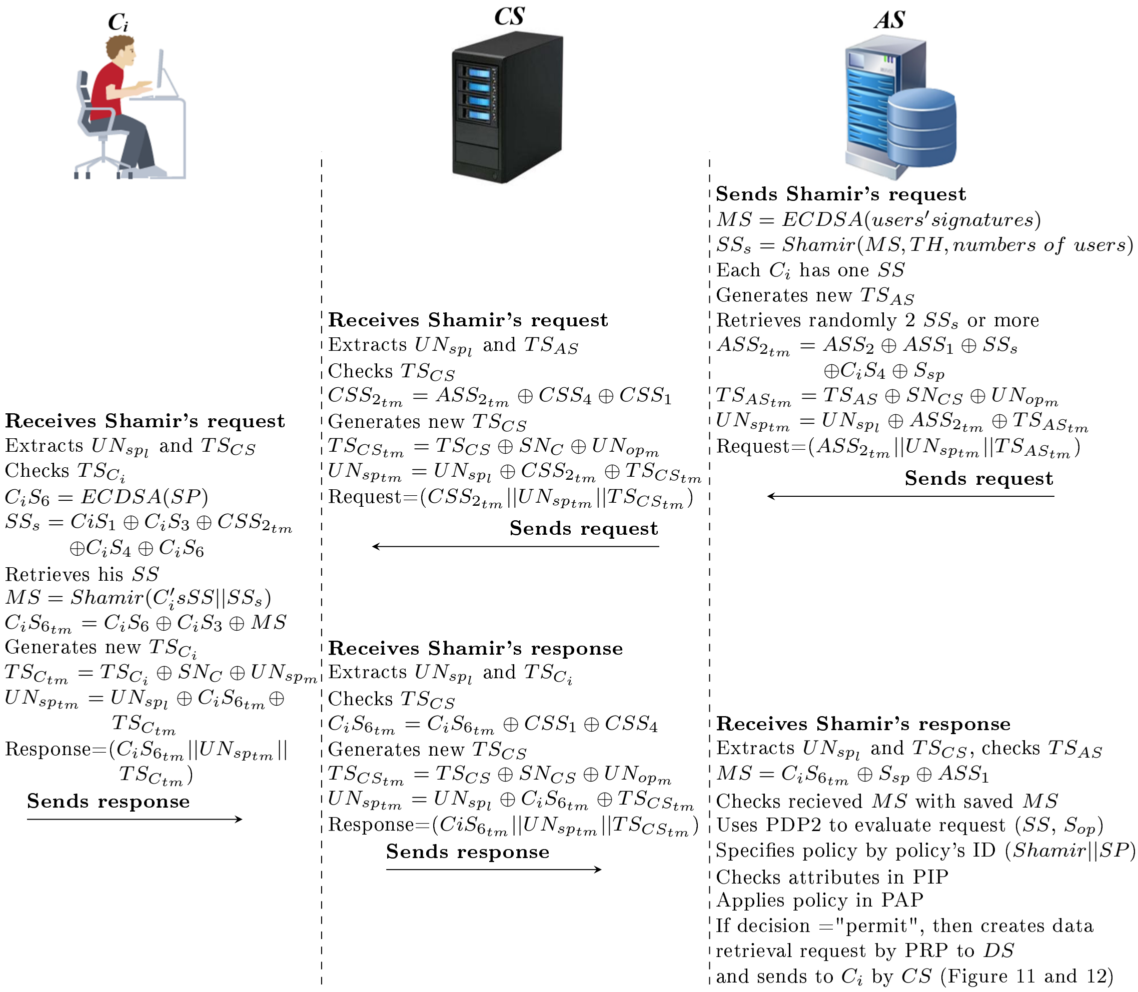


Figure 6.11: Protocol of PAX model for indirect users

checks it with the saved original the MS. After that, the AS retrieves C_i's SS depending on $S_{sp}(UR_{sp}||\ CN_{sp})$ and assigns request (SS,S_{op}) to PDP2. The AS specifies policy depending on policy's ID (Shamir||SP), checks attributes in PIP and PDP2 applies policy in PAP to produce the decision. If the decision is "permit", the AS creates a data retrieval request by PRP to the DS; otherwise the AS sends reject response to C_i by the CS.

3. The fourth and fifth protocols are similar to the third and fourth, respectively in the direct user authorisation. The DS sends the response to the C_i by the AS and CS. If C_i is advisor, relative, or emergency doctor, C_i will receive the specific patient's data; otherwise if C_i is researcher doctor, C_i will receive a set of medical records.

6.4 Summary of the Chapter

In this chapter, we have provided details of the requirements and design goals for building a new authorisation scheme. Then, we put forward an AC network model for the proposed HC application. The objective of the AC model is to authorise a differentiated level of access control to the EHR repository, for the purpose of achieving efficiency as well as preserving patient privacy. Finally, the authorisation scheme methodology has been explained in detail to authorise users in the HC application by having given a set of protocols and procedures for authorising HC users with their balanced and deserved authority.

Chapter 7: Verification of Protocols for the Security and Performance

Having developed an HC application with security features and high efficiency, along with the protocols and algorithms for protecting user authentication and privacy for the patients' health/medical records; as well the HC professionals differentiated access control to the HC patients records, it is time to verify those protocols and prove that the proposed HC application has achieved the objectives. In this chapter, we start with an introduction and description of the protocol testing tool, followed by how to use the tool to verify our developed REISCH, RAMHU and PAX.

7.1 Security Testing Tool

In this section, we will provide security testing tool details such as the automated validation of Internet security protocols and applications (AVISPA) used in our project.

7.1.1 AVISPA

For all of the protocols designed so far, it is necessary to verify their ability to resist various threats/attacks. In this section, we employ the widely used AVISPA tool, and explain how to use it to check the protocols we have designed. AVISPA is a formal tool for analysing security schemes and is applied by researchers to evaluate recent security protocols (Gupta et al. 2018, Babu & Padmanabhan 2018, Xu et al. 2018, Dong et al. 2018). This tool depends on Dolev-Yao (dy) to verify security protocols. For more details of AVISPA , see Appendix D.

7.2 Analysis of Storage Scheme

In this section, we will discuss security and performance analyses for the REISCH scheme and compare it with existing schemes. Analyses demonstrate that REISCH is efficient for use in patient data collection within the HWSN environment in terms of security and performance.

7.2.1 Possible Attacks on REISCH Scheme

In this section, we will examine the REISCH scheme theoretically with a set of threats mentioned in the threat model. We will provide a theoretical analysis of REISCH resistance to known attacks as follows:

- **Proposition 1 – MITM and replay attacks**
  The intruder (I) cannot achieve MITM and replay attacks against the REISCH scheme.
  Proof
  An intruder tries to change or delete part of data/information when transferred between the network's entities. This situation is not possible because REISCH applies the ECDSA algorithm to sign data as well as some information such as SN_{Pseud}. Additionally, an intruder cannot replay a message late due to the REISCH's entities use of timestamps such as SN_{TS} and CH_{TS}. Consequently, REISCH resists MITM and replay attacks successfully.

- **Proposition 2 – DoS attacks**
  I will not benefit from using DoS attacks against REISCH servers.
  Proof
  An intruder applies a DoS attack to destroy the availability of service in servers such as the LS and CS. The servers in REISCH initially check lightweight parameters such as $SigLsE_i$ in LS and CS_{Pseud_o} in CS before completion of the authentication process. Moreover, these parameters change randomly in the communication process between entities. This procedure allows servers to check small parameters and prevent DoS duplicate messages. Therefore, REISCH withstands DoS threats.

- **Proposition 3 – Localisation attacks**
  I cannot implement localisation attacks to deceive sensors and servers.
  Proof
  An intruder tries to use the Sybil attack by using many legitimate SN's IDs with fake data. Due to SN_i waits for random $SigLsE_i$ from the LS per round, namely, that intruder cannot deceive the LS with fake data. Also,

an intruder can employ a wormhole attack by using many SN_i to camouflage communications between network entities. Each SN_i implicitly sends SN_{SL_i} to the LS and Dif to CH_i as well as a timestamp. These parameters prevent counterfeit communications. Furthermore, if an intruder aims to apply a sinkhole attack using a node as a sink to attract all patient data from SN_i, it cannot apply to REISCH because the LS sends an unique LS_{OTP_i} including $SigLS_i(SN_{Pseud_i})$ for all SN_i. That intruder fails to detect $SigLS_i$ and SN_{Pseud_i}. Hence, REISCH strongly overcomes localisation attacks.

- **Proposition 4 – Repository attack**
  I does not have the ability to penetrate the EMR repository.
  Proof
  Assume that an intruder can penetrate datasets in LS. First, LS does not contain real patient information (real information such as the name is stored in the AS). When the intruder gets this data, he/she cannot disclose that it belongs to a particular patient. Second, the LS 'datasets are very difficult to penetrate. Furthermore, the LS contains partial data for patients because the total data and patients' history are transferred to the DS by the CS periodically. Thereupon, REISCH resists the EMR repository attack.

- **Proposition 5 – Eavesdropping attacks**
  The external intruder (EI) will not get any benefit from eavesdropping in the disclosure of personal information of collected data.
  Proof
  When an intruder eavesdrops and gets some of the messages transferred between SN_i, CH_i, LS and CS. This intruder will not benefit from these messages trapped because these messages contain no real information. Furthermore, the secret parameters are completely hidden. Thus, REISCH prevents eavesdropping attacks from revealing patient information.

- **Proposition 6 – Replication attack**
  I cannot perform node replication attacks.
  Proof
  An intruder applies node replication attack by using more than one SN_i with the same legitimate ID. In REISCH, we suppose that all SN_i are inside a specific area in the hospital or clinic. Therefore, any SN_i outside this area extremely finds it difficult to send messages from fake SN_i with the same legitimate ID. In addition, LS waits SN_m by CH_i at the same number of SN_i and the LS remove replicated SN_m or $SigSnT_3$. Also, when SN_i dies, LS records this situation in the dataset to prevent replication risks. In a result, REISCH effectively resists replication attacks.

- **Proposition 7 – Collision and preimage attacks**
  I does not have the ability to perform collision and preimage attacks against REISCH algorithms.
  Proof
  An intruder tries to implement a collision (the generation of two different messages that produce the same MD =h(m) = h(m')), preimage (the generation of a message that produces the same existing MD value as h(m) = MD) and second preimage (the generation of a different message from the received message and produce the same existing MD value) attacks when messages and signatures are transferred between REISCH's entities. These attacks cannot be implemented on REISCH protocols because our protocols use the BLAKE2bp hash instead of SHA1 which resists these attacks. Consequently, REISCH successfully prevents collision and preimage attacks.

1. **Experimental Security Analysis**
   Now it is the time to use the AVISPA to conduct a security analysis on the REISCH scheme.

   - **REISCH Scheme with AVISPA**
     As we have known that REISCH includes four essential roles: localServer (LS)), sensori (SN_i), clusterHeadi (CH_i) and centralServer (CS), as well as supporting roles: session and environment. Also, there are three sections to complete communication properly and securely: transition, composition and goal specification. The transition section is used in the essential roles to maintain the correct communication sequence. The composition section is used in the supporting roles to connect essential roles in specific sessions. The goal specification section includes security goals such as secrecy and authentication. Secrecy means known secrets only for specific entities while authentication depends on witness (freshness claim) and request (validation) processes to perform strong authentication.
     Also, our scheme uses parameters such as RCV (receiving process), SND (sending process), _inv (private key), dy (communication channel by Dolev-Yao model) and *intruder_knowledge* (known information for an intruder). We assume the the intruder uses the public key (k_i) and knows the public keys for REISCH's entities ($kSNpu$, $kCHpu$ and kLS). Figure 7.1 shows the REISCH's framework in AVISPA.

     As shown in Figure 7.4, the LS receives the start signal. Then, the LS generates and sends new $LSotpi$ for all sensors (SNi and CHi). $LSotpi$

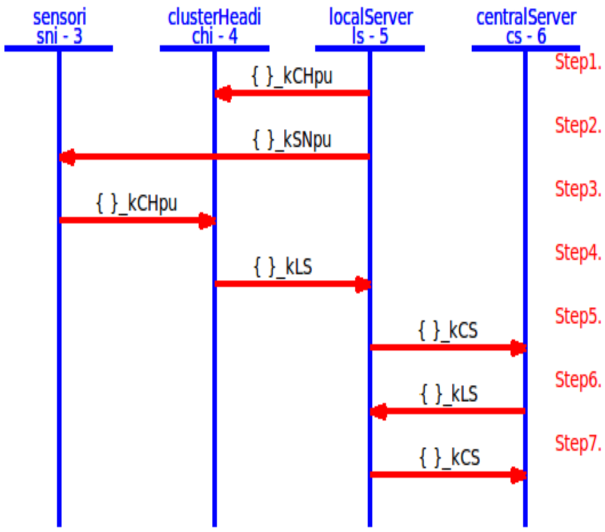


Figure 7.1: REISCH's framework in AVISPA

includes new $SigLsEi$ and pseudonym signature. Both Figure 7.2 and Figure 7.3 show that SNi and CHi receive $LSotp$ from the LS. Furthermore, SNi and CHi use freshness nonces, timestamp and signature to support reliable security. For instance, SNi uses $SigLsEi$, $SigSnEi$, $SNts$ and $SigSN$ to achieve security processes with CHi and the LS.

SNi collects data and uses one ECDSA signature with XoR operations to protect collected data and sends it to CHi. At this stage, CHi aggregates data and adds security parameters. CHi sends aggregation data to the LS. After that, the LS uses $LSotp$, $SigLS5$ and $LSrn$ to connect with the CS securely as shown in Figure 7.4. Figure 7.5 shows the CS with the storage process. The CS receives $LSotp$ and uses ECDSA($CsT2$), $CsT1$, $CsT2$ and $CSrn$ to secure communication with the LS. Figure 7.6 shows session and environment roles as well as security goals (secrecy and authentication).

REISCH applies seven secrecy and seven authentication goals. For instance, Sec1 represents secrets between SNi and the LS such as $SigSN$, $SigLsEi$ and $SigSnEi$. Also, the authentication goal, such as auth4 proves freshness between CHi and the LS such as $CHts2$, $CHotp$, $CHrni$ and $SNrni$. Additionally, the environment role includes many attacks (replay, MITM and impersonating) to test the security level in the REISCH scheme.

```
role sensori(SNi,CHi,LS:agent, KSNpu,KCHpu,KLSpu:public_key, ECDSA:hash_func, SNpseud,SNsl,
             Data:text,SND,RCV:channel(dy))
played_by SNi def=
local
     State:nat,
     SNrni,SigLsEi,SigSnEi,SNparity:text,
     MAXv:message,SigSN,SigSnT1,SigSnT2,SigSnT3,SigSnT4,SigLSi:text,
     Dif,SNts,SNtst,SNotp,Date,Time,SS,SNp,LSotpi:text
const
     sec1,sec2,sec3,auth3:protocol_id
init
     State := 0
transition
% SNi receives(LSotpi) from LS
1.State=0
  /\RCV(SNi.LSotpi'.SNrni') =|> State':=1
  /\SigSnEi':=new() /\SigLSi':={ECDSA(SNpseud)}_inv(KSNpu)
  /\SigLsEi':=xor(LSotpi',SigLSi) /\SigSN':={ECDSA(Data.SNpseud)}_inv(KLSpu)
  /\SNparity':=SigSN'/\SigSnT1':=SNparity' /\SigSnT2':=xor(SigSnT1',SigSnEi')
  /\SNts':=new()/\SNotp':=new() /\SNtst':=xor(Date,xor(Time,xor(SNotp',xor(Dif))))
  /\SNp':={(SNtst'.SNotp'.SNrni'.SNpseud.SNsl)} /\SigSnT3':=xor(SNp',SigSnT2')
  /\SigSnT4':=xor(SigLsEi',SigSnEi')
  /\secret({SigSN',SigLsEi',SigSnEi'},sec1,{SNi,LS}) /\secret({SNpseud,SNsl},sec2,{SNi,LS})
  /\secret({MAXv,SNts'},sec3,{SNi,CHi})
% SNi sends(SNm) to CHi
/\SND(CHi.SigSnT3'.SigSnT4'.SNrni'.Dif.SNtst'.SS.Data) /\witness(SNi,CHi,auth3,{SNtst',SNrni',Dif})
end role
```

Figure 7.2: SN_i role of REISCH in HLPSL

```
role clusterHeadi(CHi,LS,SNi:agent, KCHpu,KLSpu,KSNpu:public_key, ECDSA:hash_func,CHpseud,CHsl:text
                 ,MAXv:message,SND,RCV:channel(dy))
played_by CHi def=
local
     State:nat,
     SNrns,SNrni,SigLsEi,SNparityi,SNts,LSotpi:text, SigCH,SigChT1,SigChT2,SigSnT3s,SigSnT4s,
     SigSnT3i,SigSnT4i,SigLSi,Difi,SNtst,CHts1,CHts2,SS,SSi,CHp,CHparity,CHotp,CHrni,Datai,
     Datas,CHa:text
const
     sec3,sec4,sec5,auth4:protocol_id
init
     State := 0
transition
% CHi receives(LSotpi) from LS
1.State=0
  /\RCV(CHi.LSotpi'.CHrni')=|> State':=1

% CHi receives from SNi
2.State=1/\RCV(CHi.SigSnT3i'.SigSnT4i'.SNrni'.Difi'.SNtst'.SSi'.Datai')=|>State':=2
  /\CHts1':=new() /\SigSnT3s':={(SigSnT3s'.SigSnT3i')}/\SigSnT4s':={(SigSnT4s.SigSnT4i')}
  /\SNrns':=SNrni'/\Datas':=Datai' /\SigCH':={ECDSA(SigSnT3s')}_inv(KLSpu)
  /\CHparity':=SigCH' /\SigChT1':=xor(CHparity',CHts1')
  /\SigLSi':={ECDSA(CHpseud)}_inv(KCHpu) /\SigLsEi':=xor(LSotpi,SigLSi')
  /\SigChT2':=xor(SigChT1',SigLsEi') /\CHts2':=new()/\CHotp':=new()
  /\secret({MAXv,SNts},sec3,{CHi,SNi}) /\secret({SigCH,SigLsEi,CHpseud,CHsl},sec4,{CHi,LS})
  /\secret({CHotp',CHts2'},sec5,{CHi,LS})
% CHi sends(CHm) to LS
   /\SND(LS.(xor((CHts2'.CHotp'.CHrni.CHpseud.CHsl),SigChT2)).CHrni.SS.(SigSnT3s.SigSnT4s
   .SNrns'.Datas')) /\witness(CHi,LS,auth4,{CHts2',CHotp',CHrni,SNrns'})
end role
```

Figure 7.3: CH_i role of REISCH in HLPSL

- **Simulation Results**

  The simulation results described by the AVISPA tool. We have applied AVISPA with OFMC and CL-AtSe backends. Both the

OFMC (Figure 7.7) and CL-AtSe (Figure 7.8) results show that the REISCH scheme is safe against passive and active attacks (as in the SUMMARY section). Furthermore, Figure 7.7 and Figure 7.8 shows analysis details about simulation reports such as the number of sessions, goals and statistical numbers. Also, the goals of authentication and secrecy in Figure 7.6 are applied to prevent the penetration of sensors' data/information in the network. These results prove that REISCH is reliable in combatting known attacks such as replay, MITM, and impersonating.

2. **Security Comparison**

Not only is REISCH secure against the various attacks mentioned in Section 7.2.1, it is also superior over the other schemes in terms of security (Table 7.1 shows a comparison of security features between our scheme and existing schemes).

For example, compared with the scheme in Fan & Gong (2012) that uses a small key ($F_{2^{163}}$) and is extremely vulnerable to attacks, REISCH uses a key with 256-bit that resists attacks (reputable organisation recommendations). REISCH also uses BLAKE2bp to get rid of hash attacks (collision and preimage) while the scheme in Kodali (2013) focused on SHA1 performance without attention to the collision/preimage threats. In addition, all security parameters in REISCH such as SN_i's location are completely hidden, while the scheme in Lavanya & Natarajan (2017*b*) transfers some information explicitly, such as ID (the elliptic curve parameters) in the registration and authentication phases. This allows intruders to distinguish a specific SN_i. Additionally, this scheme did not address the problem of hiding SN_i location.

Although the authors in Staudemeyer et al. (2018) addressed privacy to protect the SN_i parameters, their scheme did not use the signature camouflage or SN_{OTP} that are used in REISCH to support the privacy of data signing. This makes the privacy parameters in their scheme vulnerable to analysis and easy tracking. Furthermore, REISCH outperforms the scheme in Malathy et al. (2018) which did not use the signature aggregation scheme to support security and hide signatures. The scheme in Sharavanan et al. (2018) uses ECDSA to secure heterogeneous network environments. But their scheme gives medical evaluators privileges to modify the medical parameters in the monitoring environment, SN_i's locations and even creates keys that could be the cause of an internal attack. Moreover, some information sent from SN_i to the server can clearly leak to intruders. Fortunately, REISCH does not suffer from these problems.

```
role localServer(LS,CHi,SNi,CS:agent, KLSpu,KCHpu,KSNpu,KCSpu:public_key, ECDSA:hash_func ,SNpseudi
            ,SNsli:text,SND,RCV:channel(dy))
played_by LS def=
local
     State:nat,
     SNrni,SNrns,SigLsEi,SigSnEi,SNparityi,SNp:text,
     SigChT2,SigSnT3i,SigSnT4i,SigSnT3s,SigSnT4s,SigLsT1i,SigLsT2i,SigLS1i,SigLS2i,SigLSi:text,
     CHrni,CHparityi,CHotpi,CHsli,CHts2,SS,Datai,Datas:text,LSpseudo,LSpseudn,CSpseudo,
     CSpseudn:text, LSts2,LSts3,LSts4,LsT1,LsT2,LsT3,LSotpi,LSotpii:text,
     CHpseudi,LSotp,CSotp,CSts1,CsT1,CsT2,CSrn,LSrn,SigLS3,SigLS4,SigLS5:text
const
     sec1,sec2,sec4,sec5,sec6,sec7,auth1,auth2,auth4,auth5,auth6,auth7:protocol_id
init
     State := 0
transition
% Starting signal
1.State=0
  /\ RCV(start) =|>
% LS sends (LSotpi,LSotpii)to SNi & CHi
  State':=1/\ SNrni':=new()/\SigLsEi':=new()
  /\LSotpi':=xor({ECDSA(SNpseudi)}_inv(KSNpu),SigLsEi') /\witness(LS,SNi,auth1,{SigLsEi',SNrni'})
  /\SND(SNi.LSotpi'.SNrni') /\SNrni':=new()/\SigLsEi':=new()
  /\LSotpii':=xor({ECDSA(SNpseudi)}_inv(KCHpu),SigLsEi') /\witness(LS,CHi,auth2,{SigLsEi',SNrni'})
  /\SND(CHi.LSotpii'.SNrni')

2.State=1
% LS receives(CHm) from CHi
  /\RCV(LS.(xor((CHts2'.CHotpi'.CHrni'.CHpseudi'.CHsli'),SigChT2')).CHrni'.SS'.(SigSnT3s'.SigSnT4s'
  .SNrns'.Datas')) =|>State':=2
  /\SigLS1i':={ECDSA(SigSnT3s')}_inv(KLSpu) /\CHparityi':=SigLS1i'
  /\SigLsT1i':=xor(SigLS1i',SigLsEi)
  /\SigChT2':=xor((CHts2'.CHotpi'.CHrni'.CHpseudi'.CHsli'),SigLsT1i')
  /\SNrni':=SNpseudi/\Datai':=Datas' /\SigLS2i':={ECDSA(Datai.SNpseudi)}_inv(KLSpu)
  /\SNparityi':=SigLS2i' /\SigSnT3i':=SigSnT3s'/\SigSnT4i':=SigSnT4s'
  /\SigSnEi':=xor(SigSnT4i',SigLsEi) /\SigLsT2i':=xor(SNparityi',SigSnEi')
  /\SNp':=xor(SigLsT2i',SigSnT3i') /\request(LS,CHi,auth4,{CHts2,CHotpi',CHrni',SNrns'})
  /\secret({SigLS2i,SigLsEi,SigSnEi},sec1,{LS,SNi}) /\secret({SNpseudi,SNsli},sec2,{LS,SNi})
  /\secret({CHparityi',SigLsEi,CHpseudi,CHsli},sec4,{LS,CHi})
  /\secret({CHotpi',CHts2'},sec5,{LS,CHi}) /\LSpseudn':=new()/\LSts2':=new()
  /\LSotp':=xor(SigLSi,xor(LSpseudn',LSts2'))
% LS sends(LSotp) to CS
  /\SND(CS.LSotp'.SS) /\secret({LSpseudn,CSpseudn},sec6,{LS,CS})
  /\witness(LS,CS,auth5,{CSpseudo,LSpseudn',LSts2'})

3.State=2
% LS receives(CSm) from CS
/\ RCV(LS.{ECDSA(CsT2')}_inv(KLSpu).CsT1'.CsT2'.SS'.CSrn') =|>
  State':=3/\LSts3':=new() /\CSotp':=xor(CSts1,xor(CsT1',xor(LSpseudn,CSrn')))
  /\CSpseudn':=xor(CSts1,xor(CSotp',xor(CsT2',LSpseudn)))
  /\SigLS3':={ECDSA(xor(CSts1,xor(CSotp',xor(CSpseudn',LSpseudn))))}_inv(KLSpu)
  /\secret({SigLSi,CSotp'},sec7,{LS,CS}) /\request(LS,CS,auth6,{CSpseudn',LSpseudn,CSts1})
% Prepares message to send data with mutual authentication
 /\LSts4':=new()/\LSrn':=new() /\LsT1':=xor(LSts4',xor(LSpseudn,xor(LSrn',CSrn')))
 /\SigLS4':={ECDSA(LsT1')}_inv(KCSpu) /\LsT2':=xor(CSts1,xor(CSotp',CSpseudn'))
 /\LsT3':=xor(LsT2',LSpseudn) /\SigLS5':=xor({ECDSA(Datas.LsT3')}_inv(KCSpu),SigLS4')
% LS sends(LSm) to CS
/\SND(CS.SigLS5'.SS.LSrn'.Datas) /\witness(LS,CS,auth7,{SigLS5,LSts4,LSrn'})
end role
```

Figure 7.4: *LS* role of REISCH in HLPSL

```
role centralServer(CS,LS:agent, KCSpu,KLSpu:public_key, ECDSA:hash_func,SND,RCV:channel(dy))
played_by CS def=
local
     State:nat,
     SigCS1,SigCS2,LSts4:text, LSpseudo,LSpseudn,CSpseudo,CSpseudn:text,
     CsT1,CsT2,CsT3,CsT4,LSts2,CSts1,CSts2,CSotp,CSrn,LSrn,SS,Datas,LSotp,SigLSi,SigLS5:text
const
     sec6,sec7,auth5,auth6,auth7:protocol_id
init
     State := 0
transition
% CS receives from LS
1.State=0
  /\RCV(CS.LSotp'.SS)=|> State':=1 /\CSts1':=new()/\CSpseudn':=new()
  /\CSotp':=new()/\CSrn':=new() /\CsT1':=xor(CSts1',xor(CSotp',xor(LSpseudn,CSrn')))
  /\CsT2':=xor(CSts1',xor(CSotp',xor(CSpseudn,LSpseudn)))
  /\request(CS,LS,auth5,{CSpseudo,LSpseudn,LSts2})
  /\secret({LSpseudn,CSpseudn},sec6,{CS,LS}) /\secret({SigLSi,CSotp'},sec7,{CS,LS})
% CS sends to LS
  /\SND(LS.{ECDSA(CsT2')}_inv(KLSpu).CsT1'.CsT2'.SS.CSrn')
  /\witness(CS,LS,auth6,{CSpseudn,LSpseudn,CSts1'})

% CS receives from LS
2.State=1
  /\RCV(CS.SigLS5'.SS'.LSrn'.Datas')=|> State':=2/\CSts2':=new()
  /\SigCS1':={ECDSA(xor(LSts4,xor(LSpseudn,xor(LSrn,CSrn))))}_inv(KCSpu)
  /\CsT3':=xor(CSts1,xor(CSotp,CSpseudn)) /\CsT4':=xor(LSpseudn,CsT3')
  /\SigCS2':=xor({ECDSA(Datas.CsT4')}_inv(KCSpu),SigCS1')
  /\request(CS,LS,auth7,{SigLS5,LSts4,LSrn'})
end role
```

Figure 7.5: CS role of REISCH in HLPSL

REISCH adds sufficient randomization to hide security parameters, and patient records are protected even after LS is penetrated, while the scheme in Sui & de Meer (2019) needs to support randomization and protect user information when a demand-response management unit is penetrated by an intruder. Besides, an intruder can send messages from a forged unit and deceive users after penetrating this module and revealing information.

REISCH is robust against information leakage, while the scheme in Hathaliya et al. (2019) uses a 160-bit key that is vulnerable to attacks. It explicitly sends patient identities within the encrypted message in the login and authentication phases. If an intruder can break the encryption, he/she uses this information in data disclosure. REISCH uses ECDSA-BLAKE2bp and random pseudonyms to secure data signing. The scheme in Furtak et al. (2019) is based on SHA1 and HMAC, which are vulnerable to attacks in signing and authenticating collected data. It also does not include a pseudonym mechanism to protect SN_i parameters from misbehaving.

```
role session(SNi,CHi,LS,CS:agent, KSNpu,KCHpu,KLSpu,KCSpu:public_key, ECDSA:hash_func ,MAXv:
             message, SNpseudi,SNsli,Datai:text)
def=
local
     SND1,RCV1,SND2,RCV2,SND3,RCV3,SND4,RCV4:channel(dy)
composition
     sensori(SNi,CHi,LS,KSNpu,KCHpu,KLSpu,ECDSA,SNpseudi,SNsli,Datai,SND3,RCV3)
     /\clusterHeadi(CHi,LS,SNi,KCHpu,KLSpu,KSNpu,ECDSA,SNpseudi,SNsli,MAXv,SND2,RCV2)
     /\localServer(LS,CHi,SNi,CS,KLSpu,KCHpu,KSNpu,KCSpu,ECDSA,SNpseudi,SNsli,SND1,RCV1)
     /\centralServer(CS,LS,KCSpu,KLSpu,ECDSA,SND4,RCV4)
end role

role environment()
def=
const
     kSNpu,kCHpu,kLS,kCS,ki:public_key, ecdsa:hash_func,datai,snpseudi,snsli:text,
     sni,chi,ls,cs,i:agent,maxV:message,
     sec1,sec2,sec3,sec4,sec5,sec6,sec7,auth1,auth2,auth3,auth4,auth5,auth6,auth7:protocol_id
     intruder_knowledge = {sni,chi,ls,i,kSNpu,kCHpu,kLS,ki}
composition
     session(sni,chi,ls,cs,kSNpu,kCHpu,kLS,kCS,ecdsa,maxV,datai,snpseudi,snsli)
     % Check replay attack
     %/\ session(sni,chi,ls,cs,kSNpu,kCHpu,kLS,kCS,ecdsa,maxV,datai,snpseudi,snsli)
     % Check MITM attack
     %/\ session(chi,sni,ls,cs,kSNpu,kCHpu,kLS,kCS,ecdsa,maxV,datai,snpseudi,snsli)
     % Check impersonate SNi
     /\ session(i,chi,ls,cs,kSNpu,kCHpu,kLS,kCS,ecdsa,maxV,datai,snpseudi,snsli)
     % Chekc impersonate CHi
     %/\ session(sni,i,ls,cs,kSNpu,kCHpu,kLS,kCS,ecdsa,maxV,datai,snpseudi,snsli)
     % Check impersonate LS
     %/\ session(sni,chi,i,cs,kSNpu,kCHpu,kLS,kCS,ecdsa,maxV,datai,snpseudi,snsli)
end role

goal
     secrecy_of sec1,sec2,sec3,sec4,sec5,sec6,sec7
     authentication_on auth1,auth2,auth3,auth4,auth5,auth6,auth7
end goal
environment()
```

Figure 7.6: Session, environment and goal roles of REISCH in HLPSL

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/REIESCH.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 532.61s
  visitedNodes: 1899 nodes
  depth: 7 plies
```

Figure 7.7: Simulation result of REISCH using OFMC backend

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/REIESCH.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 325 states
  Reachable  : 325 states
  Translation: 0.12 seconds
  Computation: 0.01 seconds
```

Figure 7.8: Simulation result of REISCH using CL-AtSe backend

Table 7.1: Comparison of security features between REISCH and other data collection schemes

Security feature	Fan & Gong (2012)	Lavanya & Natarajan (2017*b*)	Staudemeyer et al. (2018)	Malathy et al. (2018)	Sharavanan et al. (2018)	Sui & de Meer (2019)	Hathaliya et al. (2019)	Furtak et al. (2019)	**REISCH scheme**
Anti MITM		✓				✓	✓	✓	✓
Anti replay	✓	✓	✓			✓	✓	✓	✓
Availability	✓	✓					✓	✓	✓
Anti Sybil	✓							✓	✓
Anti Wormhole								✓	✓
Anti fake sink								✓	✓
Anti repository attack					✓				✓
Anti eavesdropping		✓	✓	✓	✓	✓	✓	✓	✓
Anti node replication								✓	✓
Anti collision/preimage		✓							✓
Pseudonym						✓	✓		✓
Homomorphic						✓			✓
Mutual authentication		✓					✓		✓

7.2.2 Performance Analysis

Analysis of performance is an important factor when designing data/records collection protocols for HWSNs. In this section, we analyse REISCH performance theoretically and experimentally, and compare it with existing schemes.

- **Theoretical Performance Analysis**
  Theoretically, REISCH uses several features that qualify it to be efficient in HWSN performance. First, it relies on the ECDSA algorithm that integrates data collected by small keys compared to public key cryptography algorithms (RSA, DSA and Elgamal). For instance, ECDSA produces 256-bit equivalent keys in security for 3072-bit keys produced by RSA, DSA, and Elgamal. Second, REISCH implicitly uses BLAKE2bp with ECDSA which is dramatically efficient in the operation of a hash function instead of SHA1. Third, REISCH uses the homomorphic property to combine signatures in CHs and significantly reduces energy dissipation. Fourth, REISCH relies on the LEACH routing protocol, which is the most efficient energy-saving protocol in WSN. Fifth, REISCH relies on rapid random pseudonyms to protect medical records rather than complex and costly processes of data encryption and k-anonymity. Finally, REISCH uses XML to support efficient patient data management. Therefore, these features allow REISCH to maintain the energy of the SNs as long as possible.

- **Experimental Performance Analysis**
  More importantly, we will evaluate the performance of REISCH in the execution of security operations in conjunction with the saved and collected data. As noted in previous sections, SNs requires performance-efficient signatures to perform services for as long as possible in patients' monitoring

Table 7.2: REISCH simulation parameters

Parameter	Value
Area of WSN	1000m*1000m
SN	200
LS location	500*500
Initial energy	25J
Size of packet	1MB
Control packet size	50B
Rounds	1000
Routing protocol	LEACH
Propagation energy	10 nJ/bit/m^2
Multi-hop propagation energy	0.0013 pJ/bit/m^4
Aggregation energy	5 nJ/bit/signal

Table 7.3: REISCH computational processes

Process type	Number of process			Running time	Storage	Energy
	SN	CH	LS			
SHA1 hash	1	1	Many	0.05529	160-bit	0.008464
BLAKE2bp hash	1	1	Many	0.040606	512-bit	0.006216
Keys generation	2	2	2	0.000859	256-bit	0.000132
Point multiplication	2	2	Many	0.000543	-	0.000083
ECDSA-SHA1 signature	1	1	-	0.072838	256-bit	0.011151
ECDSA-SHA1 verification	-	-	Many	0.073103	-	0.011191
ECDSA-BLAKE2bp signature	1	1	-	0.050046	256-bit	0.007662
ECDSA-BLAKE2bp verification	-	-	Many	0.052076	-	0.007972

and care. We will provide tests on hash algorithms (SHA and BLAKE) and the signature algorithm (ECDSA). Additionally, we apply these algorithms to HWSN to analyse performance properties such as time, storage and energy. Table 7.2 shows all the simulation parameters used in HWSN, while Table 7.3 shows computational operations in the REISCH scheme. All hash and signature algorithms are implemented by C language while WSN is designed in Octave under Ubuntu 16.04 LTS, processor Intel Core i5 2.6GHz, OS type 64-bit, Memory 4 GiB and disk 32.0 GB.

The computation of energy in our scheme is based on the Micaz sensor specification. This process uses parameters such as current (0.0567), voltage (2.7) and time to extract both power and energy using $power = current * voltage$ and $energy = time * power$. We relied on real data provided by the City of Melbourne that is licensed under CC 4.0 (City of Melbourne Open Data Team October 19, 2018). This data was generated by sensors to monitor environmental parameters such as humidity, temperature and light, as well as include some information such as timestamp and ID. We divided this data into different sizes (200K, 400K, 800K and 1M) and then converted it into an XML context. Furthermore, there are no communication channels between patients and SNs. To check performance, we implemented the SHA1-160, SHA2-256, BLAKE2s-256, BLAKE2b-512, BLAKE2sp-256, and BLAKE2bp-512 algorithms with 1MB data size as shown in Figure 7.9. Also, Figures 7.11, 7.12 and 7.13 show execution time (minimum, maximum

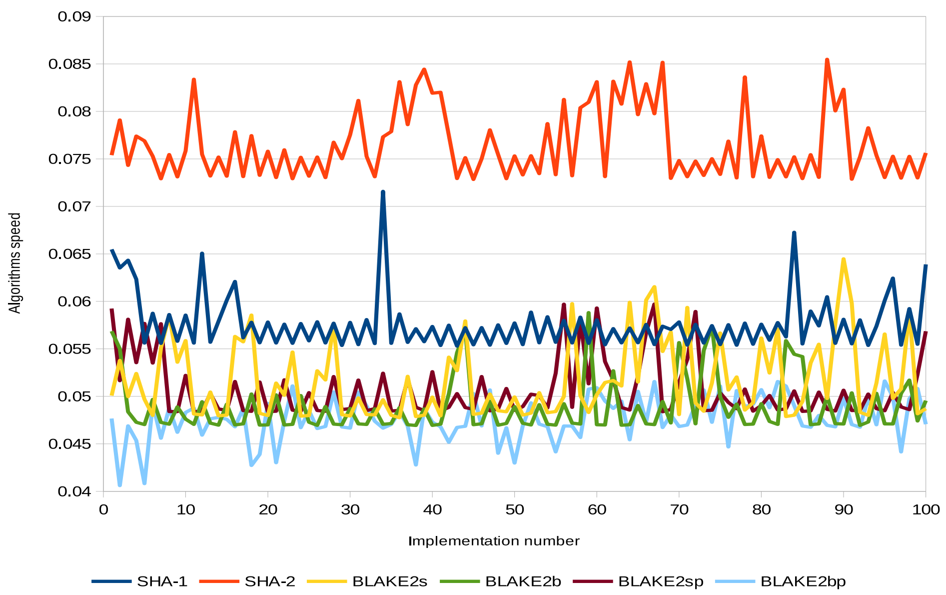


Figure 7.9: Comparison of SHA and BLAKE2 with 1MB data

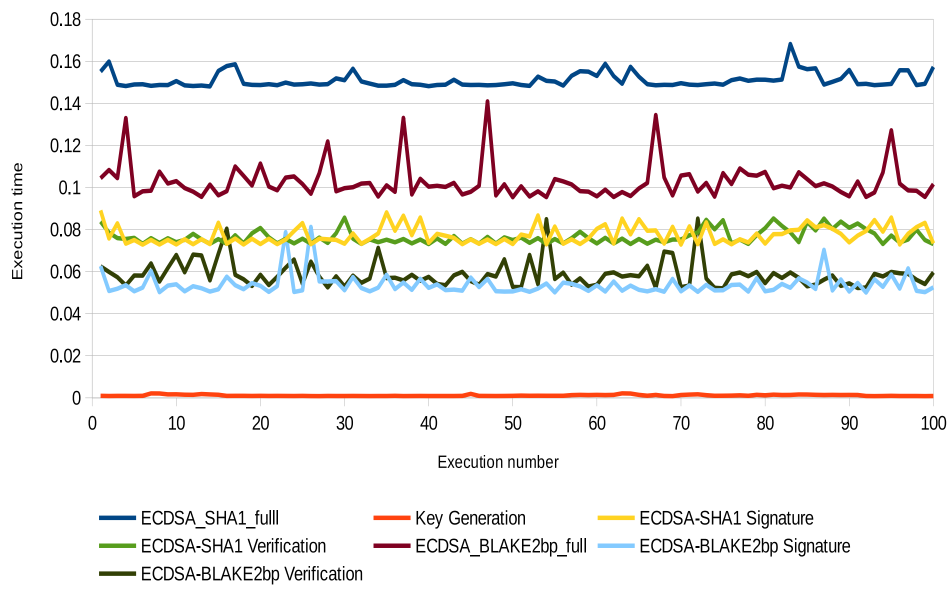


Figure 7.10: Execution time of ECDSA-SHA1 and ECDSA-BLAKE2bp with 1MB data

and average) for hash functions when using 1M data. We noticed that BLAKE2bp is the best performance in terms of execution time in all figures. In addition, Figure 7.10 shows that ECDSA-BLAKE2bp gives the best execution time of ECDSA-SHA1. Also, Figures 7.14, 7.15 and 7.16 show the execution time (minimum, maximum and average) for the ECDSA algorithms when using 1M data. Thus, the amendment to the ECDSA algorithm is entirely appropriate for the use of security measures with the longest life of the SNs from the original algorithm.

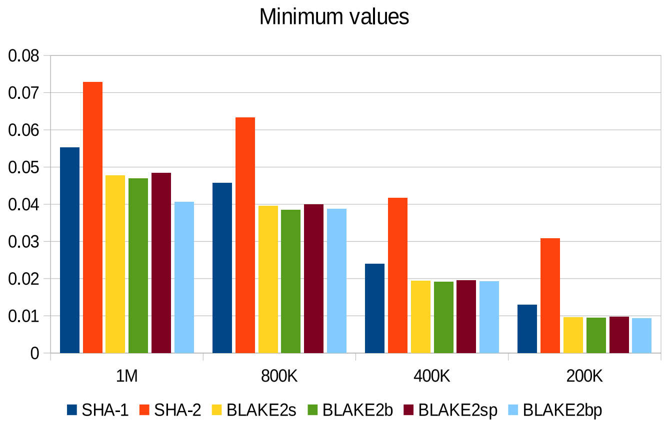


Figure 7.11: Minimum execution time of hash functions with 1MB data

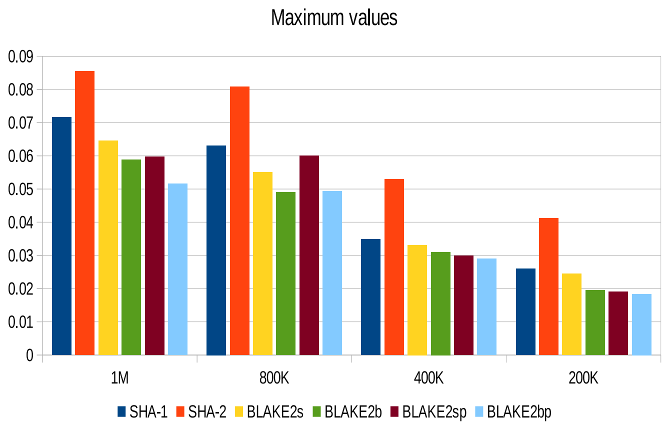


Figure 7.12: Maximum execution time of hash functions with 1MB data

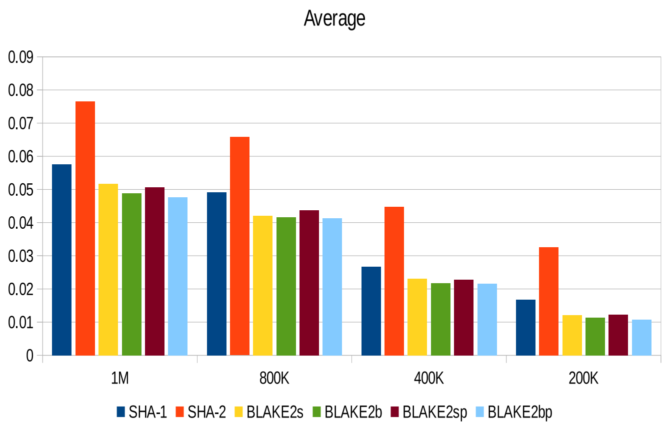


Figure 7.13: Average execution time of hash functions with 1MB data

We have computed message complexity which is the number of messages transmitted between network entities. For each round, SN and CH send one message while CH and LS receive a set of aggregated messages. Thus,

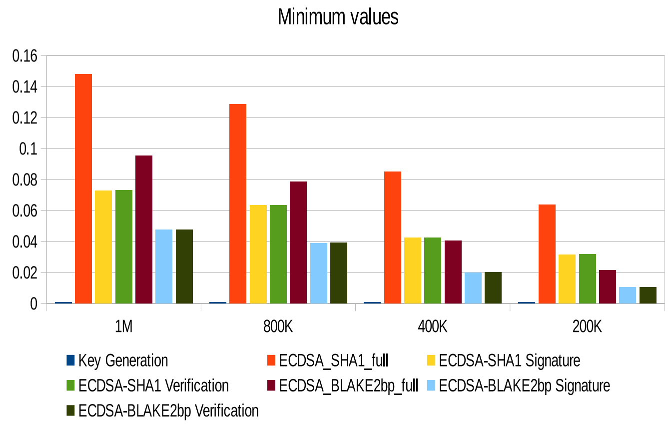


Figure 7.14: Minimum execution time of ECDSA algorithms with 1MB data

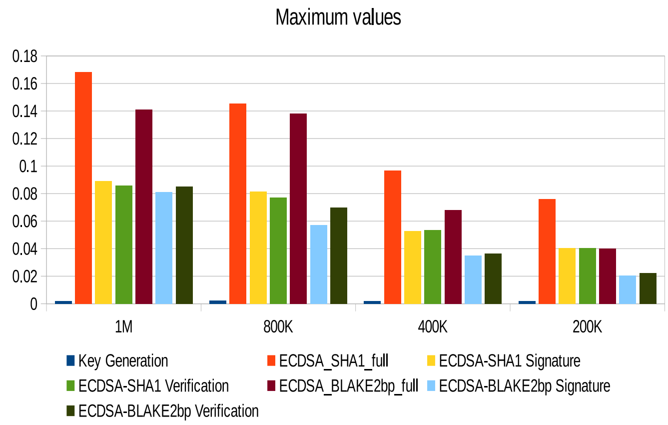


Figure 7.15: Maximum execution time of ECDSA algorithms with 1MB data

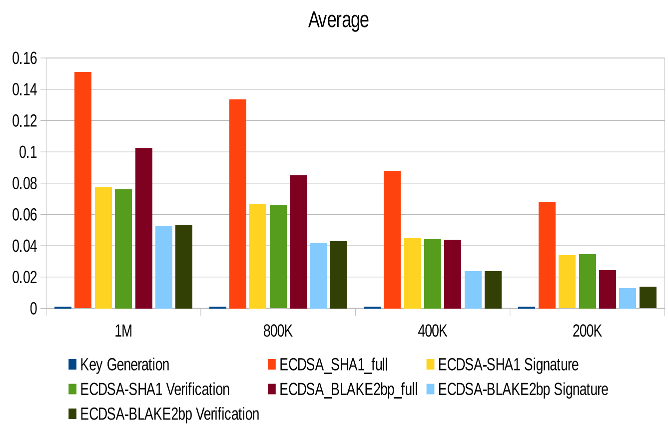


Figure 7.16: Average execution time of ECDSA algorithms with 1MB data

the message complexity with modified algorithms for SNs is (156972), CHs is (8313) and LS is (165285) while with original algorithms for SNs is (142541), CHs is (7572) and LS is (150113). Message overhead is to calculate the message size between network entities. In each round, the message overhead

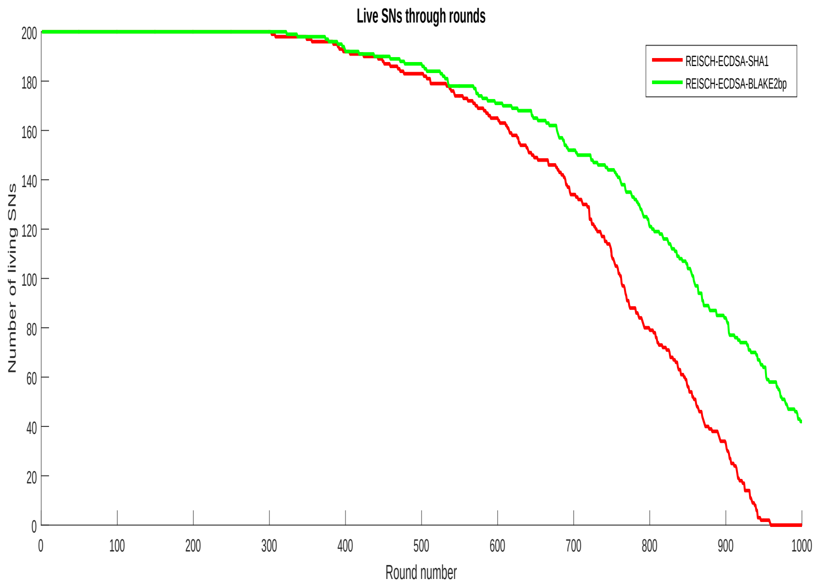


Figure 7.17: Comparison of alive SNs

of SN is $(1024 + 32)$ bytes while CH is $(15360 + 32)$ bytes. Figure 7.17 shows that REISCH-ECDSA-BLAKE2bp is better than REISCH-ECDSA-SHA1 in terms of alive SNs, namely, HWSN will have a longer life span to collect patients' data when it uses REISCH-ECDSA-BLAKE2bp. It also contributes to the energy balance of SN_s which improves the continuity of electing CH_s and collecting the most patient data in each round. We noticed that REISCH with the modified algorithm (ECDSA-BLAKE2bp) has more alive SNs by 24% than the original algorithm (ECDSA-SHA1). Furthermore, the first SN dies when using the modified algorithm in round 322, while in the original algorithm is in round 295.

- **Performance Comparison**

  Finally, we will make a comparison on performance with the existing schemes (Table 7.4 shows a comparison of the ECDSA's signature and verification (running time) between our scheme and existing schemes). Due to different environments, security parameters and network parameters such as key length, number of SNs, etc., it is very difficult to compare schemes' performance. However, we have made some comparisons to illustrate the superiority of REISCH on the existing schemes in terms of performance.

The scheme in Fan & Gong (2012) focused on accelerating ECDSA's verification based on computation results for neighbouring SNs. These computations consume additional energy. In addition, this scheme is very expensive if applied to a cluster scheme because CH needs to accomplish one PM in each signature for each SN_i and thus will consume energy in the intermediate SNs. REISCH does not need these computations because signatures' verification is performed in LS. Schemes in Kodali (2013), Lavanya & Natarajan (2017*b*), Malathy et al. (2018) used ECDSA to sign data without using homomorphic. Consequently, the performance of the SNs would be very low due to signature and verification processes in each round. The scheme in Kodali (2013) addressed the bits (8 and 32) of data processing in SHA1 but did not address the cost of energy consumption by SHA1. Also, the scheme in Lavanya & Natarajan (2017*b*) did not support the clustering environment to reduce energy consumption and the computation time to generate and verify the signature was not clearly indicated. Furthermore, the scheme in Sharavanan et al. (2018) used SHA2, which is more secure than SHA1 but performs heavy processes that significantly affect the energy of SNs. It also addresses only computations in transport while REISCH addresses computations in transport and processing using BLAKE2bp and homomorphic.

Schemes in Staudemeyer et al. (2018), Hathaliya et al. (2019), Furtak et al. (2019) rely on the use of encryption to protect data without a homomorphic property, since encryption processes extremely consume SNs resources (as mentioned in Section 3.3.4, point 4). While REISCH uses signatures and homomorphic to improve HWSN network performance. Although the scheme in Sui & de Meer (2019) uses encryption and signature of data with homomorphic, encryption can particularly affect network performance, especially through a burden on the servers.
Moreover, the scheme in Furtak et al. (2019) has implemented RSA 2048-bit algorithm which is significantly expensive in encryption operations. In addition, it uses several parameters such as many keys, 2048-bit key length and SN_i addresses (master, replica and gateway) that cause storage problems in the pre-deployment and registration phases (consumption of SNs resources). It has used a random routing of the sensor network without relying on a specific routing protocol such as LEACH. This scheme has considered the structure of the data in the SN_i memory and did not pay attention to the structure of the data as it was transferred to the servers. REISCH uses XML to support performance of the LS and CS without having to convert data formats between network devices. Recent research Kittur & Pais (2019), Kuang et al. (2019), Marino et al. (2019), Zhao et al. (2019), Liu et al. (2019) have used different ways to improve ECDSA's procedures. However, REISCH provides better performance in terms of ECDSA's signature and verification than existing

Table 7.4: Comparison of ECDSA's procedures

Running time (second)	Fan & Gong (2012)	Kodali (2013)	Malathy et al. (2018)	Kittur & Pais (2019)	Kuang et al. (2019)	Marino et al. (2019)	Zhao et al. (2019)	Liu et al. (2019)	**REISCH scheme**
Signature	0.38	0.941	0.59	0.078	0.3472	0.434	0.084	0.051	0.050
Verification	0.65	-	-	0.079	-	0.429	0.088	0.105	0.052

schemes (as shown in Table 7.4).

7.3 Analysis of RAMHU Scheme

In this section, we will analyse RAMHU's security and performance in both theoretical and experiential aspects.

7.3.1 Possible Attacks on the RAMHU Scheme

In this section, theoretical and experimental security analyses are presented. We describe how our scheme (RAMHU) applies security requirements to protect user authentication information in the healthcare system. In this section, we also show how RAMHU provides a high-security level against these attacks meanwhile providing propositions and proofs.

- **Proposition 1 – Privileged-insider attack**
  The internal intruder (II) cannot detect the private key or password (privileged-insider attack) for another legitimate user in the healthcare network.
  Proof
  II needs to eavesdrop on authentication requests between C_i and CS. After that, he/she tries to perform an analysis of these requests based on his/her access privileges to the network. First, the analysis of these requests to obtain the private key or password is infeasible because the authentication request is encrypted by the ECIES 256-bit algorithm; II cannot decrypt these requests with his private key. Second, if II breaks the encryption (which is impossible), he/she is unable to extract the PW_i value ($tmpPW_i = PW_i \oplus N_{C_i} \oplus GM \oplus C_iSig_1$) in the login phase because it is hidden and depends on the values of GM, C_iSig_1, and N_{C_i} where II does not know the GM value of the legitimate user's device, and the C_iSig_1 value depends on the CM value that is also not known to II. Therefore, RAMHU prevents a privileged insider attack.

- **Proposition 2 – Stolen device/application attack**
  I will not gain any benefit from device or client application theft (stolen

device/application attack) for use in authentication as a legitimate user.

Proof

In the first case, I stole a legitimate user device (such as a laptop) that contains a client application. In our scheme, user's PW_i, and IDs are not stored on the user's device, or in the client application. In addition, the private key is hidden and random for each authentication process by $C_iK_{pr_i} \oplus GM \oplus PW_i \oplus N_{C_i}$. Additionally, the proof in Proposition 1 shows that the PW_i value cannot be extracted from the $tmpPW_i$. If I is II, he/she also cannot use his/her IDs and PW_i to access information or other user data because both CS and AS perform matching IDs, and PW_i to determine user-related information and data. In the second case, the attacker steals only the client application. I cannot use the application on another device because it does not have OTP_i or original MAC, which prevents the application from being used on another device even if I knows the IDs and PW_i. The original MAC mechanism ($C_iSig_1 = h(CM\|N_{C_i}\|TS_{C_i})$ prevents the fake authentication request from being verified in the server because I cannot generate an original MAC address (CM) in C_iSig_1. Thus, RAMHU is resistant to stolen attacks.

- **Proposition 3 – Replay attack**

  I cannot use the login/registration/authentication request later (replay attack).

  Proof

  I tries to get a login/registration/authentication request for a legitimate user to send it later, and thus gains access to the network. This case is infeasible in our scheme because all entities (C_i, CS, and AS) use a timestamp (such as $TS_{CS} - TS_{C_i} \leq \triangle T$, where $\triangle T$ is the maximum transfer delay rate) that prevents the attacker from sending the authentication request at a later time. Furthermore, signatures and random nonces are not usable in subsequent times. Hence, RAMHU successfully resists replay attacks.

- **Proposition 4 – Man-in-the-Middle attack**

  I does not have the ability to intercept, modify, and replace authentication requests (man-in-the-middle (MITM) attack) between RAMHU's entities.

  Proof

  Assume that an I attempts to intercept encrypted login/authentication requests (such as $Enc_i = TS_{C_i}\|N_{C_i}\| \; CiSig_1\|UP_{C_i}^{CS}\|MP_{C_i}^{CS}\|tmpPW_i\|CiSig_2$) among network entities, and then modifies or replaces these requests with his/her messages to send to network entities. However, the attacker cannot replace exchanged requests between C_i, CS, and AS because, first, he/she does not know the private keys ($C_iK_{pr_i}$, CSK_{pr_i}, ASK_{pr_i}) and therefore, the decryption process is computationally infeasible with 256 bits key length and difficulty in solving ECDLP. Second, mutual authentication with PHOTON-256 signatures prevents the modification of requests between RAMHU's entities. As a result, RAMHU gracefully overcomes the MITM attack.

- **Proposition 5 – ID or Password Guessing Attack**
  EI cannot guess ID_s and PW_i (guessing attack) for legitimate users of the RAMHU protocol.
  Proof
  Assume that an EI was able to penetrate the encryption (Enc_i) between C_i and the CS (from Proposition 4, this assumption is infeasible). This EI tries to guess PW_i in a login request to use it to access the network as a legitimate user. EI cannot detect PW_i for any authorised user (either on-line or off-line) because he/she does not know the configured process to protect PW_i ($tmpPWi = PW_i \oplus N_{C_i} \oplus GM \oplus C_iSig_1$) and does not know the MAC address for that user, and thus the process of deriving PW_i is infeasible (Proposition 1). It is an extremely difficult process to guess PW_i from the $tmpPW_i$, which is 64 hex (256 bits). In addition, EI cannot detect UID_i and MID_i for any legitimate user because of the use of multi pseudonyms mechanism for users and medical centres instead of sending real information to legitimate users. As a result, RAMHU is safe against guessing attacks.

- **Proposition 6 – Client impersonation attack**
  I cannot impersonate a legitimate user or device in the network (client impersonation attack).
  Proof
  Assume that an attacker tries to impersonate a login request (such as $Enc_i = TS_{C_i} \| N_{C_i} \| CiSig_1 \| UP_{C_i}^{CS} \| MP_{C_i}^{CS} \| tmpPW_i \| CiSig_2$) for a legitimate user. This I can create TS_{C_i} and N_{C_i} but does not know PW_i and $C_iK_{pr_i}$ (Proposition 1, and 4) for the legitimate user, namely, I cannot impersonate the user identity. I also tries to impersonate the legitimate user's device by programmatically changing the MAC address to a legitimate one to gain access to the network. This case is infeasible because the original MAC check (CM) in $CiSig_1 = h(CM \| N_{C_i} \| TS_{C_i})$ detects the attacker's attempt to mimic the legitimate user's device (as proved in Proposition 2). Therefore, RAMHU withstands instances of impersonating the user's identity and device.

- **Proposition 7 – Server impersonation attack**
  I cannot impersonate (server impersonation attack) the central server (CS) and cheat the client (C_i).

Proof

Assume that an I traps login requests from C_i to CS. The attacker tries to deceive C_i by sending fake requests to C_i in order to inform them that he is a legitimate server. I needs the private key for CS to decrypt and to accomplish the attack. Mutual authentication prevents I from impersonating CS's requests (such as $Enc_i = TS_{CS} \| N_{CS} \| UP_{CS}^{C_i} \| MP_{CS}^{C_i} \| CSSig_5$) and sending them to C_i. This mechanism ensures that C_i deals with legitimate CS. Consequently, our protocol effectively resists server impersonation attack.

- **Proposition 8 – DoS attack**

  I cannot effectively perform a DoS attack against our scheme.

  Proof

  In order for I to execute a DoS attack against CS and AS, he/she needs to decrypt the login request and change its data or send the same request multiple times to destroy the servers. However, in the first case, decryption and change of signatures are infeasible as proved in Propositions 2 and 4. CS checks signature validity and rejects login requests containing fake signatures, and I cannot execute a collision or preimage attack because PHOTON-256 supplies FPR, SPR, and CR. In the second case, the attacker sends the same request multiple times. This status is infeasible because the CS or AS checks the timestamp (TS_{C_i}, TS_{CS}, TS_{AS}) for each login/authentication request and eliminates all late requests (Proposition 3) without checking the other security parameters such as PW_i, CM, and multi pseudonyms. In case I can break the encryption, he/she can change the timestamp and nonce, but cannot tamper with the signatures. RAMHU prevents this condition during C_iSig_1 and does not need to check the remaining security parameters. Therefore, RAMHU successfully resists DoS attacks.

- **Proposition 9 – Password change attack**

  I cannot detect new PW_i or change old PW_i (password change attack) to prevent legitimate users from accessing the network.

  Proof

  Assume that an I intercepts a request to change PW_i between C_i and CS. I obtains an encrypted request (such as $Enc_i = TS_{C_i} \| NC_1 \| NC_2 \| NC_3 \| UP_{C_i}^{CS} \| MP_{C_i}^{CS} \| \; tmp\_oldPW_i \| tmp\_newPW_i \| C_iSig_1$). If I can decrypt (this process is infeasible as proved in Propositions 1, and 4), he/she will find a temporary password and cannot derive new PW_i because it depends on the $C_iSig_1 = h(TS_{C_i} \| NC_1 \| UP_{C_i}^{CS} \| MP_{C_i}^{CS} \| oldPW_i)$. The signature operation (C_iSig_1) is based on the old PW_i which is not explicitly sent to CS in this phase. I does not know old PW_i

and therefore, he/she cannot create a signature to complete the PW_i change process. As a result, RAMHU provides a reliable solution against password change attacks.

- **Proposition 10 – Eavesdropping attack**
  I does not gain plaintext and useful information when an eavesdropping attack has applied to our scheme.
  Proof
  Assume that an I eavesdrops on login/authentication requests to gain information about user authentication and access to the network. However, in our scheme, I will not benefit from requests that are intercepted because RAMHU uses ECIES algorithm with key 256 bits to encrypt authentication information. The attacker can only decrypt requests by deriving private keys and this operation is infeasible (as proved in Propositions 1, and 2) due to key length, and random encryption with nonces (anonymity). Therefore, our protocol is resistant to eavesdropping attack.

- **Proposition 11 – Traceability attack**
  I cannot track exchanged login/authentication requests (traceability attack) between all entities in the RAMHU scheme.
  Proof
  I attempts to collect as many login/authentication requests as possible and then performs an analysis of those requests that helps him/her to perform user identity tracing. When an I succeeds in tracking user requests, he/she can detect and distinguish patient data. All exchanged requests among RAMHU's entities do not contain direct user information (such as username). RAMHU replaces the real user ID_s (UID_i and MID_i) with pseudonyms. Our protocol uses multi pseudonyms ($UP_{C_i}^{CS}$, UP_{CS}^{AS}, UP_{AS}^{CS}, and $UP_{CS}^{C_i}$ for users and $MP_{C_i}^{CS}$, MP_{CS}^{AS}, MP_{AS}^{CS}, and $MP_{CS}^{C_i}$ for medical centres) to prevent attackers from tracking user requests and revealing their identities when transferred between RAMHU entities (C_i, CS, and AS). Hence, RAMHU resists traceability attacks.

- **Proposition 12 – Revocation attack**
  I cannot penetrate or send a revocation request to delete or remove user account information (revocation attack) in the RAMHU protocol.
  Proof
  Assume that an I tries to penetrate a revocation request. The attacker tries to analyse the request and use it to prevent users from accessing the network's services. Depending on Propositions 1, 5, and 11, the attacker cannot extract or distinguish UID_i, MID_i and PW_i from the revocation

request. The attacker does not know and cannot extract a reason from the $tmpRR_i$, which is based on the values of RR_i, N_{C_2}, and C_iSig_1. In Propositions 2 and 8, I cannot perform collision, preimage, and second preimage attacks against the PHOTON-256 algorithm. Thus, our protocol prevents penetration of revocation request.

- **Proposition 13 – Verifier attack**
  I cannot extract users' passwords from datasets in CS (verifier attack).
  Proof
  Assume that I tries to penetrate the datasets in CS. If the attacker is EI, he/she cannot penetrate datasets because he/she does not have a K_{pr}, UID, MID, OTP, and PW_i. If the attacker is II, when he penetrates datasets in CS and wants to impersonate another user's identity. First, he/she cannot distinguish this information for a particular user because the real information for users is stored in the AS. Second, since the CS does not contain the passwords' dataset, II will not benefit from hacking datasets such as pseudonyms and cannot create a request and send it to AS because it does not know users' passwords. Therefore, RAMHU resists verifier attacks meritoriously.

- **Proposition 14 – Leakage attack**
  The attacker does not get any information leaked from requests and responses exchanged (leakage attack) between RAMHU entities.
  Proof
  Suppose that I listens to some exchanged requests among C_i, CS, and AS and tries to find any information that helps him/her to penetrate network authentication such as sending an ID explicitly, or sending a password with weak encryption. Exchanged requests between RAMHU entities such as a password update request ($Enc_i = TS_{C_i}\|N_{C_1}\|N_{C_2}\ \|N_{C_3}\ \|UP_{C_i}^{CS}\|MP_{C_i}^{CS}\|$ $tmp\_oldPW_i\|\ tmp\_newPW_i\|C_iSig_1$) show that I does not receive any leaked real information for users during transmission, such as IDs and PW_i (all information is anonymous and hidden) that could be useful in penetrating the healthcare network. Therefore, RAMHU resists leakage attacks.

1. **Experimental Security Analysis**
   In the previous subsection, we have listed various possible attacks on the RAMHU scheme. In this section, we provide the proposed scheme simulation using the AVISPA tool to verify our scheme; whether safe or unsafe. This tool has been accepted by researchers in recent years and is widely used (Amin, Kumar, Biswas, Iqbal & Chang 2018, Amin, Islam, Biswas, Khan &

Kumar 2018). It has been used to check security problems in authentication procedures and to ensure that known attacks are not able to penetrate user authentication information.

- **RAMHU Scheme with AVISPA**

  In this section, we illustrate the implementation and simulation of RAMHU with the AVISPA tool using HLPSL language. Our scheme depends on three core roles: clienti, centralServer, and attributesServer played by C_i, CS and AS respectively, in addition to the supporting roles of session, and environment, goal specification section. Each role contains parameters, variables, and local constants. Each basic role contains a transition section that indicates the sequence of communication between entities. Each supporting role contains a composition section that indicates the binding of roles and sessions.

  Asymmetric encryption has been implemented between scheme entities (C_i, CS, and AS) during public key exchange (KC_{pu}, KCS_{pu}, and KAS_{pu}) to perform confidentiality as well as mutual authentication to ensure the legitimacy of related parties in the phases of the proposed scheme (initial setup, registration, login, and authentication). Moreover, it uses nonces (N_c, N_{cs}, and N_{as}) and timestamps (TS_c, TS_{cs}, TS_{as}) to support the features of anonymity and freshness. Our scheme accomplishes 10 secrecy goals and six authentication goals as noted in the goal section in Figure 7.22. Figure 7.18 shows RAMHU's framework in AVISPA.

  C_i receives the start signal and changes the state flag (State variable) from 0 to 1. It replaces UID and MID with UP_c and MP_c and calculates the timestamp (TS_c') and new nonce (N_c') by the new() operation, and computes the signatures (C_iSig_1 and C_iSig_2). The password hiding process is also calculated in the operation of the $TmpPW$ parameter. It encrypts the registration and login request (TS_c', N_c', GM', C_iSig_1', UP_c', MP_c', OTP_i, $Temporary\_PW'$, and C_iSig_2') by the public key (KCS_{pu}) to establish a reliable communication with CS.

  C_i sends the request to CS where the transmission process is performed by the SND() operation. It achieves a set of secret goals ($sec1$ to $sec6$) with both CS and AS; these secrets are known only to and kept only by the intended parties. For example in $sec1$, UID, MID, and PW are known only to, and kept only by C_i, and AS, while in $sec3$ GM, and CM have been known and kept only to C_i and CS, since these parameters are not transmitted directly during the transition of information between network parties such as PW implicitly is in the calculation of $TmpPW$ and CM implicitly is in C_iSig_1.

C_i also achieves the goal of authentication using a statement (witness) with the parameters (C_i, CS, ci_cs_auth2, N_{cs}, TS_c), which means that C_i is a witness that the security parameters (N_{cs}, TS_c) are fresh and correct, and CS uses a statement (request) to validate parameters with the strong authentication goal (ci_cs_auth2) specified in the goal section (Figure 7.22). C_i receives the authentication response via the RCV () operation and sent from AS by CS. Then, C_i decrypts the response using its private key to verify the security parameters. If all security parameters are verified correctly, C_i performs the mutual authentication process securely.

As shown in Figure 7.20, CS receives a registration and login request by RCV () operation in state 0 and decrypts it with its private key and then checks the parameters and signatures to accomplish the secrecy and authentication goals. The CS changes the state signal from 0 to 1 and constructs an authentication request based on the security parameters (TS'_{cs}, N'_{cs}, UP_{cs}, MP_{cs}, $CSSig_3$, and $TmpPW$) and encrypts it by the public key of AS. CS performs strong authentication with AS during witness (CS, AS, as_cs_auth4, TS'_{cs}, N'_{cs}) to accomplish the authentication goal (as_cs_auth4) based on the timestamp and fresh nonce and is validated in the AS by statement (request).

In state 1, CS receives an authentication response from AS and checks the security parameters after decryption with its private key. It changes the state signal to 2 and then constructs and sends the authentication response to C_i with two strong authentication goals (cs_ci_auth1 and cs_ci_auth5) based on the security parameters (N_c, TS_{cs}, $TmpPW$, and OTP_i). The authentication process begins by sending requests from clients to the server. Therefore, the $client_i$ role includes the start signal as shown in Figure 7.19.

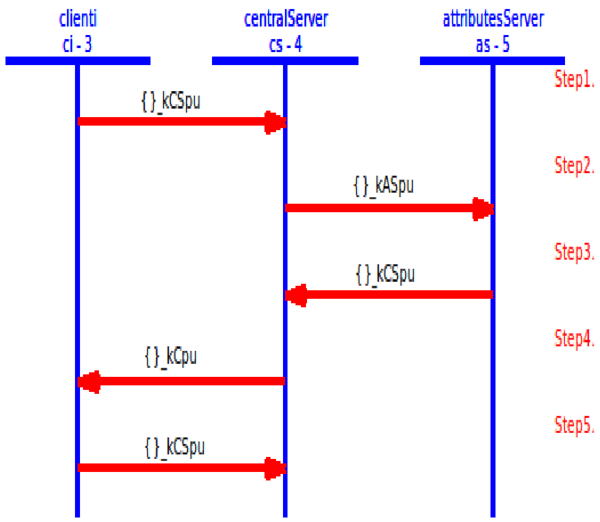


Figure 7.18: RAMHU's framework in AVISPA

```
role clienti(Ci,CS,AS:agent,  KCpu,KCSpu:public_key, H:hash_func, UID,MID,OTPi,PW,GM,CM:message
              , SND,RCV:channel(dy))
played_by Ci def=
local
     State:nat,
     TSc,TScs,Nc,Ncs:text, CiSig1,CiSig2:text, UPc,MPc,UPcs,MPcs,TmpPW:message
const
     sec1,sec2,sec3,sec4,sec5,sec6, cs_ci_auth1,ci_cs_auth2:protocol_id
init
     State := 0
transition
1. State  = 0
   /\RCV(start) =|> State':= 1 /\UPc':=UID /\MPc':=MID/\GM':=Ci /\Nc':=new() /\TSc':=new()
   /\CiSig1':= H(CM.Nc'.TSc') /\CiSig2':= H(GM'.Nc'.TSc'.CiSig1'.UPc'.MPc'.OTPi.PW)
   /\TmpPW':=xor(PW,xor(Nc',xor(GM',CiSig1')))

% Registeration and login phase
% Ci sends security parameters to CS
   /\SND({TSc'.Nc'.GM'.CiSig1'.UPc'.MPc'.OTPi.TmpPW'.CiSig2'}_KCSpu)
   /\secret({UID,MID,PW},sec1,{Ci,AS}) /\secret(PW,sec2,{Ci,CS}) /\secret({GM,CM},sec3,{Ci,CS})
   /\secret({CiSig1',CiSig2'},sec4,{Ci,CS}) /\secret({TmpPW',OTPi},sec5,{Ci,CS})
   /\secret({UPc,MPc,UPcs,MPcs},sec6,{Ci,CS})

2. State  = 1
% Ci receives authentication response from CS
   /\RCV({TScs'.Ncs'.Nc.UPcs'.MPcs'. H(TScs'.Ncs'.UPcs'.MPcs')}_KCpu)=|>
   State':= 2 /\SND({Ncs'}_KCSpu)

% Clienti checks that the received security parameters
   /\request(Ci,CS,cs_ci_auth1,{Nc,TScs}) /\request(Ci,CS,cs_ci_auth5,{TmpPW,OTPi})
% Clienti sends Ncs to prove her identity
   /\witness(Ci,CS,ci_cs_auth2,{Ncs,TSc})
end role
```

Figure 7.19: C_i role of RAMHU in HLPSL

AS receives the authentication request and decrypts it with the private key as shown in Figure 7.21. It accomplishes five secret goals, and accomplishes two authentication goals (as_cs_auth4 and as_cs_auth6) based on TS'_{cs}, N'_{cs}, and PW'. It constructs the authentication response and establishes a strong authentication when validating parameters

(TS_{as}, N'_{cs}, and PW') in CS.

Figure 7.22 displays the roles of session, environment, and goal section. In the session role, a composition process has been performed for the three roles (*clienti*, *centralServer*, and *attributeServer*) and specifies the sending and receiving channels in the Dolev Yao model.

In the environment role, the security parameters, the goals specified in the goal section, and the known information for the intruder (*intruder_knowledge*) have been defined. In this role, one or more sessions are composed, and we tested our scheme with sessions for replay, MITM, and impersonating attacks. We assume that an intruder (I) creates a public key (ki) and has knowledge of the public keys ($kCpu$, $kCSpu$, and $kASpu$) of legitimate entities in the network.
The intruder attempts to resend the registration/login or authentication requests later, intercepts/modifies these requests, or impersonates the participating entities using i_ci, i_cs, and i_as constants rather than ci, cs, and as. The results section shows that these attacks cannot penetrate the security goals in our scheme.

- **Simulation Results**
  In this section, the simulation results in the AVISPA tool are based on two backends (OFMC, and CL-AtSe). Figure 7.23 shows the simulation result with the OFMC backend and Figure 7.24 displays the simulation result with the CL-AtSe backend. From the results shown in Figures 7.23 and 7.24, our scheme clearly and accurately shows the SAFE result in the SUMMARY section, bounded number of sessions in the DETAILS section, the goals of the scheme achieved (as_specified) in the GOAL section as well as statistical numbers such as time, number of nodes, and analysed states in the STATISTICS section for both figures. Based on these results, we note that our scheme is capable of preventing passive and active attacks such as replay, MITM, and impersonating, and that the goals of the scheme in Figure 7.22 prevented the violation of legitimate user information in the network authentication.

2. **Security Comparison**
   In this section, we compare RAMHU with existing authentication schemes in terms of security. We explain RAMHU superiority to solve the deficiencies in existing authentication schemes.

```
role centralServer(Ci,CS,AS:agent, KCpu,KCSpu,KASpu:public_key, H:hash_func, SND,RCV:channel(dy))
played_by CS def=
local
     State:nat,
     UPc,MPc,UPcs,MPcs,UPas,MPas,TmpPW:message, CSSig1,CSSig2,CSSig3:text,
     TSc,TScs,TSas,Nc,Ncs,Nas:text, PW,OTPi:message, GM,CM:message
const
     sec1,sec2,sec3,sec4,sec5,sec6,sec7,sec8,sec9,sec10,
     ci_cs_auth2,cs_ci_auth1,cs_ci_auth5,cs_as_auth3,as_cs_auth4,as_cs_auth6:protocol_id
init
     State := 0
transition
% Registration and login request from Ci
1. State  = 0
   /\RCV({TSc'.Nc'.GM'.H(CM'.Nc'.TSc').UPc'.MPc'.OTPi'.TmpPW'.
          H(GM'.Nc'.TSc'.H(CM'.Nc'.TSc').UPc'.MPc'.OTPi'.PW')}_KCSpu) =|>
   CSSig1':= H(CM'.Nc'.TSc') /\CSSig2':= H(GM'.Nc'.TSc'.CSSig1'.UPc'.MPc'.OTPi.PW)
   /\PW':=xor(TmpPW,xor(Nc',xor(GM',CSSig1))) /\secret(PW',sec2,{CS,Ci})
   /\secret({GM,CM},sec3,{CS,Ci}) /\secret({CSSig1,CSSig2},sec4,{CS,Ci})
   /\secret({TmpPW',OTPi},sec5,{CS,Ci}) /\secret({UPc,MPc,UPcs,MPcs},sec6,{CS,Ci})

% Authentication request from CS to AS
   /\State':= 1 /\Ncs':=new() /\TScs':=new() /\CSSig3':= H(TScs'.Ncs'.UPcs.MPcs)
   /\TmpPW':=xor(PW,xor(Ncs',CSSig3')) /\SND({TScs'.Ncs'.UPcs.MPcs.CSSig3'.TmpPW'}_KASpu)
   /\secret(PW,sec7,{CS,AS}) /\secret({CSSig3},sec8,{CS,AS})
   /\secret({TmpPW'},sec9,{CS,AS}) /\secret({UPcs,MPcs,UPas,MPas},sec10,{CS,AS})

% Authentication response from AS to CS
2. State = 1
    /\RCV({TSas'.Nas'.Ncs.UPas'.MPas'.H(TSas'.Nas'.UPas'.MPas')}_KCSpu)=|>
% Authentication response to Ci
   State':= 2 /\Ncs':=new() /\TScs':=new()
   /\SND({TScs'.Ncs'.Nc.UPcs.MPcs.H(TScs'.Ncs'.UPcs.MPcs)}_KCpu)
   % CS prove his identity
   /\witness(CS,Ci,cs_ci_auth1,{Nc,TScs'}) /\witness(CS,Ci,cs_ci_auth5,{TmpPW,OTPi})
   /\witness(CS,AS,cs_as_auth3,{Nas',TScs'})
   /\request(CS,AS,as_cs_auth4,{Ncs',TSas}) /\request(CS,AS,as_cs_auth6,{PW})

3. State  = 2
   /\RCV({Ncs}_KCSpu) =|>
% CS checks that the received nonce and timestamp correct
   State':= 3 /\request(CS,Ci,ci_cs_auth2,{Ncs,TSc})
end role
```

Figure 7.20: CS role of RAMHU in HLPSL

```
role attributesServer(AS,Ci,CS:agent, KASpu,KCSpu:public_key, H:hash_func, SND,RCV:channel(dy))
played_by AS def=
local
     State:nat,
     UID,MID,PW:message, TmpPW:message, ASSig1,ASSig2:text,TScs,Ncs,TSas,Nas:text,
     UPcs,MPcs,UPas,MPas:message, GM,CM:message
const
     sec1,sec7,sec8,sec9,sec10,cs_as_auth3,as_cs_auth4,as_cs_auth6:protocol_id
init
     State:= 0
transition
1. State = 0
   /\RCV({TScs'.Ncs'.UPcs'.MPcs'.H(TScs'.Ncs'.UPcs'.MPcs').
         TmpPW'}_KCSpu)=|>ASSig1':= H(TScs'.Ncs'.UPcs.MPcs)
   /\PW':=xor(TmpPW,xor(Ncs',ASSig1))/\secret({UID,MID,PW},sec1,{AS,Ci}) /\secret(PW,sec7,{AS,CS})
   /\secret({ASSig1},sec8,{AS,CS}) /\secret({TmpPW'},sec9,{AS,CS})
   /\secret({UPcs,MPcs,UPas,MPas},sec10,{AS,CS}) /\State':= 1 /\Nas':=new() /\TSas':=new()
   /\ASSig2':=H(TSas'.Nas'.UPas.MPas) /\SND({TSas'.Nas'.Ncs.UPas.MPas.ASSig2'}_KCSpu)
   /\witness(AS,CS,as_cs_auth4,{Ncs',TSas'}) /\witness(AS,CS,as_cs_auth6,{PW'})
   /\request(AS,CS,cs_as_auth3,{Nas',TScs'})
end role
```

Figure 7.21: AS role of RAMHU in HLPSL

```
role session(Ci,CS,AS:agent, KCpu,KCSpu,KASpu:public_key, H:hash_func, UID,MID,OTPi,PW,GM,
             CM:message)
def=
local
     SndC,RcvC,SndCS,RcvCS,SndAS,RcvAS:channel(dy)
composition
     clienti(Ci,CS,AS,KCpu,KCSpu,H,UID,MID,OTPi,PW,GM,CM,SndC,RcvC)
     /\centralServer(Ci,CS,AS,KCpu,KCSpu,KASpu,H,SndCS,RcvCS)
     /\attributesServer(AS,Ci,CS,KASpu,KCSpu,H,SndAS,RcvAS)
end role

role environment()
def=
const
     ci,cs,as,i_ci,i_cs,i_as:agent, kCpu,kCSpu,kASpu,ki:public_key, uid,mid,otp,pw,gm,cm:message,
     h:hash_func, sec1,sec2,sec3,sec4,sec5,sec6,sec7,sec8,sec9,sec10,
     ci_cs_auth2,cs_ci_auth1,cs_ci_auth5,cs_as_auth3,as_cs_auth4,as_cs_auth6:protocol_id
     intruder_knowledge={ci,cs,as,i_ci,i_cs,i_as,kCpu,kCSpu,kASpu,ki}
composition
     session(ci,cs,as,kCpu,kCSpu,kASpu,h,uid,mid,otp,pw,gm,cm)
     % Check replay attack
     /\session(ci,cs,as,kCpu,kCSpu,kASpu,h,uid,mid,otp,pw,gm,cm)
     % Check MITM attack
     /\session(cs,ci,as,kCSpu,kCpu,kASpu,h,uid,mid,otp,pw,gm,cm)
     % Check impersonate Ci
     /\session(i,cs,as,ki,kCSpu,kASpu,h,uid,mid,otp,pw,gm,cm)
     % Chekc impersonate CS
     /\session(ci,i,as,kCpu,ki,kASpu,h,uid,mid,otp,pw,gm,cm)
     % Check impersonate AS
     /\session(ci,cs,i,kCpu,kCSpu,ki,h,uid,mid,otp,pw,gm,cm)
end role

goal
 secrecy_of sec1,sec2,sec3,sec4,sec5,sec6,sec7,sec8,sec9,sec10
 authentication_on cs_ci_auth1,ci_cs_auth2,cs_as_auth3,as_cs_auth4,cs_ci_auth5,as_cs_auth6
end goal

environment()
```

Figure 7.22: Session, environment, and goal roles of RAMHU in HLPSL

```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/RAMHU.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
 parseTime: 0.00s
 searchTime: 54.60s
 visitedNodes: 3636 nodes
 depth: 11 plies
```

Figure 7.23: Simulation result of RAMHU using OFMC backend

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/RAMHU.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed    : 324 states
  Reachable   : 64 states
  Translation: 0.52 seconds
  Computation: 0.42 seconds
```

Figure 7.24: Simulation result of RAMHU using CL-AtSe backend

The authentication schemes in He & Zeadally (2015), Giri et al. (2015)

suffer from a leak of authentication information when transferring user requests between network entities. Compared with RAMHU, all real user information is not transmitted through the network exchanges and therefore does not leak information. Farash et al. (2016) and Jiang et al. (2016) designed authentication schemes but theses schemes lacked the management and protection of user information/data on the server. RAMHU stores information on the AS and uses signatures and multi-pseudonyms to disguise user information on the AS. It also uses CS as a gateway to check requests before sending them to the AS. In Kumar et al. (2016), the authentication scheme suffers from internal attacks. If the intruder uses a sniffing program to steal user information such as MAC addresses, he/she becomes authenticated in the network as a legitimate user. In comparison, RAMHU uses CM and GM to prevent internal attacks.

Li et al. (2016) and Das et al. (2017) used a symmetric algorithm that relies on a single key and therefore runs the risk of key detection. In contrast, RAMHU relies on an asymmetric algorithm that supports scalability. Rajput, Abbas, Wang, Eun & Oh (2016) proposed an authentication scheme that suffered greatly from impersonation attacks. An intruder can use his/her device as a server and deceive users. RAMHU uses robust mutual authentication to prevent impersonation attacks. In Chandrakar & Om (2017), the authentication scheme suffers from collision and preimage attacks, and it handles real user information when requests are transferred to the network. Compared to RAMHU, it uses PHOTON to prevent collision and preimage attacks as well as using multi-pseudonyms to prevent the exchange of real information between network entities.

The schemes in Nizzi et al. (2019) and (El-Tawab et al. 2019) suffered from several problems, such as spoofing the coordinator/server by shuffle (AShA)/randomisation of fake addresses sent from intruders, key detection, breaking the integration process/vulnerability to preimage/second preimage attacks, and a single technique that does not provide security for online devices. RAMHU does not suffer from these problems because it uses CM to prevent MACs change, the public key algorithm to prevent key breaking, PHOTON to provide signature principles, and it uses a set of techniques to protect authentication processes. Table 7.5 shows a comparison between RAMHU and existing authentication schemes in various attacks resistance.

Table 7.5: Comparison of resistance in repelling the various threats between RAMHU and other authentication schemes

Attack	He & Zeadally (2015)	Giri et al. (2015)	Li et al. (2016)	Farash et al. (2016)	Kumar et al. (2016)	Jiang et al. (2016)	Rajput, Abbas, Wang, Eun & Oh (2016)	Das et al. (2017)	Chandrakar & Om (2017)	Nizzi et al. (2019)	El-Tawab et al. (2019)	**RAMHU scheme**
Privileged-insider		✓	✓			✓		✓	✓			✓
Stolen device/application			✓			✓		✓	✓			✓
Replay	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓
MITM	✓			✓		✓		✓		✓	✓	✓
Guessing		✓	✓			✓		✓	✓	✓	✓	✓
Client impersonation	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓
Server impersonation	✓	✓	✓	✓	✓	✓		✓	✓			✓
DoS		✓	✓				✓			✓		✓
Change password		✓	✓			✓		✓	✓			✓
Eavesdropping	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓
Traceability	✓		✓	✓	✓	✓	✓		✓		✓	✓
Revocation						✓	✓					✓
Verifier	✓		✓			✓		✓				✓
Leakage	✓			✓		✓	✓	✓	✓	✓		✓

7.3.2 Performance Analysis

In this section, the theoretical and experimental performance analysis and performance comparison with existing related works are presented to examine the computation and communication processes of RAMHU in improving the performance of the authentication processes.

- **Theoretical Performance Analysis**

  The authentication scheme should perform lightweight processes to support the reliable communication of users in healthcare applications. Authentication is the first process that allows users to be recognised as legitimate users of the network services. If the authentication process is vulnerable for attacks, network performance will be greatly affected by the fact that the servers will perform additional complex computations.

  RAMHU uses algorithms and techniques to support performance-efficient authentication. RAMHU uses the ECIES algorithm to encrypt user information only. This algorithm performs lightweight operations and produces small keys compared to public key cryptographic algorithms as described in Chapter 2, Table 2.2. In addition, RAMHU uses the hash function (PHOTON) that performs hash operations compared to lightweight hash functions as described in Appendix B, Table B.1. The use of robust security algorithms against attacks such as ECIES and PHOTON is a major cause of performance stability. In RAMHU, new users use OTP only once at the registration phase. They do not need to use this technique in subsequent times. This technique reduces the burden on the server processor to perform complex computations. If the attacker tries to connect to the network with a fake or previously used OTP, the server would reject this request early without having to test other authentication techniques. This contributes significantly to reducing the depletion of server capabilities.

  Furthermore, RAMHU relies on multi-pseudonyms that perform lightweight operations in protecting user information rather than using k-anonymity, which consumes the server's time and power needed to search and explore real information for legitimate users. Moreover, RAMHU supports the authentication process via the MAC address technique. This technique is extremely efficient in verifying legitimate devices connected to the network. The other solutions also rely on IP to authenticate users' devices. IP does not prove the identity of the legitimate device when connected to the network because it is the address of software and not hardware. Also, if the attacker uses a legitimate IP, he/she can perform malicious threats on the EHR repository that will significantly affect network performance. RAMHU uses lightweight operations (GM and CM) as part of the authentication processes in the examination of legitimate devices. Therefore, the aforementioned RAMHU features enable it to perform lightweight and efficient authentication.

- **Experimental Performance Analysis**
  In this section, we show RAMHU's performance in healthcare applications. We provide tests on the hash function (PHOTON 256-bit) and encryption algorithm (ECIES 256-bit). In addition, communication costs (storage overheads) and computation (execution time) are calculated to extract RAMHU's performance. Applications codes (C_i, CS and AS) are written in Java Programming Language. Also, RAMHU's results are implemented on Ubuntu 16.04 LTS, processor Intel Core i5 2.6GHz, OS type 32-bit, Memory 4 GiB and disk 32.0 GB.

  To compute the cost of communication, we compute the bits of all security parameters. Hash's MD (PHOTON) is 256-bit and the ECIES encryption key is 256-bit. The number of messages transmitted through the authentication and login protocol is 4, the password update protocol is 2 and the revocation protocol is 3. On the C_i side, it produces a 6-line file (TS_{C_i}=8, GM=17, C_iSig_1=64, $UP_{C_i}^{CS}||MP_{C_i}^{CS}$=5, N_{C_i}=8, $tmpPW_i$=64 and C_iSig_2=64). The total size of the parameters is 230 bytes. Then C_i converts the plaintext file to an encrypt file with a size from 210 bytes to 255 bytes (the minimum value is 1680 bits) and the number of bits inside the encryption file is 253 bits. When C_i receives the authentication response from CS, the size of the decryption file is from 93 bytes to 99 bytes (the minimum value is 744 bits). The total size of encryption and decryption files in C_i is 2424 bits.
  On the CS side, it produces a 4-line file (TS_{CS}=8, $UP_{CS}^{AS}||MP_{CS}^{AS}$=11, N_{CS}=8, $tmpPW_i$=64 and $CSSig_3$=64). The total size of the parameters is 155 bytes. Then CS converts the plaintext file to an encrypt file with a size of 133 bytes to 178 bytes (the minimum value is 1064 bits) and the number of bits within the encryption file is 176 bits. When CS receives the authentication response from AS, the size of the decryption file is from 99 bytes to 125 bytes (the minimum value is 792 bits). The total size of encryption and decryption files in CS is 1856 bits.
  On the AS side, it produces a 3-line file (TS_{AS}=8, $UP_{AS}^{CS}||MP_{AS}^{CS}$=17, N_{AS}=8 and $ASSig_2$=64). The total size of the parameters is 97 bytes. Then AS converts the plaintext file to an encrypted file with a size of 119 bytes to

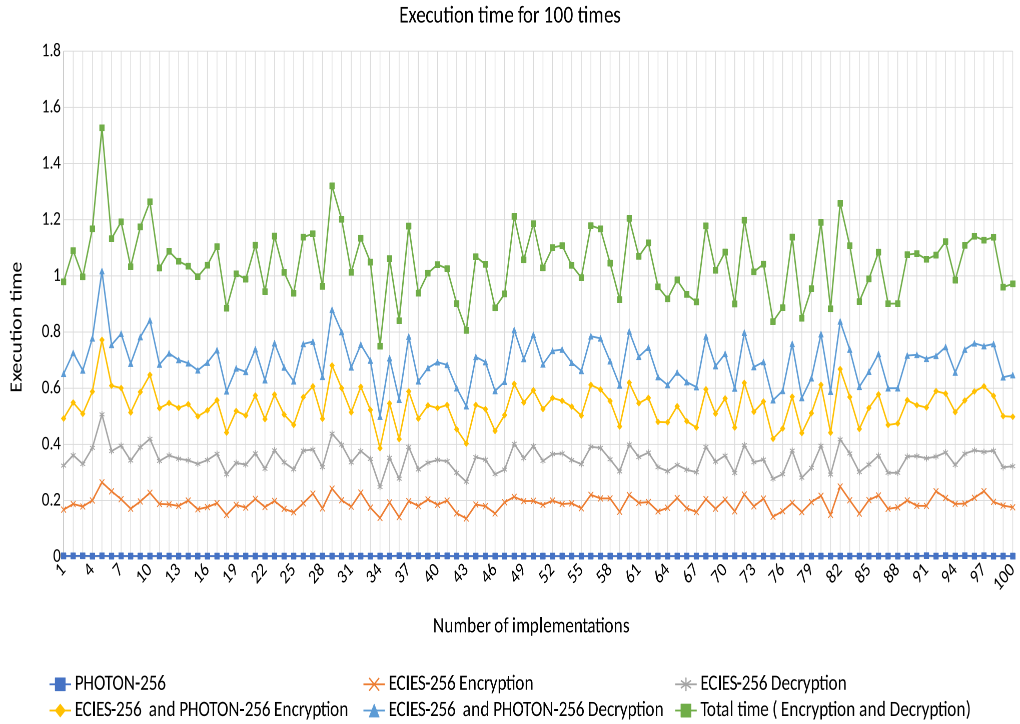


Figure 7.25: Implementations of PHOTON 256-bit and ECIES 256-bit

125 bytes (the minimum value is 952 bits) and the number of bits within the encryption file is 212 bits. When AS receives the authentication response from CS, the decryption file size is from 158 bytes to 164 bytes (the minimum value is 1264 bits). The total size of encryption and decryption files in AS is 2216 bits. Also, storage complexity in RAMHU is 8352 bits.

To compute the cost of computation, we obtained preliminary results about the running time for PHOTON-256, ECIES-256 encryption and ECIES-256 decryption (100 times, information size of 93 bytes) as shown in Figure 7.25. We noted that the execution time of PHOTON is faster than hash algorithms such as SHA-1 and SHA-256 depending on the results in Latinov (2018). The minimum execution time for the PHOTON 256-bit hash function is 0.002358 ms, ECIES 256-bit encryption is 0.133672 ms, ECIES 256-bit decryption is 0.110136 ms, ECIES 256-bit encryption and PHOTON 256-bit is 0.136186 ms and ECIES 256-bit decryption and PHOTON 256-bit is 0.112521 ms. Also, time complexity in RAMHU is 0.994828 ms.

- **Performance Comparison**

  In this section, we compare the superiority of RAMHU over the authentication schemes in the performance side. Although the implementation environment for authentication schemes varies, we make some comparisons that demonstrate the superiority of RAMHU's performance over existing schemes.

The authentication scheme in He & Zeadally (2015) relies on ECC 512-bit plus AES to generate session key, while RAMHU relies on ECC 256-bit which is NIST certified and sufficient to encrypt and secure the authentication request. As a result, RAMHU uses ECC with the best performance and has no need to use symmetric encryption that increases computation expenses. Schemes in Farash et al. (2016) and Jiang et al. (2016) are based on a single server performing all complex operations with users' devices. These processes consume server resources while RAMHU divides computations on servers CS and AS which dramatically improves performance. In addition, RAMHU performs the lowest number of hashes compared to these schemes as shown in Table 7.6 (T_h is the time complexity for hash function, T_s is the time complexity for symmetric encryption and T_H is the time complexity for bio-hash function). This makes RAMHU's performance suitable for the healthcare application environment.

Compared to RAMHU, both schemes in Jiang et al. (2016) and Das et al. (2017) use a fuzzy extractor that requires additional operations to consume an execution time of 0.442s, while RAMHU does not need these additional operations. Furthermore, the scheme in Giri et al. (2015) is based on RSA 1024-bit which is expensive in encryption and decryption while RAMHU relies on ECIES 256-bit which uses small keys and is suitable for high-performance security. The CACPPA scheme in Rajput, Abbas, Wang, Eun & Oh (2016) suffers from storage overheads. It requires 76 bytes per pseudonym as well as other storage requirements such as keys, encryption by ECIES 256-bit and signature by ECDSA 256-bit. With increasing numbers of users, this scheme will be extremely costly for server memory (communication and computational overheads). RAMHU uses only 5-17 bytes per pseudonym which saves significant memory storage. Moreover, the scheme in Chandrakar & Om (2017) stores the same user information on multiple servers which consumes storage resources as well as the computation cost to store this information. RAMHU does not deplete storage sources by repeating the same information on servers.

The scheme in El-Tawab et al. (2019) suffers from costly processes due to the use of SHA-256, while RAMHU uses PHOTON which has lightweight processes. Additionally, impersonation attacks of legitimate MACs on schemes

Table 7.6: Comparison of computation cost between RAMHU and existing authentication schemes

Scheme	Hashes number on user side	Hashes number on server side	Total	Hash type	Running time in ms	No. of messages	No. of bits
He & Zeadally (2015)	$2T_h + 2T_s$	$1T_h + 4T_s$	$3T_h + 6T_s$	Standard	0.4 +0.4	6	-
Giri et al. (2015)	$5T_h$	$4T_h$	$9T_h$	Standard	-	4	1600
Jiang et al. (2016)	$7T_h$	$7T_h$	$14T_h$	Standard	-	4	-
Kumar et al. (2016)	$5T_h$	$5T_h$	$10T_h$	Standard	-	3	-
Li et al. (2016)	$6T_h + 2T_s$	$7T_h + 6T_s$	$13T_h + 8T_s$	Standard	0.0001+0.442	4	768
Das et al. (2017)	$3T_h$	$7T_h$	$10T_h$	Standard	0.0001+0.442	4	768
Chandrakar & Om (2017)	$12T_h$+$1T_H$	$7T_h$	$19T_h$+$1T_H$	Standard	0.0005+0.02102	9	2240
RAMHU	$3T_h$	$5T_h$	$8T_h$	Lightweight	0.002358	4	253

in Nizzi et al. (2019) and El-Tawab et al. (2019) cause low network performance due to increased network-connected fake devices. RAMHU prevents the use of legitimate MACs addresses in more than one device by using the CM technique and thus maintains network performance.

7.4 Analysis of PAX Scheme

In this section, we discuss user scenarios, security and performance analysis and compare PAX with existing search and demonstrate PAX's ability to protect patient data during security and privacy implementation.

7.4.1 Direct and Indirect Users Scenarios in PAX

There are four case scenarios in PAX that involve obtaining access to patients' medical and/or health records in the EHR/EMR repository. Here we present how the user can access patient data while securing their privacy. To provide user scenarios, we introduce a number of EHR users to the PAX system, as shown in Figure 7.26.

Assume we have three patients, Sara, John, and Rose, who suffer from cancer, dementia, and diabetes respectively. Each disease requires a different level of care. For instance, a patient suffering from dementia needs a family member who assists with all of the patient's tasks and is able to access all of the patient's data. We assume that Julia is one of John's relatives. Also, there is a group of healthcare providers, including Simon, Adam, Hawa, and Abraham, who want access to the patients' medical records. These users can have different roles; for example, Adam may have the roles of advisor and doctor, and Abraham may be a doctor and an emergency doctor. Different user roles can be a major reason for breaching the privacy of medical records. Users such as patients (Sara and Rose) and the physician

(Simon) need direct authorisation to EHR data because of regular and ongoing requests to access the repository. For example, Simon is the general practitioner (GP) for Sara and needs to access her data every day or even more than once a day. Under the PAX system, only Sara and Simon can access Sara's data, as shown in Figure 7.27.

1. The first scenario use (advisor): Simon needs a consultant (such as Adam) to diagnose Sara's disease or to submit treatment suggestions (after receiving Sara's consent to seek specialist advice). Adam is not associated with Sara permanently and continuously and does not need Sara's personal information; he only needs certain details of the patient data and medical reports. Therefore, in PAX, Adam needs to enter his name (Adam), the name of the doctor (Simon), and Sara's pseudonym to access Sara's data; he does not need to know Sara's real attributes. Figure 7.27 shows Sara's data, which can be obtained by Simon and Adam. We note, from Figure 7.27, that the data received does not contain any of Sara's attributes, and Adam does not use any real attributes for Sara, which means that PAX provides a high level of security and privacy that can prevent external and internal attacks

2. The second scenario use (relative of a patient): Because the patient (John) suffers from dementia, he is unable to perform his duties. John needs a family helper (such as Julia) to access his medical data without misuse or to bypass these privileges to other medical records. Julia needs a request that contains her Sigs and John's pseudonym to be considered a legitimate user in the system but is not authorised to access John's data until the CS and AS complete the third authorisation protocol with the Shamir scheme

3. The third scenario use (researcher): Hawa is a researcher and tries to access the server's repository to use EHR in evaluating a medical study to develop a disease treatment. The researcher needs access to medical records sporadically and temporarily. The researcher is not associated with a particular patient and needs access to a set of patients data. Also, this indirect user does not need access to patient attributes. Figure 7.28 shows a set of medical records obtained by Hawa in the case of authorisation without using any of the patients' real attributes

4. The fourth scenario and the last use (emergency doctor): When Rose's health deteriorates significantly and suddenly, her doctor is not available for some reason. Rose needs an emergency doctor to assess and treat her condition quickly (e.g., Abraham). The emergency doctor needs to access Rose's data without accessing personal information. In an emergency, access to a patient's

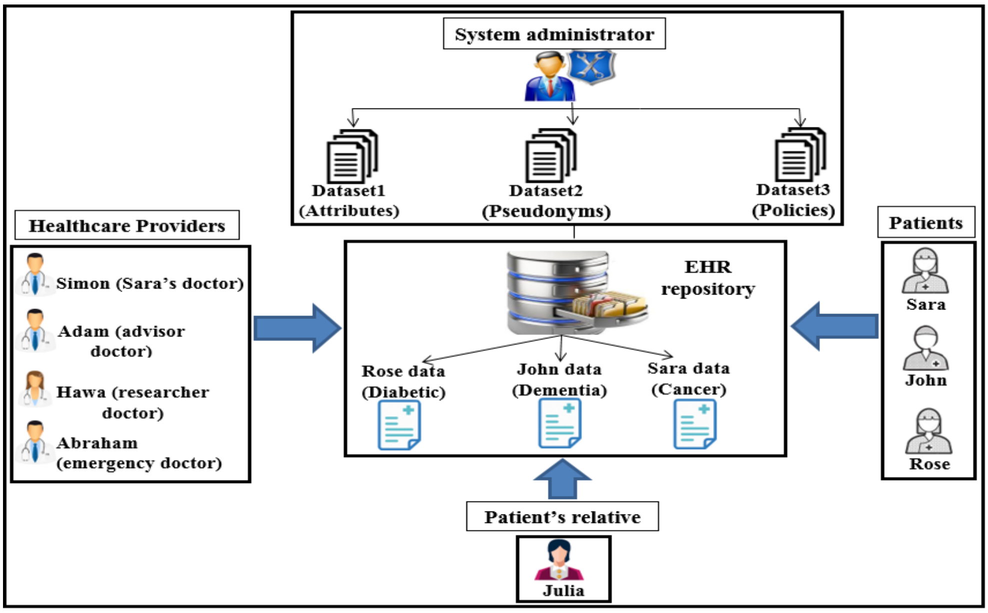


Figure 7.26: Users' scenarios in PAX

```
************************************
*           Patient's Data         *
************************************
```

Pseudonym: RN_{optm} & UN_{optm}

```
check1:    report:    status:    date:
check2:    report:    status:    date:
check3:    report:    status:    date:
check4:    report:    status:    date:
check5:    report:    status:    date:
  .          .          .          .
  .          .          .          .
  .          .          .          .
```

Figure 7.27: Part of Sarah's data

data does not require the patient's consent. Abraham should not know any secrets healthcare providers have used to authorise access to Rose's data.

PAX provides security and privacy for all previous scenarios; indirect users cannot access the patient's personal information because it is separate and completely hidden from the data. As a result, the user can retrieve this data to improve healthcare without penetrating the repository in DS.

7.4.2 Possible Attacks on PAX Scheme

Applying privacy to medical records (EHR) requires the use of access models in the authorisation of users. Integrating RBAC and ABAC gives more powerful features to PAX users. The result is an access control model based on roles and attributes that handle user requests at coarse-grained and fine-grained levels. To increase security and privacy in the authorisation model, we have added a set of mechanisms

```
*******************************
*      Patients' DataSet      *
*******************************

----------------------------------------------
No Check   Report   Status  time     date
----------------------------------------------
1  check3  Report3  still   23:21:33  2017-09-05
2  check1  Report1  ok      14:36:45  2017-09-08
3  check2  Report2  normal  17:09:57  2017-09-08
5  check3  Report3  still   17:10:09  2017-09-10
6  check2  Report2  normal  12:28:20  2017-09-11
.    .       .        .        .          .
.    .       .        .        .          .
.    .       .        .        .          .
```

Figure 7.28: Part of a group of patients' medical records

to hide and separate personal information about data. The PAX system ensures that legitimate users access their specific data and, on the other hand, the privacy of medical records is maintained. Any healthcare system should support the basic security features of confidentiality, integrity, and availability (C.I.A.) (Neubauer & Heurix 2011). There are a range of attacks that pose a serious risk to any healthcare system. PAX's security mechanisms act as countermeasures against known attacks.

- **Proposition 1 – Availability attack**
  I cannot efficiently destroy PAX scheme by availability attacks.
  Proof
  The server is vulnerable to denial of service (DOS) attacks that are intended to disable the service. In PAX, the indirect user creates a random Sig based on SS_s provided by healthcare providers. The attacker cannot use the same SS_s because the CS and AS will ignore the request. The abundance of medical records is critical to healthcare providers' flexible access. Therefore, supporting robustness in any healthcare system depends on preventing DOS attacks. Although the PAX system limits the risk of DOS attacks, it does not do so fully because the attacker can still send requests without penetrating the patient's personal information and data.

- **Proposition 2 – Data and policies datasets attacks**
  II or EI cannot penetrate data and policies datasets in EHR repository.
  Proof
  The data in the single server is considered an attractive target for attackers. Also, policies contain the attributes and roles of users, which can assist attackers in carrying out an attack to recognise and access patient data. In PAX, even if the attacker obtains a patient's data, the data would not be useful because both the stored and movable data would have pseudonym. In addition, the data is stored (on DS) separately from policies (on AS). Furthermore, PAX policies are associated with pseudonyms and anonymity

(both CS and DS do not have real attributes datasets for users), attackers are prevented from revealing subjects' and objects' attributes.

- **Proposition 3 – MITM attack**
  I cannot apply modification attacks to change authorisation requests for users.
  Proof
  User requests from clients to the server in PAX are fully protected from modification. PAX uses random nonces and Sigs to detect a changing operation (MITM) by intruders.

- **Proposition 4 – Replay attack**
  I cannot resend authorisation requests later (replay attacks) to gain access to the network.
  Proof
  The intruder cannot resend authorisation requests to the network later because PAX produces a new timestamp (TS_C, TS_{CS}, TS_{AS}, and TS_{DS}) between PAX's entities.

- **Proposition 5 – Unauthorised access attack**
  I cannot perform unauthorised access attacks against PAX scheme.
  Proof
  User access to a repository depends on authorisation policies. We use XACML v3.0 to create user policies. Integrating RBAC and ABAC into XACML prevents unauthorised users from accessing patient data.

- **Proposition 6 – Traffic analysis attack**
  EI cannot use traffic analysis attacks to detect user information.
  Proof
  To perform this attack, the hacker must analyse either the requests or the data moving between the source and the target. In PAX, if we assume that the attacker has some attributes (such as the name) and expects a specific patient, the attacker cannot use a keyword (name) and analyse it with multiple requests or medical records, even if it is the same user, to reveal its real attributes; the attacker cannot identify this data for a particular patient. Using pseudonym and anonymity prevents attackers from tracking traffic. For example, if advisor1 and advisor2 want patient1 data, the generated requests will be different. This prevents the parsing of requests.

- **Proposition 7 – Impersonation attack**
  I cannot impersonate PAX's entities (impersonation attacks).
  Proof

The intruder cannot impersonate PAX's entities (C_i, CS, AS and DS) because PAX uses secret nonces (SN_C, SN_{CS} and SN_{AS}) and secret Sigs among entities to support mutual authentication and prevent impersonation attacks.

- **Proposition 8 – Timing attack**
  EI cannot use timing attacks to penetrate security parameters in signatures.
  Proof
  This attack exploits the security procedure while calculating the time period for security operations (such as encryption and signing). PAX prevents these attacks because when the attacker gets multiple requests for the same user, the attacker will find that these requests contain different Sigs, and the attacker does not have the parameters to generate the Sig. In addition, ECDSA's Sigs with 256-bit is resistant to timing attacks.

1. **Experimental Security Analysis**
   In this section, we will use the AVISPA tool to analyse PAX authorisation procedures. Then we will present the results of PAX analysis in repelling known attacks.

   - **PAX Scheme with AVISPA**
     As we have described the AVISPA in Section 7.1.1, PAX consists of four core (essential) roles: client (C_i), centralServer (CS), attributesServer (AS) and dataServer (DS). Also, there are the supporting roles such as session, and environment, goal specification section. Essential roles include a transition section (to specify the sequence of communication operations in network framework). Supporting roles include a composition section (to specify the linking of sessions and roles).

     PAX depends on asymmetric cryptography using ECDSA with public keys (KC_{pu}, KCS_{pu}, KAS_{pu} and KDS_{pu}) to perform security requirements (integrity, authentication and non-repudiation). Moreover, PAX applies nonces (SN_C, SN_{CS}, SN_{AS} and SN_{DS}) to support anonymity and timestamps (TS_C, TS_{CS}, TS_{AS} and TS_{DS}) to support freshness. The authorisation process for indirect users is illustrated by the HLPSL scripts in Figures 7.30, 7.31, 7.32 and 7.33. Each role consists of the number of transitions, the receiving process (RCV), the sending process (SND), the sender's claim process of fresh value and correct (witness), the validation process in receiver for the sender's claim (request), the process of creating a fresh value for the nonce and timestamp (new) and the use of the private key (_inv) in PAX's entities.

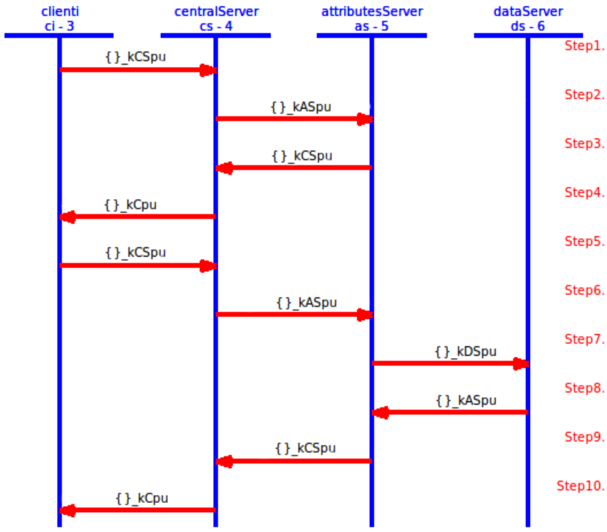


Figure 7.29: PAX's framework in AVISPA

At first, C_i receives the start signal as in Figure 7.30, then the SND and RCV operations continue until the authorisation process is completed as in Figure 7.29.
Figure 7.34 shows the roles of session, environment, and goal section. In the session role, a composition process has been performed for the four roles (*clienti*, *centralServer*, *attributeServer* and *dataServer*) and specifies the send and receive channels in the Dolev-Yao model. In the environment role, the PP, the goals specified in the goal section, and the known information for the intruder (*intruder_knowledge*) have been defined. In this role, one or more sessions are composed, and we tested our scheme with sessions for replay, MITM, and impersonating entity attacks. We assumed that an intruder (I) creates a public key (ki) and has knowledge of the public keys ($kCpu$, $kCSpu$, and $kASpu$) of PAX's entities in the network. Intruder attempts to resend legitimate user requests later, intercepts/modifies these requests, or impersonates the connecting entities using i constant rather than ci, cs, as and ds.

The results display that these attacks cannot penetrate the security goals in our scheme. Goal section describes verified goals in PAX, and provides 10 goals of secrecy (such as S_ID, O_ID, S_R and O_R representing the first secret (sec1) and known only for both ci and cs) and eight goals of authentication (such as $UNspm$, $UNopm$ and $TScs$ representing the first authentication between ci and cs).

```
role clienti(Ci,CS:agent, KCpu,KCSpu:public_key,H:hash_func, S_ID,O_ID,S_R,O_R:message,SND,
RCV:channel(dy))
played_by Ci def=
local
     State:nat,
     TSc,TScs,TSctm,TScstm,Nc,SNc,SNctm:text,
     CiS1,CiS2,CiS3,CiS4,CiS5,CiS6,CiS1tm,CiS2tm,CiS4tm,CiS6tm,CSS2tm,CSS3tm,CSS5:text,
     SP,OP,UNspl,UNspm,UNsph,UNopl,UNopm,UNoph,RNspl,RNspm,RNsph,RNopl,RNopm
     ,RNoph:text, RNspn,RNopn,UNspn,UNopn:text,AS,DS:agent, MS,SS,SSs,Decision,Data:text
const
     sec1,sec2,sec3,sec4,sec5,sec6,auth1,auth2:protocol_id
init
     State := 0
transition
1.State=0
  /\RCV(start) =|> State':=1 /\Nc':=new() /\SNc':=new() /\TSc':=new()
  /\CiS1':={H(RNspm.UNspm.Nc'.TSc')}_inv(KCpu) /\CiS2':={H(RNopm.UNopm.Nc'.TSc')}_inv(KCpu)
  /\CiS3':={H(SNc')}_inv(KCpu)/\ CiS4':={H(RNoph.UNoph.TSc')}_inv(KCpu)
  /\CiS1tm':=xor(CiS1',CiS3')/\CiS2tm':=xor(CiS2',CiS3')
  /\TSctm':=xor(TSc',xor(SNc',xor(RNspm,UNspm)))
  /\SNctm':=xor(UNspm,xor(UNopm,xor(SNc',xor(CiS4',CiS1tm'))))
  /\CiS4tm':=xor(CiS4',xor(UNspm,xor(UNopm,CiS2tm')))
  /\RNspn':=xor(RNspl,xor(CiS1tm',SNctm')) /\RNopn':=xor(RNopl,xor(CiS2tm',SNctm'))
  /\UNspn':=xor(UNspl,xor(CiS1tm',SNctm')) /\UNopn':=xor(UNopl,xor(CiS2tm',SNctm'))
% Ci sends XACML's request to CS
  /\SND(CS.CiS1tm'.RNspn'.UNspn'.Nc'.TSctm'.SNctm'
  .CiS2tm'.RNopn'.UNopn'.Nc'.CiS4tm') /\secret({S_ID,O_ID,S_R,O_R},sec1,{Ci,AS})
  /\secret({SNc',CiS3',TSc'},sec2,{Ci,CS}) /\secret({CiS1',CiS2'},sec3,{Ci,CS,AS})
  /\secret({RNspm,UNspm,RNopm,UNopm},sec4,{Ci,CS})

% Ci receives first authorisation response from CS
2. State  = 1
   /\RCV(Ci.CSS2tm'.UNspn'.TScstm')=|> State':= 2 /\UNspl':=xor(UNspn',xor(CSS2tm',TScstm'))
   /\TScs':=xor(TScstm',xor(SNc,UNopm)) /\CiS6':={H(SP)}_inv(KCpu)
   /\SSs':=xor(CiS1,xor(CiS3,xor(CSS2tm',xor(CiS4,CiS6'))))
   /\MS':= {(SS.SSs')} /\CiS6tm':=xor(CiS6',xor(CiS3,MS'))
   /\secret({OP,CiS4},sec5,{Ci,AS,DS}) /\secret({SP,CiS6,MS',SS},sec6,{Ci,AS})
   /\TSc':=new()/\TSctm':=xor(TSc',xor(SNc,UNspm)) /\UNspn':=xor(UNspl',xor(CiS6tm',TSctm'))
% Ci sends Shamir's response to CS
   /\SND(CS.CiS6tm'.UNspn'.TSctm') /\witness(Ci,CS,auth1,{UNspm,UNopm,TScs})

% Ci receives decision & data from CS
3.State=2
  /\RCV(Ci.CSS2tm'.CSS3tm'.UNspn'.TScstm'.Decision.Data)=|> State':=3
  /\UNspl':=xor(UNspn',xor(CSS2tm',TScstm')) /\TScs':=xor(TScstm',xor(SNc,UNopm))
  /\CSS5':=xor(CiS1',xor(CiS3',xor(CSS2tm',CiS4'))) /\CiS5':={H(Data)}_inv(KCSpu)
  /\CiS1':=xor(CSS2tm',xor(CiS3,xor(CiS5',CiS4))) /\CiS2':=xor(CSS3tm',xor(CiS3,xor(CiS5',CiS4)))
  /\CiS3':=xor(CiS2',xor(CSS2tm',xor(CiS5',CiS4)))
  /\CiS4':=xor(CiS1',xor(CiS3',xor(CiS5',CSS2tm'))) /\request(Ci,CS,auth2,{SNc,CiS3,CiS4,TSc})
end role
```

Figure 7.30: C_i role of PAX in HLPSL

- **Simulation Result**

  Figure 7.35 displays the simulation result with the CL-AtSe backend, PAX clearly and accurately achieves the SAFE result in the SUMMARY section, bounded number of sessions in the DETAILS section, the goals of the scheme achieved (as_specified) in the GOAL section as well as statistical numbers such as time, number of nodes, and analysed states in the STATISTICS section. It has proven that the PAX is capable of preventing passive and active attacks such as replay, MITM, and impersonating, and that the goals of the scheme in Figure 7.34 prevents the violation of legitimate users' information in the network authorisation.

```
role centralServer(CS,Ci,AS:agent, KCSpu,KCpu,KASpu:public_key, H:hash_func, SND,RCV:channel(dy))
played_by CS def=
local
     State:nat,TSc,TScs,TSas,TSctm,TScstm,TSastm,Nc,Ncs,SNc,SNcs,SNctm,SNcstm:text, CSS1,CSS2,
     CSS3,CSS4,CSS5,CSS2tm,CSS3tm,CiS1,CiS2,CiS4,CiS1tm,CiS2tm,CiS4tm,CiS6tm:text,
     ASS2,ASS3,ASS7,ASS2tm,ASS3tm:text, Decision,Data:text,
     UNspl,UNspm,UNopl,UNopm,RNspl,RNspm,RNopl,RNopm, RNspn,RNopn,UNspn,UNopn:text
const
     sec2,sec3,sec4,sec7,auth1,auth2,auth3,auth4:protocol_id
init
     State := 0
transition
% CS receives XACML's request from Ci
1.State=0
   /\RCV(CS.CiS1tm'.RNspn'.UNspn'.Nc'.TSctm'.SNctm'.CiS2tm'.RNopn'.UNopn'.Nc'.CiS4tm')=|>
   State':=1/\RNspl':=xor(RNspn',xor(CiS1tm',SNctm'))
  /\RNopl':=xor(RNopn',xor(CiS2tm',SNctm')) /\UNspl':=xor(UNspn',xor(CiS1tm',SNctm'))
  /\UNopl':=xor(UNopn',xor(CiS2tm',SNctm')) /\CiS4':=xor(CiS4tm',xor(UNspm,xor(UNopm,CiS2tm')))
  /\SNc':=xor(UNspm,xor(UNopm,xor(SNctm',xor(CiS4',CiS1tm'))))
  /\TSc':=xor(TSctm',xor(SNc',xor(RNspm,UNspm))) /\CSS1':={H(SNc')}_inv(KCpu)
  /\CiS1':=xor(CiS1tm',CSS1')/\CiS2':= xor(CiS2tm',CSS1')
  /\CSS2':={H(RNspm.UNspm.Nc'.TSc')}_inv(KCpu)
  /\CSS3':={H(RNopm.UNopm.Nc'.TSc')}_inv(KCpu)
  /\secret({SNc',CSS1',TSc'},sec2,{CS,Ci}) /\secret({CSS2',CSS3'},sec3,{CS,Ci,AS})
% CS creates authorisation request to AS
  /\SNcs':=new() /\TScs':=new() /\CSS4':= {H(SNcs')}_inv(KCSpu)
  /\CSS2tm':=xor(CSS2',CSS4') /\CSS3tm':=xor(CSS3',CSS4')
  /\Ncs':=xor(Nc',xor(TSc',xor(TScs',SNcs'))) /\TSctm':=xor(TSc',xor(SNcs',xor(RNspm,UNspm)))
  /\TScstm':=xor(TScs',xor(SNcs',xor(RNopm,UNopm)))
  /\SNcstm':=xor(UNspm,xor(UNopm,xor(SNcs',xor(CiS4',CSS2tm))))
  /\CiS4tm':=xor(CiS4',xor(UNspm,xor(UNopm,CSS3tm')))
  /\RNspn':=xor(RNspl,xor(CSS2tm',SNcstm')) /\RNopn':=xor(RNopl,xor(CSS3tm',SNcstm'))
  /\UNspn':=xor(UNspl,xor(CSS2tm',SNcstm')) /\UNopn':=xor(UNopl,xor(CSS3tm',SNcstm'))
% CS sends request to AS
  /\SND(AS.CSS2tm'.RNspn'.UNspn'.Ncs'.TSctm'.TScstm'.SNcstm'.CSS3tm'.RNopn'.UNopn'.CiS4tm')
  /\secret({RNspm,UNspm,RNopm,UNopm},sec4,{CS,AS}) /\secret({SNcs',CSS4',TScs',Nc'},sec7,{CS,AS})

% CS receives authorisation response from AS
2.State=1
  /\RCV(CS.ASS2tm'.UNspn'.TSastm')=|>State':=2 /\CSS2tm':=xor(ASS2tm',xor(CSS4,CSS1))
  /\TScs':=new() /\TScstm':=xor(TScs',xor(SNc,UNopm)) /\UNspn':=xor(UNspl,xor(CSS2tm',TScstm'))
  /\witness(CS,AS,auth3,{UNspm,UNopm,Nc,TSc,TSas})
% CS sends authorisation response to Ci
  /\SND(Ci.CSS2tm'.UNspn'.TScstm')

% CS receives Shamir's response from Ci
3. State = 2
  /\RCV(CS.CiS6tm'.UNspn'.TSctm')=|>State':=3
  /\UNspl':=xor(UNspn',xor(CiS6tm',TSctm')) /\TSc':=xor(TSctm',xor(SNc,UNspm))
  /\CiS6tm':=xor(CiS6tm',xor(CSS1,CSS4)) /\request(CS,Ci,auth1,{UNspm,UNopm,TScs})
  /\TScs':=new() /\TScstm':=xor(TScs',xor(SNcs,UNopm)) /\UNspn':=xor(UNspl,xor(CiS6tm',TScstm'))
% CS sends Shamir's response to AS
  /\SND(AS.CiS6tm'.UNspn'.TScstm')

% CS receives decision & data response from AS
4.State=3
  /\RCV(CS.ASS2tm'.ASS3tm'.UNspn'.TSastm'.Decision.Data)=|>State':=4
  /\UNspl':=xor(UNspn',xor(ASS2tm',TSastm')) /\TSas':=xor(TSastm',xor(SNcs,UNopm))
  /\ASS7':=xor(CSS2,xor(CSS4,xor(CiS4,ASS2tm))) /\CSS5':={H(Data)}_inv(KASpu)
  /\ASS2':=xor(ASS2tm',xor(CSS4,xor(ASS7,CiS4))) /\ASS3':=xor(ASS3tm,xor(CSS4,xor(ASS7,CiS4)))
  /\CiS4':=xor(ASS2tm',xor(CSS4,xor(ASS7,ASS2'))) /\request(CS,AS,auth4,{SNcs,CSS1,CiS4,TScs})
  /\TScs':=new() /\CSS2tm':=xor(CSS2,xor(CSS1,xor(CSS5,CiS4)))
  /\CSS3tm':=xor(CSS3,xor(CSS1,xor(CSS5,CiS4)))
  /\TScstm':=xor(TScs',xor(SNc,UNopm)) /\UNspn':=xor(UNspl,xor(CSS2tm',TScstm'))
% CS sends decision & data response to Ci
  /\SND(Ci.CSS2tm'.CSS3tm'.UNspn'.TScstm'.Decision.Data)
  /\witness(CS,Ci,auth2,{SNc,CSS1,CiS4,TSc})
end role
```

Figure 7.31: *CS* role of PAX in HLPSL

```
role attributesServer(AS,CS,DS:agent, KASpu,KCSpu,KDSpu:public_key, Ssp,Sop:text,H:hash_func, SND
                      RCV:channel(dy))
played_by AS def=
local
     State:nat, TSc,TScs,TSas,TSds,TSctm,TScstm,TSastm,TSdstm,Nc,Ncs,SNcs,SNas,SNcstm,SNastm:text,
     ASS1,ASS2,ASS3,ASS4,ASS5,ASS6,ASS7,ASS2tm,ASS3tm,ASS6tm:text, CiS4,CiS4tm,CiS6tm,CSS2,
     CSS3,CSS2tm,CSS3tm,DSS2tm,DSS4tm,DSS4:text, SP,OP,URsp,UNsp,URop,UNop,RNsp,RNop:text,
     UNspl,UNspm,UNsph,UNopl,UNopm,UNoph:text, RNspl,RNspm,RNsph,RNopl,RNopm,RNoph:text,
     RNspn,RNopn,UNspn,UNopn:text, S_ID,O_ID,S_R,O_R:message,Ci:agent,SSs,MS,Data,Decision:text
const
     sec1,sec3,sec4,sec5,sec6,sec7,sec8,sec9,sec10,auth3,auth4,auth5,auth6:protocol_id
init  State := 0
transition
% AS receives from CS
1.State = 0
  /\RCV(AS.CSS2tm'.RNspn'.UNspn'.Ncs'.TSctm'.TScstm'.SNcstm'.CSS3tm'.RNopn'.UNopn'.CiS4tm')=|>
  State':= 1/\RNspl':=xor(RNspn',xor(CSS2tm',SNcstm')) /\RNopl':=xor(RNopn',xor(CSS3tm',SNcstm'))
  /\UNspl':=xor(UNspn',xor(CSS2tm',SNcstm')) /\UNopl':=xor(UNopn',xor(CSS3tm',SNcstm'))
  /\CiS4':=xor(CiS4tm',xor(UNspm,xor(UNopm,CSS3tm')))
  /\SNcs':=xor(UNspm,xor(UNopm,xor(SNcstm',xor(CiS4',CSS2tm'))))
  /\TSc':=xor(TSctm',xor(SNcs',xor(RNspm,UNspm))) /\TScs':=xor(TScstm',xor(SNcs',xor(RNopm,UNopm)))
  /\Nc':=xor(Ncs',xor(TSc',xor(TScs',SNcs'))) /\ASS1':= {H(SNcs')}_inv(KCSpu)
  /\CSS2':=xor(CSS2tm',ASS1') /\CSS3':=xor(CSS3tm',ASS1')
  /\ASS2':= {H(RNspm.UNspm.Nc'.TSc')}_inv(KCSpu) /\ASS3':= {H(RNopm.UNopm.Nc'.TSc')}_inv(KCSpu)
  /\ASS4':={H(RNoph.UNoph.TSc')}_inv(KCSpu) /\SP':=URsp.UNsp/\OP':=URop.UNop
  /\secret({S_ID,O_ID,S_R,O_R},sec1,{AS,Ci}) /\secret({ASS2',ASS3'},sec3,{AS,CS,Ci})
  /\secret({RNspm,UNspm,RNopm,UNopm},sec4,{AS,CS}) /\secret({OP',CiS4'},sec5,{AS,DS,Ci})
  /\secret({SP',Ssp,MS,SSs},sec6,{AS,Ci}) /\secret({SNcs',ASS1',TScs',Nc},sec7,{AS,CS})
  /\secret(Sop,sec8,AS)
% AS creates Shamir's request
  /\TSas':=new() /\ASS2tm':=xor(ASS2,xor(ASS1,xor(SSs,xor(CiS4',Ssp))))
  /\TSastm':=xor(TSas',xor(SNcs',UNopm)) /\UNspn':=xor(UNspl',xor(ASS2tm',TSastm'))
% AS sends Shamir's request to CS
  /\SND(CS.ASS2tm'.UNspn'.TSastm')

% AS receives Shamir's response from CS
2.State=1
  /\ RCV(AS.CiS6tm'.UNspn'.TScstm')=|> State':= 2/\UNspl':=xor(UNspn',xor(CiS6tm',TScstm'))
  /\TScs':=xor(TScstm',xor(SNcs,UNopm)) /\MS':=xor(CiS6tm',xor(Ssp,ASS1))
  /\request(AS,CS,auth3,{UNspm,UNopm,Nc,TSc,TSas})
% AS creates data retrieval request
  /\SNas':=new()/\TSas':=new() /\ASS5':={H(SNas')}_inv(KASpu)
  /\ASS6':={H(RNopm.UNopm.SNas'.TSas'.CiS4)}_inv(KASpu)
  /\RNopn':=xor(RNopl,xor(TSas',SNas')) /\ASS6tm':=xor(ASS6',xor(TSas',ASS5'))
  /\TSctm':=xor(TSc,xor(SNas',UNoph)) /\TSastm':=xor(TSas',xor(SNas',UNopm))
  /\SNastm':=xor(UNopm,xor(SNas',xor(CiS4,ASS6tm')))
  /\CiS4tm':=xor(CiS4,xor(UNoph,xor(SNastm',UNopm))) /\UNopn':=xor(UNopl,xor(ASS6tm',TSastm'))
  /\secret({SNas',ASS5',TSas'},sec9,{AS,DS}) /\secret({RNoph,UNoph,ASS6'},sec10,{AS,DS})
% AS sends data retrieval request to DS
  /\SND(DS.ASS6tm'.RNopn'.UNopn'.SNastm'.TSctm'.TSastm'.CiS4tm')
  /\ witness(AS,DS,auth5,{CiS4,ASS6})

% AS receives data retrieval response from DS
3.State=2
  /\ RCV(AS.DS.DSS2tm'.DSS4tm'.UNopn'.TSdstm'.CiS4tm'.Data) =|> State':=3
  /\UNopl':=xor(UNopn',xor(DSS2tm',TSdstm')) /\TSds':=xor(TSdstm',xor(SNas,UNoph))
  /\CiS4':=xor(CiS4tm',xor(ASS5,TSds')) /\DSS4':=xor(DSS4tm',xor(ASS5,CiS4'))
  /\ASS6':=xor(DSS2tm',xor(DSS4',xor(TSds',ASS5))) /\ASS7':={H(Data)}_inv(KDSpu)
  /\request(AS,DS,auth6,{SNas,ASS5,RNoph,UNoph})
% AS creates decision & data response
  /\TSas':=new() /\ASS2tm':=xor(ASS2,xor(ASS1,xor(ASS7,CiS4)))
  /\ASS3tm':=xor(ASS3,xor(ASS1,xor(ASS7,CiS4))) /\TSastm':=xor(TSas',xor(SNcs,UNopm))
  /\UNspn':=xor(UNspl,xor(ASS2tm',TSastm'))
% AS sends data retrieval response (Data and Decision) to CS
  /\SND(CS.ASS2tm'.ASS3tm'.UNspn'.TSastm'.Decision.Data)
  /\witness(AS,CS,auth4,{SNcs,ASS1,CiS4,TScs})
end role
```

Figure 7.32: *AS* role of PAX in HLPSL

```
role dataServer(DS,AS:agent, KDSpu,KASpu:public_key, H:hash_func,SND,RCV:channel(dy))
played_by DS def=
local
     State:nat,
     TSc,TSctm,TSas,TSastm,TSds,TSdstm,SNas,SNastm:text,
     OP,UNopl,UNopm,UNoph:text, RNopl,RNopm,RNoph,RNopn,UNopn:text,
     DSS1,DSS2,DSS3,DSS4:text, DSS2tm,DSS4tm,CiS4,CiS4tm,ASS6tm:text,
     Ci:agent,Data:text
const  sec5,sec9,sec10,auth5,auth6:protocol_id
init  State := 0
transition
1.State = 0
  /\ RCV(DS.ASS6tm'.RNopn'.UNopn'.SNastm'.TSctm'.TSastm'.CiS4tm')=|>State':=1
  /\UNopl':=xor(UNopn',xor(ASS6tm',TSastm')) /\CiS4':=xor(CiS4tm',xor(UNoph,xor(SNastm',UNopm)))
  /\SNas':= xor(UNopm,xor(SNastm',xor(CiS4',ASS6tm'))) /\TSc':=xor(TSctm',xor(SNas',UNoph))
  /\TSas':=xor(TSastm',xor(SNas',UNopm)) /\RNopl':=xor(RNopn',xor(TSas',SNas'))
  /\DSS1':={H(SNas')}_inv(KASpu) /\DSS2':={H(RNopm.UNopm.SNas'.TSas'.CiS4')}_inv(KASpu)
  /\DSS3':={H(RNoph.UNoph.TSc')}_inv(KASpu)
  /\secret({OP,CiS4'},sec5,{DS,AS,Ci}) /\secret({SNas',DSS1',TSas'},sec9,{DS,AS})
  /\secret({RNoph,UNoph,DSS2'},sec10,{DS,AS}) /\request(DS,AS,auth5,{CiS4,DSS2})
% DS Creates data retrieval response
  /\TSds':=new() /\DSS4':={H(Data)}_inv(KDSpu) /\DSS4tm':=xor(DSS4',xor(DSS1',CiS4'))
  /\DSS2tm':=xor(DSS2',xor(DSS4',xor(TSds',DSS1'))) /\CiS4tm':=xor(CiS4',xor(DSS1',TSds'))
  /\TSdstm':=xor(TSds',xor(SNas',UNoph)) /\UNopn':=xor(UNopl',xor(DSS2tm',TSdstm'))
% DS sends data retrieval response to AS
  /\SND(AS.DSS2tm'.DSS4tm'.UNopn'.TSdstm'.CiS4tm'.Data)
  /\ witness(DS,AS,auth6,{SNas,DSS1,RNoph,UNoph})
end role
```

Figure 7.33: *DS* role of PAX in HLPSL

2. **Security Comparison**

   Tracking back to Chapter 2, where we reviewed the PERMIS HC system. Here we will show that the PAX has not suffered from PERMIS's problems (Chadwick et al. 2006) because each request to the healthcare provider has been signed randomly with the ECDSA algorithm, which includes both the roles (RN_s) and the pseudonyms (UN_s). In PAX, the policies are stored on the attributes server as Sigs and pseudonym rather than as explicit attributes in XACML (each user in PAX has attributes separate from other users that prevent the inheritance of attributes). Compared with Quantin et al. (2011), PAX has solved all requests' standardization and structure problems by including XACML v3.0 and ECDSA. XACML v3.0 offers standardization, and generic and rich policy language and is unified with the context of subject requests. It does not have problems converting requests from different sources. We also use ECDSA to generate very small keys relative to RSA to improve server performance.

   Furthermore, PAX does not need the keys complexity of PIPE (Riedl et al. 2008) because XACML has the flexibility to handle practitioners' and patients' requests, and we use only one high-performance signature algorithm. In PAX, all the attributes in the requests and policies are not clearly written as in Gajanayake et al. (2014). Moreover, data is anonymous to the patient when the data is transferred from the source to the target due to its link with a random pseudonym.

```
role session(Ci,CS,AS,DS:agent, KCpu,KCSpu,KASpu,KDSpu:public_key, H:hash_func, S_ID,O_ID,S_R,
             O_R:message,Ssp,Sop:text)
def=
local  SND1,RCV1,SND2,RCV2,SND3,RCV3,SND4,RCV4:channel(dy)
composition
  clienti(Ci,CS,KCpu,KCSpu,H,S_ID,O_ID,S_R,O_R,SND1,RCV1)
  /\centralServer(CS,Ci,AS,KCSpu,KCpu,KASpu,H,SND2,RCV2)
  /\attributesServer(AS,CS,DS,KASpu,KCSpu,KDSpu,Ssp,Sop,H,SND3,RCV3)
  /\dataServer(DS,AS,KDSpu,KASpu,H,SND4,RCV4)
end role

role environment()
def=
const
  ci,cs,as,ds,i:agent, kCpu,kCSpu,kASpu,kDSpu,ki:public_key,s_id,o_id,s_r,o_r:message, ssp,sop:text,
   h:hash_func, sec1,sec2,sec3,sec4,sec5,sec6,sec7,sec8,sec9,sec10,
  auth1,auth2,auth3,auth4,auth5,auth6,auth7,auth8:protocol_id

  intruder_knowledge={i,ci,cs,as,kCpu,kCSpu,kASpu,kDSpu,ki,inv(ki)}
composition
  session(ci,cs,as,ds,kCpu,kCSpu,kASpu,kDSpu,h,s_id,o_id,s_r,o_r,ssp,sop)
% Check replay attack
  /\session(ci,cs,as,ds,kCpu,kCSpu,kASpu,kDSpu,h,s_id,o_id,s_r,o_r,ssp,sop)
% Check MITM attack
  %/\session(cs,ci,as,ds,kCSpu,kCpu,kASpu,kDSpu,h,s_id,o_id,s_r,o_r,ssp,sop)
% Check impersonate Ci
  %/\session(i,cs,as,ds,ki,kCSpu,kASpu,kDSpu,h,s_id,o_id,s_r,o_r,ssp,sop)
% Check impersonate CS
  %/\session(ci,i,as,ds,kCpu,ki,kASpu,kDSpu,h,s_id,o_id,s_r,o_r,ssp,sop)
% Check impersonate AS
  %/\session(ci,cs,i,ds,kCpu,kCSpu,ki,kDSpu,h,s_id,o_id,s_r,o_r,ssp,sop)
% Check impersonate DS
  %/\session(ci,cs,as,i,kCpu,kCSpu,kASpu,ki,h,s_id,o_id,s_r,o_r,ssp,sop)
end role

goal
  secrecy_of sec1,sec2,sec3,sec4,sec5,sec6,sec7,sec8,sec9,sec10
  authentication_on auth1,auth2,auth3,auth4,auth5,auth6,auth7,auth8
end goal
environment()
```

Figure 7.34: Session, environment, and goal roles of PAX in HLPSL

Instead of one situation (emergency) as in HCPP, our project used several realistic situations such as doctor advisors, physician researchers, emergency doctors, and patients' relatives for healthcare users and used the XACML v3.0 policy language, which is effective for authorising users. The PAX does not require continuous mining (Sun et al. 2011) of patient data, but relies on the internal pseudonym to access medical records. XACS in Jo & Chung (2015) offers protection only against external attacks, whereas PAX offers protection against internal and external attacks by preventing attackers from identifying the personal information of legitimate users or patient data. In addition, the access control model in Seol et al. (2018) deals with real attributes, whereas PAX integrates signatures and pseudonyms within XACML policies and requests to prevent the clear exchange of user attributes during the

```
SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  /home/span/span/testsuite/results/PAX.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 6912 states
  Reachable  : 6912 states
  Translation: 2.12 seconds
  Computation: 139.21 seconds
```

Figure 7.35: Simulation result of PAX using CL-AtSe backend

authorisation process in healthcare applications.

The scheme in Wang et al. (2019) suffers from impersonation and timing threats. As well, the intruder can exploit cases of indirect users to trigger a DoS attack. PAX uses mutual authentication to prevent impersonation attacks and it uses random signatures to prevent timing and DoS attacks. There are a few of other scheme designed for HC systems. For instance, the scheme in Shafeeq et al. (2019) focuses on protecting the authorisation policy without paying attention to secure trusted authority and the single key in symmetric encryption. Also, their scheme relies solely on the ABAC model to determine access to the repository. PAX integrates security advantages into both RBAC and ABAC and does not have datasets/keys protection problems. Table 7.7 compares the security features provided in PAX and related works.

7.4.3 Performance Analysis

A performance analysis is conducted to examine the computation processes of PAX in improving the performance of authorisation processes.

- **Theoretical Performance Analysis**
  When the users become authenticated within the network it needs to send authorisation requests to the servers to access patient data in the EHR repository. Authorisation requests should perform lightweight processes to maintain network performance, especially if the user performs many authorisation requests at frequent and convergent times. PAX relies mainly

Table 7.7: Comparison of security features between PAX and other authorisation schemes

Security feature	Chadwick et al. (2006)	Quantin et al. (2011)	Riedl et al. (2008)	Gajanayake et al. (2014)	Sun et al. (2011)	Jo & Chung (2015)	Seol et al. (2018)	Wang et al. (2019)	Shafeeq et al. (2019)	**PAX scheme**
Mutual authentication									✓	✓
Preserving anonymity		✓	✓		✓			✓	✓	✓
Pseudonym		✓	✓		✓				✓	✓
Anti DoS	✓		✓		✓					✓
Anti dataset attack							✓	✓		✓
Anti MITM	✓	✓	✓		✓		✓	✓	✓	✓
Anti replay	✓		✓	✓	✓	✓	✓		✓	✓
Anti privileged insider					✓				✓	✓
Anti traceability			✓		✓					✓
Anti impersonation									✓	✓
Anti timing					✓					✓
Anti leakage			✓		✓			✓		✓
Authorisation policies	✓			✓		✓	✓	✓	✓	✓

on a range of techniques, such as ECDSA, XACML, Shamir scheme and random pseudonyms that perform lightweight procedures compared to other technologies.

First, PAX uses the ECDSA algorithm to sign subjects' and objects' information in both requests and policies. This algorithm is ideal for authorisation schemes compared to public key signature algorithms as described in Chapter 2, Table 2.2. To maintain performance, we were careful to reduce the number of signatures to reduce the burden on users' devices and servers. For instance, full processing associated with sending an authorisation request to access patient data requires $C_i = 6$ signatures, $CS = 5$ signatures, $AS = 7$ signatures and $DS = 4$ signatures (indirect user's case). In short, reducing the number of signatures will improve the performance of the PAX scheme. In addition, PAX uses XACML technology to build robust policies in determining access to patient data. XACML provides the best cashing storage procedures (Ilhan et al. 2015), flexibility in defining multiple rules, simplicity in configuring policy combination (Duan et al. 2016), response times, the grouping of requests and making decisions (WSO2 Team 2017) of existing technologies.

Furthermore, PAX relies on the Shamir scheme to prevent tracking of user information in special cases such as advisor, relative, researcher and emergency rather than encryption mechanisms. The Shamir scheme provides PAX with robust security while performing light operations compared to encryption procedures. PAX uses the Shamir scheme only in cases of indirect users which are lower relative to direct users such as nurses and doctors. As a result, authorisation requests with the Shamir scheme will be fewer, and the burden on the servers CS and AS will be lower. Moreover, the PAX scheme uses the Shamir scheme with $TH = 3-10$. The dependency of TH with efficient range allows the Shamir scheme to support both security and performance in PAX meritoriously (Al-Adhami et al. 2017). Finally, PAX uses random pseudonyms to prevent the sending of real information through the network instead of k-anonymity and encryption mechanisms that perform complex processes and increase overheads. Therefore, PAX is considered a performance efficient scheme because it supports lightweight and high-performance mechanisms.

- **Experimental Performance Analysis**
  Communication costs (storage overheads) and computation (execution time) are calculated to extract PAX's performance. Applications codes (C_i, CS, AS and DS) are written in Java Programming Language. The authorisation decision engine is supported on Balana XACML 3.0 open source available at https://github.com/wso2/balana, and is explained and managed by the WSO2 server. We used Maven to deal with Balana XACML as a library in Java. Also, PAX's results have been implemented on Ubuntu 16.04 LTS, processor Intel Core i5 2.6GHz, OS type 32-bit, Memory 4 GiB and disk 32.0 GB.

To compute the cost of communication, we compute the bits of all security parameters. Signature length (ECDSA) is 256-bit. The number of messages transmitted through the direct user authorisation is six and the indirect user authorisation is 10. In addition, response size is 196 bytes to 205 bytes, request size is 2091 bytes to 2098 bytes, policy size is 1483 bytes to 1492 bytes, policy with Shamir size is 1489 bytes to 1495 bytes, user information (attributes) is 586 bytes to 621 bytes, non-real patient data is 337 bytes to 500 bytes and non-real all patient data size for researchers is 1159 bytes to 2688 bytes (25 patients).

On the C_i side, it produces a request of subject attributes (C_iSig_1=128, $RN_{sp_{tm}}$=128, $UN_{sp_{tm}}$=128, N_C=8, $TS_{C_{tm}}$=16 and $SN_{C_{tm}}$=128) and object attributes (C_iSig_2=128, $RN_{op_{tm}}$=128, $UN_{op_{tm}}$=128, N_C=8 and C_iSig_4=128). The total size of the parameters is 1056 bytes and the number of bits (attributes) inside the XACML request is 4224 bits. When C_i sends response to CS, this response includes security parameters ($C_iSig_{6_{tm}}$=128, $UN_{sp_{tm}}$=128, and $TS_{C_{tm}}$=16). The total size of the parameters is 272 bytes and the number of bits inside the XACML response is 1088 bits. On the CS

side, it produces a request subject attributes ($CSSig_2$=128, $RN_{sp_{tm}}$=128, $UN_{sp_{tm}}$=128, N_{CS}=8, $TS_{C_{tm}}$=16, $TS_{CS_{tm}}$=16 and $SN_{CS_{tm}}$=128) and object attributes ($CSSig_3$=128, $RN_{op_{tm}}$=128, $UN_{op_{tm}}$=128 and C_iSig_4=128). The total size of the parameters is 1064 bytes and the number of bits inside the XACML request is 4256 bits. When CS sends the response to C_i, this response includes security parameters ($CSSig_{2_{tm}}$=128, $CSSig_{3_{tm}}$=128, $UN_{sp_{tm}}$=128, and $TS_{CS_{tm}}$=16). The total size of the parameters is 400 bytes and the number of bits inside the XACML response is 1600 bits. On the AS side, it produces a XACML request that includes security parameters ($ASSig_{6_{tm}}$=128, $RN_{op_{tm}}$=16, $UN_{op_{tm}}$=128, $SN_{AS_{tm}}$=128, $TS_{C_{tm}}$=16, $TS_{AS_{tm}}$=16 and $C_iS_{4_{tm}}$=128). The total size of the parameters is 560 bytes and the number of bits inside the XACML request is 2240 bits. Also, it produces response attributes ($ASSig_2$=128, $ASSig_3$=128, $UN_{sp_{tm}}$=128 and $TS_{AS_{tm}}$=16) . The total size of the parameters is 400 bytes and the number of bits inside the XACML request is 1600 bits. On the DS side, it produces response attributes ($DSSig_2$=128, $DSSig_4$=128, $UN_{op_{tm}}$=128, $TS_{DS_{tm}}$=16 and $C_iSig_{4_{tm}}$=128) . The total size of the parameters is 528 bytes and the number of bits inside the XACML request is 2112 bits.

To compute the cost of computation, we had obtained preliminary results about running time for ECDSA-256 key generation, signature and verification (100 times, information size of 2091 bytes) as shown in Figure 7.36. We noted execution time of ECDSA's procedures is faster than signature algorithms dependant results in Suárez-Albela et al. (2019). The minimum execution time for ECDSA 256-bit key generation is 0.002009 ms, signature is 0.002994 ms, verification is 0.003496 ms and full time for ECDSA 256-bit is 0.00921 ms.

- **Performance Comparison**
  In this section, we will compare the superiority of PAX over the authorisation schemes in the performance side. Although the implementation environment for authorisation schemes varies, we have made some comparisons that demonstrate the superiority of PAX's performance over existing schemes.

  The scheme in Chadwick et al. (2006) is based on X.509 to manage certificates/security policies and Shibboleth authentication to provide a single sign-on. Their scheme requires additional storage for certificates (PKCS#12) as well as policies. Also, the implementation of Shibboleth on servers performs complex operations affecting the performance of servers and

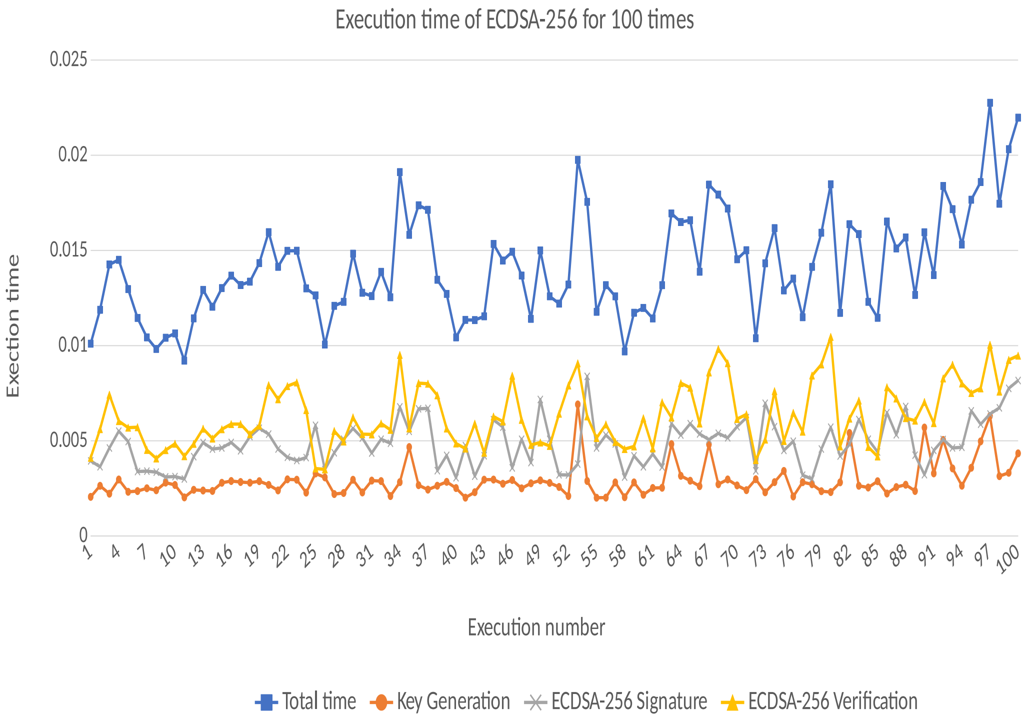


Figure 7.36: Implementations of ECDSA 256-bit

it does not correspond with the speed of the Internet. In addition, Shibboleth has a security vulnerability that may allow unauthorised users to access an EHR repository frequently and affects network performance. PAX relies on XACML for efficient policy management and ECDSA-256 bit and produces small keys for signing authorisation requests more efficiently and securely than Shibboleth. PAX also does not require additional storage to store certificates. The scheme in Riedl et al. (2008) suffers from problems with multiple key complexity expenses, the complexity of encryption and the consumption of storage sources that adversely affect server performance. While PAX uses the ECDSA public key algorithm that does not require large storage (small keys) and the use of random public/private keys without complexity (Riedl et al. 2008). The scheme in Quantin et al. (2011) has major problems with low performance: the diversity of authorisation requests contexts and the cost of encrypting data by RSA. Compared to PAX, it uses XACML which is efficient in standardising contexts of authorisation requests as well as ECDSA which provides better performance efficiency than RSA.

Moreover, instead of using four access control models (DAC, MAC, RBAC and PBAC) in the scheme (Gajanayake et al. 2014), these models further complicate authorisation processes and the difficulty of managing security policies and authorisation requests. PAX relies on RBAC and ABAC to determine user access to data with support for flexibility, ease of implementation and efficient management. The XACS scheme in Sun et al. (2011) uses XML to design an authorisation system. XML is used to structure data/information and not to specify access to the repository. Using XML authorisation policies reduces performance in responding to authorisation requests. PAX uses XACML which is essentially designed to manage efficiently authorisation policies and user requests. The HCPP scheme in Jo & Chung (2015) relies on encryption to protect the database and prospecting/searching to access legitimate user information. Both mechanisms consume storage and server time. PAX relies on signatures to protect user information without the need for full encryption of the database. It also relies on multiple pseudonyms to efficiently access user information. The scheme in Seol et al. (2018) is based on XACML 2.0, which has problems with decision precedence between 'Not Applicable' and 'Indeterminate'. Also, their scheme performs the process of signing the encrypted information in the authorisation request partly or fully. PAX relies on XACML 3.0 which is more functional, faster and better logical results in decision precedence. Also, PAX performs the signing process directly on user attributes only without overheads of encryption.

The scheme in Wang et al. (2019) suffers from many shortcomings affecting performance such as data/policy decryption costs, use of RSA for encryption and signing, costing and management of secret key, management of users' attributes by a single authority, updating of keywords list when adding new keyword and operations search by keywords consumes server time. Compared to PAX, it does not require costly decryption operations and uses XACML and pseudonyms techniques to access patient data safely and with high performance. The scheme in Shafeeq et al. (2019) has significant storage and computation costs. Their scheme requires a large size for key storage. In addition, using multi-signature is extremely expensive on network performance as the size of the signature increases with the number of signatures. Moreover, their scheme requires the size of 2500 characters to store policy and five seconds to fetch the policy. While PAX requires 256 bits for key storage, size 1465 characters for policy storage without Shamir, size 1477 characters for policy storage with Shamir and less than a second to fetch policy. Ultimately, PAX does not require encryption costs used in schemes (Riedl et al. 2008, Quantin et al. 2011, Jo & Chung 2015, Seol et al. 2018, Wang et al. 2019) but relies on signatures performed efficiently by ECDSA.
Table 7.8 shows performance comparison between PAX and existing

Table 7.8: Comparison of performance between PAX and existing authorisation schemes

Scheme	Storage	Management	Communication	Computation
Chadwick et al. (2006)	M_E	M_E (X.509)	S	H_E (Sig)
Riedl et al. (2008)	L_E	L_E	S	L_E ($Enc+Sig$)
Quantin et al. (2011)	M_E	L_E (MRSE)	S	L_E ($Enc+Sig$)
Gajanayake et al. (2014)	-	M_E (Models)	-	-
Sun et al. (2011)	M_E	L_E (SSE)	S	M_E ($Enc+Sig$)
Jo & Chung (2015)	L_E	M_E (XML)	$F+G$	M_E (Enc)
Seol et al. (2018)	H_E	H_E (XACML)	$F+S+G$	L_E ($Enc+Sig$)
Wang et al. (2019)	L_E	L_E (ABE)	$S+G$	L_E ($Enc+Sig$)
Shafeeq et al. (2019)	L_E	H_E (XACML)	$F+S+G$	M_E ($Enc+Sig$)
PAX	H_E	H_E (XACML)	$F+S+G$	H_E (Sig)

authorisation schemes (H_E is high efficiency, M_E is medium efficiency, L_E is low efficiency, F is flexibility, S is scalability, G is granularity, Enc is encryption and Sig is signature).

7.5 Summary of the Chapter

In this chapter, we introduced a security testing tool (simulation and mathematical logic) to analyse protocols security. Thereafter, we discussed and analysed the security results in terms of data collection, authentication and authorisation in preventing a wide spectrum of attacks. For each scheme, REISCH, PAX and RAMHU, we have made a theoretical and experimental analysis for both security and performance analysis. All these analyses are shown by presenting their experimental data/figures, in comparison with the existing schemes.

Chapter 8: Conclusions and Future Directions

In this chapter, we summarise our research journey and experiences, and list intended future directions. Our objective was to explore an innovative strategy to develop an efficient and secure healthcare application using the latest information technology, in particular electronic health/medical records (EHR) and a wireless sensor network (WSN).

8.1 Conclusions of the Thesis

After three or so years of arduous efforts, we have developed the general structure of a HC application that consists of four components: data-collection using WSNs, an EHR/EMRs repository, user privacy protection and the user differentiated access control.

In terms of the data collection scheme, wireless sensor networks provide unique and important care services when combined with EMR. Unfortunately, these networks suffer from performance and security problems as mentioned in the previous chapters. Therefore, we have proposed a REISCH scheme to address performance and security problems and cover gaps in existing research. As a result, the REISCH uses ECDSA-BLAKE2bp which provides better performance than the original ECDSA-SHA1 algorithm. REISCH with the modified algorithm saves more than 24% alive SNs. In addition, the results of the security analysis prove that REISCH is safe against attacks in the threat model.

In terms of an authentication scheme, healthcare systems require robust authentication to ensure that only legitimate users exchange patient data. During the investigation of previous authentication protocols, we found that they were vulnerable to some known attacks. Therefore, we proposed a new robust authentication protocol (RAMHU) to prevent internal, external, passive, and active attacks. We have used a variety of mechanisms that ensure the protection and concealment of personal information to legitimate users. Our scheme uses multi pseudonyms for both users and medical centres to prevent the transmission of real information in the authentication request and MAC address to prevent counterfeit devices from connecting to the network. We used lightweight encryption and signature algorithms (as described in Chapter 3 Sections 3.4.2 and 3.4.3) to ensure that RAMHU's efficient interaction with user requests is ensured. In addition, we provided a formal and informal security analysis to demonstrate the effectiveness of RAMHU in repelling known attacks. We conclude that the RAMHU scheme provides high-level security and performance that maintains authentication information for users against various attacks.

In terms of the authorisation scheme, the security and privacy of medical records have, in recent years, become essential requirements for the establishment of any healthcare system. To ensure the provision of security and privacy, this thesis proposes a PAX authorisation system that supports pseudonym, anonymity and XACML. Specifically, the proposed system uses pseudonyms to separate personal information about patient data, anonymity to hide subjects' information, and XACML to create distributed access control policies to authorise subjects' requests to objects' records in the EHR. Different from the large amount of theoretical investigation in the existing literature, this scheme achieves the security and privacy preservation by utilizing the pseudonym and anonymity techniques, which can reduce the unnecessary consumption of time and burden on the server. We conclude that the PAX system provides a security level that maintains patient privacy, and the system particularly protects patient information from indirect users (advisors, patients' relatives, researchers, and emergency doctors), who are considered a serious security threat to any healthcare system because they can carry out internal attacks using the privileges granted to them.

8.2 Future Directions

To further develop the proposed HC project, we intend to add some features to support security and privacy in EMR and EHR.

1. After storing collected data in the EMR, our project needs supporting security procedures between CS and DS to store partial records in the remote repository. This process requires accurate and robust security procedures to prevent intruders from modifying historical patient data

2. The NTRUSign algorithm is more efficient in term of performance than ECDSA, but is less secure (Driessen et al. 2008). We intend to investigate

the integration of NTRUSign performance features with ECDSA to improve computation operations and energy savings in the HWSN. In addition, we will try to compare NTRUSign-BLAKE2bp and ECDSA-BLAKE2bp in terms of security and performance

3. Support our scheme using ECDSA-BLAKE2bp with efficient curves such as the Edward curve, efficient PM methods such as Frobenius, and system coordinates such as λ-Projective in Oliveira et al. (2014) to improve the efficiency of patient data signing in the HWSN

4. We intend to integrate our scheme into a real HWSN environment to evaluate the efficiency and feasibility of REISCH algorithms to improve the lifetime of SNs in patient data collection as much as possible

5. To support the authentication scheme (RAMHU), we intend to add users' biometric properties such as fingerprint, iris, or sound recognition to the authentication procedures. This procedure strengthens security precautions that prevent external attackers from accessing network services. However, adding these features requires performance evaluation in the login and authentication protocols

6. RAMHU uses lightweight and efficient performance algorithms that, according to many researchers, have shown that ECIES and PHOTON are efficient encryption and signature algorithms, respectively. We intend to evaluate RAMHU in terms of efficiency and discovery of performance standards such as end-to-end request delay, throughput, and error rate

7. From other future works, we intend to use a robust security mechanism with $\oplus$ operation to support the exchange and management of public keys. Broadcasting public keys in general needs more attention to prevent various attacks. Public key storage on users' devices can also be a source of compromised authentication

8. For more testing to prevent the analysis of authorisation requests and responses, we intend to test the authorisation scheme with side-channel attacks (SCA) such as simple power analysis (SPA), differential power analysis (DPA) and template attacks. In addition to performance testing, we intend to test XACML (PDP engine) with complex combining policies and algorithms while maintaining the confidentiality of user information and preventing leakage to intruders

9. We will focus on patient data without the use of cryptographic mechanisms in examining patients' daily conditions, use real patient data to test PAX

with large data, and allow PAX to distinguish between patient history, daily status, and the purpose of data access. We will also encrypt the patients' old medical records (within a certain period) that are not frequently retrieved by healthcare providers in a manner that does not affect the efficiency of the server in providing service to users

10. We will investigate the application of a light hash algorithm to generate key ephemeral k in ECDSA and patient pseudonyms, to support increased randomization while maintaining system performance as an additional security measure to protect the privacy of medical records in the EHR.

8.3 Summary of the Chapter

We summarise our study by giving the conclusions of our project schemes (the data collection, authentication and authorisation schemes) in the construction of a robust HC application in support of security and privacy issues. Then, we described the future work to develop our project in terms of data collection, authentication and authorisation. We should conclude this thesis by saying that "writing this up is just to summarise this period of experience, it will never be at an end; it just opens a door for us to go further!".

References

AbdAllah, E. G., Zulkernine, M. & Hassanein, H. S. 2018, 'Preventing unauthorized access in information centric networking', *Security and Privacy*, vol. 1, no. 4, pp. 1–13.

Abdelraheem, M. A. 2012, 'Estimating the probabilities of low-weight differential and linear approximations on PRESENT-like ciphers', *International Conference on Information Security and Cryptology*, *Springer*, pp. 368–382.

Abdouli, A. S., Baek, J. & Yeun, C. Y. 2011, 'Survey on computationally hard problems and their applications to cryptography', *2011 International Conference for Internet Technology and Secured Transactions (ICITST)*, *IEEE*, pp. 46–52.

Abueh, Y. J. & Liu, H. 2016, 'Message authentication in driverless cars', *Technologies for Homeland Security (HST), 2016 IEEE Symposium on*, *IEEE*, pp. 1–6.

Aceto, G., Persico, V. & Pescapé, A. 2018, 'The role of information and communication technologies in healthcare: taxonomies, perspectives, and challenges', *Journal of Network and Computer Applications*, vol. 107, pp. 125–154.

Ahmadian, Z. & Jamshidpour, S. 2018, 'Linear subspace cryptanalysis of Harn's secret sharing-based group authentication scheme', *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 2, pp. 502–510.

Aitzhan, N. Z. & Svetinovic, D. 2018, 'Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams', *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 840–852.

Al-Adhami, A., Ambroze, M., Stenget, I. & Tomlinson, M. 2017, 'A 256 bit implementation of ecc-rfid based system using shamir secret sharing scheme and keccak hash function', *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, *IEEE*, pp. 165–171.

Al Ameen, M. & Kwak, K. S. 2011, 'Social issues in wireless sensor networks with healthcare perspective', *The International Arab Journal of Information Technology*, vol. 8, no. 1, pp. 52–58.

Al Ameen, M., Liu, J. & Kwak, K. 2012, 'Security and privacy issues in wireless sensor networks for healthcare applications', *Journal of medical systems*, vol. 36, no. 1, pp. 93–101.

Al-Janabi, S., Al-Shourbaji, I., Shojafar, M. & Shamshirband, S. 2017, 'Survey of main challenges (security and privacy) in wireless body area networks for healthcare applications', *Egyptian Informatics Journal*, vol. 18, no. 2, pp. 113–122.

Al Maashri, A., Pathuri, L., Awadalla, M., Ahmad, A. & Ould-Khaoua, M. 2016, 'Optimized hardware crypto engines for XTEA and SHA-512 for wireless sensor nodes', *Indian Journal of Science and Technology*, vol. 9, no. 29, pp. 1–7.

Al-Turjman, F. & Alturjman, S. 2018, 'Confidential smart-sensing framework in the IoT era', *The Journal of Supercomputing*, vol. 74, no. 10, pp. 5187–5198.

Al-Zubaidie, M., Zhang, Z. & Zhang, J. 2019*a*, 'PAX: using pseudonymization and anonymization to protect patients' identities and data in the healthcare system', *International Journal of Environmental Research and Public Health*, vol. 16, no. 9, pp. 1–36.

Al-Zubaidie, M., Zhang, Z. & Zhang, J. 2019*b*, 'RAMHU: a new robust lightweight scheme for mutual users authentication in healthcare applications', *Security and Communication Networks*, vol. 2019, pp. 1–26.

Alhaqbani, B. & Fidge, C. 2008, 'Privacy-preserving electronic health record linkage using pseudonym identifiers', *HealthCom 2008-10th International Conference on E-health Networking, Applications and Services*, *IEEE*, pp. 108–117.

Alkureishi, M. A., Lee, W. W., Lyons, M., Wroblewski, K., Farnan, J. M. & Arora, V. M. 2018, 'Electronic-clinical evaluation exercise (e-CEX): a new patient-centered EHR use tool', *Patient education and counseling*, vol. 101, no. 3, pp. 481–489.

Alturki, M. 2017, 'Achieving a secured collaborative environment in e-SiHi system users perspective on a framework to improve patients information', *2017 International Conference on Informatics, Health & Technology (ICIHT)*, *IEEE*, pp. 1–4.

Amin, R., Islam, S. H., Biswas, G., Khan, M. K. & Kumar, N. 2018, 'A robust and anonymous patient monitoring system using wireless medical sensor networks', *Future Generation Computer Systems*, vol. 80, pp. 483–495.

Amin, R., Kumar, N., Biswas, G., Iqbal, R. & Chang, V. 2018, 'A light weight authentication protocol for IoT-enabled devices in distributed cloud computing environment', *Future Generation Computer Systems*, vol. 78, pp. 1005–1019.

Amy, M., Di Matteo, O., Gheorghiu, V., Mosca, M., Parent, A. & Schanck, J. 2016, 'Estimating the cost of generic quantum pre-image attacks on SHA-2 and SHA-3', *International Conference on Selected Areas in Cryptography*, *Springer*, pp. 317–337.

Anandakumar, N. N., Peyrin, T. & Poschmann, A. 2014, 'A very compact FPGA implementation of LED and PHOTON', *International Conference in Cryptology in India*, *Springer*, pp. 304–321.

Arshad, H., Teymoori, V., Nikooghadam, M. & Abbassi, H. 2015, 'On the security of a two-factor authentication and key agreement scheme for telecare medicine information systems', *Journal of medical systems*, vol. 39, no. 8, pp. 1–10.

Asan, O. 2017, 'Providers' perceived facilitators and barriers to EHR screen sharing in outpatient settings', *Applied ergonomics*, vol. 58, pp. 301–307.

Aumasson, J.-P., Henzen, L., Meier, W. & Naya-Plasencia, M. 2013, 'Quark: a lightweight hash', *Journal of cryptology*, vol. 26, no. 2, pp. 313–339.

Aumasson, J.-P., Henzen, L., Meier, W. & Phan, R. C.-W. 2008, 'SHA-3 proposal BLAKE', *Submission to NIST*, vol. 92, pp. 1–76.

Aumasson, J.-P., Neves, S., Wilcox-O'Hearn, Z. & Winnerlein, C. 2013, 'BLAKE2: simpler, smaller, fast as MD5', *International Conference on Applied Cryptography and Network Security*, *Springer*, pp. 119–135.

Awaad, M. H. & Jebbar, W. A. 2015, 'Extending the WSN lifetime by dividing the network area into a specific zones', *International Journal of Computer Network and Information Security*, vol. 7, no. 2, pp. 33–39.

Ayyildiz, C., Erdem, H. E., Dirikgil, T., Dugenci, O., Kocak, T., Altun, F. & Gungor, V. C. 2019, 'Structure health monitoring using wireless sensor networks on structural elements', *Ad Hoc Networks*, vol. 82, pp. 68–76.

Babu, K. R. & Padmanabhan, V. 2018, 'Automated validation of DNSSEC', *Progress in Computing, Analytics and Networking*, *Springer*, pp. 51–59.

Bachiller, Y., Busch, P., Kavakli, M. & Hamey, L. 2018, 'Survey: big data application in biomedical research', *Proceedings of the 2018 10th International Conference on Computer and Automation Engineering*, *ACM*, pp. 174–178.

Barker, E. & Dang, Q. 2016, *NIST special publication 800-57 part 1, revision 4*, Technical report, National Institute of Standards and Technology, U.S. Department of Commerce, viewed 01 May 2019, <http://dx.doi.org/10.6028/NIST.SP.800-57pt1r4>.

Beglaryan, M., Petrosyan, V. & Bunker, E. 2017, 'Development of a tripolar model of technology acceptance: hospital-based physicians' perspective on EHR', *International journal of medical informatics*, vol. 102, pp. 50–61.

Beltran, V., Martinez, J. & Skarmeta, A. 2017, 'User-centric access control for efficient security in smart cities', *Global Internet of Things Summit (GIoTS), 2017*, *IEEE*, pp. 1–6.

Berger, T. P., D'Hayer, J., Marquet, K., Minier, M. & Thomas, G. 2012, 'The GLUON family: a lightweight hash function family based on FCSRs', *International Conference on Cryptology in Africa*, *Springer*, pp. 306–323.

Bhatia, T. & Verma, A. 2017, 'Data security in mobile cloud computing paradigm: a survey, taxonomy and open research issues', *The Journal of Supercomputing*, vol. 73, no. 6, pp. 2558–2631.

Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K. & Verbauwhede, I. 2011, 'SPONGENT: a lightweight hash function', *International Workshop on Cryptographic Hardware and Embedded Systems*, *Springer*, pp. 312–325.

Bogos, S., Gaspoz, J. & Vaudenay, S. 2018, 'Cryptanalysis of a homomorphic encryption scheme', *Cryptography and Communications*, vol. 10, no. 1, pp. 27–39.

Bojjagani, S. & Sastry, V. 2017, 'A secure end-to-end sms-based mobile banking protocol', *International journal of communication systems*, vol. 30, no. 15, pp. 1–19.

Bojjagani, S. & Sastry, V. 2019, 'A secure end-to-end proximity NFC-based mobile payment protocol', *Computer Standards & Interfaces*, vol. 66, pp. 1–21.

Bos, J. W., Halderman, J. A., Heninger, N., Moore, J., Naehrig, M. & Wustrow, E. 2014, 'Elliptic curve cryptography in practice', *International Conference on Financial Cryptography and Data Security*, *Springer*, pp. 157–175.

Boubiche, D. E., Boubiche, S., Toral-Cruz, H., Pathan, A.-S. K., Bilami, A. & Athmani, S. 2016, 'SDAW: secure data aggregation watermarking-based scheme in homogeneous WSNs', *Telecommunication Systems*, vol. 62, no. 2, pp. 277–288.

Bradford Networks 2012, *Top 4 network security challenges in healthcare*, viewed 09 June 2018, <http://cipherwire.net/wp-content/uploads/2013/06/>.

Brockmann, A. 2018, 'A formula that generates hash collisions', *arXiv preprint arXiv:1808.10668*, *Cornell University*.

Brossard, D., Gebel, G. & Berg, M. 2017, 'A systematic approach to implementing ABAC', *Proceedings of the 2nd ACM Workshop on Attribute-Based Access Control*, *ACM*, pp. 53–59.

Bruland, P., Doods, J., Brix, T., Dugas, M. & Storck, M. 2018, 'Connecting healthcare and clinical research: workflow optimizations through seamless integration of EHR, pseudonymization services and EDC systems', *International journal of medical informatics*, vol. 119, pp. 103–108.

Calvillo-Arbizu, J., Roman-Martinez, I. & Roa-Romero, L. M. 2014, 'Standardized access control mechanisms for protecting ISO 13606-based electronic health record systems', *IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, *IEEE*, pp. 539–542.

Cantu, M., Kim, J. & Zhang, X. 2017, 'Finding hash collisions using MPI on HPC clusters', *Systems, Applications and Technology Conference (LISAT), 2017 IEEE Long Island*, *IEEE*, pp. 1–6.

Carvalho, J. V., Rocha, Á., van de Wetering, R. & Abreu, A. 2019, 'A maturity model for hospital information systems', *Journal of Business Research*, vol. 94, pp. 388–399.

Chadwick, D., Zhao, G., Otenko, S., Laborde, R., Su, L. & Nguyen, T. A. 2006, 'Building a modular authorisation infrastructure', *The UK e-Science All Hands Meeting*, *Nottingham*, pp. 1–8.

Chandrakar, P. & Om, H. 2017, 'A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC', *Computer Communications*, vol. 110, pp. 26–34.

Chaves, R., Sousa, L., Sklavos, N., Fournaris, A. P., Kalogeridou, G., Kitsos, P. & Sheikh, F. 2016, 'Secure hashing: SHA-1, SHA-2, and SHA-3', *Circuits and Systems for Security and Privacy*, pp. 105–132.

Chen, C.-M., Fang, W., Wang, K.-H. & Wu, T.-Y. 2017, 'Comments on "an improved secure and efficient password and chaos-based two-party key agreement protocol"', *Nonlinear Dynamics*, vol. 87, no. 3, pp. 2073–2075.

Chen, Y.-Y., Lu, J.-C. & Jan, J.-K. 2012, 'A secure EHR system based on hybrid clouds', *Journal of medical systems*, vol. 36, no. 5, pp. 3375–3384.

Cheneau, T., Boudguiga, A. & Laurent, M. 2010, 'Significantly improved performances of the cryptographically generated addresses thanks to ECC and GPGPU', *computers & security*, vol. 29, no. 4, pp. 419–431.

Chiriaco, V., Franzen, A., Thayil, R. & Zhang, X. 2017, 'Finding partial hash collisions by brute force parallel programming', *Systems, Applications and Technology Conference (LISAT), 2017 IEEE Long Island*, IEEE, pp. 1–6.

Chiu, P.-H. & Hripcsak, G. 2017, 'EHR-based phenotyping: bulk learning and evaluation', *Journal of biomedical informatics*, vol. 70, pp. 35–51.

Cho, H. 2018, 'ASIC-resistance of multi-hash proof-of-work mechanisms for blockchain consensus protocols', *IEEE Access*, vol. 6, pp. 66210–66222.

Chuang, M.-C. & Chen, M. C. 2014, 'An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics', *Expert Systems with Applications*, vol. 41, no. 4, pp. 1411–1418.

City of Melbourne Open Data Team October 19, 2018, *Sensor readings, with temperature, light, humidity every 5 minutes at 8 locations*, viewed 11 April 2019, <https://data.melbourne.vic.gov.au/Environment/Sensor-readings-with-temperature-light-humidity-ev/ez6b-syvw>.

Consultants to Government and Industries 2015, *Cyberprivacy and cybersecurity for health data: building confidence in health systems*, Technical report, CGI GROUP INC., viewed 26 September 2017, <https://www.cgi.com/sites/default/files/>.

Czaja, A. S., Ross, M. E., Liu, W., Fiks, A. G., Localio, R., Wasserman, R. C., Grundmeier, R. W., Adams, W. G. & through Collaborative Electronic Reporting (CER2) Consortium, C. E. R. 2018, 'Electronic health record (EHR) based postmarketing surveillance of adverse events associated with pediatric off-label medication use: a case study of short-acting beta-2 agonists and arrhythmias', *Pharmacoepidemiology and drug safety*, vol. 27, no. 7, pp. 815–822.

Dallaglio, M., Giorgetti, A., Sambo, N., Cugini, F. & Castoldi, P. 2015, 'Provisioning and restoration with sliceability in GMPLS-based elastic optical networks', *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 2, pp. A309–A317.

Danger, J.-L., Guilley, S., Hoogvorst, P., Murdica, C. & Naccache, D. 2013, 'A synthesis of side-channel attacks on elliptic curve cryptography in smart-cards', *Journal of Cryptographic Engineering*, vol. 3, no. 4, pp. 241–265.

Das, A. K., Sutrala, A. K., Odelu, V. & Goswami, A. 2017, 'A secure smartcard-based anonymous user authentication scheme for healthcare applications using wireless medical sensor networks', *Wireless Personal Communications*, vol. 94, no. 3, pp. 1899–1933.

Davis, J. 2017, *EHR server hack threatens data of 14,000 IVF clinic patients*, Technical report, healthcare IT News, viewed 16 March 2019, <http://www.healthcareitnews.com/news/ehr-server-hack-threatens-data-14000-ivf-clinic-patients>.

De Dormale, G. M. & Quisquater, J.-J. 2007, 'High-speed hardware implementations of elliptic curve cryptography: a survey', *Journal of systems architecture*, vol. 53, no. 2, pp. 72–84.

De Meulenaer, G., Gosset, F., Standaert, F.-X. & Pereira, O. 2008, 'On the energy cost of communication and cryptography in wireless sensor networks', *2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, IEEE, pp. 580–585.

Deng, F., Wang, S., Zhang, L., Wei, X. & Yu, J. 2018, 'Establishment of attribute bitmaps for efficient XACML policy evaluation', *Knowledge-Based Systems*, vol. 143, pp. 93–101.

Dhillon, P. K. & Kalra, S. 2018, 'Multi-factor user authentication scheme for IoT-based healthcare services', *Journal of Reliable Intelligent Environments*, vol. 4, no. 3, pp. 141–160.

Di Pietro, R., Guarino, S., Verde, N. V. & Domingo-Ferrer, J. 2014, 'Security in wireless ad-hoc networks–a survey', *Computer Communications*, vol. 51, pp. 1–20.

Dikshit, P. & Singh, K. 2017, 'Efficient weighted threshold ECDSA for securing bitcoin wallet', *Asia Security and Privacy (ISEASP), 2017 ISEA*, *IEEE*, pp. 1–9.

Diro, A. A., Chilamkurti, N. & Kumar, N. 2017, 'Lightweight cybersecurity schemes using elliptic curve cryptography in publish-subscribe fog computing', *Mobile Networks and Applications*, vol. 22, no. 5, pp. 848–858.

Dobraunig, C., Eichlseder, M. & Mendel, F. 2015, 'Analysis of SHA-512/224 and SHA-512/256', *International Conference on the Theory and Application of Cryptology and Information Security*, *Springer*, pp. 612–630.

Dong, Q., Chen, M., Li, L. & Fan, K. 2018, 'Cloud-based radio frequency identification authentication protocol with location privacy protection', *International Journal of Distributed Sensor Networks*, vol. 14, no. 1, pp. 1–12.

Donovan, F. 2018, *417K patients exposed in latest phishing attack at AU health*, Technical report, Health IT Security, viewed 04 June 2018, <https://healthitsecurity.com/news/417k-patients-exposed-in-latest-phishing-attack-at-au-health>.

Dou, Y., Weng, J., Ma, C. & Wei, F. 2017, 'Secure and efficient ECC speeding up algorithms for wireless sensor networks', *Soft Computing*, vol. 21, no. 19, pp. 5665–5673.

Driessen, B., Poschmann, A. & Paar, C. 2008, 'Comparison of innovative signature algorithms for WSNs', *Proceedings of the first ACM conference on Wireless network security*, *ACM*, pp. 30–35.

Duan, L., Zhang, Y., Chen, S., Zhao, S., Wang, S., Liu, D., Liu, R. P., Cheng, B. & Chen, J. 2016, 'Automated policy combination for secure data sharing in cross-organizational collaborations', *IEEE Access*, vol. 4, pp. 3454–3468.

Dwivedi, A. D., Srivastava, G., Dhar, S. & Singh, R. 2019, 'A decentralized privacy-preserving healthcare blockchain for IoT', *Sensors*, vol. 19, no. 2, pp. 1–17.

El Barachi, M. & Alfandi, O. 2013, 'The design and implementation of a wireless healthcare application for WSN-enabled IMS environments', *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, *IEEE*, pp. 892–897.

El-Semary, A. M. & Abdel-Azim, M. M. 2013, 'New trends in secure routing protocols for wireless sensor networks', *International Journal of Distributed Sensor Networks*, vol. 9, no. 5, pp. 1–16.

El-Tawab, S., Yorio, Z., Salman, A., Oram, R. & Park, B. B. 2019, 'Origin-destination tracking analysis of an intelligent transit bus system using internet of things', *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, *IEEE*, pp. 139–144.

Elhoseny, M., Yuan, X., El-Minir, H. K. & Riad, A. M. 2016, 'An energy efficient encryption method for secure dynamic WSN', *Security and Communication Networks*, vol. 9, no. 13, pp. 2024–2031.

Emmanuel, N., Khan, A., Alam, M., Khan, T. & Khan, M. K. 2018, 'Structures and data preserving homomorphic signatures', *Journal of Network and Computer Applications*, vol. 102, pp. 58–70.

Entzeridou, E., Markopoulou, E. & Mollaki, V. 2018, 'Public and physician's expectations and ethical concerns about electronic health record: benefits outweigh risks except for information security', *International journal of medical informatics*, vol. 110, pp. 98–107.

Esiner, E. & Datta, A. 2019, 'Two-factor authentication for trusted third party free dispersed storage', *Future Generation Computer Systems*, vol. 90, pp. 291–306.

Ever, Y. K. 2018, 'Secure-anonymous user authentication scheme for e-healthcare application using wireless medical sensor networks', *IEEE Systems Journal*, vol. 13, no. 1, pp. 456–467.

Fan, J., Guo, X., De Mulder, E., Schaumont, P., Preneel, B. & Verbauwhede, I. 2010, 'State-of-the-art of secure ECC implementations: a survey on known side-channel attacks and countermeasures', *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, *IEEE*, pp. 76–87.

Fan, J. & Verbauwhede, I. 2012, 'An updated survey on secure ECC implementations: attacks, countermeasures and cost', *Cryptography and Security: From Theory to Applications*, *Springer*, pp. 265–282.

Fan, S., Wang, W. & Cheng, Q. 2016, 'Attacking openssl implementation of ECDSA with a few signatures', *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, *ACM*, pp. 1505–1515.

Fan, X. & Gong, G. 2012, 'Accelerating signature-based broadcast authentication for wireless sensor networks', *Ad Hoc Networks*, vol. 10, no. 4, pp. 723–736.

Farash, M. S., Nawaz, O., Mahmood, K., Chaudhry, S. A. & Khan, M. K. 2016, 'A provably secure RFID authentication protocol based on elliptic curve for healthcare environments', *Journal of medical systems*, vol. 40, no. 7, pp. 1–7.

Fernández-Alemán, J. L., Señor, I. C., Lozoya, P. Á. O. & Toval, A. 2013, 'Security and privacy in electronic health records: a systematic literature review', *Journal of biomedical informatics*, vol. 46, no. 3, pp. 541–562.

Franeková, M., Holečko, P., Bubeníková, E. & Kanáliková, A. 2017, 'Transport scenarios analysis within C2C communications focusing on security aspects', *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, *IEEE*, pp. 461–466.

Furtak, J., Zieliński, Z. & Chudzikiewicz, J. 2019, 'A framework for constructing a secure domain of sensor nodes', *Sensors*, vol. 19, no. 12, pp. 1–30.

Gajanayake, R., Iannella, R. & Sahama, T. 2014, 'Privacy oriented access control for electronic health records', *Electronic Journal of Health Informatics*, vol. 8, no. 2, pp. 1–8.

Ganiga, R., Pai, R. M., MM, M. P. & Sinha, R. K. 2018, 'Private cloud solution for securing and managing patient data in rural healthcare system', *Procedia Computer Science*, vol. 135, pp. 688–699.

Gao, Y., Ao, H., Feng, Z., Zhou, W., Hu, S. & Tang, W. 2018, 'Mobile network security and privacy in WSN', *Procedia Computer Science*, vol. 129, pp. 324–330.

García-Holgado, A., Marcos-Pablos, S., Therón-Sánchez, R. & García-Peñalvo, F. J. 2019, 'Technological ecosystems in the health sector: a mapping study of European research projects', *Journal of medical systems*, vol. 43, no. 4, pp. 1–11.

Garrett, B. 2016, *VORMETRIC data threat report*, Technical report, 451 RESEARCH, Thales Group, viewed 15 June 2018, https://www.thalesesecurity.com/.

Giechaskiel, I., Cremers, C. & Rasmussen, K. B. 2018, 'When the crypto in cryptocurrencies breaks: Bitcoin security under broken primitives', *IEEE Security & Privacy*, vol. 16, no. 4, pp. 46–56.

Giri, D., Maitra, T., Amin, R. & Srivastava, P. 2015, 'An efficient and robust RSA-based remote user authentication for telecare medical information systems', *Journal of medical systems*, vol. 39, no. 1, pp. 1–9.

Gold, R., Cottrell, E., Bunce, A., Middendorf, M., Hollombe, C., Cowburn, S., Mahr, P. & Melgar, G. 2017, 'Developing electronic health record (EHR) strategies related to health center patients' social determinants of health', *The Journal of the American Board of Family Medicine*, vol. 30, no. 4, pp. 428–447.

Grace, P. & Surridge, M. 2017, 'Towards a model of user-centered privacy preservation', *Proceedings of the 12th International Conference on Availability, Reliability and Security*, *ACM*, pp. 1–8.

Griggs, K. N., Ossipova, O., Kohlios, C. P., Baccarini, A. N., Howson, E. A. & Hayajneh, T. 2018, 'Healthcare blockchain system using smart contracts for secure automated remote patient monitoring', *Journal of medical systems*, vol. 42, no. 7, pp. 1–7.

Guesmi, R., Farah, M., Kachouri, A. & Samet, M. 2016, 'A novel chaos-based image encryption using DNA sequence operation and secure hash Algorithm SHA-2', *Nonlinear Dynamics*, vol. 83, no. 3, pp. 1123–1136.

Guo, J., Peyrin, T. & Poschmann, A. 2011, 'The PHOTON family of lightweight hash functions', *Annual Cryptology Conference*, *Springer*, pp. 222–239.

Gupta, S., Parne, B. L. & Chaudhari, N. S. 2018, 'An efficient handover AKA protocol for wireless network using Chameleon Hash function', *2018 4th International Conference on Recent Advances in Information Technology (RAIT)*, *IEEE*, pp. 1–7.

Haftu, G. G. 2019, 'Information communications technology and economic growth in Sub-Saharan Africa: a panel data approach', *Telecommunications Policy*, vol. 43, no. 1, pp. 88–99.

Hamidi, H. 2019, ‘An approach to develop the smart health using Internet of things and authentication based on biometric technology’, *Future generation computer systems*, vol. 91, pp. 434–449.

Harkanson, R. & Kim, Y. 2017, ‘Applications of elliptic curve cryptography: a light introduction to elliptic curves and a survey of their applications’, *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, *ACM*, pp. 1–7.

Harman, L. B., Flite, C. A. & Bond, K. 2012, ‘Electronic health records: privacy, confidentiality, and security’, *AMA Journal of Ethics*, vol. 14, no. 9, pp. 712–719.

Harran, M., Farrelly, W. & Curran, K. 2018, ‘A method for verifying integrity & authenticating digital media’, *Applied computing and informatics*, vol. 14, no. 2, pp. 145–158.

Hathaliya, J. J., Tanwar, S., Tyagi, S. & Kumar, N. 2019, ‘Securing electronics healthcare records in Healthcare 4.0: A biometric-based approach’, *Computers & Electrical Engineering*, vol. 76, pp. 398–410.

He, D. & Zeadally, S. 2015, ‘Authentication protocol for an ambient assisted living system’, *IEEE Communications Magazine*, vol. 53, no. 1, pp. 71–77.

Heart, T., Ben-Assuli, O. & Shabtai, I. 2017, ‘A review of PHR, EMR and EHR integration: a more personalized healthcare and public health policy’, *Health Policy and Technology*, vol. 6, no. 1, pp. 20–25.

Heigl, M., Schramm, M. & Fiala, D. 2019, ‘A lightweight quantum-safe security concept for wireless sensor network communication’, *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, *IEEE*, pp. 906–911.

Hoceini, O., Afifi, H. & Aoudjit, R. 2017, ‘Authentication based elliptic curves digital signature for ZigBee networks’, *International Conference on Mobile, Secure, and Programmable Networking*, *Springer*, pp. 63–73.

Homsirikamol, E., Rogawski, M. & Gaj, K. 2011, ‘Comparing hardware performance of round 3 SHA-3 candidates using multiple hardware architectures in Xilinx and Altera FPGAs’, *Ecrypt II Hash Workshop*, *Vol. 2011*, pp. 1–15.

Huang, W., Langberg, M., Kliewer, J. & Bruck, J. 2016, ‘Communication efficient secret sharing’, *IEEE Transactions on Information Theory*, vol. 62, no. 12, pp. 7195–7206.

Ijaz, R. & Pasha, M. A. 2017, ‘Area-efficient and high-throughput hardware implementations of TAV-128 hash function for resource-constrained IoT devices’, *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, Vol. 2, *IEEE*, pp. 832–835.

Ilhan, Ö. M., Thatmann, D. & Küpper, A. 2015, ‘A performance analysis of the xacml decision process and the impact of caching’, *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, *IEEE*, pp. 216–223.

Imran, M., Rashid, M. & Shafi, I. 2018, ‘Lopez Dahab based elliptic crypto processor (ECP) over GF (2^{163}) for low-area applications on FPGA’, *2018 International Conference on Engineering and Emerging Technologies (ICEET)*, *IEEE*, pp. 1–6.

Imran, M., Shafi, I., Jafri, A. R. & Rashid, M. 2017, 'Hardware design and implementation of ECC based crypto processor for low-area-applications on FPGA', *2017 International Conference on Open Source Systems & Technologies (ICOSST)*, *IEEE*, pp. 54–59.

Iqbal, U. & Shafi, S. 2019, 'A provable and secure key exchange protocol based on the elliptical curve Diffe–Hellman for WSN', *Advances in Big Data and Cloud Computing*, *Springer*, pp. 363–372.

Islam, S. H., Amin, R., Biswas, G., Farash, M. S., Li, X. & Kumari, S. 2017, 'An improved three party authenticated key exchange protocol using hash function and elliptic curve cryptography for mobile-commerce environments', *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 3, pp. 311–324.

Jariwala, V. & Jinwala, D. 2012, 'A novel approach for secure data aggregation in wireless sensor networks', *arXiv preprint arXiv:1203.4698*, *Cornell University*.

Javadi, S. S. & Razzaque, M. 2013, 'Security and privacy in wireless body area networks for health care applications', *Wireless Networks and Security*, *Springer*, pp. 165–187.

Jessica Davis 2019, *Breach Tally of Oregon DHS Phishing Attack Reaches 645K Patients*, viewed 10 August 2019, <https://healthitsecurity.com/news/breach-tally-of-oregon-dhs-phishing-attack-reaches-645k-patients>.

Jiang, Q., Khan, M. K., Lu, X., Ma, J. & He, D. 2016, 'A privacy preserving three-factor authentication protocol for e-health clouds', *The Journal of Supercomputing*, vol. 72, no. 10, pp. 3826–3849.

Jin, X., Krishnan, R. & Sandhu, R. 2012, 'A unified attribute-based access control model covering DAC, MAC and RBAC', *IFIP Annual Conference on Data and Applications Security and Privacy*, *Springer*, pp. 41–55.

Jo, S.-M. & Chung, K.-Y. 2015, 'Design of access control system for telemedicine secure XML documents', *Multimedia Tools and Applications*, vol. 74, no. 7, pp. 2257–2271.

Johnson, D., Menezes, A. & Vanstone, S. 2001, 'The elliptic curve digital signature algorithm (ECDSA)', *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63.

Joye, M. 2008, 'Fast point multiplication on elliptic curves without precomputation', *International Workshop on the Arithmetic of Finite Fields*, *Springer*, pp. 36–46.

Kale, S. S. & Bhagwat, D. 2018, 'A Secured IoT based webcare healthcare controlling system using BSN', *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, *IEEE*, pp. 816–821.

Kapusta, K., Memmi, G. & Noura, H. 2019, 'Additively homomorphic encryption and fragmentation scheme for data aggregation inside unattended wireless sensor networks', *Annals of Telecommunications*, pp. 1–9.

Kavun, E. B. & Yalcin, T. 2010, 'A lightweight implementation of Keccak hash function for radio-frequency identification applications', *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, *Springer*, pp. 258–269.

Khatoun, R. & Zeadally, S. 2017, ‘Cybersecurity and privacy solutions in smart cities’, *IEEE Communications Magazine*, vol. 55, no. 3, pp. 51–59.

Kittur, A. S. & Pais, A. R. 2019, ‘A new batch verification scheme for ECDSA* signatures’, *Sādhanā*, vol. 44, no. 7, pp. 1–12.

Knellwolf, S. & Khovratovich, D. 2012, ‘New preimage attacks against reduced SHA-1’, *Advances in Cryptology–Crypto 2012*, *Springer*, pp. 367–383.

Koczkodaj, W. W., Mazurek, M., Strzałka, D., Wolny-Dominiak, A. & Woodbury-Smith, M. 2018, ‘Electronic health record breaches as social indicators’, *Social Indicators Research*, pp. 1–11.

Kodali, R. K. 2013, ‘Implementation of ECDSA in WSN’, *2013 International Conference on Control Communication and Computing (ICCC)*, *IEEE*, pp. 310–314.

Körber, O., Keller, J. & Holmbacka, S. 2018, ‘Energy-efficient execution of cryptographic hash functions on big. LITTLE architecture’, *2018 13th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, *IEEE*, pp. 1–7.

Kuang, B., Fu, A., Yu, S., Yang, G., Su, M. & Zhang, Y. 2019, ‘ESDRA: an efficient and secure distributed remote attestation scheme for IoT swarms’, *IEEE Internet of Things Journal*, pp. 1–12.

Kumar, N., Kaur, K., Misra, S. C. & Iqbal, R. 2016, ‘An intelligent RFID-enabled authentication scheme for healthcare applications in vehicular mobile cloud’, *Peer-to-Peer Networking and Applications*, vol. 9, no. 5, pp. 824–840.

Kumar, P. & Lee, H.-J. 2011, ‘Security issues in healthcare applications using wireless medical sensor networks: a survey’, *Sensors*, vol. 12, no. 1, pp. 55–91.

Kumar, V., Shekhawat, R. S. & Bohra, M. K. 2018, ‘Secure data aggregation in WSNs: A two level framework’, *Proceedings of the 11th International Conference on Security of Information and Networks*, *ACM*, pp. 1–7.

Latinov, L. 2018, *Md5, sha-1, sha-256 and sha-512 speed performance*, viewed 20 February 2019, <https://automationrhapsody.com/md5-sha-1-sha-256-sha-512-speed-performance/>.

Lavanya, M. & Natarajan, V. 2017*a*, ‘Lightweight key agreement protocol for IoT based on IKEv2’, *Computers & Electrical Engineering*, vol. 64, pp. 580–594.

Lavanya, M. & Natarajan, V. 2017*b*, ‘LWDSA: light-weight digital signature algorithm for wireless sensor networks’, *Sādhanā*, pp. 1–15.

Levine, M. E., Albers, D. J. & Hripcsak, G. 2018, ‘Methodological variations in lagged regression for detecting physiologic drug effects in EHR data’, *arXiv preprint arXiv:1801.08929*, *Cornell University*.

Li, J., Zhang, W., Kumari, S., Choo, K.-K. R. & Hogrefe, D. 2018, ‘Security analysis and improvement of a mutual authentication and key agreement solution for wireless sensor networks using chaotic maps’, *Transactions on Emerging Telecommunications Technologies*, vol. 29, no. 6, pp. 1–17.

Li, X., Niu, J., Kumari, S., Liao, J., Liang, W. & Khan, M. K. 2016, 'A new authentication protocol for healthcare applications using wireless medical sensor networks with user anonymity', *Security and Communication Networks*, vol. 9, no. 15, pp. 2643–2655.

Liu, C.-H. & Chung, Y.-F. 2017, 'Secure user authentication scheme for wireless healthcare sensor networks', *Computers & Electrical Engineering*, vol. 59, pp. 250–261.

Liu, X., Xia, Y., Yang, W. & Yang, F. 2018, 'Secure and efficient querying over personal health records in cloud computing', *Neurocomputing*, vol. 274, pp. 99–105.

Liu, Y., Yang, C., Wang, Y., Zhu, L. & Ji, W. 2018, 'Cheating identifiable secret sharing scheme using symmetric bivariate polynomial', *Information Sciences*, vol. 453, pp. 21–29.

Liu, Y., Zhao, Y., Tian, A., Yu, Y. & Du, X. 2019, 'Blockchain based privacy-preserving software updates with proof-of-delivery for Internet of Things', *Journal of Parallel and Distributed Computing*, vol. 132, pp. 141–149.

Liu, Z., Huang, X., Hu, Z., Khan, M. K., Seo, H. & Zhou, L. 2017, 'On emerging family of elliptic curves to secure internet of things: ECC comes of age', *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 3, pp. 237–248.

Lokshina, I. & Lanting, C. 2019, 'A qualitative evaluation of IoT-driven eHealth: knowledge management, business models and opportunities, deployment and evolution', *Data-Centric Business and Applications*, *Springer*, pp. 23–52.

Lu, Y. & Sinnott, R. O. 2018, 'Semantic privacy-preserving framework for electronic health record linkage', *Telematics and Informatics*, vol. 35, no. 4, pp. 737–752.

Lu, Y., Zhai, J., Zhu, R. & Qin, J. 2016, 'Study of wireless authentication center with mixed encryption in WSN', *Journal of Sensors*, vol. 2016, pp. 1–7.

Luo, F., Wang, F., Wang, K. & Chen, K. 2019, 'A more efficient leveled strongly-unforgeable fully homomorphic signature scheme', *Information Sciences*, vol. 480, pp. 70–89.

Luo, P., Athanasiou, K., Fei, Y. & Wahl, T. 2017, 'Algebraic fault analysis of SHA-3', *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, *IEEE*, pp. 151–156.

Luo, P., Li, C. & Fei, Y. 2016, 'Concurrent error detection for reliable SHA-3 design', *Great Lakes Symposium on VLSI, 2016 International*, *IEEE*, pp. 39–44.

Madhavi, A. & Lincke, S. 2018, 'Security risk assessment in electronic health record system', *2018 IEEE Technology and Engineering Management Conference (TEMSCON)*, *IEEE*, pp. 1–4.

Malathy, S., Geetha, J., Suresh, A. & Priya, S. 2018, 'Implementing elliptic curve cryptography with ACO based algorithm in clustered WSN for border surveillance', *2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB)*, *IEEE*, pp. 1–5.

Manogaran, G., Varatharajan, R., Lopez, D., Kumar, P. M., Sundarasekar, R. & Thota, C. 2018, 'A new architecture of Internet of Things and big data ecosystem for secured smart healthcare monitoring and alerting system', *Future Generation Computer Systems*, vol. 82, pp. 375–387.

Marino, F., Moiso, C. & Petracca, M. 2019, 'PKIoT: a public key infrastructure for the Internet of Things', *Transactions on Emerging Telecommunications Technologies*, pp. 1–13.

Martin, J., Mayberry, T., Donahue, C., Foppe, L., Brown, L., Riggins, C., Rye, E. C. & Brown, D. 2017, 'A study of MAC address randomization in mobile devices and when it fails', *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 365–383.

Masoud, M. Z., Jaradat, Y. & Jannoud, I. 2015, 'On preventing ARP poisoning attack utilizing Software Defined Network (SDN) paradigm', *2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, *IEEE*, pp. 1–5.

Mathur, A., Newe, T., Elgenaidi, W., Rao, M., Dooly, G. & Toal, D. 2017, 'A secure end-to-end IoT solution', *Sensors and Actuators A: Physical*, vol. 263, pp. 291–299.

Matte, C., Cunche, M., Rousseau, F. & Vanhoef, M. 2016, 'Defeating MAC address randomization through timing attacks', *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, *ACM*, pp. 15–20.

Mattos, D. M. F. & Duarte, O. C. M. B. 2016, 'AuthFlow: authentication and access control mechanism for software defined networking', *Annals of Telecommunications*, vol. 71, no. 11-12, pp. 607–615.

Mayer, H. 2016, 'ECDSA security in Bitcoin and Ethereum: a research survey', *CoinFabrik*, vol. 28, pp. 1–10.

Meddah, N., Jebrane, A. & Toumanari, A. 2017, 'Scalable Lightweight ABAC Scheme for Secure Sharing PHR in Cloud Computing', *International Conference on Advanced Information Technology, Services and Systems*, *Springer*, pp. 333–346.

Mehmood, A., Natgunanathan, I., Xiang, Y., Poston, H. & Zhang, Y. 2018, 'Anonymous authentication scheme for smart cloud based healthcare applications', *IEEE access*, vol. 6, pp. 33552–33567.

Meri, A., Hasan, M., Danaee, M., Jaber, M., Safei, N., Dauwed, M., Abd, S. K., Al-bsheish, M. et al. 2019, 'Modelling the utilization of cloud health information systems in the Iraqi public healthcare sector', *Telematics and Informatics*, vol. 36, pp. 132–146.

Merrill, N. 2017, 'Better not to know?: the SHA1 collision & the limits of polemic computation', *Proceedings of the 2017 Workshop on Computing Within Limits*, *ACM*, pp. 37–42.

Mi, B., Huang, D. & Wan, S. 2018, 'NTRU implementation of efficient privacy-preserving location-based querying in VANET', *Wireless Communications and Mobile Computing*, vol. 2018, pp. 1–11.

Mozaffari-Kermani, M. & Azarderakhsh, R. 2015, 'Reliable hash trees for post-quantum stateless cryptographic hash-based signatures', *2015 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS), IEEE*, pp. 103–108.

Mozaffari-Kermani, M., Azarderakhsh, R. & Aghaie, A. 2017, 'Fault detection architectures for post-quantum cryptographic stateless hash-based secure signatures benchmarked on ASIC', *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 16, no. 2, pp. 1–19.

Muthee, V., Bochner, A. F., Osterman, A., Liku, N., Akhwale, W., Kwach, J., Prachi, M., Wamicwe, J., Odhiambo, J., Onyango, F. et al. 2018, 'The impact of routine data quality assessments on electronic medical record data quality in Kenya', *PloS one*, vol. 13, no. 4, pp. 1–14.

Nabil, G., Naziha, K., Lamia, F. & Lotfi, K. 2012, 'Hardware implementation of elliptic curve digital signature algorithm (ECDSA) on Koblitz curves', *2012 8th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP), IEEE*, pp. 1–6.

Najaftorkaman, M., Ghapanchi, A. H., Talaei-Khoei, A. & Ray, P. 2015, 'A taxonomy of antecedents to user adoption of health information systems: a synthesis of thirty years of research', *Journal of the Association for Information Science and Technology*, vol. 66, no. 3, pp. 576–598.

Naya-Plasencia, M. & Peyrin, T. 2012, 'Practical cryptanalysis of ARMADILLO2', *Fast Software Encryption*, *Springer*, pp. 146–162.

Neubauer, T. & Heurix, J. 2011, 'A methodology for the pseudonymization of medical data', *International Journal of Medical Informatics*, vol. 80, no. 3, pp. 190–204.

Nizzi, F., Pecorella, T., Esposito, F., Pierucci, L. & Fantacci, R. 2019, 'Iot security via address shuffling: The easy way', *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3764–3774.

Oliveira, T., López, J., Aranha, D. F. & Rodríguez-Henríquez, F. 2014, 'Two is the fastest prime: lambda coordinates for binary elliptic curves', *Journal of Cryptographic Engineering*, vol. 4, no. 1, pp. 3–17.

Oliveira, T., López, J. & Rodríguez-Henríquez, F. 2018, 'The Montgomery ladder on binary elliptic curves', *Journal of Cryptographic Engineering*, vol. 8, no. 3, pp. 241–258.

Osmani, V., Li, L., Danieletto, M., Glicksberg, B., Dudley, J. & Mayora, O. 2018, 'Processing of electronic health records using deep learning: a review', *arXiv preprint arXiv:1804.01758*, *Cornell University*.

Ostad-Sharif, A., Arshad, H., Nikooghadam, M. & Abbasinezhad-Mood, D. 2019, 'Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme', *Future Generation Computer Systems*, vol. 100, pp. 882–892.

Othman, S. B., Alzaid, H., Trad, A. & Youssef, H. 2013, 'An efficient secure data aggregation scheme for wireless sensor networks', *2013 Fourth International Conference on Information, Intelligence, Systems and Applications (IISA), IEEE*, pp. 1–4.

Paganini, P. 2014, *Risks and Cyber Threats to the Healthcare Industry*, Technical report, INFOSEC Institute, viewed 21 December 2018, <http://resources.infosecinstitute.com/risks-cyber-threats-healthcare-industry/#gref>.

Pan, W., Zheng, F., Zhao, Y., Zhu, W.-T. & Jing, J. 2017, 'An efficient elliptic curve cryptography signature server with GPU acceleration', *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 1, pp. 111–122.

Park, S.-y. & Kim, K. 2018, 'A study on the processing and reinforcement of message digest through two-dimensional array masking', *2018 International Conference on Information Networking (ICOIN), IEEE*, pp. 540–544.

Pawar, P. M., Nielsen, R. H., Prasad, N. R. & Prasad, R. 2018, 'GSHMAC: green and secure hybrid medium access control for wireless sensor network', *Wireless Personal Communications*, vol. 100, no. 2, pp. 267–281.

Porambage, P., Braeken, A., Schmitt, C., Gurtov, A., Ylianttila, M. & Stiller, B. 2015, 'Group key establishment for secure multicasting in IoT-enabled Wireless Sensor Networks', *2015 IEEE 40th Conference on Local Computer Networks (LCN), IEEE*, pp. 482–485.

Prakasha, K., Gowda, P., Acharya, V., Muniyal, B. & Khandelwal, M. 2018, 'Enhanced Authentication and Key Agreement Mechanism Using PKI', *International Conference on Applications and Techniques in Information Security, Springer*, pp. 40–51.

proofpoint 2018, *A 12-month analysis of ransomware, email fraud and other healthcare threats—and how you can stop them*, Technical report, NASDAQ:PFPT, viewed 13 July 2018, <https://www.proofpoint.com/sites/default/files/pfpt-us-tr-2018-healthcare-threat-report-181005.pdf>.

Quantin, C., Jaquet-Chiffelle, D.-O., Coatrieux, G., Benzenine, E. & Allaert, F.-A. 2011, 'Medical record search engines, using pseudonymised patient identity: an alternative to centralised medical records', *international journal of medical informatics*, vol. 80, no. 2, pp. e6–e11.

Rafik, M. B. O. & Mohammed, F. 2013, 'The impact of ECC's scalar multiplication on wireless sensor networks', *2013 11th International Symposium on Programming and Systems (ISPS), IEEE*, pp. 17–23.

Rajput, U., Abbas, F. & Oh, H. 2016, 'A hierarchical privacy preserving pseudonymous authentication protocol for VANET', *IEEE Access*, vol. 4, pp. 7770–7784.

Rajput, U., Abbas, F., Wang, J., Eun, H. & Oh, H. 2016, 'CACPPA: a cloud-assisted conditional privacy preserving authentication protocol for VANET', *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), IEEE*, pp. 434–442.

Rantos, K., Fysarakis, K., Manifavas, C. & Askoxylakis, I. G. 2018, 'Policy-controlled authenticated access to LLN-connected healthcare resources', *IEEE Systems Journal*, vol. 12, no. 1, pp. 92–102.

Rasjid, Z. E., Soewito, B., Witjaksono, G. & Abdurachman, E. 2017, 'A review of collisions in cryptographic hash function used in digital forensic tools', *Procedia Computer Science*, vol. 116, pp. 381–392.

Rathert, C., Mittler, J. N., Banerjee, S. & McDaniel, J. 2017, 'Patient-centered communication in the era of electronic health records: what does the evidence say?', *Patient education and counseling*, vol. 100, no. 1, pp. 50–64.

Rezaeibagha, F., Win, K. T. & Susilo, W. 2015, 'A systematic literature review on security and privacy of electronic health record systems: technical perspectives', *Health Information Management Journal*, vol. 44, no. 3, pp. 23–38.

Riedl, B., Grascher, V., Fenz, S. & Neubauer, T. 2008, 'Pseudonymization for improving the privacy in e-health applications', *Hawaii International Conference on System Sciences, Proceedings of the 41st Annual*, *IEEE*, pp. 255–255.

Riedl, B., Grascher, V. & Neubauer, T. 2007, 'Applying a threshold scheme to the pseudonymization of health data', *13th Pacific Rim International Symposium on Dependable Computing (PRDC 2007)*, *IEEE*, pp. 397–400.

Rijmen, V. & Oswald, E. 2005, 'Update on SHA-1', *Cryptographers' Track at the RSA Conference*, *Springer*, pp. 58–71.

Saha, S., Das, R., Datta, S. & Neogy, S. 2016, 'A cloud security framework for a data centric wsn application', *Proceedings of the 17th International Conference on Distributed Computing and Networking*, *ACM*, pp. 1–6.

Sánchez, Y. K. R., Demurjian, S. A. & Baihan, M. S. 2017, 'Achieving RBAC on RESTful APIs for mobile apps using FHIR', *2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*, *IEEE*, pp. 139–144.

Sarkar, B. K. 2017, 'Big data for secure healthcare system: a conceptual design', *Complex & Intelligent Systems*, vol. 3, no. 2, pp. 133–151.

Sartoli, S. & Namin, A. S. 2019, 'Modeling adaptive access control policies using answer set programming', *Journal of Information Security and Applications*, vol. 44, pp. 49–63.

Save, T. & Chhatani, R. 2015, 'Elliptic curve cryptography for EAACK intrusion detection system', *Proceedings of the Sixth International Conference on Computer and Communication Technology 2015*, *ACM*, pp. 355–359.

Senteio, C., Veinot, T., Adler-Milstein, J. & Richardson, C. 2018, 'Physicians' perceptions of the impact of the EHR on the collection and retrieval of psychosocial information in outpatient diabetes care', *International journal of medical informatics*, vol. 113, pp. 9–16.

Seol, K., Kim, Y.-G., Lee, E., Seo, Y.-D. & Baik, D.-K. 2018, 'Privacy-preserving attribute-based access control model for XML-based electronic health record system', *IEEE Access*, vol. 6, pp. 9114–9128.

Sghaier, A., Zeghid, M. & Machhout, M. 2016, 'Fast hardware implementation of ECDSA signature scheme', *International Symposium on Signal, Image, Video and Communications (ISIVC)*, *IEEE*, pp. 343–348.

Shafeeq, S., Alam, M. & Khan, A. 2019, 'Privacy aware decentralized access control system', *Future Generation Computer Systems*, pp. 420–433.

Shankar, S. K., Tomar, A. S. & Tak, G. K. 2015, 'Secure medical data transmission by using ECC with mutual authentication in WSNs', *Procedia Computer Science*, vol. 70, pp. 455–461.

Sharavanan, P., Sridharan, D. & Kumar, R. 2018, 'A privacy preservation secure cross layer protocol design for IoT based wireless body area networks using ECDSA framework', *Journal of medical systems*, vol. 42, no. 10, pp. 1–11.

Shen, J., Gui, Z., Ji, S., Shen, J., Tan, H. & Tang, Y. 2018, 'Cloud-aided lightweight certificateless authentication protocol with anonymity for wireless body area networks', *Journal of Network and Computer Applications*, vol. 106, pp. 117–123.

Shi, Z., Ma, C., Cote, J. & Wang, B. 2012, 'Hardware implementation of hash functions', *Introduction to Hardware Security and Trust*, *Springer*, pp. 27–50.

Shickel, B., Tighe, P. J., Bihorac, A. & Rashidi, P. 2018, 'Deep EHR: a survey of recent advances in deep learning techniques for electronic health record (EHR) analysis', *IEEE journal of biomedical and health informatics*, vol. 22, no. 5, pp. 1589–1604.

Shrestha, N., Alsadoon, A., Prasad, P., Hourany, L. & Elchouemi, A. 2016, 'Enhanced e-health framework for security and privacy in healthcare system', *2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC)*, *IEEE*, pp. 75–79.

Sittig, D. F., Belmont, E. & Singh, H. 2018, 'Improving the safety of health information technology requires shared responsibility: it is time we all step up', *Healthcare*, vol. 6, no. 1, pp. 7–12.

Siwicki, B. 2016, *Brute force ransomware attacks on the rise*, Technical report, healthcare IT News, viewed 03 April 2018, <https://www.healthcareit.com.au/article/brute-force-ransomware-attacks-rise>.

Sojka-Piotrowska, A. & Langendoerfer, P. 2017, 'Shortening the security parameters in lightweight WSN applications for IoT-lessons learned', *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, *IEEE*, pp. 636–641.

Staudemeyer, R. C., Pöhls, H. C. & Wójcik, M. 2018, 'The road to privacy in IoT: beyond encryption and signatures, towards unobservable communication', *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, *IEEE*, pp. 14–20.

Stevens, M. 2013, 'New collision attacks on SHA-1 based on optimal joint local-collision analysis', *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, *Springer*, pp. 245–261.

Stinson, D. R. & Wei, R. 2018, 'Combinatorial repairability for threshold schemes', *Designs, Codes and Cryptography*, vol. 86, no. 1, pp. 195–210.

Suárez-Albela, M., Fraga-Lamas, P., Castedo, L. & Fernández-Caramés, T. 2019, 'Clock frequency impact on the performance of high-security cryptographic cipher suites for energy-efficient resource-constrained iot devices', *Sensors*, vol. 19, no. 1, pp. 1–16.

Sugier, J. 2017, 'Simplifying FPGA implementations of BLAKE hash algorithm with block memory resources', *Procedia Engineering*, vol. 178, pp. 33–41.

Sui, Z. & de Meer, H. 2019, 'BAP: a batch and auditable privacy preservation scheme for demand-response in smart grids', *IEEE Transactions on Industrial Informatics*, pp. 1–13.

Sun, J., Zhu, X., Zhang, C. & Fang, Y. 2011, 'HCPP: cryptography based secure EHR system for patient privacy and emergency healthcare', *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*, *IEEE*, pp. 373–382.

Sun, W., Cai, Z., Li, Y., Liu, F., Fang, S. & Wang, G. 2018, 'Security and privacy in the medical Internet of Things: a review', *Security and Communication Networks*, vol. 2018, pp. 1–9.

Teguig, E., Touati, Y. & Ali-Cherif, A. 2017, 'ECC based-approach for Keys authentication and security in WSN', *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)*, *IEEE*, pp. 1–4.

The AVISPA Team 2006, *AVISPA v1.1 user manual*, viewed 19 June 2017, <http://www.avispa-project.org>.

Thiranant, N., Lee, Y. S. & Lee, H. 2015, 'Performance comparison between RSA and elliptic curve cryptography-based QR code authentication', *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, *IEEE*, pp. 278–282.

Timpner, J., Schürmann, D. & Wolf, L. 2016, 'Trustworthy parking communities: helping your neighbor to find a space', *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 1, pp. 120–132.

Tiwari, H. D. & Kim, J. H. 2018, 'Novel method for DNA-based elliptic curve cryptography for IoT devices', *ETRI Journal*, vol. 40, no. 3, pp. 396–409.

Trakadas, P., Zahariadis, T., Leligou, H., Voliotis, S. & Papadopoulos, K. 2008, 'Analyzing energy and time overhead of security mechanisms in wireless sensor networks', *2008 15th International Conference on Systems, Signals and Image Processing*, *IEEE*, pp. 137–140.

Turkmen, F., den Hartog, J., Ranise, S. & Zannone, N. 2017, 'Formal analysis of XACML policies using SMT', *Computers & Security*, vol. 66, pp. 185–203.

U.S. Department of Health and Human Services 2018, *Breaches affecting 500 or more individuals*, viewed 25 December 2018, <https://ocrportal.hhs.gov/ocr/breach/breach_report.jsf#>.

Vanhoef, M., Matte, C., Cunche, M., Cardoso, L. S. & Piessens, F. 2016, 'Why MAC address randomization is not enough: an analysis of Wi-Fi network discovery mechanisms', *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, *ACM*, pp. 413–424.

Varchola, M., Drutarovsky, M., Repka, M. & Zajac, P. 2015, 'Side channel attack on multiprecision multiplier used in protected ECDSA implementation', *2015 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, *IEEE*, pp. 1–6.

Vatsalan, D., Sehili, Z., Christen, P. & Rahm, E. 2017, 'Privacy-preserving record linkage for big data: current approaches and research challenges', *Handbook of Big Data Technologies*, *Springer*, pp. 851–895.

Verma, G. K., Singh, B. & Singh, H. 2018, 'Bandwidth efficient designated verifier proxy signature scheme for healthcare wireless sensor networks', *Ad Hoc Networks*, vol. 81, pp. 100–108.

Wander, A. S., Gura, N., Eberle, H., Gupta, V. & Shantz, S. C. 2005, 'Energy analysis of public-key cryptography for wireless sensor networks', *Third IEEE international conference on pervasive computing and communications*, *IEEE*, pp. 324–328.

Wang, H. & Li, Q. 2006, 'Efficient implementation of public key cryptosystems on mote sensors (short paper)', *International Conference on Information and Communications Security*, *Springer*, pp. 519–528.

Wang, H., Ning, J., Huang, X., Wei, G., Poh, G. S. & Liu, X. 2019, 'Secure fine-grained encrypted keyword search for e-healthcare cloud', *IEEE Transactions on Dependable and Secure Computing*, pp. 1–13.

Wang, H., Sheng, B. & Li, Q. 2006, 'Elliptic curve cryptography-based access control in sensor networks', *International Journal of Security and Networks*, vol. 1, no. 3-4, pp. 127–137.

Wang, X., Yin, Y. L. & Yu, H. 2005, 'Finding collisions in the full SHA-1', *Annual International Cryptology Conference*, *Springer*, pp. 17–36.

Washington Health Care Authority 2016, *Data breach for Apple Health (Medicaid) managed care plan*, Technical report, Washington State, Health Care Authority, viewed 10 March 2019, <https://www.hca.wa.gov/about-hca/apple-health-medicaid/data-breach-apple-health-medicaid-managed-care-plan-what-clients>.

Wazid, M., Zeadally, S., Das, A. K. & Odelu, V. 2016, 'Analysis of security protocols for mobile healthcare', *Journal of medical systems*, vol. 40, no. 11, pp. 1–10.

WSO2 Team 2017, *WSO2 Identity Server*, Technical report, WSO2 Inc., viewed 18 October 2018, <https://docs.wso2.com/display/IS530/WSO2+Identity+Server+Documentation>.

Xia, W., Jiang, H., Feng, D., Douglis, F., Shilane, P., Hua, Y., Fu, M., Zhang, Y. & Zhou, Y. 2016, 'A comprehensive study of the past, present, and future of data deduplication', *Proceedings of the IEEE*, vol. 104, no. 9, pp. 1681–1710.

Xiao, L., Li, Y., Han, G., Liu, G. & Zhuang, W. 2016, 'PHY-layer spoofing detection with reinforcement learning in wireless networks', *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 10037–10047.

Xu, G., Liu, J., Lu, Y., Zeng, X., Zhang, Y. & Li, X. 2018, 'A novel efficient MAKA protocol with desynchronization for anonymous roaming service in global mobility networks', *Journal of Network and Computer Applications*, vol. 107, pp. 83–92.

Xu, S., Li, C., Li, F. & Zhang, S. 2012, 'An improved sliding window algorithm for ECC multiplication', *World Automation Congress (WAC), 2012*, *IEEE*, pp. 335–338.

Yang, Q., Zhu, X., Fu, H. & Che, X. 2015, 'Survey of security technologies on wireless sensor networks', *Journal of Sensors*, vol. 2015, pp. 1–9.

Yang, Y., Chen, F., Sun, Z., Wang, S., Li, J., Chen, J. & Ming, Z. 2018, 'Secure and efficient parallel hash function construction and its application on cloud audit', *Soft Computing*, pp. 1–19.

Yang, Y., Zhang, X., Yu, J., Zhang, P. et al. 2017, 'Research on the hash function structures and its application', *Wireless Personal Communications*, vol. 94, no. 4, pp. 2969–2985.

Yeh, K.-H. 2016, 'A secure IoT-based healthcare system with body sensor networks', *IEEE Access*, vol. 4, pp. 10288–10299.

Yoshida, H., Watanabe, D., Okeya, K., Kitahara, J., Wu, H., Küçük, Ö. & Preneel, B. 2007, 'MAME: a compression function with reduced hardware requirements', *International Workshop on Cryptographic Hardware and Embedded Systems, Springer*, pp. 148–165.

Yuehong, Y., Zeng, Y., Chen, X. & Fan, Y. 2016, 'The internet of things in healthcare: an overview', *Journal of Industrial Information Integration*, vol. 1, pp. 3–13.

Zhang, G. & Liu, M. 2017, 'A distinguisher on PRESENT-like permutations with application to SPONGENT', *Science China Information Sciences*, vol. 60, no. 7, pp. 1–13.

Zhang, Y., Sun, L., Song, H. & Cao, X. 2014, 'Ubiquitous wsn for healthcare: recent advances and future prospects', *IEEE Internet of Things Journal*, vol. 1, no. 4, pp. 311–318.

Zhang, Y. & Zhang, B. 2017, 'A new testing method for XACML 3.0 policy based on ABAC and data flow', *2017 13th IEEE International Conference on Control & Automation (ICCA), IEEE*, pp. 160–164.

Zhao, Y., Yu, Y., Li, Y., Han, G. & Du, X. 2019, 'Machine learning based privacy-preserving fair data trading in big data market', *Information Sciences*, vol. 478, pp. 449–460.

Zhong, H., Zhao, R., Cui, J., Jiang, X. & Gao, J. 2016, 'An improved ECDSA scheme for wireless sensor network', *International Journal of Future Generation Communication and Networking*, vol. 9, no. 2, pp. 73–82.

Zhou, J., Cao, Z., Dong, X. & Vasilakos, A. V. 2017, 'Security and privacy for cloud-based IoT: challenges', *IEEE Communications Magazine*, vol. 55, no. 1, pp. 26–33.

Appendices

Appendix A: REISCH

A.1 SHA Hash Function

Secure hash algorithm (SHA) is one of the traditional hash algorithms that provide integrity and authentication when used with digital signatures (Guesmi et al. 2016). For example, SHA1 was used in the ECDSA algorithm to perform the signature process in public key cryptography. SHA consists of several varieties SHA0, SHA1, SHA2 and SHA3. The National Security Agency (NSA) replaced SHA0 with SHA1 to improve security by adding rotation to the compression function. Both SHA2 and SHA3 consist of SHA-224, SHA-256, SHA-384 and SHA-512, but SHA3 uses a different structure than the rest of the SHA family. SHA0, SHA1 and SHA2 are built on the basis of the Merkle-Damgard structure as shown in Figure A.1 (Shi et al. 2012) and were designed by the NSA. SHA3 is also known as KECCAK and is built on sponge construction as shown in Figure B.2. It uses two-stage absorbing and squeezing. Since 2007, NIST has adopted KECCAK because of the practical attacks on SHA1 and SHA2. KECCAK became the rival standard in 2015 (Shi et al. 2012). However, some research (Amy et al. 2016, Luo et al. 2016, 2017) have indicated that SHA3 can suffer from fault injection threats.

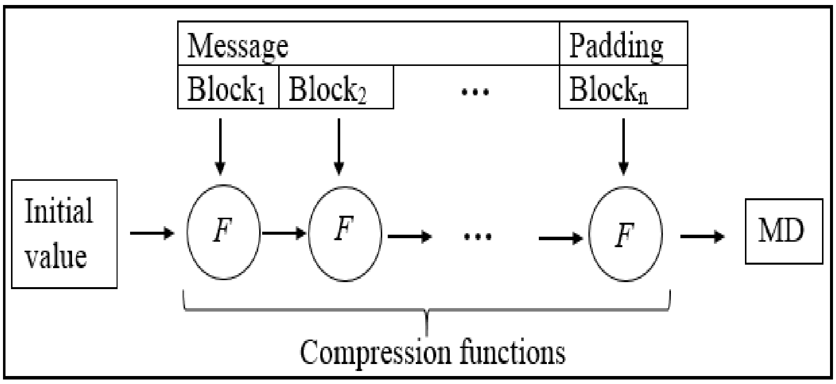


Figure A.1: The Merkle-Damgard construction of SHA (0, 1 and 2) hash functions

SHA is a one-way function consisting of two phases that divide the message into blocks of the same size (such as 512 or 1024). A set of zeros is added and followed

Table A.1: Comparison of SHA family

Algorithm	SHA1	SHA2				SHA3			
MD	160	224	256	384	512	224	256	384	512
Word size	32	32	32	64	64	64	64	64	64
Block size	512	512	512	1024	1024	1152	1088	832	576
Message size	$<2^{64}$	$<2^{64}$	$<2^{64}$	$<2^{128}$	$<2^{128}$	-	-	-	-
Iterations	80	64	64	80	80	24	24	24	24
Security	80	112	128	192	256	112	128	192	256
Weak security	Yes, practical such as Collision and preimage	Yes, practical such as preimage and length extension				Yes, theoretical such as fault injection			
Performance	Fast	Less				Lowest			
Year	1995	2004				2015			
Designer	NSA					Guido Bertoni and et al.			
Construction	Merkle–Damgård					Sponge			

by one at the end of the last block of the message (Kodali 2013). This phase is called preprocessing or padding. The second stage is the MD computation. At this stage, all message blocks are entered into the iterations (SHA1 (80), SHA2 (64) and SHA3 (256)) one by one, containing constants and logic operations (OR, AND, XOR) in compression function (F) to produce MD. Each hash algorithm produces a fixed length of MD, such as 160 for SHA1, 224, 256, 384 and 512 For SHA2 and SHA3 (Boubiche et al. 2016, Chaves et al. 2016). Table A.1 shows comparison of SHA1, SHA2 and SHA3 (Guesmi et al. 2016, Dobraunig et al. 2015). Many existing schemes to employed to collect data in WSN (Kodali 2013, Boubiche et al. 2016, Al Maashri et al. 2016, Lu et al. 2016, Saha et al. 2016, Elhoseny et al. 2016) have used a SHA algorithm to support integrity and authentication. However, these schemes do not address the collision and preimage problems in the SHA algorithm (Xia et al. 2016, Rasjid et al. 2017, Chiriaco et al. 2017, Merrill 2017, Yang et al. 2017, Giechaskiel et al. 2018, Brockmann 2018, Park & Kim 2018).

A.2 BLAKE Hash Function

Aumasson et al. (2008) proposed a BLAKE hash algorithm to overcome efficiency problems in previous hash algorithms. This algorithm offers several features, such as simplicity, speed and parallel operations in hardware and software implementations. It is immune to second preimage, side-channel and length-extension attacks. BLAKE implements HAIFA construction which is an enhanced version of Merkle-Damgård. This development of construction is accomplished by adding a salt and a counter to the algorithm stages to prevent security vulnerability for second preimage attacks in Merkle-Damgård. Also, BLAKE's local wide-pipe structure makes collision attacks impossible (Shi et al. 2012). BLAKE uses the LAKE hash algorithm and compresses the message blocks in hash-tree constructions with Bernstein's stream cipher ChaCha which is a variation of Salsa20-256 (Mozaffari-Kermani & Azarderakhsh 2015). NIST was considered a BLAKE of competing algorithms in

Table A.2: Versions of BLAKE hash function

BLAKE version	Word	Message	Block	MD	Salt	Round
BLAKE-28	32 bits	$<2^{64}$	512	224	128	14
BLAKE-32	32 bits	$<2^{64}$	512	256	128	14
BLAKE-48	64 bits	$<2^{128}$	1024	384	256	16
BLAKE-64	64 bits	$<2^{128}$	1024	512	256	16
BLAKE2s	32 bits	-	256	128-256	64	10
BLAKE2b	64 bits	-	512	160-512	128	12

the final round of hash algorithms, such as KECCAK, Skein and Grøstl (Cho 2018). BLAKE supports several versions: 244, 256, 384 and 512. Then, Aumasson, Neves, Wilcox-O'Hearn & Winnerlein (2013) developed BLAKE2 to improve the speed in software implementation and reduce memory. BLAKE2 has 32% less memory than BLAKE. In addition, BLAKE2 contains two versions, BLAKE2s and BLAKE2b, to be used with 32 bits and 64 bits platform respectively. Table A.2 shows the BLAKE family (Chaves et al. 2016).

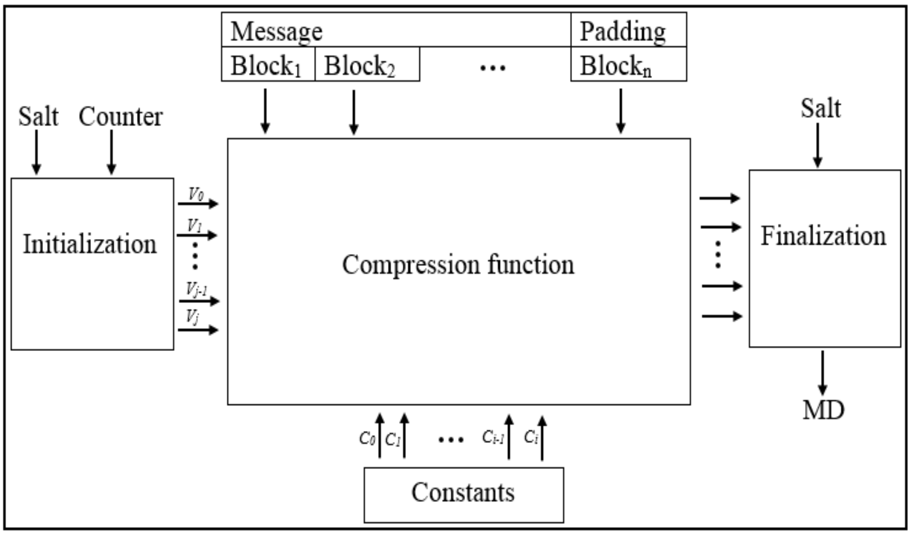


Figure A.2: Architecture of BLAKE hash function

Figure A.2 shows the architecture of the BLAKE hash function. In BLAKE, the message is divided into blocks, and the last block is padded with 1 followed by zeros to complete the last block size to 512 or 1024 bits. BLAKE consists of two parts: the compression function and iteration mode. Compression function consists of chain value, message blocks, salt value and counter value. The BLAKE compression function uses three phases: initialization, round functions and finalization. The initialization phase uses the chain value, salt and counter to create a matrix 4 * 4 and produces a 16-word value for different initializing states (V). These states are entered into the round function (r) with parallel rounds in the round functions phase. The output of this phase is a new V that is used to generate the chain value for the finalization phase. In the finalization phase of the chain, salt and new state values are applied with $\oplus$ operations to produce a new chain value. BLAKE is one of the fastest hash algorithms and has strong security (Chaves et al. 2016, Körber et al. 2018). Much research has pointed out that BLAKE is a conspicuously suitable algorithm for source restricted devices (Mozaffari-Kermani & Azarderakhsh 2015, Homsirikamol et al. 2011, Sugier 2017, Mozaffari-Kermani et al. 2017, Yang et al. 2018, Prakasha et al. 2018).

A.3 ECDSA with BLAKE Hash

The hash algorithm is one of the important processes used by the ECDSA algorithm to complete the signing. The ECDSA algorithm uses a secure hash algorithm (SHA1) provided by NIST in 1995 to preserve the message integrity. This algorithm produces a digest 160-bit size with 6122 gate-equivalent (GE). It needs the complex operations to accomplish message digestion. In digital signatures, the hash algorithm should be collision (Wang et al. 2005), preimage and second preimage resistance (Knellwolf & Khovratovich 2012). Rijmen & Oswald (2005), Wang et al. (2005), Knellwolf & Khovratovich (2012), Stevens (2013) pointed out that this algorithm can suffer from practical attacks such as collision and preimage. But SHA1 is still used in many signature algorithms, such as ECDSA.

We propose using the BLAKE2 hash algorithm for signature in the ECDSA rather than the SHA1. The BLAKE2 algorithm produces digest 256-bit size with 1058 GE. This algorithm was developed in 2013 in order to reduce the memory, energy, speed requirements and resistance to attacks. BLAKE2 is faster and lighter than the SHA family of algorithms and even message digest (MD5), as shown in Figure A.3 (Aumasson, Neves, Wilcox-O'Hearn & Winnerlein 2013). Also, this algorithm is resistant to attacks, such as collisions, multicollisions, distinguishers, preimage and second preimage.

The ECDSA algorithm is used to ensure the integrity of messages. This algorithm prevents the attacker from changing the message data, since any change in the message will be discovered by the receiver. ECDSA produces the signing of the message (r, s). The r parameter is the value used to calculate the signature in the process of signature verification, and s is the signature. The sender signs the message (m) by ECDSA and gets (r, s). The sender sends the message m, r and s to the receiver, who checks the message. During this transfer, r and s can be attacked to extract the private key (K_{pr}). If the attacker is able to get K_{pr}, this attacker can produce the same original signatures. A side channel attack (SCA) can penetrate ECDSA signatures and thus get K_{pr}. Where K_{pr} is the security in this algorithm, if the attacker discovers the private key, data is easily penetrated. These attacks rely on information leaked to the ephemeral key (k) and the value of r is publicly available.

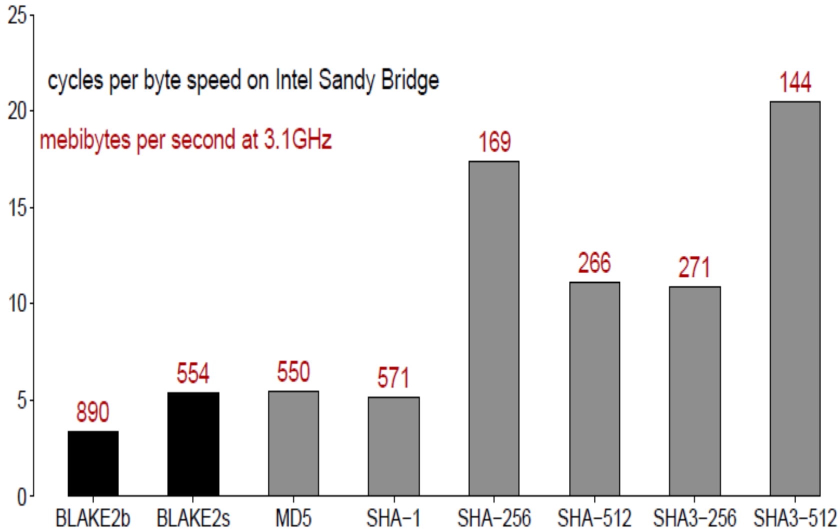


Figure A.3: Comparison of hash functions speed

Appendix B: RAMHU

B.1 Elliptic Curve Integrated Encryption Scheme (ECIES)

This algorithm was independently proposed by Neal Koblitz and Victor Miller in 1985 (Imran et al. 2018). It depends on elliptic curve discrete logarithm problem (ECDLP), ECDLP means difficulty obtaining k from P and Q (where k is the integer, and P and Q are two points on the curve). The small parameters used in ECC help to perform computations quickly. These computations are important in constrained-source and large environments (such as HC systems) that require processing power, memory, bandwidth, or power consumption (Diro et al. 2017). ECC provides encryption (elliptic curve integrated encryption scheme (ECIES)), signature (elliptic curve digital signature algorithm (ECDSA)), and exchange keys (elliptic curve Diffie-Hellman (ECDH)) approaches (Teguig et al. 2017). Many operations are performed in ECC algorithms (described in four layers), as shown in Figure B.1 (Nabil et al. 2012). ECIES has provided confidentiality and integrity, and has proven to be extremely efficient in its performance, as it uses small keys. Thus, the cost of computation is small compared with other public key cryptography algorithms, such as RSA. Table 2.2 in Chapter 2 shows a comparison of key sizes for public key encryption algorithms in addition to some information about these algorithms.

ECC uses two finite fields (prime field and binary field). The binary field uses two types to represent basis (normal and polynomial basis) (Imran et al. 2017), and is well suited to implementation in hardware (Nabil et al. 2012). The prime field is well suited to implementation in software more than hardware, and the prime field is more suited to security operations than the binary field. Let F_q indicate field type, if q=p (where $p > 3$) then ECC uses prime field (F_p). In the second case, if q=2 then ECC uses binary field (F_{2^m}) where m is the prime integer (Oliveira et al. 2018). ECC consists of a set of points (x_i, y_i), where x_i, y_i are integers and the point at infinity (O). It uses O to provide an identity for the Abelian group rule that

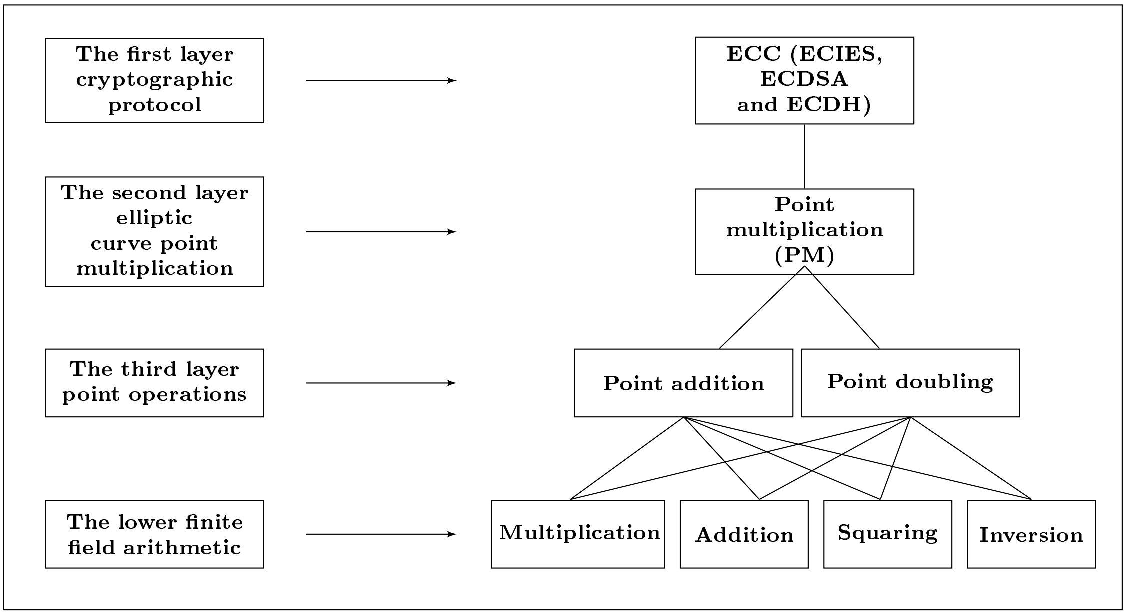


Figure B.1: Arithmetic operations in ECC hierarchy

satisfies long-form for the Weierstrass equation (with Affine coordinate):

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \tag{B.1}$$

When the prime field is used over ECC, the simplified equation is as follows:

$$y^2 = x^3 + ax + b \tag{B.2}$$

Where $a, b \in F_p, 4a^3 + 27b^2 \neq 0(modp)$. The law of chord-and-tangent is used in ECC to add two points on the curve (Pan et al. 2017). Let us suppose that P and Q are two points on the curve; these two points have coordinates $(x_1, y_1), (x_2, y_2)$ respectively and the sum of these two points is equal to a new point $R(x_3, y_3)$ (i.e $P(x_1, y_1)$+$Q(x_2, y_2)$=$R(x_3, y_3)$) (Wang et al. 2006, Islam et al. 2017).

ECC uses two operations for addition that are point addition(P+Q) and point doubling (P+P) (Figure B.1), as in the following equations:

In the case of the point addition $(P + Q)$ where P and $Q \in E(F_p)$:

with using the slope $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$

$$x_3 = \lambda^2 - x_1 - x_2, \; y_3 = \lambda^2(x_1 - x_3) - y_1 \tag{B.3}$$

In the case of the point doubling $(P + P)$ where $P \in E(F_p)$:

with using slope $\lambda = \frac{3x_1^2 + a}{2y_1}$

$$x_3 = \lambda^2 - 2x_1, \; y_3 = \lambda^2(x_1 - x_3) - y_1 \tag{B.4}$$

Algorithm 1 ECC encryption and decryption algorithm:

1: Alice and Bob use same parameters domain $D = \{a, b, q, G, n, h\}$, where a, b are coefficient, q is field type, G is base point, n is order point and h is cofactor.
2: Alice selects random integer AK_{pr} as private key.
3: Alice generates public key $K_{pu} = AK_{pr}G$ and sends K_{pu} and G to Bob.
4: Bob receives message m from Alice, selects random integer as private key BK_{pr} where $BK_{pr} \leq n$.
5: Bob encrypts m with point P in elliptic curve $E(F_q)$.
6: Bob computes $C_1 = P + BK_{pr}K_{pu}$, $C_2 = BK_{pr}G$ and sends C_1 and C_2 to Alice.
7: Alice receives Bob's m and decrypts the m by computing $C_1 - AK_{pr}C_2$ to obtain plaintext.

When the binary field is used over ECC, the simplified equation is as follows:

$$y^2 + xy = x^3 + ax^2 + b \tag{B.5}$$

Where $a, b \in F_{2^m}, b \neq 0$ (Johnson et al. 2001), as addition operations are in the following form:
In the case point addition $(P + Q)$ where P and $Q \in E(F_{2^m})$:
with using the slope $\lambda = \frac{y_1+y_2}{x_1+x_2}$

$$x_3 = \lambda^2 + \lambda + x_1 + x_2 + a, \; y_3 = \lambda(x_1 + x_3) + x_3 + y_1 \tag{B.6}$$

In the case point doubling $(P + P)$ where $P \in E(F_{2^m})$:

$$x_3 = x_1^2 + \frac{b}{x_1^2}, \; y_3 = x_1^2 + (x_1 + \frac{y_1}{x_1})x_3 + x_3 \tag{B.7}$$

ECC operations for encryption and decryption are explained through the following algorithm 1 (Zhong et al. 2016):

B.2 Lightweight Hash-Function Algorithm

The hash function is a method that takes the variable length of data as an input and produces a constant length of the size called message digest (MD). This algorithm is called the one-way algorithm. Namely, when the data is encoded into the MD, the process cannot be reversed to access the original data (Wazid et al. 2016). It is eminently useful in data signature processes, offering high efficiency compared to traditional cryptography algorithms. Recently, lightweight hash algorithms (such as PHOTON, QUARK, and SPONGENT) have emerged to improve efficiency and security rather than traditional hash algorithms (such as MD5, SHA1, and SHA2). Many advantages can be provided in the hash algorithm to satisfy the signature principle, such as first preimage resistant (FPR), second preimage resistant (SPR) and collision resistant (CR) (Giri et al. 2015). Suppose that h is a hash function, MD is a hash result and m is a message, the definition of the signature principles is as follows:

- **FPR**: This principle is also known as one-wayness. When the attacker gets a MD, it is arithmetically difficult to find m and produces the same MD ($h(m)$ = MD) (Harran et al. 2018)

- **SPR**: This principle is also known as weak collision resistant. When the attacker gets MD and m, it is difficult to calculate a different message (m') where $m \neq m'$ to produce the same MD ($h(m) = h(m')$) (Aitzhan & Svetinovic 2018)

- **CR**: This principle is also known as strong collision resistant. It is computationally difficult for the attacker to calculate two different messages (m and m') $m \neq m'$ and to produce the same MD ($h(m) = h(m')$). CR is not the FPR guarantee (Esiner & Datta 2019, Cantu et al. 2017).

The PHOTON is one of the lightweight hash function algorithms. This algorithm is tremendously suitable for projects that require a robust and reliable signature. This algorithm is based on a sponge-like construction and AES-like primitive for domain extension and permutation efficiently (Anandakumar et al. 2014, Porambage et al. 2015, Ijaz & Pasha 2017). PHOTON is available in several versions (80, 128, 160, 224, and 256). It is a balance between the efficiency in the execution of computations on the one hand and security in the implementation of the features of the signature principles (FPR, SPR and CR) on the other hand.

PHOTON algorithm uses sponge construction and applies two phases of absorbing and squeezing as shown in the Figure B.2. The message (m) in the PHOTON is divided into n blocks ($m_0, ..., m_{n-1}$) after adding padding by appending a "1" bit followed many zeros. In the absorbing phase, the internal state (t-bit) is composed of the capacity (c-bit) and the bitrate (r-bit). The t is first initialized with some fixed value. In each iteration, this algorithm achieves the process of changing for t by computing $r + c$ and uses the exclusive-or ($\oplus$) operation between message blocks. After each $\oplus$ operation, the permutation (P) operation of the internal state has been performed. After the end of the absorbing phase, when all the blocks of messages are treated, the squeezing phase begins. At this stage, the input of the squeezing phase is the output of the absorbing phase (internal state and permutation). This phase continues to squeeze the input until the desired MD obtains, the hash output size is 64 $\leq$MD$\leq$ 256. More details are available in (Guo

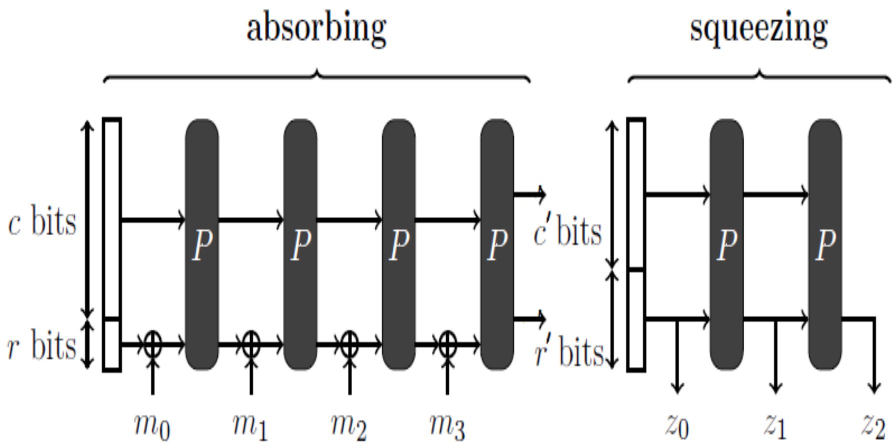


Figure B.2: The PHOTON hash function

et al. 2011).

Table B.1 provides a comparison of the lightweight hash algorithms (the latest version of these algorithms) in terms of security and efficiency (Yoshida et al. 2007, Aumasson, Henzen, Meier & Naya-Plasencia 2013, Kavun & Yalcin 2010, Guo et al. 2011, Bogdanov et al. 2011, Berger et al. 2012, Anandakumar et al. 2014). Standard hash algorithms, however, are still used in applications such as SHA1 (5527 GE, MD 160-bit) and SHA2 (10868 GE, MD 256-bit). Lightweight hash algorithms, such as PHOTON (2177 GE, MD 256-bit), provide the perfect solution for handling complex signatures in large HC systems.

As Table B.1 shows that the PHOTON-256 (2177 GE) offers the best performance compared to all the lightweight hash functions. It also shows that the PHOTON-256 offers a high level of security through the application of signature features FPR (224), SPR (128) and CR (128). Linear and differential attacks are the most powerful attacks in the MD analysis of hash functions. Compared to PHOTON, ARMDILLO and SPONGENT-256 also offer signature features, but both are vulnerable to attacks. ARMIDLLO2 has been attacked by local linearization (practical semi-free-start collision attack) (Naya-Plasencia & Peyrin 2012). All versions of SPONGENT have been attacked by linear distinguishers (23 rounds) (Abdelraheem 2012) and (13 rounds) (Zhang & Liu 2017). Both need the most computations (3281 GE and 8653 GE) compared with PHOTON-256 (2177 GE). PHOTON is a reliable algorithm against linear and differential attacks (Guo et al. 2011). It has a high level of security and efficiency; therefore, it is suitable for our scheme as a signing mechanism in our authentication protocol between parties of the HC applications (clients and servers).

Table B.1: Comparison of lightweight hash function algorithms

Algorithm	Performance		MD	Security			Author (s)	Year
	Gate equivalent area	Throughput (kbps)		FPR	SPR	CR		
SQUASH	2646 GE	0.15	64	64	64	0	Shamir	2005
MAME	8100 GE	146.7	256	-	-	-	Yoshida et al.	2007
C-PRESENT-192	8048 GE	59.26	192	192	192	96	Bogdanov et al.	2008
ARMADILLO2	8653 GE	9.38	256	256	256	128	Bald et al.	2010
S-QUARK	2296 GE	3.13	256	224	112	112	Aumasson et al.	2010
KECCAK-f[400]	5090 GE	14.4	128	128	128	64	Kavun and Yalcin	2010
GLUON	4724 GE	32	224	224	112	112	Berger et al.	2011
SPONGENT-256	3281 GE	11.43	256	240	128	128	Bogdanov et al.	2011
PHOTON-256	2177 GE	0.88	256	224	128	128	Guo et al.	2011

Appendix C: PAX

C.1 Elliptic Curve Digital Signature Algorithm (ECDSA)

Proposed by Scott Vanstone in 1992 (Johnson et al. 2001, Fan et al. 2016), the elliptic curve digital signature algorithm (ECDSA) is an asymmetric signature algorithm. It depends on the use of the points of the curve to integrate and sign data. It has been used to provide integrity, authentication, and non-repudiation properties in the communications network with limited capacity in terms of power and processing. The algorithm depends on the elliptic curve discrete logarithm problem (ECDLP) (Sghaier et al. 2016, Dikshit & Singh 2017). ECDSA uses small parameters which expedite performance of computations, thus reducing time and storage (Sojka-Piotrowska & Langendoerfer 2017). These features are extremely important for large organisations and constrained-source devices, such as WSN, because these networks require processing power, memory, bandwidth and power consumption (Dou et al. 2017).

The data integrity of messages is exceedingly important in networks because attackers can modify messages when they are transferred from source to destination (Bachiller et al. 2018). Many organisations, such as ISO (1998), ANSI (1999) and, IEEE and NIST (2000), use it as a standard (Hoceini et al. 2017). This algorithm is similar to the digital signature algorithm (DSA). Both algorithms depend on the discrete logarithm problem (DPL), but the ECDSA algorithm uses a set of points on the curve and the generating keys are notably small. The ECDSA algorithm with key length 160-bit provides the equivalent of symmetric cryptography with a key length of 80-bit (Driessen et al. 2008, Abueh & Liu 2016). It is significantly convenient for devices with constrained-source because it uses tremendously small keys and provides computation speed in the signature (Cheneau et al. 2010). Moreover, four-point multiplication operations are used in the ECDSA algorithm: one is in public key generation, one for signature generation and two for signature verification (Xu et al. 2012). In addition, this algorithm consists of three procedures: key generation, signature generation, and signature verification. These operations are explained as follows:

- **Key generation:**
  1. Select a random or pseudorandom integer d in the interval $[1, n-1]$.
  2. Compute $K_{pu} = K_{pr}G$
  3. Public key is K_{pu}, private key is K_{pr}.
- **Signature generation:**
  1. Select a random or pseudorandom integer k, $1 \leq k \leq n$-1.
  2. Compute SHA1(m) and convert this bit string to an integer e.
  3. Compute $kG = (x_1, y_1)$ and convert x_1 to an integer $\bar{x}_1$.
  4. Compute $r = x_1 \bmod n$. If $r = 0$ then go to step 1.
  5. Compute $k^{-1} \bmod n$.
  6. Compute s= $k^{-1}(e + K_{pr}r)$mod n. If $s = 0$ then go to step 1.
  7. Signature for the message m is (r, s).
- **Signature verification:**
  1. Verify that r and s are integers in the interval [1,n-1].
  2. Compute SHA1(m) and convert this bit string to an integer e.
  3. Compute $w = s^{-1} \bmod n$.
  4. Compute $u_1 = ew \bmod n$ and $u_2 = rw \bmod n$.
  5. Compute $X = u_1G + u_2K_{pu}$.
  6. If $X = \theta$ then reject the signature. Otherwise, convert the x-coordinate x_1 of X to an integer $\bar{x}_1$, and compute $v = \bar{x}_1 \bmod n$.
  7. Accept the signature if and only if $v = r$.

The ECDSA algorithm is unsuitable for signing messages (integration) if used poorly and incorrectly. Validation of domain parameters is significantly important for ensuring strong security against a variety of attacks. This algorithm becomes strong if the parameters are well validated (Save & Chhatani 2015, Franeková et al. 2017). The authors' recommendations are to update the validation scheme. In our authorisation scheme, we used ECDSA 256-bit to add a high-security level and we took care to consume system resources. Furthermore, it supports the provision of anonymity of users' policies and requests. More details about ECDSA's signature and verification are available in Rafik & Mohammed (2013).

Appendix D: Security Test Tool

D.1 AVISPA

After designing any authentication, authorisation or data collection scheme, this scheme includes a set of protocols and each protocol should be checked and its accuracy verified under a test model, such as the Dolev-Yao to analyse, trace and detect the possibility of attack theoretically. However, this analysis needs to be simulated in a practical manner to detect errors and hidden traces of the protocol's designer, threats tracking, statistics analysing and accurate results, checking several techniques on the same protocol (Al-Zubaidie et al. 2019*a*,*b*, Bojjagani & Sastry 2019, Iqbal & Shafi 2019, Ostad-Sharif et al. 2019). The AVISPA tool provides the features listed above, and offers ease, simplicity, robustness, and applicability for implementing security protocols such as authentication, authorisation and data collection (Bojjagani & Sastry 2017). AVISPA is a formal tool for analysing security schemes and is applied by researchers to evaluate recent security protocols (Gupta et al. 2018, Babu & Padmanabhan 2018, Xu et al. 2018, Dong et al. 2018).

The AVISPA tool is a push-button (as shown in Figure D.1), testing/proofing model and is based on high-level protocol specification language (HLPSL). This language uses directives and expressive terms to represent security procedures. It is integrated with four backends that are the On-the-Fly Model-Checker (OFMC), the Constraint-Logic-based Attack Searcher (CL-AtSe), the SAT-based Model-Checker (SATMC), and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) to perform the simulation in AVISPA. Each backend gives the result of simulation analysis statistics that is different from the other. More details are available in The AVISPA Team (2006). SATMC and TA4SP backends do not work with security protocols that implement the XoR gateway; therefore, we relied on OFMC and CL-AtSe backends to simulate the proposed project's schemes (authentication (RAMHU), authorisation (PAX) and data collection (REISCH)).

- OFMC: It analyses protocols problems and builds infinite tree in demand-driven manner. Also, state-space in this backend has formatted by

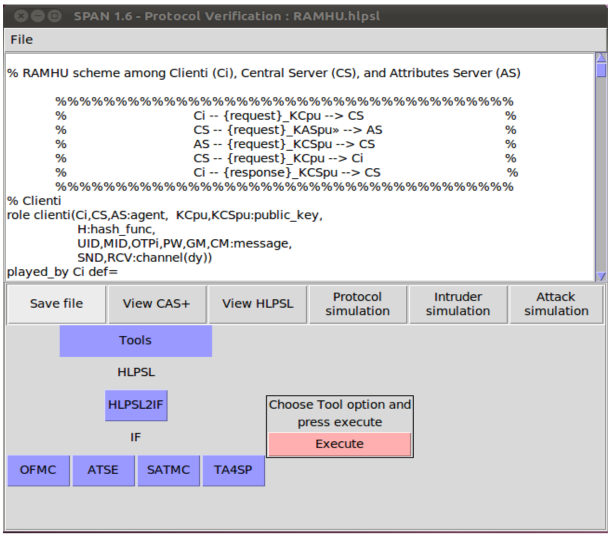


Figure D.1: AVISPA's interface

symbolic techniques. It provides detection of falsification and verification for protocols.

- CL-AtSe: It translates a security protocol specification to a set of constraints. This process allows for the backend to effectively detect attacks on schemes. It provides internal checking and translation.

To implement the security protocols in AVISPA, the protocols should first be written in HLPSL and then converted to the low-level language that is read directly by the backends, which is an intermediate format (IF) by the hlpsl2if compiler, and then converted to an output format (OF) to extract and describe the result of analysis by one of four backends. Figure D.2 shows AVISPA's architecture. The result of the analysis accurately describes whether the protocol is safe or not safe, with some statistical numbers.

HLPSL is modular, and role-oriented. This language allows the completion of security protocol procedures as well as intruder actions. To represent security/privacy scenarios and build simulation structures, HLPSL uses roles, including basic roles such as clients and servers (clienti, centralServer, attributesServer, dataServer, localServer, sensori and clusterHeadi), and composition (session and environment) roles that control the sequence of sending and receiving actions between clients and servers. In addition, communication channels between network entities are governed by the Dolev-Yao (dy) model. Many basic types are used in HLPSL to represent variables, constants, and algorithms (symmetric/asymmetric/hash) such as agent, public_key, message, text, nat, const, and hash_func; in addition to some symbols and terms that have shown in Table D.1. More details are provided in The AVISPA Team (2006). The security

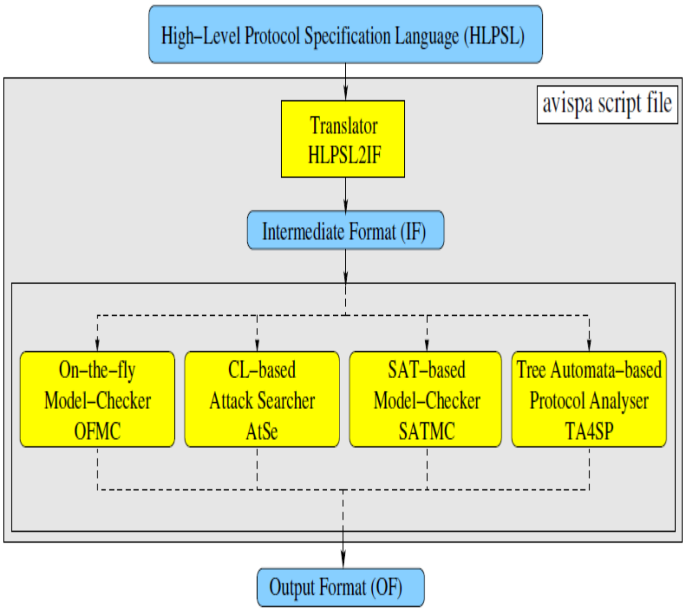


Figure D.2: AVISPA's architecture (The AVISPA Team 2006)

Table D.1: Some HLSPL's symbols and statements

Symbol	Description
init	Initial value for the state
def	Role definition
state	Sequential value for transition
new	Fresh value
start	Beginning signal for role
:=	Assign Mark
xor	Exclusive or operation
.	Concatenation
{}_*Kp*	Asymmetric encryption with public key
{}_*invKp*	Asymmetric signature with public key
played_by	Used to link the role with the intended entity
= | >	Reaction transitions to relate event with act
/\	Conjunction
protocol_id	Goal identifier
secrecy_of	The goal of protecting the secret between entities permanently
authentication_on	The goal of strong authentication between entities
intruder_knowledge	What intruder knows about network

schemes (authentication, authorisation and data collection) in AVISPA depend on security features in the goal specification. Each protocol contains a set of goals (authentication and secret) in authenticating each party with the other. The goals in secrecy_of demonstrate that secrets are not exposed or hacked to non-intended entities, while the goals in authentication_on demonstrate that strong authentication has been applied between entities based on witness and request. Our simulations are based on the AVISPA tool with the current version v.1.1 (13/02/2006) available on the website in The AVISPA Team (2006).